Artificail Intelligence:
Swarm Intelligence
ปัญญาประดิษฐ์: ปัญญาเชิงกลุ่ม

ศ. ดร. บุญเจริญ ศิริเนาวกุล

ปัญญาประดิษฐ์

ปัญญาเชิงกลุ่ม

ศ.ดร.บุญเจริญ ศิริเนาวกุล

คำนำ

หนังสือเล่มนี้เป็นการจัดทำครั้งที่ 2 หลังจากที่การพิมพ์ครั้งแรกได้รับความกรุณาจากบริษัทสำนักพิมพ์ท้อป จำกัด ช่วยในการจัดพิมพ์ออกมาเป็นรูปเล่มที่เป็นกระดาษเมื่อปี 2556 ต้องขอขอบคุณสำนักพิมพ์นี้เป็นอย่างยิ่งที่กรุณาให้ความช่วยเหลือในการพิมพ์ครั้งแรกเป็นอย่างดี

ตามที่ผมได้เคยแจ้งไว้กับสำนักพิมพ์ไว้แล้วว่า หลังจากหมดสัญญากับสำนักพิมพ์ ผมตั้งใจจะเผยแพร่หนังสือเล่มนี้ในฉบับอิเล็กทรอนิกส์แบบ open access แด่ผู้อ่านที่สนใจทุกท่าน

เอกสารฉบับนี้เป็นตำราที่ผมเขียนเพื่อ ขอตำแหน่งศาสตราจารย์ในสาขาวิศวกรรมคอมพิวเตอร์ ปัจจุบันงานวิจัยของผมมีความก้าวหน้าไปมากกว่าเนื้อหาที่ผมเขียนไว้ในตำรามาก แต่เนื่องจากเวลาที่มีอยู่อย่างจำกัด ผมจึงยังไม่ได้ปรับปรุงเนื้อหาของตำราเล่มนี้ เพราะต้องการรีบเผยแพร่

ตำรานี้แม้ว่าจะเขียนมานานแล้ว แต่เนื้อหาทั้งหมดยังทันสมัยอยู่ หลาย ๆ เรื่องที่ผมเรียบเรียงมาจากบทความหลายชิ้นที่ผมส่งตีพิมพ์ในวารสารนานาชาติสมัยนั้น และยังได้รับการอ้างอิงอยู่ในปัจจุบัน โดยเฉพาะจากนักวิจัยระดับแนวหน้าของโลกในด้านนี้ เช่น D Karaboga และ W Gao เป็นต้น ผลการทดลองของอัลกอริธึมที่เขียนอยู่ในตำราเล่มนี้ยังเป็น state of the art ที่ถูกนำไป benchmark กับอัลกอริธึมใหม่ ๆ จนถึงวันนี้

สำหรับการปรับปรุงเนื้อหาของตำราเล่มนี้ ต้องขอเวลาอีกสักหน่อย

ศ.ดร.บุญเจริญ ศิริเนาวกุล
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
14 พย 2562

สารบัญ

บทที่ 1 ปัญญาเชิงกลุ่ม 1

1.1 คำนำ 1

1.2 การหาค่าเหมาะที่สุด 3

1.2.1 การหาค่าเหมาะที่สุดแบบคอมบินาทอเรียล 3

1.2.2 การหาค่าเหมาะที่สุดเชิงตัวเลข 4

1.2.3 ฮิวริสติก 4

1.2.4 เมตาฮิวริสติก 6

1.3 เซลฟ์ออกาไนเซชัน 8

1.3.1 ประวัติการศึกษาเรื่องเซลฟ์ออกาไนเซชัน 8

1.3.2 ลักษณะของเซลฟ์ออกาไนเซชัน 10

1.3.3 อิเมอร์เจนซ์ 16

1.4 ปัญญาเชิงกลุ่ม 18

1.4.1 สติกเมอร์จี 19

1.4.2 หลักการของปัญญาเชิงกลุ่ม 23

1.4.3 การลู่เข้าสู่คำตอบ เอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน 24

1.4.4 ฟังก์ชันเทียบเคียง 26

1.5 สรุป 32

1.6 แบบฝึกหัด 36

บทที่ 2 การค้นหาแบบฮิวริสติก 39

2.1 คำนำ 39

2.2 การค้นหาแบบกำหนดทิศทาง 39
2.2.1 การค้นหาแบบลึกก่อน 40
2.2.2 การค้นหาแบบกว้างก่อน 43
2.3 การค้นหาแบบฮิวริสติก 44
2.3.1 เจเนอเรตแอนด์เทสต์ 46
2.3.2 อัลกอริธึมกรีดี้ 50
2.3.3 การค้นหาแบบที่ดีที่สุดก่อน 55
2.3.4 บรานช์แอนด์เบานด์ 57
2.3.5 การค้นหาแบบฮิลไคลมิ่ง 62
2.3.6 อัลกอริธึมของฮิลไคลมิ่งธรรมดา 62
2.3.7 อัลกอริธึมของสตีเพสต์-แอสเซนต์ฮิลไคลมิ่ง 65
2.3.8 อัลกอริธึมเอ-สตาร์ 68
2.3.9 มีน-เอ็นด์อะแนไลซิส 72
2.3.10 การเล่นเกม 76
2.4 สรุป 83
2.5 แบบฝึกหัด 87
บทที่ 3 ซีมูเลตเต็ดอันนีลลิ่ง 93
3.1 คำนำ 93
3.2 การหาค่าเหมาะที่สุดด้วยซีมูเลตเต็ดอันนีลลิ่ง 96
3.2.1 อัลกอริธึมซีมูเลตเต็ดอันนีลลิ่ง 99
3.2.2 ขั้นตอนการวนรอบ 100
3.2.3 คำตอบข้างเคียง 100
3.3 การปรับสมรรถนะ 101
3.3.1 ตารางการอบอ่อน 101

3.3.2 การลู่เข้าสู่คำตอบ 102

3.4 ตัวอย่างการหาค่าเหมาะที่สุดด้วยซีมูเลตเต็ดอันนีลลิ่ง 105

3.4.1 การเดินทางของเซลส์แมน 105

3.4.2 การจัดวางแพทเทิร์น 108

3.5 การทำงานขั้นสูงของอัลกอริธึม 113

3.5.1 ซีมูเลตเต็ดอันนีลลิ่งสำหรับตัวแปรที่มีค่าต่อเนื่อง 113

3.5.2 ซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัว 118

3.5.3 การทดสอบสมรรถนะ 121

3.6 สรุป 122

3.7 แบบฝึกหัด 125

บทที่ 4 อัลกอริธึมพันธุการ 127

4.1 คำนำ 127

4.2 หลักการของอัลกอริธึมพันธุการ 127

4.2.1 อัลกอริธึมพันธุการ 130

4.2.2 ตัวดำเนินการของอัลกอริธึม 130

4.2.3 วิธีการต่างๆ ของอัลกอริธึมพันธุการ 131

4.3 การเลือกสรรโครโมโซม 133

4.3.1 การเลือกสรรแบบวงล้อรูเล็ตต์ 133

4.3.2 การเลือกสรรแบบจัดลำดับ 135

4.3.3 การเลือกสรรแบบอื่น 136

4.4 การเข้ารหัสโครโมโซมและการแลกเปลี่ยนยีน 137

4.4.1 การเข้ารหัสแบบไบนารี 137

4.4.2 การเข้ารหัสแบบเปอร์มิวเตชัน 139

4.4.3 การเข้ารหัสแบบค่า 141
4.4.4 การเข้ารหัสแบบโครงสร้างต้นไม้ 141
4.5 ตัวอย่างการหาค่าเหมาะที่สุดด้วยอัลกอริธึมพันธุการ 143
4.5.1 การหาค่าสูงสุดของเลขไบนารี 143
4.5.2 การเดินทางของเซลส์แมน 146
4.5.3 การวางผังงานโดยอาศัยอัลกอริธึมพันธุการ 150
4.6 อัลกอริธึมพันธุการขั้นสูง 158
4.6.1 การทำงานแบบแบ่งกลุ่มประชากร 158
4.6.2 การครอสโอเวอร์แบบหลายพ่อแม่พันธุ์ 160
4.6.3 การเข้ารหัสแบบค่าจำนวนจริง 164
4.7 การปรับสมรรถนะ 167
4.7.1 พารามิเตอร์แบบสุ่ม 167
4.7.2 การลู่เข้าสู่คำตอบ 168
4.7.3 การทดสอบกับฟังก์ชันเทียบเคียง 170
4.8 สรุป 171
4.9 แบบฝึกหัด 174
บทที่ 5 การหาค่าเหมาะที่สุดแบบฝูงมด 179
5.1 คำนำ 179
5.1.1 พฤติกรรมของแมลง 179
5.1.2 การทดลองกับสะพานคู่ 181
5.2 การหาค่าเหมาะที่สุดแบบฝูงมด 183
5.2.1 เมตาฮิวริสติกของการหาค่าที่เหมาะที่สุดแบบฝูงมด 185
5.2.2 ขั้นตอนการหาค่าเหมาะสุดแบบฝูงมด 187
5.2.3 การลู่เข้าหาคำตอบ 189

5.3 ตัวอย่างการหาค่าเหมาะที่สุดแบบฝูงมด 192
5.3.1 การเดินทางของเซลส์แมน 192
5.3.2 การหาเส้นทางเดินรถ 203
5.3.3 การวางผังโรงงาน 206
5.4 ความหลากหลายของวิธีการหาค่าเหมาะที่สุดแบบฝูงมด 211
5.4.1 ระบบมด 211
5.4.2 ระบบมดอีลีท 212
5.4.3 ระบบมดแบบการจัดอันดับ 213
5.4.4 ระบบมดแบบสูงสุดต่ำสุด 214
5.4.5 ระบบฝูงมด 215
5.4.6 การหาค่าเหมาะที่สุดสำหรับตัวแปรที่มีค่าจำนวนจริง 216
5.4.7 การทดสอบสมรรถนะ 219
5.5 สรุป 220
5.6 แบบฝึกหัด 224
บทที่ 6 การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค 229
6.1 คำนำ 229
6.2 การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค 232
6.2.1 อัลกอริธึมของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค 233
6.2.2 การทำงานของการหาค่าเหมาะที่สุดแบบฝูงอนุภาค 235
6.2.3 พารามิเตอร์ของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค 237
6.2.4 ขั้นตอนการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค 238
6.2.5 การลู่เข้าสู่คำตอบ 240
6.2.6 การทดสอบสมรรถนะ 242
6.3 ความหลากหลายของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค 244

6.3.1 น้ำหนักอิเนอร์เทีย 244
6.3.2 สัมประสิทธิ์การหดตัว 245
6.3.3 โทโพโลยีของค่าข้างเคียง 246
6.3.4 การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคแบบคอมบินาโทเรียว 247

6.4 ตัวอย่างการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค 250
6.4.1 การหาค่าเหมาะที่สุดของฟังก์ชัน $f(x)=3+x_0^2+x_1^2$ 250
6.4.2 การเดินทางของเซลส์แมน 254
6.4.3 การตัดสต็อกแบบสองมิติ 258

6.5 สรุป 263

6.6 แบบฝึกหัด 265

บทที่ 7 การหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ 269

7.1 คำนำ 269

7.2 การหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ 272
7.2.1 การปรับปรุงคำตอบที่เป็นไปได้ของผึ้งเอ็มพลอย 274
7.2.2 การเลือกคำตอบที่เป็นไปได้ของผึ้งรับสาร 274
7.2.3 การปรับปรุงคำตอบที่เป็นไปได้ของผึ้งรับสาร 275
7.2.4 การหลีกเลี่ยงคำตอบที่ลู่เข้าสู่ทางตันของผึ้งค้นหา 275
7.2.5 อัลกอริธึมของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ 276

7.3 ความหลากหลายของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ 276
7.3.1 เบสโซฟา เอบีซี 276
7.3.2 การปรับปรุงการทำงานของเอบีซีแบบอื่น 280
7.3.3 การลู่เข้าสู่คำตอบและการทดสอบสมรรถนะ 281

7.4 ตัวอย่างการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ 285
7.4.1 การหาค่าเหมาะที่สุดของ ฟังก์ชัน $f(x) = x_1^2 + x_2^2$ 286

7.4.2 การเดินทางของเซลส์แมน....292
7.4.3 การจัดตารางการผลิตด้วยเบสโซฟา เอบีซี....296
7.5 สรุป....307
7.6 แบบฝึกหัด....311
บทที่ 8 ภูมิคุ้มกันประดิษฐ์....315
8.1 คำนำ....315
8.2 ระบบภูมิคุ้มกันทางชีวภาพ....316
8.2.1 เซลล์ที่ทำหน้าที่ภูมิคุ้มกัน....316
8.2.2 กลไกของระบบภูมิคุ้มกัน....317
8.2.3 การทำงานของระบบภูมิคุ้มกัน....318
8.3 หลักการของการเลือกโคลนอล....322
8.3.1 อัลกอริธึมการเลือกโคลนอล....325
8.3.2 การเลือกโคลนอลสำหรับแก้ปัญหาการเดินทางของเซลส์แมน....327
8.4 อัลกอริธึมการเลือกเชิงลบ....329
8.4.1 อัลกอริธึมการเลือกเชิงลบ....330
8.4.2 การประยุกต์ใช้งาน....334
8.5 โครงข่ายภูมิคุ้มกัน....337
8.5.1 การทำงานของโครงข่ายภูมิคุ้มกัน....338
8.5.2 โครงข่ายภูมิคุ้มกันประดิษฐ์สำหรับแก้ปัญหาการเดินทางของเซลส์แมน. 340
8.5.3 โครงข่ายภูมิคุ้มกันประดิษฐ์สำหรับการวิเคราะห์ข้อมูล....343
8.6 สรุป....347
8.6.1 การเข้ารหัส....347
8.6.2 การวัดความเหมือนและแอฟฟินิตี....348

8.6.3 การเลือกแบบโคลนอล และการเลือกเชิงลบ 350
8.6.4 โซมาติกไฮเปอร์มิวเตชัน(Somatic Hypermutation)..... 352
8.7 แบบฝึกหัด 353
เอกสารอ้างอิง..... 356
ดัชนีคำศัพท์..... 378

บทที่ 1 ปัญญาเชิงกลุ่ม

Swarm Intelligence

1.1 คำนำ

แม้ว่าการศึกษาในเรื่องปัญญาเชิงกลุ่ม (Swarm Intelligence) จะมีมานานกว่าร้อยปีแล้วก็ตาม แต่การประยุกต์ใช้ศาสตร์ทางด้านนี้เพื่อนำมาแก้ปัญหาทางวิศวกรรมนับว่าเป็นเรื่องใหม่มาก โดยเฉพาะอย่างยิ่งการนำมาประยุกต์ใช้ในการแก้ปัญหาที่อาศัยคอมพิวเตอร์เป็นเครื่องมือ วิธีการปัญญาเชิงกลุ่มจะนำมาใช้กับ***การหาค่าเหมาะที่สุด*** (Optimization) เป็นส่วนใหญ่ โดยเฉพาะการหาค่าเหมาะที่สุดของปัญหาที่แก้ได้ยาก

ในยุคสงครามโลกครั้งที่สอง การหาคำตอบของปัญหาหลายเรื่องทางด้านการทหาร เป็นการทำเพื่อเอาชนะกันในการสู้รบ ปัญหาเหล่านี้เช่น การหาระบบการส่งกำลังบำรุงที่มีประสิทธิภาพที่สุด การสร้างระบบคงคลังให้มีขนาดที่เหมาะสมสำหรับเก็บเสบียง และการจัดตารางแผนปฏิบัติการสู้รบ เป็นต้น ปัญหาดังที่กล่าวมาต้องการวิธีการแก้ปัญหาลักษณะใหม่ ที่แตกต่างจากการคำนวณในยุคก่อนหน้านั้น เพราะการหาคำตอบของปัญหานั้น ไม่ใช่การหาเพียงคำตอบที่ใช้สำหรับการแก้ปัญหานั้นเท่านั้น แต่เป็นการหาคำตอบที่ดีที่สุดของปัญหานั้นด้วย การหาคำตอบเช่นนี้จะหมายถึงการหาคำตอบที่ระบุถึงคุณภาพของการแก้ปัญหา เช่น แผนการส่งกำลังบำรุงที่มีประสิทธิภาพที่สุด ขนาดของคงคลังที่เหมาะที่สุด และตารางการปฏิบัติการที่สั้นที่สุด เป็นต้น เทคนิคทางคณิตศาสตร์ใหม่ๆ เช่น การวิจัยดำเนินงาน (Operation Research) และโปรแกรมเชิงเส้น (Linear Programming) จึงถูกคิดค้นขึ้นเพื่อนำมาใช้ในการหาค่าที่ดีที่สุด หลังจากการสิ้นสุดของสงคราม การพัฒนาทางด้านอุตสาหกรรม การพานิชย์ การเดินทางขนส่ง และการสื่อสารโทรคมนาคม ได้เจริญเติบโตขึ้นอย่างรวดเร็ว ความพยายามที่จะหาคำตอบเชิงคุณภาพของปัญหาที่มีความซับซ้อน เช่น ระยะทางที่สั้นที่สุดของการขนส่ง ระบบคงคลังสิ้น

ค้าที่สิ้นเปลืองน้อยที่สุด การใช้พื้นที่การทำงานให้เกิดประโยชน์สูงสุด และการจัดตารางการทำงานของเครื่องจักรที่มีประสิทธิภาพสูงสุด เริ่มเป็นที่สนใจมากขึ้น

ปัญหาเช่นนี้มีลักษณะพิเศษ ความซับซ้อนของปัญหาจะขึ้นอยู่กับจำนวน (n) เช่น ปัญหาการเดินทางของเซลส์แมน การหาระยะทางที่สั้นที่สุด ถ้าจำนวนเมือง (n) ที่ผู้เดินทางจะต้องเดินทางผ่านมีมาก การหาระยะทางที่สั้นที่สุดก็จะยากขึ้น การหาตารางการทำงานของเครื่องจักรก็เช่นกัน ถ้าจำนวนเครื่องจักร (n1) มีมาก และมีจำนวนของลักษณะงาน (n2) ที่ต้องทำมาก การหาตารางที่ดีที่สุดก็จะยากขึ้นด้วย นั่นหมายความว่า เมื่อจำนวน (n) มากขึ้น จำนวนคำตอบที่เป็นไปได้ก็จะมากขึ้นเป็นทวีคูณ การหาค่าที่เหมาะที่สุดก็จะซับซ้อนขึ้น ในแทบทุกกรณี จำนวนคำตอบที่เป็นไปได้จะมากจนไม่สามารถใช้วิธีการทางคณิตศาสตร์ธรรมดามาแก้ปัญหาได้ วิธีการที่เรียกว่าฮิวริสติก (Heuristic) จึงถูกนำมาใช้ และมีการพัฒนาขึ้นเป็นลำดับ ฮิวริสติกเป็นวิธีการประมาณการอย่างมีหลักการ เช่น ในการหาเส้นทางที่สั้นที่สุดของเซลส์แมนที่จะต้องเดินผ่านเมืองจำนวนมาก มีเส้นทางที่เป็นไปได้สำหรับการเดินทางให้ครบทุกเมืองจำนวนมาก การนำเส้นทางทั้งหมดมาเปรียบเทียบกันจึงเป็นไปไม่ได้ วิธีการแบบฮิวริสติกคือ การให้เซลส์แมนเริ่มต้นที่จากเมืองใดเมืองหนึ่ง จากนั้นให้เซลส์แมนเดินทางไปยังเมืองถัดไปที่มีระยะทางสั้นที่สุด ทำเช่นนี้เรื่อยๆ จนครบทุกเมืองเราก็จะได้เส้นทางที่สั้น แต่วิธีการแบบฮิวริสติกก็ไม่รับประกันว่าจะได้คำตอบที่ดีที่สุด เมื่อวิธีการทางด้านฮิวริสติกมีข้อจำกัด วิธีการใหม่ที่เรียกว่าเมตาฮิวริสติกก็ถูกคิดค้นขึ้นอีก

วิธีการของเมตาฮิวริสติดมักจะพัฒนามาจากปรากฏการณ์ทางธรรมชาติ เทคนิคทางธรรมชาติที่สำคัญอย่างหนึ่งคือ เซลฟ์ออกาไนเซชัน (Self Organization) ซึ่งพัฒนามาจากพฤติกรรมของสัตว์สังคมในการแก้ปัญหาและการดำเนินชีวิต เช่น พฤติกรรมการสร้างรังของปลวก การหาเส้นทางที่สั้นที่สุดของมด การรักษาระยะห่างของนกในระหว่างการบิน และการหาอาหารของผึ้ง พฤติกรรมของแมลงและสัตว์เหล่านี้เป็นพฤติกรรมแบบที่เรียกว่า เซลฟ์ออกาไนเซชัน ที่เป็นแรงดลใจของการสร้างแบบจำลอง เพื่อใช้สำหรับการแก้ปัญหาการหาค่าเหมาะที่สุด ที่เรียกว่าปัญญาเชิงกลุ่ม (Swarm Intelligence) (Bonabeau, 2000)

1.2 การหาค่าเหมาะที่สุด

ปัญหาการหาค่าเหมาะที่สุด (Optimization) คือ ปัญหาที่เกี่ยวกับการหาค่าที่ดีที่สุดเท่าที่เป็นไปได้จากคำตอบที่เป็นไปได้ (Possible Solution) ที่มีจำนวนมาก ลักษณะของปัญหาที่ต้องอาศัยวิธีการหาค่าเหมาะที่สุด ต้องเป็นปัญหาที่ยากและมีความซับซ้อนมาก ปัญหาเหล่านี้ถูกจำแนกประเภทด้วยการพิจารณาจากชนิดของตัวแปรของปัญหานั้นว่า ค่าของตัวแปรเป็นแบบไม่ต่อเนื่องกัน (Discrete) หรือเป็นแบบต่อเนื่องกัน (Continuous) การหาค่าเหมาะที่สุดของปัญหาที่ค่าของตัวแปรเป็นแบบไม่ต่อเนื่องกันเรียกว่า การหาค่าเหมาะที่สุดแบบคอมบินาทอเรียล (Combinatorial Optimization) สำหรับการหาค่าเหมาะที่สุดของปัญหาที่ค่าของตัวแปรเป็นแบบต่อเนื่องกันจะเรียกว่า การหาค่าเหมาะที่สุดเชิงตัวเลข (Numerical Optimization)

1.2.1 การหาค่าเหมาะที่สุดแบบคอมบินาทอเรียล

การหาค่าเหมาะที่สุดแบบคอมบินาทอเรียล ไม่ใช่การแก้ปัญหาเชิงตัวเลข แต่เป็นการแก้ปัญหาที่มีลักษณะเชิงวัตถุประสงค์ (Objective) ที่มีปริมาณเป็นจำนวนเต็ม ด้วยการเปรียบเทียบแบบเปอร์มิวเตชัน จากเซทที่มีจำนวนจำกัด (Finite Set) เซทหนึ่ง ที่ประกอบด้วยคำตอบที่เป็นไปได้ของปัญหานั้นชุดหนึ่ง คำตอบที่เป็นไปได้เหล่านี้ไม่จำเป็นที่จะต้องเป็นคำตอบที่เหมาะที่สุดของปัญหา คำตอบที่เป็นไปได้นี้ เป็นเพียงชุดของคำตอบทั้งหมดที่มีความสอดคล้องกับเงื่อนไขของปัญหาเชิงวัตถุประสงค์นั้น ทำให้คำตอบที่เป็นไปได้ทั้งหมดของปัญหาหนึ่งๆ มีจำนวนมหาศาล จำนวนคำตอบที่เป็นไปได้ทั้งหมดนี้รวมกันเรียกว่าปริภูมิของการค้นหา (Search Space)

ตัวอย่างของปัญหาแบบคอมบินาทอเรียล ได้แก่ การเดินทางของเซลส์แมน (Travelling Salesman Problem: TSP) การจัดตาราง (Scheduling Problems) ปัญหาเหล่านี้จัดว่าเป็นปัญหาที่ยากและมีความซับซ้อนมาก วิธีสำหรับการแก้ปัญหาดังกล่าวสามารถจำแนกได้เป็นการแก้ปัญหาแบบสมบูรณ์ (Complete) และการแก้ด้วยการประมาณการ (Approximate) สำหรับการ**แก้ปัญหาแบบสมบูรณ์**นั้นเป็นการหาคำตอบที่จะได้ค่าที่ดีที่สุดแน่นอนจากคำตอบที่เป็นไปได้ทั้งหมดด้วยการเปรียบเทียบ การแก้ปัญหาแบบนี้จะต้องมีขอบเขตปัญหาที่มีขนาดเล็ก เช่น การหาเส้นทางที่สั้นที่สุดของการเดินทางไปยังเมือง 4 เมือง ที่มีเส้นทางเชื่อมโยงกันทุกเมือง แต่ละเมืองที่เชื่อมกันมีเพียงเส้นทางเดียว และการเดินทางจะไม่มีการเดินเส้นทางที่ซ้ำกัน ในกรณีนี้ เส้นทางที่เป็นไปได้มีจำนวนเท่ากับ 3!

หรือ (n-1)! ซึ่งเป็นจำนวนที่ไม่มาก การเปรียบเทียบแบบทุกเส้นจึงเป็นไปได้ แต่ถ้าเป็นปัญหาของการเดินทางในกรณีเดียวกัน แต่จำนวนเมืองเท่ากับ 100 เมือง เส้นทางที่เป็นไปได้ทั้งหมดเท่ากับ 99! การเปรียบเทียบเส้นทางทั้งหมดด้วยการแก้ปัญหาแบบสมบูรณ์อาจจะใช้เวลานานมาก ในกรณีการแก้ปัญหาแบบนี้จึงต้องอาศัย**การประมาณการ** สำหรับปัญหาการเดินทางในลักษณะนี้จะมีชื่อที่เรียกกันทั่วไปว่า **การเดินทางของเซลส์แมน**

1.2.2 การหาค่าเหมาะที่สุดเชิงตัวเลข

การหาค่าเหมาะที่สุดเชิงตัวเลข คือปัญหาที่ต้องการหาค่าที่เป็นตัวเลขเพื่อเป็นคำตอบของสมการที่เป็นความสัมพันธ์ของตัวแปรของปัญหาหนึ่งๆ การแก้ปัญหาอาจจะหาได้ด้วยวิธีการประมาณการ เพราะหากต้องแก้สมการด้วยมือหรือใช้กรรมวิธีอื่นอาจใช้เวลานานมาก หรืออาจจะแก้ปัญหานั้นไม่ได้เลย

ตัวอย่างของการหาค่าเหมาะที่สุดเชิงตัวเลข เช่น การหาค่าของตัวแปร x จากสมการ $minf(x) = x^2$ ซึ่งเป็นการหาค่าของ x ที่จะทำให้ฟังก์ชัน $f(x)$ มีค่าน้อยที่สุด ในกรณีนี้เป็นโจทย์ที่ค่อนข้างง่าย เมื่อเราพิจารณาจากสมการเราก็พอที่จะทราบได้ว่าค่าของ x จะเท่ากับ 0 แต่ถ้าใช้คอมพิวเตอร์หาค่าของ x โดยที่คอมพิวเตอร์ไม่ทราบวิธีแก้สมการ วิธีที่คอมพิวเตอร์ใช้ในการแก้ปัญหานี้ก็คือ การสุ่มค่าของ x ออกมา แล้วแทนค่าลงในสมการเพื่อหาค่าของ $f(x)$ ถ้าหาก เรากำหนดขอบเขตของค่า x ให้เป็นจำนวนเต็มที่มีค่าตั้งแต่ -3 ถึง 3 ซึ่งเป็นขอบเขตที่ห่างจากค่าที่ดีที่สุดไม่มาก การหาค่าของ x ก็คงไม่ยากนัก หรือเราอาจจะใช้วิธีการแก้ปัญหาแบบสมบูรณ์ได้ โดยการสร้างค่าของ x ที่เป็นเลขจำนวนเต็มออกมาทั้งหมดแล้วแทนค่า จากนั้นก็ทำการเปรียบเทียบเพื่อหาค่าที่ต่ำที่สุด ในกรณีนี้ แต่ถ้าค่า x เป็นเลขจำนวนจริงขึ้นมา การหาค่าของ x เพื่อให้ได้ $minf(x)$ ก็จะใช้เวลานานมาก ยิ่งถ้าไม่มีการกำหนดขอบเขตค่าของ x ด้วยแล้ว การหาค่า x ก็จะยิ่งใช้เวลานานมากออกไปอีก ดังนั้นการแก้ปัญหานี้จึงควรใช้วิธีประมาณการ หรือที่เรียกอีกอย่างหนึ่งว่า ฮิวริสติก (Heuristic)

1.2.3 ฮิวริสติก

วิธีการแก้ปัญหาด้วยการประมาณการ หรือมักนิยมเรียกว่า ฮิวริสติก เป็นวิธีการที่นิยมมากในระยะ 30 กว่าปีที่ผ่านมา อย่างไรก็ตาม วิธีการประมาณการนี้ ทำให้การรับประกันว่าคำตอบที่ได้เป็น

คำตอบที่ดีที่สุดนั้นอาจไม่เป็นจริง แต่สิ่งที่ได้มาก็คือระยะเวลาของการคำนวณที่ลดลงมาก ฮิวริสติกเป็นเทคนิคการประมาณการที่ได้มาจากการปฏิบัติในการแก้ปัญหา การเรียนรู้ และการค้นคว้า วิธีการของฮิวริสติกถูกนำมาใช้เพื่อทำให้เวลาที่ใช้ในการหาคำตอบของปัญหาการหาค่าเหมาะที่สุดที่มีขอบเขตกว้างได้เร็วขึ้น ในขณะที่การหาคำตอบด้วยวิธีการคำนวณแบบธรรมดาทำไม่ได้ ตัวอย่างของวิธีการที่เรียกว่าฮิวริสติก เช่น การเลือกสิ่งที่ดีที่สุดก่อน หรือที่เรียกกันว่า กรีดี (Greedy) ที่ใช้กับการทอนสตางค์ และการลองผิดลองถูก (Trial and error) ที่ใช้กับการเลือกหนังสือในห้องสมุด เป็นต้น สำหรับการแก้ปัญหาการเดินทางของเซลส์แมนตามที่ได้กล่าวมาข้างต้น สามารถใช้วิธีการฮิวริสติกแบบการเลือกสิ่งที่ดีที่สุดก่อนได้ วิธีการคือ ให้เริ่มต้นที่เมืองใดเมืองหนึ่งก่อน หลังจากนั้นให้เลือกเมืองที่จะเดินทางต่อไปด้วยการเลือกเมืองที่ใกล้ที่สุดเสมอ ทำเช่นนี้ไปเรื่อยๆ จนเดินทางไปครบทุกเมือง ก็จะทำให้เราได้เส้นทางเดินที่เหมาะที่สุด ในการแก้ปัญหาในลักษณะนี้ ผลที่ได้มักจะถูกเรียกว่าเป็นค่าที่เหมาะที่สุด ทั้งนี้ก็เพราะว่าไม่มีอะไรมารับประกันว่าคำตอบที่ได้จะดีที่สุด ในขณะที่การใช้วิธีเปรียบเทียบธรรมดาเพื่อหาคำตอบที่ดีที่สุด อาจจะทำให้การหาคำตอบเป็นไปไม่ได้เลย และที่สำคัญคือ ใช้เวลาในการหาคำตอบนานมาก

การแก้ปัญหาด้วยวิธีการฮิวริสติกสามารถแบ่งออกได้เป็น 2 ประเภทคือ วิธีคอนสตรัคทีฟ (Constructive Method) และวิธีการค้นหาแบบโลคอล (Local Search Method) (C. and Roli, A. Blum, 2003) อัลกอริธึมแบบคอนสตรัคทีฟจะสร้างคำตอบที่เป็นไปได้จากเริ่มต้น เมื่อเริ่มต้น อัลกอริธึมจะเพิ่มองค์ประกอบให้กับคำตอบทีละส่วน และทำซ้ำไปเรื่อยๆ จนกระทั่งได้คำตอบออกมา วิธีการนี้เป็นวิธีการที่เร็ว แต่ในหลายกรณี คำตอบที่ได้มักจะไม่ค่อยดีนัก ตัวอย่างของอัลกอริธึมแบบนี้ได้แก่ การแก้ปัญหาการเดินทางของเซลส์แมนตามที่กล่าวมาข้างต้น สำหรับอัลกอริธึมการค้นหาแบบโลคอลนั้นจะเริ่มด้วยคำตอบเบื้องต้นค่าใดค่าหนึ่ง คำตอบเบื้องต้นนี้อาจจะมาจากการสุ่ม หรืออาจจะใช้วิธีการอื่นสร้างคำตอบขึ้นมา และกำหนดให้คำตอบนี้เป็นคำตอบปัจจุบัน จากนั้นให้แทนที่คำตอบปัจจุบันด้วยคำตอบที่เลือกมาจากคำตอบข้างเคียง (Neighborhood Solution) ของคำตอบปัจจุบัน ที่ดีกว่า ซึ่งคำตอบข้างเคียงมีคำนิยามอย่างไปทางการดังนี้

ถ้าหากว่า s เป็นสมาชิกของเซท S โดยที่ S เป็นเซทของคำตอบในฟังก์ชัน $N : S \longrightarrow 2^S$ แล้ว และเรามีเซทของ s จำนวนหนึ่งจาก s ทั้งหมดที่เขียนแทนด้วย N(s) โดยที่ N(s) เป็นเซทย่อยใน S เราจะเรียกสมาชิกของ N(s) นี้ว่าคำตอบข้างเคียง (Neighborhood)

ความหมายอย่างง่ายๆ ก็คือ ถ้าในฟังก์ชันหนึ่ง มีคำตอบที่เป็นไปได้อยู่หลายคำตอบ และในบรรดาคำตอบที่เป็นไปได้เหล่านั้น มีชุดของคำตอบ (หลายคำตอบ) อยู่ชุดหนึ่งที่มีความเหมาะสมกับฟังก์ชันมากกว่าชุดอื่น ชุดคำตอบชุดนี้เรียกว่าชุดคำตอบข้างเคียง หรือคำตอบเพื่อนบ้านกัน (Neighbouring Solutions)

ในระยะ 20 กว่าปีที่ผ่านมา อัลกอริธึมแบบประมาณการใหม่ๆ เกิดขึ้นจำนวนมาก ส่วนใหญ่มีหลักการพื้นฐานคล้ายกันคือ การขยายความสามารถของวิธีการฮิวริสติกธรรมดาให้มีกรอบการทำงานในระดับที่สูงขึ้น โดยมีเป้าหมายที่จะเพิ่มประสิทธิภาพและประสิทธิผลของการค้นหาคำตอบในปริภูมิปัญหา วิธีการเหล่านี้ปัจจุบันเรียกว่า เมตาฮิวริสติก (Metaheuristic)

1.2.4 เมตาฮิวริสติก

เมตาฮิวริสติกถูกออกแบบมาสำหรับการหาค่าเหมาะที่สุด ที่มีความซับซ้อนมาก ถูกนำเสนอขึ้นมาครั้งแรกโดยโกลเวอร์ (F. Glover, 1986) โดยการนำเสนอเมตาฮิวริสติกที่ชื่อ**ตาบู (Tabu Search)** เมตาฮิวริสติกนี้ให้ทั้งประสิทธิภาพที่สูงขึ้น และมีประสิทธิผลที่ดี ในขณะที่วิธีการอื่นๆ ไม่สามารถทำได้ สิ่งที่เป็นประโยชน์ในการปฏิบัติเป็นอย่างมากสำหรับวิธีการของเมตาฮิวริสติกก็คือ ตัวมันเองให้ประสิทธิผลที่ดีพร้อมสามารถปรับใช้กับปัญหาการหาค่าเหมาะที่สุดได้ทุกรูปแบบ ซึ่งต่างจากวิธีการฮิวริสติกธรรมดาที่ออกแบบสำหรับการแก้ปัญหาแบบหนึ่งๆ เท่านั้น เมื่อมีปัญหาประเภทใหม่ ก็ต้องออกแบบวิธีการแก้ปัญหาหรือฮิวริสติกแบบใหม่ทุกครั้ง ยิ่งไปกว่านั้นบทเรียนจากการแก้ปัญหาหนึ่งก็ไม่สามารถทำให้มีลักษณะทั่วไปที่ดีพอสำหรับการแก้ปัญหาที่ต่างชนิดกันได้ ซึ่งต่างจากเมตาฮิวริสติกที่เป็นยุทธศาสตร์สำหรับการแก้ปัญหาที่มีลักษณะทั่วไป โดยที่ไม่ต้องไปพัฒนาวิธีการเฉพาะสำหรับปัญหาต่างๆ จึงทำให้เมตาฮิวริสติกเป็นที่นิยมกันอย่างกว้างขวางในฐานะที่เป็นอัลกอริธึมที่มีลักษณะทั่วไป และเมตาฮิวริสติกที่ดีจะให้คำตอบได้ใกล้เคียงกับค่าที่ดีที่สุดมาก โดยใช้เวลาในการคำนวณที่สมเหตุสมผล

วิธีการโดยทั่วไปของเมตาฮิวริสติกจะเริ่มต้นจากคำตอบ หรือชุดของคำตอบเริ่มต้น (Initial Solution) จากนั้นพัฒนาให้คำตอบนี้ให้ดีขึ้นเรื่อยๆ ด้วยหลักการของการค้นหาคำตอบ โดยที่โครงสร้างของการค้นหาคำตอบของวิธีการต่างๆ จะมีองค์ประกอบที่เหมือนกัน ในแต่ละขั้นตอนของอัลกอริธึม จะมีคำตอบที่เป็นไปได้ (s_k) ที่แสดงถึงคำตอบปัจจุบันของอัลกอริธึม ในเมตาฮิวริสติกหลายๆ ชนิด เช่น ซีมูเลตเต็ดอันนีลลิ่ง (Simulated Annealing) และตาบู (Tabu) การหาคำตอบที่เป็นไปได้จะเป็นวิธีค้นหา

แบบคำตอบต่อคำตอบ ซึ่งหมายความว่าค่าของ s_k จะมีค่าเดียวหรือจุดเดียว และ $s_k \in S$ เมื่อ S คือ ปริภูมิคำตอบ (Solution Space) คือ s_k เป็นเพียงบางส่วนของปริภูมิคำตอบ S ในขณะที่เมตาฮิวริสติกบางชนิดเช่น อัลกอริธึมพันธุการ (Generic Algorithm) การหาคำตอบที่เป็นไปได้ในแต่ละขั้นตอน s_k จะมีลักษณะเป็นชุด ดังนั้น s_k เป็นชุดของคำตอบที่เป็นไปได้ $s_k \subseteq S$ แต่อย่างไรก็ตามโครงสร้างพื้นฐานของการค้นหายังเหมือนกัน ไม่ว่าคำตอบที่เป็นไปได้จะมีคำตอบเดียว หรือแบบเป็นชุด ถ้ากำหนดว่า N(s) เป็นเซทของคำตอบข้างเคียง ชุดของคำตอบที่เป็นไปได้ $\{s_k\} \subset N(s)$ จะถูกเลือกและประเมินสอบ (Evaluate) การประเมินสอบนี้จะประกอบด้วย การคำนวณหาสมรรถนะของคำตอบที่เป็นไปได้และการเปรียบเทียบสมรรถนะของคำตอบที่เป็นไปได้ของ s_k แต่ละตัว และบางครั้งกับคำตอบที่เป็นไปได้อื่น บนฐานของการประเมินนี้ คำตอบที่เป็นไปได้อาจจะได้รับเลือกให้เป็นคำตอบปัจจุบัน $s_{k+1} = s_k$ หรือถูกปฏิเสธไม่ให้เป็นคำตอบปัจจุบัน ซึ่งกรอบของเมตาฮิวริสติกจะมีลักษณะทั่วไปดังนี้ (S. Ólafsson, 2006)

- สร้างคำตอบเริ่มต้น s_0 ให้ $s_0 = s_c$ และ k=0
- ทำซ้ำตามขั้นตอนดังต่อไปนี้
  - ให้คำตอบที่ดีที่สุดในปัจจุบันเป็น s_c
  - หาคำตอบข้างเคียงกับคำตอบปัจจุบัน N(s)
  - เลือกคำตอบที่เป็นไปได้จากคำตอบข้างเคียง $\{s_k\} \subset N(s)$
  - เลือก $\{s_k\}$ ให้เป็นคำตอบปัจจุบันและให้ค่าที่ดีที่สุดในรอบต่อไป $s_{k+1} = s_k$ ถ้าค่าของ s_k ดีกว่าค่าของ s_c หรือปฏิเสธค่า s_k แล้วใช้ s_c ตัวเดิม ถ้าค่าของ s_k แย่กว่าค่าของ s_c และให้ค่าที่ดีที่สุดในรอบต่อไป $s_{k+1} = s_c$
  - เพิ่มค่าของ k = k+1
- หยุดการทำงานเมื่อเป็นไปตามเงื่อนไขที่กำหนด และได้ค่า s_c เป็นคำตอบ

จากกรอบการทำงานของเมตาฮิวริสติกข้างต้น จะพบว่าในเมตาฮิวริสติกจะไม่ได้กำหนดวิธีการค้นหาที่ชัดเจนว่าจะทำอย่างไรสำหรับการหาคำตอบที่เป็นไปได้ และการหาค่าของคำตอบเหล่านั้น แต่เมตาฮิวริสติกจะกำหนดเป็นยุทธศาสตร์การแก้ปัญหาที่มีลักษณะทั่วไป ในทางปฏิบัติ การค้นหาคำตอบ

แบบนี้จะต้องปรับใช้ให้ตรงกับลักษณะเฉพาะของปัญหาที่เราต้องการแก้ เช่นกัน ในทางธรรมชาติ เซลฟ์ออกาไนเซชัน (Self-Organization) ก็เป็นตัวอย่างหนึ่งที่ถูกนำมาประยุกต์ใช้เป็นเมตาฮิวริสติก

1.3 เซลฟ์ออกาไนเซชัน

เซลฟ์ออกาไนเซชัน (Self Organization) เป็นศาสตร์ที่มีการศึกษามานาน ทั้งในเชิงทฤษฎีและการทดลอง การศึกษาเรื่องเซลฟ์ออกาไนเซชัน นั้นเป็นการทำความเข้าใจพฤติกรรมของธรรมชาติทั้งที่เป็นพฤติกรรมของสิ่งมีชีวิต และไม่มีชีวิต มีนักวิทยาศาสตร์หลายท่านได้ทุ่มเทกับการทำความเข้าใจในเรื่องเหล่านี้ เพื่อหาข้อสรุปที่เป็นระบบสำหรับการประยุกต์ใช้งานต่อไป ในการศึกษาเรื่องของเซลฟ์ออกาไนเซชันนี้ มีหลายเรื่องที่มีข้อยุติ และหลายเรื่องที่ยังอยู่ในระหว่างการค้นคว้าเพิ่มเติม

1.3.1 ประวัติการศึกษาเรื่องเซลฟ์ออกาไนเซชัน

บุคคลที่ได้รับการยอมรับในวงการ เซลฟ์ออกาไนเซชัน อย่างเป็นเรื่องเป็นราวในยุคแรกมีหลายท่าน เช่น รอส แอสบี้ (W.R. Ashby, 1947) ที่กล่าวถึงหลักการของ ระบบเซลฟ์ออกาไนซิง โดยการให้ความหมายของ เครื่องจักร (Machine) ออกาไนเซชัน (Organization) อินทิเกรชัน (Integration) และเซลฟ์ออกาไนเซชัน (Self-Organization) แอสบี้ได้อธิบายว่า เซลฟ์ออกาไนเซชัน เกิดจากการเปลี่ยนองค์ประกอบ (Organization) ภายในของตัวระบบมันเอง แทนที่จะเป็นการเปลี่ยนจากองค์ประกอบภายนอก

ในช่วงที่ใกล้กัน ฟาร์เรย์ และคลาร์ค (B.G. and Clark, W.A. Farley, 1954) ได้นำเสนอแนวคิดและนิยามเกี่ยวกับ ระบบเซลฟ์ออกาไนซิง (Self Organizing) ด้วยการสังเคราะห์ระบบ และได้สร้างแบบจำลองด้วยคอมพิวเตอร์ โดยที่ในตอนเริ่มต้นระบบเป็นองค์กร (Organization) ที่ถูกสร้างขึ้นมาโดยการจัดองค์กรแบบสุ่ม (Randomly Organized) เมื่อผ่านการทำงานไประยะหนึ่ง ระบบจะปรับปรุงตัวมันเองให้สามารถทำงานง่ายๆ ตามที่กำหนดไว้ได้

ในอีก 5 ปีต่อมา ฟอร์สเตอร์ (H. V. Foerster, 1960) ได้อาศัยกฎข้อสองของอุณหพลศาสตร์ (Thermodynamics) เพื่ออธิบายให้เห็นถึงความสัมพันธ์ของระบบเซลฟ์ออกาไนซิง กับภาวะแวดล้อม (Environment) พร้อมทั้งแสดงให้เห็นถึงวิธีการที่แม่เหล็กที่ถูกจับใส่ในกล่อง เรียงตัวกันเป็นรูปร่างที่

เป็นระเบียบหลังจากที่จับกล่องนั้นเขย่า เพื่อพิสูจน์ว่าการจัดระเบียบมาจากความไม่เป็นระเบียบ (Order came from disorder)

ในปีเดียวกัน ค.ศ. 1959 กราซเซ (P.P. Grassé, 1984) ได้การศึกษาพฤติกรรมของปลวก และเสนอทฤษฎีเกี่ยวกับ สติกเมอร์จี (Stigmergy) ซึ่งหมายถึง “งานกระตุ้นผู้ทำงาน (*the work excites the workers)*” ความหมายของมันก็คือการทำงานร่วมกันในกลุ่มไม่จำเป็นที่จะต้องอาศัยการสื่อสารถึงกันโดยตรงก็ได้ การสื่อสารทางอ้อมโดยผ่านสิ่งแวดล้อมก็สามารถทำได้เช่นกัน จากพฤติกรรมของปลวกที่ใช้ฟีโรโมน (Pheromone) ที่วางไว้บนดินข้างจอมปลวก เพื่อการสื่อสารระหว่างกันในการสร้างรัง การสร้างรังของปลวกทำโดยที่ไม่มีการควบคุมจากส่วนกลาง หรือมีปลวกตัวใดตัวหนึ่งเป็นหัวหน้าฝูงคอยสั่งการในการก่อสร้าง การกระทำนี้ นับเป็น เซลฟ์ออกาไนซิง ของการทำงานในหมู่แมลง ที่อาศัยการสื่อสารด้วยการวางฟีโรโมนบนพื้นดิน (ผ่านสภาพแวดล้อม) เพื่อการสื่อสารระหว่างกัน

ในช่วงทศวรรษที่ 70 ผู้ที่ได้รับรางวัลโนเบลชื่อ อิลยา พริโกจิน (Glansdorff, 1971) และทีมงานได้ใช้อุณหพลศาสตร์อธิบายเรื่อง เซลฟ์ออกาไนเซชัน อีกครั้ง แนวคิดที่สำคัญคล้ายกับคำอธิบายของ ฟอร์สเตอร์ ที่กล่าวว่าในระบบเปิด (Open System) ค่าของเอ็นโทรฟีจะลดลงเมื่อมีพลังงานจากภายนอกถูกใส่เข้ามาในระบบ หรืออีกนัยหนึ่ง สสาร (Matter) จะจัดการตัวเองภายใต้แรงกดดันจากภายนอกเพื่อที่จะเปลี่ยนไปสู่สถานะ (State) ใหม่ เมื่อเอ็นโทรฟีลดลง จากการเปรียบเทียบผลการศึกษาของ พริโกจิน และ ฟอร์สเตอร์ กับผลการศึกษาของ กราซเซ จะมีความแตกต่างกันโดยพื้นฐานคือ ในกรณีของกราซเซ สติกเมอร์จี เป็นผลที่เกิดจากการเปลี่ยนแปลงจากภายในของมันเอง (ตัวปลวก หรือ มด) แต่ในกรณีพริโกจิน และฟอร์สเตอร์นั้น เซลฟ์ออกาไนซิง เป็นการเปลี่ยนแปลงเกิดจากแรงกดดันจากภายนอกระบบ

ในตอนเริ่มต้นของการศึกษาเรื่องเซลฟ์ออกาไนเซชัน ผู้ศึกษาส่วนใหญ่จะอยู่ในกลุ่มของการศึกษาเรื่อง ไซเบอร์เนติกส์ (Cybernetics) ชีววิทยา และทฤษฎีระบบ (System Theory) ในระยะเวลาต่อมาการศึกษาในเรื่องนี้ได้การกระจายไปแทบทุกสาขา เช่น ฟิสิกส์ และวิทยาการคอมพิวเตอร์ ในทางฟิสิกส์ มีการศึกษาเกี่ยวกับการก่อตัวอย่างเป็นรูปทรงทางธรรมชาติ เช่น เกล็ดหิมะ ลายของทรายบนชายหาด การแตกตัวอย่างสมมาตรโดยธรรมชาติ เช่น ลายแตกของผิวดิน เป็นต้น ใน

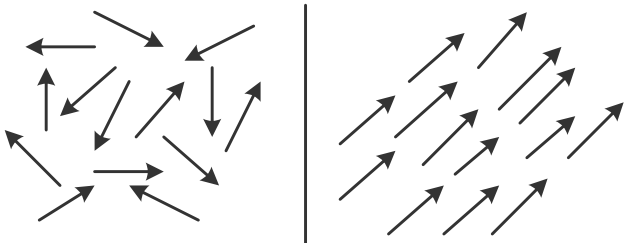
ด้านวิทยาการคอมพิวเตอร์ การทำงานแบบ เซลฟ์ออกาไนเซชัน ถูกนำมาใช้ในการจำลองแบบสำหรับการแก้ปัญหา เช่นการหาค่าเหมาะที่สุด (Optimization) เป็นต้น

ในช่วง 20 กว่าปีที่ผ่านมา การวิจัยมุ่งไปเพื่อสร้างซอฟต์แวร์จำลองกลไกการทำงานของ เซลฟ์ออกาไนเซชัน โดยเฉพาะการจำลองกลไกการทำงานของธรรมชาติ และฝูงสัตว์ อัลกอริธึมที่จำลองการทำงานแบบเซลฟ์ออกาไนเซชัน ได้แก่ ซีมูลเลเต็ด แอลนิลลิง (Simulated Annealing) อัลกอริธึมพันธุการ (Genetic Algorithm) การหาค่าเหมาะที่สุดแบบฝูงมด (Ant Colony Optimization) การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค (Particle swarm optimization) และการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ (Artificial Bee Colony Algorithm) เป็นต้น

1.3.2 ลักษณะของเซลฟ์ออกาไนเซชัน

ตัวอย่างของเซลฟ์ออกาไนซ์ที่มักถูกอ้างถึงเสมอมี 2 ตัวอย่างคือ การเป็นแม่เหล็ก และการถ่ายเทความร้อนของเรย์ลี เบนนาร์ด (Rayleigh-Benard Convection)

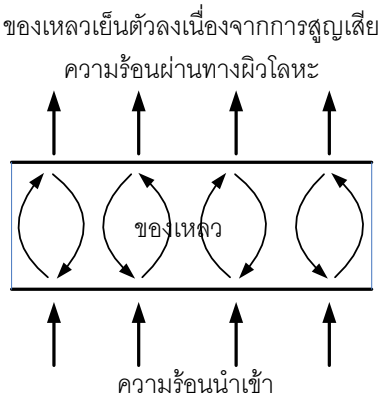
ตัวอย่างแรก การเป็นแม่เหล็ก จากการพิจารณาโลหะที่สามารถทำเป็นแม่เหล็กได้เช่นเหล็ก ที่เนื้อในของเหล็กจะประกอบด้วยชิ้นส่วนของแม่เหล็กเล็กๆ จำนวนมาก เรียกว่าสปิน (Spins) เรียงตัวกันอย่างไม่เป็นระเบียบ เนื่องจากการเรียงตัวแบบไม่เป็นระเบียบนี้เป็นผลให้สนามแม่เหล็กที่เกิดขึ้นในแต่ละสปินมีทิศทางที่สลับขั้วกัน และหักล้างกันเองจนหมด เหล็กทั้งแท่งจึงไม่แสดงสนามแม่เหล็กออกมา การเรียงตัวที่ไม่เป็นระเบียบของสปินนี้เกิดจากอุณหภูมิ ยิ่งอุณหภูมิสูงขึ้น การเรียงตัวของสปินจะยิ่งกระจัดกระจายมากขึ้น ในทางกลับกัน เมื่ออุณหภูมิเย็นตัวลง สปินแต่ละตัวก็เริ่มเรียงตัวกันอย่างเป็นระเบียบมากขึ้น จนทำให้สปินทุกตัวชี้ไปในทางเดียวกันหมด ก่อให้เกิดการเสริมกันของสนามแม่เหล็กในแต่ละสปิน และทำให้เหล็กทั้งแท่งมีสนามแม่เหล็กเกิดขึ้น รูปที่ 1.1 แสดงการเรียงตัวกันของสปิน ที่มีการเรียงตัวอย่างไม่เป็นระเบียบทางซ้ายมือ และการเรียงตัวอย่างเป็นระเบียบทางขวามือ ซึ่งการเรียงตัวของสปินอย่างเป็นระเบียบทางขวามือเกิดจากการลดอุณหภูมิให้กับสปินของเหล็กที่อยู่ทางซ้ายมือ และก่อให้เกิดเป็นสนามแม่เหล็กขึ้น



การเรียงตัวอย่างไม่เป็นระเบียบ | การเรียงตัวอย่างเป็นระเบียบ

รูปที่ 1.1 การเรียงตัวของสปินในเหล็ก

อีกตัวอย่างหนึ่ง เป็นตัวอย่างของการถ่ายเทความร้อนที่เรียกว่า การหมุนเวียนของเรย์ลี เบนนาร์ด เป็นการทดลองในเรื่องการพาความร้อนของของเหลว ในการทดลองเรย์ลี เบนนาร์ด ได้นำของเหลวมาใส่ไว้ในระหว่างกลางของแผ่นนำความร้อน 2 แผ่นที่อยู่ข้างล่างและบน แล้วให้ความร้อนกับแผ่นที่อยู่ข้างล่าง เพื่อที่จะทำให้เกิดความแตกต่างของอุณหภูมิระหว่างแผ่นล่างกับแผ่นบน เมื่อความแตกต่างของอุณหภูมิมากถึงจุดหนึ่ง อนุภาคของของเหลวที่มีอิสระในการเคลื่อนที่ก็จะเริ่มเคลื่อนเพื่อถ่ายเทพลังงานความร้อน ระหว่างแผ่นนำความร้อนที่อยู่ข้างล่างและบน เรย์ลี เบนนาร์ดได้พบการหมุนเวียนของของเหลวเกิดขึ้นมาตาม รูปที่ 1.2 แสดงด้านหน้าตัดของการพาความร้อน (Convection) จากการจำลองแบบจะเห็นทิศทางการไหลเวียนของของเหลวที่ไหลขึ้นและลงจะเสริมกันในระหว่างวงที่อยู่ข้างกันเสมอ เมื่อความร้อนของแผ่นล่างสูงขึ้น ทำให้ความแตกต่างของอุณหภูมิมีมากขึ้น การหมุนเวียนก็จะเร็วขึ้น รูปแบบการหมุนเวียนนี้ถือเป็นการถ่ายเทความร้อนที่มีประสิทธิภาพสูงสุดที่เกิดขึ้นโดยธรรมชาติ จากผลการทดลอง เรย์ลี และเบนนาร์ดได้แสดงให้เห็นความพยายามของระบบทั้งระบบในการรักษาสมดุลของตัวมันเอง ในการป้องกันการรบกวนจากภายนอกที่เป็นความร้อน เมื่อมีการป้อนความร้อนเข้ามาในระบบ ซึ่งถือว่าเป็นการรบกวนจากภายนอก ระบบก็จะหาทางปรับตัวมันเอง ด้วยการถ่ายเทความร้อนที่มีประสิทธิภาพ เพื่อทำให้ระบบทั้งหมดกลับไปสู่ภาวะสมดุล



รูปที่ 1.2 แสดงด้านหน้าตัดของการพาความร้อน (Convection)

สำหรับคำจำกัดความของเซลฟ์ออกาไนเซชัน ยังเป็นที่สับสนและก็ยังเป็นที่ถกเถียงกันอยู่ (G.D.M., Gleizes, M.P., and Karageorgos, A. Serugendo, 2006) เพื่อที่จะทำให้มีความเข้าใจในเบื้องต้น วูลฟ์ และฮอลโฮลเว็ด (T.D. and Holvoet, T. Wolf, 2004) ได้ให้คำจำกัดความสำหรับการใช้งาน (Working Definition) ของเซลฟ์ออกาไนเซชันว่า

เซลฟ์ออกาไนเซชัน เป็นกระบวนการที่เป็นพลวัตที่ระบบสร้างขึ้นมาเพื่อรักษาโครงสร้างของมันเอง โดยไม่มีการควบคุมจากภายนอก ซึ่งหมายถึงไม่มีการกำหนดทิศทางการทำงาน การจัดการ การแทรกแซง การผลักดัน และการมีส่วนเกี่ยวข้องจากสิ่งที่อยู่ภายนอกระบบ ทั้งนี้ไม่รวมถึงการนำเข้าข้อมูลจากภายนอกระบบ ตราบใดที่ข้อมูลเหล่านั้นไม่เป็นคำสั่งควบคุม (Control Instructions) โดยมีหลักการพื้นฐานอยู่ 4 ประการ

- การป้อนกลับเชิงบวก เป็นการเสริมการกระทำของระบบ
- การป้อนกลับเชิงลบ เป็นการปรับสมดุล และสร้างความมั่นคงให้กับระบบ
- การขยายของความแปรปรวน เช่น ความไม่แน่นอน ความผิดพลาด และการสุ่มหาที่เกิดขึ้นกับระบบ
- การกระทำที่ทำร่วมกันขององค์ประกอบภายในหรือประชากรในองค์กรทั้งหมด

จากตัวอย่างที่กล่าวมาข้างต้น เมื่อนำมาเปรียบเทียบกับหลักการพื้นฐาน 4 ประการ ตัวอย่างแรกการเรียงตัวของสปิน การป้อนกลับเชิงบวก เป็นการที่สนามแม่เหล็กของแต่ละสปินทำงานเสริมกัน การ

ป้อนกลับเชิงลบ เป็นการที่แต่ละสปินกระตุ้นซึ่งกันและกันให้เรียงตัวอย่างเป็นระเบียบ การขยายความแปรปรวนเป็นสิ่งรบกวนจากภายนอกที่เป็นตัวกระตุ้นให้เกิดการเปลี่ยนแปลงซึ่งก็คืออุณหภูมิที่เปลี่ยนไป สำหรับการทำงานร่วมกันขององค์ประกอบภายใน ก็คือการที่สปินแต่ละตัวมีส่วนช่วยกันทำให้เหล็กทั้งแท่งเป็นแม่เหล็ก

สำหรับตัวอย่างของการถ่ายเทความร้อน การป้อนกลับเชิงบวก เป็นการที่อนุภาคของของเหลวในระบบที่มีการไหลเวียนอย่างมีประสิทธิภาพ เพื่อการถ่านเทความร้อน การป้อนกลับเชิงลบเป็นการที่อนุภาคของของเหลว ทำให้ระบบเกิดความสมดุลด้วยการถ่ายเทความร้อนจากแผ่นนำความร้อนที่มีอุณหภูมิที่สูงกว่าไปสู่ที่ต่ำกว่า การขยายความแปรปรวน เป็นความร้อนที่ถูกใส่เข้ามาในระบบ เพื่อสร้างความแตกต่างของอุณหภูมิ และการทำงานร่วมกันขององค์ประกอบ ก็เป็นการทำงานของแต่ละอนุภาคในของเหลว

คามาซีน (Camazine, 2003) ได้อธิบายเรื่อง เซลฟ์ออกาไนเซชัน ในหนังสือของเขาว่า เป็นกระบวนการที่รูปแบบ (Pattern) ที่อยู่ในระดับภาพรวม (Global level) ของระบบ เกิดจาก (Emerge) การปฏิสัมพันธ์กันของส่วนประกอบต่างๆ ในระดับล่าง (Lower level) ของระบบ โดยอาศัยกฎที่เกิดจากข้อมูลภายในระบบ (Local Information) กำหนดวิธีการปฏิสัมพันธ์ระหว่างประชากรของระบบโดยที่ไม่มีความเกี่ยวพันกับรูปแบบที่อยู่ชั้นสูงขึ้นไป

หรือกล่าวอีกนัยหนึ่งว่า ในระบบของเซลฟ์ออกาไนเซชัน รูปแบบและองค์การ (Organization) ที่เกิดขึ้นใหม่พัฒนามาจากการปฏิสัมพันธ์ภายในของระบบเอง โดยไม่มีการแทรกแซงจากอิทธิพลภายนอก เช่น ผู้นำหรือผู้ที่ควบคุมระบบ รูปแบบหรือองค์กรที่เกิดขึ้นใหม่นี้อุบัติ (Emerge) จากตัวของมันเอง ไม่ใช่การบงการจากภายนอก ผลจากการทดลองและทางทฤษฎีทำให้เราทราบว่ารูปแบบใหม่นี้เกิดจากกฎง่ายๆ เพียง 2-3 ข้อเท่านั้นที่ทำซ้ำๆ กันภายในระบบหลายๆ รอบ

ตัวอย่างเช่น การบินของฝูงนก ที่มีระยะห่างระหว่างกันอย่างพอดีที่ไม่ทำให้เกิดการชนกัน และมีทิศทางในการบินไปทิศเดียวกัน โดยไม่มีตัวใดตัวหนึ่งที่คอยกำกับการบินของทั้งฝูง โดยที่นกแต่ละตัวจะเป็นผู้ที่กำหนดระยะห่างระหว่างกัน พร้อมกับบินไปในทิศทางเดียวกัน ซึ่งเป็นเพียงกฎง่ายๆ ของนกแต่ละตัว

การอธิบายเรื่องเซลฟ์ออกาไนเซชันนั้น นอกจากหลักการทั้ง 4 และนิยามของตัวมันเองแล้ว ยังมีปรากฏการณ์ของเซลฟ์ออกาไนเซชันที่น่าสนใจอีกดังต่อไปนี้

ปฏิกิริยาท้องถิ่นและการเปลี่ยนแปลงในภาพรวม

การเปลี่ยนแปลงที่เกิดขึ้นไม่ว่าจะเป็นตัวอย่างของการกลายเป็นแม่เหล็ก หรือการถ่ายเทความร้อนของของเหลว การเปลี่ยนแปลงจะเริ่มต้นจากจุดเล็กๆ ภายในตัวมันก่อน จากนั้นการเปลี่ยนแปลงนั้นก็จะค่อยๆ แพร่ขยายไปจนครอบคลุมระบบทั้งระบบ คือเหล็กทั้งแท่ง หรือของเหลวในภาชนะทั้งหมด ปฏิกิริยานี้ เริ่มต้นที่ท้องถิ่นและขยายตัวไปสู่ภาพรวม ทำให้ระบบทั้งระบบเปลี่ยนแปลงสภาพเป็นสภาพใหม่ที่เรียกว่า อิเมอร์เจนซ์ (Emergence) ซึ่งเป็นการเปลี่ยนเชิงระบบ ไม่ใช่การเปลี่ยนแปลงที่เกิดขึ้นเล็กน้อย หรืออย่างธรรมดา ถ้าจะอธิบายการเปลี่ยนแปลงนี้ในเรื่องการกลายเป็นแม่เหล็กของเหล็ก ด้วยเหตุผลของการจัดเรียงอย่างเป็นระเบียบของสปินนี้เริ่มต้นจาก การเหนี่ยวนำของสปินแต่ละตัว โดยที่มีสปินบางตัวเปลี่ยนทิศทางการชี้ และมีผลให้เกิดการเหนี่ยวนำให้สปินตัวที่อยู่ข้างเคียงไม่กี่ตัวชี้ไปในทิศเดียวกัน จากนั้นการเหนี่ยวนำก็จะขยายวงกว้าง และทำให้สปินทั้งระบบชี้ไปในทิศเดียวกันหมด เป็นผลให้เกิดการเปลี่ยนสภาพ (หรืออิเมอร์เจนซ์) ของโลหะจากเหล็กธรรมดากลายเป็นแม่เหล็ก สำหรับตัวอย่างของการถ่ายเทความร้อนก็เช่นกัน เริ่มต้นจากโมเลกุลของเหลวที่ด้านล่างได้รับความร้อนมากขึ้น และทำให้เกิดความแตกต่างของอุณหภูมิเมื่อเทียบกับโมเลกุลที่อยู่ด้านบน โมเลกุลของน้ำที่ร้อนก็จะเริ่มเคลื่อนตัวขึ้น และส่วนที่เย็นก็จะไหลลงมาแทนที่ ในส่วนของโมเลกุลที่ไหลขึ้น ก็จะเหนี่ยวนำให้โมเลกุลที่อยู่ข้างๆ ลอยตัวขึ้นไปด้วย และเช่นกัน ในส่วนที่ไหลลงก็จะเหนี่ยวนำให้โมเลกุลที่อยู่ข้างเคียงไหลลงมาด้วย จนทำให้เกิดการไหลของของเหลวที่วนเป็นลูปที่เสริมกันหลายลูป ดังรูปที่ 1.2

การควบคุมอย่างกระจาย

ในการเกิดปรากฏการณ์ที่เป็นเซลฟ์ออกาไนซ์นั้น จะเป็นการเกิดที่ต่างคนต่างเกิด ไม่ได้มีการควบคุมใดๆ จากส่วนกลาง หรือการมีศูนย์กลางการควบคุม การควบคุมจะมีขึ้นอย่างกระจัดกระจายทั่วทั้งระบบ โดยเกิดขึ้นที่โมเลกุลหนึ่งและสร้างการเปลี่ยนแปลงกับโมเลกุลที่อยู่ข้างเคียง ซึ่งเป็นการควบคุมที่เกิดขึ้นที่ท้องถิ่น (Local Control) ลักษณะการควบคุมที่ทำให้เกิดการเปลี่ยนแปลงนี้จะเกิดขึ้นเป็นหย่อมๆ ทั่วทั้งระบบ โดยมีการเปลี่ยนแปลงที่เป็นลักษณะเดียวกัน

ความแข็งแกร่งและการฟื้นคืนสภาพได้

ระบบที่เกิดขึ้นจากเซลฟ์ออกาไนซ์จะเป็นระบบที่มีความแข็งแกร่ง ทนทานต่อการรบกวนจากภายนอก และสามารถฟื้นคืนสภาพได้ ถ้ามีการเปลี่ยนแปลงเกิดขึ้นในระบบ ให้ลองคิดถึงแท่งเหล็กที่กลายสภาพเป็นแม่เหล็ก จะเห็นได้ว่าในแต่ละองค์ประกอบเล็กๆ ที่เป็นสปิน จะมีการเหนี่ยวนำต่อกันระหว่างตัวเองและเพื่อนที่อยู่ข้างเคียงโดยตลอด ทำให้การจัดเรียงตัวเป็นไปในลักษณะที่เสริมแรงกันและปรากฏเป็นขั้วเหนือใต้ขึ้น การเกิดขึ้นของระบบเป็นการเสริมแรงในระดับสปิน ที่เป็นองค์ประกอบที่เล็กมาก ดังนั้นระบบจึงมีความแข็งแกร่งมาก ในบางกรณีที่มีแรงกระทำจากภายนอกที่รุ่นแรงมากมากระทบกับระบบ เช่นมีสนามแม่เหล็กอื่นมารบกวน ถ้าหากว่า มีสปินบางตัวเริ่มจะเปลี่ยนทิศไป สปินที่อยู่ข้างเคียงก็จะเหนี่ยวนำให้สปินนั้นกลับมาอยู่ทิศเดิมได้

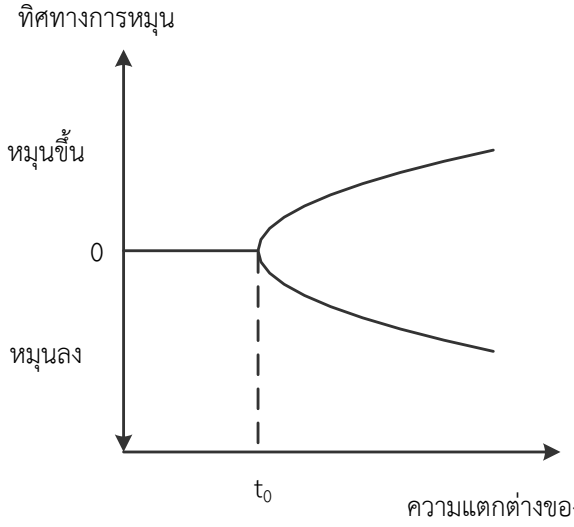
ในระบบของการถ่ายเทความร้อนก็เช่นกัน ระบบการไหลเวียนของของเหลวจะเป็นไปในลักษณะนั้นอย่างมั่นคง เนื่องจากการไหลเวียนเป็นการเสริมแรงกันของทั้งระบบ และมีความทนทานต่อแรงกระทบจากภายนอกที่เป็นความร้อนที่ถูกใส่เข้ามา

ไบฟาร์เคชัน

การเปลี่ยนแปลงที่เกิดจากเซลฟ์ออกาไนซ์ ไม่ได้เกิดขึ้นอย่างค่อยเป็นค่อยไป แต่เกิดจากการสะสมของการเปลี่ยนแปลงภายใน จนถึงจุดแตกหัก แล้วทำให้เกิดการเปลี่ยนแปลงทั้งระบบ ที่เรียกว่าไบฟาร์เคชัน (Bifurcation) ดูจากตัวอย่างของสปินที่เปลี่ยนสภาพเหล็กเป็นแม่เหล็ก การเปลี่ยนไม่ได้เกิดขึ้นจากการค่อยๆ เปลี่ยน แต่การเปลี่ยนจะเกิดขึ้นที่การเปลี่ยนทิศทางการชี้ของสปินสะสม เมื่อการเปลี่ยนสะสมไปถึงจุดหนึ่ง แท่งเหล็กก็จะกลายเป็นแม่เหล็กโดยทันที จากนั้นแม่เหล็กก็จะเพิ่มเสถียรภาพของตนเองขึ้นไปเรื่อยๆ จนถึงจุดสมดุล และที่จุดเปลี่ยนนี้จะเป็นไปได้ใน 2 ทิศทาง ทำให้เหล็กกลายเป็นแม่เหล็ก N-S หรือ S-N ในระบบการถ่ายเทความร้อนของเรย์ลี เบนนาร์ด ก็เช่นกันที่การเปลี่ยนแปลงสามารถเกิดได้ 2 ทิศทาง คือน้ำวนขึ้น และหมุนลง ดังแสดงในกราฟรูปที่ 1.1

ความแตกต่างของตัวอย่างของแม่เหล็ก และการถ่ายเทความร้อน ก็คือ ตัวอย่างของแม่เหล็กเมื่อระบบพัฒนาถึงจุดสูงสุดคือสถานะเสถียร (Equilibrium) สำหรับตัวอย่างการถ่ายเทความร้อนจุดสูงสุดคือสถานะสมดุล (Stationary State) ทั้งสองสถานะมีความแตกต่างกัน

สถานะเสถียรในทางอุณหพลศาสตร์ หมายถึงสถานะมีพลังงานศักย์ (Potential Energy) เหลืออยู่น้อยมาก การที่จะถึงจุดนี้ได้จะต้องไม่มีพลังงานจลน์หลงอยู่ในระบบ ในระบบนี้ถ้าไม่มีพลังงานรบกวนจากภายนอก ระบบจะคงสถานะเช่นนี้ไปตลอดกาล ในขณะที่ในระบบการถ่ายเทความร้อน เป็นระบบที่มีการใส่พลังงานเข้ามาในระบบอยู่ตลอดเวลา การรักษาความสมดุลของระบบจะเกิดจากการถ่ายเทความร้อนไปยังสภาพแวดล้อมที่อยู่ข้างบน สำหรับระบบแบบนี้เป็นระบบยืดหยุ่นกว่า แต่เปราะบาง และสามารถปรับตัวเองให้เข้ากับสภาวะแวดล้อมที่เปลี่ยนไปได้



รูปที่ 1.1 กราฟไบฟาร์เคชันของการถ่ายเทความร้อนของเรย์ลี เบนนาร์ด t_0 คือจุดไบฟาร์เคชัน (Heylighen)

1.3.3 อิเมอร์เจนซ์

เมื่อระบบมีการเปลี่ยนแปลงจากสถานะธรรมดาให้กลายเป็นสถานะใหม่ เช่นการเปลี่ยนจากเหล็กให้เป็นแม่เหล็ก องค์ประกอบที่เกิดการเปลี่ยนแปลง จะเกิดแรงเสริมกันจนทำให้ทั้งระบบมีการเปลี่ยนแปลง การเปลี่ยนแปลงนี้ เกิดจากการป้อนกลับเชิงบวก (Positive Feedback) เมื่อมีการเปลี่ยนแปลงที่เสริมกันจากการป้อนกลับเชิงบวก ทำให้ระบบเกิดการเปลี่ยนที่เร็วและรุ่นแรงมากจนกระทั่งเกิดการเปลี่ยนสภาพ การเปลี่ยนลักษณะนี้จะเรียกว่า **อิเมอร์เจนซ์ (Emergnece)** ระบบทั้งระบบก็จะอยู่ในสภาพสมดุล ทำให้การป้อนกลับเชิงบวกไม่มีผลอีกต่อไป ซึ่งการเปลี่ยนแปลงนี้จะไม่เป็นการเปลี่ยนแปลงที่มีลักษณะเชิงเส้น แต่จะเป็นแบบไบฟาร์เคชัน ในขณะเดียวกัน อาจจะมีการ

เปลี่ยนแปลงอื่นที่เกิดขึ้นได้เช่นกัน คือการทำให้ระบบเสียความสมดุล และกลายสภาพไปอยู่ในสถานะดั้งเดิม การเปลี่ยนในลักษณะนี้จะเกิดจากการป้อนกลับเชิงลบ เมื่อมีการเปลี่ยนแปลงใดๆ ที่เกิดจากการป้อนกลับเชิงลบขึ้น ระบบโดยการป้อนกลับเชิงบวก ก็จะทำให้ระบบกลับมายังจุดที่สมดุลอีก

เป็นที่น่าสังเกตว่า การเปลี่ยนแปลงแบบเซลฟ์ออกาไนเซชัน จะเริ่มที่สถานะปกติที่เป็นสถานะเริ่มต้น จากนั้นก็จะเกิดการเปลี่ยนแปลงภายใน อย่างอิสระ และรวดเร็ว จนทำให้เกิดอิเมอร์เจนซ์ และกลายสภาพเป็นระบบใหม่ที่มีความเสถียร เมื่อมีผลกระทบจากภายนอกที่ทำให้ระบบเสียความสมดุล ระบบก็จะปรับตัวเองอย่างรวดเร็วให้กลับไปสู่ความเสถียรอีก เราจะเห็นว่าในรอบของการปรับตัว จะขึ้นอยู่กับความรุนแรงของการเสียสมดุล ไม่ว่าระดับของการเสียสมดุลจะมากหรือน้อย ระบบก็จะฟื้นตัวกลับได้ ถ้าสมมุติขั้นตอนการปรับตัวมี 5 ขั้นคือ 1-2-3-4-5 ที่ขั้น 5 ถือว่าระบบมีความสมดุล การฟื้นตัวของระบบจะสามารถเริ่มจากช่วงกลางของระบบก็ได้ เช่นที่ขั้นตอนที่ 4 หรือ 3 เป็นต้น ระบบก็จะปรับตัวกลับมาสู่ความเสถียรอีก ซึ่งการรักษาความเสถียรของระบบอาจจะเป็น 1-2-3-4-5-**4-5** คือเมื่อกระบวนการดำเนินมาถึงขั้นตอนที่ 5 ซึ่งเป็นขั้นตอนที่เสถียรแล้ว และเมื่อมีแรงกระทบจากภายนอก ทำให้ระบบเสียสมดุลและความเสียหายเกิดขึ้นที่ขั้นตอน 4 ระบบก็จะกลับไปทำที่ขั้นตอนที่ 4 และ 5 จนระบบกลับมาภาวะเสถียรอีก หรือ ถ้าความเสียหายเกิดขึ้นที่ขั้นตอน 3 ระบบก็จะไปเริ่มที่ขั้นตอน 3 ดังนี้ 1-2-3-4-5-***3-4-5*** ดังนั้นการฟื้นตัวจะเป็นลักษณะที่มีลำดับขั้นที่ไม่จำเป็นที่จะต้องเริ่มจากจุดเริ่มต้นเสมอไป ซึ่งระบบนี้เรียกว่าองค์กรแบบปิด (Organizational Closure)

อิเมอร์เจนซ์ เป็นจุดสำคัญของระบบ จากตัวอย่างที่กล่าวมา อิเมอร์เจนซ์เป็นผลมาจากการที่ประชากรภายในระบบมีปฏิสัมพันธ์กันอย่างง่ายๆ จนเป็นผลให้ระบบโดยรวมเข้าสู่ภาวะสมดุลที่มีเสถียรภาพ และกลายเป็นระบบที่มีลักษณะใหม่ที่แตกต่างจากภาวะพื้นฐานเดิม หากพิจารณาจากคำนิยามที่เป็นทางการของอิเมอร์เจนซ์คือ การเกิดขึ้นของสิ่งใหม่ (Novel) และการโคฮีเร้นท์ (Coherent) กันของ โครงสร้าง แพทเทริน (Pattern) และคุณสมบัติ ในระหว่างกระบวนการทำงานของเซล์ฟออกาไนเซชัน ของระบบที่ซับซ้อน (Complex system) (Jeffrey, 1999)

โดยที่ระบบที่ซับซ้อน หมายถึงการเชื่อมกันภายในของประชากรต่างๆ ที่ทำให้เกิดคุณสมบัติอย่างใดอย่างหนึ่งหรือมากกว่านั้น ซึ่งคุณสมบัติดังกล่าวไม่สามารถพิจารณาได้จากประชากรอันใดอันหนึ่ง

ตัวอย่างที่แสดงให้เห็นการเกิดอิเมอร์เจนซ์ นอกจากปรากฏการณ์ทางฟิสิกส์ ดังที่กล่าวมาในเรื่อง แม่เหล็ก และการพาความร้อนแล้ว ในพฤติกรรมของแมลงทางธรรมชาติก็มี เช่นการสร้างรังของปลวก การหาอาหารของมด และผึ้ง เป็นต้น ในทางชีววิทยา การทำงานในร่างกายของสิ่งมีชีวิตก็ถือว่าเป็นอิเมอร์เจนซ์จากเซล์ฟออกาไนเซชันเช่นกัน

1.4 ปัญญาเชิงกลุ่ม

ปัญญาเชิงกลุ่ม (Swarm Intelligence) เป็นอิเมอร์เจนซ์ที่เกิดจากเซล์ฟออกาไนเซชันที่มาจากการทำงานร่วมกันขององค์ประกอบในทางธรรมชาติทั้งจากสิ่งมีชีวิตและไม่มีชีวิต ปัญญาเชิงกลุ่มมีคุณลักษณะที่สำคัญคือ ตัวระบบจะเป็นกลุ่มของสมาชิกที่เป็นอิสระ โดยที่สมาชิกแต่ละตัวจะมีลักษณะเดียวกัน เช่น ฝูงมด ฝูงนก และฝูงผึ้ง เป็นต้น ภายในกลุ่มที่ประกอบด้วยสมาชิกแต่ละตัวนั้น จะมีการติดต่อหรือประสานกันบนพื้นฐานของกฎเชิงพฤติกรรมอย่างง่ายๆ ที่อาศัยประโยชน์จากข้อมูลข่าวสารที่มีอยู่ในพื้นที่หรือท้องถิ่นนั้น ด้วยการแลกเปลี่ยนกันโดยตรงหรือการแลกเปลี่ยนผ่านสื่อกลางในการติดต่อกัน

การติดต่อปฏิสัมพันธ์กันดังกล่าว จะทำให้ระบบทั้งหมดเกิดเป็นพฤติกรรมแบบเซลฟ์ออกาไนซ์ ซึ่งพฤติกรรมเชิงกลุ่มในลักษณะนี้มีผลให้เกิดผลลัพธ์ที่น่าอัศจรรย์ เช่น การหาทางเดินที่สั้นที่สุดของมด การบินหาอาหารของฝูงนก และการหาแหล่งอาหารที่ดีที่สุดของผึ้ง เป็นต้น โดยที่ผลลัพธ์ที่ยอดเยี่ยมนี้จะไม่มีทางเกิดขึ้นได้ถ้าสมาชิกในกลุ่มไม่ได้ทำงานร่วมกันในลักษณะข้างต้น สิ่งสำคัญก็คือ การทำงานร่วมกันแบบกลุ่มนี้ จะไม่มีสมาชิกตัวใดตัวหนึ่งเป็นหัวหน้าหรือผู้ประสานงาน แต่สมาชิกทุกตัวจะทำหน้าที่ตามกฎง่ายๆ ดังกล่าวเท่านั้น ในหลายกรณี พฤติกรรมของสมาชิกในกลุ่มจะเป็นแบบสถิติอยู่บ้าง ทั้งนี้ขึ้นอยู่กับความน่าจะเป็นของการรับรู้ข้อมูลในท้องถิ่นนั้นของเพื่อนๆ รอบข้าง

สัตว์และแมลงที่อาศัยกันอยู่อย่างเป็นกลุ่ม เช่น นก ค้างคาว มด หรือผึ้ง มักจะถูกเรียกว่าเป็นสัตว์สังคม การทำงานร่วมกันของสัตว์และแมลงภายในฝูงมักจะก่อให้เกิดสิ่งที่น่าประหลาดใจเสมอ ในสัตว์อย่าง เช่น นก หรือค้างคาว ถ้าหากว่ามันบินเป็นฝูง เราจะเห็นพวกมันบินไปด้วยความเร็วสูง โดยที่แต่ละตัวจะกำหนดระยะห่างระหว่างกัน ไม่ว่านกหรือค้างคาวเหล่านั้นจะบินหักเหไปในทิศทางใด หรือด้วยความเร็วเท่าไร เราจะไม่เคยเห็นว่านกหรือค้างคาวเหล่านั้นบินชนกันเองในฝูง ด้วยการกำหนดระยะห่างระหว่างกัน ซึ่งเป็นข้อกำหนดง่ายๆ แต่เมื่อทุกตัวทำตามนั้น เราก็จะเห็นปรากฏการณ์ที่

มหัศจรรย์ ทำให้การบินเป็นฝูงในท้องฟ้ามีรูปแบบที่สวยงาม คล้ายกับการแปรขบวนหรือการเดินสวนสนามของสัตว์บนท้องฟ้า

1.4.1 สติกเมอร์จี

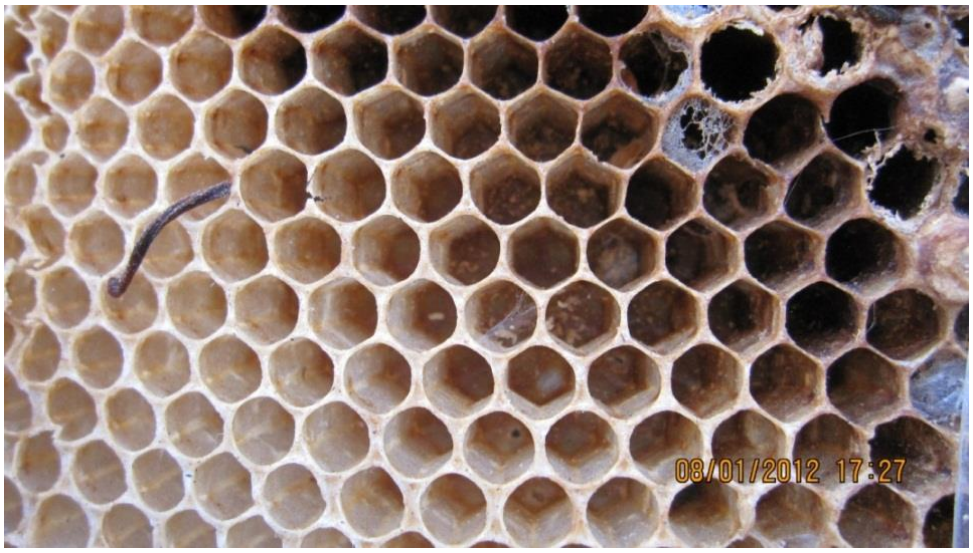
สำหรับแมลงก็เช่นกัน การสังเกตการณ์การทำงานของแมลง ได้มีการศึกษามานานแล้วตั้งแต่ปี 1959 โดยนักกีฏวิทยาชื่อดังชาวฝรั่งเศสชื่อ ปิแอร์ พอล กราเซ (Pierre-Paul Grassé) ได้สังเกตเห็นพฤติกรรมของแมลงที่มีการทำงานประสานงานกันเป็นกลุ่มและมีการสื่อสารระหว่างกัน เขาเรียกการทำงานร่วมกันของแมลงในลักษณะนี้ว่า สติกเมอร์จี (Stigmergy) จากการติดตามพฤติกรรมการทำงานของปลวก เขาพบว่า ปลวกสามารถสร้างรังที่มีความซับซ้อนและมีขนาดใหญ่มาก โดยที่ปลวกไม่มีการวางแผนการทำงานล่วงหน้า ไม่มีหัวหน้าที่คอยควบคุมการทำงาน ในทางตรงกันข้าม ปลวกแต่ละตัวจะช่วยกันทำงาน และแต่ละตัวในฝูงจะรักษากฎเกณฑ์การทำงานอย่างง่ายๆ ด้วยการใช้ฟีโรโมน (Pheromone) ที่อยู่ในน้ำลายเคี้ยวผสมกับดินเพื่อบอกปลวกตัวอื่นๆ ให้มาคายกองดินเพิ่ม ปลวกใช้แรงกระตุ้นอย่างง่ายๆ ที่ขึ้นอยู่กับสภาพแวดล้อมในเวลานั้น กราเซ ตั้งคำว่า สติกเมอร์จี ซึ่งหมายถึง การกระตุ้นให้ทำงาน และกระบวนการนี้ไม่ได้เกิดขึ้นเฉพาะกับปลวกเท่านั้น แต่ยังเกิดขึ้นกับแมลงอีกหลายชนิดเช่น มด ผึ้ง และต่อ ในหลายๆ กิจกรรมจำนวนมากด้วย

กระบวนการสติกเมอร์จี และอิเมอร์เจนซ์ เป็นกระบวนการ 2 กระบวนการที่คล้ายกัน มีความแตกต่างเพียงอย่างเดียวคือ วิธีการส่งผ่านข้อมูลระหว่างกันของประชากรในระบบ ในกรณีของสติกเมอร์จี การส่งผ่านข้อมูลระหว่างประชากรต่างๆ ในระบบ จะต้องผ่านสื่อกลาง เช่นการวางฟีโรโมนไว้กับดินเพื่อการสื่อสารระหว่างกันของมดและปลวก แต่ในอิเมอร์เจนซ์ การส่งผ่านข้อมูล จะเป็นการส่งถึงกันในระหว่างประชากรแต่ละตัวในระบบ เช่นกรณีของสปินแต่ละตัวของแท่งเหล็ก ที่การกระตุ้นเกิดขึ้นกับสปินแต่ละตัวในเหล็กโดยตรง ดังนั้นสติกเมอร์จีจึงเกิดขึ้นเฉพาะในสัตว์หรือแมลง

องค์ประกอบที่สำคัญของการเกิดปัญญาเชิงกลุ่มคือ เซลฟ์ออกาไนเซชัน และสติกเมอร์จี ปัญญาเชิงกลุ่ม เป็นพฤติกรรมเชิงกลุ่ม ที่สมาชิกของกลุ่มจะกระจายกันทำงาน (Decentralized) โดยมีจุดมุ่งหมายเดียวกันแบบระบบเซลฟ์ออกาไนเซชัน ทั้งในทางธรรมชาติและในการประดิษฐ์ โดยทั่วไปแล้วปัญญาเชิงกลุ่มจะประกอบด้วยกลุ่มประชากรของสมาชิกที่มีปฏิสัมพันธ์ระหว่างกันภายในกลุ่ม และกับสภาพแวดล้อมที่กลุ่มของสมาชิกนั้นอยู่

รูปที่ 1.3 การบินของฝูงค้างคาว ตัวอย่างของเซลฟ์ออกาไนเซชัน ของสัตว์(ภาพโดยปรีชา รอดอิ่ม)

แรงดลใจของการศึกษาในเรื่องนี้จะมาจากธรรมชาติ โดยเฉพาะทางด้านชีววิทยา ซึ่งสมาชิกเหล่านี้จะดำเนินการตามกฎง่ายๆ ไม่กี่ข้อ และแม้ว่าจะไม่มีการควบคุมจากส่วนกลางที่คอยกำกับว่าสมาชิกแต่ละตัวจะต้องทำงานอย่างไร การปฏิสัมพันธ์ระหว่างกันนั้นก็ก่อให้เกิด “ความอัจฉริยะ” ขึ้นในองค์รวม โดยที่สมาชิกแต่ละตัวไม่ได้ตั้งใจให้เกิดปรากฏการณ์นั้น ตัวอย่างทางธรรมชาติของปัญญาเชิงกลุ่มได้แก่ การบินของค้างคาว (ดูรูปที่ 1.3) การหาเส้นทางเดินของอาณานิคมมด การสร้างรังของปลวก การบินไปหาแหล่งอาหารของฝูงนก และการสร้างรังผึ้ง (ดูรูปที่ 1.4) เป็นต้น



รูปที่ 1.4 โครงสร้างของรังผึ้งรูปหกเหลี่ยมที่เกิดจากการสร้างของผึ้งแต่ละตัว โครงสร้างลักษณะนี้จะช่วยพยุงให้รังไม่เกิดความเสียหายอันเนื่องมาจากแรงโน้มถ่วง

จากการทำงานของแมลงในรูปแบบของสติกเมอร์จี ทำให้ มาร์โค โดริโก (Bonabeau, 2000) ได้พัฒนาและนำเสนอวิธีการหาค่าเหมาะที่สุดแบบฝูงมด (Ant Colony Optimization) ขึ้นในปี 1992 โดริโก ได้พบรายละเอียดของการหาอาหารของมดว่า มดสามารถหาระยะทางที่สั้นที่สุดในการหาอาหารได้ ด้วยการใช้ฟีโรโมนเช่นกัน มดจะทิ้งฟีโรโมนตามทางบนพื้นดินที่มันเดินผ่าน มดแต่ละตัวจะเลือกเส้นทางที่มีกลิ่นฟีโรโมนแรงมากที่สุดในการเดินทาง ทันทีที่มดพบแหล่งอาหาร มดจะบอกปริมาณและคุณภาพของอาหารด้วยฟีโรโมนนี้ ร่องรอยของฟีโรโมนจะเป็นตัวนำให้มดตัวอื่นๆ เดินไปแหล่งอาหาร โดริโกได้จำลองวิธีการหาอาหารของมดดังกล่าวมาใช้ในการหาค่าเหมาะที่สุด เช่นปัญหาการเดินทางของเซลส์แมน (Travelling Salesman)

เช่นเดียวกันกับเรื่องของมด การศึกษาเรื่องพฤติกรรมของการหาอาหารของฝูงผึ้งก็มีมานานแล้วเช่นกัน คาร์ล วอน ฟริส์ช (Frisch, 1956) ผู้ซึ่งได้รับรางวัลโนเบลสาขาสรีรวิทยาหรือการแพทย์ (The Nobel Prize in Physiology or Medicine) เป็นคนแรกที่ได้อธิบายถึงความสามารถของผึ้งในการสื่อสาร จากการเต้นระบำส่ายท้อง (Waggle Dance) ซึ่งเป็นการสื่อสารให้ผึ้งตัวอื่นได้รับทราบถึงระยะทาง ทิศทาง และคุณภาพของแหล่งอาหาร ที่ตัวมันไปพบมา คาราโบกา (Karaboga, 2005) ได้

อธิบายถึงความสัมพันธ์ขององค์ประกอบของผึ้งและการหาอาหารโดยการแบ่งผึ้งเป็น 3 กลุ่ม โดยกลุ่มแรกเรียกว่า ผึ้งเอ็มพลอย (Employed Bee) จะทำหน้าที่ในการหาอาหาร และนำข้อมูลที่มันได้รับมาไปบอกกล่าวให้กับเพื่อนๆ ถึงคุณภาพของแหล่งอาหาร ซึ่งขึ้นอยู่กับองค์ประกอบหลายอย่างเช่น ระยะห่างจากรัง ความเข้มข้นและจำนวนของพลังงานในแหล่งอาหาร และความยากง่ายในการนำอาหารออกจากแหล่ง กลุ่มที่สองคือ ผึ้งค้นหา (Scouts) กลุ่มนี้จะทำการสำรวจพื้นที่รอบๆ รังของมันเพื่อหาแหล่งอาหารแหล่งใหม่ และกลุ่มที่สามคือ ผึ้งรับสาร (Onlookers) เป็นกลุ่มที่จะรออยู่ที่รังและเลือกแหล่งอาหารจากข่าวสารที่ได้รับมาจากผึ้งหาอาหารที่เป็นผึ้งเอ็มพลอย

ส่วนที่สำคัญที่สุดของการทำให้ผึ้งสามารถเลือกแหล่งอาหารที่ดีคือ การระดมความรู้เกี่ยวกับแหล่งอาหารที่มาจากผึ้งแต่ละตัว ความรู้นี้ได้มาจากการแลกเปลี่ยนมูลข่าวสารระหว่างกันในฝูงผึ้ง จากพื้นที่ในรังผึ้งทั้งหมด ผึ้งจะจัดสรรพื้นที่กลางไว้เป็นเวทีของการแลกเปลี่ยนเรียนรู้ สำหรับผึ้งใช้ในการสื่อสารกันเกี่ยวกับแหล่งอาหารที่ผึ้งแต่ละตัวไปพบมา วิธีการที่ผึ้งใช้ในการสื่อสารกันคือการเต้นรำแบบส่ายท้อง

เนื่องจากมีข้อมูลข่าวสารอยู่เป็นจำนวนมากในพื้นที่ของเวทีเต้นรำ ผึ้งรับสารจะติดตามข้อมูลข่าวสารนี้จากการชมการเต้นรำแบบส่ายท้องของผึ้งเอ็มพลอยที่ละตัว ซึ่งการเต้นของผึ้งที่กลับมาจากแหล่งอาหารนี้จะสัมพันธ์กับคุณภาพและระยะทางของแหล่งอาหารที่มันพบมา ถ้าแหล่งอาหารมีคุณภาพสูง การเต้นของผึ้งก็จะนาน ถ้าคุณภาพของแหล่งอาหารต่ำการเต้นก็จะสั้น สำหรับระยะทางของแหล่งอาหาร ผึ้งจะแสดงออกด้วยขนาดของวงเลขแปดที่มันเต้น โดยที่วงกว้างจะหมายถึงระยะทางไกล การตัดสินใจของผึ้งรับสารจะขึ้นอยู่กับข้อมูลที่ได้จากการเต้นเหล่านี้ และมันก็จะเลือกแหล่งอาหารที่ให้ประโยชน์สูงสุด กระบวนการที่ผึ้งเอ็มพลอยให้ข้อมูลให้กับผึ้งรับสารในการตัดสินใจนี้เรียกว่าการรับสมัครสมาชิก (Recruitment)

จากพฤติกรรมของธรรมชาติ และสัตว์ดังที่กล่าวมาทั้งหมด เราจะเห็นระบบที่เป็น เซลฟ์ออกาไนเซชันที่เกิดจากการทำงานของสัตว์ทั้งฝูง ถ้าจะยกตัวอย่าง ในกรณีของมด การวางฟีโรโมนซึ่งถือว่าเป็นสื่อกลางของการสื่อสารนั้นจะมีทั้งระบบการป้อนกลับเชิงบวก ที่เมื่อมีฟีโรโมนมากขึ้น มดตัวถัดไปก็จะเลือกทางเดินนี้มากขึ้น แล้ววางฟีโรโมนเพิ่มขึ้นอีก สำหรับการป้อนกลังเชิงลบ ก็คือการที่ฟีโรโมนระเหยได้ ในเส้นทางที่มดไม่เลือกเดิน เมื่อเวลาผ่านไปนานขึ้น ฟีโรโมนก็จะค่อยๆ จางไปจนหายหมด

สำหรับการสร้างความแปรปรวนในระบบ ก็จะหมายถึงการที่มดบางตัวเดินสุ่มออกไปหาอาหารยังแหล่งใหม่ๆ โดยที่ไม่สนใจว่าเส้นทางเดินนั้นจะมีฟีโรโมนอยู่หรือไม่ ซึ่งการทำเช่นนี้ของมดก็จะเป็นการประกันว่ามดจะมีแหล่งอาหารที่ดีและอยู่ใกล้รังของมันมากที่สุดเสมอ และสุดท้ายก็แน่นอนที่สุดที่มดจะทำงานกันเป็นกลุ่มที่มีจำนวนมาก เมื่อพิจารณาผลของการทำงานโดยรวมของมดทั้งฝูงแล้ว เราจะเห็นความเป็นอัจฉริยะของการทำงานร่วมกันเป็นฝูงแบบเซลฟ์ออกาไนเซชัน ผลที่ได้ออกมาก็คือปัญญาเชิงกลุ่ม (Swarm Intelligence)

มีข้อที่น่าสังเกตว่า ในระหว่างการทำงานของสัตว์ที่เป็นกลุ่มเช่นมดดังที่กล่าวมา มีกลไกที่สำคัญอยู่ 2 อย่างคือ กระบวนการที่มดเลือกเส้นทางเดินจากความแรงของฟีโรโมนที่อยู่บนเส้นทางนั้น และการที่มีมดบางตัวชอบเดินสุ่มออกไปหาแห่งอาหารใหม่ๆ เสมอ กระบวนการทั้งสองเป็นกลไกที่สำคัญเมื่อเรานำมาประยุกต์ใช้สำหรับการหาค่าที่เหมาะที่สุดที่เรียกว่า เอ็กซ์พล้อยเตชัน และเอ็กซ์พลอเรชัน ซึ่ง เอ็กซ์พล้อยเตชัน หมายถึงกระบวนการที่มดเลือกเส้นทางที่ดีที่สุดจากการดมกลิ่นของฟีโรโมน และเอ็กซ์พลอเรชัน หมายถึงกระบวนการที่มดบางตัวเดินออกไปหาแหล่งอาหารใหม่ เพื่อหาแหล่งอาหารใหม่ๆ ให้กับมดทั้งรัง กลไกทั้งสองเป็นกลไกสำคัญสำหรับการหาค่าเหมาะที่สุดของเมตาฮิวริสติก และการที่มดพบเส้นทางที่สั้นที่สุดจากรังไปยังแหล่งอาหารก็หมายถึงกระบวนการค้นหาได้พบคำตอบที่เหมาะสมแล้ว ซึ่งในเชิงของเมตาฮิวริสติกจะเรียกว่า การลู่เข้าสู่คำตอบ (Convergence)

1.4.2 หลักการของปัญญาเชิงกลุ่ม

ดังที่ได้กล่าวมาหลายครั้งแล้วว่า ปัญญาเชิงกลุ่มมีพื้นฐานมากจากองค์ประกอบหรือประชากรธรรมดาที่อยู่ในระบบทำงานร่วมกันจนเป็นผลให้เกิดผลงานที่ซับซ้อนมาก เมื่อนำเอาผลงานที่ได้จากการทำงานของประชากรแต่ละอันมารวมกันแล้ว ยังสู้ผลงานที่เกิดขึ้นนี้ไม่ได้ ซึ่งการเกิดขึ้นของปัญญาเชิงกลุ่มเป็นผลมาจากสติกเมอร์จีและเซลฟ์ออกาไนเซชัน โดยที่ปัญญาเชิงกลุ่มมีหลักการ 5 อย่าง (Millonas, 1994) ดังนี้

- หลักการแห่งความใกล้ชิด (The Proximity Principle) : ประชากรในระบบจะต้องสามารถจัดการกับปริภูมิและเวลาในการคำนวณ
- หลักการแห่งคุณภาพ (The Quality Principle) : ประชากรในระบบจะต้องสามารถตอบสนองต่อปัจจัยคุณภาพในสภาวะแวดล้อม

- หลักแห่งการตอบสนองที่หลากหลาย (The Principle of Diverse Response) : ประชากรในระบบจะต้องมีความสามารถ ในการตอบสนองที่หลากหลาย และจะไม่มุ่งเน้นไปในกิจกรรมที่มีช่องทางที่แคบ
- หลักการแห่งเสถียรภาพ (The Principle of Stability) : ประชากรในระบบไม่ควรเปลี่ยนโหมดของพฤติกรรมในทุกครั้งที่สภาพแวดล้อมเปลี่ยน
- หลักการของการปรับตัวได้ (The Principle of Adaptability) : ประชากรในระบบจะต้องมีการเปลี่ยนแปลงโหมดของพฤติกรรมในทุกครั้งที่มีปัจจัยใดปัจจัยหนึ่งครอบงำการทำงานของทั้งระบบมากเกินไปจนเป็นผลให้การทำงานแย่ลง

1.4.3 การลู่เข้าสู่คำตอบ เอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน

การลู่เข้าสู่คำตอบ (Convergence) เป็นกระบวนการที่ระบบปรับตัวเข้าสู่ภาวะสมดุลจนเกิดเป็นอิเมอร์เจนซ์ เมื่อนำมาอธิบายกับกระบวนการเมตาฮิวริสติกแล้ว การลู่เข้าสู่คำตอบเป็นกระบวนการของการค้นหาคำตอบที่ คำตอบที่ค้นพบใหม่ในแต่ละรอบ (Iteration) ของการค้นหานั้นจะดีขึ้นมาเรื่อยๆ อย่างมีนัยสำคัญ จนกระทั่งค้นหาพบคำตอบที่เหมาะที่สุด ในการให้คำนิยามอย่างเป็นทางการนั้น การลู่เข้าสู่คำตอบสามารถนิยามได้ว่า

ถ้าค่า p_n เมื่อค่า n มีค่าจาก 0 ถึง ∞ เป็นลำดับของการลู่เข้าสู่คำตอบที่จะไปถึงจุดคงที่ p โดยที่ $p_n \neq 0$ สำหรับทุกค่า n ถ้า λ และ α เป็นค่าที่เกิดขึ้นในสมการ

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}} = \lambda$$

ดังนั้น p_n เมื่อ n เคลื่อนที่จาก 0 ไป ∞ แล้วลู่เข้าสู่จุด p ด้วยลำดับ α ของค่าคงที่ความผิดพลาดอซิมพ์โทติก (Asymptotic Error Constant) λ

โดยทั่วไปแล้ว เราอาจจะกล่าวได้ว่า เมตาฮิวริสติกเป็นยุทธศาสตร์ระดับสูงที่ใช้สำหรับการสำรวจปริภูมิค้นหา (Search Space) ด้วยวิธีการต่างๆ สิ่งสำคัญที่ทำให้การแก้ปัญหาแบบเมตาฮิวริสติกมีประสิทธิภาพและประสิทธิผลสูงคือ การจัดความสมดุลอย่างมีพลวัตระหว่างไดเวอร์สิฟิเคชัน

(Diversification) และ อินเทนสิฟิเคชัน (Intensification) คำว่าไดเวอร์สิฟิเคชัน หมายถึงการสำรวจปริภูมิค้นหา ในขณะที่ อินเทนสิฟิเคชัน หมายถึงการใช้ประโยชน์จากประสบการณ์ในการค้นหาสะสม (C. and Roli, A. Blum, 2003) คำทั้งสองนี้นำมาจากวิธีการค้นหาแบบตาบู (Tabu Search) ในบางครั้งคำว่า เอ็กซ์พลอเรชัน (Exploration) อาจนำมาใช้แทนคำ ไดเวอร์สิฟิเคชัน และคำ เอ็กซ์พล้อยเตชัน (Exploitation) ก็นำมาใช้แทน อินเทนสิฟิเคชัน เหมือนกัน ซึ่งการใช้คำทั้ง 2 ชุดเป็นที่นิยมพอๆ กัน สำหรับหนังสือเล่มนี้ เราจะใช้คำว่า เอ็กซ์พลอเรชัน และ เอ็กซ์พล้อยเตชัน

ดังที่ได้กล่าวมาแล้วว่า การสร้างความสมดุลของ เอ็กซ์พลอเรชัน และ เอ็กซ์พล้อยเตชัน มีความสำคัญมาก เอ็กซ์พลอเรชัน มีหน้าที่ที่สำคัญในการจำแนกบริเวณในปริภูมิค้นหาที่มีคำตอบที่มีคุณภาพสูง ในขณะที่ เอ็กซ์พล้อยเตชัน จะหลีกเลี่ยงการค้นหาในพื้นที่ของปริภูมิค้นหาที่ค้นหาไปแล้วหรือพื้นที่ที่มีคำตอบที่ไม่มีคุณภาพ การค้นหาที่ดำเนินการโดยเมตาฮิวริสติกจะต้องฉลาดพอ ที่จะทำให้กลไกของเอ็กซ์พลอเรชัน ทำการค้นหาบนพื้นที่ใหม่ ในปริภูมิปัญหา ที่มีคำตอบคุณภาพสูงได้ ไม่ว่าเมื่อใดก็ตาม ที่พบว่าบริเวณที่ทำการค้นหาอยู่นั้นมีคำตอบที่มีคุณภาพไม่ดีพอ กลไกของเอ็กซ์พลอเรชันต้องทำการหาพื้นที่ใหม่โดยทันที ในขณะที่เอ็กซ์พล้อยเตชันจะทำการค้นหาคำตอบที่ดีที่สุดในบริเวณที่เอ็กซ์พลอเรชันกำหนดให้ โดยการใช้ประโยชน์จากประสบการณ์ในการค้นหาที่สะสมมา การกำหนดพื้นที่ใหม่ของเอ็กซ์พลอเรชันนั้น ส่วนใหญ่แล้วจะเกิดเป็นบริเวณ เนื่องจากการค้นหาของเอ็กซ์พล้อยเตชันจะนำเอาคำตอบที่เป็นไปได้ที่เป็นเพื่อนบ้านกัน มาเปรียบเทียบกับคำตอบปัจจุบัน เพื่อเลือกคำตอบที่ดีกว่า จุดสำคัญของการสร้างความสมดุลนี้ก็คือ การกำหนดว่า เมื่อไหร่เมตาฮิวริสติกจะทราบว่าเอ็กซ์พล้อยเตชันควรหยุดการค้นหาในพื้นที่นี้ เนื่องจากคำตอบที่ได้ไม่มีคุณภาพดีพอ และเอ็กซ์พลอเรชันควรทำการหาพื้นที่ใหม่ได้แล้ว ปัญหาก็จะอยู่ที่ว่าเราจะทราบได้อย่างไรว่าพื้นที่ที่ทำการค้นหาอยู่นั้นได้คำตอบที่ไม่ดีพอ หรืออาจจะเรียกว่าเป็น โลคอลมินิมั่ม (Local Minimum) และแน่ใจได้อย่างไรว่าพื้นที่ใหม่ที่เอ็กซ์พลอเรชันหามาให้จะมีคำตอบที่ดีกว่า พื้นที่ที่เอ็กซ์พล้อยเตชันกำลังค้นหาอยู่ในปัจจุบัน ถ้าหากว่าเราให้น้ำหนักกับเอ็กซ์พล้อยเตชันมากเกินไป โอกาสที่การค้นหาจะได้คำตอบที่เป็นโลคอลมินิมั่มก็มีมาก แต่ถ้าเราให้น้ำหนักกับเอ็กซ์พลอเรชันมากเกินไป การค้นหาก็จะใช้เวลายาวนานมาก แต่ผลดีคือจะไม่มีโอกาสที่ได้คำตอบที่ไม่ดีเช่นกัน ดังนั้นความสมดุลจึงเป็นเรื่องที่มีความสำคัญดังกล่าวมาแล้ว

ในการลู่เข้าสู่คำตอบ เอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน เป็นสามกระบวนการที่มีผลต่อกัน ผลของการค้นหาจาก เอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน จะต้องมีการปรับสมดุลที่เหมาะสม การลู่เข้าสู่คำตอบจึงจะไม่ไปติดค้างอยู่ที่ โลคอลมินิมั่ม ในแง่ของ เซลฟ์ออกาไนเซชัน กระบวนการเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน จึงเป็นกลไกที่สำคัญที่ทำให้เกิดวิวัฒนาการใหม่ที่เรียกว่า อิเมอร์เจนซ์ ซึ่งเป็นผลจาก เซลฟ์ออกาไนเซชัน ในการหาค่าเหมาะที่สุดกระบวนการเหล่านี้ก็คือกระบวนการค้นหาคำตอบได้ลู่เข้าสู่คำตอบ (Converse) ที่เหมาะสมที่สุด

1.4.4 ฟังก์ชันเทียบเคียง

ในการออกแบบอัลกอริธึมสำหรับการหาค่าเหมาะที่สุดเชิงตัวเลข โดยเฉพาะอย่างยิ่งอัลกอริธึมประเภทปัญญาเชิงกลุ่ม การหาสมรรถนะของอัลกอริธึมเป็นเรื่องที่สำคัญ เพื่อที่จะพิสูจน์ให้ได้ว่าอัลกอริธึมที่ออกแบบมานั้นสามารถแก้ปัญหาได้จริงในหลายรูปแบบ โดยเฉพาะปัญหาที่แก้ยาก ซึ่งหมายถึงปัญหาที่ปริภูมิปัญหามีความซับซ้อนและมีขนาดใหญ่มาก โดยเฉพาะอย่างยิ่งเมื่อปัญหาเหล่านั้นมีกับดักที่เป็นโลคอลมินิมั่ม หลายรูปแบบ

ในการหาสมรรถนะดังที่กล่าวมานั้น เป็นเรื่องยากที่จะหาตัวอย่างจริงมาเป็นบททดสอบ เพราะในตัวอย่างจริง บางครั้งเราก็ไม่ทราบคำตอบที่แท้จริง และก็อาจจะเป็นการยากที่จะรู้ว่าตัวอย่างของปัญหาแบบไหนมีความยากง่ายอย่างไร เพื่อแก้ปัญหาดังกล่าวข้างต้น นักวิจัยในกลุ่มนี้จึงคิดค้นวิธีการหาสมรรถนะของอัลกอริธึมด้วยนำฟังก์ชันคณิตศาสตร์ ที่รู้คำตอบที่ดีที่สุดแล้ว และทราบลักษณะของปริภูมิปัญหาว่ามีรูปร่างอย่างไร เช่น มี โลคอลมินิมั่ม มากน้อย จึงถูกนำมาใช้ มีผู้ออกแบบฟังก์ชันคณิตศาสตร์เพื่อการทดสอบจำนวนมาก เรียกว่าฟังก์ชันเทียบเคียง (Benchmark Functions)

ฟังก์ชันที่ถูกออกแบบมาสำหรับการทดสอบอัลกอริธึมนั้น จึงมักจะต้องออกแบบมาเฉพาะ และจะต้องมี โลคอลมินิมั่ม (Local Minimum) ประกอบกับโกลบอลมินิมั่ม (Global Minimum) อัลกอริธึมที่ดีจะต้องหา โกลบอลมินิมั่ม ให้พบ ในขณะที่อัลกอริธึมพบ โลคอลมินิมั่ม ก็จะต้องไม่ติดกับ และหลงคิดว่าค่าที่ได้เป็นค่าที่ดีที่สุดแล้ว มีผู้ที่คิดฟังก์ชันและวิธีการทดสอบอยู่จำนวนมาก สำหรับฟังก์ชันที่ออกแบบมาแล้ว มีคุณสมบัติที่น่าสนใจคือ ฟังก์ชันยูนิโมเดล (Unimodal Function) หมายถึงฟังก์ชันที่มีลักษณะของปริภูมิการค้นหา ที่ราบเรียบ ไม่มีโลคอลมินิมั่ม มีแต่ โกบอล มินิมั่ม แต่ถ้าฟังก์ชันใดมีโล

คอลมินิมั่มมากกว่าหนึ่งจุด และมีโกลบอลมินิมั่มอยู่ด้วย ฟังก์ชันนั้นจะถูกเรียกว่าเป็น มัลติโมเดล (Multimodal Function)

ฟังก์ชันแยกได้ (Separable Function) ฟังก์ชันประเภทนี้พิจารณาจากตัวแปรที่อยู่ภายในฟังก์ชันนั้น ถ้าตัวแปรทุกตัวในฟังก์ชันเป็นอิสระต่อกัน (Independent) การแก้ปัญหาทำได้โดยการแยกฟังก์ชันให้เป็นปัญหาย่อย (Sub-problem) ที่ประกอบด้วยตัวแปรแต่ละตัว แล้วแก้ปัญหาย่อยเหล่านั้นทีละตัว โดยกำหนดให้ตัวแปรที่เหลือเป็นค่าคงที่ เราก็สามารถหาค่าเหมาะที่สุดของฟังก์ชันนั้นได้ สำหรับปัญหาที่ตัวแปรในฟังก์ชันจะต้องขึ้นต่อกัน หมายถึงฟังก์ชันนั้นมีตัวแปรที่มีความสัมพันธ์ต่อกัน ฟังก์ชันประเภทนี้จะถูกเรียกว่า ฟังก์ชันที่ไม่สามารถแยกได้ (Non-separable Function) ตัวอย่างต่อไปนี้ดังตารางที่ 1.1 จะแสดงฟังก์ชันในลักษณะต่างๆ

- ฟังก์ชันรูปทรงกลม (Sphere Function)
- ฟังก์ชันของโรเซนบร็อก (Rosenbrock's Function)
- ฟังก์ชันของกรีวังค์ (Griewangk's Function)
- ฟังก์ชันของเรสตริจิน (Rastrigin's Function)
- ฟังก์ชันของแอคเลย์ (Ackley's Function)
- ฟังก์ชันของเชฟเฟอร์ (Schaffer's Function)

จากตารางที่ 1.1 ฟังก์ชันรูปทรงกลม (Sphere function) เป็นฟังก์ชันที่ถูกนำเสนอโดย เด จอง (De Jong, 1988) เป็นฟังก์ชันอย่างง่ายที่เป็นแบบคอนเวค (Convex) มีจุดที่ดีที่สุดหรือ โกลบอลมินิมั่มเพียงจุดเดียว โรเซนบร็อก (H.H Rosenbrock, 1960) คล้ายกับฟังก์ชันรูปทรงกลม ที่ไม่มี โลคอลมินิมั่ม แต่มีโกลบอลมินิมั่ม ยาวเรียงกันคล้ายหุบเหวเป็นรูปพาราโบล่า (Parabola) ของสมการ $y = x^2$ เนื่องจาก โกลบอลมินิมั่ม ที่เรียงตัวกันไม่เป็นเส้นตรงนี้ ทำให้อัลกอริธึมหลายอันลู่เข้าสู่คำตอบช้าเพราะการเปลี่ยนทิศทางของการค้นหาบ่อยๆ

ชื่อฟังก์ชัน	นิยาม	มัลติโมเดล
ทรงกลม	$f(x) = \sum_{i=1}^{n} x_i^2$	ไม่ใช่
ขอบเขต	$-5.12 \le x_i \le 5.12$, i = 1, 2, ... , n.	

ค่าดีที่สุด	$x^* = (0, ..., 0), f(x^*) = 0.$	
โรเซนบร็อก	$f(x,y) = (1-x)^2 + 100(y-x^2)^2$	ไม่ใช่
ขอบเขต	$x_i \in [-2.048, 2.048]$	
ค่าดีที่สุด	$(x, y) = (1,1), f(x, y) = 0.$	
กรีวังค์	$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	ใช่
ขอบเขต	$x_i \in [-600, 600]$	
ค่าดีที่สุด	$x^* = (0,...,0), f(x^*) = 0.$	
เรสตริจิน	$f(x) = An + \sum_{i=1}^{n} [x_i^2 - A\cos(2\pi x_i)]$	ใช่
ขอบเขต	$x_i \in [-5.12, 5.12]$	
ค่าดีที่สุด	$x = 0, f(x) = 0.$	
แอคเลย์	$f(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)}$	ใช่
ขอบเขต	$-15 \le x_i \le 30, i = 1, 2, \ldots, n.$	
ค่าดีที่สุด	$x^* = (0,...,0), f(x^*) = 0.$	
เชฟเฟอร์	$f(x) = 0.5 + \frac{\{\sin(\sqrt{x_0^2 + x_1^2})\}^2 - 0.5}{\{1.0 + 0.001(x_0^2 + x_1^2)\}^2}$	ใช่
ขอบเขต	$\lvert x_i \rvert \le 100$	
ค่าดีที่สุด	$x = 0, f(x^*) = 0.$	

ตารางที่ 1.1 ฟังก์ชันที่ใช้สำหรับการเทียบเคียงสมรรถนะ

เรสตริจิน (L.A. Rastrigin, 1974) สร้างขึ้นมาจากฟังก์ชันรูปทรงกลมที่เพิ่มตัวมอดูเลเตอร์ (Modulator) ฟังก์ชัน $\alpha\cos(2\pi x_i)$ เข้าไป ทำให้ที่พื้นผิวของกราฟมีโลคอลมินิมั่ม เกิดขึ้นจำนวนมหาศาลตลอดระยะทางไปจนถึงโกลบอลมินิมั่ม

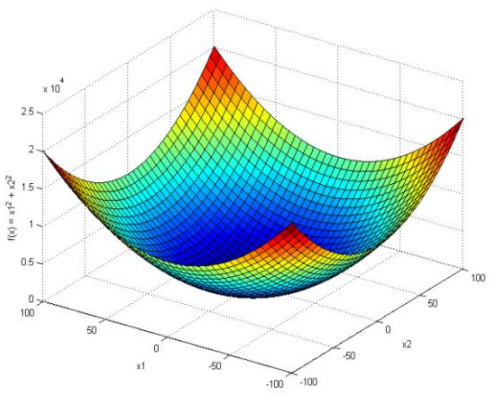
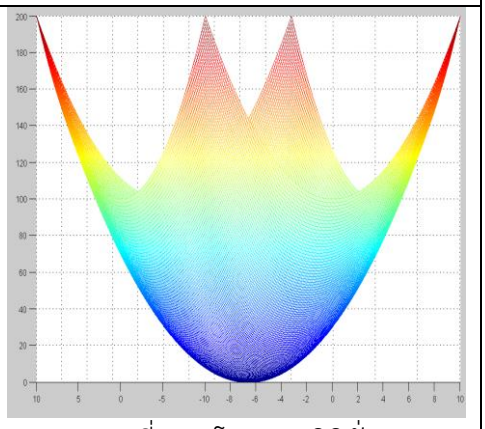
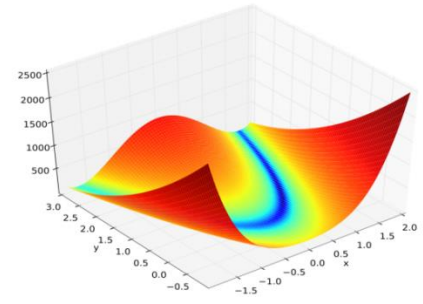
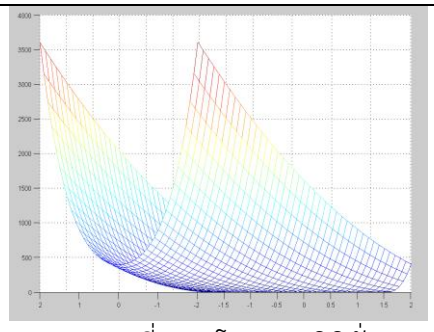
ฟังก์ชันของเรสตริจินและฟังก์ชันของแอคเลย์ (D. Ackley, 1987) มีลักษณะที่คล้ายกันคือมีจำนวนโลคอลมินิมั่มจำนวนมาก และรูปแบบของการเกิดโลคอลมินิมั่ม จะคล้ายกันหมดทั้งกราฟ ในปริภูมิการค้นหาของทั้งสองฟังก์ชัน สำหรับกรณีของฟังก์ชันของเชฟเฟอร์ (J.D. Schaffer, 1987) นั้น

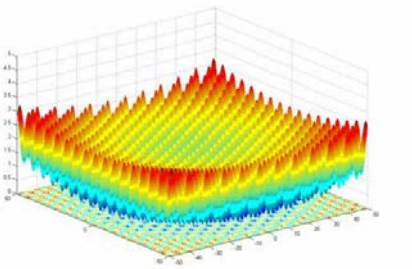
รูปแบบของการเกิดโลคอลมินิมั่ม จะต่างกันอย่างสิ้นเชิง โดยเฉพาะรูปแบบของการเกิดโลคอลมินิมั่มในปริภูมิการค้นหาส่วนที่อยู่ตรงขอบ ซึ่งจะเป็นจุดที่ยากมากในการค้นหาโกลบอลมินิมั่ม ที่อยู่ในส่วนกลางของกราฟ ถ้าหาการค้นหาเริ่มต้นที่จุดที่อยู่ตรงชายขอบ การหาจุดที่เป็น โกลบอลมินิมั่ม จะทำได้ยากมาก การหาคำตอบสำหรับการแก้ฟังก์ชันของเชฟเฟอร์นั้นต้องประสานการทำงานของเอ็กซ์พลอเรชันและเอ็กซ์พล้อยเตชันอย่างดี

จากตารางที่ 1.2 กราฟที่วาด 2 มิติ (D=2) ตามที่แสดงข้างต้นของฟังก์ชันต่างๆ จะเห็นว่าฟังก์ชันรูปทรงกลม (Sphere function) จะไม่มีโลคอลมินิมั่ม และมีโกลบอลมินิมั่มเพียงจุดเดียว ฟังก์ชันในลักษณะนี้จะเรียกว่า ยูนิโมเดล (Unimodal) สำหรับกราฟของโรเซนบร็อกก็จะไม่มี โลคอลมินิมั่มเช่นกัน แต่โกลบอลมินิมั่มจะมีหลายจุด เป็นแนวคล้ายหุบเขา มีรูปร่างเป็นพาราโบล่า (Parabola) ฟังก์ชันในลักษณะนี้จะเรียกว่า ยูนิโมเดล เช่นกัน สำหรับฟังก์ชันนี้ ถ้าหากว่าเราเพิ่มค่า D > 2 กราฟของฟังก์ชันนี้จะกลายเป็น มัลติโมเดล (Multimodal) ทันที สำหรับฟังก์ชันที่เหลือทั้งหมด จะเป็นแบบมัลติโมเดล ซึ่งหมายถึงฟังก์ชันที่มี โลคอลมินิมั่ม คอยดักเอาไว้ สำหรับการทดสอบการทำงานของอัลกอริธึมนั้น แต่ละฟังก์ชันจะมีลักษณะที่ไม่เหมือนกัน ฟังก์ชันรูปทรงกลมจะเป็นการทดสอบที่ง่ายที่สุด ไม่มีโลคอลมินิมั่ม ที่จะทำให้ได้คำตอบที่ผิด สำหรับฟังก์ชันที่เหลือ จะมีโลคอลมินิมั่ม ทั้งหมด ในกรณีที่เมื่อจำนวนมิติของการค้นหามากกว่าสองมิติ (และการแก้ปัญหาในทางปฏิบัติก็มักจะเป็นเช่นนั้นด้วย) เช่นการจัดตารางการทำงานของเครื่องจักรในการผลิตสินค้าของระบบการผลิต ความยากของการค้นหาก็จะมีลักษณะที่ต่างกัน เช่นฟังก์ชันกรีวังค์ ฟังก์ชันของเรสตริจิน และฟังก์ชันของแอคเลย์ จะมีจำนวนโลคอลมินิมั่มจำนวนมาก แต่รูปแบบของการเกิดโลคอลมินิมั่ม จะคล้ายกันหมดทั้งกราฟ สำหรับกรณีของฟังก์ชันของเชฟเฟอร์นั้น รูปแบบของการเกิดโลคอลมินิมั่ม จะต่างกันอย่างสิ้นเชิง โดยเฉพาะรูปแบบของกราฟในส่วนที่อยู่ตรงขอบ ถ้าการค้นหาเริ่มต้นที่จุดนั้น การหาจุดที่เป็น โกลบอลมินิมั่ม จะทำได้ยากมาก

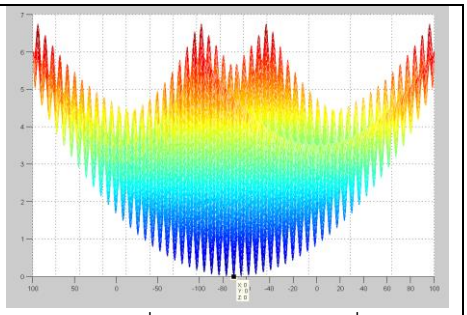
ความยากของการหาค่าที่เหมาะที่สุดกับฟังก์ชันที่มีความซับซ้อนดังกล่าว จะยากขึ้นไปอีกเมื่อจำนวนมิติของปัญหานั้นมีมากขึ้น เหตุผลของการเกิดปัญหาดังกล่าวมีอยู่สองประการคือ ประการแรก จำนวนคำตอบในปริภูมิค้นหา จะมีขนาดใหญ่ขึ้น (หรือมากขึ้น) ในลักษณะของเอ็กซ์โพเนนเชียล (Exponential) เมื่อจำนวนมิติของปัญหามากขึ้น ดังนั้นยุทธศาสตร์ในการค้นหาที่มีประสิทธิภาพจึงมี

ความจำเป็น เพื่อทำให้ได้คำตอบในเวลาที่กำหนด ประการที่สอง ลักษณะของปัญหาจะเปลี่ยนไป เมื่อขนาดของปัญหาใหญ่ขึ้น เช่นในกรณีของฟังก์ชันของโรเซนบร็อก เมื่อจำนวนมิติของปัญหาเพิ่มขึ้น รูปแบบของปัญหาก็จะเปลี่ยนจากยูนิโมเดล เป็นมัลติโมเดล สำหรับปัญหาที่ยากขึ้น ตัง (K., Li, X., Suganthan, P.N., Yang, Z. and Weise, T. Tang, 2010) ได้สร้างวิธีในการจำแนกชนิดของฟังก์ชันเป็น ฟังก์ชันที่แยกได้ (Separable Function) และฟังก์ชันที่แยกไม่ได้ (Nonseparable Function) และทำฟังก์ชันทดสอบให้ครอบคลุมลักษณะของฟังก์ชันดังกล่าวด้วยการเพิ่มมิติของตัวแปรในฟังก์ชันทดสอบที่กล่าวมาข้างต้นให้มีมิติที่ซับซ้อนขึ้น โดยให้อยู่ในรูปแบบของ ออโธโกนอล เมตริกซ์ (Orthogonal Matrix) และการเปอร์มิวเตชันแบบสุ่ม (Random Permutation)

1. ฟังก์ชันรูปทรงกลม	
	
รูปแบบของปริภูมิการค้นหา	จุดที่แสดงโกลบอลมินิมั่ม
2. ฟังก์ชันโรเซนบร็อก	
	
รูปแบบของปริภูมิการค้นหา	จุดที่แสดงโกลบอลมินิมั่ม
3. ฟังก์ชันกรีวังค์	

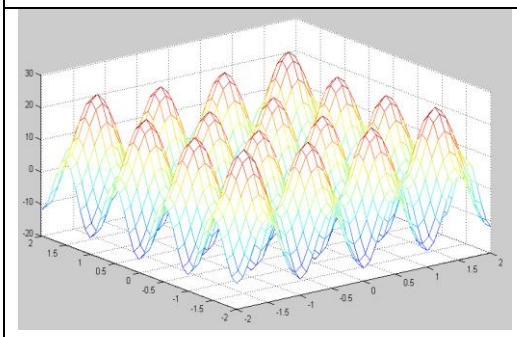


รูปแบบของปริภูมิการค้นหา

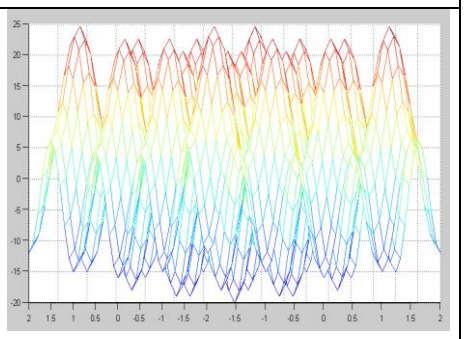


จุดที่แสดงโกลบอลมินิมั่ม

4. ฟังก์ชันเรสตริจิน

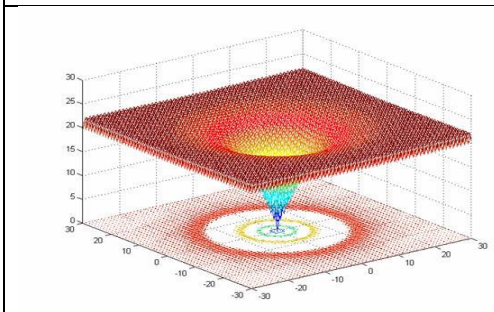


รูปแบบของปริภูมิการค้นหา

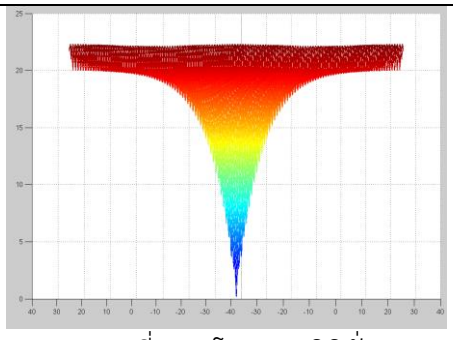


จุดที่แสดงโกลบอลมินิมั่ม

5. ฟังก์ชันแอคเลย์

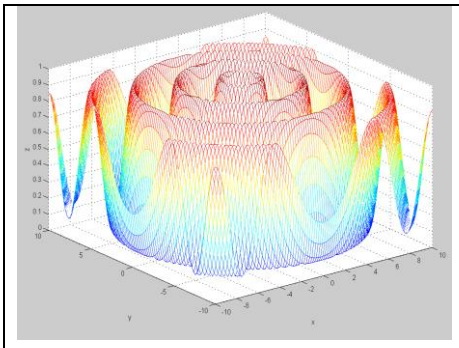


รูปแบบของปริภูมิการค้นหา

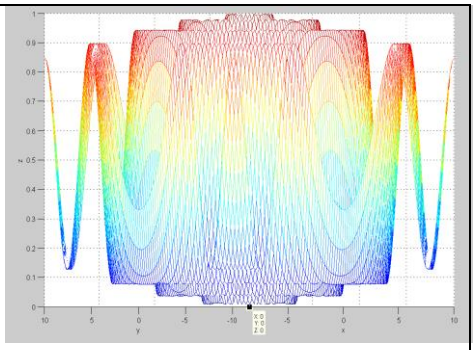


จุดที่แสดงโกลบอลมินิมั่ม

6. ฟังก์ชันเชฟเฟอร์



รูปแบบของปริภูมิการค้นหา



จุดที่แสดงโกลบอลมินิมั่ม

ตารางที่ 1.2 ภาพสามมิติของฟังก์ชันการเทียบเคียงสมรรถนะ

1.5 สรุป

วิธีการของปัญญาเชิงกลุ่มมักนำมาใช้กับการหาค่าเหมาะที่สุด (Optimization) ปัญหาการหาค่าเหมาะที่สุดคือ ปัญหาที่เกี่ยวกับการหาค่าที่ดีที่สุดจากคำตอบที่เป็นไปได้ทั้งหมด (Possible Solution) การจำแนกชนิดของการหาค่าที่เหมาะที่สุดจะพิจารณาจากค่าของตัวแปรว่าเป็นแบบไม่ต่อเนื่องกันหรือเป็นแบบต่อเนื่องกัน ปัญหาที่มีตัวแปรแบบไม่ต่อเนื่องกันจะเรียกว่า การหาค่าเหมาะที่สุดแบบคอมบินาทอเรียล สำหรับแบบต่อเนื่องกันจะเรียกว่า การหาค่าเหมาะที่สุดเชิงตัวเลข ตัวอย่างของปัญหาแบบคอมบินาทอเรียล ได้แก่ การเดินทางของเซลส์แมน และปัญหาการจัดตาราง ปัญหาเหล่านี้จัดว่าเป็นปัญหาที่ยากและมีความซับซ้อนมาก วิธีสำหรับการแก้ปัญหาดังกล่าวสามารถจำแนกได้เป็นการแก้แบบสมบูรณ์และการแก้ด้วยการประมาณการ การ**แก้ปัญหาแบบสมบูรณ์**นั้นเป็นการหาคำตอบที่จะได้เป็นค่าที่ดีที่สุดแน่นอน ดังนั้นขอบเขตปัญหาจึงมีขนาดเล็กเช่น การหาเส้นทางที่สั้นที่สุดของการเดินทางไปยังเมือง 4 เมือง สำหรับปัญหาที่มีขนาดใหญ่ขึ้นจะต้องอาศัย**การประมาณการ** สำหรับ**การหาค่าเหมาะที่สุดเชิงตัวเลข** คือปัญหาที่ต้องการหาค่าที่เป็นตัวเลขเพื่อเป็นคำตอบของสมการ ซึ่งอาจจะหาได้จากการประมาณการเพราะหากต้องแก้สมการด้วยมือหรือกรรมวิธีอื่นอาจใช้เวลานาน

วิธีการแก้ปัญหาด้วยการประมาณการ หรือ ฮิวริสติก (Heuristic) เป็นการหาคำตอบให้กับการหาค่าเหมาะที่สุดสำหรับปัญหาที่มีขอบเขตของปริภูมิการค้นหากว้าง ในขณะที่การหาคำตอบด้วยวิธีการของคอมพิวเตอร์แบบธรรมดาทำไม่ได้

การแก้ปัญหาด้วยวิธีการประมาณการ โดยทั่วไปสามารถแบ่งได้เป็น 2 ประเภทคือ คอนสตรัคทีฟ และการค้นหาแบบโลคอล อัลกอริธึมแบบคอนสตรัคทีฟจะสร้างคำตอบที่เป็นไปได้จากเริ่มต้น เมื่อเริ่มต้น อัลกอริธึมจะเพิ่มองค์ประกอบให้กับคำตอบทีละส่วน และทำซ้ำไปเรื่อยๆ จนกระทั่งได้คำตอบออกมา

สำหรับอัลกอริธึมการค้นหาแบบโลคอลนั้น จะเริ่มด้วยคำตอบเบื้องต้นค่าใดค่าหนึ่ง คำตอบเบื้องต้นนี้อาจจะมาจากการสุ่ม หรืออาจจะใช้วิธีการอื่นสร้างคำตอบขึ้นมาและกำหนดให้คำตอบนี้เป็นคำตอบปัจจุบัน จากนั้นให้แทนที่คำตอบปัจจุบันด้วยคำตอบที่เลือกมาจากคำตอบข้างเคียงของคำตอบปัจจุบันที่ดีกว่า วิธีการค้นหาแบบโลคอลนี้ปัจจุบันเรียกว่า เมตาฮิวริสติก (Metaheuristics)

เมตาฮิวริสติกถูกออกแบบมาสำหรับการแก้ปัญหาการหาค่าเหมาะที่สุดที่มีความซับซ้อน โดยที่ให้ทั้งประสิทธิภาพที่สูง และมีประสิทธิผลที่ดี ในขณะที่วิธีการอื่นๆ ไม่สามารถทำได้ สิ่งที่เป็นประโยชน์ในการปฏิบัติเป็นอย่างมากสำหรับวิธีการของเมตาฮิวริสติกก็คือ ตัวมันเองให้ประสิทธิผลที่ดีและสามารถปรับใช้ได้กับปัญหาการหาค่าเหมาะที่สุดได้ทุกรูปแบบ วิธีการทางเมตาฮิวริสติกจะใช้กระบวนการทางธรรมชาติที่สามารถสร้างวิวัฒนาการให้กับตนเองได้มาเป็นแนวทางในการออกแบบสร้างเมตาฮิวริสติก วิธีการนั้นคือ เซลฟ์ออกาไนเซชัน

เซลฟ์ออกาไนเซชัน เป็นกระบวนการที่เป็นพลวัตที่ระบบจะได้มาและรักษาโครงสร้างของมันเอง โดยไม่มีการควบคุมจากภายนอก ซึ่งหมายถึงไม่มีการกำหนดทิศทางการทำงาน การจัดการ การแทรกแซง การผลักดัน และการมีส่วนเกี่ยวข้องจากสิ่งที่อยู่ภายนอกระบบ ทั้งนี้ไม่รวมถึงการนำเข้าข้อมูลจากภายนอกระบบตราบใดที่ข้อมูลเหล่านั้นไม่เป็นคำสั่งควบคุม (Control instructions) โดยมีหลักการพื้นฐานอยู่ 4 ประการ

- การป้อนกลับเชิงบวก เป็นการเสริมการกระทำของระบบ
- การป้อนกลับเชิงลบ เป็นการปรับสมดุล และสร้างความมั่นคงให้กับระบบ

- การขยายของความแปรปรวน เช่นความไม่แน่นอน ความผิดพลาด และการสุ่ม
- การกระทำที่ทำร่วมกันขององค์ประกอบในองค์กรทั้งหมด

สติกเมอร์จี เป็นรูปแบบหนึ่งของ เซลฟ์ออกาไนเซชัน ที่อาศัยการวางฟีโรโมน ในสภาพแวดล้อมมาเป็นตัวช่วยให้การทำงานเชิงกลุ่มของแมลงเกิดพลัง

ปัญญาเชิงกลุ่ม เป็นผลจากพฤติกรรมของ เซลฟ์ออกาไนเซชัน สัตว์หลายชนิดที่มีพฤติกรรมเชิงกลุ่มแบบเซลฟ์ออกาไนเซชัน แล้วทำให้เกิดเป็นปัญญาที่ฉลาดในการแก้ปัญหาต่างๆ ได้ สัตว์ที่มีพฤติกรรมเช่นนี้ได้แก่ นก ค้างคาว มด ปลวก และผึ้ง เป็นต้น

ในการจำลองพฤติกรรมของสัตว์มาใช้ในการหาค่าที่เหมาะที่สุดนั้น เนื่องจากพฤติกรรมของสัตว์จะมีวิธีการแก้ปัญหาด้วยการหาคำตอบเบื้องต้นที่เป็นไปได้ก่อน ในลักษณะของเมตาฮิวริสติก จากนั้นค่อยๆ พัฒนาคำตอบไปเรื่อยๆ ให้คำตอบที่ดีขึ้นจนกระทั่งได้คำตอบที่ดีที่สุดเท่าที่เป็นไปได้ วิธีการที่การค้นหาคำตอบค่อยๆ พัฒนาไปสู่คำตอบที่ดีกว่านี้จะเรียวกว่า **การลู่เข้าสู่คำตอบ** ซึ่งการลู่เข้าสู่คำตอบเพื่อให้ได้ค่าที่ดีที่สุดเท่าที่เป็นไปได้นั้น อัลกอริธึมของการค้นหาจะต้องหลีกเลี่ยงการลู่เข้าสู่คำตอบก่อนเวลาอันควร (Immature Convergence) และจะต้องหลีกเลี่ยง โลคอลมินิมั่ม ด้วย วิธีการเมตาฮิวริสติกที่จำลองมาจากการทำงานของสัตว์นั้น จะมีเทคนิคที่สำคัญสองอย่างคือ เอ็กซ์พลอเรชัน และ เอ็กซ์พล้อยเตชัน

เอ็กซ์พลอเรชัน มีหน้าที่สำคัญในการจำแนกบริเวณในปริภูมิค้นหาที่มีคำตอบที่มีคุณภาพสูง จะหลีกเลี่ยงการค้นหาในพื้นที่ของปริภูมิค้นหาที่ค้นหาไปแล้ว หรือพื้นที่ที่มีคำตอบที่ไม่มีคุณภาพ ด้วยการไปสำรวจหาพื้นที่ใหม่เพื่อการหาคำตอบ ในขณะที่ เอ็กซ์พล้อยเตชัน เป็นการหาคำตอบที่ดีที่สุดในบริเวณที่กำหนดเฉพาะ เมตาฮิวริสติกที่ดีจะต้องฉลาดพอที่จะทำให้กลไกของเอ็กซ์พลอเรชันทำการค้นหาของพื้นที่ในปริภูมิปัญหาที่มีคำตอบที่มีคุณภาพสูงให้ได้ แต่ถ้าเมื่อใดที่พบว่าบริเวณที่ทำการค้นหาอยู่นั้นมีคำตอบที่มีคุณภาพไม่ดีพอ กลไกของเอ็กซ์พลอเรชันต้องทำการหาพื้นที่ใหม่โดยทันที ในขณะที่เอ็กซ์พล้อยเตชันจะทำการค้นหาคำตอบที่ดีที่สุดในบริเวณที่เอ็กซ์พลอเรชันกำหนดให้

วิธีการเปรียบเทียบสมรรถนะการทำงานของเมตาฮิวริสติกแต่ละอันว่าเมตาฮิวริสติกใดทำงานดีกว่าเมตาฮิวริสติกอื่นอย่างไรนั้น ต้องอาศัยฟังก์ชันเทียบเคียง เช่น ฟังก์ชันรูปทรงกลม ฟังก์ชันของโรเซนบร็อก ฟังก์ชันของกรีวังค์ ฟังก์ชันของเรสตริจิน ฟังก์ชันของแอคเลย์ และฟังก์ชันของเชฟเฟอร์ เป็นต้น

1.6 แบบฝึกหัด

1. การหาค่าเหมาะที่สุดคืออะไร และมีการจำแนกเป็นกี่ชนิด แต่ละชนิดมีวิธีการแก้ปัญหาที่แตกต่างกันอย่างไร
2. เหตุใดการหาเส้นทางที่สั้นที่สุดของการเดินทางของเซลส์แมนจึงเป็นการหาค่าเหมาะที่สุด โดยเฉพาะในกรณีที่มีเมืองของการเดินทางจำนวนมาก
3. วิธีการแบบฮิวริสติกคืออะไร และใช้สำหรับการหาเส้นทางที่สั้นที่สุดของเซลส์แมนได้อย่างไร
4. เมตาฮิวริสติกคืออะไร และต่างจากฮิวริสติกอย่างไร ในแง่ของการการหาค่าที่เหมาะที่สุด
5. อธิบายและให้รายละเอียดกรอบการทำงานของเมตาฮิวริสติก
6. หลักการพื้นฐานของ เซลฟ์ออกาไนเซชัน 4 ประการมีอะไรบ้าง
7. การปฏิสัมพันธ์กันของประชากรในระบบ เหตุใดจึงเป็นปฏิกิริยาท้องถิ่น และปฏิกริยานี้มีผลต่อระบบโดยรวมอย่างไร
8. เหตุใดเซลฟ์ออกาไนเซชัน จึงเป็นระบบที่ทนทานต่อการเปลี่ยนแปลง และถ้าระบบได้รับผลกระทบจากสิ่งแวดล้อมภายนอกจนระบบเสียความสมดุล ระบบการถ่ายเทความร้อนของเรย์ลี เบนนาร์ด จะมีวิธีการฟื้นตัวอย่างไร
9. สภาพแวดล้อมคืออะไร และมีผลต่อเซลฟ์ออกาไนเซชันอย่างไร
10. อธิบายกระบวนการของ เซลฟ์ออกาไนเซชัน ด้วยวิธีการของเมตาฮิวริสติก โดยใช้ตัวอย่างของสปินในเหล็ก และการพาความร้อนของน้ำ
11. อธิบายกลไกของ เซลฟ์ออกาไนเซชัน จากตัวอย่างดังต่อไปนี้

11.1. การที่ฝูงมดเดินทางจากรังไปหาอาหาร มดหาวิธีที่สั้นที่สุดในการเดินทางได้อย่างไร วิธีการของฝูงมดนี้เป็น เซลฟ์ออกาไนเซชัน หรือไม่อย่างไร

11.2. ท่านคิดว่าการที่ค้างคาวบินไปหาอาหาร เป็นเซลฟ์ออกาไนเซชัน หรือไม่ อย่างไร

12. ปัญญาเชิงกลุ่ม มีองค์ประกอบหลักที่สำคัญคืออะไร และมีหลักการอย่างไร

13. คำว่า ประชากรและสภาพแวดล้อมของปัญญาเชิงกลุ่มอะไรบ้าง และมีความสำคัญต่อระบบอย่างไร

14. อธิบายความหมายและกลไกของปัญญาเชิงกลุ่ม จากตัวอย่างดังต่อไปนี้

14.1. การหาอาหารของมดเป็นปัญญาเชิงกลุ่มหรือไม่ อย่างไร

14.2. การหาอาหารของผึ้งนับว่าเป็นปัญญาเชิงกลุ่มที่มีความสามารถในการเลือกแหล่งอาหารที่ดี ผึ้งมีวิธีการอย่างไร

15. อธิบายและให้ความหมายของคำดังต่อไปนี้

15.1. อิเมอร์เจนซ์ (Emergence)

15.2. สติกเมอร์จี (Stigmergy)

15.3. ไบฟาร์เคชัน (Bifurcation)

16. อธิบายความหมายและความสัมพันธ์ของคำดังต่อไปนี้

16.1. เอ็กซ์พลอเรชัน (Exploration)

16.2. เอ็กซ์พล้อยเตชัน (Exploitation)

16.3. การลู่เข้าสู่คำตอบ (Convergence)

17. ให้บอกถึงบุคคลสำคัญที่บุกเบิกแนวคิดของเรื่องเหล่านี้ พร้อมทั้งให้รายละเอียดของเรื่องที่ค้นพบ

17.1. เซลฟ์ออกาไนเซชัน

17.2. เมตาฮิวริสติก

17.3. ปัญญาเชิงกลุ่ม

18. ฟังก์ชันเทียบเคียงมีความสำคัญอย่างไร และใช้ประโยชน์ในแง่ใดบ้าง

19. ให้ใช้ Mathlab เขียนกราฟของสมการฟังก์ชันต่างๆ ในตารางที่ 1.1 แล้วให้หาค่าของโกลบอลมินิมั่ม จากการสังเกตุจากกราฟ

บทที่ 2 การค้นหาแบบฮิวริสติก

Heuristic Search

2.1 คำนำ

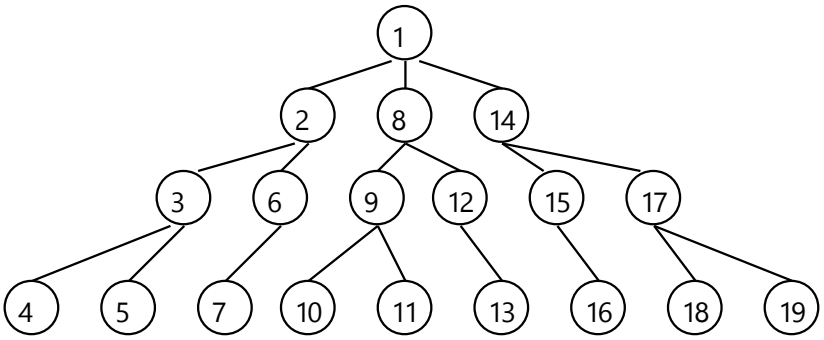
การค้นหาคำตอบหรือการค้นหาข้อมูลทางคอมพิวเตอร์ มักจะกระทำบนโครงสร้างข้อมูลแบบต้นไม้และกราฟ ทั้งนี้เพราะโครงสร้างข้อมูลในลักษณะนี้สามารถทำให้การค้นหาสะดวก และสามารถพลิกแพลงการค้นหาได้ง่าย ในความเป็นจริงแล้ว การค้นหาข้อมูลบางครั้งสามารถกระทำบนโครงสร้างข้อมูลชนิดอื่นก็ได้ เช่น เชิงเส้น (Linear) แต่การจัดโครงสร้างข้อมูลแบบเชิงเส้นมีข้อจำกัด เพราะการค้นหาด้วยวิธีการดังกล่าวเป็นแบบเรียงลำดับ (Sequential Search) เท่านั้น และใช้ได้กับข้อมูลที่มีขนาดเล็ก ดังนั้นเพื่อความสะดวกในการค้นหาข้อมูลที่มีขนาดใหญ่ ก่อนการค้นหาหรือระหว่างการค้นหา ควรจัดข้อมูลทั้งหมดให้อยู่ในรูปแบบของต้นไม้หรือกราฟก่อน การค้นหาข้อมูลบนโครงสร้างต้นไม้และกราฟสามารถจำแนกได้ 2 แบบคือ การค้นหาแบบกำหนดทิศทางและการค้นหาแบบฮิวริสติก (Heuristic Search)

2.2 การค้นหาแบบกำหนดทิศทาง

การค้นหาแบบกำหนดทิศทาง เป็นการค้นหาแบบเริ่มต้นการสำรวจจากโหนดหนึ่งไปยังอีกโหนดหนึ่ง โดยอาศัยทิศทางเป็นตัวกำหนดการค้นหา ไม่ต้องมีข้อมูลอะไรมาช่วยเสริมการตัดสินใจว่าจะเดินทางต่อไปอย่างไร หรือกล่าวอย่างง่ายๆ คือ การจะหยิบข้อมูลใดมาตรวจสอบในขั้นต่อไป ไม่ต้องอาศัยข้อมูลใดๆ ทั้งสิ้น นอกจากทิศทาง ตัวอย่างเช่น การค้นหาแบบลึกก่อน (Depth First Search) และการค้นหาแบบกว้างก่อน (Breadth First Search)

2.2.1 การค้นหาแบบลึกก่อน

การค้นหาแบบลึกก่อนเป็นการค้นหาที่กำหนดทิศทางจากรูปของโครงสร้างต้นไม้ เริ่มต้นจากโหนดราก (Root Node) ที่อยู่บนสุด แล้วเดินลงมาให้ลึกที่สุด เมื่อถึงโหนดปลายทาง (Terminal Node) ให้ย้อนขึ้นมาที่กิ่ง (Branch) แยกต่ำสุดของกิ่งเดียวกัน แต่ยังไม่ได้ทำการสำรวจ แล้วทำการสำรวจโหนดในกิ่งนั้น ถ้าพบโหนดที่ต้องการ การค้นหาก็สิ้นสุด ถ้าไม่พบ ให้ทำการสำรวจจนถึงโหนดปลายทาง ถ้ายังไม่พบ ให้ย้อนขึ้นมายังกิ่งแยกต่ำสุดที่ยังไม่สำรวจถัดไป แล้วทำการสำรวจโหนดบนกิ่งแยกนั้น ทำเช่นนี้สลับกันไปเรื่อยๆ จนพบโหนดที่ต้องการหาหรือสำรวจได้ทำครบทุกโหนด



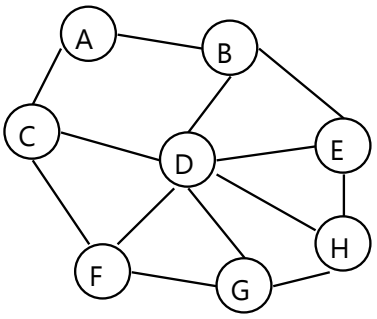
รูปที่ 2.1 ลำดับการเดินทางบนโหนดของการค้นหาแบบลึกก่อน

จากรูปที่ 2.1 การค้นหาแบบลึกก่อนจะมีลำดับการสำรวจโหนดดังตัวเลขที่กำกับไว้ในแต่ละโหนด หมายเลขที่กำกับอยู่ที่โหนดจะแสดงลำดับของการค้นหาแบบลึกก่อน เมื่อการค้นหาในกิ่งแรกมาถึงโหนดปลายทางคือ 4 แล้ว การค้นหาจะย้ายไปที่กิ่งแยกต่ำสุดและยังไม่ได้มีการสำรวจ ในที่นี้คือกิ่งของโหนด 3 ให้ทำการสำรวจโหนด 5 ซึ่งเป็นโหนดปลายทางอีก จากนั้นก็ย้อนกลับไปที่กิ่งแยกต่ำสุดถัดไปที่ยังไม่ได้มีการสำรวจ ในที่นี้คือโหนด 2 และให้ทำการสำรวจโหนด 6 และโหนด 7 ตามลำดับ ทำเช่นนี้เรื่อยๆ จนครบทุกโหนด หรือพบโหนดที่ต้องการ ในการค้นหาแบบลึกก่อน เราจะเห็นว่า เมื่อมีการสำรวจโหนดในกิ่งใดกิ่งหนึ่งที่เริ่มต้นจากโหนดรากลงมา เราจะต้องมีการบันทึกว่ามีโหนดใดบ้างที่มีกิ่งแยกออกไป ทั้งนี้เพื่อทำให้การสำรวจเมื่อลงไปจนถึงโหนดปลายทางแล้ว ย้อนกลับขึ้นไปหาโหนดต่ำสุดที่มีกิ่งแยกจะทำง่ายขึ้น ในการทำงานจริงของอัลกอริธึมการใช้โครงสร้างข้อมูลสแต็ก (Stack) มาช่วยใน

การค้นหา จะทำให้การสำรวจทำได้ง่ายขึ้น โดยเฉพาะอย่างยิ่ง เมื่อการสำรวจจะต้องมีการย้อนกลับขึ้นมาหาโหนดที่มีกิ่งแยกที่ต่ำที่สุด และยังไม่ได้ทำการสำรวจ ซึ่งการทำงานจะมีขั้นตอนดังนี้

กำหนดให้โหนดใดโหนดหนึ่งในกราฟเป็นโหนดเริ่มต้น ให้การสำรวจเริ่มจากโหนดเริ่มต้นนี้ จากนั้นนำโหนดที่อยู่ติดกับโหนดที่กำลังสำรวจอยู่ และยังไม่ได้ทำการสำรวจและยังไม่ได้อยู่ในสแต็กมาเก็บไว้ในสแต็ก เมื่อสำรวจโหนดนั้นเสร็จ ให้ป๊อป (Pop) ตัวบนสุดของโหนดออกมาทำการสำรวจ แล้วนำโหนดข้างเคียงทั้งหมดที่ยังไม่ได้สำรวจมาต่อท้ายสแต็ก แล้วป๊อปตัวบนสุดออกมาสำรวจ ทำเช่นนี้เรื่อยๆ จนกระทั่งพบโหนดที่ต้องการ หรือสำรวจครบทุกโหนด

จากรูปที่ 2.2 การสำรวจจะเริ่มต้นที่ A และนำโหนดข้างเคียง C และ B มาเก็บไว้ในสแต็กตามตารางที่ 2.1 แถวที่ 1 เมื่อสำรวจ A เสร็จ ป๊อปข้อมูลจากสแต็กออกมาได้ C ทำการสำรวจ C และนำโหนดข้างเคียงกับ C ที่ยังไม่ได้ทำการสำรวจและยังไม่ได้อยู่ในสแต็กมาใส่สแต็ก ในที่นี้คือ F และ D พุช (Push) ใส่สแต็ก ในสแต็กตอนปัจจุบันมี F D B อยู่ ตามตารางที่ 2.1 แถวที่ 2 เมื่อสำรวจ C เสร็จ ป๊อป F ออกมาทำการสำรวจ แล้วนำโหนดข้างเคียงที่ยังไม่ได้สำรวจและยังไม่ได้อยู่ในสแต็กมาใส่สแต็ก ซึ่งก็คือ G ดังนั้นข้อมูลในสแต็กจะเป็น G D B ตามตารางที่ 2.1 แถวที่ 3 ทำเช่นนี้ไปเรื่อยๆ จนจบการทำงาน จะได้ลำดับการสำรวจคือ (A C F G H E D B) ทั้งหมดตามตารางที่ 2.1 คอลัมน์ที่ 2



รูปที่ 2.2 โครงสร้างข้อมูลแบบกราฟ

แถวที่	โหนดที่สำรวจ	สแต็ก
1	A	C B
2	C	F D B

3	F	G D B
4	G	H D B
5	H	E D B
6	E	D B
7	D	B
8	B	

ตารางที่ 2.1 ลำดับการค้นหาแบบลึกก่อน

การค้นหาข้อมูลบนโครงสร้างกราฟแบบนี้ มีข้อที่น่าสังเกตคือ โหนดที่เริ่มต้นการสำรวจจะต้องมีการกำหนดไว้ก่อนล่วงหน้า และอัลกอริธึมของการค้นหาแบบลึกก่อนที่ใช้สำหรับโครงสร้างข้อมูลแบบกราฟ สามารถใช้กับโครงสร้างข้อมูลแบบต้นไม้ได้ด้วย

อัลกอริธึม การค้นหาข้อมูลแบบลึกก่อน

ให้สถานะ 1 หมายถึงโหนดที่ยังไม่สำรวจ สถานะ 2 หมายถึงโหนดที่อยู่ในสแต็ก สถานะ 3 หมายถึงโหนดที่ทำการสำรวจแล้ว

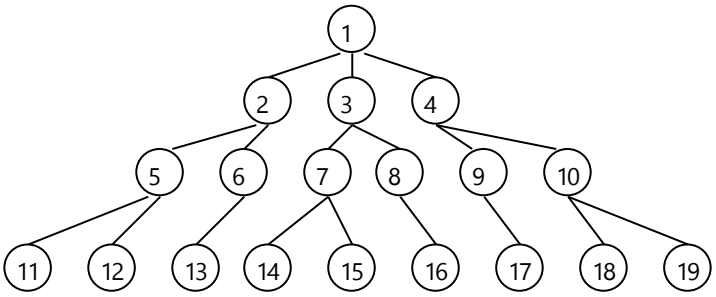
1. ให้โหนดทุกโหนดมีสถานะเป็น 1 และนำโหนดเริ่มต้นไปเก็บไว้ในสแต็ก พร้อมกับเปลี่ยนสถานะของโหนดเริ่มต้นนี้เป็น 2
2. ถ้าโหนดในสแต็กมีมากกว่า 0 ให้นำโหนดบนสุดในสแต็กออกมาสำรวจ และเปลี่ยนสถานะเป็น 3
   2.1. ถ้าโหนดที่ทำการสำรวจอยู่คือโหนดเป้าหมาย ให้รายงานว่าโหนดที่สำรวจคือโหนดเป้าหมาย และให้ข้ามไปทำขั้นตอนที่ 4
   2.2. ถ้าโหนดที่สำรวจไม่ใช่โหนดเป้าหมาย ให้นำโหนดข้างเคียงที่มีสถานะเป็น 1 ทั้งหมด (ถ้ามี) มาเปลี่ยนสถานะเป็น 2 แล้วนำใส่ไว้ในสแต็ก และกลับไปทำข้อที่ 2
3. รายงานการค้นหาล้มเหลว
4. การทำงานสิ้นสุด

หมายเหตุ : ลักษณะการใส่ข้อมูลเข้าและการนำข้อมูลออกของโครงสร้างข้อมูลแบบสแต็ก จะเป็นในลักษณะ “ข้อมูลที่นำเข้าก่อนจะถูกนำมาใช้ทีหลัง”

2.2.2 การค้นหาแบบกว้างก่อน

การค้นหาแบบกว้างก่อนเป็นการกำหนดทิศทางการค้นหาแบบที่ทำการสำรวจทีละระดับของโครงสร้างต้นไม้ โดยเริ่มจากโหนดราก (ระดับที่ 0) แล้วลงมาระดับที่ 1 ทำการสำรวจจากซ้ายไปขวา เมื่อเสร็จระดับที่ 1 ไประดับที่ 2 ทำการสำรวจจากซ้ายไปขวาเช่นกัน ทำเช่นนี้เรื่อยๆ จนพบโหนดที่ต้องการ หรือจนสำรวจครบทุกโหนด ตามรูปที่ 2.3 ลำดับของการสำรวจโหนดเป็นตามหมายเลขที่กำกับไว้บนโหนด

สำหรับการค้นหาแบบกว้างก่อนจะอาศัยการจัดการโครงสร้างข้อมูลแบบคิว (Queue) มาช่วย ซึ่งเป็นวิธีการเช่นเดียวกับการค้นหาแบบลึกก่อน คือให้เริ่มต้นสำรวจที่โหนดเริ่มต้น แล้วนำโหนดข้างเคียงเก็บไว้ในคิว เมื่อสำรวจโหนดเริ่มต้นเสร็จ ให้นำข้อมูลในคิวออกมาสำรวจ แล้วนำโหนดข้างเคียงที่ยังไม่ได้สำรวจและไม่ได้อยู่ในคิวใส่คิวไว้ ทำเช่นนี้ไปเรื่อยๆ จนพบโหนดที่ต้องการ หรือเมื่อสำรวจครบทุกโหนด สำหรับโหนดข้างเคียงในที่นี้หมายถึงโหนดที่มีเส้นเชื่อม (Link) เชื่อมกับโหนดที่กำลังสำรวจ หรือกล่าวอีกนัยหนึ่งคือ โหนดที่อยู่ติดกันและมีทางเดินเชื่อมหากัน



รูปที่ 2.3 ลำดับการค้นหาแบบกว้างก่อนบนโครงสร้างต้นไม้

เช่นเดียวกับการค้นหาแบบลึกก่อน การค้นหาแบบกว้างก่อนโดยใช้การจัดการข้อมูลคิวมาช่วย ต้องมีการกำหนดโหนดเริ่มต้น และวิธีการนี้สามารถใช้ได้กับข้อมูลบนโครงสร้างแบบต้นไม้ตามรูปที่ 2.3 ได้ด้วยเช่นกัน

อัลกอริธึม การค้นหาข้อมูลแบบกว้างก่อน

ให้สถานะ 1 หมายถึงโหนดที่ยังไม่สำรวจ สถานะ 2 หมายถึงโหนดที่อยู่ในคิว สถานะ 3 หมายถึงโหนดที่ทำการสำรวจแล้ว

1. ให้โหนดทุกโหนดมีสถานะเป็น 1 และนำโหนดเริ่มต้นไว้ในคิว เปลี่ยนสถานะเป็น 2
2. ถ้าจำนวนโหนดในคิวมีมากกว่า 0 นำโหนดแรกในคิวออกมาสำรวจ และเปลี่ยนสถานะเป็น 3
   2.1. ถ้าโหนดที่ทำการสำรวจอยู่คือโหนดเป้าหมาย รายงานโหนดที่สำรวจคือโหนดเป้าหมาย และให้ข้ามไปทำขั้นตอนที่ 4
   2.2. ถ้าโหนดที่สำรวจไม่ใช่โหนดเป้าหมาย ให้นำโหนดข้างเคียงที่มีสถานะเป็น 1 ทั้งหมด (ถ้ามี) เปลี่ยนสถานะเป็น 2 แล้วนำใส่ไว้ในคิว และกลับไปทำข้อที่ 2
3. รายงานการค้นหาล้มเหลว
4. การทำงานสิ้นสุด

หมายเหตุ : ลักษณะการใส่ข้อมูลเข้าและการนำข้อมูลออกของโครงสร้างข้อมูลแบบคิว จะเป็นในลักษณะ “ข้อมูลที่นำเข้าก่อนจะถูกนำมาใช้ก่อน”

2.3 การค้นหาแบบฮิวริสติก

ความแตกต่างระหว่างการค้นหาข้อมูลแบบธรรมดาและการค้นหาแบบฮิวริสติก (Heuristic Search) นั้น อยู่ที่ว่าการค้นหาข้อมูลแบบธรรมดาจะทำการตรวจสอบข้อมูลทีละตัวจนครบทุกตัว ถ้าข้อมูลที่มีขนาดไม่ใหญ่มาก การค้นหาแบบนี้จะให้คำตอบที่ถูกต้องเสมอ แต่ในบางครั้งข้อมูลมีขนาดใหญ่มาก การตรวจสอบข้อมูลอาจจะต้องทำกันหลายๆ ล้านครั้ง โดยเฉพาะอย่างยิ่งในโดเมนของปัญหาที่ค่าของตัวแปรเป็นจำนวนจริง ทำให้การเปรียบเทียบข้อมูลทุกตัวเพื่อหาคำตอบที่ถูกต้องเป็นไปไม่ได้ ลักษณะของปัญหาในแบบนี้ เช่น ปัญหาที่เป็นเชิงตัวเลข สำหรับปัญหาที่เป็นคอมบินาทอเรียลเช่น การหาทางเดินที่สั้นที่สุดของเมือง 100 เมือง ที่แต่ละเมืองมีทางเดินที่เชื่อมต่อกัน ถ้าจะหาเส้นทางที่สั้นที่สุดเราจะต้องทำการเปรียบเทียบเส้นทางถึง 99! ครั้ง หรือ (n – 1)! เมื่อ n คือจำนวนเมือง ซึ่งจะต้องทำการเปรียบเทียบกันจำนวนมหาศาล และอาจจะไม่สามารถเปรียบเทียบให้เสร็จได้ แม้จะใช้เวลาตลอดชั่วชีวิตคน ดังนั้นการที่จะตอบคำถามที่ว่า เส้นทางที่สั้นที่สุดคือเส้นทางใดนั้นจะยากมาก และอาจจะหา

ไม่ได้เลยก็ได้ ดังนั้นการแก้ปัญหาแบบนี้จึงต้องอาศัยวิธีการของฮิวริสติก วิธีการนี้จะเลือกคำตอบที่เหมาะสมให้กับการค้นหาเท่านั้น แต่อาจจะไม่ได้คำตอบที่ดีที่สุด

ถ้าลองกลับไปนึกถึงการค้นหาแบบลึกก่อนหรือกว้างก่อน เราจะเห็นข้อมูลปรากฏในรูปของโครงสร้างต้นไม้หรือกราฟ การค้นหาข้อมูลก็คือการสำรวจข้อมูล หรือโหนดบนโครงสร้างต้นไม้ หรือกราฟ ถ้าต้นไม้หรือกราฟนั้นมีจำนวนโหนดมากมายขนาด 99! โหนด การสำรวจทุกโหนดดังที่กล่าวมาข้างต้นของกิ่งทุกกิ่งนั้นเป็นไปไม่ได้

ในเรื่องของฮิวริสติกนั้น เครื่องมือสำคัญที่ช่วยการค้นหาแบบฮิวริสติกคือ ฮิวริสติกฟังก์ชัน (Heuristic Function) ที่ทำหน้าที่ในการวัดขนาดของความน่าจะเป็นในการแก้ปัญหา เพื่อกำหนดทิศทางของการค้นหาคำตอบ กล่าวอีกนัยหนึ่งคือ ฮิวริสติกฟังก์ชันจะเป็นตัวบอกว่าการสำรวจครั้งต่อไปจะเลือกสำรวจโหนดที่กิ่งใด โดยบอกเป็นขนาดของความน่าจะเป็น และขนาดของความน่าจะเป็นนี้จะถูกแสดงด้วยตัวเลข ((Newell, 1965); (Doran, 1996))

การวัดขนาดของความน่าจะเป็นในการแก้ปัญหา จะกระทำได้โดยการพิจารณาถึงวิธีการต่างๆ ที่ใช้ในการแก้ปัญหา ณ สถานะหนึ่งว่าจะสามารถแก้ปัญหาได้ตามที่ต้องการหรือไม่ โดยกำหนดเป็นน้ำหนักที่ให้กับการแก้ปัญหาของแต่ละวิธี น้ำหนักเหล่านี้จะถูกแสดงด้วยตัวเลขที่กำกับไว้กับโหนดต่างๆ ในกระบวนการค้นหา และค่าเหล่านี้จะเป็นตัวที่ใช้ในการประมาณความน่าจะเป็นว่าเส้นทางที่ผ่านโหนดนั้นมีความน่าจะเป็นในการนำไปสู่หนทางการแก้ปัญหาได้มากน้อยเพียงใด

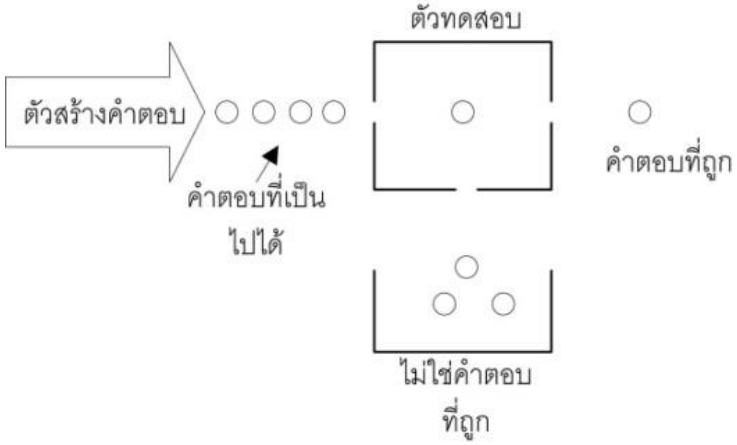
จุดประสงค์ของฮิวริสติกฟังก์ชันก็คือ การกำกับทิศทางของกระบวนการค้นหา เพื่อให้อยู่ในทิศทางที่ได้ประโยชน์สูงสุด โดยการบอกว่าเราควรเลือกเดินเส้นทางไหนก่อน ในกรณีที่มีเส้นทางมากกว่าหนึ่งเส้นทางต้องเลือก

กระบวนการค้นหาแบบฮิวริสติก โดยปกติแล้วจะต้องอาศัยฮิวริสติกฟังก์ชัน ทำให้การแก้ปัญหาหนึ่งๆ จะดีหรือไม่นั้น ขึ้นอยู่กับฮิวริสติกฟังก์ชันว่าจะวัดขนาดของความน่าจะเป็นแม่นยำหรือไม่ ดังนั้นการค้นหาแบบนี้จึงไม่มีอะไรเป็นหลักประกันว่าจะได้สิ่งที่ไม่ดีออกมาด้วย ในรายละเอียดที่จะกล่าวถึงดังต่อไปนี้ เป็นการอธิบายกระบวนการแบบฮิวริสติกเพื่อทำความเข้าใจในเรื่องดังกล่าว

2.3.1 เจเนอเรตแอนด์เทสต์

หลักการของการแก้ปัญหาด้วยวิธีการเจเนอเรตแอนด์เทสต์ (Generate and Test) นี้ก็คือ การที่ระบบสร้างคำตอบที่เป็นไปได้ออกมา แล้วทำการตรวจสอบคำตอบที่เป็นไปได้นั้นว่าถูกต้องหรือไม่ ถ้าถูกต้อง เราก็จะได้คำตอบ แต่ถ้าไม่ถูกต้อง ระบบก็จะทำการสร้างคำตอบที่เป็นไปได้ใหม่ออกมาอีก กระบวนการจะทำเช่นนี้ต่อไปเรื่อยๆ จนพบคำตอบ หรือจนไม่สามารถสร้างคำตอบที่เป็นไปได้ออกมาอีกแล้ว

องค์ประกอบสำคัญของเจเนอเรตแอนด์เทสต์ก็คือตัวสร้างคำตอบ (Generator) และตัวทดสอบ (Tester) ตัวสร้างคำตอบจะสร้างคำตอบที่เป็นไปได้ออกมาเรื่อยๆ และตัวตรวจสอบจะตรวจสอบคำตอบที่สร้างออกมาว่าถูกหรือผิด การค้นหาจะสิ้นสุดนั้นมีหลายวิธีคือ เมื่อพบคำตอบที่ต้องการ (ในกรณีนี้จะได้คำตอบแรกที่พบเพียงคำตอบเดียว) หรือเมื่อพบคำตอบได้จำนวนเท่าที่ต้องการ หรือเมื่อพบคำตอบทั้งหมด (ในกรณีนี้การค้นหาจะต้องค้นหาจากคำตอบที่เป็นไปได้ทั้งหมด) วิธีการตรวจสอบเช่นนี้เป็นวิธีการที่ง่ายที่สุด การทำงานคล้ายกับการลองผิดลองถูก รูปที่ 2.4 แสดงหลักการของการทำงานแบบนี้



รูปที่ 2.4 การทำงานของเจเนอเรตแอนด์เทสต์

อัลกอริธึม เจเนอเรตแอนด์เทสต์

1. สร้างคำตอบที่เป็นได้ออกมาหนึ่งชุด คำตอบเหล่านี้อาจหมายถึงสถานะใหม่ที่สร้างจากสถานะเริ่มต้น หรือเป็นการสร้างเส้นทาง (Path) ใหม่จากสถานะใดๆ ที่มีอยู่
2. ตรวจสอบว่าคำตอบที่สร้างขึ้นมานั้นถูกต้องหรือไม่ โดยการเปรียบเทียบสถานะที่สร้างขึ้นใหม่ หรือจากสถานะปลายทางของเส้นทาง (Path) ที่สร้างขึ้นกับสถานะเป้าหมาย
3. ถ้าการตรวจสอบในข้อที่ 2 พบคำตอบ เป็นอันว่าการค้นหาสิ้นสุด แต่ถ้าไม่พบคำตอบให้ย้อนกลับไปทำข้อที่ 1 ใหม่

ตัวอย่าง ถ้าเรามีสมการ 2(x + 3) = 20 หาค่าของ x โดยใช้เจเนอเรตแอนด์เทสต์

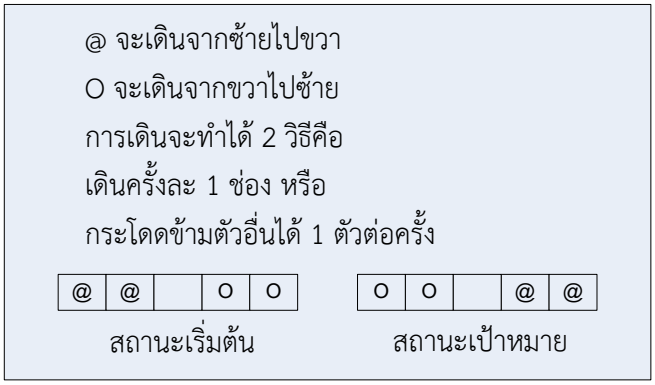
วิธีคือ สุ่มค่า x ออกมา แล้วทดสอบด้วยการแทนค่า x ลงในสมการ เช่น ให้ x = 1 ได้ค่า 2(1 + 3) = 8 ซึ่งไม่ตรงกับคำตอบที่เท่ากับ 20 แล้วทดลองสมมติเลขตัวใหม่ ให้ x = 6 จะได้ 2(6 + 3) = 18 ซึ่งก็ยังไม่ตรงอีก ทำการสมมติตัวเลขตัวใหม่ขึ้นมาเรื่อยๆ จนกว่าจะพบคำตอบ (เมื่อได้ค่า x = 7)

การสมมติค่าของ x ออกมาก็คือ *ตัวสร้างคำตอบ (Generator)* สร้างคำตอบที่เป็นไปได้ออกมา เมื่อนำมาใส่ในสมการ ตัวสมการจะเป็น *ตัวทดสอบ (Tester)* ว่าคำตอบนั้นเท่ากับ 20 หรือไม่ ในการสร้างคำตอบที่เป็นไปได้ออกมานี้ ถ้าการสร้างเป็นการสร้างที่เริ่มจากตัวเลขน้อย แล้วค่อยๆ เพิ่มขึ้น เช่น จากตัวอย่างข้างบน เช่นนี้ไปเรื่อยๆ จนกว่าจะพบคำตอบ การแก้แบบนี้ ถ้า x มีค่าตามสมการนั้นอยู่จริง เราก็จะต้องหาคำตอบนั้นได้แน่นอน

ในการแก้ปัญหาแบบเจเนอเรตแอนท์เทสต์ดังกล่าวข้างต้น เป็นเพียงการสร้างคำตอบที่เป็นไปได้ออกมาทีละคำตอบ ในบางครั้งการสร้างคำตอบที่เป็นไปได้อาจจะออกมาเป็นชุด ที่เป็นการหาเส้นทางจากโหนดรากจนถึงโหนดปลายทาง จากตัวอย่างดังต่อไปนี้

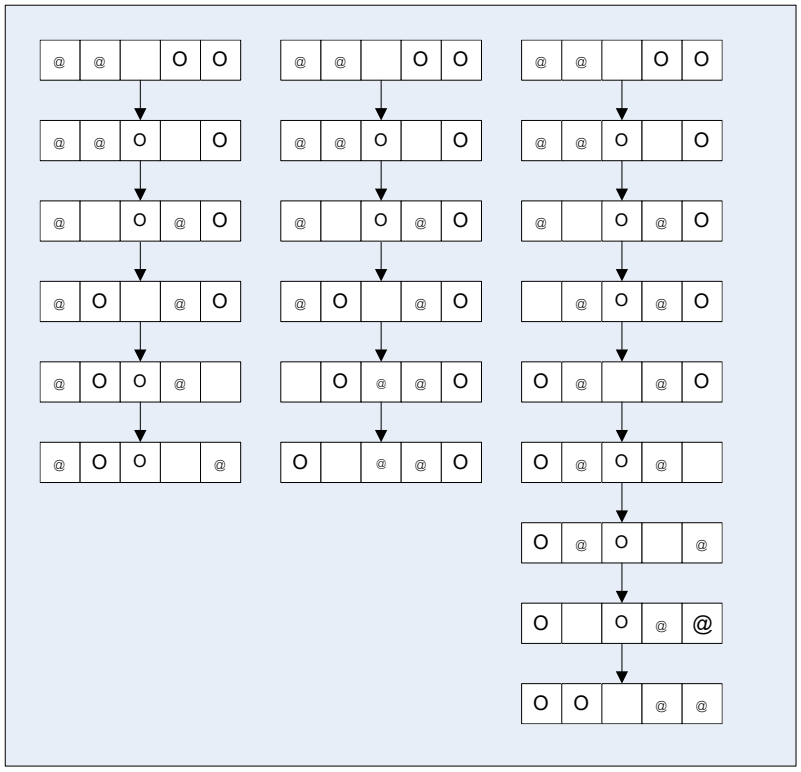
ตัวอย่าง การเล่นเกม @ และ O โดยการเดินตัว @ และ O จากตำแหน่งที่อยู่ในสถานะเริ่มต้นดังตารางที่อยู่ซ้ายมือ ให้ตัวเดิน @ และ O ย้ายตำแหน่งไปเป็นดังตารางเป้าหมายที่ตารางขวามือ โดยมีกฎของการเดินว่า @ จะเดินไปทางขวาเท่านั้น และ O จะเดินไปทางซ้ายเท่านั้น และทั้งสองตัวจะสามารถกระโดดข้ามตัวเดินอื่นได้เพียวตัวเดียว และจะต้องเป็นในทิศทางที่ตัวเดินนั้นเดินได้ด้วย

ในรูปที่ 2.5 เป็นโจทย์ตัวอย่างการแก้ปัญหาด้วยวิธีการของเจเนอเรตแอนด์เทสต์ ซึ่งเรามีสถานะเริ่มต้นและสถานะเป้าหมายตามที่กำหนด และทั้ง O และ @ มีทิศทางการเดินทางตามที่กำหนด เราต้องการที่จะเปลี่ยนสถานะเริ่มต้นไปเป็นสถานะเป้าหมายด้วยกฎตามที่กล่าวไว้ข้างต้น



รูปที่ 2.5 โจทย์ของการเดินเกม @ และ O

การแก้ปัญหาเริ่มต้นด้วยการสร้างคำตอบที่เป็นไปได้ ในกรณีนี้คำตอบที่เป็นไปได้ถูกสร้างออกมาเป็นเส้นทางเดิน (Path) ของการแก้ปัญหาตาม รูปที่ 2.6



รูปที่ 2.6 แสดงคำตอบที่เป็นไปได้ 3 คำตอบ

การสร้างคำตอบของปัญหานี้ ทำโดยการทดลองสร้างคำตอบด้วยการเดินหมากแบบลองผิดลองถูกไปเรื่อยๆ เมื่อผิดให้เริ่มต้นใหม่ จากรูปที่ 2.6 เราจะเห็นว่าการเดินของ 2 ครั้งแรกเดินมาถึงทางตันแล้วเริ่มต้นใหม่ จนกระทั่งมาถึงการเดินครั้งที่ 3 จึงได้คำตอบ ในการเดินแต่ละครั้ง ทั้ง 3 ครั้งเราจะเห็นเป็นเส้นทาง (Path) ของการแก้ปัญหา ซึ่งเป็นการเปรียบเทียบสถานะปลายทางของเส้นทางที่สร้างขึ้นกับสถานะเป้าหมาย ถ้าสถานะปลายทางกับสถานะเป้าหมายตรงกัน เส้นทางทั้งเส้นทางคือคำตอบของการแก้ปัญหา

เจเนอเรตแอนด์เทสต์ที่สร้างคำตอบเป็นไปได้ ด้วยการสุ่ม (Random) จากคำตอบที่เป็นไปได้ทั้งหมด ซึ่งหมายความว่า ไม่จำเป็นที่จะต้องสร้างคำตอบที่เป็นไปได้ทั้งหมดออกมาทดสอบ แต่จะสุ่มมาเพียงบางคำตอบจากคำตอบที่เป็นได้ที่สร้างขึ้นมา การทำเช่นนี้จะทำให้การค้นหาทำได้รวดเร็วขึ้น แต่

ข้อเสียก็คือ มีความเป็นไปได้สูงที่จะไม่พบคำตอบที่ต้องการ การสร้างคำตอบแบบนี้เราจะเรียกว่า *อัลกอริธึมพิพิธภัณฑ์อังกฤษ (British Museum Algorithm)*

จากลักษณะของปัญหาที่เป็นแบบสองขั้วคือ วิธีการที่หนึ่ง ระบบจะต้องสร้างคำตอบที่เป็นไปได้ทั้งหมดออกมาตรวจสอบ การค้นหาแบบนี้ เรียกว่า *การค้นหาแบบสมบูรณ์ (Complete Search)* กับอีกวิธีหนึ่งคือ การเดาสุ่ม ว่าจะสร้างคำตอบที่เป็นไปได้ตัวไหนออกมา การค้นหาแบบนี้เรียกว่า *การค้นหาแบบสุ่ม (Random Search)* ในการสร้างคำตอบที่เป็นไปได้นั้น เราสามารถหาวิธีตรงกลางได้ด้วยการสร้างตัวตรวจสอบขึ้นมา ตัวตรวจสอบนี้จะทำหน้าที่ช่วยในการตัดสินว่าคำตอบที่เป็นไปได้คำตอบใดควรสร้างออกมา หรือไม่ควรสร้างออกมา ด้วยการตรวจสอบความน่าจะเป็นที่สร้างออกมาว่าจะพบคำตอบหรือไม่ ตัวตรวจสอบนี้ก็คือ *ฮิวริสติกฟังก์ชัน (Heuristic Function)* ในหัวข้อต่อไปนี้เราจะได้กล่าวถึงวิธีการแบบต่างๆ ในการสร้างคำตอบที่เป็นไปได้ ด้วยการอาศัยฮิวริสติกฟังก์ชัน และกระบวนการค้นหาคำตอบแบบนี้ เรียกว่า *การค้นหาแบบฮิวริสติก (Heuristic Search)*

2.3.2 อัลกอริธึมกรีดี้

อัลกอริธึมกรีดี้ (Greedy Algorithm) เป็นการค้นหาแบบดีที่สุดก่อน (Best First Search) ที่ง่ายที่สุด หลักการของการค้นหาแบบนี้คือ การเลือกโหนดที่ดีที่สุดตลอดเวลา ในการเลือกโหนดที่ดีที่สุดของกรีดี้จะอาศัยค่าที่ได้จากการวัดจากสถานะปัจจุบันไปยังสถานะเป้าหมาย ถ้ากำหนดว่า h คือค่าที่ได้จากการวัดจากสถานะปัจจุบันไปยังสถานะเป้าหมาย ดังนั้นในการเลือกโหนดที่จะดำเนินการต่อไปจะเลือกจากโหนดที่มีค่า h ดีที่สุด

อัลกอริธึม กรีดี้: ให้ h คือค่าที่คำนวณจากสถานะปัจจุบันถึงสถานะเป้าหมาย และ SUCC เป็นหน่วยความจำชั่วคราวที่ใช้สำหรับเก็บค่าของการคำนวณที่เกิดขึ้น

1. เลือกสถานะเริ่มต้นมาหนึ่งโหนดให้เป็นสถานะปัจจุบัน
2. หาค่า h ของสถานะปัจจุบัน

   2.1. ถ้า h = 0 แสดงว่าสถานะปัจจุบันคือสถานะเป้าหมาย ให้รายงานคำตอบ และเลิกทำงาน

2.2. ถ้า h ≠ 0 ให้สร้างสถานะลูกให้กับสถานะปัจจุบัน แล้วนำไปเก็บไว้ใน SUCC จากนั้นหาค่าของ h ให้กับสถานะลูกทุกตัว เลือกสถานะลูกที่มีค่า h ดีที่สุดออกจาก SUCC และกำหนดให้สถานะลูกนั้นเป็นสถานะปัจจุบัน

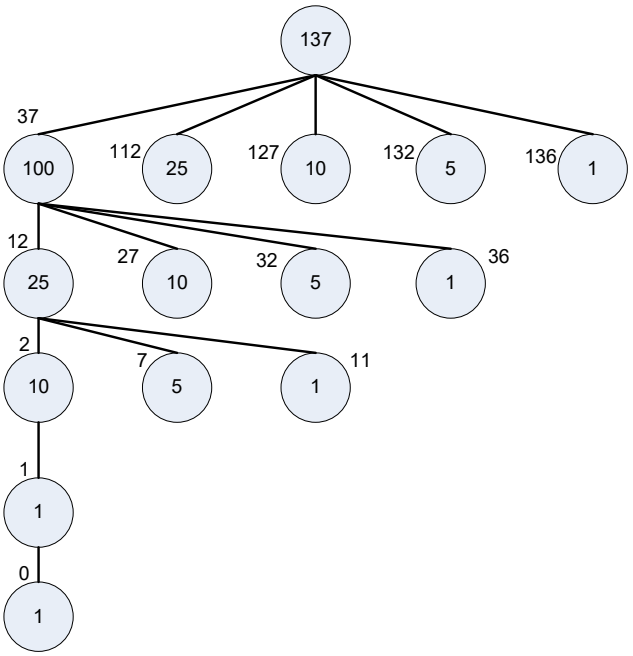
3. ล้างความจำใน SUCC แล้วกลับไปที่ขั้นตอนที่ 2

ตัวอย่าง การแลกเหรียญให้ได้จำนวนน้อยที่สุดจากจำนวนเงินที่กำหนด เช่น สมมติว่าในประเทศหนึ่งมีเงินเหรียญใช้อยู่ 5 ขนาดคือ 1 สตางค์, 5 สตางค์, 10 สตางค์, 25 สตางค์ และ 1 บาท ถ้าเราต้องการแลกเงินจำนวน 137 สตางค์ จะได้เหรียญน้อยที่สุดจำนวนกี่เหรียญ

ที่เหรียญ 1 บาท (= 100 สตางค์) ค่าของ h = (137 – 100) = 37
ที่เหรียญ 25 สตางค์ ค่าของ h = (137 – 25) = 112
ที่เหรียญ 10 สตางค์ ค่าของ h = (137 – 10) = 127
ที่เหรียญ 5 สตางค์ ค่าของ h = (137 – 5) = 132
ที่เหรียญ 1 สตางค์ ค่าของ h = (137 – 1) = 136
ให้เลือกเหรียญ 1 บาท ค่าเงินที่เหลือเท่ากับ 137 – 100 = 37 หาค่าของ 37
ที่เหรียญ 1 บาท (100 สตางค์) ใช้ไม่ได้ เพราะค่าของเหรียญมากกว่าค่าที่เหลือ
ที่เหรียญ 25 สตางค์ ค่าของ h = (37 – 25) = 12
ที่เหรียญ 10 สตางค์ ค่าของ h = (37 – 10) = 27
ที่เหรียญ 5 สตางค์ ค่าของ h = (37 – 5) = 32
ที่เหรียญ 1 สตางค์ ค่าของ h = (37 – 1) = 36
ให้เลือกเหรียญ 25 สตางค์ ค่าของเงินที่เหลือเท่ากับ 37 – 25 = 12 หาค่าของ 12 ต่อ
ที่เหรียญ 1 บาท (= 100 สตางค์) ใช้ไม่ได้ เพราะค่าของเหรียญมากกว่าค่าที่เหลือ
ที่เหรียญ 25 สตางค์ ใช้ไม่ได้ เพราะค่าของเหรียญมากกว่าค่าที่เหลือ
ที่เหรียญ 10 สตางค์ ค่าของ h = (12 – 10) = 2
ที่เหรียญ 5 สตางค์ ค่าของ h = (12 – 5) = 7
ที่เหรียญ 1 สตางค์ ค่าของ h = (12 – 1) = 11
ให้เลือกเหรียญ 10 สตางค์ ค่าของเงินที่เหลือเท่ากับ 12 – 10 = 2 หาค่าของ 2 ต่อ

ที่เหรียญ 1 บาท (= 100 สตางค์) ใช้ไม่ได้ เพราะค่าของเหรียญมากกว่าค่าที่เหลือ
ที่เหรียญ 25 สตางค์ ใช้ไม่ได้ เพราะค่าของเหรียญมากกว่าค่าที่เหลือ
ที่เหรียญ 10 สตางค์ ใช้ไม่ได้ เพราะค่าของเหรียญมากกว่าค่าที่เหลือ
ที่เหรียญ 5 สตางค์ ใช้ไม่ได้ เพราะค่าของเหรียญมากกว่าค่าที่เหลือ
ที่เหรียญ 1 สตางค์ ค่าของ h = (2 – 1) = 1
ให้เลือกเหรียญ 1 สตางค์ และมีค่าที่เหลือเท่ากับ 1
และให้เลือกเหรียญ 1 สตางค์ อีกครั้งก็จะได้คำตอบ

ดังนั้นสรุปว่า เราจะได้เหรียญบาท 1 เหรียญ เหรียญ 25 สตางค์ 1 เหรียญ เหรียญ 10 สตางค์ 1 เหรียญ และเหรียญ 1 สตางค์ 2 เหรียญ การแก้ปัญหาในเรื่องนี้สามารถเขียนเป็นโครงสร้างต้นไม้ได้ ดังแสดงในรูปที่ 2.7



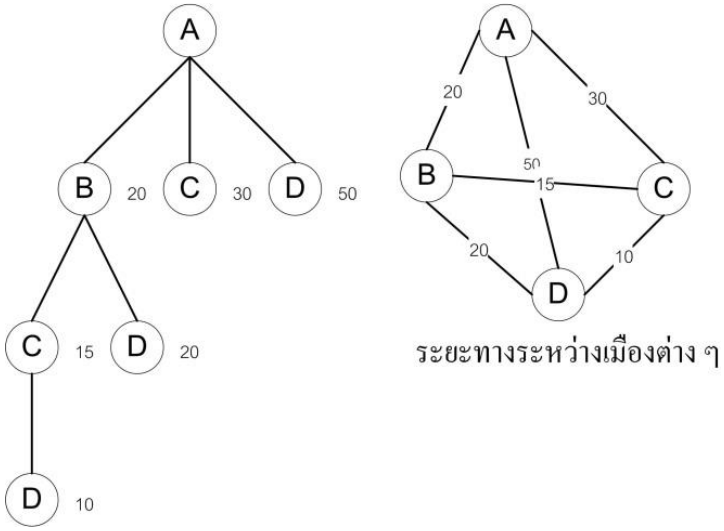
รูปที่ 2.7 โครงสร้างต้นไม้แสดงการทอนเหรียญด้วยกรีดีอัลกอริธึม

ตัวอย่าง การแก้ปัญหาแบบฮิวริสติกที่มักจะถูกหยิบมาเป็นตัวอย่างในการอธิบายบ่อยมากคือ *การเดินทางของเซลส์แมน (Traveling Salesman)* ปัญหาคือ

"เซลส์แมนคนหนึ่งมีชื่อของเมืองที่ตัวเองจะต้องเดินทางอยู่ในมือ เขามีแผนที่จะเดินทางไปยังแต่ละเมืองเพียงครั้งเดียว และทุกเมืองที่เขาจะต้องเดินทางไปมีเส้นทางต่อถึงกันหมด เขาต้องการหาเส้นทางของการเดินทางที่สั้นที่สุดเท่าที่จะเป็นไปได้ของการเดินทางครั้งนี้ โดยที่เขาจะเริ่มต้นเดินทางที่เมืองใดก็ได้ และสิ้นสุดที่เมืองใดก็ได้"

การแก้ปัญหาแบบนี้ด้วยวิธีการปกติก็คือ การหาเส้นทางเดินที่เป็นไปได้ทั้งหมด แล้วนำมาเปรียบเทียบกัน และเลือกเส้นทางที่สั้นที่สุด แต่การทำเช่นนี้จะทำได้ก็ต่อเมื่อเมืองต่างๆ ที่จะต้องเดินทางมีจำนวนน้อย จำนวนเส้นทางที่เป็นไปได้ทั้งหมดสำหรับ n เมืองจะได้เท่ากับ (n – 1) * (n – 2) * (n – 3) ... 1 หรือเท่ากับ (n – 1)! เส้นทาง ตัวอย่างเส้นทางที่เป็นไปได้ทั้งหมดของเมืองจำนวน 11 เมืองคือ 10! หรือ 3,628,800 เส้นทาง ถ้านำมาเปรียบเทียบกันทั้งหมดคงจะใช้เวลานานมาก ยิ่งถ้ามีสัก 100 เมือง คิดว่าคงเป็นเรื่องที่เป็นไปไม่ได้ที่จะหาเส้นทางที่เป็นไปได้ทั้งหมดจนครบ

วิธีการที่ค่อนข้างง่ายในการแก้ปัญหานี้ก็คือ วิธีการในการหาเส้นทางที่สั้นที่สุด (Shortest Path) ที่เสนอโดยเฮลด์และคาร์ป (Held, 1971) วิธีการก็คือ ให้เริ่มต้นจากเมืองใดเมืองหนึ่งโดยการสุ่ม จากนั้นทำการวัดจากเมืองที่เริ่มต้นนี้ไปยังเมืองที่เหลือ แล้วเลือกเดินทางไปยังเมืองที่ใกล้ที่สุด และจากเมืองที่เดินทางมานี้ วัดระยะทางจากเมืองนี้ไปยังเมืองที่เหลือ แล้วเลือกเดินทางไปยังเมืองที่ใกล้ที่สุด ทำเช่นนี้เรื่อยๆ จนครบทุกเมือง ดังรูปที่ 2.8



รูปที่ 2.8 การแก้ปัญหาการเดินทางของเซลส์แมนด้วยการหาระยะทางที่สั้นที่สุด

ในรูปที่ 2.8 สมมติว่าเราเริ่มต้นที่เมือง A จากนั้นวัดระยะทางจาก A ไป B จาก A ไป C และจาก A ไป D ได้ระยะทางออกมาเป็น 20, 30 และ 50 ตามลำดับ เราเลือกเดินทางไป B เป็นเมืองที่สอง จากนั้นวัดจาก B ไป C และ B ไป D ได้ระยะทางเป็น 15 และ 20 เลือกเดินทางไป C จากนั้นไป D เป็นเมืองสุดท้าย แล้วย้อนกลับมา A ที่เป็นเมืองตั้งต้น

	A	B	C	D
A	0	20	30	50
B		0	15	20
C			0	10
D				0

ตารางที่ 2.2 แสดงระยะทางระหว่างเมืองต่างๆ ของการเดินทาง

ตัวอย่าง จากตารางที่ 2.2 แสดงระยะทางระหว่างเมืองต่างๆ ของการเดินทางมีเมือง A B C D อยู่ 4 เมือง และมีระยะทางจากเมืองหนึ่งไปอีกเมืองหนึ่งมีค่าตามตารางที่ 2.2 เราเริ่มต้นด้วยการหาระยะทางหลักก่อนคือ A B C D รวมระยะทางแล้วได้ดังนี้

A B C D = 20 + 15 + 10 = 45

หาระยะทางที่สอง ด้วยการสุ่ม โดยเริ่มจากเมือง A C B เมื่อเดินทางมาถึง B ระยะทางที่ได้ก็เท่ากับระยะทางหลักแล้ว ทั้งที่ยังเดินไม่ครบทุกเมือง ดังนี้

A C B = 30 + 15 = 45

ดังนั้นการเดินทางที่เริ่มต้นด้วยเมือง A C B เราจะไม่เลือกทั้งหมด ถัดมาสร้างเส้นทางใหม่จาก A C D B เมื่อเดินครบทุกเมือง จะได้เท่ากับ 60 ซึ่งไกลกว่าระยะทางหลักที่หาได้ครั้งแรก ดังนั้นไม่เลือกเส้นทางนี้

A C D B = 30 + 10 + 20 = 60

สร้างเส้นทางใหม่ เมื่อเริ่มเดินทางจาก A D รวมกัน ระยะทางก็เกินเส้นทางหลักไปแล้วคือ

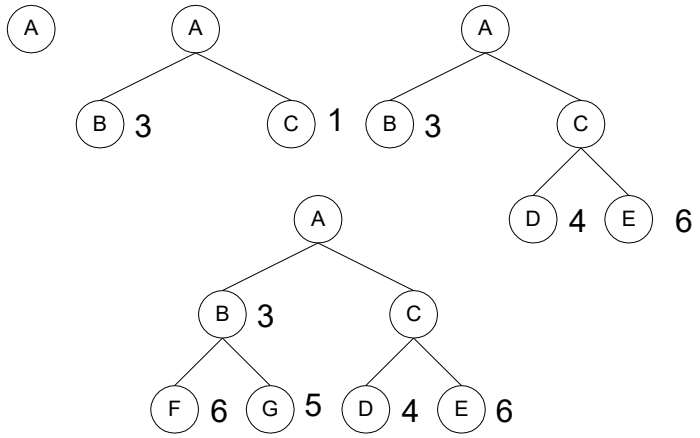
A D = 50

ดังนั้นเส้นทางที่เริ่มต้นด้วย A D จะไม่เลือกทั้งหมด เราจะได้เส้นทางที่ดีที่สุดคือ A B C D

2.3.3 การค้นหาแบบที่ดีที่สุดก่อน

การค้นหาแบบที่ดีที่สุดก่อน (Best First Search) เป็นกระบวนการค้นหาคำตอบ ที่นำเอาข้อดีของทั้งการค้นหาแบบลึกก่อน (Depth First Search) และการค้นหาแบบกว้างก่อน (Breadth First Search) มารวมกันเป็นวิธีการเดียว โดยที่แต่ละขั้นของการค้นหาในโหนดลูกนั้น การค้นหาแบบดีที่สุดก่อนจะเลือกเอาโหนดที่ดีที่สุด (Most Promising) การที่จะทราบว่าโหนดใดดีที่สุดนั้น สามารถทำได้โดยอาศัยฮิวริสติกฟังก์ชัน ซึ่งฮิวริสติกฟังก์ชันนี้จะทำหน้าที่เหมือนตัววัดผล และให้ผลของการวัดนี้ออกมาเป็นคะแนน รูปที่ 2.9 เป็นตัวอย่างของการค้นหาแบบดีที่สุดก่อน ขั้นตอนนี้เริ่มจากขั้นตอนที่ 1 สร้างโหนดราก (Root Node) ในขั้นตอนที่ 2 สร้างโหนดลูก B และ C แล้วตรวจสอบโหนด B และ C ด้วยฮิวริสติกฟังก์ชัน ได้ผลออกมาเป็นคะแนนคือ 3 และ 1 ตามลำดับ ให้เลือกโหนด C เป็นโหนดต่อไปที่เราสนใจ เพราะมีค่าน้อยกว่า (*หมายเหตุ :* ในการเลือกนี้จะเลือกค่ามากสุดหรือน้อยสุดก็ได้ ขึ้นอยู่กับลักษณะของปัญหา) แล้วสร้างโหนดลูกให้กับโหนด C ในขั้นตอนที่ 3 ได้โหนด D และ E แล้วตรวจสอบคะแนนได้ 4 และ 6 ตามลำดับ จากนั้นทำการเปรียบเทียบค่าของโหนดท้ายสุดหรือเทอร์มินอลโหนด (Terminal Node) ทุกโหนด ว่าโหนดใดมีค่าดีที่สุด ในที่นี้จะต้องเลือกโหนด B เพราะมีคะแนนเพียง 3

(เลือกคะแนนต่ำสุด) แล้วสร้างโหนดลูกตามขั้นตอนที่ 4 ได้ F และ G แล้วตรวจสอบคะแนนได้ 6 และ 5 คะแนน ตามลำดับ ทำเช่นนี้เรื่อยๆ จนพบคำตอบหรือจนไม่สามารถสร้างโหนดต่อไปได้อีก



รูปที่ 2.9 แสดงขั้นตอนของการค้นหาแบบที่ดีที่สุดก่อน

อัลกอริธึม การค้นหาแบบที่ดีที่สุดก่อน: กำหนดให้ SUCC เป็นหน่วยความจำชั่วคราวที่ใช้สำหรับเก็บโหนดปลายทาง (Terminal Node) หรือสถานะปลายทางใหม่ที่ถูกสร้างขึ้น และให้คำว่าโหนดและสถานะเป็นคำที่มีความหมายเดียวกัน

1. เริ่มด้วย SUCC ที่มีเพียงสถานะเริ่มต้นเท่านั้น
2. ทำกระบวนการดังต่อไปนี้จนกว่าจะพบเป้าหมาย หรือว่าไม่มีสถานะใดเหลืออยู่ใน SUCC
   2.1. เลือกสถานะที่ดีที่สุดใน SUCC ออกมา และกำหนดให้เป็นสถานะปัจจุบัน แล้วเอาสถานะปัจจุบันนี้ออกจาก SUCC
   2.2. สร้างสถานะลูกให้กับสถานะปัจจุบัน
   2.3. สำหรับสถานะลูกแต่ละตัว ให้ทำดังต่อไปนี้
      2.3.1.ถ้าสถานะลูกที่สร้างขึ้นมาใหม่ไม่เป็นสถานะเดียวกันกับสถานะที่อยู่ใน SUCC ให้หาค่าของสถานะนั้นโดยใช้ฮิวริสติกฟังก์ชัน แล้วใส่สถานะที่สร้างใหม่เข้าไปใน SUCC และบันทึกให้สถานะปัจจุบันเป็นสถานะพ่อแม่ (Parent State)

2.3.2.ถ้าสถานะลูกที่สร้างใหม่เป็นสถานะเดียวกันกับสถานะที่อยู่ใน SUCC ให้เปรียบเทียบว่าสถานะที่สร้างขึ้นมาใหม่นี้ เมื่อคำนวณค่าผ่านสถานะแม่ปัจจุบันกับสถานะแม่เดิมทางใดดีกว่าให้เลือกทางนั้น พร้อมกับปรับค่าของสถานะนั้นตามค่าใหม่ที่ดีกว่า

2.3.4 บรานช์แอนด์เบานด์

บรานช์แอนด์เบานด์ (Branch and Bound) เสนอโดยโฮโลวิตซ์และซาฮ์นี (Horowitz, 1974) (Horowitz, 1978) หลักการของการแก้ปัญหาก็คือ เริ่มต้นด้วยการหาเส้นทางขึ้นมาเส้นทางหนึ่ง แล้วพยายามหาเส้นทางอื่นขึ้นมาเปรียบเทียบ ถ้าเส้นทางใหม่ที่ได้ดีกว่า ให้นำเส้นทางใหม่ที่ได้เป็นเส้นทางหลักในการเปรียบเทียบกับเส้นทางอื่นต่อไป แต่ถ้าเส้นทางใหม่ที่สร้างขึ้น เมื่อวัดแล้วแม้มีเพียงบางส่วนที่ยาวกว่าเส้นทางหลัก ก็ให้ยกเลิกเส้นทางนั้น ทำเช่นนี้เรื่อยๆ จนกระทั่งได้เส้นทางที่ดีที่สุด

อัลกอริธึม บรานช์แอนด์เบานด์

1. สร้างเส้นทางที่เป็นไปได้มาเส้นทางหนึ่ง และให้ถือเป็นเส้นทางที่ดีที่สุดชั่วคราว
2. สร้างเส้นทางใหม่ขึ้นมาอีก แล้วทำการเปรียบเทียบกับเส้นทางในข้อแรก
   - **2.1** ถ้าเส้นทางที่สร้างใหม่ดีกว่าเส้นทางแรก ให้ใช้เส้นทางนี้เป็นเส้นทางที่ดีที่สุด
   - **2.2** ถ้าเส้นทางที่สร้างใหม่ แย่กว่าเส้นทางที่ดีที่สุดชั่วคราวแม้จะเพียงบางส่วน ให้ยกเลิกการพิจารณาเส้นทางเส้นนั้นทั้งหมด
3. ทำเช่นนี้เรื่อยๆ จนครบทุกความเป็นไปได้ของเส้นทาง

ตัวอย่าง การจัดตารางการทำงานโดยอาศัยวิธีการบรานช์แอนด์เบานด์ ซึ่งกำหนดให้มีคนอยู่ 4 คนคือ a b c และ d และมีงานอยู่ 4 ชิ้นคือ 1 2 3 และ 4 ตามลำดับ มีตารางค่าใช้จ่ายของการทำงานของแต่ละคนต่องานแต่ละชิ้นเป็นดังตารางที่ 2.3 เมตริกซ์ค่าใช้จ่ายในการทำงานจากบราสสาร์ด

	1	2	3	4
a	11	12	18	40
b	14	15	13	22
c	11	17	19	23

d	17	14	20	28

ตารางที่ 2.3 เมตริกซ์ค่าใช้จ่ายในการทำงานจากบราสสาร์ด (Brassard, 1996)

เพื่อที่จะหาขอบเขตของค่าใช้จ่ายสูงสุด สำหรับการพิจารณาว่าจะเลือกหรือไม่เลือกทางเลือกใหม่ที่เกิดขึ้น เราทดลองกำหนดโดยให้มีการมอบหมายงานในเบื้องต้นก่อนดังนี้ a = 1, b = 2, c = 3 และ d = 4 ได้ค่าใช้จ่ายรวมเท่ากับ 11 + 15 + 19 + 28 = 73 และทดลองให้ a = 4, b = 3, c = 2 และ d = 1 จะได้ค่าออกมาเป็น 40 + 13 + 17 + 17 = 87 ซึ่งสูงกว่าค่า 73 ดังนั้นให้เลือกค่า 73 เป็นกรอบค่าสูงสุด (Upper Bound Value)

ในการดูค่าต่ำสุดที่เป็นไปได้ ให้ดูตามแถวนอนก่อน โดยการเลือกค่าต่ำสุดของแต่ละแถวมา จะได้ 11 + 13 + 11 + 14 = 49 และทดลองดูตามแถวตั้งบ้าง จะได้ 11 + 12 + 13 + 22 = 58 ซึ่งได้ค่าที่ดีกว่า ให้เลือก 58 เป็นกรอบค่าต่ำสุด (Lower Bound Value)

เริ่มหาคำตอบจากการหาค่า a ที่เหมาะสมที่สุดก่อน ด้วยการกำหนดให้ a = 1, a = 2, a = 3 และ a = 4 ดูว่าค่า a ใดดีที่สุดดังนี้

a = 1	11 + 14 + 13 + 22	60
a = 2	12 + 11 + 13 + 22	58
a = 3	18 + 11 + 14 + 22	65
a = 4	40 + 11 + 14 + 13	78

ตารางที่ 2.4 การหาค่า a ที่เหมาะที่สุด

จากตารางที่ 2.4 การหาค่า a ที่เหมาะที่สุด เมื่องานที่ 1 ถูกมอบหมายให้ a เป็นผู้ทำ ดังนั้น a จึงไม่สามารถทำงานอื่นได้อีก สำหรับงานที่ 2 ให้เลือกช่องที่มีค่าใช้จ่ายต่ำสุด ยกเว้นงานที่ a ทำไปแล้ว (แถวนอน a) โดยดูจากช่องในแนวตั้งของงานที่ 2 ได้ค่าใช้จ่ายเท่ากับ 14 งานที่ 3 ทำเช่นเดียวกัน ได้ค่าใช้จ่ายต่ำสุดคือ 13 และงานที่ 4 ได้ค่าใช้จ่ายเท่ากับ 22

ในการเลือกค่าใช้จ่ายสำหรับงานที่ 2, 3 และ 4 ให้เลือกโดยคิดว่าระบบต้องการได้ค่าที่ต่ำที่สุด โดยไม่ต้องสนใจว่าใครจะเป็นคนทำ หรือใครจะทำซ้ำหรือไม่ ยกเว้น a และเช่นกัน เมื่อ a = 2, a = 3 และ a = 4 ก็ใช้วิธีการเดียวกัน จะได้ผลออกมาตามตารางที่ 2.4

ค่า a ที่ดีที่สุดคือ 2 เพราะจะให้ค่าใช้จ่ายออกมาต่ำสุด และในขณะเดียวกัน เมื่อค่าของ a = 4 ค่าทั้งหมดจะเท่ากับ 78 มีค่ามากกว่าค่ากรอบสูงสุด (73) ดังนั้นคำตอบที่จะเกิดจาก a = 4 ในตอนเริ่มต้นจึงไม่มีความจำเป็นที่จะต้องพิจารณาอีกต่อไป

จากนั้นเลือกค่า b ที่เหมาะสมที่สุด ด้วยการกำหนดค่า a = 2 ดังนี้

a = 2, b = 1	12 + 14 + 19 + 23	68
a = 2, b = 3	12 + 13 + 11 + 23	59
a = 2, b = 4	12 + 22 + 11 + 19	64

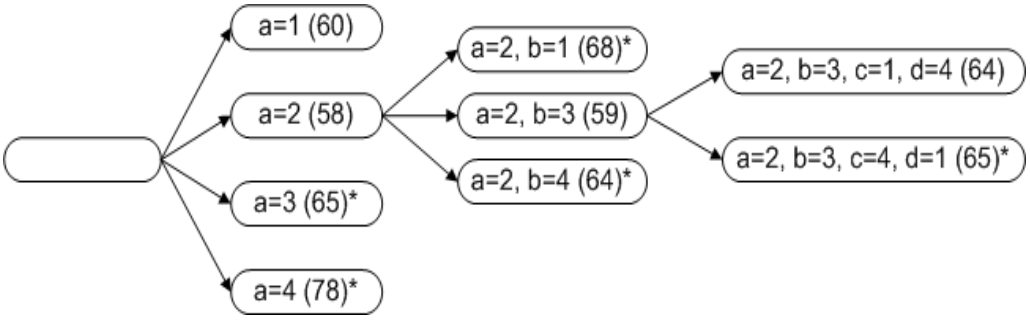
ตารางที่ 2.5 การหาค่า b ที่เหมาะที่สุด

จากตารางที่ 2.5 ได้ค่าของ b = 3 เป็นค่าที่ดีที่สุด ต่อไปพิจารณาเลือกงาน c และ d ได้ผลออกมาตามตารางที่ 2.6

a = 2, b = 3, c = 1, d = 4	12 + 13 + 11 + 28	64
a = 2, b = 3, c = 4, d = 1	12 + 13 + 23 + 17	65

ตารางที่ 2.6 การหาค่า c และ d ที่เหมาะที่สุด

จากการคำนวณมาที่ได้ออกมาตารางที่ 2.6 จะเห็นว่า a = 2, b = 3, c = 1, d = 4 ให้ผลลัพธ์ออกมาดีที่สุด ตอนนี้เราจะต้องย้อนกลับไปดูที่โหนดทั้งหมดที่สร้างออกมา แล้วพิจารณาซ้ำอีกครั้ง โดยการตั้งค่ากรอบสูงสุดใหม่เป็น 64 ตามค่าใหม่ที่เราหามาได้ดังตารางที่ 2.6 การหาค่า c และ d ที่เหมาะที่สุดตามรูปที่ 2.10



รูปที่ 2.10 โครงสร้างต้นไม้ของการแก้ปัญหาบางส่วน

เห็นได้ว่ามีค่าเหลือเพียงโหนดเดียวที่ควรทำการทดสอบว่า จะได้ค่าที่ดีกว่าค่าที่เราได้มาแล้วหรือไม่ คือเมื่อ a = 1 (โหนดที่มีเครื่องหมาย * หมายความว่าเป็นโหนดที่มีค่ามากกว่ากรอบสูงสุด) ให้ทำการทดลองเช่นที่ผ่านมา จะได้โหนดที่ขยายออกมาใหม่ทั้งหมด เมื่อพิจารณาให้ a = 1 และหาค่า b ที่ดีที่สุดจะได้ดังตารางที่ 2.7 การหาค่า b ที่เหมาะที่สุด

a = 1, b = 2	11 + 15 + 19 + 23	68
a = 1, b = 3	11 + 13 + 14 + 23	61
a = 1, b = 4	11 + 22 + 14 + 19	66

ตารางที่ 2.7 การหาค่า b ที่เหมาะที่สุด

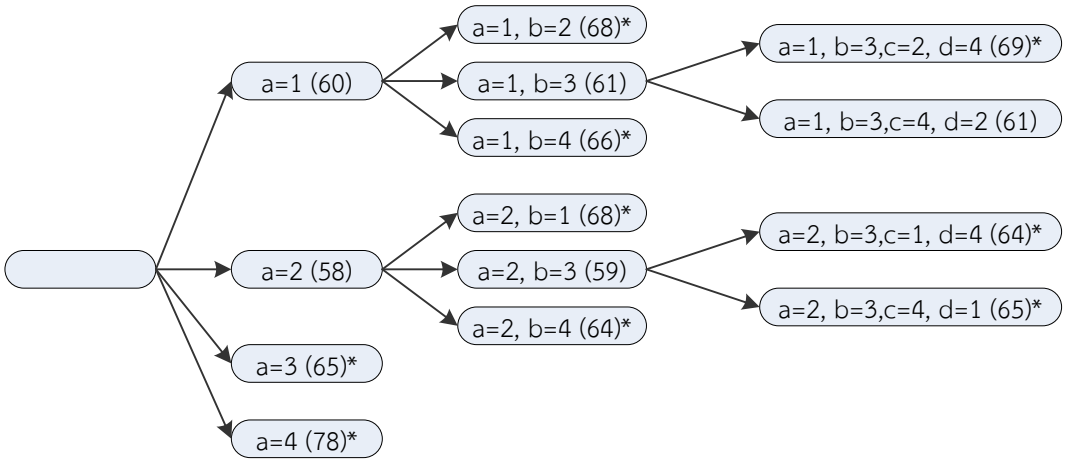
ค่า b = 3 เป็นค่าที่ดีที่สุด ให้เลือก a = 1 และ b = 3 จากนั้นหาค่าที่ดีที่สุดของ c และ d ดังตารางที่ 2.8

a = 1, b = 3, c = 2, d = 4	11 + 13 + 17 + 28	69
a = 1, b = 3, c = 4, d = 2	11 + 13 + 23 + 14	61

ตารางที่ 2.8 การหาค่า c และ d ที่เหมาะที่สุด

เมื่อเขียนเป็นกราฟที่เราได้ทำการแยกโหนดออกมาทั้งหมด แล้วกำหนดให้ค่าที่ดีที่สุดค่าใหม่เท่ากับ 61 เป็นค่ากรอบบนที่ดีที่สุด จะเห็นว่าไม่มีโหนดใดที่เราจะต้องพิจารณาอีก เพราะโหนดที่เรา

สร้างมาทั้งหมดมีค่ามากกว่า 61 ที่เราได้มาใหม่ ดังกราฟในรูปที่ 2.11 กราฟแสดงโหนดทั้งหมดของการทำงานแบบบรานช์แอนด์เบานด์



รูปที่ 2.11 กราฟแสดงโหนดทั้งหมดของการทำงานแบบบรานช์แอนด์เบานด์

เมื่อเปรียบเทียบการแก้ปัญหาของบรานช์แอนด์เบานด์กับวิธีเจเนอเรตแอนด์เทสต์ ที่เราต้องสร้างเส้นทางทั้งหมด แล้วนำระยะทางที่ได้ทั้งหมดมาเปรียบเทียบกัน การแก้ปัญหาของบรานช์แอนด์เบานด์จะดีกว่า เพราะการเปรียบเทียบของบรานช์แอนด์เบานด์ไม่จำเป็นที่จะต้องหาระยะทางของเส้นทางสมบูรณ์ (Complete Path) บางครั้งเมื่อวัดระยะทางได้ระยะหนึ่ง แล้วเห็นว่าแย่กว่าระยะทางที่มีอยู่เดิม เราสามารถยกเลิกการวัดเส้นทางนั้นต่อไปได้ ซึ่งทำให้ประหยัดเวลาในการค้นหาได้มาก

วิธีการแก้ไขปัญหาแบบนี้ แม้จะดีกว่าวิธีเจเนอเรตแอนด์เทสต์ การเปรียบเทียบจะไม่ทำทุกเส้นทาง แต่การแก้ปัญหาแบบนี้ก็ยังใช้เวลาในการแก้นานอยู่ดี เพราะการแก้เช่นนี้ยังจำเป็นต้องอาศัยการสร้างคำตอบที่เป็นไปได้จำนวนมากมาเปรียบเทียบกัน ดังนั้นการแก้ปัญหา เช่น การเดินทางของเซลส์แมนหรือการจัดตารางการทำงานด้วยบรานช์แอนด์เบานด์จึงยังไม่ดีนักถ้ามีจำนวนเมืองมากๆ หรือจำนวนคนและงานมากๆ

2.3.5 การค้นหาแบบฮิลไคลมิ่ง

การค้นหาแบบฮิลไคลมิ่ง (Hill Climbing) นำเสนอโดยเพิร์ล (Pearl, 1984) เป็นวิธีการค้นหาข้อมูลที่มีลักษณะคล้ายกับการปีนภูเขา การที่นักปีนเขาจะเดินทางไปถึงยอดภูเขา นักปีนเขาจะต้องมองก่อนว่ายอดเขาอยู่ที่ใด แล้วนักปีนเขาจะต้องพยายามไปจุดนั้นให้ได้ ลองนึกภาพของการปีนภูเขาหัวโล้นที่มองเห็นแต่ยอด และนักปีนเขากำลังปีนภูเขาจากเบื้องล่างที่มีเส้นทางเต็มไปหมด เพื่อที่จะเดินทางไปถึงยอดภูเขาโดยเร็วที่สุด นักปีนเขาจะมองไปที่ยอดเขาแล้วสังเกตว่าทิศทางใดที่เมื่อปีนแล้วจะยิ่งใกล้ยอดเขา และหลีกเลี่ยงทิศทางที่ทำให้ตัวเองห่างจากยอดเขา นักปีนเขาจะต้องทำเช่นนี้ไปเรื่อยๆ จนกระทั่งถึงยอดเขา

ความจริงแล้วการค้นหาวิธีนี้เป็นวิธีที่ดัดแปลงมาจากเจเนอเรตแอนด์เทสต์ ต่างกันตรงที่วิธีการของเจเนอเรตแอนด์เทสต์ให้คำตอบของตัวทดสอบ (Tester) ออกมาเป็น yes และ no คือคำตอบนั้นถูกหรือผิด แต่ของฮิลไคลมิ่งจะให้เป็นค่าที่บอกถึงลักษณะของคำตอบออกมา ค่าเหล่านี้เช่น ความยาวของระยะทาง ค่าใช้จ่ายที่ต้องใช้ และเวลาที่ใช้ในการทำงาน เป็นต้น

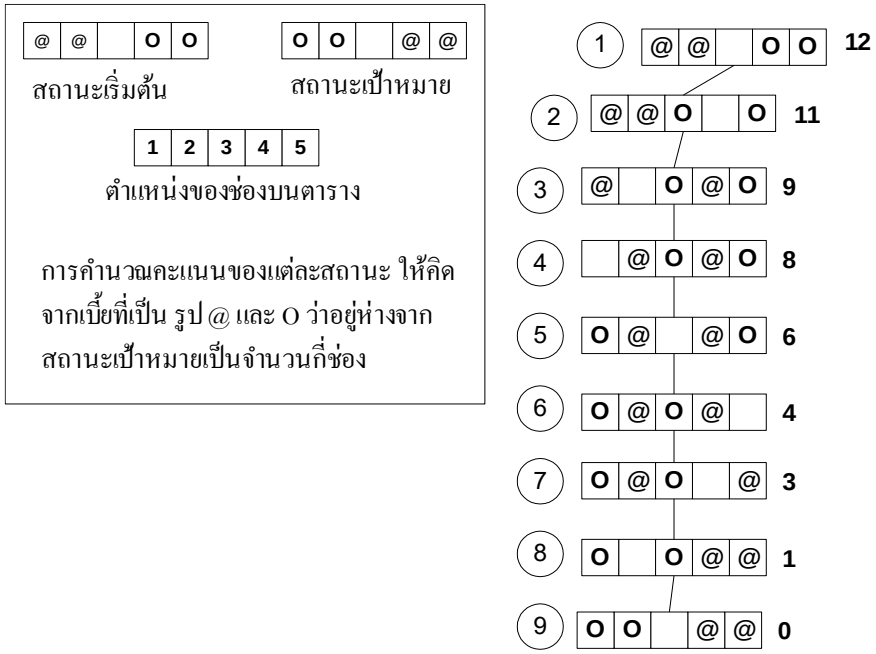
2.3.6 อัลกอริธึมของฮิลไคลมิ่งธรรมดา

การค้นหาแบบฮิลไคลมิ่งธรรมดา (Simple Hill Climbing) นั้นมีหลักการง่ายๆ คือ ในแต่ละโหนดหรือสถานะ ให้สร้างลูกออกมา 1 ตัว แล้วตรวจสอบว่าโหนดที่สร้างขึ้นมานี้ดีกว่าโหนดปัจจุบันหรือไม่ ด้วยการอาศัยฮิวริสติกฟังก์ชัน (Heuristic function) ที่ในเจเนอเรตแอนด์เทสต์ เรียกว่า *ตัวทดสอบ* ถ้าการทดสอบให้ผลดีกว่า ให้โหนดที่สร้างใหม่นี้เป็นโหนดปัจจุบัน แล้วสร้างโหนดลูกออกมาจากโหนดที่ได้เลือกใหม่นี้ แล้วทำการตรวจสอบต่อ ถ้าผลการทดสอบไม่ดีกว่า ให้สร้างลูกตัวใหม่จากเดิมขึ้นมาตรวจสอบจนกว่าจะพบลูกที่ดีกว่าสถานะปัจจุบัน อัลกอริธึมจะทำวนเวียนเช่นนี้ไปเรื่อยๆ จนพบคำตอบ เมื่อเขียนเป็นขั้นตอนแล้วจะได้ดังนี้

อัลกอริธึม ฮิลไคลมิ่ง: การทำงานของอัลกอริธึมนี้เริ่มจาก *สถานะเริ่มต้น (Initial State)* เป็นโหนดรากของโครงสร้างต้นไม้ และให้เรียกโหนดนี้ว่า *สถานะปัจจุบัน (Current State)* แล้วสร้างโหนดลูกให้กับสถานะปัจจุบัน และทำการตรวจสอบว่าใช่สิ่งที่เราต้องการหาหรือไม่ ด้วยการเปรียบเทียบสถานะปัจจุบันกับ*สถานะเป้าหมาย (Goal State)* ในการสร้างโหนดลูกหรือ*สถานะใหม่ (New State)*

ให้กับสถานะปัจจุบัน จะต้องอาศัย*ตัวดำเนินการ (Operator)* ซึ่งตัวดำเนินการแต่ละตัวจะสร้างสถานะใหม่ให้กับสถานะปัจจุบันได้ทีละหนึ่งสถานะ

1. ให้สถานะเริ่มต้นเป็นสถานะปัจจุบัน
2. ตรวจสอบสถานะปัจจุบันกับสถานะเป้าหมาย ถ้าสถานะปัจจุบันนี้คือสถานะเป้าหมาย ก็ให้แสดงคำตอบและเลิกการทำงาน แต่ถ้าสถานะเริ่มต้นไม่ได้เป็นสถานะเป้าหมาย ให้ไปทำงานต่อข้อถัดไป
3. ให้ทำตามกระบวนการข้างล่างนี้จนกว่าจะพบคำตอบ หรือจนกระทั่งไม่มีตัวดำเนินการใดๆ ที่จะใช้กับสถานะปัจจุบัน เพื่อสร้างสถานะใหม่
   3.1. เลือกตัวดำเนินการที่ยังไม่ได้ใช้การกับสถานะปัจจุบันมาใช้ เพื่อสร้างสถานะใหม่ขึ้นมาสถานะหนึ่งแล้วไปทำงานต่อที่ข้อ 3.2
   3.2. ตรวจสอบสถานะใหม่ดังนี้
      3.2.1. ถ้าเป็นคำตอบ ให้แสดงคำตอบและเลิกการทำงาน
      3.2.2. ถ้าไม่ใช่คำตอบ แต่สถานะที่ได้ใหม่ดีกว่าสถานะปัจจุบัน ให้สถานะใหม่นี้เป็นสถานะปัจจุบัน
      3.2.3. ถ้าไม่ใช่คำตอบ และสถานะที่ได้แย่กว่าสถานะปัจจุบัน ให้กลับไปทำซ้ำที่ข้อที่ 3.1 ใหม่
4. ให้เลิกทำงาน และรายงานว่าการค้นหาประสบความล้มเหลว



รูปที่ 2.12 แสดงการแก้ปัญหาด้วยวิธีการฮิลไคลมิ่ง

ตัวอย่าง ในรูปที่ 2.12 การทำงานจะเริ่มจากสถานะเริ่มต้น ซึ่งมีค่าเท่ากับ 12 การหาค่าของสถานะนี้เราใช้ฮิวริสติกฟังก์ชัน ซึ่งวิธีการคำนวณจากการหาระยะห่างของตำแหน่งปัจจุบันกับตำแหน่งในสถานะเป้าหมาย เช่น @ ตัวแรกของสถานะเริ่มต้นอยู่ที่ตำแหน่งที่ 1 (ซ้ายสุด) จะต้องไปอยู่ตำแหน่งที่ 4 ของสถานะเป้าหมาย ดังนั้น @ ตัวแรกอยู่ห่างจากเป้าหมาย 3 ตำแหน่ง @ ตัวที่สองของสถานะเริ่มต้นจะต้องไปอยู่ที่ตำแหน่งที่ 5 ของสถานะเป้าหมาย ซึ่งก็อยู่ห่างจากตำแหน่งเป้าหมาย 3 ตำแหน่ง และเช่นกัน O ทั้งสองตัวก็อยู่ห่างจากตำแหน่งเป้าหมาย 3 ตำแหน่ง ดังนั้นค่าของสถานะเริ่มต้นจะเท่ากับ 3+3+3+3 = 12 วิธีการคำนวณนี้คือ ฮิวริสติกฟังก์ชัน การออกแบบฮิวริสติกฟังก์ชันนี้ขึ้นอยู่กับศิลปะของการแก้ปัญหา ว่าใครจะสามารถคิดค้นฮิวริสติกฟังก์ชันออกมาแก้ปัญหาได้อย่างไร ซึ่งเป็นสิ่งที่ยากที่สุดของการแก้ปัญหาแบบฮิวริสติกนี้

ที่สถานะเริ่มต้น โหนดที่ 1 มีค่าที่ได้จากการคำนวณด้วยฮิวริสติกฟังก์ชันเท่ากับ 12 สร้างลูกออกมาโหนดที่ 2 ซึ่งมีตัวดำเนินการคือกฎในการเคลื่อนที่ของ O จะได้สถานะใหม่มีค่า 11 สถานะใหม่ที่ได้ไม่ตรงกับสถานะเป้าหมาย แต่ดีกว่าสถานะเดิม ให้โหนด 2 เป็นโหนดปัจจุบัน สร้างลูกออกมาอีก ได้

โหนด 3 โดยใช้กฎการกระโดดของ @ เป็นตัวดำเนินการ สถานะใหม่มีค่า 9 สถานะใหม่ที่ได้ไม่ตรงกับสถานะเป้าหมาย แต่ดีกว่าโหนดปัจจุบัน ให้โหนด 3 เป็นโหนดปัจจุบัน สร้างลูกออกมาเป็นโหนด 4 โดยใช้กฎในการเลื่อน @ เป็นตัวดำเนินการ สถานะใหม่มีค่าเท่ากับ 8 และทำเช่นนี้ไปเรื่อยๆ จนได้โหนด 9 ซึ่งมีค่าเป็น 0 การค้นหาก็สิ้นสุดเพราะเมื่อเปรียบเทียบกับสถานะเป้าหมายแล้วตรงกัน

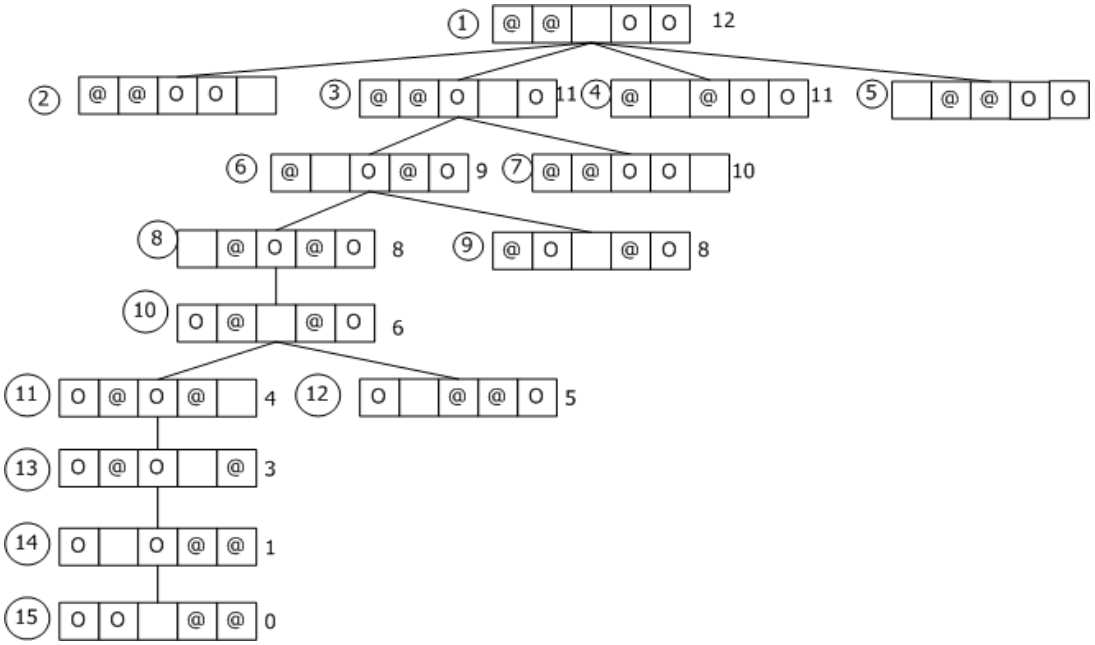
2.3.7 อัลกอริธึมของสตีเพสต์-แอสเซนต์ฮิลไคลมิ่ง

วิธีการนี้จะต่างจากฮิลไคลมิ่งธรรมดาคือ ฮิลไคลมิ่งแบบธรรมดาจะสร้างโหนดลูกออกมาทีละตัว แล้วเลือกโหนดลูกที่ดีกว่าโหนดแม่ตัวแรกที่พบ โดยไม่สนใจลูกที่เหลือ แล้วสร้างโหนดใหม่รุ่นถัดไปเพื่อเปรียบเทียบต่อไปอีก แต่สำหรับสตีเพสต์-แอสเซนต์ฮิลไคลมิ่ง (Steepest-Ascent Hill Climbing) นั้นจะสร้างโหนดลูกทุกตัวออกมาก่อน แล้วเลือกตัวที่ดีที่สุดของลูกทั้งชุด

อัลกอริธึม สตีเพสต์-แอสเซนต์ฮิลไคลมิ่ง: การทำงานของอัลกอริธึมนี้จะเพิ่มตัวแปรที่ชื่อว่า BVAL ทำหน้าที่เป็นดัชนี คอยชี้ไปยังสถานะ/โหนดลูกที่เพิ่งสร้างใหม่และมีค่าดีที่สุด โดยในตอนเริ่มต้นให้ดัชนีนี้ชี้ไปยังสถานะที่แย่ที่สุดก่อน

1. ตรวจสอบสถานะเริ่มต้น ถ้าสถานะเริ่มต้นนี้คือสถานะเป้าหมาย ให้แสดงคำตอบออกมาและยกเลิกการทำงาน แต่ถ้าสถานะเริ่มต้นไม่ได้เป็นสถานะเป้าหมาย ให้เปลี่ยนสถานะนี้เป็นสถานะปัจจุบัน และทำต่อข้อที่ 2
2. ให้ทำตามกระบวนการข้างล่างนี้จนกว่าจะพบคำตอบ หรือจนกระทั่งคาดว่าไม่สามารถหาคำตอบได้
   2.1 ให้ BVAL เป็นสถานะที่ไม่ว่าสถานะลูกใดๆ ที่เกิดใหม่จะต้องดีกว่าสถานะนี้
   2.2 สำหรับตัวดำเนินการที่จะใช้กับสถานะปัจจุบันแต่ละตัว ให้ดำเนินการดังนี้
      2.2.1 สร้างสถานะใหม่จากสถานะปัจจุบัน
      2.2.2 ตรวจสอบสถานะใหม่ ถ้าเป็นคำตอบ ให้แสดงผลและเลิกการทำงาน ถ้าไม่ใช่ให้เปรียบเทียบสถานะใหม่กับ BVAL ถ้าสถานะใหม่ดีกว่า ให้สถานะใหม่นี้เป็น BVAL ถ้าสถานะใหม่ไม่ดีกว่า ไม่ต้องทำอะไรทั้งนั้น
      2.2.3 เปลี่ยนตัวดำเนินการตัวใหม่ที่ยังไม่ได้ใช้งาน และกลับไปทำข้อที่ 2.2.1
   2.3 ถ้า BVAL ดีกว่าสถานะปัจจุบัน ให้สถานะปัจจุบันคือ BVAL

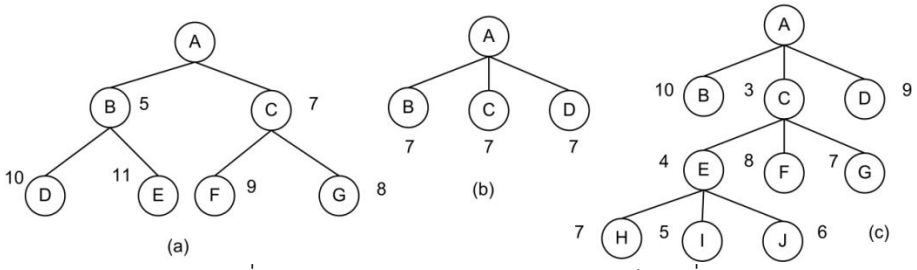
ตัวอย่าง ในรูปที่ 2.13 การแก้ปัญหาเริ่มจากสถานะเริ่มต้น สมมติว่าสถานะเริ่มต้นนี้สร้างโหนดลูกออกมาได้ 4 โหนดคือ 2, 3, 4 และ 5 จากรูปเราจะเห็นได้ชัดเจนว่าโหนด 2 และ 5 เป็นทางตัน ไม่สามารถสร้างตาเดินได้อีก สำหรับโหนด 3 และ 4 มีค่าเท่ากันคือ 11 ให้เลือกโหนดแรกที่ถูกสร้างขึ้นก่อนคือ 3 (ความจริงแล้ว การที่โหนดแม่สร้างลูกออกมาได้ค่าเท่ากันหมด ทำให้การเลือกโหนดที่ดีที่สุดมีปัญหา ให้ดูเรื่อง “เพลทู” ในหัวข้อถัดไป ในที่นี้เราแก้ปัญหาด้วยวิธีนี้ก่อน) จากนั้นให้นำ 3 มาสร้างลูกออกมาได้ 6 และ 7 ซึ่งมีค่าเท่ากับ 9 และ 10 ตามลำดับ ให้เลือกโหนด 6 ซึ่งมีค่าดีกว่า สร้างโหนดลูกได้ออกมาเป็น 8 และ 9 มีค่าเท่ากันคือ 8 ทั้งคู่ ให้เลือกโหนดที่สร้างก่อนคือ 8 สร้างลูกออกมาเป็นโหนด 10 (สร้างโหนดลูกได้เพียงโหนดเดียว) มีค่าเท่ากับ 6 ซึ่งดีกว่าโหนดแม่ และทำเช่นนี้เรื่อยๆ จนพบคำตอบตามรูปที่ 2.13



รูปที่ 2.13 แสดงการแก้ปัญหาแบบสตีเพสต์-แอสเซนต์

การแก้ปัญหาด้วยฮิลไคลมิ่งนั้น แม้จะมีข้อดีหลายประการ แต่การแก้ด้วยวิธีนี้ก็ยังมีปัญหาที่อาจจะเกิดขึ้นคือ

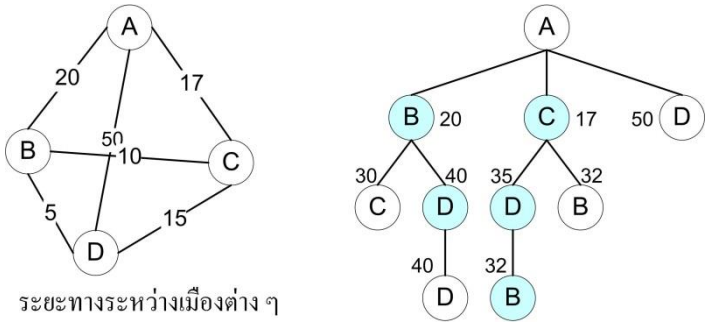
- **โลคอลแม็กซิมั่ม (Local Maximum) :** เป็นส่วนที่บอกสถานะที่ดีที่สุดเมื่อเทียบกับสถานะข้างเคียงเท่านั้น แต่ถ้าเทียบกับสถานะอื่นๆ ที่อยู่ห่างออกไป หรือการตรวจสอบกับโหนดที่อยู่ขั้น (Generation) ต่อไปแล้ว ไม่แน่ว่าจะได้ผลออกมาดีที่สุด ในรูปที่ 2.14 (a) เราจะเห็นว่าที่โหนด B และ C มีค่าเท่ากับ 5 และ 7 ตามลำดับ ซึ่งค่าของ B น้อยกว่า C แต่ว่าลูกของ B กลับมีค่ามากกว่าลูกของ C ดังนั้นในการเลือกโหนดที่ดีที่สุดระหว่าง B และ C คือ B จะมีผลให้โหนด G ซึ่งควรจะเป็นโหนดที่ดีที่สุดไม่ได้รับการพิจารณา เพราะในระดับบน โหนด C ไม่ได้รับเลือก
- **เพลทู (Plateau) :** ในกรณีที่ปัญหาอยู่บนระดับเดียวกัน และผลของการหาค่าจากฮิวริสติกฟังก์ชันที่ได้คะแนนเท่ากันหมด จะไม่สามารถตัดสินใจได้ว่าจะเลือกเส้นทางเส้นใด ในรูปที่ 2.14(b) จะเห็นว่าโหนดลูกทั้ง 3 ของ A มีค่าเท่ากันหมด ทำให้เราไม่สามารถเลือกโหนดที่ดีที่สุดได้ ซึ่งเรื่องนี้เราสามารถแก้ได้ด้วยการสร้างโหนดลูกให้กับโหนดเหล่านั้น แล้ววัดกันที่โหนดลูก ดูว่าโหนดใดดีกว่า บางครั้งโหนดที่สร้างออกมามีมาก ทำให้สร้างโหนดลูกของโหนดเหล่านั้นไม่ไหว เราอาจจะใช้วิธีการเลือกแบบสุ่มก็ได้



รูปที่ 2.14 ปัญหาของการค้นหาแบบฮิลไคลมิ่ง

- **ริดจ์ (Ridge) :** เป็นกรณีที่การค้นหาได้เดินไปในทางที่ดีที่สุดตลอดเวลาเมื่อเปรียบเทียบกับโหนดข้างเคียง แต่เมื่อเทียบกับโหนดที่ดีที่สุดในระดับที่ผ่านมาจะแย่ลงเรื่อยๆ การค้นหาในลักษณะนี้จะมีลักษณะเช่นเดียวกับการเดินบนสันเขาเล็กๆ ที่ดูเหมือนว่าจะพาไปสู่ยอดเขาได้ แต่ยิ่งเดินยิ่งต่ำลง ในรูปที่ 2.14(c) ถ้าสมมติว่าค่าน้อยคือค่าที่ดีกว่าค่ามาก ดังนั้นในการเลือกแต่ละระดับเราจะเลือกค่าน้อย จากโหนดราก A สร้างลูกออกมาเป็น B, C และ D มีค่าเท่ากับ 10, 3 และ 9 ตามลำดับ เราจะต้องเลือกโหนด C เพราะมีค่าดีที่สุด จากนั้นลูกของ C

ที่ถูกสร้างขึ้นมาคือ E, F และ G มีค่า 4, 8 และ 7 ตามลำดับ เลือกโหนด E ลูกของ E คือ H, I และ J ที่มีค่าเท่ากับ 7, 5 และ 6 ตามลำดับ เลือก I เพราะมีค่าที่ดีที่สุด เราจะเห็นว่าในแต่ละชั้น โหนดที่ดีที่สุดที่ถูกเลือกจะมีค่าแย่ลงไปเรื่อยๆ คือจาก C ที่มีค่า 3 ถัดมา E มีค่า 4 และ I มีค่า 5 ตามลำดับ (อย่าลืมว่าเรากำลังเลือกโหนดที่มีค่าน้อยซึ่งถือว่าดี) ซึ่งค่าที่ได้จะแย่ลงเรื่อยๆ แต่เมื่อเปรียบเทียบกับโหนดอื่นๆ ที่อยู่ในระดับเดียวกันแล้วยังถือว่าดีกว่า การแก้ปัญหาในเรื่องนี้ทำได้ยากมาก



รูปที่ 2.15 แสดงการค้นหาแบบบีม

เพื่อที่จะแก้ปัญหาดังกล่าวอย่างง่ายๆ ในการเลือกลูกที่ออกมาแต่ละครั้งของฮิลไคลมิ่ง แทนที่จะเลือกลูกที่ดีที่สุดออกมาเพียงตัวเดียว เราก็ทำการเลือกลูกที่ดีที่สุดออกมามากกว่า 1 ตัว เช่น ให้เท่ากับ w ตัว เราเรียกการค้นหาแบบนี้ว่า *การค้นหาแบบบีม (Beam Search)* และเราเรียกค่า w นี้ว่า *บีมวิดธ์ (Beam Width)* เพื่อแก้ปัญหาโลคอลแม็กซิมั่ม และค่าของ w ก็ขึ้นอยู่กับว่าเราเป็นผู้กำหนด เช่น ถ้าเราให้ w = 2 ในการเลือกแต่ละระดับ เราจะเลือกลูกที่ดีที่สุด 2 ตัวเสมอ รูปที่ 2.15 แสดงการทำงานของการค้นหาแบบบีมที่กำหนดให้ w มีค่าเท่ากับ 2

2.3.8 อัลกอริธึมเอ-สตาร์

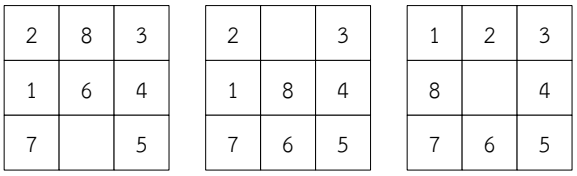
การค้นหาด้วยอัลกอริธึมเอ-สตาร์ (A* Algorithm) (Hart, 1968) เป็นอีกแบบของการค้นหาแบบดีที่สุดก่อน วิธีการเลือกโหนดที่จะใช้ในการดำเนินการต่อจะพิจารณาจากโหนดที่ดีที่สุด ความแตกต่างระหว่างเอ-สตาร์กับการค้นหาแบบดีที่สุดก่อนคือส่วนของฮิวริสติกฟังก์ชัน การค้นหาแบบดีที่สุดก่อน ค่าของฮิวริสติกฟังก์ชันจะวัดจากโหนดราก (สถานะเริ่มต้น) ถึงโหนดปัจจุบัน (สถานะปัจจุบัน) แต่

ค่าฮิวริสติกฟังก์ชันของเอ-สตาร์จะเกิดจากการรวมกันของค่า 2 ค่าคือ ค่าที่วัดจากสถานะปัจจุบันไปยังสถานะเริ่มต้น รวมกับค่าที่วัดจากสถานะปัจจุบันไปยังสถานะเป้าหมาย ถ้ากำหนดให้ตัวแปร f แทนค่าของฮิวริสติกฟังก์ชันของเอ-สตาร์ และ g เป็นฟังก์ชันที่ใช้วัดค่า (Cost) จากสถานะเริ่มต้นจนถึงสถานะปัจจุบัน h′ เป็นฟังก์ชันที่ใช้วัดค่าจากสถานะปัจจุบันถึงสถานะเป้าหมาย ดังนั้น

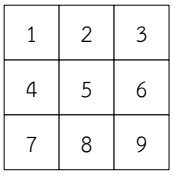
$$f = g + h'$$

อัลกอริธึม เอ-สตาร์: ให้ OPEN เป็น list ของโหนดที่เกิดใหม่และยังไม่ผ่านการสำรวจ และ CLOSED เป็น list ของโหนดที่สำรวจแล้ว ให้โหนดเป้าหมายชื่อ node_goal และโหนดเริ่มต้นชื่อ node_start ใส่ node_start เข้าไปใน OPEN list ให้ดำเนินขั้นตอนดังต่อไปนี้จนกว่าจะไม่มีโหนดใน OPEN

1. เลือกโหนด n จาก OPEN โดยที่โหนด n เป็นโหนดที่มีค่า f ต่ำสุด และเพิ่ม n เข้าไปใน CLOSED
2. ถ้า n เหมือนกับ node_goal แสดงว่าพบคำตอบ ดังนั้นให้ค่าของ n ออกมา Solution(n) ถ้า n ไม่ใช่ node_goal ให้สร้างลูกของ n ออกมาเป็น n′
3. ให้ดำเนินตามขั้นตอนดังต่อไปนี้สำหรับโหนดลูกทุกตัวของ n ที่เป็น n′
   3.1. ให้สร้างเส้นเชื่อม (Link) จากโหนด n′ ไปหาโหนดพ่อแม่ (Parent Node) ของมันคือ n หาค่าของ h(n′) โดยคำนวณค่าจากโหนด n′ ไปยัง node_goal หาค่า g(n′) ใหม่จากการรวมค่า g(n′) เดิมกับค่าที่ได้จากโหนด n′ ไปยัง n และคำนวณค่า f(n′) จากการรวม g(n′) กับ h(n′)
   3.2. ถ้า n′ อยู่ใน OPEN และมีโหนดใดโหนดหนึ่งที่มีค่า f ดีเท่ากันหรือดีกว่า f(n′) ให้ทิ้ง n′
   3.3. ถ้า n′ อยู่ใน CLOSED และมีโหนดใดโหนดหนึ่งที่ค่า f เท่ากันหรือดีกว่า f(n′) ให้ทิ้ง n′
   3.4. ถ้า n′ ดีกว่า ให้เอาโหนด n′ ที่อยู่ใน OPEN และ CLOSED ที่เหมือนกันออกไป
   3.5. เพิ่ม n′ เข้าไปใน OPEN



สถานะเริ่มต้น สถานะปัจจุบัน สถานะเป้าหมาย

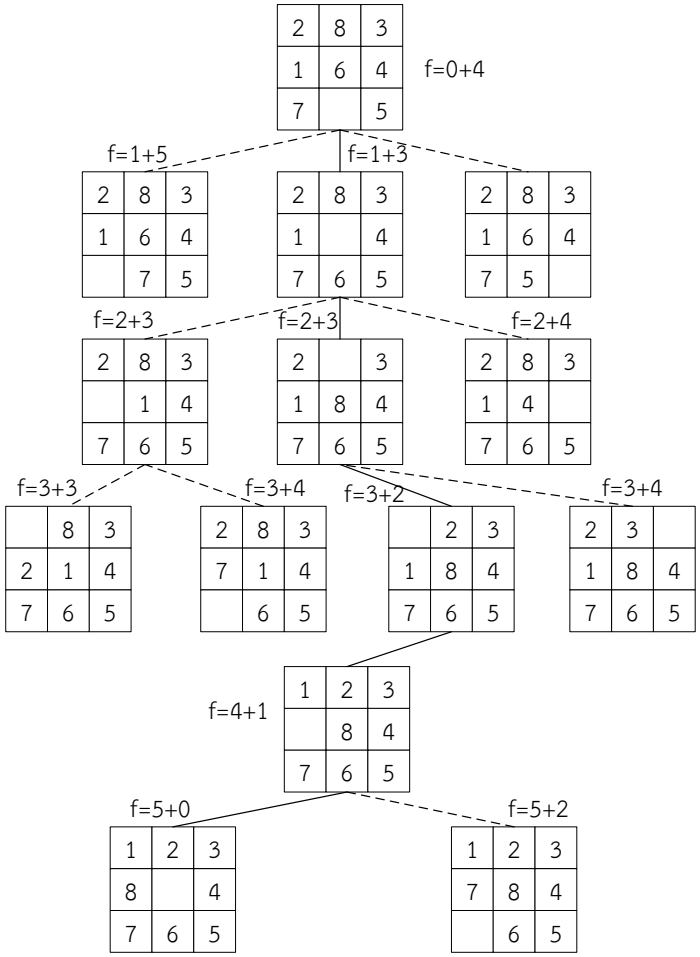


ตำแหน่งของตาราง

การหาค่าของ g ทำได้โดยการเปรียบเทียบสถานะปัจจุบันกับสถานะเริ่มต้น คือดูว่าช่องตารางใดที่ไม่ตรงกัน ให้นับเป็น 1 คะแนน ยกเว้นช่องว่างของสถานะปัจจุบัน ถ้าดูจากตำแหน่งบนตาราง จะเห็นว่าช่องที่ไม่ตรงกันคือ ช่อง 5 และ 8 ได้ g=2 สำหรับ h’ เป็นการเปรียบเทียบสถานะปัจจุบันกับสถานะเป้าหมาย มีช่องที่ไม่ตรงกันคือ ช่องที่ 1, 4 และ 5 ได้ h'=3 รวม f=2+3 = 5

รูปที่ 2.16 การกำหนดสถานะต่างๆ และวิธีการหาค่าของตาราง

ตัวอย่าง การหาคำตอบแบบเอ-สตาร์ของปัญหา 8-Puzzle ในการคำนวณค่าของ g คือการเปรียบเทียบสถานะปัจจุบันกับสถานะเริ่มต้น และค่า h′ คือการเปรียบเทียบสถานะปัจจุบันกับสถานะเป้าหมาย ซึ่งในการคำนวณค่า g และ h′ มีวิธีการดังรูปที่ 2.6



รูปที่ 2.17 การแก้ปัญหา 8-Puzzle ด้วยเอ-สตาร์ (Luger, 1997)

สำหรับวิธีการแก้ปัญหา 8-Puzzle ด้วยวิธีการของเอ-สตาร์จะเป็นดังปรากฏในรูปที่ 2.17 เริ่มจากสถานะเริ่มต้นเป็นโหนดราก จากนั้นสร้างลูกที่เป็นสถานะใหม่ออกมา 3 โหนดมีค่าของ f คือ 6 4 และ 6 ผลคือโหนดที่มีค่า 4 ดีที่สุดให้เลือกเป็นโหนดปัจจุบัน แล้วสร้างสถานะใหม่ออกมาได้ 3 สถานะ หาค่าของ f ออกมาได้ 5 5 และ 6 ให้เลือกโหนดที่มีคะแนน 5 ทั้งสองโหนด เป็นสถานะปัจจุบัน สร้างโหนดลูกออกมาจากโหนดปัจจุบันทั้ง 2 โหนดออกมาได้ 4 โหนด มีค่า f เท่ากับ 6 7 5 และ 7 โหนดที่มีค่าเท่ากับ 5 เป็นโหนดที่ดีที่สุด ให้เป็นโหนดปัจจุบัน สร้างโหนดลูกออกมาได้ 1 โหนดมีค่า f เท่ากับ 5 เช่นกัน แต่ค่า h’ ดีกว่าโหนดปัจจุบัน ให้เลือกโหนดสร้างใหม่เป็นโหนดปัจจุบัน สร้างโหนดลูกออกมาได้

อีก 2 โหนด เราจะได้โหนดลูกใหม่เหมือนกับโหนดเป้นหมาย โดยที่มีค่า h’ เป็น 0 คือไม่มีความแตกต่างระหว่างโหนดปัจจุบัน และโหนดเป้าหมาย

2.3.9 มีน-เอ็นด์อะแนไลซิส

มีน-เอ็นด์อะแนไลซิส (Mean-End Analysis) เป็นเทคนิคในการแก้ปัญหาแบบตัวแก้ปัญหาทั่วไป (General Problem Solver: GPS) เสนอโดยเนเวล ซอว์ และไซม่อน (Newell, 1960) การแก้ปัญหาแบบนี้จะต่างกับการแก้ปัญหาแบบปริภูมิสถานะ (State Space) ที่จะต้องเลือกว่าทางเดินใดที่ทำการทดลองต่อจากบรรดาทางเดินที่ถูกสร้างขึ้นมาใหม่ หรือกล่าวอีกนัยหนึ่งก็คือ การเลือกว่าจะเลือกโหนดลูกโหนดใดเพื่อดำเนินการต่อไป

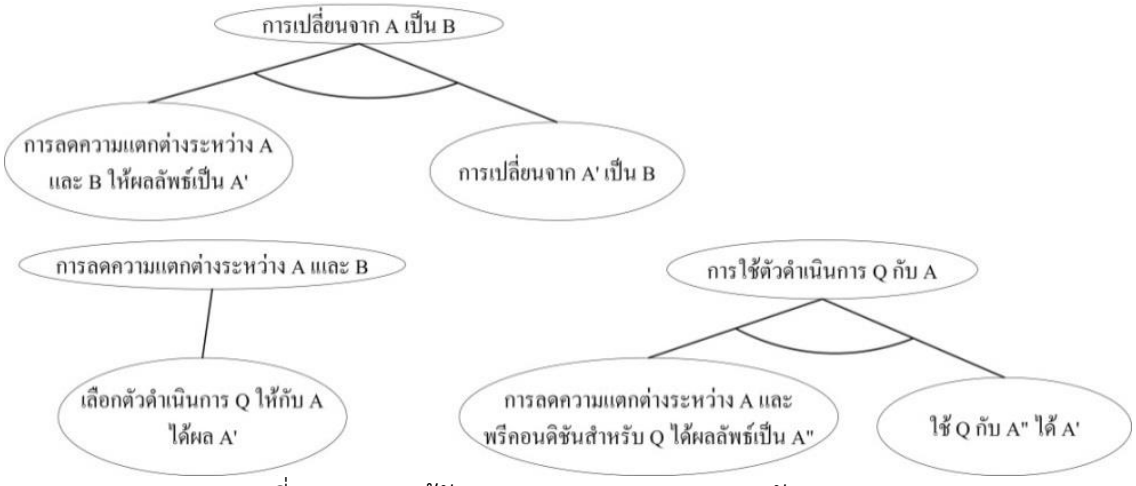
การแก้ปัญหาแบบมีน-เอ็นด์อะแนไลซิส เป็นการแก้ปัญหาด้วยการลดความแตกต่างระหว่างสถานะปัจจุบันกับสถานะเป้าหมายที่ต้องการ โดยอาศัยตัวดำเนินงาน หรือกฎ เพื่อที่จะสร้างสถานะใหม่จากสถานะปัจจุบันในปริภูมิคำตอบ สถานะที่เกิดใหม่นี้จะต้องเป็นสถานะที่อยู่ในระหว่างกลางของสถานะปัจจุบันกับสถานะเป้าหมาย เพื่อช่วยลดความแตกต่าง ในการสร้างสถานะใหม่ อัลกอริธึมจะเลือกตัวดำเนินการที่เกี่ยวข้องเพียงตัวเดียวเท่านั้น เพื่อใช้กับสถานะปัจจุบัน ถ้าหากว่ามีตัวดำเนินการหรือกฎมากกว่าหนึ่งตัวที่สามารถใช้กับสถานะปัจจุบันได้ ตัวดำเนินการหรือกฎเหล่านี้จะถูกจัดลำดับการทำงานด้วยวิธีการค้นหาแบบลึกก่อน (Depth First Search) ตราบใดที่ตัวดำเนินการที่เลือกไว้สามารถแก้ปัญหาได้ และทางเดินนั้นยังใช้ได้ ในการค้นหาลำดับของตัวดำเนินงานนั้นต้องอาศัยการสำรอง (Back Up) เพื่อป้องกันความยุ่งยากที่อาจจะเกิดขึ้นในกรณีที่ทางเดินปัจจุบันใช้ไม่ได้

ลักษณะพิเศษของมีน-เอ็นด์อะแนไลซิสก็คือ เมื่อตัวดำเนินการที่เกี่ยวข้องถูกเลือกมาเพื่อลดความแตกต่างนั้น อาจจะไม่สามารถทำให้ผลลัพธ์ของสถานะปัจจุบันกลายเป็นสถานะเป้าหมายโดยทันที แทนที่จะเลิกใช้ตัวดำเนินการนี้ มีน-เอ็นด์อะแนไลซิสจะพยายามเปลี่ยนผลลัพธ์ไปเป็นจุดประสงค์อื่นแทน และทำให้จุดประสงค์นี้กลายเป็นผลลัพธ์ที่เหมาะสมสำหรับการเป็นอินพุตของตัวดำเนินการตัวต่อไป การดำเนินการจะวนเช่นนี้ไปเรื่อยๆ จนกว่าผลลัพธ์จะเป็นจุดประสงค์เป้าหมายที่เราต้องการ กลวิธีนี้จะเป็นแบบการเรียกซ้ำ (Recursive)

วิธีการ ผลลัพธ์คือผลของการเปลี่ยนสถานะจากจุดประสงค์ปัจจุบัน ซึ่งผลลัพธ์นี้อาจจะเป็นจุดประสงค์ที่ต้องการได้ ลักษณะสำคัญของผลลัพธ์ที่เกิดจากวิธีการนี้ที่สำคัญมี 3 อย่างคือ

1. การเปลี่ยน (Transforming) จากจุดประสงค์ A เป็นจุดประสงค์ B
2. การลด (Reducing) ความแตกต่างระหว่างจุดประสงค์ A และจุดประสงค์ B โดยการดัดแปลงจุดประสงค์ A
3. การใช้งาน (Applying) ตัวดำเนินการ Q กับจุดประสงค์ A

สำหรับวิธีการแก้ปัญหาของการลดทอนปัญหาหรือมีน-เอ็นด์อะแนไลซิส ในการแก้ปัญหาแบบตัวแก้ปัญหาทั่วไป หรือ GPS สามารถสรุปได้ดังแสดงในรูปที่ 2.18



รูปที่ 2.18 การแก้ปัญหา 3 แบบของการลดทอนปัญหา (Barr, 1982)

การเลือกตัวดำเนินการ การพยายามเปลี่ยนจุดประสงค์ A ไปเป็นจุดประสงค์ B วิธีการเปลี่ยนจะใช้กระบวนการเปรียบเทียบ เพื่อที่จะหาความแตกต่างของจุดประสงค์ทั้ง 2 อัน ซึ่งความแตกต่างที่เป็นไปได้ทั้งหมดจะถูกนำมาเรียงลำดับไว้ โดยการจัดให้อยู่ในรูปของตารางที่เชื่อมต่อกับตัวดำเนินการหรือกฎที่สามารถลดความแตกต่างในแต่ละแบบไว้ ในระหว่างการแก้ปัญหาด้วยมีน-เอ็นด์อะแนไลซิสนี้มีสิ่งที่พึงระวังในการป้องกันการเดินทางบนทางเดินที่เราไม่ต้องการ (ผิดทาง) ดังนี้

1. ผลลัพธ์แต่ละอันจะต้องมีความแตกต่างน้อยกว่าผลลัพธ์เดิมคือการลดความแตกต่าง

2. สำหรับแต่ละคู่ของโหนด AND ที่แทนผลลัพธ์ย่อยที่ถูกสร้างโดยการเปลี่ยนและการใช้นั้น ผลลัพธ์ย่อยอันที่สองจะนำมาใช้ได้ดีกว่า
3. จุดประสงค์ที่ถูกสร้างใหม่ล่าสุด ไม่ควรจะใหญ่กว่าจุดประสงค์ที่ปรากฏผลลัพธ์ที่อยู่สูงสุดหรือจุดประสงค์เริ่มต้น
4. ผลลัพธ์ใดที่เคยปรากฏมาแล้ว ไม่ควรจะปรากฏอีกในการแก้ปัญหาแต่ละครั้ง

ตัวอย่าง การใช้งานของหุ่นยนต์ ตารางที่ 2.9 คำสั่งการทำงานของหุ่นยนต์ เป็นกฎหรือตัวดำเนินการที่ใช้กับการเคลื่อนที่ของหุ่นยนต์ ซึ่งใช้สำหรับลดความแตกต่างระหว่างจุดประสงค์ปัจจุบันกับจุดประสงค์ที่ต้องการหรือสถานะเป้าหมาย ภายในกฎจะมีองค์ประกอบที่สำคัญคือ ตัวดำเนินการ (Operator) พรีคอนดิชั่น (Precondition) และผลลัพธ์ (Result) วิธีการของกฎคือ ตัวดำเนินการจะใช้การได้ก็ต่อเมื่อพรีคอนดิชั่นนั้นถูกทำให้เป็นจริง และผลของการดำเนินการก็คือส่วนที่เป็นผลลัพธ์ เช่น คำสั่งที่ใช้สำหรับการเคลื่อนย้ายวัตถุ PUSH(obj,loc) มีพรีคอนดิชั่นเป็น at(robot,obj)^large(obj)^ clear(obj)^armempty ซึ่งหมายความว่าหุ่นยนต์จะต้องอยู่ตรงที่วัตถุ at(robot,obj) วัตถุต้องชิ้นใหญ่ large(obj) วัตถุต้องไม่มีอะไรวางทับอยู่ clear(obj) และแขนหุ่นยนต์ต้องว่างเปล่า armempty มีผลลัพธ์เป็น at(obj,loc)^at(robot,loc) ซึ่งหมายถึงวัตถุก็จะอยู่ในที่ที่กำหนด และหุ่นยนต์ก็จะอยู่ ณ ที่นั้นด้วย สำหรับตัวดำเนินการอื่นๆ ก็จะมีวิธีการอธิบายเช่นเดียวกัน

ตัวดำเนินการ	พรีคอนดิชั่น	ผลลัพธ์
PUSH(obj, loc)	at(robot,obj)^large(obj)^clear(obj)^ armempty	at(obj, loc)^at(robot, loc)
CARRY(obj, loc)	at(robot, obj)^small(obj)	at(obj,loc)^at(robot, loc)
WALK(loc)	None	at(robot, loc)
PICKUP(obj)	at(robot, obj)	holding(obj)
PUTDOWN(obj)	holding(obj)	~holding(obj)
PLACE(obj1, obj2)	at(robot, obj2)^holding(obj1)	on(obj1, obj2)

ตารางที่ 2.9 คำสั่งการทำงานของหุ่นยนต์ (Rich, 1991)

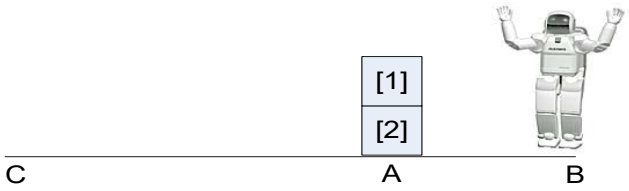
	PUSH	CARRY	WALK	PICKUP	PUTDOWN	PLACE

Move object	*	*				
Move robot			*			
Clear object				*		
Get object on object						*
Get arm empty					*	*
Be holding object				*		

ตารางที่ 2.10 ตารางแสดงความแตกต่าง (E. and Knight, K. Rich, 1991)

ในตารางที่ 2.10เป็นตารางที่แสดงความแตกต่าง (Difference Table) ของการเคลื่อนที่ของหุ่นยนต์ จะเห็นว่าในการ Move object นั้นจะต้องอาศัยคำสั่ง PUSH และ CARRY เท่านั้น ในการเคลื่อนย้ายหุ่นยนต์ Move robot จะต้องอาศัย WALK และในการ Clear object จะต้องอาศัย PICKUP เป็นต้น ในที่นี้ Move object, Move robot และ Clear object คือจุดประสงค์ และ PUSH, CARRY และ WALK คือตัวดำเนินการ ในตารางนี้ * คือตัวเชื่อมที่บอกว่าตัวดำเนินการใดใช้สำหรับจุดประสงค์ใด สังเกตว่าตัวดำเนินการอาจจะมี 2 ตัวที่สามารถใช้กับจุดประสงค์เดียว เพื่อลดความแตกต่าง

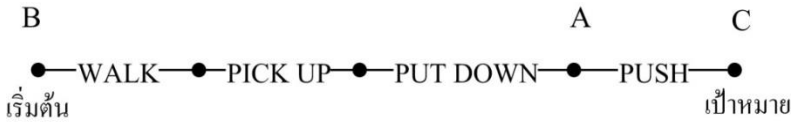
ตัวอย่าง สมมติว่าเราต้องการให้หุ่นยนต์ย้ายแท่งเหล็กจาก A ไปยัง C โดยที่บนแท่งเหล็ก [2] มีเหล็กอีกแท่ง [1] วางทับอยู่ กำหนดให้ว่า ปัจจุบันหุ่นยนต์ยืนอยู่ที่ B และมือของหุ่นยนต์ว่างเปล่า ดังแสดงในรูปที่ 2.19 ดังนี้



รูปที่ 2.19 แสดงสถานะเริ่มต้นของแท่งเหล็กและหุ่นยนต์

ในการย้ายแท่งเหล็กนี้ ก่อนอื่นจะต้องเคลื่อนย้ายแท่งเหล็กแท่งบน [1] ออกก่อน การเคลื่อนย้าย หรือ Move Object จะต้องเลือก PUSH หรือ CARRY โดยดูจากตารางความแตกต่างในตารางที่ 2.10 ทดลองเลือก PUSH ก่อน ในการที่จะใช้ตัวดำเนินการ PUSH ได้ จะต้องทำให้เงื่อนไขของกฎที่เป็นพรี

คอนดิชั่นให้บรรลุก่อนจากกฎในตารางที่ 2.9 พรีคอนดิชั่นคือ at(robot,obj)^large(obj)^clear(obj)^ armempty ในกรณีของ at(robot,loc) หรือ at(robot,obj) นั้นจะต้องอาศัย WALK(loc) ดูตารางที่ 2.9 สำหรับ clear(obj) จะต้องอาศัย PICKUP(obj) โดยดูจากตารางแสดงความแตกต่าง ดังนั้นเราเลือกการลดทอนอันแรกคือ at(robot,obj) มากระทำก่อนโดยอาศัย WALK(loc) ที่ไม่มีพรีคอนดิชั่น เมื่อเลือก WALK(loc) แล้วจะได้ผลลัพธ์คือ at(robot,loc) จากนั้นเลือก PICKUP(obj) ซึ่งจะได้ผลลัพธ์เป็น holding(obj) แต่พรีคอนดิชั่นของ PUSH ที่เราจะกระทำจะต้องเป็น armempty การแก้ปัญหา holding(obj) จึงต้องเลือก PUTDOWN(obj) จะได้ผลลัพธ์ ~holding(obj) ซึ่งก็หมายถึง clear(obj) และ armempty ดังนั้นเงื่อนไขของพรีคอนดิชั่นของ PUSH(obj,loc) ก็ประสบความสำเร็จ เราจึงเลือกตัวดำเนินการ PUSH(obj,loc) ได้ และผลลัพธ์ก็ออกมาเป็น at(obj,loc) และ at(robot,loc) นั่นคือการย้ายแท่งเหล็กนี้ก็จะบรรลุผล ขั้นตอนของการใช้ตัวดำเนินการเป็นตามรูปที่ 2.20



รูปที่ 2.20 สรุปขั้นตอนของการใช้ตัวดำเนินการย้ายแท่งเหล็ก

2.3.10 การเล่นเกม

ในการศึกษาเกี่ยวกับเรื่องการเล่นเกม (Game Playing) ด้วยการใช้คอมพิวเตอร์ช่วยได้มีการทำกันมานานแล้ว เริ่มตั้งแต่สมัยแรกเมื่อปี พ.ศ. 2493 คล็อด แชนนอน (Shannon, 1950) ได้เขียนบทความที่อธิบายถึงวิธีการที่สามารถใช้เครื่องจักรในการเล่นหมากรุก และในอีก 2-3 ปีถัดมา อลัน ทิวริ่ง (Turing, 1950) ได้อธิบายวิธีการเขียนโปรแกรมเล่นหมากรุก และในต้นทศวรรษที่ 1990 อาร์เทอร์ แซมมูเอล (Samuel, 1959) ได้สร้างโปรแกรมที่ใช้สำหรับเล่นเกมได้เป็นคนแรก

ลองพิจารณาดูการเล่นเกม เช่น การเล่นหมากรุก หมากฮอส ที่มีผู้เล่นสองคนที่แข่งกัน โดยการผลัดกันเป็นผู้เดินหมากฝ่ายละครั้ง ในอดีตที่ผ่านมา การเล่นเกมเป็นเรื่องที่ได้รับการพิสูจน์แล้วว่าเป็นเรื่องที่ท้าทายมากสำหรับผู้ที่เขียนโปรแกรม ในบางครั้ง แม้ว่าเราจะรู้วิธีการที่จะเอาชนะได้เป็นอย่างดีแล้วก็ตาม แต่การเอาชนะได้ก็ต้องอาศัยการคำนวณด้วยคอมพิวเตอร์เป็นเวลานานมาก ในปัจจุบัน การสร้างเกมด้วยโปรแกรมคอมพิวเตอร์เป็นเรื่องที่มีความสำเร็จมากพอสมควร เบื้องหลังของความสำเร็จใน

เรื่องนี้มาจากการพัฒนาทางด้านความเร็วของฮาร์ดแวร์ของคอมพิวเตอร์ แต่นั่นไม่ใช่ความสำเร็จของเรื่องราวทั้งหมด ความสำเร็จอีกส่วนหนึ่งและเป็นส่วนที่สำคัญมากไม่แพ้การพัฒนาด้านความเร็วของคอมพิวเตอร์ก็คือ การพัฒนาด้านอัลกอริธึมของซอฟต์แวร์ ในปัจจุบันมีความคิดและทฤษฎีใหม่ๆ ทางด้านฮิวริสติกที่มาช่วยในการควบคุมการค้นหาของการเล่นเกมของคอมพิวเตอร์ การที่เราจะไปดูเรื่องของการเล่นเกมอย่างเต็มรูปแบบ เราควรทำความเข้าใจเทคนิคมาตรฐานของการเล่นเกมกันเสียก่อน

กระดานสถิต (Static Board) ในการเล่นเกมที่มีผู้เล่นสองคน ที่แต่ละฝ่ายต่างก็ต้องการชัยชนะ และต่างฝ่ายต่างก็ต้องการได้คะแนนดีที่สุด ถ้าสมมติว่าเรากำลังเล่นหมากรุกอยู่ การที่เราเดินหมากไปหนึ่งตา เราต้องสามารถประเมินได้ว่าการเดินหมากตานั้นจะทำให้เราได้คะแนนเป็นเท่าไร และทำให้ฝ่ายตรงข้ามได้คะแนนเท่าไร ซึ่งจะเป็นตัวช่วยให้เราสามารถคาดการณ์ถึงชัยชนะได้ ถ้าเราคิดให้ลึกกว่านั้นอีกว่า เวลาจะเดินหมากแต่ละตัว เราจะเลือกตัวใดในการเดินแต่ละครั้ง เพื่อทำให้เราได้เปรียบคู่แข่งมากที่สุด ซึ่งหมายความว่า เราจะต้องหาความเป็นไปได้ของตาเดินในตารางหมากรุกทั้งหมด และคำนวณออกมาเป็นคะแนน การทำเช่นนี้เป็นสิ่งที่เป็นไปได้ยาก เพราะตาเดินที่เป็นไปได้ของหมากรุกมีจำนวนมาก ในขณะเดียวกัน ฝ่ายตรงข้ามก็มีตาเดินที่เป็นไปได้จำนวนมากเช่นกัน ยิ่งถ้าเราต้องการคำนวณล่วงหน้าไปหลายชั้น ยิ่งทำได้ยากมากขึ้น ถ้าเราให้ตาเดินที่เป็นไปได้ทั้งหมดจากการเดินตาแรกไปจนตาสุดท้ายที่จะได้รับชัยชนะเป็นปริภูมิปัญหา (Problem Space) ปริภูมิปัญหานี้จะใหญ่มาก โดยเฉพาะอย่างยิ่งในการเดินหมากรุก และการทำเช่นนั้นก็ไม่ได้หมายความว่าจะทำให้เราชนะเสมอ เพราะการเล่นหมากรุกเป็นสิ่งที่สามารถเปลี่ยนแปลงยุทธวิธีได้ตลอดเวลา ดังนั้นในการคิดคะแนนเรามักจะคิดจากเส้นทางที่เป็นไปได้จำนวนหนึ่งเท่านั้น และไม่จำเป็นที่จะต้องคำนวณตั้งแต่ต้นจนจบ การคำนวณในลักษณะนี้มีประโยชน์ต่อการประเมิน แต่ไม่ใช่การคิดที่สมบูรณ์แบบ ซึ่งคะแนนที่คิดคำนวณในลักษณะนี้จะเรียกว่า *การประเมินกระดานสถิต (Static Board Evaluation)*

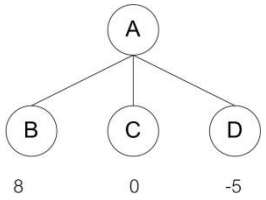
การประเมินกระดานสถิต เป็นกระบวนการฮิวริสติกที่ใช้แสดงสถานะของหมากบนกระดานนั้นออกมาเป็นคะแนน ซึ่งเราจะเรียกค่านี้ว่า *ค่าสถิต (Static Value)* ในการเดินหมากแต่ละครั้ง ค่าที่ได้จากการประเมินกระดานสถิตจะขึ้นอยู่กับตำแหน่งของหมากต่างๆ บนกระดาน เมื่อมีหมากตัวใดตัวหนึ่งเปลี่ยนไป ค่าที่ได้จะเปลี่ยนไปด้วย ในการใช้กระบวนการประเมินกระดานสถิต กระบวนการนี้ไม่ใช่ใช้ได้เฉพาะกับการเล่นหมากรุกเท่านั้น แต่สามารถประยุกต์ใช้กับการเล่นเกมต่างๆ ได้ทุกชนิด ในการคำนวณ

ค่าของกระดานสถิต ค่าสถิตนี้จะถูกจำแนกออกเป็น 2 ชุดคือ ชุดที่เป็นค่าของฝ่ายเราและชุดที่เป็นค่าของฝ่ายตรงข้าม และค่าสถิตรวมของทั้งกระดานจะเกิดจากผลต่างของค่าของทั้งสองชุด สำหรับการคำนวณโดยทั่วไปแล้ว เราจะให้ชุดหนึ่งเป็นจำนวนบวก และอีกชุดหนึ่งเป็นจำนวนลบ สำหรับค่าที่ออกเป็น 0 ถือว่าทั้งสองฝ่ายเสมอกัน

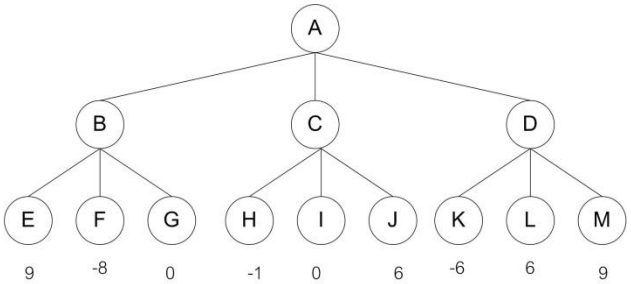
ค่ามินแม็กซ์ (Minmax Value) ในการเล่นเกม กระบวนการหาตาเดินที่ดีที่สุดจะแตกต่างกันในแต่ละระดับ การเลือกตาเดินที่ดีที่สุดจะมีวิธีการพิจารณาที่ต่างกัน ขึ้นอยู่กับว่าใครเป็นผู้ที่จะเล่นในตานั้น ถ้าฝ่ายเราเป็นฝ่ายเล่น การเลือกตาเดินที่ดีที่สุดสำหรับเราก็จะเลือกจากโหนดที่จะทำให้เรามีคะแนนสูงสุดเท่าที่เป็นไปได้ แต่ถ้าตานั้นเป็นของฝ่ายตรงกันข้าม เราก็ต้องคิดว่าฝ่ายตรงกันข้ามจะเลือกโหนดที่ดีที่สุดสำหรับเขา และทำให้เกิดผลแย่ที่สุดสำหรับเรา ถ้าเราจะเทียบเรื่องนี้เป็นคะแนน เมื่อถึงตาของเราเป็นผู้เดิน เราจะเลือกตัวที่ทำให้เราได้คะแนนมากที่สุด พอถึงตาของเขา เขาจะเลือกโหนดที่ทำให้เราได้คะแนนต่ำที่สุด และทำเช่นนี้ไปเรื่อยๆ จนกว่าจะรู้ผลแพ้ชนะ การทำเช่นนี้ก็เท่ากับว่าในตาหนึ่งเราเลือกโหนดที่ให้กระดานสถิตมีคะแนนสูงสุด ในอีกตาหนึ่งเราเลือกโหนดที่ให้กระดานสถิตมีคะแนนต่ำสุด เราเรียกวิธีการเช่นนี้ว่า *มินแม็กซ์ (Minmax)* และคะแนนของการคิดค่าบนกระดานจะเรียกว่า *ค่ามินแม็กซ์ (Minmax Value)* ในการเล่นเกม ถ้าผลของการประเมินกระดานสถิตออกมามีค่ามากกว่า 0 มาก จะหมายความว่าผู้เล่นที่ทำคะแนนสูง (Max Play) เป็นฝ่ายได้เปรียบ ในทางกลับกัน ถ้าการประเมินกระดานสถิตออกมาเป็นลบมาก หมายถึงผู้เล่นที่ทำคะแนนต่ำเป็นฝ่ายได้เปรียบ วิธีการหาค่ามินแม็กซ์นี้นำเสนอครั้งแรกโดยอลัน ทิวริ่ง (Alan Turing) เมื่อทศวรรษที่ 1950

สำหรับโหนดที่ตำแหน่งต่างๆ ที่สามารถไปถึงได้จากตำแหน่งเริ่มต้นใดๆ ที่กำหนดไว้ของต้นไม้ที่มีรากเป็นตำแหน่งเริ่มต้น และลูกของโหนดที่ตำแหน่งใดๆ สามารถไปถึงได้ด้วยการเดินเพียงครั้งเดียว จะเรียกต้นไม้นี้ว่า *ต้นไม้เกม (Game Tree)* และตำแหน่งต่างๆ เหล่านั้นจะเรียกว่า *โหนด (Node)* โหนดที่ผู้เล่นคะแนนสูงเล่นเรียกว่า *แม็กซ์โหนด (Max Node)* สำหรับโหนดที่ผู้เล่นคะแนนต่ำเล่นเรียกว่า *มินโหนด (Min Node)* ลูกของแม็กซ์โหนดคือมินโหนด ในทางกลับกัน ลูกของมินโหนดคือแม็กซ์โหนด การเปลี่ยนจากโหนดหนึ่งไปยังโหนดลูกจะเรียกว่า *ตาเดิน (Ply) บนต้นไม้เกม* สำหรับการเล่นหมากรุกจะเรียกว่า *การเดิน (Move)*

ฟังก์ชันสำหรับการหาค่าสถิตสามารถใช้ได้กับทุกๆ โหนดบนต้นไม้เกม แต่อย่างไรก็ตาม เทคนิคของการค้นหาข้อมูลจะต้องนำมาใช้สำหรับการหาค่าสูงสุดหรือต่ำสุดที่แท้จริงด้วย เช่น ถ้าเราพิจารณาที่แม็กซ์โหนดใดๆ ผู้ที่เล่นค่าสูงสุดจะต้องการเลือกโหนดลูก (ของแม็กซ์โหนด) ที่มีค่าสูงสุด แต่ว่าลูกของแม็กซ์โหนดคือมินโหนด และการเลือกค่าจะต้องเลือกค่าที่ต่ำที่สุด ณ จุดนี้ ดังนั้นการเลือกค่าที่ดีที่สุดของแม็กซ์โหนดอาจจะเป็นผลให้ฝ่ายตรงกันข้ามที่เล่นค่าต่ำทำคะแนนได้ดีกว่าได้ ถ้าการเลือกนั้นไม่ได้อาศัยโหนดลูกมาช่วยพิจารณา ดังตัวอย่างตามรูปที่ 2.21 คือโหนด A เมื่อโหนด A สร้างทางเดินที่เป็นไปได้ของการเล่นเกมออกมาเป็น B, C และ D ซึ่งเป็นรูปแบบของการเดิน สมมติว่าเมื่อเดินมาที่ B ได้ 8 คะแนน เดินมาที่ C ได้ 0 คะแนน ถ้าเดินมาที่ D จะได้ –5 คะแนน



รูปที่ 2.21 เมื่อพิจารณาจากจุดเริ่มต้น

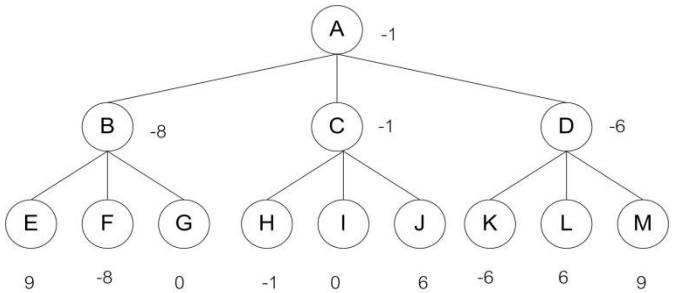


รูปที่ 2.22 เมื่อพิจารณาจากการเดินของฝ่ายตรงข้าม

ในรูปที่ 2.22 ถ้าเราเลือกที่จะเดิน B ตามคะแนนของการเดินเกมในตาแรก เราจะเห็นว่าผลลัพธ์ออกมาเป็น E, F และ G ซึ่งมีคะแนนออกมาเป็น 9, –8 และ 0 ตามลำดับ ในการเลือกในตานี้คือการเลือกโหนด E, F หรือ G เราไม่ได้เป็นฝ่ายเลือก เพราะเราได้เลือกไปก่อนหน้าแล้ว เมื่อฝ่ายตรงข้ามเป็นผู้เลือก เขาจะต้องเลือกโหนดหรือตาเดินที่ทำให้เขาได้เปรียบที่สุดคือโหนด F ซึ่งจะทำให้ฝ่ายตรงข้ามได้คะแนนเป็น –8 ถ้าเราคิดใหม่ ในช่วงที่เราเดินตาแรก เราไม่เลือก B แต่มาเลือก C แทน คะแนน

ที่ฝ่ายตรงข้ามจะได้สูงสุดจะเป็น –1 และถ้าเราเลือก D ให้ตาเดินก่อนหน้า ฝ่ายตรงข้ามจะได้คะแนนสูงสุดเป็น –6 เมื่อพิจารณามาถึงตอนนี้ เราจะทราบได้เลยว่าโหนดที่เราควรเลือกในการเดินครั้งแรกคือ C มากกว่า B ทั้งที่การเดินมาที่ C ในตาแรกจะได้คะแนนเพียง 0 (ดูรูปที่ 2.21) ดังนั้นในการพิจารณาการเล่นเกม เราไม่สามารถดูคะแนนได้จากการเดินเพียงตาเดียวได้ ในทางกลับกัน การพิจารณานี้เราจะต้องดูไปล่วงหน้า และจะสามารถดูไปล่วงหน้ากี่ตาก็ได้

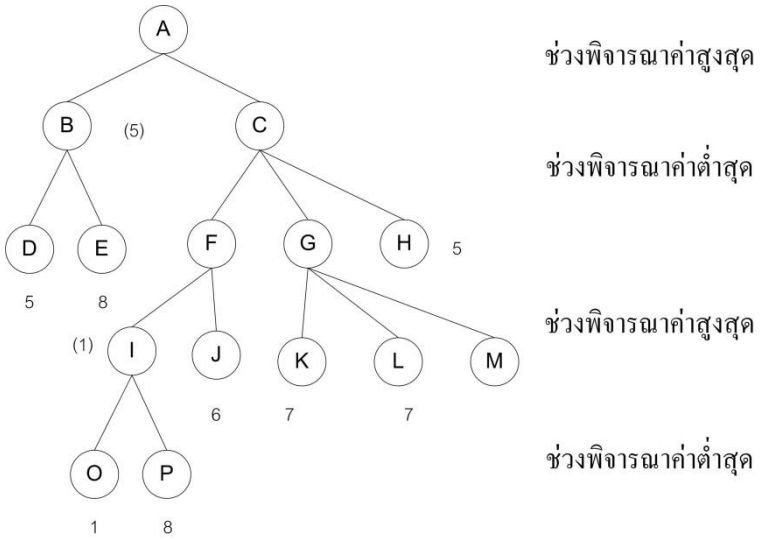
รูปที่ 2.23 ค่าสถิติเป็นคะแนนการคิดค่าของการเดินในแต่ละตา การคิดคะแนนตอนนี้จะดูคะแนนจากโหนดในระดับสุดท้ายที่ฝ่ายตรงข้ามเป็นฝ่ายเลือก ซึ่งเขาจะเลือกตาเดินหรือโหนดที่ดีที่สุดสำหรับตนเอง เมื่อเลือกแล้วก็ส่งผ่านค่าที่ได้ขึ้นไปยังโหนดที่อยู่สูงกว่าตามรูปที่ 2.23 โหนด E, F และ G มีค่าที่ดีที่สุดสำหรับฝ่ายตรงข้ามคือ –8 จากโหนด F ค่า –8 นี้จะถูกส่งไปยัง B สำหรับโหนด H, I และ J ได้ค่าที่ดีที่สุดสำหรับฝ่ายตรงข้ามคือ –1 จากโหนด H และค่า –1 นี้จะถูกส่งไปยัง C และเช่นกัน สำหรับโหนด K, L และ M ได้ค่าที่ดีที่สุดสำหรับฝ่ายตรงข้ามเท่ากับ –6 จากโหนด K และค่านี้จะถูกส่งผ่านไปยัง D เราจะสังเกตได้ว่า การเลือกในระดับนี้ เราเลือกคะแนนที่ดีที่สุดสำหรับฝ่ายตรงข้าม ก็เท่ากับเป็นการเลือกค่าต่ำที่สุด เรียกว่า *ช่วงพิจารณาค่าต่ำสุด* จากโหนด B, C และ D คะแนนที่ได้เป็น –8, –1 และ –6 ตามลำดับ เพื่อทราบว่าที่โหนด A เราจะเลือกโหนดใดระหว่าง B, C หรือ D นั้น เราจะต้องเลือกค่าที่ดีที่สุดสำหรับฝ่ายเรา คือต้องเลือกโหนดที่มีคะแนนสูงสุด ซึ่งก็คือ –1 จากโหนด C การเดินตานี้เราจะเลือกโหนดที่มีค่ามาก หรือเรียกว่า *ช่วงพิจารณาค่าสูงสุด*



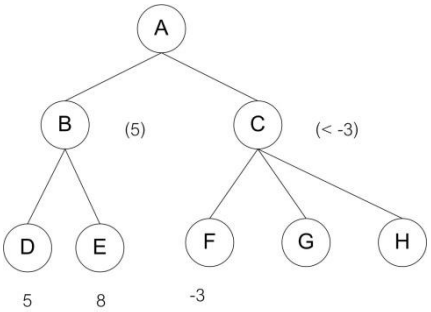
รูปที่ 2.23 เมื่อพิจารณาจากการเดินหลังจากฝ่ายตรงข้ามเดินแล้ว

จากรูปที่ 2.24 ช่วงแรกเป็นช่วงของการเลือกระหว่างโหนด B และ C การพิจารณาว่าจะเลือกโหนดใด ให้พิจารณาจากค่าสูงสุด เพราะการเลือกครั้งแรก เราจะเป็นฝ่ายเลือกคะแนนให้สูงสุด สำหรับ

ระดับถัดไป เป็นการเลือกระหว่างโหนด D, E, F, G และ H เราจะเลือกโหนดต่ำสุด เพราะว่าในระดับนี้เป็นตาของฝ่ายตรงข้าม ซึ่งคาดว่าฝ่ายตรงข้ามจะเลือกโหนดที่ดีที่สุดสำหรับเขา และทำให้เราได้คะแนนต่ำสุด การพิจารณาในระดับถัดไปเป็นตาของฝ่ายเราอีก เราจะเลือกคะแนนสูงสุดอีก และถัดไปเป็นของฝ่ายตรงข้ามอีก ก็จะเลือกคะแนนต่ำสุดอีก การเลือกจะเป็นเช่นนี้สลับกันไปเรื่อยๆ



รูปที่ 2.24 การแบ่งระดับต่างๆ ของการพิจารณา

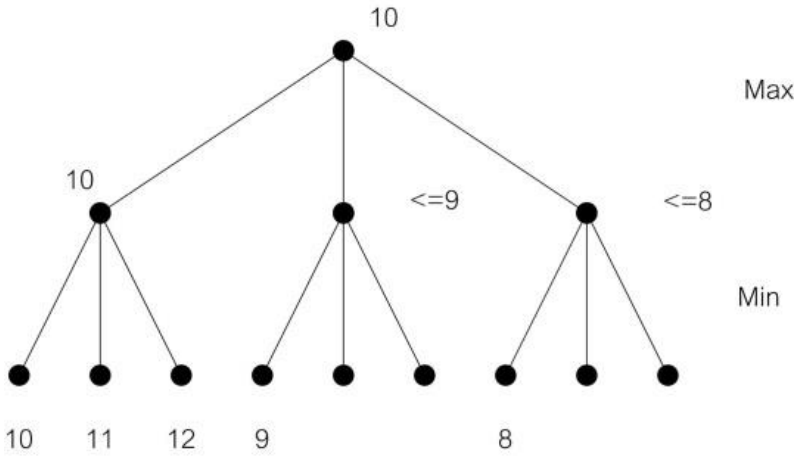


รูปที่ 2.25 การหาค่าแอลฟาและเบต้า

แอลฟา–เบต้าคัตออฟ (Alpha-Beta Cutoff) การเพิ่มความสามารถให้กับการค้นหาแบบมินแม็กซ์สามารถทำได้ด้วยการใส่ค่าสูงสุดและต่ำสุดเข้าไปเพื่อจำกัดขอบเขตของการค้นหาให้แคบลงได้อีก ลองพิจารณาดูในรูปที่ 2.25 เป็นต้นไม้ย่อยของการเล่นเกมที่มี 2 ตา พิจารณาต้นไม้ที่อยู่ใต้ B ถ้าหากว่า

เราเป็นฝ่ายเดิน จะพบว่าเมื่อเดินตาแรกเสร็จแล้ว (คือ B) ฝ่ายตรงข้ามจะเลือกระหว่าง D และ E ซึ่งเขาเลือกตัวที่มีคะแนนน้อยที่สุดสำหรับเรา ซึ่งจะได้โหนด D มีค่าเท่ากับ 5 ดังนั้นที่ A จึงประกันได้ว่าค่าต่ำที่สุดที่เราจะได้คือ 5 เมื่อเราเลือกที่จะเดินมาที่ B ในตาแรก ในทางตรงกันข้าม ถ้าเราเลือกเดินมาที่ C ค่าที่ได้ของ A จะเป็น –3 ซึ่งเป็นผลดีต่อฝ่ายตรงข้ามมากกว่า ทั้งนี้เพราะเมื่อเราสำรวจมาถึง F และได้ค่าเป็น –3 เราก็ทราบได้ทันทีว่าถ้าเลือกเดินมาที่ C จาก A นั้น ค่าที่ได้จะแย่กว่าแน่ๆ เพราะฝ่ายตรงข้ามจะทำให้เราได้คะแนนเท่ากับ –3 ดังนั้นค่าต่อไปของโหนด G และ H จึงไม่จำเป็นที่จะทราบต่อไปอีกแล้ว ไม่ว่าค่าที่ได้จะออกมาดีสำหรับเราเท่าไรก็ตาม ทำให้เราสามารถตัดการพิจารณาของ 2 โหนดนี้ (G และ H) ออกไปได้ และค่า 5 นี้เป็นการประกันคะแนนขั้นต่ำของฝ่ายเรา

กล่าวโดยสรุปคือ ถ้าเราเลือกเดินมาที่ B ผู้ที่เลือกค่ามากหรือแม็กซ์โหนดจะชนะ และเราเรียกค่า 5 นี้ว่า *ค่าเบต้า* ในทางตรงกันข้าม ถ้าเราเลือกเดินทางมาที่ C ทำให้ผู้ที่เลือกค่าต่ำหรือมินโหนดเป็นผู้ชนะ และเราเรียกค่า –3 นี้ว่า *ค่าแอลฟา*



รูปที่ 2.26 ต้นไม้ของเกมขนาด 2 ตา

ลองดูเกมแบบ 2 ตาในรูปที่ 2.26 สำหรับต้นไม้เกมนี้ แต่ละโหนดจะมีลูกอยู่ 3 ตัว หมายความว่า ในแต่ละตำแหน่งจะมีทางเลือกอยู่ 3 ทาง สำหรับการตัดสินใจเลือกตาเดิน เราสนใจที่จะคำนวณค่าของมินแม็กซ์ของต้นไม้นี้ ค่ามินแม็กซ์ของลูกตัวแรกจากตำแหน่งรากคือ 10

ตอนนี้มาดูลูกตัวที่ 2 ของโหนดราก เนื่องจากตำแหน่งรากเป็นแม็กซ์โหนด หรือเป็นการเล่นเพื่อหาค่าสูงสุด ดังนั้นลูกทุกตัวของรากนี้จะเป็นมินโหนด โดยการพิจารณาลูกเพียงตัวเดียว เราก็ทราบว่าลูกตัวที่ 2 ของรากจะต้องมีค่าน้อยกว่าหรือเท่ากับ 9 เพราะลูกตัวแรกมีค่าเท่ากับ 10 จึงทำให้เราทราบว่าลูกตัวที่ 2 จะมีค่าได้สูงสุดไม่เกิน 9 ซึ่งเราก็ไม่จำเป็นที่จะต้องไปหาค่าที่แท้จริงของมัน เพราะตัวที่มีค่ามากที่สุดของมันถูกพบแล้ว และเช่นกัน สำหรับโหนดที่ 3 เราก็ไม่มีความจำเป็นที่จะต้องไปหาค่าของมันอีก ตัวอย่างของการพิจารณาค่าง่ายๆ เหล่านี้เราเรียกว่า *แอลฟา-เบต้าพรุนนิ่ง (α-β Pruning)* สำหรับการหาค่ามินแม็กซ์ของเกมที่เล่น n-ตา เมื่อ n คือค่าคงที่

การแก้ปัญหาแบบที่กล่าวไปข้างต้น สามารถใช้กับการหาคำตอบกับการแก้ปัญหาที่มีความลึก (Depth) เท่ากับกี่ตาก็ได้ ซึ่งทำให้การแก้ปัญหาแบบนี้ลดขนาดของปริภูมิปัญหา (Problem Space) ลงได้เป็นอย่างมาก สมมติว่าถ้าทุกๆ โหนดของต้นไม้เกมนี้สามารถแตกลูกออกมาได้จำนวนเท่ากันเสมอคือ b นั่นหมายความว่า โหนดหนึ่งโหนดมีลูกอยู่ b โหนด และจำนวนตาของการคิดคำนวณคือ n (เท่ากับจำนวนระดับของต้นไม้) ดังนั้นถ้าหากว่าเราต้องการค้นหาโหนดที่ดีที่สุดด้วยวิธีมินแม็กซ์ปกติ เราจะต้องทำการหาค่าของโหนดสุดท้ายทั้งหมด b^n แต่ถ้าใช้วิธีการของแอลฟา-เบต้าคัตออฟแล้ว จำนวนการค้นหาจะลดลงมาเหลือ nb – 1 เท่านั้น เช่น ถ้ากำหนดว่าจำนวนตา n ที่ใช้ในการคำนวณทั้งหมดเท่ากับ 2 และจำนวนโหนดลูก b เท่ากับ 3 ถ้าใช้การคำนวณปกติ โหนดที่ต้องคำนวณทั้งหมดเท่ากับ $3^2 = 9$ ถ้าใช้วิธีการของแอลฟา-เบต้าคัตออฟ จะได้เท่ากับ 2*3 – 1 = 5 เท่านั้น

2.4 สรุป

การค้นหาข้อมูลบนโครงสร้างต้นไม้และกราฟสามารถจำแนกได้ 2 แบบคือ การค้นหาแบบกำหนดทิศทางและการค้นหาแบบฮิวริสติก (Heuristic Search)

การค้นหาแบบกำหนดทิศทาง เป็นการค้นหาแบบที่เริ่มต้นการสำรวจจากโหนดหนึ่งไปยังอีกโหนดหนึ่ง โดยอาศัยทิศทางเป็นตัวกำหนดการค้นหา ไม่ต้องอาศัยข้อมูลช่วยในการตัดสินใจ ตัวอย่างคือการค้นหาแบบลึกก่อน (Depth First Search) และการค้นหาแบบกว้างก่อน (Breadth First Search)

การค้นหาแบบลึกก่อน เริ่มต้นจากโหนดราก แล้วเดินลงมาให้ลึกที่สุด เมื่อถึงโหนดปลายทางให้ย้อนขึ้นมาที่กิ่งแยกต่ำสุดของกิ่งเดียวกันที่ยังไม่ได้ทำการสำรวจ แล้วทำการสำรวจโหนดในกิ่งนั้น ถ้า

พบโหนดที่ต้องการ การค้นหาก็สิ้นสุด ถ้าไม่พบให้ทำการสำรวจต่อ ทำเช่นนี้สลับกันเรื่อยไปจนพบโหนดที่ต้องการหาหรือสำรวจให้ครบทุกโหนด

การค้นหาแบบกว้างก่อน เป็นการค้นหาแบบที่ทำการสำรวจทีละระดับของโครงสร้างต้นไม้ โดยเริ่มจากโหนดราก แล้วลงมาระดับที่ 1 จากซ้ายไปขวา ระดับที่ 2 จากซ้ายไปขวาเช่นกัน ทำเช่นนี้ไปเรื่อยๆ จนพบโหนดที่ต้องการ หรือจนครบทุกโหนด

การค้นหาแบบฮิวริสติก เป็นการค้นหาคำตอบที่เหมาะสมบนฐานข้อมูลที่มีจำนวนมหาศาลจนนับไม่ถ้วน ด้วยการใช้ฮิวริสติกฟังก์ชันเป็นเครื่องมือในการกำหนดทิศทางของการค้นหา ผลลัพธ์ที่ได้จะเป็นคำตอบที่ดี แต่อาจจะไม่ได้คำตอบที่ดีที่สุด การค้นหาแบบฮิวริสติกที่สำคัญได้แก่

เจเนอเรตแอนด์เทสต์ เป็นการค้นหาด้วยการสร้างคำตอบที่เป็นไปได้ออกมา แล้วทำการตรวจสอบคำตอบนั้นเรื่อยๆ จนพบคำตอบ หรือจนไม่สามารถสร้างคำตอบที่เป็นไปได้ออกมาอีก

อัลกอริธึมกรีดี้ เป็นการค้นหาแบบดีที่สุดก่อน หลักการของการค้นหาแบบนี้คือ การเลือกโหนดที่ดีที่สุดตลอดเวลา ในการเลือกโหนดที่ดีที่สุดของกรีดี้จะอาศัยค่าที่ได้จากการวัด จากสถานะปัจจุบันไปยังสถานะเป้าหมาย

การค้นหาแบบฮิลไคลมิ่ง เป็นวิธีที่ดัดแปลงมาจากเจเนอเรตแอนด์เทสต์ ต่างกันตรงที่วิธีการของเจเนอเรตแอนด์เทสต์ให้คำตอบของการตรวจสอบคำตอบว่าใช่หรือไม่ใช่ แต่ของฮิลไคลมิ่งจะให้ค่าออกมาเป็นว่าอะไรใกล้ที่สุดเพิ่มอีกด้วย การค้นหาแบบฮิลไคลมิ่งมี 2 ชนิดคือ

ฮิลไคลมิ่งแบบธรรมดา การทำงานของการค้นหาแบบนี้มีหลักการง่ายๆ คือ ในแต่ละโหนดหรือสถานะ ให้สร้างลูกออกมา 1 ตัว แล้วตรวจสอบว่าโหนดที่สร้างขึ้นมานี้ดีกว่าโหนดปัจจุบันหรือไม่ ถ้าดีกว่า ให้โหนดที่สร้างใหม่นี้เป็นโหนดปัจจุบัน แล้วทำการตรวจสอบต่อ ถ้าไม่ดีกว่า ให้สร้างลูกตัวใหม่ขึ้นมาตรวจสอบจนกว่าจะพบลูกที่ดีกว่าสถานะปัจจุบัน ทำเช่นนี้ไปเรื่อยๆ จนพอใจ เราก็จะได้คำตอบที่ดีที่สุดเท่าที่หาได้

สตีเพสต์-แอสเซนต์ฮิลไคลมิ่ง จะต่างจากฮิลไคลมิ่งธรรมดาคือ ฮิลไคลมิ่งแบบธรรมดาจะสร้างโหนดลูกออกมาทีละตัว แล้วเลือกโหนดลูกที่ดีกว่าโหนดแม่ตัวแรกที่พบ โดยไม่สนใจลูกที่เหลือ แล้วสร้างโหนดใหม่รุ่นถัดไปเพื่อเปรียบเทียบต่อไปอีก แต่สำหรับสตีเพสต์-แอสเซนต์นั้นจะสร้างโหนดลูกทุกตัวออกมาก่อน แล้วค่อยเลือกตัวที่ดีที่สุดของลูกทั้งชุด

ปัญหาของการค้นหาแบบฮิวริสติกคือ

- *โลคอลแม็กซิมั่ม* เป็นส่วนที่บอกสถานะที่ดีที่สุดเมื่อเทียบกับสถานะข้างเคียงเท่านั้น แต่ถ้าเทียบกับสถานะอื่นๆ ที่อยู่ห่างออกไปหรือการตรวจสอบขั้นต่อไปแล้ว ไม่แน่ว่าจะได้ผลออกมาดีที่สุด
- *เพลทู* ในกรณีที่ปัญหาอยู่บนระดับเดียวกัน และผลของการหาค่าจากฮิวริสติกฟังก์ชันที่ได้คะแนนเท่ากันหมดจะไม่สามารถตัดสินใจได้ว่าจะเลือกเส้นทางใด
- *ริดจ์* เป็นกรณีที่การค้นหาได้เดินไปในทางที่ดีที่สุดตลอดเวลาเมื่อเปรียบเทียบกับโหนดข้างเคียง แต่เมื่อเทียบกับโหนดที่ดีที่สุดในระดับที่ผ่านมาจะแย่ลงเรื่อยๆ

บรานช์แอนด์เบานด์ เริ่มต้นด้วยการหาเส้นทางขึ้นมาหนึ่งเส้นทาง แล้วพยายามหาเส้นทางอื่นขึ้นมาเปรียบเทียบ ถ้าเส้นทางใหม่ที่ได้ดีกว่า ให้นำเส้นทางใหม่ที่ได้เป็นเส้นทางหลักในการเปรียบเทียบกับเส้นทางอื่นต่อไป แต่ถ้าเส้นทางใหม่ที่สร้างขึ้น เมื่อวัดแล้วแม้เพียงบางส่วนที่ยาวกว่าเส้นทางหลัก ก็ให้ยกเลิกเส้นทางนั้น ทำเช่นนี้เรื่อยๆ จนกระทั่งได้เส้นทางที่ดีที่สุด บรานช์แอนด์เบานด์มีการกำหนดกรอบค่าสูงสุดและกรอบค่าต่ำสุดเพื่อช่วยลดขนาดของปริภูมิค้นหา ซึ่งทำให้การค้นหาคำตอบแบบนี้ทำได้เร็วขึ้น

การค้นหาแบบเอ-สตาร์ เป็นอีกแบบของการค้นหาแบบดีที่สุดก่อน วิธีการเลือกโหนดที่จะใช้ในการดำเนินการต่อจะพิจารณาจากโหนดที่ดีที่สุด โดยการวัดจากสถานะปัจจุบันไปยังสถานะเริ่มต้นรวมกับค่าที่วัดจากสถานะปัจจุบันไปยังสถานะเป้าหมาย

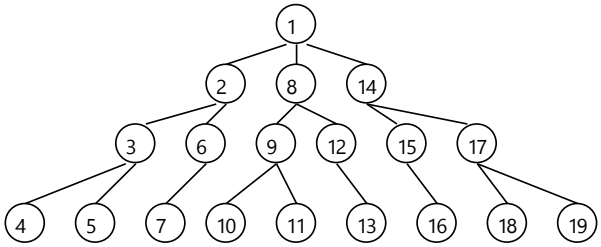
มีน-เอ็นด์อะแนไลซิส เป็นการหาคำตอบจากการหาความแตกต่างของจุดประสงค์ปัจจุบันกับจุดประสงค์ที่ต้องการ โดยอาศัยตัวดำเนินการเพื่อช่วยในการลดความแตกต่างของสถานะ ลักษณะพิเศษของมีน-เอ็นด์อะแนไลซิสก็คือ เมื่อตัวดำเนินการที่เกี่ยวข้องถูกเลือกมาเพื่อลดความแตกต่างนั้น อาจจะ

ไม่สามารถทำให้ผลลัพธ์ของสถานะปัจจุบันกลายเป็นสถานะเป้าหมายโดยทันที แทนที่จะเลิกใช้ตัวดำเนินการนี้ มีน-เอ็นด์อะแนไลซิสจะพยายามเปลี่ยนผลลัพธ์ไปเป็นจุดประสงค์อื่นแทน และทำให้จุดประสงค์นี้กลายเป็นผลลัพธ์ที่เหมาะสมสำหรับการเป็นอินพุตของตัวดำเนินการตัวต่อไป การดำเนินการจะวนเช่นนี้ไปเรื่อยๆ จนกว่าผลลัพธ์จะได้จุดประสงค์ตามเป้าหมายที่เราต้องการ

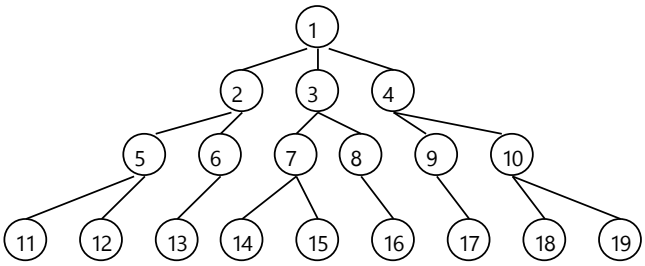
การเล่นเกม เป็นฮิวริสติกที่พิจารณาว่ามีผู้เล่นสองคนที่ต่างฝ่ายก็ต้องการชัยชนะ และต่างฝ่ายต่างก็ต้องการได้คะแนนดีที่สุด เรียกว่า *ค่ามินแม็กซ์* ซึ่งจะเป็นคะแนนที่ใช้สำหรับการเลือกตาเดิน ถ้าเราให้ตาเดินทั้งหมดของเกมเป็นปริภูมิปัญหา (Problem Space) ปริภูมิปัญหานี้จะใหญ่มาก ดังนั้นในการคิดคะแนน เรามักจะคิดจากเส้นทางที่เป็นไปได้จำนวนหนึ่งเท่านั้น และไม่จำเป็นที่จะต้องคำนวณตั้งแต่ต้นจนจบ คะแนนที่คิดคำนวณในลักษณะนี้จะเรียกว่า *การประเมินกระดานสถิต* การประเมินกระดานสถิตเป็นกระบวนการฮิวริสติกที่ใช้แสดงสถานะของหมากบนกระดานนั้นออกมาเป็นคะแนน เรียกว่า *ค่าสถิต (Static Value)* โดยการใช้ค่าแอลฟา-เบต้าคัตออฟเป็นเครื่องมือในการลดขนาดของปริภูมิปัญหา

2.5 แบบฝึกหัด

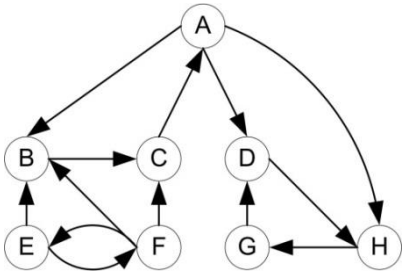
1. จากรูปโครงสร้างต้นไม้ด้านล่าง ให้เขียนตารางการค้นหาแบบลึกก่อน โดยอาศัยสแต็กในการจัดโครงสร้างข้อมูล



2. จากรูปโครงสร้างต้นไม้ด้านล่าง ให้เขียนตารางการค้นหาแบบกว้างก่อน โดยอาศัยคิวในการจัดโครงสร้างข้อมูล

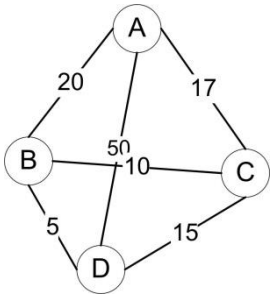


3. จากรูปกราฟที่มีการกำหนดทิศทางด้านล่าง ให้แสดงว่าการค้นหาแบบลึกก่อนเดินทางอย่างไร โดยกำหนดให้โหนด A เป็นโหนดเริ่มต้น



4. จากรูปที่มีการกำหนดทิศทางของข้อที่ 3 ให้แสดงว่าการค้นหาแบบกว้างก่อนเดินทางอย่างไร โดยกำหนดให้โหนด A เป็นโหนดเริ่มต้น

5. การค้นหาแบบฮิวริสติกแตกต่างจากการค้นหาแบบทั่วไปอย่างไร
6. เจเนอเรตแอนด์เทสต์มีหลักการอย่างไร ถ้าสมมติว่าเรามีสมการ 3x + 8 = 20 ให้อธิบายการหาค่าของ x ด้วยวิธีการอัลกอริธึมพิพิธภัณฑ์อังกฤษเพื่อหาคำตอบ
7. ให้ทดลองแก้ปัญหาดังแสดงไว้ในรูปแผนที่แสดงระยะทางระหว่างเมืองด้านล่างด้วยวิธีเจเนอเรตแอนด์เทสต์



8. ให้ใช้บรานช์แอนด์เบานด์แก้ปัญหาตามรูปแสดงแผนที่ระยะทางระหว่างเมืองของข้อ 7 ถ้าเรากำหนดให้ A เป็นจุดเริ่มต้นของการเดินทาง
9. ให้ใช้บรานช์แอนด์เบานด์แก้ปัญหาของค่าในตารางเมตริกซ์ดังต่อไปนี้

	1	2	3	4
A	4	1	4	8
B	74	10	88	82
C	62	88	8	76
D	11	74	81	21

10. ให้อธิบายจุดอ่อนของการค้นหาแบบฮิลไคลมิ่งทั้ง 3 ชนิดดังต่อไปนี้ พร้อมทั้งยกตัวอย่าง
    10.1 โลคอลแมกซิมัม
    10.2 เพลทู
    10.3 ริดจ์

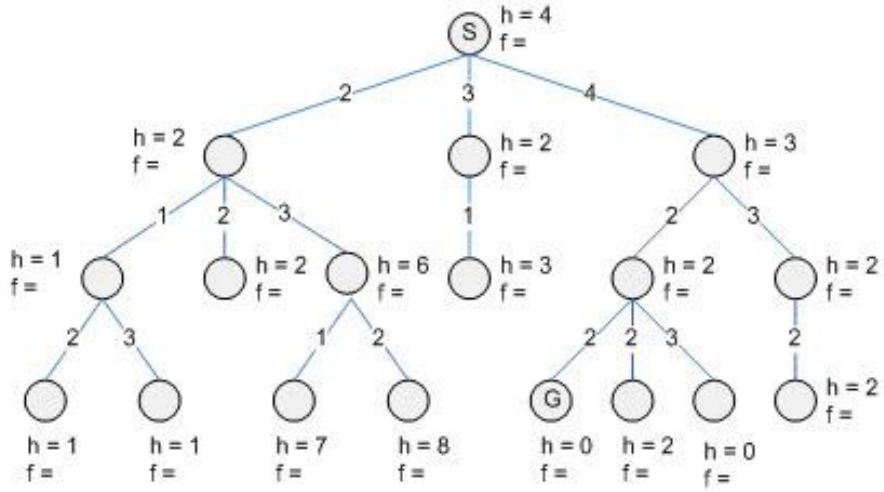
11. อธิบายถึงข้อดีของการค้นหาแบบลึกก่อนและแบบกว้างก่อน ในแง่ของขนาดของหน่วยความจำที่ต้องใช้ พร้อมยกตัวอย่างการทำงานที่เหมาะกับการค้นหาของแต่ละแบบ
12. สตีเพสต์-แอสเซนต์ดีกว่าฮิลไคลมิ่งอย่างไร และให้บอกความแตกต่างของการทำงานของอัลกอริธึมทั้ง 2 แบบนี้
13. ฮิลไคลมิ่งมีปัญหาอะไร สตีเพสต์-แอสเซนต์จะแก้ปัญหาของฮิลไคลมิ่งได้อย่างไร และแก้ได้ทั้งหมดหรือไม่
14. ให้บอกความแตกต่างและความเหมือนกันของฮิลไคลมิ่งและการค้นหาแบบกว้างก่อน
15. จากรูปการจัดวางเหรียญที่มีเหรียญ @ อยู่ทางด้านซ้ายมือ และ O อยู่ทางด้านขวามือ เหรียญทั้งสองสามารถเคลื่อนที่ได้ทิศเดียวคือ @ เคลื่อนไปทางขวา และ O เคลื่อนไปทางซ้าย ห้ามถอยหลัง การเคลื่อนของเหรียญทั้งสองมี 2 วิธีคือ เลื่อนไปยังช่องว่างที่อยู่ถัดไปได้ 1 ช่อง และสามารถกระโดดข้ามได้ 1 และ 2 ช่องไปยังช่องว่างได้ จากรูปดังแสดงข้างล่าง ให้เดินเหรียญที่เป็น O ทั้งหมดไปอยู่ทางด้านซ้ายมือของ @ โดยช่องว่างจะอยู่ที่ใดก็ได้

@	@	@		O	O	O

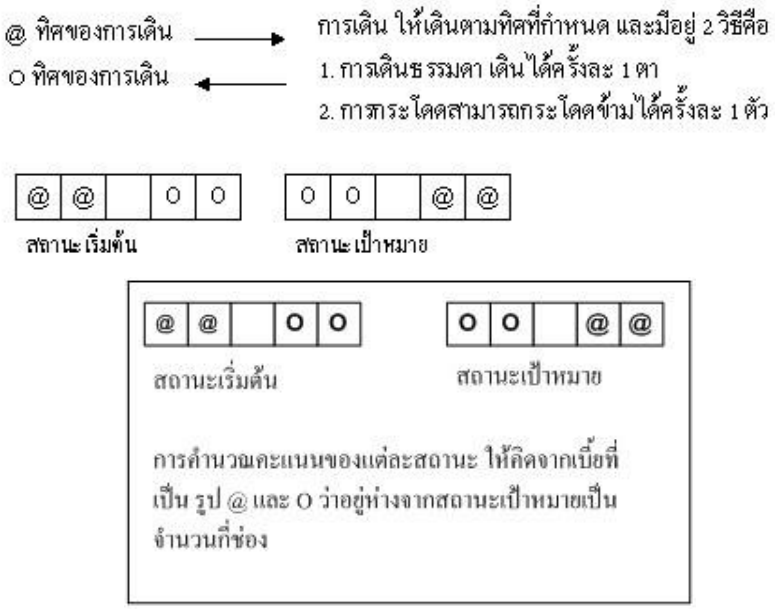
การจัดวางเหรียญ

    16.1 ให้แก้ปัญหาข้างต้นด้วยฮิลไคลมิ่ง
    16.2 ให้แก้ปัญหาข้างต้นด้วยสตีเพสต์-แอสเซนต์ฮิลไคลมิ่ง
    16.3 ให้แก้ปัญหาข้างต้นด้วยการค้นหาแบบดีที่สุดก่อน
    16.4 ให้แก้ปัญหาข้างต้นด้วยเอ-สตาร์

16. จากรูปด้านล่าง ถ้าค่าที่อยู่บนเส้นเชื่อมเป็นค่าดำเนินการสำหรับการเปลี่ยนสถานะ และค่า h คือค่าที่วัดจากสถานะปัจจุบันไปยังสถานะเป้าหมาย ให้เติมค่าของ f ที่อยู่ข้างโหนดทุกโหนด แล้วเขียนตัวเลขลำดับของโหนดที่ทำการสำรวจ



17. จากตัวอย่างในรูปต่อไปนี้ ให้แสดงการแก้ปัญหาแบบเอ-สตาร์ของปัญหาดังกล่าว



18. ให้ใช้เอ-สตาร์แก้ปัญหาดังต่อไปนี้ ถ้ากำหนดว่าสถานะเริ่มต้นคือ Start และสถานะเป้าหมาย Goal ดังแสดงในรูปข้างล่าง

Start

2	8	3
1	6	4
7		5

Goal

1	2	3
8		4
7	6	5

19. จากรูปข้างล่าง ถ้าสถานะปัจจุบันเป็น at(robot,A)^at(1,A)^on(1, 2)^armempty ให้ใช้คำสั่งและตารางแสดงความแตกต่างของมีน-เอ็นด์อะแนไลซิสดังที่กล่าวมาแล้ว ทำให้สถานะเป้าหมายเป็น at(2,B)

20. การเล่นเกมหมากรุกที่มีตารางหมากรุกขนาด 3 × 3 มีเม็ดทั้งหมด 6 ตัว เป็นขาว 3 ตัว และดำ 3 ตัว วางเรียงกันอยู่คนละฟากของกระดาน ให้เม็ดสีขาวเป็นฝ่ายที่เดินก่อนเสมอ ผู้เล่นทั้งสองฝ่ายผลัดกันเดินคนละครั้ง การเดินของเม็ดจะเดินตรงไปข้างหน้า เฉียงไปทางหน้าซ้าย หรือเฉียงไปทางหน้าขวาก็ได้ ครั้งละ 1 ช่อง การกินฝ่ายตรงข้าม สามารถทำได้ในกรณีที่เดินตาเฉียง และมีฝ่ายตรงข้ามขวางอยู่ ผู้ชนะจะต้องมีเงื่อนไขใดเงื่อนไขหนึ่งดังต่อไปนี้คือ 1) เม็ดใดเม็ดหนึ่งของฝ่ายหนึ่งสามารถเดินไปถึงปลายอีกข้างหนึ่ง 2) เม็ดของฝ่ายตรงข้ามถูกกินหมด 3) เมื่อฝ่ายตรงข้ามต้องเดินแต่ไม่สามารถเดินต่อไปได้ ให้ใช้มินแม็กซ์ร่วมกับแอลฟา-เบต้าพรุนนิ่งแก้ปัญหาดังกล่าวข้างต้น

บทที่ 3 ซีมูเลตเต็ดอันนีลลิ่ง

Simulated Annealing

3.1 คำนำ

ในการศึกษาเพื่อจำลองการทำงานของธรรมชาติมาใช้สำหรับการหาค่าเหมาะที่สุดนั้น เริ่มจากการนำหลักการของการอบอ่อน (Annealing) มาใช้ ในทางอุณหพลศาสตร์ (Thermodynamics) การอบอ่อนเป็นกระบวนการของสสารเมื่อถูกทำให้มีอุณหภูมิสูงขึ้นแล้ว การทำให้อุณหภูมิลดลง สสารจะไม่เปลี่ยนคุณสมบัติไปเป็นแก้วแทนที่การเป็นผลึกของสสาร ในการทำให้อุณหภูมิลดลงเป็นกระบวนการที่สำคัญ เพราะถ้าอุณหภูมิลดลงเร็วเกินไป สสารก็จะกลายเป็นแก้ว (Glass) ดังนั้นในการลดอุณหภูมิจึงต้องควบคุมการลดอุณหภูมิในระยะเวลาที่เหมาะสม ในปี 2496 เมโทรโปลิส และคณะ (N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. Metropolis, 1953) ได้นำเสนอวิธีการลดอุณหภูมิของกระบวนการอบอ่อน และในปี 2526 เคิร์กแพทริก และคณะ (S., Gelatt, Jr. C.D., Vecchi, M.P. Kritpatrick, 1983) ได้นำวิธีการดังกล่าวมาพัฒนาเป็นอัลกอริธึมสำหรับการหาค่าเหมาะที่สุดเรียกว่า **ซีมูเลตเต็ดอันนีลลิ่ง (Simulated Annealing)** โดยอธิบายวิธีการดังกล่าวด้วยกลศาสตร์สถิติ (Statistical Mechanics)

กลศาสตร์สถิติเป็นศาสตร์ทางฟิสิกส์ที่ว่าด้วยการกลั่นตัวของสสาร เป็นกระบวนวิธีในการบอกถึงคุณสมบัติในองค์รวมของอะตอมทั้งหมดที่มีอยู่ในสสาร ด้วยการวิเคราะห์จากตัวอย่างของอะตอมในสสารนั้น ว่าเป็นของเหลวหรือของแข็ง โดยปกติแล้ว สสารจะมีสถานะพื้นฐาน ที่บอกถึงภาวะสมดุลของอะตอมในสสารที่เรียกว่าสถานะดั้งเดิม (Ground state) โดยปกติแล้วอุณหภูมิจะเป็นเงื่อนไขในการเปลี่ยนสถานะของสสารจากสถานะดั้งเดิม ในความเป็นจริง การลดอุณหภูมิให้ต่ำลงนั้นไม่ได้เป็นเงื่อนไขเดียวสำหรับการหาสถานะพื้นฐานของสสาร ระยะเวลาของการลดอุณหภูมิก็จะมีส่วนสำคัญในการเปลี่ยนแปลงนี้ด้วย เช่นการที่คริสตัลกลับมาอยู่ในภาวะเดิมหลังจากถูกทำให้หลอมเหลว จะต้องมีการลดอุณหภูมิในระยะเวลาที่เหมาะสม

เนื่องจากจำนวนของอะตอมในสสารมีอยู่จำนวนมาก ดังนั้นการที่จะรู้ว่าอุณหภูมิของสสารนั้นอยู่ในภาวะสมดุลหรือไม่ จึงต้องพิจารณาจากความเป็นไปได้ของสสารที่จะอยู่ในภาวะสมดุลที่อุณหภูมิเฉลี่ยของอะตอมในพื้นที่ตัวอย่างของสสารนั้น ซึ่งอุณหภูมิของอะตอมในพื้นที่ตัวอย่างนี้จะเป็นส่วนหนึ่งของอุณหภูมิของสสารโดยรวม การเปลี่ยนแปลงอุณหภูมิของอะตอมในพื้นที่นี้ จะเป็นตัวสะท้อนการเปลี่ยนแปลงสถานะของสสารทั้งหมด ถ้าสถานะของสสารที่อุณหภูมิอะตอมเฉลี่ยหนึ่งๆ คือ r_i ดังนั้นความน่าจะเป็นที่สสารที่จะเข้าสู่ภาวะสมดุลจะพิจารณาจากแฟคเตอร์ของความน่าจะเป็นของโบลซ์แมนน์ (p) คือ

$$\square = \square^{\frac{-\square\{\square_{\square}\}}{\square_{\square}\square}}$$ สมการ 3.1

เมื่อ $E\{r_i\}$ เป็นพลังงานเฉลี่ยของอะตอมที่สถานะต่างๆ ของการลดอุณหภูมิหนึ่ง ค่า k_B คือค่าคงที่โบลซ์แมนน์ และ T คืออุณหภูมิ เมื่อพิจารณาในทางกลศาสตร์สถิติแล้ว p จะเป็นตัวที่บอกถึงโอกาสที่ เมื่อสสารเมื่อถูกหลอมเหลว และถูกลดอุณหภูมิลง ณ ที่อุณหภูมิต่างๆ อะตอมของสสารนั้น จะมีสถานะเป็นของเหลวหรือของแข็ง ถ้าเป็นของแข็ง สสารนั้นจะเป็นผลึกแข็ง หรือผลึกที่มีคุณสมบัติที่ผิดไปจากเดิม (หรือกลายเป็นแก้ว) ในระหว่างการลดอุณหภูมินั้น โอกาสที่สสารนั้นจะกลับกลายเป็นผลึกแข็งซึ่งเป็นคุณสมบัติดั้งเดิมนั้นมีไม่มากนัก

ในการทดลองที่พิจารณาจากสถานะของวัสดุที่อุณหภูมิต่ำ เช่นการลดอุณหภูมิของคริสตัลจากสภาวะของเหลว จะต้องทำการลดอุณหภูมิอย่างระมัดระวัง ที่เรียกว่าการอบอ่อน (Annealing) โดยเริ่มต้นด้วยการเพิ่มอุณหภูมิกับสสารให้หลอมเหลว จากนั้นก็ทำการลดอุณหภูมิอย่างช้าๆ และใช้เวลาอย่างยาวนานขึ้นที่บริเวณอุณหภูมิใกล้จุดแข็งตัว ถ้าหากสถานะของคริสตัลถูกทำให้หลุดออกจากความสมดุล การทำให้คริสตัลกลับสู่สภาพเดิม ด้วยการลดอุณหภูมิลงอย่างรวดเร็ว โดยเฉพาะในช่วงอุณหภูมิใกล้ถึงจุดแข็งตัว จะทำให้ผลที่ได้ออกมาเป็นผลึกมีความเสียหาย (หรือผลึกของคริสตัลนั้นอาจจะแข็งตัวเป็นแก้ว) ซึ่งถือว่าการลดอุณหภูมินี้ก่อให้เกิดความเสียหาย หรือกล่าวอีกนัยหนึ่งคือ การลดอุณหภูมินี้ทำให้เกิดค่าเหมาะสุดแบบโลคอล (Local optimum) ซึ่งเป็นการลดอุณหภูมิที่ไม่พึงประสงค์ สิ่งที่เราต้องการคือการลดอุณหภูมิแล้ว ไม่ได้ก่อให้เกิดการเปลี่ยนแปลงต่อผลึกคริสตัล ดังนั้นการหาสถานะของ

การลดอุณหภูมิของผลึกคริสตัล ด้วยการคำนวณค่าพลังงานให้เป็นไปตามค่าที่กำหนดโดยไม่ทำให้เกิดการเปลี่ยนสถานะ เป็นปัญหาการหาค่าเหมาะที่สุดแบบคอมบินาโทเรียล

วิธีการหาค่าเหมาะที่สุดสามารถแบ่งออกเป็น 2 วิธีคือ แบบแบ่งส่วนแล้วแก้ปัญหา (Divide and conquer) และแบบการพัฒนาทีละขั้น (Iterative improvement) แบบแบ่งส่วนแล้วแก้ปัญหาเป็นวิธีการที่เริ่มจากการแบ่งปัญหาเป็นปัญหาย่อยหลายๆ อัน จากนั้นก็แก้ปัญหาย่อยเหล่านั้นทุกอัน นำผลลัพธ์ของปัญหาย่อยทุกอันมารวมกันเป็นผลลัพธ์ของปัญหารวม การแก้ปัญหาลักษณะนี้ ปัญหาย่อยแต่ละอันจะต้องเป็นอิสระจากกัน และวิธีการแยกปัญหาใหญ่ให้เป็นปัญหาย่อยจะต้องมีความเหมาะสม ทั้งนี้เพื่อทำให้ผลลัพธ์รวมที่เกิดจากการรวมของผลลัพธ์ย่อยไม่เกิดความผิดพลาด สำหรับแบบการพัฒนาทีละขั้นนั้นจะเริ่มจากคำตอบเริ่มต้น โดยที่เราทราบแล้วว่าเป็นคำตอบนี้ แต่ยังไม่ใช่คำตอบที่เหมาะที่สุด จากนั้นก็จะใช้วิธีการแก้ปัญหาด้วยกลวิธีต่างๆ เพื่อหาคำตอบใหม่มาเปรียบเทียบกับคำตอบเดิม ถ้าคำตอบใหม่ที่ได้ดีกว่าคำตอบเก่า ก็ให้นำคำตอบใหม่ไปแทนที่ และคำตอบที่แย่กว่าก็ตัดทิ้งไป จากนั้นวนกลับไปที่วิธีการแก้ปัญหาเพื่อหาคำตอบใหม่ แล้วทำการเปรียบเทียบอีก ถ้าได้คำตอบที่ดีกว่า ก็ให้ใช้คำตอบที่ได้ใหม่แทนที่คำตอบเดิม แต่ถ้าคำตอบใหม่แย่กว่าให้ใช้คำตอบเดิมต่อไป ทำเช่นนี้เรื่อยๆ จนกว่าจะได้คำตอบที่คิดว่าเหมาะสมที่สุด

ในการแก้ปัญหาแบบการพัฒนาทีละขั้น มีกระบวนการที่เหมือนกับวิธีการของกลศาสตร์สถิติ เพราะวิธีการของกลศาสตร์สถิติจะอาศัยการคำนวณความน่าจะเป็น p ที่บอกความเป็นไปได้ของการเปลี่ยนสถานะของสสารจากสมการ 3.1 สำหรับฮิวริสติกฟังก์ชันของการค้นหาคำตอบแบบพัฒนาทีละขั้น จะทำหน้าที่เหมือนกับการคำนวณค่า p จากพลังงาน ในการเลือกคำตอบที่เป็นไปได้ในแต่ละขั้น ก็เหมือนกับการหาค่าของอุณหภูมิที่ลดลงของการอบอ่อน และการเลือกคำตอบที่เหมาะสมที่สุด ก็เสมือนหนึ่งการลดอุณหภูมิมาจนถึงจุดแข็งตัวของสสาร นอกจากนั้นยังมีสิ่งที่เหมือนกันอย่างหนึ่งก็คือ ขั้นตอนของการลดอุณหภูมิกับการเลือกคำตอบที่เหมาะที่สุดในแต่ละขั้น หรือรอบ (Iteration) ของการค้นหา

ในปี 2496 เมโทรโปลิส (N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. Metropolis, 1953) ได้เสนออัลกอริธึมในการลดอุณหภูมิของกลศาสตร์สถิติ ที่เป็นรูปแบบทั่วไปสำหรับการพัฒนาทีละขั้น (Iterative improvement) เพื่อการควบคุมการลดอุณหภูมิของอะตอม

ในการค้นหาคำตอบที่ดีขึ้น เมโทรโปลิสได้นำเสนออัลกอริธึมที่สามารถจำลองแบบวิธีการที่อะตอมจะเข้าสู่ภาวะสมดุลที่อุณหภูมิต่างๆ โดยที่กำหนดว่า ในแต่ละขั้นตอนของอัลกอริธึม อะตอมตัวหนึ่งๆ จะสามารถมีสถานะที่แตกต่างออกไปเล็กน้อยโดยการสุ่ม เป็นผลให้พลังงานเปลี่ยนไปเท่ากับ ΔE ถ้าหากว่า ΔE ≤ 0 การเปลี่ยนแปลงนั้นก็จะได้รับการยอมรับ และสถานะของมันจะกลายเป็นสถานะใหม่ของอะตอม แล้วเริ่มทำงานในรอบต่อไป ด้วยการสุ่มสถานะใหม่ของอะตอมที่แตกต่างออกไป ถ้า ΔE > 0 การพิจารณาว่าจะยอมรับสถานะนี้เป็นสถานะใหม่หรือไม่ ให้พิจารณาจากความน่าจะเป็นจากสมการดังนี้

$$\square(\Delta\square) = \square^{\frac{-\Delta\square}{\square\square\square}}$$ สมการ 3.2

จากนั้น สร้างตัวเลขสุ่มของค่า (0,1) มาหนึ่งค่า แล้วทำการเปรียบเทียบกับค่าของ p(ΔE) ถ้าค่าสุ่มที่ได้น้อยกว่าค่าของ p(ΔE) เราจะกำหนดให้สถานะใหม่เป็นสถานะปัจจุบันของอะตอมแทนที่สถานะเดิม แต่ถ้าค่าสุ่มที่ได้มามากกว่าค่าของ p(ΔE) ให้ใช้สถานะเดิมเป็นสถานะปัจจุบัน ด้วยการทำซ้ำกระบวนการดังกล่าวหลายๆ รอบของการเปลี่ยนค่าของสถานะอะตอม ตามการเปลี่ยนค่า T ของสมการ 3.2 ข้างต้น

อัลกอริธึมของเมโทรโปลิส เป็นการสร้างคำตอบที่เป็นไปได้ที่เป็นทางเลือกให้กับการค้นหาคำตอบของการหาค่าเหมาะที่สุด การที่อะตอมแต่ละตัวสามารถสร้างสถานะที่แตกต่างเป็น p(ΔE) จากการปรับค่าของ T นี้ ที่เป็นเสมือนการสร้างคำตอบที่เป็นทางเลือกใหม่ที่เป็นไปได้ ของการหาค่าเหมาะที่สุดแบบพัฒนาทีละขั้น จากคำตอบที่เป็นไปได้นี้ ค่าของ p(ΔE) จะเป็นเสมือนฮิวริสติกฟังก์ชันที่ใช้ในการตัดสินว่าจะเลือกหรือไม่เลือกคำตอบที่เป็นไปได้อันใหม่ที่สร้างขึ้น

3.2 การหาค่าเหมาะที่สุดด้วยซีมูเลตเต็ดอันนีลลิ่ง

ดังที่ได้กล่าวมาแล้วว่า การค้นหาแบบฮิลไคลมิ่งอาจจะเกิดปัญหาของโลคอลแม็กซิมั่มได้ง่าย เพราะการเลือกคำตอบที่เหมาะสมในแต่ละระดับ โหนดลูกที่ดีที่สุดจะถูกเลือกเพียงโหนดเดียว แล้วสร้างโหนดลูกชุดใหม่ออกมาอีกจากโหนดที่เลือกไว้ในระดับแรก ปัญหาก็เกิดตรงที่โหนดที่ดีที่สุดในแต่ละ

ระดับก็ไม่รับประกันว่าจะสร้างโหนดลูกที่ดีที่สุดของระดับถัดไปได้เสมอ ดังนั้นการเลือกโหนดที่ดีที่สุดเพียงโหนดเดียวในแต่ละระดับ จึงเป็นการเสี่ยงที่จะให้ได้คำตอบที่ไม่ดีได้ การเลือกโหนดที่มีค่ารองลงมาเพื่อเป็นการสำรอง แล้วสร้างโหนดลูกมาเพื่อการพิจารณาเปรียบเทียบเพิ่มเติมจึงเป็นเรื่องที่ดี แต่ปัญหาที่ตามมาก็คือว่า หลักในการพิจารณาว่าโหนดที่มีค่ารองลงมานั้น จะต้องมีความแตกต่างจากโหนดที่ดีที่สุดสักเท่าไร ถ้าโหนดที่รองลงมามีค่าฟิตเน็สที่คำนวณได้ ต่างจากโหนดที่ดีที่สุดมาก การเลือกโหนดที่รองลงมาอาจจะไม่มีความจำเป็น แต่ถ้าโหนดที่ดีที่สุดและโหนดที่รองลงมามีค่าฟิตเน็สต่างกันน้อยมาก การเลือกโหนดที่รองลงมาจะมีความสำคัญมาก ในการตัดสินใจว่า ค่าฟิตเน็สนั้นมีความแตกต่างขนาดไหนที่เรียกว่าน้อย และขนาดใหนที่เรียกว่ามาก จึงเป็นเรื่องที่ต้องมีวิธีการช่วยในการตัดสินใจ และการตัดสินใจนี้จะต้องอาศัยฟังก์ชันในการคำนวณมากำหนด

เพื่อที่จะแก้ปัญหาข้างต้น เคิร์กแพทริก (S., Gelatt, Jr. C.D., Vecchi, M.P. Kritpatrick, 1983) ได้นำเสนอวิธีการค้นหาแบบใหม่คือ ซีมูเลตเต็ดอันนีลลิ่ง (Simulated Annealing) ที่ใช้วิธีการทางฟิสิกส์มาช่วยในการแก้ปัญหา โดยอาศัยหลักการลดอุณหภูมิของโลหะ เมื่อมีความร้อนสูงมากให้ลงมาอยู่ที่อุณหภูมิปกติ โดยพิจารณาว่าการลดหรือความแตกต่างของอุณหภูมิที่ลดลงนี้จะไม่ทำให้เกิดการเปลี่ยนแปลงคุณสมบัติของโลหะเมื่อโลหะอบอ่อนตัว (Annealing) การทำให้โลหะนั้นเย็นลงอย่างช้าๆ จนถึงอุณหภูมิที่โลหะแข็งตัว เป็นกระบวนการในการลดพลังงานที่สะสมในโลหะ กระบวนการลดลงของพลังงานนี้มีความน่าจะเป็นที่ไม่นำไปสู่ความเสียหายสามารถคำนวณได้จาก

$$\square = \square^{\frac{-\Delta\square}{\square\square}}$$

เมื่อ p คือค่าความน่าจะเป็นที่จะไม่เกิดโลคอลแม็กซิมั่ม

ΔE คือค่าความเปลี่ยนแปลงของพลังงานที่เป็นบวกของระดับพลังงาน

K คือค่าคงที่ของโบลซ์แมนน์ (Boltmann Constant) ของโลหะ ในอัลกอริธึมนี้กำหนดให้ K เท่ากับ 1

T คืออุณหภูมิ

ในการนำเรื่องนี้มาใช้กับการค้นหาแบบฮิวริสติก เคิร์กแพทริกได้ปรับสูตรการคำนวณใหม่ โดยการตัดค่า K ออก ซึ่งในการทำงานจริงค่านี้จะเท่ากับ 1

$$□ = □^{\frac{-Δ□}{□}}$$ สมการ 3.3

สำหรับการคำนวณจริง มีเรื่องที่จะต้องทำความเข้าใจเพิ่มเติมเกี่ยวกับค่าของ T คือ เมื่อค่าของ T ลดลงความน่าจะเป็นของการเกิด P ก็น้อยลงด้วย สำหรับการลดค่าของ T นี้จะมีความหมายต่อสถานภาพของโลหะด้วย คือถ้าอุณหภูมิลดลงเร็วเกินไป โอกาสที่จะเกิดความเสียหายกับวัตถุ (เกิดจากโลคอลแม็กซิมั่ม) จะมีมาก ในทางตรงกันข้าม ถ้าเราลดค่าของ T น้อยเกินไป ก็จะเป็นการเสียเวลา ดังนั้นอัตราการลดของ T จึงมีความสำคัญ ซึ่งอัตราการลดของ T นี้ ในทางฟิสิกส์จะเรียกว่า *ตารางอบอ่อน (Annealing Schedule)* เหตุการณ์นี้เกิดขึ้นเช่นเดียวกับการค้นหาข้อมูลเหมือนกันคือ ในช่วงแรกของการค้นหา จะเปรียบเสมือนช่วงที่มีพลังงานสูงอยู่และอุณหภูมิก็ยังสูงอยู่โอกาสของการเกิดโกลบอลแม็กซิมั่มจะมีไม่มาก เพราะการสร้างปริภูมิปัญหา (Problem Space) อยู่ในช่วงเริ่มต้น และปริภูมิปัญหายังมีขนาดใหญ่ เมื่อการค้นหาดำเนินไปถึงช่วงปลายหรือช่วงใกล้ถึงเป้าหมาย ขนาดของปริภูมิปัญหาจะเล็กลง โอกาสของการเกิดโกลบอลแม็กซิมั่มจะมีมาก ดังนั้นในการกำหนดขั้นตอนการค้นหาและการกำหนดอัตราการลดลงของ T จึงเป็นเรื่องที่ต้องคำนึงถึง ซึ่งในการค้นหาข้อมูลยังไม่มีการกำหนดอัตราการลดของ T ที่ตายตัวออกมา ในทางปฏิบัติ การกำหนดอัตราการลดค่า T นี้ ให้ใช้วิธีการทดลองว่าการลดแบบใดจะให้ผลดีที่สุด แล้วเอาค่านั้นมาเป็นเกณฑ์ สำหรับค่า E คือค่าคะแนนของโหนดที่คำนวณได้จากค่าของฮิวริสติกฟังก์ชัน และ ΔE คือความแตกต่างระหว่างค่าของโหนดในสถานะปัจจุบันกับค่าของโหนดในสถานะใหม่sinvFsof-hk’g8up’

ด้วยวิธีการจำลองกระบวนการทางฟิสิกส์ ในแต่ละขั้นของซีมูเลตเต็ดอันนีลลิ่งจะเป็นการแทนกระบวนการในการหาคำตอบใหม่ที่ดีกว่าเพื่อใช้แทนคำตอบเดิม ด้วยการสร้างคำตอบใหม่ที่ได้จากการสุ่มขึ้นมา ในการสุ่มหาคำตอบใหม่นี้จะสุ่มจากคำตอบข้างเคียงของคำตอบปัจจุบัน คำตอบใหม่ที่ได้จะได้รับการยอมรับให้เป็นคำตอบใหม่ ด้วยค่าของความน่าจะเป็นที่พิจารณาจากค่าความต่างของพลังงานที่คำนวณได้และค่าเป้าหมายของอุณหภูมิ T ซึ่งอุณหภูมินี้จะมีค่าลดลงเรื่อยๆ ในระหว่างการดำเนินการของกระบวนการอบอ่อน ในช่วงแรกของการอบอ่อนค่า T ยังมีค่ามากอยู่ การเลือกคำตอบที่เป็นไปได้ระหว่างคำตอบเดิมและคำตอบที่หามาได้ใหม่ เกือบจะเป็นการสุ่ม แต่การเลือกจะดีขึ้นเรื่อยๆ เมื่อค่าของ T เข้าสู่ 0 ซึ่งกระบวนดังกล่าวถือว่าเป็นข้อดีที่ในช่วงแรกของการค้นหาคำตอบ คำตอบที่ได้จะไม่ติดกับค่าของโลคอลมินิมั่ม

3.2.1 อัลกอริธึมซีมูเลตเต็ดอันนีลลิ่ง

อัลกอริธึมที่จำลองการทำงานของซีมูเลตเต็ดอัลนีลลิง มีขั้นตอนการทำงานที่มีรายละเอียดดังนี้

อัลกอริธึม ซีมูเลตเต็ดอันนีลลิ่ง (Simulated Annealing) : BEST เป็นดัชนีที่ชี้ไปยังสถานะปัจจุบัน SUCC เป็นหน่วยความจำชั่วคราวที่ใช้เก็บสถานะ (โหนด) ที่ผ่านการคัดเลือกจากสถานะที่เพิ่งสร้างขึ้นมาใหม่ โดยการพิจารณาจากค่า p

1. ตรวจสอบสถานะเริ่มต้น ถ้าสถานะเริ่มต้นนี้คือสถานะเป้าหมาย ให้แสดงคำตอบ และยุติการทำงาน แต่ถ้าสถานะเริ่มต้นไม่ได้เป็นสถานะเป้าหมายด้วย เปลี่ยนสถานะนี้เป็นสถานะปัจจุบัน (Current State) และทำต่อข้อที่ 2
2. กำหนดให้ BEST เป็นดัชนี (Pointer) ใน SUCC และให้ดัชนีนี้ชี้มาที่สถานะปัจจุบัน
3. กำหนดตารางอันนีลลิ่ง (Annealing Schedule) ของ T
4. ให้ทำตามกระบวนการข้างล่างนี้จนกว่าจะพบคำตอบ หรือจนกระทั่งไม่มีโหนดใน SUCC
   4.1. สร้างสถานะใหม่จากสถานะปัจจุบัน โดยพิจารณาจาก SUCC
   4.2. ตรวจสอบสถานะใหม่ที่ได้ดังนี้
      4.2.1. ถ้าสถานะใหม่ที่ได้คือเป้าหมายให้ return ค่าที่ได้ออกมา และเลิกการทำงาน
      4.2.2. ถ้าไม่ใช่สถานะเป้าหมาย แต่ค่าที่ได้ดีกว่าค่าของสถานะปัจจุบัน ให้เก็บค่าที่ได้นี้ใน SUCC และกำหนดให้เป็นสถานะปัจจุบัน และให้ BEST ชี้ที่สถานะปัจจุบันนี้ด้วย
      4.2.3. ถ้าสถานะใหม่ที่ได้ไม่ใช่เป้าหมาย และมีค่าแย่กว่าสถานะปัจจุบัน ให้คำนวณค่า ΔE = ค่าของสถานะปัจจุบัน – ค่าของสถานะใหม่ แล้วหาค่า p ออกมา ขณะเดียวกันให้สุ่มตัวเลข 0-1 ออกมา 1 จำนวน ให้เป็นค่าของ p’ แล้วนำ p มาเปรียบเทียบกับค่า p’ ถ้าค่าของ p มากกว่า กำหนดให้สถานะใหม่นี้เป็นสถานะปัจจุบัน นำเข้าไปเก็บใน SUCC
   4.3. เปลี่ยนแปลงค่า T ตามตารางอันนีลลิ่ง
5. ให้ค่า return ของ BEST ออกมา

3.2.2 ขั้นตอนการวนรอบ

ในแต่ละขั้นของการทำงาน กระบวนการซีมูเลตเต็ดอันนีลลิ่งจะพิจารณาคำตอบข้างเคียง s' ของคำตอบปัจจุบัน s ด้วยค่าความน่าจะเป็นว่าสถานะขั้นต่อไปของอะตอมจะเป็น s' หรือ s ซึ่งการเลือกนี้จะมีผลต่อไปกับระบบในการเลือกคำตอบที่เป็นไปได้ในช่วงที่ค่าของพลังงานลดลง กระบวนการนี้จะถูกทำซ้ำจนกว่าจะได้คำตอบที่ดี หรือจนกว่าจะยุติการทำงานของคอมพิวเตอร์

3.2.3 คำตอบข้างเคียง

คำตอบข้างเคียงเป็นสถานะใหม่ของปัญหา ที่ถูกสร้างขึ้นมาด้วยการปรับเปลี่ยนสถานะปัจจุบัน หรือคำตอบปัจจุบันที่ผ่านวิธีการบางอย่าง เช่น ในการแก้ปัญหาการเดินทางของเซลส์แมน การหาคำตอบข้างเคียง อาจจะหมายถึงการนำเส้นทางเดินที่เดินผ่านเมืองต่างๆ ไปแล้ว และได้รับเลือกให้เป็นคำตอบในปัจจุบัน มาจัดเรียงเส้นทางของการเดินผ่านเมืองต่างๆ ดังกล่าวใหม่ ให้เกิดเส้นทางเส้นเดินใหม่ ในการหาคำตอบข้างเคียงดังกล่าว เป็นวิธีการที่จะช่วยให้กระบวนการในการหาคำตอบมีทางเลือกมากขึ้น และยังเป็นการป้องกันการเกินโลคอลแม็กซิมั่ม

การค้นหาด้วยการหาคำตอบข้างเคียง เป็นวิธีการที่เป็นมาตรฐานในการค้นหาคำตอบที่ดี เมื่อการค้นหาคำตอบในรอบหนึ่งผ่านไป การค้นหาคำตอบที่ดีที่สุดด้วยการเปรียบเทียบกับคำตอบข้างเคียงก็จะเกิดขึ้น กระบวนการนี้จะถูกทำซ้ำไปเรื่อยๆ จะกว่าจะได้คำตอบที่ต้องการ แต่กระบวนการนี้ก็มีจุดอ่อนเช่นกัน เมื่อการค้นหาดำเนินการไปเรื่อยๆ และถ้าสมมุติว่าเราไม่สามารถหาคำตอบข้างเคียงที่ดีกว่าคำตอบในปัจจุบันได้แล้ว ก็ไม่ได้มีหลักประกันว่า ในปริภูมิของคำตอบนี้ไม่มีคำตอบที่ดีกว่านี้อีกแล้ว และนี่จึงเป็นเหตุผลของการเกิดโลคอลแม็กซิมั่ม ดังนั้นในหลายๆ กรณี การสร้างคำตอบข้างเคียงขึ้นมานั้น ไม่เพียงแค่ใช้ในการเปรียบเทียบกับคำตอบปัจจุบันเท่านั้น แต่ยังใช้สำหรับเป็นช่องทางในการสำรวจปริภูมิคำตอบใหม่ๆ ด้วย ในการทำสำรวจปริภูมิคำตอบ อัลกอริธึมจะต้องยอมรับคำตอบที่แย่กว่าคำตอบปัจจุบันด้วย เพื่อเป็นหลักประกันว่า ถ้าเราสามารถทำให้คอมพิวเตอร์ทำการค้นหาได้อย่างที่เราพอใจแล้ว เราจะได้คำตอบที่ดีที่สุดหรือโกลบอลมินิมั่มแน่นอน

3.3 การปรับสมรรถนะ

ในซีมูเลตเต็ดอันนีลลิ่ง การปรับสมรรถนะจะขึ้นอยู่กับพารามิเตอร์ พารามิเตอร์ที่สำคัญของอัลกอริธึมก็คือตารางการอบอ่อน ซึ่งเป็นเทคนิคในการลดอุณหภูมิ การปรับพารามิเตอร์นี้มีผลต่อสมรรถนะในการทำงานของอัลกอริธึมเป็นอย่างมาก ทั้งในแง่ของการลู่เข้าสู่คำตอบที่ดีและคุณภาพของคำตอบที่ได้

3.3.1 ตารางการอบอ่อน

ตารางการอบอ่อน (Annealing schedule) เป็นกระบวนการลดอุณหภูมิ (T) เพื่อทำให้ระบบเข้าสู่ภาวะสมดุล ดังนั้นการเปลี่ยนแปลงของอุณหภูมิ T เป็นเงื่อนไขสำคัญของการอบอ่อน ในการค้นหาคำตอบที่ดีก็เช่นกัน การลดอุณหภูมิลงก็จะเป็นหัวใจของการค้นหา การนำการเปลี่ยนแปลงของอุณหภูมิเข้ามาใส่ในอัลกอริธึมจึงเป็นเรื่องสำคัญ และมีผลต่อวิธีการของการค้นหาคำตอบด้วย ในการกำหนดตารางของการอบอ่อนนั้น ให้เริ่มต้นเมื่อ T มีค่าสูง จากนั้นก็ลดอุณหภูมิลงมาเรื่อยๆ ซึ่งในช่วงของการลดอุณหภูมินั้น เราจะต้องกำหนดขึ้นมาเอง การลดอุณหภูมิจะดำเนินการไปจนกระทั่งสิ้นสุดการค้นหาคำตอบ และค่าของ T ต่ำสุดจะต้องเป็น 0 ด้วยวิธีการดังกล่าว อัลกอริธึมจะมีขอบเขตของปริภูมิค้นหาที่กว้างขวางขึ้นในช่วงต้นของการค้นหา ซึ่งทำให้มีโอกาสที่จะเลือกคำตอบที่ดีได้มากขึ้น จากนั้นเมื่ออุณหภูมิลดลง ขอบเขตของการค้นหาก็เล็กลง ทำให้การค้นหามุ่งเจาะจงลงไปที่การค้นหาคำตอบที่ดีขึ้นเรื่อยๆ จนได้คำตอบที่เหมาะสม

นูรานีและแอนเดอร์สัน (Y. and Andersen, B. Nourani, 1998) ได้รวบรวมวิธีการของการจัดตารางการอบอ่อนไว้ในการศึกษาเปรียบเทียบประสิทธิภาพของแต่ละตาราง ตารางที่ถูกนำเสนอก่อนในช่วงแรกได้แก่ การกำหนดวิธีการลดอุณหภูมิแบบคงที่ โดยที่ค่าคงที่นี้เมื่อนำไปคูณกับอุณหภูมิเดิมการเปลี่ยนแปลงก็จะเป็นแบบ เอ็กโพเนนเชียล (Exponential) ซึ่งมีสมการดังนี้

$$□(□) = □□_0$$ สมการ 3.4

แต่ถ้านำไปลบกับอุณหภูมิเดิมก็จะเป็นแบบเส้นตรง (Linear)

$$□(□) = □_0 − □□$$ สมการ 3.5

การลดอุณหภูมิของทั้งสองแบบนี้ถูกนำเสนอโดยเคิร์กแพทริกและคณะ โดยมีค่า T(t) เป็นค่าการลดของอุณหภูมิตามตาราง ค่า α และ η เป็นค่าคงที่ และ t เป็นตัวนับขั้นตอนการลดอุณหภูมิ ต่อมาจีแมน (S. and Geman, D. Geman, 1984) ได้นำเสนอวิธีการลดอุณหภูมิใหม่แบบลอกาลิธึม ดังสมการ 3.6 โดยที่ค่า c และ d เป็นค่าคงที่ และ d ถูกกำหนดให้เท่ากับ 1 แต่อย่างไรก็ตาม สมการดังกล่าวไม่ได้รับการยอมรับมากนักเนื่องจากตัวสมการทำให้ T นั้นลดอุณหภูมิช้ามาก และทำให้หลักการดั้งเดิมของโบลซ์แมนน์ถูกละเลย

$$\square(\square) = \frac{\square}{\log(\square+\square)}$$ สมการ 3.6

ในอีกไม่นานต่อมา ซูและฮาร์ทเลย์ (H. and Hartley, R. Szu, 1987) ได้นำเสนอวิธีการที่เรียกว่าการอบอ่อนอย่างรวดเร็ว โดยการใช้หลักการกระจายของโคชี (Cauchy distribution) โดยการนำเสนอตารางการลดอุณหภูมิใหม่ตามสมการ 3.7

$$\square(\square) = \frac{\square_0}{\square}$$ สมการ 3.7

ในปี 2533 ฮอฟ์ฟแมนน์ และสารามอน (Hoffmann, 1990) ได้นำเสนอวิธีการลดอุณหภูมิของ T ในรูปแบบของลอการิธึมดังสมการ 3.8

$$\square(\square) \approx \frac{\square-1}{\square\square\square}$$ สมการ 3.8

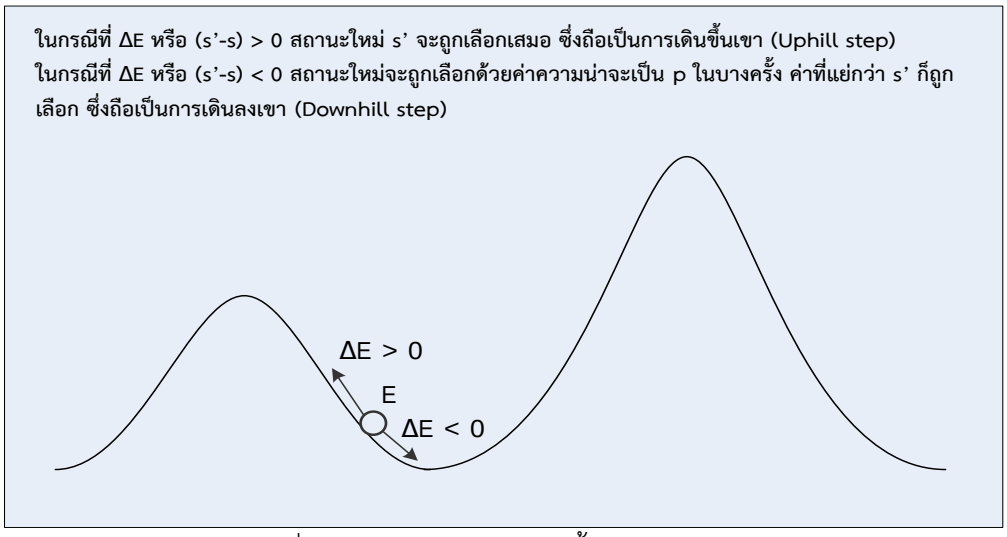
โดยที่ D เป็นค่าคงที่ ในการศึกษาเรื่องตารางการอบอ่อนนั้นยังมีอีกมาก เช่น วิธีการของนัลตันและสารามอน (J. D. and Salamon, P. Nulton, 1988) และของ แอนเดอร์เซนและกอร์ดอน (B. and Gordon, J. M. Andresen, 1994) เป็นต้น

3.3.2 การลู่เข้าสู่คำตอบ

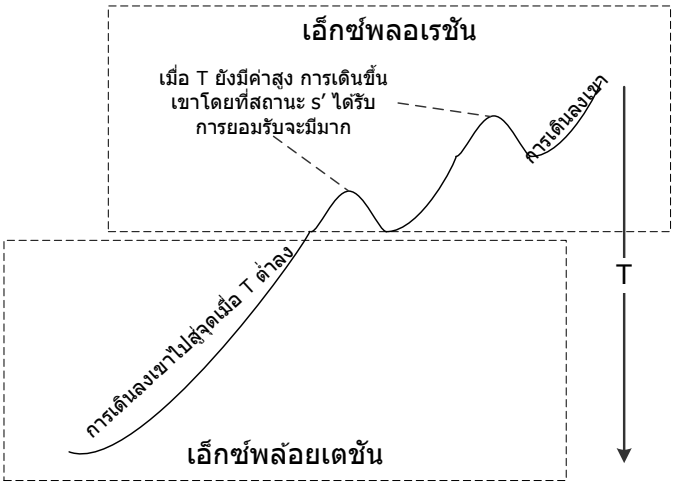
จากอัลกอริธึมของเมโทรโปลิสที่แสดงไว้ในสมการที่ 3 เป็นการหาค่าของความน่าจะเป็น p ที่จะยอมรับสถานะข้างเคียง s’ ที่สร้างขึ้นใหม่ ในกรณีที่ s’ < s (สำหรับกรณีที่ s’ > s สถานะ s’ จะถูกเลือกอย่างแน่นอน) ความหมายก็คือ ระบบจะยอมรับสถานะที่แย่กว่า (s’) เป็นสถานะปัจจุบัน คือ การเดินย้อนไปในจุดที่แย่กว่าหรือการเดินลงเขา (Downhill step) ในกรณีที่เราเลือกค่าที่ดีกว่าเสมอ ไม่ว่าจะเป็นในกรณีที่ s’ > s แล้วเลือก s’ จะเรียกว่าการเดินขึ้นเขา (Uphill step) ดังแสดงในรูปที่ 3.1 ที่แสดงการเลือกเดินขึ้นหรือลงเขาตามการพิจารณาของ p

จากสมการ 3.3 ค่าของ p นั้นจะมีความสัมพันธ์โดยตรงกับค่าของ ΔE และค่า T ถ้าเราดูความสัมพันธ์ระหว่าง p ความน่าจะเป็น (ของการไม่เกิดโลคอลแม็กซิมั่ม) กับ ΔE (ความแตกต่างของพลังงาน) ก่อน ความสัมพันธ์นี้เป็นแบบผกผัน คือเมื่อ ΔE มีค่ามากขึ้น ค่าของ p จะมีค่าน้อยลง ความหมายก็คือว่า ถ้าพลังงานของสถานะ s ที่เป็นสถานะปัจจุบัน กับสถานะ s' ซึ่งเป็นสถานะข้างเคียงมีความแตกต่างกันมาก ค่าของ p จะมีค่าน้อย โอกาสที่จะยอมรับสถานะของ s' ก็จะมีน้อยลงด้วย ในทางตรงกันข้าม ถ้าความแตกต่างของพลังงานในสถานะ s และ s' มีน้อย โอกาสที่ s' จะได้รับการยอมรับก็มีมากขึ้น โอกาสของการเดินลงเขาจะมีมากขึ้น

เมื่อพิจารณาความสัมพันธ์ระหว่าง p และ T ความสัมพันธ์จะเป็นไปในลักษณะที่ตามกัน คือ เมื่อ T มีค่ามาก ค่าของ p มีค่ามากตามไปด้วย ตัวอย่างเช่น เมื่อค่า T □ ∞ ค่าของ p จะเป็น 1 ซึ่งหมายถึง s' หรือสถานะใหม่ที่อยู่ข้างเคียงจะได้รับการยอมรับเสมอ และเมื่อค่าของ T □ 0 ค่าของ p จะเป็น 0 ค่าของ s' จะถูกปฏิเสธเสมอ ดังนั้นผลของค่า T จึงมีอิทธิพลเป็นอย่างมากต่อการค้นหาคำตอบ



รูปที่ 3.1 กราฟแสดงการเดินขึ้นเขา และการเดินลงเขา



รูปที่ 3.2 แสดงการลู่เข้าสู่คำตอบเมื่อค่า T เปลี่ยน

ดังที่ได้กล่าวมาแล้วว่า ในช่วงแรกของการค้นหาคำตอบ เราจะกำหนดค่า T ให้มีค่าสูง เพื่อทำให้การเลือกคำตอบใหม่ๆ เป็นไปได้อย่างกว้างขวาง เปรียบเสมือนการเดินสุ่มสำรวจพื้นที่เพื่อหาสถานะใหม่ๆ มาทำการค้นหาคำตอบ ดังนั้นการค้นหาคำตอบจะมีการเดินลงเขาอยู่เสมอ เพราะค่า p ที่ได้มีค่าสูง และมีโอกาสที่ s' จะได้รับการยอมรับสูง ในทางเมตาฮิวริสติกแล้ว กระบวนการเช่นนี้ถือเป็นเอ็กซ์พลอเรชัน ดังนั้นในช่วงแรกของการค้นหา โอกาสที่ s' จะถูกเลือกจะมีสูงมาก เมื่อเราลดอุณหภูมิลงเรื่อยๆ ค่า T จะน้อยลงจนเหลือ 0 ทำให้โอกาสที่ s' จะได้รับการยอมรับน้อยมาก เพราะค่า p มีค่าเป็น 0 ดังนั้นในการสำรวจจึงมุ่งแต่การเดินขึ้นเขาเสมอ ด้วยการเลือก s และ s' ที่ดีกว่าเสมอ ในทางเมตาฮิวริสติก กระบวนการในช่วงนี้จะเรียกว่า เอ็กซ์พล้อยเตชัน ดังแสดงในรูปที่ 3.2 การลู่เข้าสู่คำตอบของการค้นหาแบบซีมูเลตเต็ด อันนีลลิ่ง

จากรูปที่ 3.2 แสดงการลู่เข้าสู่คำตอบเมื่อค่า T เปลี่ยน ถ้าให้ s' คือคำตอบข้างเคียงของ s ที่เป็นคำตอบที่เป็นไปได้ หรือคำตอบที่ดีที่สุดที่อุณหภูมิต่างๆ ดังที่ได้กล่าวมาแล้ว เมื่ออุณหภูมิ T ยังมีค่าสูงอยู่ โอกาสที่ s' คือคำตอบข้างเคียง หรือคำตอบที่แย่กว่าถูกเลือกจะมีมาก ซึ่งเป็นช่วงที่อัลกอริธึมมีโอกาสในการสำรวจคำตอบที่กว้าง การค้นหาคำตอบในช่วงนี้จะเรียกว่า เอ็กซ์พลอเรชัน (Exploration) เมื่อค่า T ลดลง โอกาสที่คำตอบข้างเคียง หรือคำตอบที่แย่กว่าจะถูกเลือกมีน้อยลง ทำให้เอ็กซ์พลอเรชันมีน้อยลง แต่จะเป็นช่วงที่การค้นหาของอัลกอริธึมมุ่งไปที่การหาคำตอบที่เหมาะที่สุด ช่วงนี้จะ

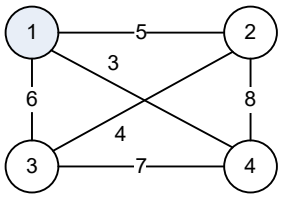
เรียกว่า เอ็กซ์พล้อยเตชัน (Exploitation) และการลู่เข้าหาคำตอบที่เหมาะที่สุดของอัลกอริธึมซีมูเลตเต็ดอันนีลลิ่งจะเกิดขึ้นในช่วงนี้

3.4 ตัวอย่างการหาค่าเหมาะที่สุดด้วยซีมูเลตเต็ดอันนีลลิ่ง

เพื่อเป็นตัวอย่างในการประยุกต์ใช้งานซีมูเลตเต็ดอันนีลลิ่ง ในหัวข้อนี้จะนำเสนอตัวอย่างของการหาค่าเหมาะที่สุดที่เป็นแบบคอมบินาทอเรียล (Combinatorial Optimization) ซึ่งตัวอย่างคือการแก้ปัญหาการเดินทางของเซลส์แมนและการจัดวางแพทเทิร์น

3.4.1 การเดินทางของเซลส์แมน

เพื่อที่จะทำความเข้าใจในการแก้ปัญหาการเดินทางของเซลส์แมนด้วยวิธีการหาค่าเหมาะที่สุดแบบซีมูเลตเต็ดอันนีลลิ่ง อย่างเป็นรูปธรรม รูปที่ 3.3 เป็นตัวอย่างที่มีเมืองที่จะต้องเดินทาง 4 เมืองคือ 1, 2, 3 และ 4 โดยที่แต่ละเมืองมีระยะทางต่อไปนี้



รูปที่ 3.3 ระยะทางระหว่างเมือง

การแก้ปัญหาการเดินทางของเซลส์แมนด้วยซีมูเลตเต็ดอันนีลลิ่ง เริ่มจากการกำหนดสถานะเริ่มต้น ที่เป็นคำตอบ เราสามารถสร้างคำตอบโดยเริ่มต้นด้วยการสุ่ม เช่นการเดินทางเริ่มต้น s คือ 1→2→3→4→1 ซึ่งมีระยะทางรวมทั้งหมดเท่ากับ 19 ในการหาคำตอบข้างเคียง เราอาจจะทำได้ด้วยการสลับระหว่างเมือง 1 และ 2 และได้เส้นทางใหม่ s’ คือ 2→1→3→4→2 ระยะทางทั้งหมดเท่ากับ 26 เนื่องจาก (s’–s) > 0 หรือค่าของ s ดีกว่าค่าของ s’ (เนื่องจากเราพิจารณาจากค่าที่ดีเป็นค่าน้อย) ดังนั้นการพิจารณาว่าจะยอมรับ s’ หรือไม่ให้พิจารณาจากค่า p ของสมการที่ 3.3

เพื่อที่จะหาค่าของ p ก่อนอื่นเราจะต้องกำหนดตารางอบอ่อนของ T ในที่นี้เราจะกำหนดตารางของ T โดยให้ T เป็นแบบเอ็กซ์โปเนนเชียลดังนี้ $T_{k+1} = \alpha T_k$ เมื่อ $\alpha = 0.3$ และให้ T สูงสุดมีค่าเท่ากับ 100 และต่ำสุดเป็น 0

ดังนั้นค่า p ของ ΔE คือ $e^{-(26-19)/100}$ = 0.93

สุ่มหาค่า p’ จาก 0 – 1 ได้เท่ากับ 0.32

ค่าของ p > p’ ดังนั้น s’ ถูกเลือก ให้เป็นสถานะปัจจุบัน

คำตอบปัจจุบันใหม่คือ 2 ⟶1⟶3⟶4⟶2 ระยะทางทั้งหมดเท่ากับ 26

หาคำตอบข้างเคียงใหม่ ด้วยการสลับระหว่างเมือง 3 และ 4 และได้เส้นทางใหม่ s’ คือ 2 ⟶ 1 ⟶ 4 ⟶ 3 ⟶ 2 ระยะทางทั้งหมดเท่ากับ 19 เนื่องจาก (s’–s) < 0 หรือค่าของ s’ ดีกว่าค่า s ดังนั้นยอมรับ s’ เป็นสถานะปัจจุบัน

สถานะปัจจุบันคือ 2 ⟶ 1 ⟶ 4 ⟶ 3 ⟶ 2 ระยะทางทั้งหมดเท่ากับ 19

หาคำตอบข้างเคียงใหม่ ด้วยการสลับระหว่างเมือง 1 และ 4 ของเส้นทางปัจจุบัน ได้เส้นทางใหม่ s’ คือ 2 ⟶ 4 ⟶ 1 ⟶ 3 ⟶ 2 ระยะทางทั้งหมดเท่ากับ 21 เนื่องจาก (s’–s) > 0 หรือค่าของ s ดีกว่า s’ ดังนั้นการพิจารณาว่าจะยอมรับ s’ ให้พิจารณาจากค่า p และค่า T = 30*0.3 = 9 ผลที่ได้ออกมาดังนี้

ค่า p ของ ΔE คือ $e^{-(21-19)/9}$ = 0.80

สุ่มหาค่า p’ จาก 0 – 1 ได้เท่ากับ 0.62

ค่าของ p < p’ ดังนั้น s’ จะถูกเลือก ให้ใช้ s’ เป็นสถานะปัจจุบัน

ได้เส้นทางใหม่ s’ คือ 2⟶4⟶1⟶3⟶2 ระยะทางทั้งหมดเท่ากับ 21

หาคำตอบข้างเคียงใหม่ ด้วยการสลับระหว่างเมือง 1 และ 2 ของเส้นทางปัจจุบัน ได้เส้นทางใหม่ s’ คือ 1⟶4⟶2⟶3⟶1 ระยะทางทั้งหมดเท่ากับ 21 เนื่องจาก (s’–s) = 0 หรือค่าของ s’ เท่ากับ s ดังนั้นการพิจารณาว่าจะยอมรับ s’ ให้พิจารณาจากค่า p และค่า T = 9*0.3 = 2.7 ผลที่ได้ออกมาดังนี้

ค่า p ของ ΔE คือ $e^{-(21-21)/2.7}$ = 1

ให้ใช้ s’ เป็นสถานะปัจจุบัน

ได้เส้นทางใหม่ s’ คือ 1⟶4⟶2⟶3⟶1 ระยะทางทั้งหมดเท่ากับ 21

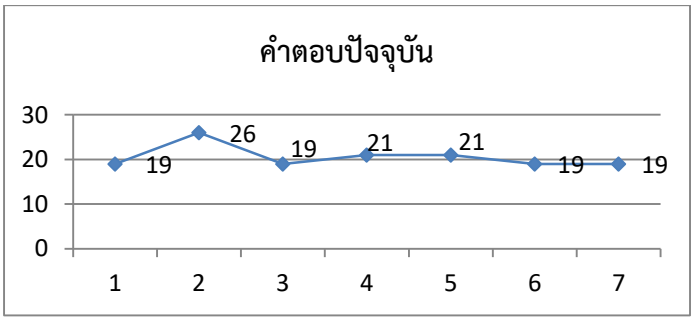
หาคำตอบข้างเคียงใหม่ ด้วยการสลับระหว่างเมือง 3 และ 2 ของเส้นทางปัจจุบัน ได้เส้นทางใหม่ s’ คือ 1⟶4⟶3⟶2⟶1 ระยะทางทั้งหมดเท่ากับ 19 เนื่องจาก (s’–s) < 0 หรือค่าของ s’ ดีกว่า s ดังนั้นยอมรับ s’ เป็นสถานะปัจจุบัน ค่า T = 2.7*0.3 = 0.81

ได้เส้นทางใหม่ s’ คือ 1⟶4⟶3⟶2⟶1 ระยะทางทั้งหมดเท่ากับ 19

หาคำตอบข้างเคียงใหม่ ด้วยการสลับระหว่างเมือง 3 และ 4 ของเส้นทางปัจจุบัน ได้เส้นทางใหม่ s’ คือ 1⟶3⟶4⟶2⟶1 ระยะทางทั้งหมดเท่ากับ 26 หรือค่าของ s ดีกว่า s’ ดังนั้นการพิจารณาว่าจะยอมรับ s’ หรือไม่ พิจารณาจากค่า p และค่า T = 0.81*0.3 ≈ 0 ผลที่ได้ออกมาดังนี้

ค่า p ของ ΔE คือ $e^{-(26-19)/0} = 0$

ได้เส้นทางปัจจุบัน s คือ 1⟶4⟶3⟶2⟶1 ระยะทางทั้งหมดเท่ากับ 19



รูปที่ 3.4 การลู่เข้าสู่คำตอบของปัญหาการเดินทางของเซลส์แมน

นับตั้งแต่รอบนี้เป็นต้นไป ค่าของ p จะออกมาเป็น 0 ตลอด ดังนั้นการสร้างคำตอบข้างเคียงใหม่ ถ้าไม่ดีกว่าคำตอบปัจจุบัน จะไม่ได้รับการยอมรับเลย

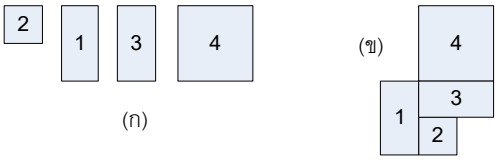
จากรูปที่ 3.4 เป็นการแสดงการลู่เข้าสู่คำตอบของปัญหาการเดินทางของเซลส์แมนตามที่ได้ดำเนินการมา แกน x จะเป็นจำนวนรอบของการทำงาน แกน y คือค่าของระยะทางรวม จะสังเกตได้ว่าในการทำงานรอบแรกๆ คำตอบที่มีค่าแย่กว่าจะมีโอกาสได้รับการคัดเลือกเพื่อเป็นคำตอบปัจจุบัน แต่

เมื่อการทำงานในรอบที่มากขึ้น ค่าของ T จะลดลง และทำให้โอกาสที่คำตอบปัจจุบันจะเป็นคำตอบที่แย่กว่าเดิมนั้นเป็นไปไม่ได้

3.4.2 การจัดวางแพทเทิร์น

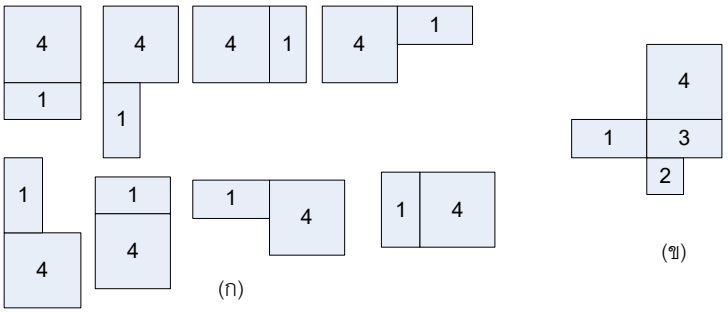
การแก้ปัญหาการจัดวางแพทเทิร์นเป็นวิธีการจัดวางแพทเทิร์นที่มีขนาดต่างกันหลายอันให้วางอยู่บนพื้นที่ที่น้อยที่สุด โดยที่แพทเทิร์นทุกอันจะไม่ซ้อนทับกัน

เพื่อเป็นการแสดงตัวอย่าง ในที่นี่จะใช้แพทเทิร์นที่เป็นรูปสี่เหลี่ยม ขนาด 1x1 จำนวน 1 รูป ขนาด 1x2 จำนวน 2 รูป และขนาด 2x2 จำนวน 1 รูป ดังรูปที่ 3.5(ก) โดยที่เราสร้างการจัดวางเริ่มต้นต้นด้วยการวางแบบสุ่มออกมาได้ดังรูปที่ 3.5(ข) ให้เป็นแพทเทิร์นจัดวางปัจจุบันของสถานะ s คำนวณพื้นที่ที่ใช้ในการจัดวางทั้งหมดโดยการนำด้านที่ยาวที่สุดคุณด้วยด้านที่กว้างที่สุด ได้เท่ากับ 4x3 = 12 หน่วย



รูปที่ 3.5 ขนาดของแพทเทิร์นที่จะทำการจัดวางและแพทเทิร์นจัดวางเริ่มต้น

เพื่อเป็นการสร้างคำตอบข้างเคียง เราจะใช้เทคนิคที่พัฒนาโดย ดากลีและตาโตกู (C.H. and Tatogu, M.Y. Dagli, 1987) และพัฒนาต่อโดย บุญเจริญ ศิริเนาวกุล และ ไพรัช ธัชยพงษ์ (B. and Thajchayapong, P. Sirinaovakul, 1994) วิธีการก็คือ เราจะให้แพทเทิร์นหนึ่งวางอยู่ที่มุมของอีกแพทเทิร์นหนึ่งเสมอ จากนั้นให้หมุนแพทเทิร์นที่วางอยู่ข้างเคียงกันทั้งหมดทีละแพทเทิร์นดังแสดงในรูปที่ 3.6(ก) แสดงแพทเทิร์นที่ 1 หมุนรอบแพทเทิร์นที่ 4 ดังนั้นแพทเทิร์นข้างเคียงของแพทเทิร์นปัจจุบันตามรูปที่ 3.5(ข) คือรูปที่ 3.6(ข) ให้สถานะใหม่ของรูป รูปที่ 3.6(ข) เป็นสถานะ s’ คำนวณพื้นที่ที่ใช้ในการจัดวางทั้งหมด ได้เท่ากับ 4x4 = 16 หน่วย

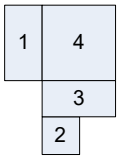


รูปที่ 3.6 การจัดวางแพทเทิร์นและการสร้างแพทเทิร์นข้างเคียง

เนื่องจาก (s'-s) > 0 หรือแพทเทิร์นข้างเคียงมีพื้นที่มากกว่าแพทเทิร์นปัจจุบัน ดังนั้นการพิจารณาว่าจะยอมรับ s' แทน s หรือไม่จะต้องหาจากค่า p โดยที่กำหนดให้ T มีค่าสูงสุดเท่ากับ 100 และมีค่า α = 0.3 ผลที่ได้เป็นดังนี้

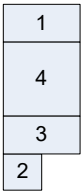
$p = e^{-(16-12)/100} = 0.96$

หาค่า p' จากการสุ่มได้เป็น 0.63 ดังนั้น p > p' ให้ยอมรับ s' เป็นสถานะปัจจุบัน สร้างคำตอบข้างเคียงของสถานะปัจจุบันได้ดังรูปที่ 3.7 เป็นสถานะ s' มีค่า T = 30 คำนวณพื้นที่ที่ใช้ทั้งหมดได้เท่ากับ 3x4 = 12 หน่วย ดีกว่าสถานะปัจจุบัน s ที่ใช้พื้นที่เท่ากับ 16 หน่วย ดังนั้นให้ s' เป็นสถานะปัจจุบัน s



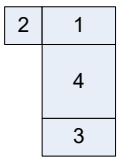
รูปที่ 3.7 สถานะข้างเคียงของแพทเทิร์นปัจจุบันโดยหมุนแพทเทิร์น 1

สร้างคำตอบหรือแพทเทิร์นข้างเคียงของสถานะปัจจุบันได้ดังรูปที่ 3.8 เป็นสถานะ s' มีค่า T = 9 คำนวณพื้นที่ที่ใช้ทั้งหมดได้เท่ากับ 5x2 = 10 หน่วย ดีกว่าสถานะปัจจุบัน s ที่ใช้พื้นที่เท่ากับ 12 หน่วย ดังนั้นให้ s' เป็นสถานะปัจจุบัน s



รูปที่ 3.8 สถานะข้างเคียงของแพทเทิร์นปัจจุบันโดยหมุนแพทเทิร์น 1

สร้างแพทเทิร์นข้างเคียงของสถานะปัจจุบันได้ดังรูปที่ 3.9 เป็นสถานะ s’ มีค่า T = 2.7 คำนวณพื้นที่ที่ใช้ทั้งหมดได้เท่ากับ 3x4 = 12 หน่วย ใช้พื้นที่มากกว่าแพทเทิร์นปัจจุบัน s ซึ่งใช้พื้นที่ 10 หน่วย



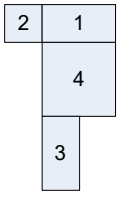
รูปที่ 3.9 สถานะข้างเคียงของแพทเทิร์นปัจจุบันโดยหมุนแพทเทิร์น 2

เนื่องจาก (s’-s) > 0 หรือแพทเทิร์นข้างเคียงมีพื้นที่มากกว่าแพทเทิร์นปัจจุบัน ดังนั้นการพิจารณาว่าจะยอมรับ s’ แทน s หรือไม่จะต้องหาจากค่า p ได้ผลเป็นดังนี้

$p = e^{-(12-10)/2.7} = 0.47$

หาค่า p’ จากการสุ่มได้เป็น 0.42 ดังนั้น p > p’ ให้รับ s’ เป็นสถานะปัจจุบัน

สร้างแพทเทิร์นข้างเคียงของสถานะปัจจุบันได้ดังรูปที่ 3.10 เป็นสถานะ s’ มีค่า T = 0.81 คำนวณพื้นที่ที่ใช้ทั้งหมดได้เท่ากับ 3x5 = 15 หน่วย ซึ่งใช้พื้นที่มากกว่าแพทเทิร์นปัจจุบัน s ซึ่งใช้พื้นที่ 12 หน่วย



รูปที่ 3.10 สถานะข้างเคียงของแพทเทิร์นปัจจุบันโดยหมุนแพทเทิร์น 3

เนื่องจาก (s'-s) > 0 หรือแพทเทิร์นข้างเคียงมีพื้นที่มากกว่าแพทเทิร์นปัจจุบัน ดังนั้นการพิจารณาว่าจะยอมรับ s' แทน s หรือไม่จะต้องหาจากค่า p ได้ผลเป็นดังนี้

$p = e^{-(15-12)/0.81} = 0.024$

หาค่า p' จากการสุ่มได้เป็น 0.32 ดังนั้น p > p' ให้ s ยังเป็นสถานะปัจจุบัน

สร้างแพทเทิร์นข้างเคียงของสถานะปัจจุบันได้ดังรูปที่ 3.11 เป็นสถานะ s' มีค่า T = 0.243 คำนวณพื้นที่ที่ใช้ทั้งหมดได้เท่ากับ 3x3 = 9 หน่วย ดีกว่าสถานะปัจจุบัน s ที่ใช้พื้นที่เท่ากับ 12 หน่วย ดังนั้นให้ s' เป็นสถานะปัจจุบัน s

2	1
3	4

รูปที่ 3.11 สถานะข้างเคียงของแพทเทิร์นปัจจุบันโดยหมุนแพทเทิร์น 3

สร้างแพทเทิร์นข้างเคียงของสถานะปัจจุบันด้วยการหมุนแพทเทิร์น 4 ไปรอบแพทเทิร์นที่เหลือให้อยู่ข้างแพทเทิร์น 1 ดังรูปที่ 3.12(ก) เป็นสถานะ s' มีค่า T = 0.729 คำนวณพื้นที่ที่ใช้ทั้งหมดได้เท่ากับ 4x5 = 20 หน่วย ซึ่งใช้พื้นที่มากกว่าสถานะปัจจุบัน s ที่ใช้พื้นที่เท่ากับ 9 หน่วย

เนื่องจาก (s'-s) > 0 หรือแพทเทิร์นข้างเคียงมีพื้นที่มากกว่าแพทเทิร์นปัจจุบัน ดังนั้นการพิจารณาว่าจะยอมรับ s' แทน s หรือไม่จะต้องหาจากค่า p ได้ผลเป็นดังนี้

$p = e^{-(15-12)/0.81} = 2.79 \times 10^{-7}$

หาค่า p' จากการสุ่มได้เป็น 0.41 ดังนั้น p > p' ให้ s ยังเป็นสถานะปัจจุบัน

สร้างแพทเทิร์นข้างเคียงของสถานะปัจจุบันด้วยการหมุนแพทเทิร์น 4 ไปรอบแพทเทิร์นที่เหลือให้อยู่บนแพทเทิร์น 2 และ 1 ดังรูปที่ 3.12(ข) เป็นสถานะ s' มีค่า T = 0.022 คำนวณพื้นที่ที่ใช้ทั้งหมดได้เท่ากับ 3x5 = 15 หน่วย ซึ่งใช้พื้นที่มากกว่าสถานะปัจจุบัน s ที่ใช้พื้นที่เท่ากับ 9 หน่วย

เนื่องจาก (s'-s) > 0 หรือแพทเทิร์นข้างเคียงมีพื้นที่มากกว่าแพทเทิร์นปัจจุบัน ดังนั้นการพิจารณาว่าจะยอมรับ s' แทน s หรือไม่จะต้องหาจากค่า p ได้ผลเป็นดังนี้

$p = e^{-(15-9)/0.022} \approx 0$

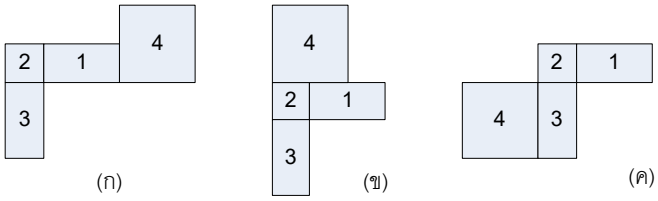
หาค่า p' จากการสุ่มได้เป็น 0.22 ดังนั้น p > p' ให้ s ยังเป็นสถานะปัจจุบัน

สร้างแพทเทิร์นข้างเคียงของสถานะปัจจุบันด้วยการหมุนแพทเทิร์น 4 ไปรอบแพทเทิร์นที่เหลือให้อยู่ข้างแพทเทิร์น 3 ดังรูปที่ 3.12(ค) เป็นสถานะ s' มีค่า T = 0 คำนวณพื้นที่ที่ใช้ทั้งหมดได้เท่ากับ 3x5 = 15 หน่วย ซึ่งใช้พื้นที่มากกว่าสถานะปัจจุบัน s ที่ใช้พื้นที่เท่ากับ 9 หน่วย

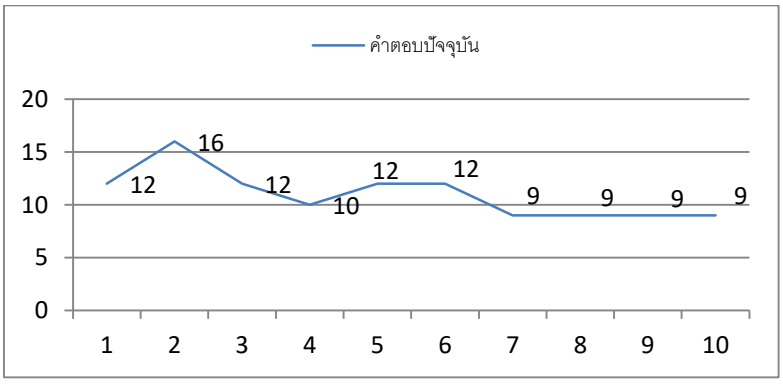
เนื่องจาก (s'-s) > 0 หรือแพทเทิร์นข้างเคียงมีพื้นที่มากกว่าแพทเทิร์นปัจจุบัน ดังนั้นการพิจารณาว่าจะยอมรับ s' แทน s หรือไม่จะต้องหาจากค่า p ได้ผลเป็นดังนี้

$p = e^{-(15-9)/0} = 0$

หาค่า p' จากการสุ่มได้เป็น 0.81 ดังนั้น p > p' ให้ s ยังเป็นสถานะปัจจุบัน



รูปที่ 3.12 การหาแพทเทิร์นข้างเคียงด้วยการหมุนแพทเทิร์น 4



รูปที่ 3.13 การลู่เข้าสู่คำตอบของการแก้ปัญหาการวางแพทเทิร์น

จากรูปที่ 3.13 การลู่เข้าสู่คำตอบของปัญหาการวางแพทเทิร์นนี้จะมีลักษณะที่คล้ายกับปัญหาของการเดินทางของเซลส์แมน ที่การค้นหาจะยอมรับค่าที่แย่กว่าถึง 2 ครั้ง คือครั้งแรกจากค่า 12 ไปเป็น 16 และครั้งที่สองจาก 10 ไปเป็น 12 จากกราฟของการลู่เข้าสู่คำตอบ เราจะเห็นการเกิดโลคอลมินิมั่ม ที่ 12 ในการทำงานรอบแรก และที่การทำงานรอบที่ 4 ก่อนที่จะเข้าสู่โกลบอลมินิมั่มที่การทำงานรอบที่ 7 เป็นต้นไป ในช่วงของการยอมรับค่าที่แย่กว่านี้เป็นเสมือนการเดินขึ้นเขา ซึ่งจะเกิดขึ้นเป็นครั้งเป็นคราว เนื่องจาก T ยังมีค่าสูง ทำให้ระบบมีโอกาสที่จะทำการสำรวจปริภูมิคำตอบที่เป็นไปได้มากขึ้น ซึ่งในระบบถึงว่าเป็นช่วงของเอ็กซ์พลอเรชัน และที่จุดการทำงานตั้งแต่รอบที่ 7 เป็นต้นไปนี้ เนื่องจากค่าของ T ต่ำมากดังนั้น การยอมรับค่าที่แย่กว่าจะเป็นไปได้น้อย ทำให้ระบบต้องเลือกคำตอบที่ดีที่สุดตลอดเวลา ซึ่งถือเป็นเอ็กซ์พล้อยเตชันของระบบ

3.5 การทำงานขั้นสูงของอัลกอริธึม

ซีมูเลตเต็ดอันนีลลิ่ง เป็นอัลกอริธึมที่ถูกออกแบบมาใช้สำหรับการแก้ปัญหาคอมบินาทอเรียล สำหรับการนำซีมูเลตเต็ดอันนีลลิ่งมาใช้กับปัญหาเชิงตัวเลขนั้น จะต้องมีการดัดแปลงกระบวนการทำงานของอัลกอริธึม

3.5.1 ซีมูเลตเต็ดอันนีลลิ่งสำหรับตัวแปรที่มีค่าต่อเนื่อง

โคราณาและคณะ (Corana, 1987) เป็นคนแรกๆ ที่นำเสนอวิธีการซีมูเลตเต็ดอันนีลลิ่งสำหรับการแก้ปัญหาค่าเหมาะที่สุดเชิงตัวเลข โดยที่มีค่าของพารามิเตอร์ที่ใช้ในการแก้ปัญหาเป็นเลขจำนวนจริง กล่าวอีกนัยหนึ่งคือ ตัวแปรของปัญหามีค่าเป็นเลขจำนวนต่อเนื่อง ลักษณะของปัญหาคือ ให้ X เป็นเวคเตอร์ที่อยู่ในโดเมนของเลขจำนวนจริง R^n โดยมี $(x_1, x_2,..., x_n)$ เป็นคอมโพเน้นท์ (Components) ของเวคเตอร์ ให้ f(x) เป็นฟังก์ชันที่เราต้องการหาค่าเหมาะที่สุด และให้ x_n เป็นตัวแปร เมื่อค่าของ x คือ $a_1 < x_1 < b_1, \ldots, a_n < x_n < b_n$ โดยที่ x_n เป็นตัวแปรในเวคเตอร์ที่มีทั้งหมด n ตัว ตัวแปร x_i แต่ละตัวมีค่าอยู่ในขอบเขตระหว่าง a_i และ b_i เมื่อ i = 1 ... n โดยที่ขอบเขตนี้เป็นขอบเขตจำกัด และมีช่วงที่ต่อเนื่อง (Continuous interval) สำหรับค่าของ f ไม่จำเป็นต้องมีค่าต่อเนื่องแต่ต้องมีขอบเขตที่ชัดเจน (Bounded)

การทำงานของอัลกอริธึมนี้จะเป็นแบบวนรอบ โดยเริ่มต้นจากเวคเตอร์ X_0 เป็นจุดเริ่มต้น จากจุดนี้ อัลกอริธึมจะหาจุดต่อๆ ไปคือ X_0, X_1, ... , X_i, ... ที่มุ่งไปสู่โกลบอลมินิมั่ม ที่ฟังก์ชัน f มีค่าน้อยที่สุด จุดพิกัด (Coordinate) ที่เป็นทางเลือกใหม่จะถูกสร้างขึ้นตามทิศทางรอบๆ จุดปัจจุบันของ X_i ด้วยการเดินแบบสุ่ม และค่าของจุดพิกัดใหม่จะมีการแจกแจงแบบสม่ำเสมอ (Uniform distribution) ที่มีช่วงของการแจกแจงที่มี X_i อยู่ตรงกลาง และตรงกลางระหว่างช่วงห่างของจุดพิกัดจะถูกบันทึกว่าเป็นสเต็ป (Step) ของเวคเตอร์ V ถ้าจุดที่ได้ตกอยู่นอกโดเมนที่กำหนดของฟังก์ชัน f ให้สร้างจุดใหม่โดยการสุ่ม ให้ทำการสุ่มและทดสอบเช่นนี้ไปจนกระทั่งได้จุดใหม่ที่อยู่ในโดเมนที่กำหนด จุดที่เป็นตัวเลือกใหม่ X' จะได้รับการยอมรับหรือปฏิเสธขึ้นอยู่กับเกณฑ์ของเมโทรโพลิส (Metropolis criterion) ดังนี้

if $(\Delta f) \leq 0$ then ให้ยอมรับจุดที่ได้มาใหม่ $X_{i+1} = X'$

else ให้ยอมรับจุดใหม่ด้วยความน่าจะเป็นดังนี้

$p(\Delta f) = \exp^{(-\Delta f/T)}$

เมื่อ $(\Delta f) = f(X') - f(x_i)$ และ T เป็นอุณหภูมิ

ที่จุดหนึ่งๆ ของ T การเรียงกันของค่า X_0, X_1, ... , X_i, ... จะไม่ลดลง ยกเว้นในกรณีที่ T = 0 เมื่อ T มีค่าสูง จุดที่ถูกสร้างขึ้นมาโดยส่วนใหญ่จะได้รับการยอมรับ

การทำงานของอัลกอริธึมจะเริ่มที่ค่า T_0 มีค่าสูง จุดต่างๆ จะถูกสร้างขึ้นจนกระทั่งระบบเข้าสู่ภาวะสมดุล ซึ่งหมายความว่าลำดับของจุด X_i ที่ค่าเฉลี่ยของฟังก์ชัน f ไปถึงจุดที่เสถียร เมื่อค่า i เพิ่มขึ้นในเฟสนี้ สเต็ปเวคเตอร์ (Step vertor) V_m ก็จะถูกปรับให้ดีขึ้นตามพฤติกรรมของฟังก์ชัน f จุดที่ดีที่สุดจะถูกจดจำไว้เป็น X_{opt}

เมื่อได้ค่าที่เป็นจุดเสถียร X_{opt} แล้วค่าของ T ก็จะลดลง แล้วจุดต่างๆ ก็จะถูกสร้างขึ้นมาเพื่อหาจุด X_{opt} ใหม่ การทำงานจะวนรอบเช่นนี้ไปเรื่อยๆ

การสิ้นสุดของอัลกอริธึมจะเกิดเมื่อค่าของ T มีค่าต่ำเพียงพอ จนเมื่อลดค่า T ให้ต่ำกว่านี้ก็จะไม่มีประโยชน์อะไรต่อการทำให้ค่าที่ได้ดีขึ้น สำหรับการทำงานของอัลกอริธึมอย่างละเอียดของโครานามีดังต่อไปนี้

ขั้นตอนที่ 0 (กำหนดค่าเริ่มต้น)

กำหนดค่าเริ่มต้นของตัวแปรดังนี้

X_0, V_0 และ T_0

ค่า □ แทนเกณฑ์การยุติการทำงาน และ □□ เป็นเลขนับที่ต่อเนื่องของการลดอุณหภูมิเพื่อทดสอบการยุติการทำงาน

ค่า N_S เป็นการเปลี่ยนแปลงของสเต็ป และเกณฑ์การเปลี่ยนแปลงคือ C

ค่า N_T เป็นการทดสอบการลดอุณหภูมิ และค่าสัมประสิทธิ์ของการลดคือ r_T

ตั้งค่าของ i, j, m, k เป็น 0

ค่า i คือดัชนีของจุดต่างๆ ที่เกิดขึ้นในโดเมน

ค่า j เป็นดัชนีที่แสดงจำนวนรอบตามทิศทางทั้งหมด

ค่า m เป็นค่าที่บอกการปรับสเต็ปที่ต่อเนื่อง

ค่า k บอกการลดอุณหภูมิที่ต่อเนื่อง

ตั้งค่า h = 1

ค่า h เป็นดัชนีที่แสดงทิศทางตามการเดินทางของจุดที่สร้างขึ้น โดยเริ่มจากจุดที่ได้รับการยอมรับหลังสุด

คำนวณค่า $f_0 = f(X_0)$

กำหนดให้ $X_{opt} = X_0$, $f_{opt} = f_o$

กำหนดให้ $n_u = 0$, u = 1, ..., n

กำหนดให้ $f_u^* = f_0$, u = 0. -1, ..., -□□ – 1.

ขั้นตอนที่ 1

ให้เริ่มต้นจากจุด X_i และสร้างจุดใหม่ขึ้นมาด้วยการสุ่ม X’ ตามทิศทางของ h

$$X' = X_i + rV_{mh}E_h$$

โดยที่ r เป็นค่าสุ่มที่มีช่วงอยู่ระหว่าง [-1,1] ตัวแปร E_h เป็นเวคเตอร์ที่แสดงจุดพิกัดที่ทิศทางที่ h และ V_{mh} คือเป็นองค์ประกอบของสเต็ปเวคเตอร์ V_m ในทิศทางเดียวกัน

ขั้นตอนที่ 2

ถ้าจุดพิกัดที่ h ของ X' วางอยู่นอกโดเมนของ f ซึ่งก็คือ x_h ต่ำกว่า a_h หรือ x_h สูงกว่า b_n (นอกขอบเขต $a_h < x_h < b_n$) ให้ย้อนกลับไปที่ขั้นตอนที่ 1

ขั้นตอนที่ 3

คำนวณค่า f' = f(X')

ถ้า f' < f_i ให้ยอมรับค่าใหม่

ให้ X_{i+1} = X', f_{i+1} = f', i = i+1 และ $n_h = n_h + 1$

ถ้า f' < f_{opt} ให้ X_{opt} = X' และ f_{opt} = f' endif

else ถ้า (f' > f_i) การยอมรับหรือปฏิเสธให้เป็นพิจารณาจาก

□ = □□□$^{(\frac{□□-□□}{□□})}$

สุ่มค่า p' ที่มีช่วงระหว่าง [0,1] แล้วเปรียบเทียบกับ p ที่คำนวณมาได้ ถ้า p' < p ให้ยอมรับค่าใหม่

ให้ X_{i+1} = X', f_{i+1} = f', i = i+1 และ $n_h = n_h + 1$

ขั้นตอนที่ 4

h = h +1

ถ้า h □ n ให้ไปที่ขั้นตอนที่ 1

ถ้าไม่ใช่ ให้ h =1 และ j = j+1

ขั้นตอนที่ 5

ถ้า j < N_S ให้ไปที่ขั้นตอนที่ 1

มิฉะนั้น ปรับปรุงสเต็ปเวคเตอร์ของ V_m

สำหรับแต่ละทิศทางของ u องค์ประกอบใหม่ของสเต็ปเวคเตอร์ V_u' คือ

$$\square_{\square}^{\square} = \square_{\square\square}\left(1 + \square_{\square}\frac{\frac{\square_{\square}}{\square_{\square}} - 0.6}{0.4}\right); \square\square - \square_{\square} > 0.6\square_{\square}$$

$$\square_{\square}^{\square} = \frac{\square_{\square\square}}{\left(1+\square_{\square}\frac{0.4-\frac{\square_{\square}}{\square_{\square}}}{0.4}\right)}; \square\square - \square_{\square} < 0.4\square_{\square}$$

$$\square_{\square}^{\square} = \square_{\square\square}$$

ให้ค่า V_{m+1} =V', j = 0, n_u = 0 (u=1,...,n) และ m = m+1

จุดประสงค์ของการใช้สมการข้างต้น คือ เพื่อการควบคุมความยาวของสเต็ปในการทำให้เปอร์เซ็นต์เฉลี่ยของการยอมรับค่าใหม่อยู่ที่ประมาณครึ่งหนึ่งของจำนวนค่าใหม่ทั้งหมด ตัวแปร c_u เป็นพารามิเตอร์ที่ใช้สำหรับการควบคุมการเปลี่ยนแปลงของสเต็ปตามทิศทางแต่ละอันของที่ตำแหน่ง u

ขั้นตอนที่ 6

ถ้า m<N_T ให้ไปที่ขั้นตอนที่ 1 ถ้าไม่ใช่ให้ลดอุณหภูมิ T_K

มิฉะนั้นให้ $T_{K+1} = r_T \cdot T_K$, $f_K^* = f_i$, k = k+1, และ m = 0

ไม่มีประโยชน์ที่จะลดอุณหภูมิทุก N_S แต่การลดที่รอบของ N_T จะเหมาะสมกว่า

ขั้นตอนที่ 7 (เกณฑ์การหยุด)

ถ้า $|\square_{\square}^{*} - \square_{\square-\square}^{*}| \leq \square, \square = 1, \ldots, \square$ และ $\square_{\square}^{*} - \square_{\square\square\square} \leq \square$ ให้ยุติการค้นหา

มิฉะนั้น i = i+1, $X_i = X_{opt}$, $f_i = f_{opt}$

กลับไปที่ขั้นตอนที่ 1

จากการทดลองของโคราณาพารามิเตอร์ต่างๆ ที่เหมาะสำหรับการทำงานของอัลกอริธึมนี้ก็คือ

N_S = 20, N_T = max(100, 5*n), $\square_{\square}$ = 4, r_T = 0.85

c_i = 2 (i = 1, ..., n)

การปรับกระบวนการทำงาน

ในการทำงาน ควรจะให้ระบบมีสัดส่วนของการยอมรับและการปฏิเสธเท่ากับร้อยละ 50 ให้ได้

3.5.2 ซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัว

ซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัว (Adaptive Simulated Annealing Algorithm) นำเสนอครั้งแรกโดย อิงเบอร์ (Ingber, 1989) (Ingber, 1993) เป็นการหาค่าเหมาะที่สุดเชิงตัวเลขที่ดัดแปลงเพิ่มเติมจากซีมูเลตเต็ดอันนิลลิ่ง โดยการสร้างพารามิเตอร์สำหรับการควบคุมตารางอุณหภูมิและการปรับเปลี่ยนของอุณหภูมิแบบอัตโนมัติ ในระหว่างการทำงาน ซึ่งทำให้อัลกอริธึมนี้ไม่มีผลกระทบจากการปรับเปลี่ยนอุณหภูมิที่เป็นแบบกำหนดเองโดยผู้ใช้งาน อัลกอริธึมนี้ทำงานโดยการใช้พารามิเตอร์ที่มีค่าเป็นเลขจำนวนจริง และมีจำนวนมิติของพารามิเตอร์เป็นแบบไฮเปอร์คิว (Hypercube) เท่ากับ D อัลกอริธึมนี้มีองค์ประกอบใหญ่ที่สำคัญอยู่ 3 อย่างคือ

การสร้างฟังก์ชันความหนาแน่นของความน่าจะเป็น (Generating Probability Density Functions)

การยอมรับฟังก์ชันความหนาแน่นของความน่าจะเป็น (Acceptance Probability Density Functions) และ

ตารางอุณหภูมิ (Temperature Schedules)

การสร้างฟังก์ชันความหนาแน่นของความน่าจะเป็น กำหนดจากพารามิเตอร์ที่มีมิติเท่ากับ D ของ p^i ที่มีช่วงอยู่ระหว่าง $[A_i, B_i]$ โดยที่ i มีค่าเท่ากับ 1, ..., D และที่ p^i ตัวที่ k ซึ่งเป็นสถานะปัจจุบัน (p_k^i) การสร้างจุดใหม่ขึ้นมา จะทำโดยการแจกแจงที่เกิดจากผลคูณของการแจกแจงของพารามิเตอร์แต่ละตัวของ $g^i(y^i; T_i)$ ที่มีตัวแปรสุ่ม y^i ที่อยู่ในช่วง [-1, 1] โดยกำหนดว่า $\square_{\square}^{\square\square\square} = \square_{\square}^{\square\square\square} + \square_{\square}(\square_{\square} - \square_{\square})$ และอุณหภูมิ T_i ในรูปของ g^i คือ

$$\square^{\square}(\square^{\square}, \square_{\square}) = \frac{1}{2(|\square^{\square}| + \square_{\square})\square\square\left(1 + \frac{1}{\square_{\square}}\right)}$$

การยอมรับฟังก์ชันความหนาแน่นของความน่าจะเป็น เป็นการพิจารณาเปรียบเทียบระหว่างค่าจากฟังก์ชันที่ใช้ของจุด 2 จุดคือ $C(p_{k+1})$-$C(p_k)$ กับค่าที่ได้จากตัวแปรสุ่มแบบยูนิฟอร์ม U ที่อยู่ในช่วงระหว่าง [0, 1] ของการทดสอบ บอลท์ซแมนน์ (Boltzmann) ดังนี้

$$\square\square\square\left(\frac{\left(\square(\square^{\square\square\square}) - \square(\square^{\square\square\square})\right)}{\square_{\square\square\square\square}}\right) > U$$

เมื่อ T_{cost} เป็นอุณหภูมิที่ใช้ในการทดสอบครั้งนี้ ดังนั้นค่าใหม่จะได้รับการยอมรับถ้าสมการข้างต้นเป็นจริง มิฉะนั้นจุดเดิมจะถูกใช้สำหรับการทำงานในรอบถัดไป

ตารางอุณหภูมิ เกี่ยวกับตารางอุณหภูมิของซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัวนั้นจะเป็นเรื่องของตารางอุณหภูมิการอบอ่อนซ้ำ (Reannealing temperature schedule) ตารางอบอ่อนสำหรับพารามิเตอร์ของอุณหภูมิ T_i จากอุณหภูมิเริ่มต้น T_{i0} จะเป็นดังนี้

$$\square_{\square}(\square_{\square}) = \square_{0\square}\square\square\square(-\square_{\square}\square_{\square}^{\frac{1}{\square}})$$

พารามิเตอร์ของอุณหภูมินี้สามารถปรับได้เป็นช่วงๆ หรืออาจจะเพิ่มขึ้นตามค่าที่มีอยู่เดิม การทำอันนิลลิ่งซ้ำของค่า (Cost) อุณหภูมิจะทำให้เกิดการเริ่มต้นใหม่ของสเกลของอันนิลลิ่ง ซึ่งมีเกณฑ์การยอมรับค่า (Cost) เป็นดังนี้

$$\square_{\square\square\square\square}(\square_{\square\square\square\square}) = \square_{0\square\square\square\square}\square\square\square(-\square_{\square\square\square\square}\square_{\square\square\square\square}^{\frac{1}{\square}})$$

ค่า T_{0cost} ตัวใหม่เป็นค่าใช้จ่ายของอุณหภูมิเริ่มต้นที่เกิดขึ้นในปัจจุบัน ค่า T_{cost} เป็นค่าใช้จ่ายของอุณหภูมิ ค่า k_{cost} เป็นดัชนีเวลาของการอันนีลลิ่ง ค่า c_{cost} เป็นแฟกเตอร์การปรับ และ D เป็นจำนวนมิติของพารามิเตอร์

เมื่อพิจารณาอ็อปเจ็คทีฟฟังก์ชันของเวคเตอร์ P ในรูปของสเกล่า E เราจะได้

$$E = f(P)$$

เมื่อ P = [p_1, ..., p_D] ซึ่งเป็นเวคเตอร์ที่จำนวนมิติเท่ากับ D และ p_i จะอยู่ในช่วงของ [A_i,B_i] โดยที่ขอบเขตสูงสุด **A** = [A_1, ..., A_D] และขอบเขตต่ำสุด **B** = [B_1, ..., B_D] การทำงานของซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัวหรือ ASA คือการหาจุดที่ดีที่สุดของ P ในการทำให้ E มีค่าต่ำสุด การทำงานจะแบ่งเป็นขั้นตอนดังนี้ (Shi, 2003)

ขั้นตอนที่ 1 เลือกค่าเริ่มต้น P^{init} เพื่อเป็นข้อมูลป้อนเข้าด้วยการสุ่ม กำหนดค่า c พารามิเตอร์ควบคุมอัตราการอบอ่อน และพารามิเตอร์เวลาการอบอ่อน k_{cost} และ k_i เป็น 0 กำหนดค่าอุณหภูมิการยอมรับเริ่มต้น $T_{cost}(k_{cost}) = f(P^{init})$ กำหนดให้อุณหภูมิการสร้างเริ่มต้น (Initial generating temperature) $T_i(k_i) = 1$ โดย i = 1, ..., D

ขั้นตอนที่ 2 สร้างจุดขึ้นมาใหม่ตามสมการดังนี้

$$□_□^{□□□} = □_□^{□□□} + □_□(□_□ - □_□)$$

ค่า y_i เป็นค่าสุ่มที่มีการแจกแจงอย่างสม่ำเสมอ (Uniform distribution) $u_i \in [0,1]$

$$□_□ = □□□(□_□ - 0.5)□_□(□_□) \times \left\{\left[1 + \frac{1}{□_□}\right]^{|2□_□-1|} - 1\right\}$$

ถ้าจุดที่สร้างขึ้นใหม่ P^{new} ไม่ได้อยู่ในขอบเขตของโดเมน ก็จะถูกยกเลิก แล้วสร้างใหม่ จนได้จุดใหม่ที่ต้องการ

ขั้นตอนที่ 3 จุดที่สร้างขึ้นมาใหม่จะได้รับการยอมรับหรือไม่จะพิจารณาจากฟังก์ชันดังต่อไปนี้

$$□_{□□□□□□} = \frac{1}{1 + □□□\left(\frac{\left(□(□^{□□□}) - □(□^{□□□})\right)}{□_{□□□□}}\right)}$$

ถ้าค่า P_{accept} ที่ได้มีค่ามากกว่าค่าที่ได้จากการสุ่ม [0,1] จุดที่สร้างใหม่ P^{new} จะได้รับการยอมรับ

ขั้นตอนที่ 4 หลังจากที่จุดใหม่ทุกจุด N_{accept} ได้รับการยอมรับ ให้ทำการอบอ่อนซ้ำ (Re-Annealing) โดยเริ่มต้นด้วยการหาเซ้นซิติวิตี้ (Sensitivity) ดังนี้

$$□_□ = \left| \frac{□(□^{□□□□} + (\eta □^{□□□□} □_□)□_□) - □(□^{□□□□})}{\eta □^{□□□□} □_□} \right|$$

เมื่อ P^{best} เป็นจุดที่ดีที่สุดเท่าที่หามา η เป็นสเต็ปที่มีขนาดเล็ก e_i เป็นเวคเตอร์ที่มีมิติขนาด D ซึ่งมีค่าในมิติที่ i เป็น 1 สำหรับเลขที่เหลือในมิติอื่นๆ คือ 0 และให้ $s_{max} = \max\{s_i, i = 1, ..., D\}$ ดังนั้น ในแต่ละรอบของอุณหภูมิ $T_i(k_i)$ และพารามิเตอร์เวลาการอบอ่อน k_i จะถูกเริ่มต้นใหม่ดังนี้

$$□_□(□_□) = \frac{□_{□□□}}{□_□} □_□(□_□)$$

$$□_□ = \left(\frac{1}{□} □□ \left(\frac{□_□(0)}{□_□(□_□)} \right) \right)^{□}$$

ขั้นตอนที่ 5 หลังจากที่จุดใหม่ทุกจุด N_{gen} ได้รับการสร้าง ให้เริ่มกระบวนการอบอ่อนดังนี้

$$□_□(□_□) = □_{0□} \exp(-□_□ □_□^{\frac{1}{□}})$$

$$□_{□□□□}(□_{□□□□}) = □_{0□□□□} □□□(-□_{□□□□} □_{□□□□}^{\frac{1}{□}})$$

ถ้า N_{gen} ใหม่ยังสร้างไม่ครบให้ไปที่ขั้นตอนที่ 2

ขั้นตอนที่ 6 กระทำงานจะยุติเมื่ออัลกอริธึมยอมรับ $N_{total\text{-}accept}$ ทั้งหมดหรือสร้าง $N_{total\text{-}gen}$ ทั้งหมดแล้ว มิฉะนั้นให้ไปที่ขั้นตอนที่ 2

3.5.3 การทดสอบสมรรถนะ

เนื่องจากซีมูเลตเต็ดอันนีลลิ่ง ไม่สามารถนำมาใช้กับการคำนวณตัวเลขแบบต่อเนื่องได้โดยตรง ดังนั้นการทดลองจึงต้องใช้ซีมูเลตเต็ดอันนีลลิ่งที่สามารถแก้ปัญหาเชิงตัวเลขได้ สโตร์นและคณะ (Storn, 1997) ได้ทำการทดลองหาสมรรถนะของอัลกอริธึม โดยนำซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัว (Adaptive simulated annealing: ASA) มาใช้ในการทดลองกับฟังก์ชันเทียบเคียงจำนวนมาก โดยตั้งค่าพารามิเตอร์ 2 ตัวคือ มาตราส่วนอัตราอุณหภูมิ (Temperature Ration Scale: TRS) และ มาตราส่วนการอบอ่อน (Temperature Annealing Scale: TAS) เพื่อที่จะหาจำนวนครั้งของฟังก์ชันที่ถูกเรียกใช้งาน (Number of Function Evaluations: nfe) ในการทดลองของสโตร์น การที่คำตอบได้ค่าต่ำว่า 10^{-6} ให้ถือว่า อัลกอริธึมนั้นพบค่าเหมาะที่สุด หรือค่าที่ต่ำที่สุด (เท่ากับ 0) แล้ว

ฟังก์ชัน	TRS	TAS	nfe
ทรงกลม	10^{-5}	10	397
ฟังก์ชันของโรเซนบร็อก	10^{-5}	10,000	11,275
ฟังก์ชันของกรีวังค์	10^{-5}	0.1	-

ตารางที่ 3.1 การทดสอบการทำงานของซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัว

จากผลการทดลองกับฟังก์ชันรูปทรงกลมเป็นฟังก์ชันที่ซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัวสามารถหาคำตอบได้ง่ายที่สุด ซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัวจะเริ่มมีปัญหาในการหาค่าเหมาะที่สุดของฟังก์ชันของโรเซนบร็อกบ้าง คือใช้เวลาในการหาคำตอบนานขึ้น เนื่องจากฟังก์ชันจะถูกเรียกใช้งานหลายครั้งขึ้น หรือกล่าวอีกนัยหนึ่งคือมีจำนวนรอบของการทำงานมากขึ้น สำหรับฟังก์ชันของกรีวังค์ ซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัวไม่สามารถหาค่าที่เหมาะที่สุดพบ

3.6 สรุป

ซีมูเลตเต็ดอันนีลลิ่งเป็นอัลกอริธึมที่จำลองกระบวนการทางฟิสิกส์ เพื่อใช้ในกระบวนการในการหาค่าเหมาะที่สุดแบบคอมบินาโทเรียล ที่เริ่มจากการสุ่มหาคำตอบเบื้องต้นขึ้นมา จากนั้นวิธีการของซีมูเลตเต็ดอันนีลลิ่ง จะทำการค้นหาคำตอบใหม่ที่ดีกว่าเพื่อใช้แทนคำตอบเดิม และทำการพัฒนาคำตอบของระบบให้ดีขึ้น โดยที่พิจารณาว่าถ้าคำตอบที่หาได้ใหม่ดีกว่าคำตอบปัจจุบัน ให้ใช้คำตอบใหม่แทนที่คำตอบปัจจุบัน แต่ถ้าคำตอบใหม่ไม่ดีกว่า ให้พิจารณาจากความน่าจะเป็นของโบลซ์แมนน์ (p) โดยที่มีค่าของ ΔE และ T เป็นปัจจัยสำคัญ โดยเฉพาะค่า T ที่ในช่วงแรกของการพิจารณาเปรียบเทียบ จะมีค่า T สูง ซึ่งทำให้ระบบสามารถยอมรับคำตอบที่แย่กว่ามาเป็นคำตอบปัจจุบันได้ แต่เมื่อการพิจารณาไปถึงรอบที่มากขึ้นค่า T จะถูกกำหนดให้มีค่าน้อยลง เพื่อจำกัดการพิจารณาให้การค้นหามุ่งเป้าไปที่คำตอบที่ดีกว่าเสมอ ซึ่งขั้นตอนของการทำงานของอัลกอริธึมจะประกอบด้วย

ขั้นตอนการวนรอบ ในแต่ละขั้นของการทำงาน จะพิจารณาเปรียบเทียบคำตอบข้างเคียง s' ของคำตอบปัจจุบัน s ด้วยค่าความน่าจะเป็น กระบวนการนี้จะถูกทำซ้ำจนกว่าจะได้คำตอบที่ดี หรือจนกว่าจะยุติการทำงานของคอมพิวเตอร์

การสร้างคำตอบข้างเคียง เป็นสถานะใหม่ของปัญหาที่ถูกสร้างขึ้นมาด้วยการปรับเปลี่ยนสถานะปัจจุบัน หรือคำตอบปัจจุบันด้วยวิธีการบางอย่าง

การยอมรับความน่าจะเป็น เป็นกระบวนการที่อัลกอริธึมของเมโทรโปลิส พิจารณาว่าจะยอมรับสถานะข้างเคียง s' ที่สร้างขึ้นใหม่ที่แย่กว่า s เป็นสถานะปัจจุบันหรือไม่ ซึ่งก็คือการเดินย้อนขึ้นไปในจุดที่แย่กว่า หรือการเดินลงเขา (Downhill step) ซึ่งมีความสัมพันธ์โดยตรงกับค่าของ ΔE และค่า T โดยตรง และค่า T จะมีผลอย่างมากต่อการยอมรับความน่าจะเป็น คือเมื่อ T มีค่ามาก โอกาสที่สถานะใหม่ที่อยู่ข้างเคียงจะได้รับการยอมรับมีสูง แต่เมื่อค่าของ T □ 0 สถานะใหม่ทีแย่กว่าจะถูกปฏิเสธเสมอ

การลู่เข้าสู่คำตอบ (Convergence) ในช่วงแรกของการค้นหาคำตอบให้ค่า T มีค่าสูง และมีโอกาสที่สถานะใหม่ที่แย่กว่าจะได้รับการยอมรับสูง ซึ่งถือเป็นช่วงของเอ็กซ์พลอเรชัน เมื่อเราลดอุณหภูมิลงเรื่อยๆ ค่า T จะน้อยลงจนเหลือ 0 ทำให้โอกาสของสถานะที่มีค่าแย่กว่าจะได้รับการยอมรับน้อยมาก กระบวนการในช่วงนี้จะเรียกว่า เอ็กซ์พล้อยเตชัน

ตารางการอบอ่อน (Annealing schedule) เป็นการกำหนดวิธีการเปลี่ยนแปลงอุณหภูมิ T เป็นเงื่อนไขสำคัญของการอบอ่อน เพื่อกำหนดผลต่อวิธีการของการค้นหาคำตอบ ในการกำหนดตารางของการอบอ่อนนั้น ให้เริ่มต้นเมื่อ T มีค่าสูง จากนั้นก็ลดอุณหภูมิลงมาเรื่อยๆ ในช่วงของการลดอุณหภูมิ มีวิธีการที่สำคัญคือ แบบเชิงเส้น แบบเอ็กซ์โพเนนเชียล และแบบลอกาลิธึม

ดังที่ได้กล่าวมาแล้วว่า ซีมูเลตเต็ด อันนีลลิ่งเป็นอัลกอริธึมที่จำลองกระบวนการทางฟิสิกส์ ที่มีศาสตร์มาจากกลศาสตร์สถิติ ซึ่งกระบวนการทั้งสองนี้มีความเหมือนและต่าง มีข้อดีข้อเสียดังการเปรียบเทียบดังต่อไปนี้

การเปรียบเทียบระหว่างกลศาสตร์สถิติกับการคำนวณด้วยซีมูเลตเต็ด อันนีลลิ่ง

ฟิสิกส์	ซีมูเลตเต็ด อันนีลลิ่ง
สถานะของอะตอม พลังงาน	คำตอบที่เป็นไปได้ของปัญหา ค่าที่ได้จากฮิวริสติก ฟังก์ชัน

สถานะสมดุล	โลคอลมินิมั่ม
สถานะพื้นฐาน	โกลบอลมินิมั่ม
อุณหภูมิ	พารามิเตอร์ควบคุม
การอบอ่อน	การค้นหาด้วยการลดค่า T
การกระจายโบล์สแมนน์-กิบบ์ส	ความน่าจะเป็นในการเลือกคำตอบข้างเคียง

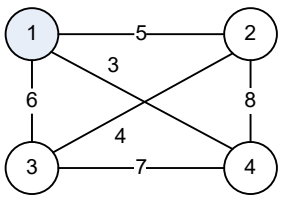
ข้อดี	ข้อไม่ดี
☐ ซีมูเลตเต็ด อันนีลลิ่งมีความสามารถในการหลีกเลี่ยงการติดอยู่ในโลคอลมินิมั่ม ☐ ซีมูเลตเต็ด อันนีลลิ่ง รับประกันว่าการลู่เข้าสู่คำตอบจะให้ค่าที่ดีได้ ถ้ามีการวนหาคำตอบในจำนวนรอบที่มากพอ	☐ การกำหนดตารางการลดอุณหภูมิเป็นเรื่องยาก เนื่องจากการกำหนดจำนวนรอบของการค้นหาคำตอบ จะขึ้นอยู่กับการลดอุณหภูมิ ☐ การกำหนดอุณหภูมิเริ่มต้นนั้นยาก ถ้าเราเริ่มที่อุณหภูมิสูงไป ก็จะเป็นการเสียเวลาในการค้นหา ถ้ากำหนดต่ำไปคุณภาพของคำตอบที่ได้จะไม่ดี

ซีมูเลตเต็ด อันนีลลิ่ง เป็นเมตาฮิวริสติกที่เหมาะกับปัญหาที่เป็นคอมบินาทอเรียล การประยุกต์ใช้ซีมูเลตเต็ด อันนีลลิ่ง กับการหาค่าเหมาะที่สุดเชิงตัวเลขทำได้เช่นกัน ตัวอย่างเช่นซีมูเลตเต็ดอันนีลลิ่งแบบมอนเตคาร์โล (Vanderbilt, 1984) ซีมูเลตเต็ดอันนีลลิ่งสำหรับตัวแปรที่มีค่าต่อเนื่อง และซีมูเลตเต็ดอันนีลลิ่งแบบปรับตัว

ในการทดสอบสมรรถนะของอัลกอริธึมโดยใช้ฟังก์ชันเทียบเคียง การหาค่าเหมาะที่สุดสำหรับฟังก์ชันที่มีโลคอลมินิมั่มจำนวนมาก อัลกอริธึมซีมูเลตเต็ดอันนีลลิ่งยังทำได้ไม่ดีนัก

3.7 แบบฝึกหัด

1. ท่านเข้าใจวิธีการลดอุณหภูมิ T ของการอบอ่อนสัมพันธ์กับการค้นหาคำตอบในปริภูมิคำตอบอย่างไร
2. กลศาสตร์สถิติคืออะไร และสามารถนำมาใช้กับการหาค่าเหมาะที่สุดได้อย่างไร
3. ให้อธิบายขั้นตอนที่สำคัญของการค้นหาแบบซีมูเลตเต็ดอันนีลลิ่งดังต่อไปนี้ว่าทำได้อย่างไร
   3.1. การสร้างคำตอบเริ่มต้น
   3.2. การสร้างคำตอบข้างเคียง
   3.3. การเลือกคำตอบสำหรับสถานะใหม่
4. อธิบายบทบาทของเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชันของซีมูเลตเต็ดอันนีลลิ่งตามหัวข้อต่อไปนี้
   4.1. ความหมายของเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน
   4.2. เอ็กซ์พลอเรชันเกิดขึ้นช่วงใดของการหาคำตอบ
   4.3. เอ็กซ์พล้อยเตชันเกิดขึ้นช่วงใดของการหาคำตอบ
   4.4. การสร้างสมดุลของเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน มีประโยชน์อย่างไร
5. ให้อธิบายตารางอบอ่อนในเรื่องดังต่อไปนี้
   5.1. ตารางอบอ่อนคืออะไร
   5.2. มีบทบาทต่อการหาค่าที่เหมาะที่สุดอย่างไร
   5.3. ยกตัวอย่างของตารางอบอ่อนมาอย่างน้อย 3 ชนิด พร้อมทั้งอธิบายความแตกต่างของแต่ละตาราง
6. อธิบายว่าอัลกอริธึมซีมูเลตเต็ดอันนีลลิ่งเป็นเซลฟ์ออกาไนเซชันอย่างไร
7. จากปัญหาการเดินทางของเซลส์แมน ตามแสดงดังรูปข้างล่าง ให้แก้ปัญหานี้โดยการใช้ตารางอบอ่อนแบบเส้นตรง $T(t) = T_0 - \eta t$

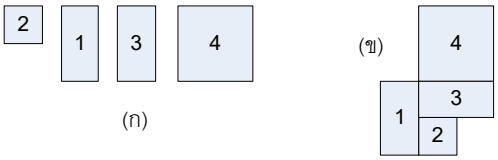


7.1. หาคำตอบของปัญหา

7.2. เขียนกราฟของการลู่เข้าสู่คำตอบ

7.3. เปรียบเทียบกราฟของการลู่เข้าสู่คำตอบที่ได้ใหม่กับกราฟที่ได้จากตัวอย่างที่มีอยู่ในหนังสือว่ามีความเหมือนหรือต่างกันอย่างไร เพราะอะไร

8. จากปัญหาการจัดวางแพทเทิร์นดังแสดงตามรูปข้างล่าง ให้ใช้ตารางอบอ่อนตามสมการ $T(t) = T_0/t$



8.1. หาคำตอบของปัญหาใหม่

8.2. เขียนกราฟของการลู่เข้าสู่คำตอบใหม่

8.3. เปรียบเทียบกราฟของการลู่เข้าสู่คำตอบที่ได้ใหม่กับกราฟที่ได้จากตัวอย่างว่ามีความเหมือนหรือต่างกันอย่างไร เพราะอะไร

9. เขียนโปรแกรมซีมูเลตเต็ดอันนีลลิ่งเพื่อหาค่าเหมาะสุดของสมการดังต่อไปนี้ พร้อมแสดงกราฟของการลู่เข้าสู่คำตอบ

9.1. $f(x) = x_1^2 + x_2^2$

9.2. $f(x) = An + \sum_{i=1}^{n} [x_i^2 - A\cos(2\pi x_i)]$

10. เขียนโปรแกรมซีมูเลตเต็ดอันนีลลิ่งเพื่อหาค่าเหมาะสุดของสมการ $f(x) = x_1^2 + x_2^2$ แล้วทดลองปรับค่าของตารางการอบอ่อน เพื่อหากราฟของการลู่เข้าสู่คำตอบดังต่อไปนี้

10.1. กำหนดการลดอุณหภูมิอย่างคงที่

10.2. การลดอุณหภูมิแบบเอ็กซ์โพเนนเชียล

10.3. การลดแบบลอการิธึมของฮอฟแมนน์

บทที่ 4อัลกอริธึมพันธุการ

Genetic Algorithm

4.1 คำนำ

อัลกอริธึมพันธุการ (Genetic Algorithm) หรือจีเอ (GA) เป็นอัลกอริธึมที่จำลองกระบวนการในการสืบทอดเผ่าพันธุ์ของสิ่งมีชีวิต ที่การผสมพันธุ์เกิดจากโครโมโซม 2 ตัวจับคู่เพื่อแลกเปลี่ยนยีน และการแลกเปลี่ยนยีนนี้ทำให้โครโมโซมซึ่งเป็นตัวกำหนดคุณลักษณะของสิ่งมีชีวิต มีการถ่ายทอดคุณสมบัติที่เป็นทั้งลักษณะเด่นและลักษณะด้อยซึ่งกันและกัน การวิวัฒนาการนี้เป็นไปแบบธรรมชาติ จากการสังเกตซากของฟอสซิลซี้ให้เห็นว่า วิวัฒนาการทางธรรมชาตินี้เป็นไปอย่างรวดเร็วมาก ใช้เวลาเพียงไม่กี่แสนปีเท่านั้น ที่ทำให้สิ่งมีชีวิตเซลล์เดียวสามารถพัฒนามาเป็นสิ่งมีชีวิตที่มีโครงสร้างสลับซับซ้อนมากๆ เช่น สัตว์ และคนได้ ผู้ที่นำกระบวนการนี้มาใช้กับการหาค่าเหมาะที่สุด (Optimization) คือ จอห์น ฮอลแลนด์ (Holland, 1975) และคณะ

อัลกอริธึมพันธุการเป็นการค้นหาแบบฮิวริสติกที่จำลองกระบวนการทำงานของกฎการคัดเลือกพันธุ์ทางธรรมชาติ การค้นหานี้เป็นการใช้ประโยชน์จากการเลือกสรรพันธุ์ ที่เป็นเสมือนกับการค้นหาแบบสุ่ม (Random) เพื่อหาค่าที่เหมาะที่สุด โดยการใช้ข้อมูลทางประวัติศาสตร์ เพื่อนำไปสู่การหาพันธุ์ใหม่ที่ดีขึ้น อัลกอริธึมนี้การจำลองกฎการวิวัฒนาการทางธรรมชาติของชาร์ลส์ ดาร์วิล (Charles Darwin) ที่ว่าด้วย “ความอยู่รอดของผู้ที่เหมาะสมที่สุด (Survival of the Fittest)” เพราะว่าในธรรมชาติที่ทรัพยากรมีอยู่อย่างจำกัด สิ่งมีชีวิตที่ปรับตัวให้ได้ดีที่สุดเท่านั้นจึงจะอยู่รอดได้ ในการปรับตัวดังกล่าว สิ่งมีชีวิตใช้กระบวนการเลือกสรร เพื่อการปรับตัวเองให้เหมาะสมกับสภาพแวดล้อมนั้น การเลือกสรรนี้เปรียบเสมือนกระบวนการในการค้นหาสิ่งที่เหมาะที่สุด (Optimization) วิธีหนึ่ง

4.2 หลักการของอัลกอริธึมพันธุการ

อัลกอริธึมพันธุการจะหาผู้ที่เหมาะสมที่สุดในระหว่างการวิวัฒนาการ เพื่อใช้ในการแก้ปัญหาในแต่ละรุ่น (Generation) องค์ประกอบสำคัญของอัลกอริธึมนี้คือประชากรของสตริง (Strings) สตริง

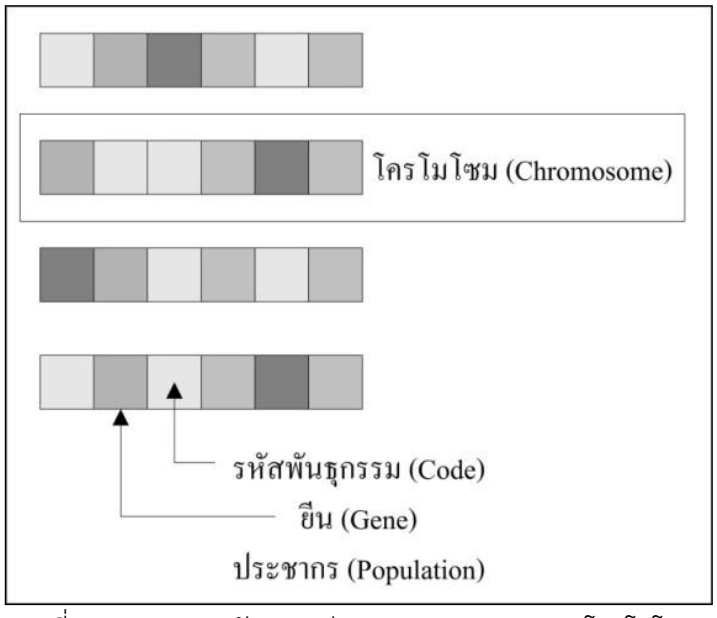
เหล่านี้เกิดจากการเรียงตัวของตัวแปรหรือบิต (bits) ต่างๆ ที่แสดงคุณสมบัติของสตริง ในอัลกอริธึมหนึ่งๆ จะมีจำนวนของสตริงจำนวนมากและมีคุณสมบัติที่หลากหลาย ที่เรียกว่าประชากร ซึ่งการเรียงตัวของตัวแปรเป็นสตริงนี้เป็นการเลียนแบบวิธีการของโครโมโซมที่มียีนในการแสดงคุณสมบัติ สำหรับสมาชิกหรือสตริงแต่ละตัวในประชากรนี้จะแสดงถึงสถานะ (State) หรือคำตอบที่เป็นไปได้ในปริภูมิปัญหา (Problem Space) และสมาชิกแต่ละตัวของประชากรเหล่านี้จะต้องผ่านกระบวนการของการวิวัฒนาการ สตริงที่เป็นตัวแทนของประชากรในอัลกอริธึมพันธุการจะเลียนแบบการจัดโครงสร้างของพันธุการ และเลียนแบบพฤติกรรมของโครโมโซม โดยมีวิธีการพื้นฐานดังนี้

- สตริงแต่ละตัวจะแข่งขันกันเพื่อชิงทรัพยากรและการจับคู่
- สตริงที่ชนะในการแข่งขันเหล่านี้จะมีโอกาสสร้างลูกหลานมากกว่าสตริงที่แพ้
- รหัสพันธุกรรมที่ดีของสตริงจะถูกถ่ายทอดไปยังประชากรในรุ่นต่อไป ซึ่งเป็นไปได้ที่สตริงที่อยู่ในรุ่นลูกจะมีคุณสมบัติดีกว่ารุ่นพ่อแม่
- ประชากรในรุ่นหลังจะถูกพัฒนาให้มีคุณสมบัติที่เหมาะสมกับสภาพแวดล้อมมากขึ้น

สตริงแต่ละตัวของประชากรในปริภูมิปัญหา (Problem Space) จะถูกเขียนไว้ในรูปของเวกเตอร์ (Vector) ของตัวแปร (หรือบิต) ที่มีความยาวจำกัด โดยปกติแล้วตัวแปรเหล่านี้จะนิยมแสดงค่าเป็น [0,1] ตัวอย่างเช่น 10010110 เป็นสตริงที่อยู่ในรูปของเวกเตอร์ขนาดความยาวของตัวแปรเท่ากับ 8 หลัก สำหรับตัวแปรในแต่ละหลักของสตริงนี้จะเป็นตัวแสดงคุณสมบัติที่เปรียบเสมือนยีน (Gene) ของโครโมโซม นั่นก็หมายความว่าโครโมโซมจะประกอบด้วยยีนหรือตัวแปรหลายๆ ตัวเหล่านี้ประกอบกัน โดยสตริงแต่ละตัวจะอาศัยค่าของตัวแปรเหล่านี้แสดงถึงระดับของความสามารถในการแข่งขันเพื่อความอยู่รอดที่เรียกว่า *ฟิตเน็ส (Fitness)* และสตริงที่มีค่าฟิตเน็สสูงจะถูกเลือกสรรให้อยู่หรือสืบพันธุ์ต่อในรุ่นถัดไป โดยอาศัยกระบวนการ “การเลือกสรร (Selection)” สำหรับอัลกอริธึมพันธุการจะมีความโน้มเอียงที่โครโมโซมลูกหลานจะดีกว่าตัวพ่อแม่ เนื่องจากวิธีนี้เป็นการรวมยีนที่มีคุณสมบัติดีของโครโมโซมพ่อและแม่

จาก รูปที่ 4.1 แสดงลักษณะของโครโมโซมที่ประกอบด้วยยีน (Gene) ยีนแต่ละตัวจะมีรหัสพันธุกรรมของยีนเป็นตัวบอกลักษณะของโครโมโซม การรวมกันของโครโมโซมจำนวนมากจะเรียกว่า

ประชากร (Population) ในอัลกอริธึมพันธุการ จำนวนประชากรจะถูกคงไว้ที่จำนวนคงที่ N ด้วยการใช้ค่าฟิตเน็ส (Fitness) เป็นตัวคัดเลือก พ่อและแม่จะถูกเลือกให้จับคู่ด้วยค่าของฟิตเน็สนี้เพื่อสร้างลูกหลานด้วยการครอสโอเวอร์ (Crossover) กล่าวคือ พ่อแม่ที่มีค่าฟิตเน็สสูงจะมีโอกาสสูงที่ถูกเลือกมาทำครอสโอเวอร์ และลูกๆ ที่ถูกสร้างออกมาจะได้รับการถ่ายทอดคุณสมบัติจากพ่อและแม่ ในขณะที่มีการจับคู่กันของพ่อและแม่เพื่อสร้างลูกใหม่ ในระหว่างการดำเนินงานของอัลกอริธึม จำนวนประชากรบางส่วนที่มีค่าฟิตเน็สต่ำจะต้องถูกคัดออกเพื่อเว้นที่ไว้สำหรับลูกที่เกิดใหม่ ซึ่งในที่สุดก็จะทำให้เกิดประชากรรุ่นใหม่ ด้วยวิธีการเช่นนี้ เมื่อประชากรของสตริงผ่านการคัดเลือกและการครอสโอเวอร์ด้วยจำนวนครั้งที่มากเพียงพอ จะทำให้สตริงส่วนที่แข็งแรงจะอยู่รอดเพื่อสร้างลูก และส่วนที่อ่อนแอจะถูกคัดออกไป จนกระทั่งได้ประชากรที่ดีเหลืออยู่



รูปที่ 4.1 การแสดงลักษณะต่างๆ ของประชากรของโครโมโซม

สำหรับสตริงในรุ่นใหม่ โดยเฉลี่ยแล้วจะมีคุณสมบัติที่ดีกว่าสตริงในรุ่นก่อน และผู้สืบทอดในแต่ละรุ่นจะประกอบด้วยค่าฟิตเน็สของยีนที่ดีกว่าเสมอ จากนั้นกระบวนการเลือกสรรจะทำการคัดเลือกสตริงหรือโครโมโซม สำหรับการจับคู่เพื่อการสร้างลูกหลานรุ่นต่อไป วิวัฒนาการจะเป็นเช่นนี้ไปเรื่อยๆ

จนกระทั่งความแตกต่างของคุณสมบัติที่ดีของประชากรรุ่นใหม่กับประชากรรุ่นเดิมไม่มีความแตกต่างกันมาก เมื่อนั้นก็หมายความว่าการวิวัฒนาการมาถึงจุดสุดยอดแล้ว และการแก้ปัญหาก็จะสิ้นสุดลง

4.2.1 อัลกอริธึมพันธุการ

อัลกอริธึมพันธุการมีขั้นตอนในการแก้ปัญหา 4 ขั้นตอนคือ

1. สร้างประชากรเริ่มต้นจากการสุ่มตัวอย่าง
2. คำนวณค่าฟิตเน็สสำหรับประชากรแต่ละตัว
3. สร้างประชากรใหม่ด้วยวิธีการดังต่อไปนี้
   3.1 **การเลือกสรร (Selection)** เป็นการสำเนาหรือการคัดสรรสตริงที่ดีจำนวนหนึ่ง
   3.2 **ครอสโอเวอร์ (Crossover)** เป็นกระบวนการของการจับคู่ของสตริงที่ดีจำนวนหนึ่ง
   3.3 **มิวเตชัน (Mutation)** เป็นกระบวนการปรับปรุงค่าของตัวแปรหรือบิตบางตัวในสตริงบางสตริงแบบสุ่ม
4. สตริงที่ดีที่สุดหลังจากการทำงานของอัลกอริธึมของประชากรหลายรุ่นจะเป็นคำตอบของปัญหา (Koza, 1992)

4.2.2 ตัวดำเนินการของอัลกอริธึม

การทำงานของอัลกอรธึมพันธุการจะประกอบด้วยตัวดำเนินการที่สำคัญดังต่อไปนี้

ตัวดำเนินการเลือกสรร (Selection Operator) เป็นฟังก์ชันที่ทำหน้าที่ในการเลือกสตริงโดยที่ระบบจะเปิดโอกาสให้สตริงหรือโครโมโซมที่มีคุณสมบัติที่ดี มีโอกาสส่งต่อยีนหรือตัวแปรของมันไปยังรุ่นต่อไป การพิจารณาคุณสมบัติของสตริงจะดูจากค่าฟิตเน็ส ซึ่งค่านี้หาได้จากอ็อบเจ็กทีฟฟังก์ชัน (Objective Function) หรือฟังก์ชันการประเมิน (Evaluation Function)

ตัวดำเนินการครอสโอเวอร์ (Crossover Operator) เป็นตัวดำเนินการหลัก ที่แสดงความแตกต่างของอัลกอริธึมพันธุการกับเทคนิคของการหาค่าเหมาะที่สุด (Optimization) แบบอื่นๆ วิธีการก็คือ สตริงจะถูกเลือกออกมาหนึ่งคู่จากประชากรที่มีอยู่โดยใช้ตัวดำเนินการเลือกสรร จากนั้นหาตำแหน่งของบิต (Bit) ของตัวแปรที่จะทำการครอสโอเวอร์ด้วยวิธีการสุ่ม แยกสตริงออกเป็นสองส่วนตาม

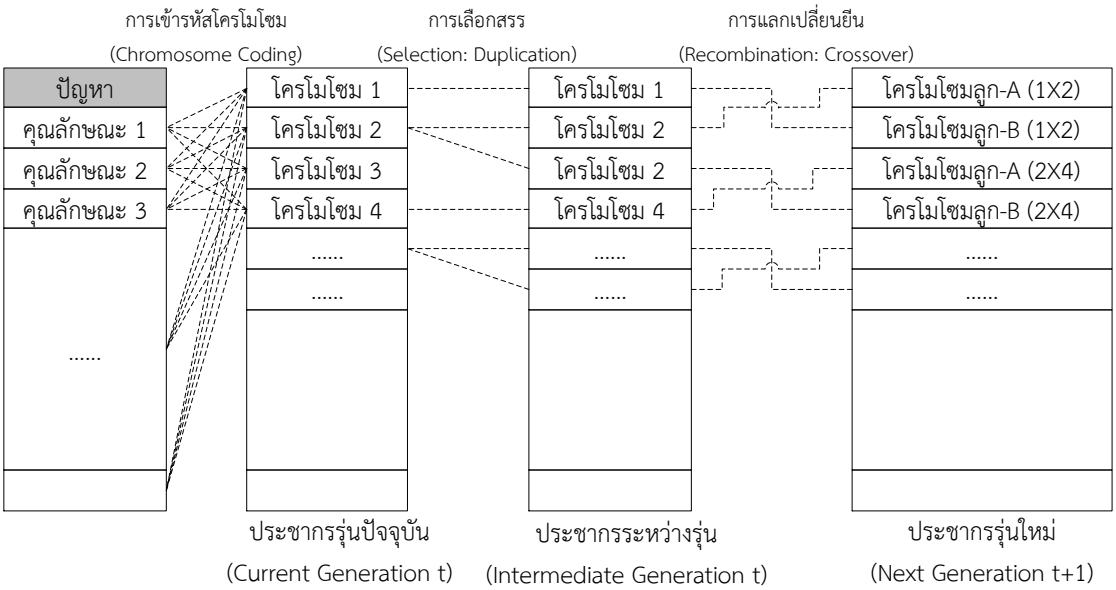
ตำแหน่งของบิตที่สุ่มขึ้นมาของสตริงทั้งสอง แล้วทำการจับคู่ใหม่แบบสลับกัน ลูกที่เกิดขึ้นใหม่และมีคุณสมบัติที่ดีกว่าจะถูกส่งต่อไปเป็นประชากรในรุ่นต่อไป ด้วยวิธีการนี้เองที่สามารถทำให้ประชากรในรุ่นใหม่มีคุณภาพดีขึ้น

ตัวดำเนินการมิวเตชัน (Mutation Operator) เป็นการดำเนินการที่มีเกิดขึ้นไม่บ่อยนัก แต่จะเกิดขึ้นเป็นบางครั้งบางคราวที่ตัวแปรบางตัวของสตริงหรือที่บิตใดบิตหนึ่งของสตริง โดยที่การเกิดขึ้นนี้เป็นไปแบบสุ่ม จุดประสงค์ของการเปลี่ยนแปลงนี้ก็เพื่อให้เกิดความหลากหลายของคุณสมบัติของสตริงขึ้นในหมู่ประชากร

4.2.3 วิธีการต่างๆ ของอัลกอริธึมพันธุการ

จากอัลกอริธึมดังกล่าวข้างต้น เราสามารถจำแนกขั้นตอนการทำงานของอัลกอริธึมพันธุการที่สำคัญมีอยู่ 3 ขั้นตามลำดับ ดังแสดงไว้ที่รูปที่ 4.2 คือ

- การเข้ารหัสโครโมโซม (Chromosome Coding)
- การเลือกสรรโครโมโซม (Selection)
- ขั้นตอนการแลกเปลี่ยนยีน (Recombination)



รูปที่ 4.2 ขั้นตอนการทำงานของอัลกอริธึมพันธุการใน 1 รุ่น (Whitley, 1994)

การทำงานเริ่มจาก การเข้ารหัสโครโมโซม (Chromosome Encoding) ซึ่งจะเป็นการบอกวิธีการกำหนดค่าของตัวแปรในสตริง และการออกแบบโครงสร้างของตัวแปรที่จะประกอบกันเป็นสตริง ในการกำหนดค่าของตัวแปรนั้น จะได้จากการนำปัญหามาวิเคราะห์หา คุณลักษณะ (Attributes) เพื่อนำมาสร้างเป็นตัวแปรของสตริงหรือโครโมโซมของปัญหานั้น และเช่นกันค่าที่บอกคุณลักษณะเหล่านี้โดยทั่วไปแล้วจะมีลักษณะที่สำคัญคือ เลขไบนารี ตัวเลข หรืออักขระ เป็นต้น สำหรับโครงสร้างของตัวแปรที่จะประกอบกันเป็นสตริงหรือโครโมโซมนั้น จะมีโครงสร้างที่ใช้กันโดยทั่วไป 2 โครงสร้างเท่านั้นคือ โครงสร้างเวกเตอร์และโครงสร้างต้นไม้

เมื่อเรากำหนดวิธีการเข้ารหัสได้แล้ว ขั้นตอนต่อไปคือกระบวนการในการสร้างประชากรเริ่มต้น เพื่อให้เป็นประชากรในรุ่นปัจจุบัน (Current Generation) ขั้นตอนนี้โดยทั่วไปแล้วจะใช้วิธีการสุ่มค่าให้กับตัวแปรทุกตัวในสตริงหรือโครโมโซมขึ้นมา การสุ่มตัวแปรขึ้นมา 1 ชุดจะเป็นประชากรเริ่มต้น 1 ตัว สำหรับจำนวนทั้งหมดจะเป็นกี่ชุดนั้นขึ้นอยู่กับขนาดของปัญหาที่ต้องการแก้

จากประชากรเริ่มต้นเหล่านี้ โครโมโซมจะถูกคัดด้วยกระบวนการที่เหมาะสมเพื่อเป็นตัวแทนในการสร้างประชากรในรุ่นต่อไป วิธีการของการเลือกสรรนี้มีหลายอย่าง แต่โดยหลักการคือการสร้าง

สำเนาของโครโมโซม โครโมโซมที่มีค่าฟิตเน็สสูงจะมีโอกาสถูกสำเนามากกว่าโครโมโซมที่มีค่าฟิตเน็สต่ำ จำนวนโครโมโซมทั้งหมดที่ถูกสำเนานี้เรียกว่า ประชากรระหว่างรุ่น (Intermediate Generation) ซึ่งประชากรระหว่างรุ่นกลุ่มนี้จะทำการแลกเปลี่ยนยีน (Recombination) ขึ้นเพื่อสร้างประชากรรุ่นใหม่ (Next Generation) ดังแสดงไว้ในรูปที่ 4.2 สำหรับการแลกเปลี่ยนยีนนี้มีวิธีการที่สำคัญอยู่ 2 วิธีคือ การครอสโอเวอร์ (Crossover) และการมิวเตชัน (Mutation) ซึ่งต่อไปเมื่อกล่าวถึงกระบวนการแลกเปลี่ยนยีน จะหมายถึงวิธีการทั้ง 2 นี้เท่านั้น

เนื่องจาก การเข้ารหัสโครโมโซมและการแลกเปลี่ยนยีนเป็นเรื่องที่สัมพันธ์กันโดยตรง ดังนั้นในตอนต่อไปเรื่องทั้งสองจะถูกกล่าวพร้อมกันภายใต้หัวข้อการเข้ารหัสโครโมโซมและการแลกเปลี่ยนยีน สำหรับการเลือกสรรโครโมโซมจะอธิบายแยกหัวข้อออกมา ดังมีรายละเอียดต่อไปนี้

4.3 การเลือกสรรโครโมโซม

จากรูปที่ 4.2 วิทเลย์ได้สรุปว่า จำนวนประชากรในรุ่นปัจจุบัน จำนวนหนึ่งจะถูกเลือกสรรให้เป็นประชากรระหว่างรุ่นก่อนที่จะทำการแลกเปลี่ยนยีน กระบวนการนี้เป็นการพิจารณาถึงโอกาสของโครโมโซมแต่ละตัวจะจับคู่กันเพื่อการแลกเปลี่ยนยีน วิธีการในการเลือกโครโมโซมจากประชากรรุ่นปัจจุบันให้เป็นประชากรระหว่างรุ่นนั้นมีหลายวิธีดังนี้

4.3.1 การเลือกสรรแบบวงล้อรูเล็ตต์

การเลือกสรรแบบวงล้อรูเล็ตต์ (Roulette Wheel Selection) นี้พิจารณาจากฟิตเน็สฟังก์ชัน (Fitness function) ที่คำนวณมาจาก f_i/f เมื่อ f_i คือค่าฟิตเน็สที่คำนวณได้จากโครโมโซม i หรือฟังก์ชันการประเมิน (Evaluation function) ของโครโมโซม i และ f คือค่าฟิตเน็สเฉลี่ยของ f_i ที่ได้จากการคำนวณของฟังก์ชันการประเมินของโครโมโซมทั้งหมด

จากตารางที่ 4.1 ในคอลัมน์แรกเป็นโครโมโซมแต่ละตัวที่มีค่าฟิตเน็ส ซึ่งแสดงไว้ในคอลัมน์ที่สอง สำหรับคอลัมน์ที่สามเป็นตัวเลขสุ่มที่เราสร้างขึ้นให้กับโครโมโซมแต่ละตัว สำหรับคอลัมน์สุดท้ายเป็นจำนวนโครโมโซมที่จะสร้างขึ้นของประชากรระหว่างรุ่นเพื่อการแลกเปลี่ยนยีนต่อไป จำนวนโครโมโซมของประชากรระหว่างรุ่นนี้ได้มาจากเลขจำนวนเต็มของค่าฟิตเน็สหารด้วยค่าฟิตเน็สเฉลี่ยตาม

คอลัมน์ที่สองของตารางที่ 4.1 ในกรณีที่ค่าฟิตเน็สนั้นมีค่าทศนิยมตามหลัง ให้นำค่าทศนิยมนั้นมารวมกับค่าสุ่มที่เราสร้างขึ้นในคอลัมน์ที่สาม ถ้ารวมกันแล้วมีค่ามากกว่า 1 ให้เพิ่มจำนวนโครโมโซมนั้นอีก 1 ตัว แต่ถ้ารวมกันแล้วไม่ถึง 1 ก็ไม่ต้องเพิ่ม จากนั้นเราก็จะได้โครโมโซมของประชากรระหว่างรุ่นตามตารางที่ 4.2 ในตารางนี้ โครโมโซมทุกตัวจะถูกกำกับหมายเลขไว้ จาก 1 ถึง S ตามตารางที่ 4.2 เมื่อ S ซึ่งเป็นจำนวนโครโมโซมทั้งหมดที่เลือกออกมา จากนั้นกระบวนการจับคู่ของโครโมโซมเพื่อการแลกเปลี่ยนยีนก็จะเกิดขึ้น ในการจับคู่ โครโมโซมแต่ละตัวจะถูกเลือกโดยการสุ่มตัวเลขตั้งแต่ 1 ถึง S ในการเลือกนี้เราก็จะเห็นได้ชัดเจนว่า โครโมโซมใดมีจำนวนมากกว่าโอกาสที่ถูกเลือกก็จะมากกว่าด้วย

โครโมโซม	ค่าฟิตเน็ส (f_i/f)	ค่าสุ่ม	จำนวนโครโมโซม
โครโมโซม 1	1.7	0.3	2
โครโมโซม 2	1.4	0.5	1
โครโมโซม 3	1.2	0.8	2
โครโมโซม 4	1		1
โครโมโซม 5	0.6	0.4	1
โครโมโซม 6	0.1	0.3	0

ตารางที่ 4.1 การแสดงวิธีการคำนวณหาจำนวนโครโมโซมจากค่าฟิตเน็สและค่าสุ่ม

โครโมโซม	เลขที่
โครโมโซม 1	1
โครโมโซม 1	2
โครโมโซม 2	3
โครโมโซม 3	4
โครโมโซม 3	5
โครโมโซม 4	6
โครโมโซม 5	7

ตารางที่ 4.2 จำนวนโครโมโซมของประชากรระหว่างรุ่นที่ถูกสร้างขึ้นตั้งแต่เลขที่ 1 ถึง S (S=7)

4.3.2 การเลือกสรรแบบจัดลำดับ

การเลือกสรรแบบจัดลำดับ (Rank Selection) เป็นอีกวิธีหนึ่งที่น่าสนใจ เพราะการเลือกสรรแบบวงล้อลูเล็ตต์นั้นจะมีปัญหาความลำเอียงเกิดขึ้น ถ้าค่าฟิตเน็สของโครโมโซมแต่ละตัวมีค่าแตกต่างกันมากๆ เช่น 4.0 กับ 0.4 ตามตารางที่ 4.3 เป็นการเปรียบเทียบวิธีการสร้างโครโมโซมระหว่าง การเลือกสรรแบบวงล้อลูเล็ตต์ (คอลัมน์ 4) และการเลือกสรรแบบจัดลำดับ (คอลัมน์ 6) ผลของความแตกต่างนี้จะทำให้โครโมโซมที่มีค่าฟิตเน็สต่ำ มีโอกาสได้รับการเลือกสรรเพื่อการแลกเปลี่ยนยีนน้อยมาก โอกาสที่จะเกิดการครอสโอเวอร์ของโครโมโซมเดียวกันก็จะมีสูง เพราะโครโมโซมที่มีค่าฟิตเน็สสูงจะเป็นประชากรส่วนใหญ่ของประชากรระหว่างรุ่น เป็นผลให้ไม่เกิดการพัฒนาคุณภาพของโครโมโซมในรุ่นถัดไป เมื่อเป็นเช่นนี้ การลู่เข้าสู่คำตอบก็จะไม่เกิดขึ้น เพื่อที่จะป้องกันปัญหาดังกล่าว วิธีการเลือกสรรแบบจัดลำดับจะนำโครโมโซมทั้งหมดมาเรียงจากน้อยไปหามาก แล้วจำนวนโครโมโซมที่ถูกเลือกไปเป็นโครโมโซมของประชากรระหว่างรุ่น มีจำนวนเท่ากับลำดับการเรียงดังแสดงไว้ในตารางที่ 4.3 คอลัมน์ 6

โครโมโซม	ค่าฟิตเน็ส (f_i/f)	ค่าสุ่ม	จำนวนโครโมโซม	ลำดับที่	จำนวนโครโมโซม
โครโมโซม 1	0.1	0.61	0	1	1
โครโมโซม 2	0.2	0.72	0	2	2
โครโมโซม 3	0.3	0.81	1	3	3
โครโมโซม 4	0.4	0.4.8	0	4	4
โครโมโซม 5	4.0		4	5	5

ตารางที่ 4.3 แสดงค่าการเลือกแบบจัดลำดับ

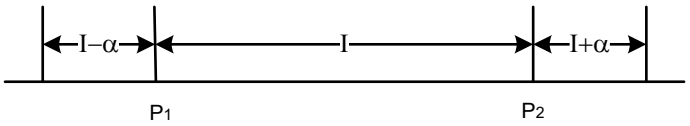
จากนั้นการจับคู่เพื่อการแลกเปลี่ยนยีนแบบสุ่มก็จะเกิดขึ้น จากการสังเกตผลของจำนวนโครโมโซมจะเห็นชัดว่า การเลือกสรรแบบจัดลำดับจะมีการกระจายของจำนวนโครโมโซมสม่ำเสมอกว่าแบบวงล้อลูเล็ตต์ที่มีเพียง โครโมโซม 3 จำนวน 1 ตัวและโครโมโซม 5 อีก 4 ตัว สำหรับโครโมโซมอื่นจะไม่มีโอกาสได้รับเลือกเพื่อการแลกเปลี่ยนยีนเลย แต่อย่างไรก็ตาม การเลือกโครโมโซมแบบจัดลำดับนี้ก็มีปัญหาเช่นกัน เพราะการเลือกสรรแบบนี้จะทำให้การลู่เข้าสู่คำตอบ (Convergence) ช้า เนื่องจากโครโมโซมที่มีค่าฟิตเน็สต่ำจะมีโอกาสในการแลกเปลี่ยนยีนมากขึ้น

4.3.3 การเลือกสรรแบบอื่น

นอกเหนือจากกระบวนการเลือกสรร 2 แบบดังกล่าวข้างต้น บางครั้งผู้ที่ออกแบบอัลกอริธึมพันธุการสามารถสร้างความลำเอียงให้กับการเลือกสรรได้เช่นกัน ถ้ากำหนดว่าโครโมโซมที่ดี มีค่าฟิตเน็สสูงจำนวนหนึ่งที่มีไม่มากนัก จะต้องได้รับการเลือกสรรเพื่อการแลกเปลี่ยนยีนเสมอเพื่อลดความเสี่ยงที่มาจากการสุ่มตัวอย่างแบบวงล้อรูเล็ตต์ และเช่นกัน ก็กำหนดไว้ว่าโครโมโซมที่ไม่ดีจำนวนหนึ่งก็จะถูกคัดออกเสมอเพื่อลดความเสี่ยงจากการนำโครโมโซมที่มีคุณลักษณะไม่ดีมาสู่กระบวนการแลกเปลี่ยนยีน การเลือกสรรที่มีการสร้างความลำเอียงในลักษณะนี้เรียกว่า **การเลือกแบบสเตดีสเตท (Steady-State Selection)**

ในขณะเดียวกัน การเลือกสรรแบบที่กำหนดความลำเอียงให้กับโครโมโซมที่ดีจำนวน 2-3 โครโมโซม เพื่อให้ได้รับการเลือกสรรให้ไปเป็นประชากรในรุ่นต่อไปโดยไม่ต้องผ่านการแลกเปลี่ยนยีน สำหรับโครโมโซมที่เหลือให้ผ่านกระบวนการตามปกติจะทำให้การค้นหาลู่เข้าหาคำตอบได้เร็วขึ้น เพราะประชากรที่มีค่าฟิตเน็สที่ดีที่สุดในแต่ละรุ่นจะถูกคัดให้เป็นประชากรของรุ่นใหม่เสมอ ทำให้ประชากรที่ดีจำนวนหนึ่งจะไม่ถูกคัดออก อันเนื่องมาจากการสุ่มและการแลกเปลี่ยนยีน การเลือกสรรที่มีการกำหนดความลำเอียงแบบนี้เรียกว่า **การเลือกแบบอีลีทิซึม (Elitism Selection)**

ในการคัดเลือกโครโมโซมที่ได้จากการครอสโอเวอร์ ยังมีอีกหลายวิธี (Wright, 1991) (Ono, 1996) แต่วิธีที่ได้มีการกล่าวถึงมาก และมีรายงานว่าสามารถประยุกต์ใช้ได้อย่างกว้างขวางคือ **บีแอลเอ็กซ์-แอลฟา(BLX-α)** (Eshelman, 1993) วิธีการนี้ได้กำหนดกรอบการเลือกโครโมโซมที่เกิดใหม่ไว้ดังรูปที่ 4.3



รูปที่ 4.3 ขอบเขตการเลือกสรรโครโมโซมใหม่แบบบีแอลเอ็กซ์-แอลฟา

วิธีการของบีแอลเอ็กซ์-แอลฟาคือ โครโมโซมใหม่ที่จะเป็นประชากรในรุ่นถัดไปจะเลือกจากโครโมโซมที่มีค่าในช่วงของ $[I-\alpha, I+\alpha]$ ตามรูปที่ 4.3 โดยที่ p_1 และ p_2 คือโครโมโซมพ่อและแม่ตามลำดับ

4.4 การเข้ารหัสโครโมโซมและการแลกเปลี่ยนยีน

การเข้ารหัสโครโมโซมเป็นหัวใจของการแก้ปัญหาด้วยอัลกอริธึมพันธุการ วิธีการคือการนำคุณลักษณะของปัญหามาใช้ในการเข้ารหัส สำหรับปัญหาหนึ่งๆ แล้ว จะประกอบด้วยคุณลักษณะที่มีผลต่อการแก้ปัญหาหลายอย่าง การมองคุณลักษณะของปัญหาในเชิงเปรียบเทียบก็จะเหมือนกับตัวแปรในสมการคณิตศาสตร์ การกำหนดค่าให้กับตัวแปรในสมการคณิตศาสตร์ก็เพื่อให้สมการคณิตศาสตร์สามารถแสดงผลออกมาได้ ในทำนองเดียวกัน การกำหนดค่าของคุณลักษณะของปัญหาจะเป็นผลลัพธ์ของการแก้ปัญหานั้น สำหรับวิธีการของอัลกอริธึมพันธุการ การเข้ารหัสถูกออกแบบให้อยู่ในรูปของสตริงที่ใช้แทนโครโมโซมหรือสตริง โครโมโซมนี้จะประกอบด้วยคุณลักษณะต่างๆ เรียงตัวกันเป็นเส้น ที่ทำหน้าที่เหมือนยีนในโครโมโซม ค่าของยีนในโครโมโซมหรือค่าของคุณลักษณะในสตริง จะเป็นตัวบอกคำตอบของโครโมโซม หรือของปัญหานั้น ในการเรียงตัวของคุณลักษณะในสตริง บางกรณีจะมีการเรียงตัวแบบโครงสร้างต้นไม้ก็ได้ สำหรับการแลกเปลี่ยนยีน (Recombination) เป็นการทำหน้าที่เสมือนการผสมพันธุ์เพื่อการสร้างประชากรชุดใหม่ให้กับประชากรเดิม การแลกเปลี่ยนยีนนั้นจะทำในสองลักษณะคือ การครอสโอเวอร์และมิวเตชัน (Mutation) เพื่อที่จะทำความเข้าในวิธีการเข้ารหัสและการแลกเปลี่ยนยีนของโครโมโซม ในหัวข้อต่อไปจะอธิบายวิธีการเข้ารหัสดังนี้

4.4.1 การเข้ารหัสแบบไบนารี

การเข้ารหัสแบบไบนารี (Binary Encoding) นี้เป็นที่ใช้กันมากที่สุด การเข้ารหัสแบบนี้จะเป็นการแทนค่าของปัญหาด้วยสตริงของตัวแปรชุดหนึ่งที่ตัวแปรในสตริงนั้นมีค่าเป็น 0 หรือ 1 ซึ่งค่าของตัวแปรนี้จะเป็นค่าที่แสดงคุณลักษณะของยีนแต่ละตัวในโครโมโซม ดังแสดงในรูปที่ 4.4 ดังนี้

โครโมโซม A	1100001110101001
โครโมโซม B	1111000010110011

รูปที่ 4.4 ตัวอย่างของโครโมโซมแบบไบนารี

การครอสโอเวอร์ ของโครโมโซมที่เข้ารหัสแบบไบนารี มีวิธีการคือ สมมุติว่า S1=000000 เป็นสตริงชุดแรก และ S2=111111 เป็นสตริงชุดที่สอง จากนั้นทำการสุ่มตำแหน่งของบิตที่จะทำการครอสโอเวอร์ สมมุติว่าได้เท่ากับ 2 ดังนั้นการแบ่งสตริงเพื่อการครอสโอเวอร์จะกลายเป็น

S1=[00]0000 และ

S2=[11]1111

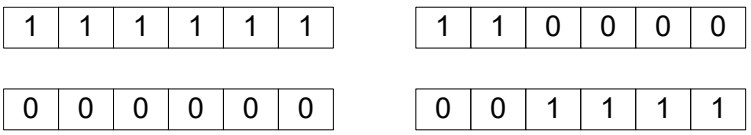
เมื่อทำการครอสโอเวอร์แล้ว จะได้สตริงในรุ่นลูกเป็น

S1’ = 110000 และ

S2’ = 001111

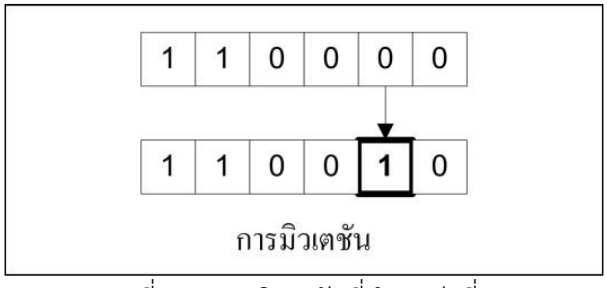
ลูกที่เกิดขึ้นใหม่ (S1′ และ S2′) จะถูกส่งต่อไปเป็นประชากรของรุ่นใหม่ และด้วยวิธีการนี้เองที่ทำให้ประชากรในรุ่นใหม่มีการปรับปรุงคุณสมบัติ รูปที่ 4.5 การครอสโอเวอร์เกิดที่ตำแหน่งที่ 2

การมิวเตชันของโครโมโซมที่เข้ารหัสแบบไบนารี เป็นการปรับปรุงคุณสมบัติของสตริงอีกวิธีหนึ่ง มิวเตชันจะเกิดขึ้นไม่บ่อยนัก เมื่อมีการทำครอสโอเวอร์ให้เกิดประชากรใหม่หลายๆ รุ่น การมิวเตชันจะเกิดขึ้นสักครั้งหนึ่ง โดยที่การมิวเตชันจะเกิดขึ้นแบบสุ่มที่บิตใดบิตหนึ่งของโครโมโซม บิตที่ถูกสุ่มให้เกิดการมิวเตชันจะถูกเปลี่ยนค่าของคุณลักษณะของบิต (หรือยีน) ซึ่งจะทำให้คุณลักษณะของโครโมโซมโดยรวมเปลี่ยนไปด้วย รูปที่ 4.6 การมิวเตชันที่ตำแหน่งที่ 5 ตัวอย่างเช่น ถ้า S1′=110000 เป็นสตริงในประชากรที่เราจะทำการมิวเตชัน เมื่อผ่านการมิวเตชันที่บิตที่ 5 แล้วผลที่ได้จะกลายเป็น S1″=110010



การครอสโอเวอร์

รูปที่ 4.5 การครอสโอเวอร์เกิดที่ตำแหน่งที่ 2



รูปที่ 4.6 การมิวเตชันที่ตำแหน่งที่ 5

4.4.2 การเข้ารหัสแบบเปอร์มิวเตชัน

การเข้ารหัสแบบเปอร์มิวเตชัน (Permutation Encoding) การเข้ารหัสแบบนี้จะเหมาะกับปัญหาการจัดลำดับ เช่นปัญหาการเดินทางของเซลส์แมน หรือปัญหาของการจัดตารางการทำงาน เป็นต้น ในการเข้ารหัสแบบนี้ ค่าของคุณลักษณะทุกตัวหรือบิตในโครโมโซมจะบอกลำดับ เช่น โครโมโซม 24351 อาจจะบอกว่า การเรียงลำดับของงานที่จะต้องทำคือ ให้ทำงานเรียงตามชิ้นงานหมายเลข 24351 เป็นต้น

โครโมโซม A	1 4 6 2 3 7 5 9 8
โครโมโซม B	9 1 2 8 3 7 4 6 5

รูปที่ 4.7 ตัวอย่างโครโมโซมแบบเปอร์มิวเตชัน

การครอสโอเวอร์ของโครโมโซมที่เข้ารหัสแบบเปอร์มิวเตชัน มีวิธีการเริ่มต้นเช่นเดียวกับวิธีแบบไบนารี ด้วยการหาตำแหน่งของยีนที่จะทำการครอสโอเวอร์ของคู่โครโมโซม เช่นที่ตำแหน่ง k จากนั้นยีนที่ตำแหน่ง 1 ถึง k ของโครโมโซมตัวแรกจะถูกยกมาเป็นยีนเริ่มต้นของโครโมโซมลูกตัวแรก สำหรับยีนส่วนที่เหลือให้สแกนจากยีนของโครโมโซมอีกตัวหนึ่งที่มีคุณลักษณะไม่ซ้ำกันมาเรียงต่อ ดังแสดงในรูปที่ 4.8

A	[1 4 6 2] 3 7 5 9 8		[1 4 6 2] 9 8 3 7 5
B	[9 1 2 8] 3 7 4 6 5		[9 1 2 8] 4 6 3 7 5

รูปที่ 4.8 การครอสโอเวอร์ของโครโมโซมที่เข้ารหัสแบบเปอร์มิวเตชัน

รูปที่ 4.8 การครอสโอเวอร์ของโครโมโซม A และ B ที่ตำแหน่งที่ k=4 ได้ลูกออกมา 2 ชุด (ซึ่งจะไปเป็นประชากรรุ่นใหม่) ชุดแรก เรานำ 4 ตัวแรกของโครโมโซม A คือ [1 4 6 2] มา สำหรับส่วนที่เหลือก็สแกนจากโครโมโซม B โดยเลือกเอาเฉพาะส่วนที่ไม่ซ้ำกับ 1 ถึง k ของโครโมโซม A มา ดังนี้ ตัวแรกของ โครโมโซม B คือ 9 ซึ่งไม่ซ้ำ นำมาต่อกับโครโมโซมลูกชุดแรกที่บิตที่ 5 ตัวที่สองและสามคือ 1 และ 2 ซ้ำให้เอาออก ตัวที่สี่ ห้า หก คือ 8 3 7 ไม่ซ้ำให้นำมาต่อหลัง 9 บิตที่ 7 และ 8 คือ 4 และ 6 จะซ้ำกับบิตที่ 1 ถึง k แรกของโครโมโซม A ให้นำออก สำหรับตัวสุดท้ายคือ 5 ซึ่งไม่ซ้ำ ให้นำมาต่อหลัง 7 ก็จะได้โครโมโซมลูกตัวแรกเป็น [1 4 6 2] 9 8 3 7 5 และเช่นกันสำหรับโครโมโซมลูกอีกตัวก็ทำเช่นเดียวกันโดยยก 1 ถึง k แรกของโครโมโซม B มาคือ [9 1 2 8] สำหรับส่วนที่เหลือให้สแกนจากโครโมโซม A จะได้ 4 6 3 7 5 ทำให้โครโมโซมลูกใหม่อีกตัวคือ [9 1 2 8] 4 6 3 7 5

การมิวเตชันของโครโมโซมที่เข้ารหัสแบบเปอร์มิวเตชัน จะเกิดจากการสลับตำแหน่งของลำดับดังแสดงในรูปที่ 4.9 เช่น กำหนดว่ามิวเตชันที่เกิดขึ้นโดยการสุ่มที่บิตที่ k=3 ของโครโมโซม [1 4 6 2 9 8 3 7 5] และค่าที่เปลี่ยนโดยการสุ่มเช่นกันจาก 6 เป็น 3 ดังนั้นการทำมิวเตชันนี้จะเป็นการสลับตำแหน่งกันของยีนตำแหน่งที่สามและเจ็ด โดยสลับเลข 6 ของตำแหน่งที่สามแทนที่เลข 3 ของบิตที่ 7 และเช่นกัน เลข 3 ของบิตที่ 7 จะย้ายมาแทนที่เลข 6 ของตำแหน่งที่สามดังรูปที่ 4.9 ดังนี้

1 4 6 2 9 8 3 7 5		1 4 **3** 2 9 8 **6** 7 5

รูปที่ 4.9 แสดงการเกิดมิวเตชัน

ตัวอย่างการใช้งานการเข้ารหัสแบบเปอร์มิวเตชันคือ การหาระยะทางที่สั้นที่สุดของเซลส์แมน โดยที่โครโมโซมแต่ละตัวจะบอกชื่อของเมืองที่เซลส์แมนจะต้องเดินทาง และตำแหน่งของบิตจะบอกลำดับของการเดินทาง ดังรูปที่ 4.9 สตริง 1 4 **3** 2 9 8 **6** 7 5 จะบอกลำดับการเดินทางของเมืองจากเมือง 1 ไปเมือง 4 ไปเมือง 3 ไปเมือง 2 ไปเมือง 9 เรื่อยๆ จนถึงเมือง 5 ซึ่งเป็นเมืองสุดท้าย แล้วกลับมาที่เมือง 1 เป็นอันสิ้นสุดการเดินทาง

4.4.3 การเข้ารหัสแบบค่า

การเข้ารหัสแบบค่า (Value encoding) นี้ ค่าของคุณลักษณะของยีนในโครโมโซมจะเป็นค่าที่สัมพันธ์กับปัญหา ซึ่งค่านี้เป็นได้ทั้งเลขจำนวนจริง ตัวเลขจำนวนเต็ม หรือ อักษร ที่แทนคุณลักษณะของโครโมโซม ดังรูปที่ 4.10 เช่น

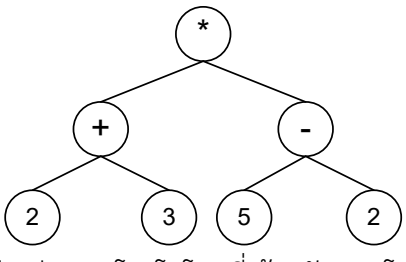
โครโมโซม A	1.1 1.3 1.6 1.0 1.2
โครโมโซม B	ASDFJKLQWUIJ
โครโมโซม C	(top), (bottom), (right), (left)

รูปที่ 4.10 ตัวอย่างของโครโมโซมแบบค่า

การเข้ารหัสแบบค่านี้ มีวิธีการการทำครอสโอเวร์และมิวเตชัน เช่นเดียวกับการเข้ารหัสแบบไบนารี โดยที่เทคนิคที่นำไปใช้กับการเข้ารหัสแบบไบนารีได้ก็สามารถนำมาใช้กับการเข้ารหัสแบบค่าได้เช่นเดียวกัน

4.4.4 การเข้ารหัสแบบโครงสร้างต้นไม้

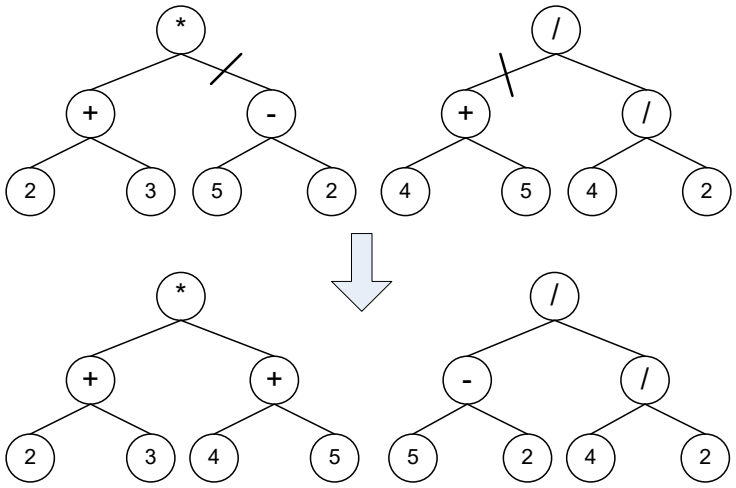
การเข้ารหัสแบบโครงสร้างต้นไม้ (Tree edcoding) เป็นรูปร่างของโครโมโซม ซึ่งจะต่างจากโครโมโซมโดยทั่วไปที่มีโครงสร้างแบบเชิงเส้นดังที่กล่าวมาแล้วข้างต้น การเข้ารหัสแบบนี้จะเหมาะกับโครโมโซมที่มีโครงสร้างที่โหนดปลายทาง (Terminal node) เป็นตัวแปรที่บ่งบอกคุณลักษณะ และมีโหนดภายใน (Internal node) เป็นตัวบอกความสัมพันธ์ของโหนดปลายทาง เช่นสมการคณิตศาสตร์ ที่จะประกอบด้วยโหนดภายในที่เป็นคำสั่ง โหนดภายในเหล่านี้แสดงความสัมพันธ์ของโหนดปลายทางที่เป็นตัวแปร ดังตัวอย่างสมการ (2+3)*(5-2) สามารถสร้างเป็นโครโมโซมดังแสดงไว้ในรูปที่ 4.11 เป็นต้น



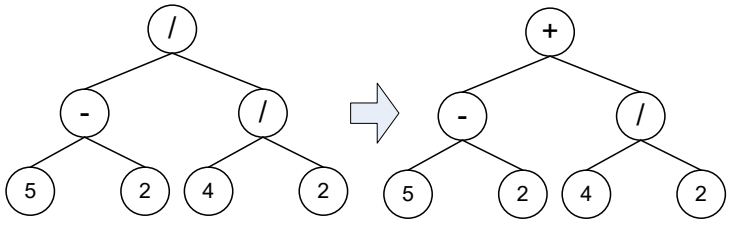
รูปที่ 4.11 ตัวอย่างของโครโมโซมที่เข้ารหัสแบบโครงสร้างต้นไม้

การครอสโอเวอร์ของโครโมโซมที่เข้ารหัสแบบโครงสร้างต้นไม้ จะเป็นการแลกเปลี่ยนกิ่งของต้นไม้ ดังตัวอย่างที่แสดงไว้ในรูปที่ 4.12 เป็นการครอสโอเวอร์ของโครโมโซมสองชุด คือ (2+3)*(5-2) กับ (4+5)/(4/2) กิ่งที่ทำการครอสโอเวอร์สำหรับโครโมโซมแรกคือ (5-2) และโครโมโซมชุดที่สองคือ (4+5) เมื่อทำการครอสโอเวอร์แล้ว ผลที่ได้จะเป็นไปตามโครงสร้างต้นไม้สองรูปล่างในรูปที่ 4.12 ผลของการครอสโอเวอร์เมื่อเขียนเป็นสมการคณิตศาสตร์ได้ดังนี้ (2+3)*(4+5) กับ (5-2)/(4/2)

มิวเตชันของโครโมโซมที่เข้ารหัสแบบโครงสร้างต้นไม้ เกิดที่โหนดภายใน (Internal node) เป็นส่วนใหญ่ ซึ่งในการแทนค่าสมการคณิตศาสตร์เป็นแบบโครงสร้างต้นไม้นั้น การเปลี่ยนแปลงฟังก์ชันจะเกิดที่ตัวดำเนินการคณิตศาสตร์ (Operator) มากกว่าการเกิดที่โหนดปลายทาง (Terminal node) ที่เป็นตัวแปร หรือเป็นค่าของตัวแปร ดังรูปที่ 4.13 มิวเตชันของโครโมโซมแบบโครงสร้างต้นไม้



รูปที่ 4.12 การครอสโอเวอร์ของโครงสร้างต้นไม้



รูปที่ 4.13 มิวเตชันของโครโมโซมแบบโครงสร้างต้นไม้

จากรูปที่ 4.13 การมิวเตชันเกิดขึ้นที่โหนดราก ทำให้สมการเปลี่ยนจาก (5-2)/(4/2) กลายเป็น (5-2)+(4/2) ตัวอย่างของปัญหาที่ใช้วิธีการของการเข้ารหัสแบบโครงสร้างต้นไม้ เช่น การหาฟังค์ชันจากค่าที่กำหนดให้ได้ค่าสูงที่สุด ตัวอย่างโครงสร้างต้นไม้ข้างต้น เป็นโครโมโซมที่แสดงสมการทางคณิตศาสตร์ มีโหนดปลายทางที่เป็นตัวเลข และโหนดภายในที่เป็นเครื่องหมายของฟังค์ชันต่างๆ ถ้าเราอยากให้สมการนี้ได้คำตอบออกมามากที่สุด ก็สามารถทำได้โดยการเปลี่ยนค่าฟังค์ชันต่างๆ ของโหนดภายใน โดยการคงค่าของโหนดปลายทางไว้

4.5 ตัวอย่างการหาค่าเหมาะที่สุดด้วยอัลกอริธึมพันธุการ

ในตอนนี้ จะเป็นการนำเสนอตัวอย่างการประยุกต์ใช้งานอัลกอริธึมพันธุการเพื่อทำให้เกิดความเข้าใจในการทำงานมากขึ้น ตัวอย่างแรกเป็นการหาค่าสูงสุดของเลขไบนารี เป็นตัวอย่างที่แสดงวิธีการทำงานของอัลกอริธึมพันธุการอย่างง่าย ตัวอย่างที่สองเป็นการประยุกต์ใช้อัลกอริธึมพันธุการในการแก้ปัญหาการเดินทางของเซลส์แมน ซึ่งเป็นปัญหาพื้นฐานที่สามารถนำไปประยุกต์ใช้งานต่อได้อีกหลายด้าน และตัวอย่างสุดท้ายเป็นการแก้ปัญหาเรื่องการวางผังงาน ซึ่งเป็นตัวอย่างของการใช้อัลกอริธึมในการแก้การออกแบบการจัดวางพื้นที่โรงงาน

4.5.1 การหาค่าสูงสุดของเลขไบนารี

กำหนดให้โครโมโซมที่มีสตริงขนาด 4 บิต แต่ละบิตมีค่าเป็นได้แค่ 0 และ 1 เท่านั้น ค่าของสตริงชุดนี้คำนวณจากเลขไบนารี 4 บิต ให้มีค่าเป็นจำนวนฐาน 10 ซึ่งค่านี้ เราเรียกว่าค่าการประเมิน (Evaluation value) สมมุติ เราไม่ทราบว่า โครโมโซมไบนารี 4 บิต ที่มีค่าสูงสุดมีค่าเท่าไร เราจะใช้อัลกอริธึมพันธุการสำหรับการหาโครโมโซม 4 บิตที่มีค่าสูงสุด โดยที่เรามีโครโมโซมเริ่มต้นจำนวน 4 ตัวที่ได้มาจากการสุ่ม ตามตารางที่ 4.4

จากตารางที่ 4.4 คอลัมน์แรกเป็นโครโมโซมที่ได้มาจากการสุ่ม คอลัมน์ที่ 2 เป็นค่าการประเมิน (Evaluation value) ของสตริงทั้ง 4 สำหรับคอลัมน์ที่ 3 เป็นค่าฟิตเน็ส (Fitness value) ของสตริงทั้ง 4 ที่คำนวณจาก f_i/f หรือค่าการประเมินของสตริงนั้น (f_i) หารด้วยค่าเฉลี่ยของค่าการประเมินของสตริงทุกตัว (f)

โครโมโซม	ค่า	Fitness	เลขสุ่ม	จำนวนโครโมโซม
1100	12	1.50	0.51	2
1010	10	1.25	0.62	1
0111	7	0.88	0.43	1
0011	3	0.38	0.50	0

ตารางที่ 4.4 แสดงค่าของโครโมโซมโดยใช้วิธีการเลือกสรรแบบวงล้อลูเล็ตต์

เพื่อสร้างสตริงของประชากรระหว่างรุ่น ในตัวอย่างนี้เราจะใช้วิธีการของวงล้อลูเล็ตต์ ด้วยการสร้างเลขสุ่มที่มีค่าระหว่าง 0-1 ตามคอลัมน์ที่ 4 ของตารางที่ 4.4 จากนั้นเราจะได้จำนวนโครโมโซมของประชากรระหว่างรุ่นตามคอลัมน์ที่ 5 ของตารางที่ 4.4

จากการจับคู่แบบวงล้อลูเล็ตต์ เราได้โครโมโซม 1 [1100] จับคู่เพื่อการแลกเปลี่ยนยีนกับโครโมโซม 2 [1010] และโครโมโซม 1 [1100] จับคู่เพื่อการแลกเปลี่ยนยีนกับโครโมโซม 3 [0111] ผลการจับคู่ของโครโมโซมเมื่อทำครอสโอเวอร์ที่ยีนตำแหน่งที่ 2 แล้วได้ผลตามตารางที่ 4.5 ดังนี้

โครโมโซม 1 x โครโมโซม 2	โครโมโซม 1 x โครโมโซม 3
1110	1111
1000	0100

ตารางที่ 4.5 แสดงผลการครอสโอเวร์ของโครโมโซมในรุ่นแรก

ผลจากการทำครอสโอเวอร์ตามที่แสดงไว้ในตารางที่ 4.5 เป็นประชากรในรุ่นที่สอง และประชากรเหล่านี้จะถูกนำมาเลือกสรรอีกครั้งด้วยวิธีการแบบวงล้อลูเล็ตต์ ผลของการคำนวณค่าการประเมิน ค่าฟิตเน็ส ค่าสุ่ม และจำนวนโครโมโซมที่จะใช้สำหรับการแลกเปลี่ยนยีน จะเป็นไปตามผลที่แสดงไว้ในตารางที่ 4.6 ดังนี้

โครโมโซม	ค่า	Fitness	เลขสุ่ม	จำนวนโครโมโซม
1111	15	1.46	0.55	2
1110	14	1.37	0.64	2

1000	8	0.78	0.30	1
0100	4	0.39	0.52	0

ตารางที่ 4.6 ประชากรในรุ่นที่สองและกระบวนการเลือกสรรโครโมโซมสำหรับการแลกเปลี่ยนยีน

โครโมโซม1 x โครโมโซม2	โครโมโซม1 x โครโมโซม3
1110	1100
1111	1011

ตารางที่ 4.7 การครอสโอเวอร์ของโครโมโซมรุ่นสอง

จากการจับคู่แบบสุ่มด้วยวงล้อลูเล็ตต์ ผลการทำครอสโอเวอร์ที่บิตที่สองของแต่ละตัวจะเป็นไปตามผลที่แสดงไว้ที่ตารางที่ 4.7 และผลการเลือกสรรโครโมโซมแบบวงล้อลูเล็ตต์จะได้ผลของการสร้างโครโมโซมสำหรับการแลกเปลี่ยนยีนเพื่อการสร้างประชากรในรุ่นสามตามตารางที่ 4.8

โครโมโซม	ค่า	Fitness	เลขสุ่ม	จำนวนโครโมโซม
1111	15	1.15	0.55	1
1110	14	1.08	0.44	1
1100	12	0.93	0.30	1
1011	11	0.84	0.72	1

ตารางที่ 4.8 ประชากรในรุ่นที่สองและกระบวนการเลือกสรรโครโมโซม

จากนั้นกระบวนการทำงานจะทำซ้ำกันไปจนกระทั่งได้คำตอบที่เราพอใจ กระบวนการมิวเตชันจะเกิดหลังจากการทำงานของครอสโอเวอร์ผ่านไปถึงรุ่นที่กำหนด การกำหนดนี้ส่วนใหญ่มาจากการสุ่มการมิวเตชันจะเกิดที่โครโมโซมตัวใดตัวหนึ่ง และที่บิตใดบิตหนึ่งของโครโมโซมนั้น เช่น ถ้าการสุ่มเลือกที่จะเกิดมิวเตชันของโครโมโซมตัวที่ 3 ให้เกิดมิวเตชันที่บิต 3 ผลที่ได้จะเกิดการเปลี่ยนแปลงจากเดิมที่เป็น 1100 จะกลายเป็น 1110 เป็นต้น เมื่อเกิดมิวเตชันแล้ว กระบวนการเลือกสรรก็จะเกิดขึ้นได้ และก็จะเริ่มกระบวนการสร้างประชากรใหม่

4.5.2 การเดินทางของเซลส์แมน

กำหนดว่า มีเมืองอยู่ 4 เมืองคือ A B C D ระยะทางระหว่างเมืองดังกล่าวได้แสดงไว้ในตารางที่ 4.9 แสดงระยะทางระหว่างเมือง มีเซลส์แมนคนหนึ่งที่จะต้องเดินทางไปขายของที่เมืองทั้ง 4 โดยเริ่มต้นที่เมืองใดเมืองหนึ่งก็ได้ และจะต้องเดินทางไปยังเมืองที่เหลือจนครบ เมื่อเดินครบทุกเมืองแล้วให้กลับมายังเมืองที่เริ่มต้นอีกครั้ง ให้หาเส้นทางที่สั้นที่สุดสำหรับการเดินทางของเซลส์แมนท่านนี้

	A	B	C	D
A	0	20	30	50
B		0	15	20
C			0	10
D				0

ตารางที่ 4.9 แสดงระยะทางระหว่างเมือง

การแก้ปัญหานี้ด้วยอัลกอริธึมพันธุการ จะเริ่มต้นด้วยการสุ่มลำดับของเมืองออกมา 3 เส้นทาง ดังแสดงไว้ในตารางที่ 4.10 คอลัมน์แรก คอลัมน์ที่สองเป็นค่าการประเมิน (Evaluation value) ของแต่ละโครโมโซม ด้วยวิธีการเลือกสรรแบบจัดลำดับ ซึ่งก็คือระยะทางรวมของแต่ละเส้นทาง ในคอลัมน์ที่สาม เป็นการจัดลำดับของโครโมโซมหรือเส้นทางทั้งสาม และคอลัมน์ที่สี่เป็นจำนวนโครโมโซมที่ใช้สำหรับการแลกเปลี่ยนยีน

โครโมโซม	ค่า	ลำดับ	จำนวนโครโมโซม
DCBAD	95	1	2
ABCDA	95	1	2
CBDAC	115	2	1

ตารางที่ 4.10 โครโมโซมเริ่มต้นและการเลือกสรรแบบจัดลำดับ

การครอสโอเวอร์ ด้วยวิธีการสุ่มตัวอย่าง การจับคู่ของโครโมโซมจะเป็นดังนี้คือ โครโมโซม 1 x โครโมโซม 2 ชุดที่สองเป็น โครโมโซม 1 x โครโมโซม 3 ดังแสดงไว้ที่แถวบนสุดของตารางที่ 4.11

สำหรับแถวที่สอง และแถวที่สามเป็นผลของการครอสโอเวอร์ ที่ทำให้เกิดประชากรในรุ่นใหม่ตามที่แสดงไว้ในตารางที่ 4.12

[DCBAD]x[ABCDA]	[DCBAD]x[CBDAC]
DCABD=80	DCBAD=95
ABDCA=80	CBDAC=115

ตารางที่ 4.11 การครอสโอเวอร์

โครโมโซม	ค่า	ลำดับ	จำนวนโครโมโซม
DCABD	80	1	3
ABDCA	80	1	3
DCBAD	95	2	2
CBDAC	115	3	1

ตารางที่ 4.12 การเลือกสรรแบบจัดลำดับ

สำหรับตารางที่ 4.12 การแสดงวิธีการเลือกสรรแบบจัดลำดับของประชากรในรุ่นสองเพื่อสร้างประชากรระหว่างรุ่น จำนวนโครโมโซมของเส้นทางที่เกิดใหม่นี้จะถูกนำไปจับคู่แบบสุ่มเพื่อการแลกเปลี่ยนยีนสำหรับการสร้างประชากรในรุ่นสาม การทำงานก็จะไปเริ่มที่การสร้างประชากรในรุ่นต่อไปเรื่อยๆ จนกระทั่งพบคำตอบที่พอใจ

มิวเตชัน การเกิดมิวเตชันของโครโมโซม เมื่อเกิดที่บิตใด ก็จะทำให้เกิดการสลับที่กับบิตที่เกี่ยวข้อง เช่น ถ้าหากการมิวเตชันเกิดที่โครโมโซมที่ 3 ที่บิตที่ 2 ของเส้นทาง DCBAD คือ C และ C สามารถจะเปลี่ยนเป็น B หรือ A ก็ได้ ถ้าเป็น B เมือง C และ B ก็จะสลับตำแหน่งกัน ผลที่ได้จะเป็น DBCAD แต่ถ้า C เป็น A โครโมโซมใหม่ที่ได้จะเป็น DABCD การเกิดมิวเตชันครั้งนี้ ถ้าบิตที่ 2 เปลี่ยนจาก C เป็น D จะทำให้ C กลายเป็นเมืองเริ่มต้น และสิ้นสุดการเดินทาง ดังนั้นโครโมโซมใหม่จะเป็น CDBAC

จากนั้นก็จะเกิดประชากรในรุ่นใหม่ กระบวนการแลกเปลี่ยนยีนก็จะเกิดขึ้นอีก กระบวนการดังกล่าวจะเกิดซ้ำจนกระทั่งได้โครโมโซมที่ดีที่สุด

การครอสโอเวอร์ เพื่อทำให้เข้าใจการทำงานของอัลกอริธึมพันธุการในการแก้ปัญหาการเดินทางของเซลส์แมนมากขึ้น ในตอนนี้จะนำเสนอวิธีการแก้ปัญหาการเดินทางของเซลส์แมนที่นำเสนอโดยอาเมด (Ahmed, 2010) ซึ่งมีวิธีการครอสโอเวอร์ของโครงโมโซมด้วยวิธีการซีเคว้นเชียล คอนสตรัคทีฟ ครอสโอเวอร์ (Sequential constructive crossover: SCX) ดังนี้

อัลกอริธึมของซีเควนเชียล คอนสตรัคทีฟ ครอสโอเวอร์

ขั้นตอนที่ 1: เริ่มจากโหนดที่ 1 และกำหนดให้ชื่อ p

ขั้นตอนที่ 2: หาโหนดที่เดินทางต่อจาก p ของโครโมโซมพ่อแม่ ที่ยังไม่ได้ไปเยี่ยมจาก จากหน้าไปหลัง ถ้าหากว่าโครโมโซมพ่อหรือแม่ ไม่มีเส้นทางที่เดินต่อจาก p ให้ใช้โหนดเรียงตามลำดับขั้นต่อไปนี้เป็นตัวเลือก (2,3,4,5,6,7) โดยเลือกโหนดที่ยังไม่ได้ถูกเลือก

ขั้นตอนที่ 3: ถ้ากำหนดว่า โหนด a และ b คือโหนดที่มาจากโครโมโซมพ่อและแม่ตามลำดับ หาระยะทาง distance(p,a) และ distance(p,b)

ขั้นตอนที่ 4: ถ้า distance(p,a) < distance(p,b) เป็นจริง ให้เลือก a ถ้าไม่ใช่ให้เลือก b เป็นโหนดที่จะต้องเดินทางต่อไป

ขั้นตอนที่ 5: ถ้าเดินครบทุกเมือง ให้หยุดทำงาน ถ้ายังไม่ครบทุกเมืองให้กลับไปที่ขั้นตอนที่ 2:

ถ้ากำหนดให้เส้นทางเดินของเมือง 7 เมือง มีระยะทางระหว่างเมืองตามตารางที่ 4.13 ซึ่งมีรายละเอียดดังต่อไปนี้

	1	2	3	4	5	6	7
1	0	70	85	91	45	63	28
2		0	60	46	28	30	42
3			0	18	76	40	30
4				0	20	80	18

5					0	41	60
6						0	32
7							0

ตารางที่ 4.13 ระยะทางระหว่างเมือง

และกำหนดว่า (1,5,7,3,6,4,2) มีระยะทางเท่ากับ 301 และ (1,6,2,4,3,5,7) มีระยะทางเท่ากับ 293 เป็นเส้นทางเริ่มต้น และให้เป็นโครโมโซมพ่อและแม่ ตามลำดับ เริ่มจากโหนดแรกของโครโมโซมพ่อและแม่ ซึ่งเท่ากับ (1) เลือกเมืองที่จะเดินถัดไปจากโครโมโซมทั้งสอง ซึ่งก็คือ 5 และ 6 โดยที่

distance(1,5) = 45 สำหรับ

distance(1,6) = 63

ดังนั้นเลือก 5 เนื่องจากระยะทางสั้นกว่า ดังนั้นเส้นทางที่ได้ในเบื้องต้นคือ (1,5) ต่อไปเลือกเมืองที่สามารถเดินจาก 5 ต่อไปคือเมือง 7 ทั้งจากของโครโมโซมพ่อและแม่ เลือกเมือง 7 เป็นเมืองถัดไป เราจะได้ (1,5,7) จาก 7 เมืองถัดไปของโครโมโซมพ่อคือ 3 ของโครโมโซมแม่ไม่มี ดังนั้นใช้ (2,3,4,5,6,7) ซึงเป็นข้อกำหนดจากอัลกอริธึมในขั้นตอนที่ 2 ได้เมือง 2 เป็นเมืองแรกที่ยังไม่ได้เดินผ่าน

distance(7,3) = 30

distance(7,2) = 42

เลือกเมือง 3 จะได้เมืองทั้งหมดเป็น (1,5,7,3) จากนั้นหาเส้นทางจาก 3 ไปเมืองที่ยังไม่ผ่านจากพ่อและแม่ ได้

distance(3,6) = 40

distance(3, (**2**,3,4,5,6,7)) = distance(3,**2) =** 60 (2 ยังไม่เคยถูกเลือก)

เลือกเมือง 6 จะได้เมืองทั้งหมดเป็น (1,5,7,3,6) จากนั้นหาเส้นทางจาก 6 ไปเมืองที่ยังไม่ผ่านจากพ่อและแม่ ได้

distance(6,4) = 80

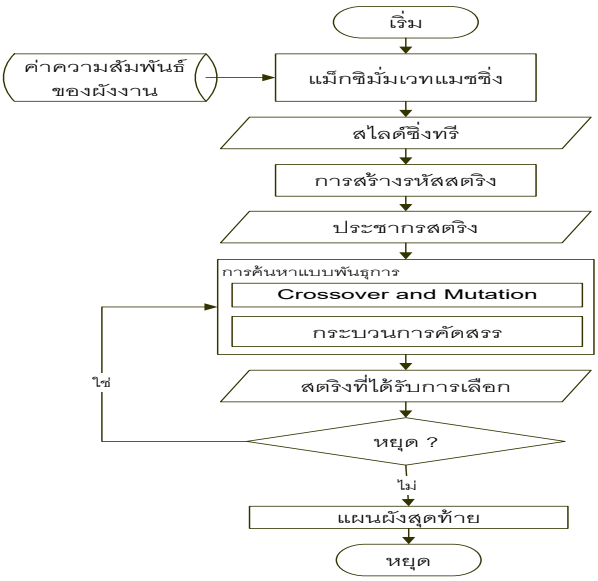
distance(6, (**2**,3,4,5,6,7)) = 30

เลือกเมือง 2 จะได้เมืองทั้งหมดเป็น (1,5,7,3,6,2) และเมืองสุดท้ายคือ 4 ดังนั้นเส้นทางทั้งหมดคือ (1,5,7,3,6,2,4) ระยะทางเท่ากับ 45+60+30+40+30+46 = 254 ซึ่งได้ระยะทางที่ดีขึ้น

สำหรับการใช้อัลกอริธึมพันธุการ เพื่อแก้ปัญหาการเดินทางของเซลส์แมนนั้นยังมีวิธีการอีกจำนวนมากเช่นของ มูนและคณะ (Moon, 2002) และไบร์แอนท์ (Bryant, 2000) เป็นต้น

4.5.3 การวางผังงานโดยอาศัยอัลกอริธึมพันธุการ

วิธีการนี้นำเสนอโดยบุญเจริญและธิติพงษ์ (Sirinaovakul, 2007) แบ่งขั้นตอนการวางผังเป็น 2 ขั้นตอนคือ การสร้างสไลซิ่งทรี (Slicing Tree) และอัลกอริธึมพันธุการ (Genetic Algorithm) การสร้างสไลซิ่งทรีจะอาศัยเทคนิคของคลัสเตอริ่งที่เรียกว่า *แม็กซิมั่มเวทแมชชิ่ง (Maximum Weight Matching)* ในการวางแผนผังเริ่มต้น (Initial Layout) จากนั้นระบบนำผลที่ได้จากการจัดวางนี้ไปเข้ากระบวนการค้นหาผังงานที่ดีที่สุดโดยอัลกอริธึมพันธุการ



รูปที่ 4.14 ผังงานของการทำงาน

ในกระบวนการของอัลกอริธึมพันธุการประกอบด้วย 2 ส่วนคือ การสร้างรหัสสตริง (String Coding) และการค้นหาแบบพันธุการ (Genetic Search) ในส่วนของการสร้างรหัสสตริง จะสร้างแผนผังเริ่มต้นที่ได้จากการใช้แม็กซิมั่มเวทแมชชิ่ง จากนั้นส่วนการค้นหาแบบพันธุการจะทำการเปลี่ยนรหัสสตริง และเลือกสรรสตริงที่ดีที่สุดออกมาเป็นผังงานสุดท้าย ดังแสดงไว้ในรูปที่ 4.14 ผังงานของการทำงาน

การสร้างสไลซิ่งทรี (Slicing tree construction) นี้ เราจะอาศัยเทคนิคของแม็กซิมั่มเวทแมชชิ่ง (Gabow, 1973) เพื่อช่วยในการจับคู่ ซึ่งสไลซิ่งทรีนี้จะทำหน้าที่เสมือนแผนผังเบื้องต้น เทคนิคของแม็กซิมั่มเวทแมชชิ่งคือการจับคู่หน่วยงานด้วยค่าความสัมพันธ์ เมื่อจับคู่กันได้ทั้งหมดแล้วจะต้องทำให้ผลรวมของค่าความสัมพันธ์ของคู่หน่วยงานทุกคู่มีค่าสูงสุด วิธีการทำเช่นนี้จะอาศัยเทคนิคของเวทกราฟ (Weight Graph) โดยให้โหนดของกราฟแทนหน่วยงานหรือการรวมกันของหน่วยงาน (แผนผังย่อย) ให้เส้นเชื่อมแสดงความเป็นไปได้ของการจับคู่ของหน่วยงานหรือแผนผังย่อย หรือจะกล่าวอีกนัยหนึ่งก็คือ โหนดที่มีเส้นเชื่อมถึงกันคือหน่วยงานหรือแผนผังย่อยที่สามารถจับคู่กันได้ และให้น้ำหนัก (Weight) ที่แสดงอยู่บนเส้นเชื่อมเป็นค่าของความสัมพันธ์ของหน่วยงาน ดังนั้นเราสามารถเขียนสมการความสัมพันธ์รวมออกมาได้ดังนี้

$$\text{Maximize } \alpha = \sum_{i,j=1}^{n} \omega_{i,j} x_{i,j} \qquad \text{มการ 4.1}$$

Subject to

$$x_{i,j} = \begin{cases} 0 & \text{for } 1 \le i \le n \text{ and } 1 \le j \le n \text{ if node i is not connected to node j,} \\ 0 & \text{for } 1 \le i \le n \text{ and } 1 \le j \le n \text{ if node i and node j are combined,} \\ 1 & \text{for } 1 \le i \le n \text{ and } 1 \le j \le n \text{ if node i is connected to node j.} \end{cases}$$

เมื่อ α คือผลรวมของค่าความสัมพันธ์ของคู่หน่วยงาน

$\omega_{i,j}$ คือความสัมพันธ์ของหน่วยงาน i และหน่วยงาน j

n คือจำนวนหน่วยงานหรือแผนผังย่อย

อัลกอริธึมการสร้างสไลซิ่งทรีเริ่มด้วยการใช้แม็กซิมั่มเวทแมชชิ่ง โดยการหาคู่ของหน่วยงานที่เมื่อจับคู่กันได้หมดแล้ว ผลรวมทั้งหมดของค่าความสัมพันธ์ของทุกคู่มีค่าสูงสุด ด้วยค่าความสัมพันธ์ของหน่วยงาน i และหน่วยงาน j หรือ $\omega_{i,j}$ ดังแสดงไว้ในสมการที่ 4.1

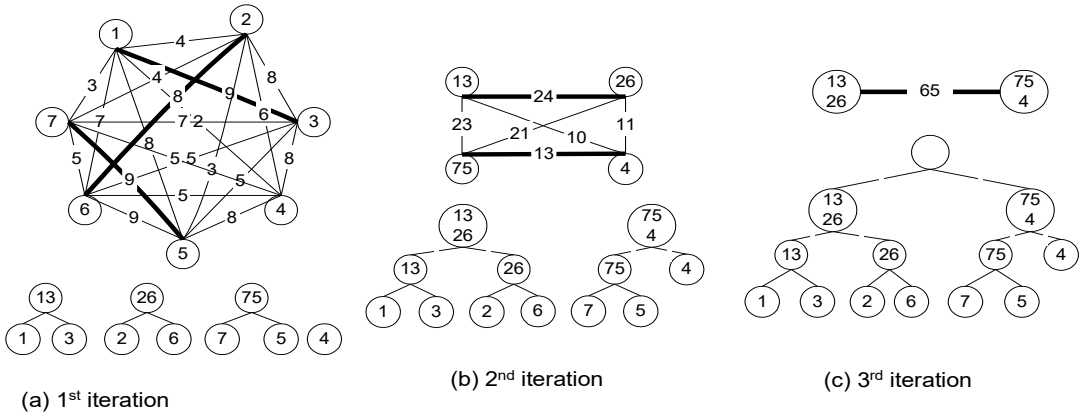
เมื่อได้แล้ว คู่ของหน่วยงานทั้งหมดจะถือว่าเป็นระดับล่างสุดของต้นไม้ ดังแสดงตาม

รูปที่ 4.15(a) เราจะได้การจับคู่ของหน่วยงาน 1 และ 3 หน่วยงาน 2 และ 6 และหน่วยงาน 7 และ 5 สำหรับหน่วยงาน 4 ไม่สามารถจับคู่กับใครได้ ให้เหลือไว้สำหรับการจับคู่ในรอบต่อไป จากนั้นให้นำผังงานที่จับคู่กันแล้ว กำหนดให้เป็นแผนผังย่อยสำหรับการพิจารณาเช่นเดียวกันอีก คือให้ทำการจับคู่เพื่อให้ได้แผนผังย่อยทั้งหมดที่มีค่ารวมของค่าความสัมพันธ์สูงสุด และการคำนวณผลรวมของค่าความสัมพันธ์ของแผนผังย่อยนี้จะเกิดจากความสัมพันธ์ของหน่วยงานแต่ละตัวรวมกัน

รูปที่ 4.15(b) แสดงผลลัพธ์ของการจับคู่ในระดับที่สอง ซึ่งผลที่ได้คือ แผนผังย่อย 1-3 จะจับคู่กับหน่วยงานย่อย 2-6 และแผนผังย่อย 7-5 จะจับคู่กับหน่วยงาน 4 ทำให้รวมกันเป็นแผนผังย่อยที่ใหญ่ขึ้นคือแผนผังย่อย 1-3-2-6 และแผนผังย่อย 7-5-4 จากนั้นทำการจับคู่อีก ทำเช่นนี้ไปเรื่อยๆ จะกระทั่งหน่วยงานทุกหน่วยงานถูกนำมารวมกันเป็นแผนผังเดียว ซึ่งก็หมายถึงการรวมของโหนดทั้งหมดมาอยู่ที่โหนดราก (Root Node) ตาม

รูปที่ 4.15(c) เป็นผลลัพธ์ของการรวมหน่วยงาน 7 ผังเป็นสไลซิ่งทรีจากสไลซิ่งทรีที่ได้เป็นผลลัพธ์สุดท้าย เราทราบว่าหน่วยงานควรจะจับคู่กันอย่างไร แต่ไม่ทราบว่าหน่วยงานที่จับคู่กันนี้จะวางอยู่อย่างไรของกันและกัน ดังนั้นเมื่อเทียบกับสไลซิ่งทรีใน

รูปที่ 4.15 แล้ว จะเห็นว่าที่หน่วยงาน นอกจากจะมีการจับคู่กันแล้ว โหนดที่อยู่ในระดับที่อยู่สูงขึ้นไปยังบอกตำแหน่งของการวางหน่วยงานด้วย ดังนั้นสไลซิ่งทรีตารางที่ 4.14(c) นี้จึงถูกเรียกว่าเอ็มทีสไลซิ่งทรี (Empty Slicing Tree)

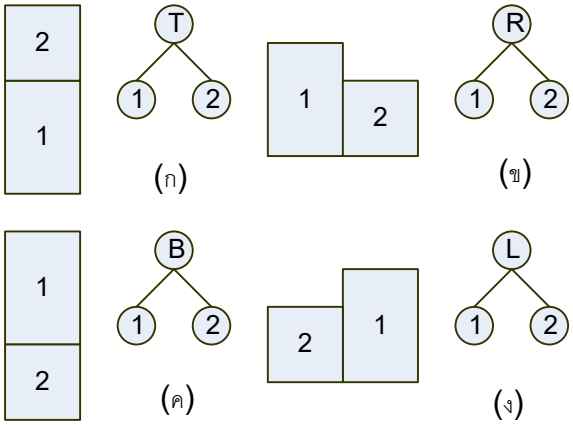


รูปที่ 4.15 แสดงการสร้างสไลซิ่งทรี

อัลกอริธึมพันธุการ (Genetic Algorithm) เป็นกระบวนการที่ให้คำตอบที่เหมาะสมที่สุดในการค้นหาคำตอบ (Goldberg, 1998) ในส่วนของอัลกอริธึมพันธุการที่นำมาใช้ในการแก้ปัญหานี้แบ่งออกได้เป็น 2 ส่วนคือ (1) ส่วนของการสร้างรหัส เป็นส่วนที่ทำหน้าที่ในการจำลองปัญหาการจัดวางผังงานให้อยู่ในรูปของสตริง ซึ่งในขั้นตอนนี้จะเป็นการจำลองเอ็มทีสไลซิ่งทรีให้อยู่ในรูปของสตริง และ (2) ส่วนของการค้นหาพันธุการ (Genetic Search) ในส่วนนี้ อัลกอริธึมจะทำการสร้างตำแหน่งการวางของหน่วยงานให้กับเอ็มทีสไลซิ่งทรีโดยการสุ่ม เช่น สุ่มให้หน่วยงานที่ 1 วางอยู่ด้านซ้ายของหน่วยงานที่ 3 เป็นต้น และตัวที่กำหนดตำแหน่งของหน่วยงานเราจะเรียกว่า *ตัวดำเนินการ (Operator)* เมื่อเอ็มทีสไลซิ่งทรีถูกทำเป็นสไลซิ่งทรีด้วยการใส่ค่าตัวดำเนินงานลงไปแล้ว ตัวระบบจะทำการคำนวณค่าฟิตเน็ส (Fitness Value) ของแผนผังนั้นจากสไลซิ่งทรีที่สร้างขึ้น ในการสุ่มค่าตัวดำเนินงานจะมีการสุ่มออกมาหลายตัวเลือก แต่ละตัวเลือกก็จะสร้างแผนผังออกมา 1 ชุด และระบบจะคำนวณค่าฟิตเน็สของหน่วยงานออกมา หน่วยงานที่มีค่าฟิตเน็สที่ดีที่สุดจะได้รับการคัดเลือก

การสร้างรหัสสตริง ในการสร้างรหัสสตริง เราจะใช้ตัวดำเนินการที่อยู่บนสไลซิ่งทรีเป็นตัวแทนในการสร้าง เพื่อใช้เป็นโครโมโซม (Chromosome) ของประชากรในการคัดเลือก ในการออกแบบระบบ เราออกแบบให้ตัวดำเนินการของรหัสสตริงมีองค์ประกอบ 3 ส่วนคือ ตัวดำเนินการตัด (Cutting Operator) ตำแหน่งของการวาง (Allocation Position) และการหมุน (Orientation) ซึ่งนิยามและความยาวของตัวดำเนินการแต่ละตัวเป็นดังนี้

- **ตัวดำเนินการตัด (Cutting Operator)** คือตัวดำเนินการที่บอกตำแหน่งการวางของหน่วยงาน 2 หน่วยหรือชุด (ในกรณีผังงานย่อย) ว่าตัวที่สองวางอยู่ตำแหน่งใดของตัวแรก ตัวดำเนินการตัดทั้งหมดมีอยู่ 4 ชนิดคือ T (บน) B (ล่าง) L (ซ้าย) และ R (ขวา) ดังตัวอย่างในรูปที่ 4.16

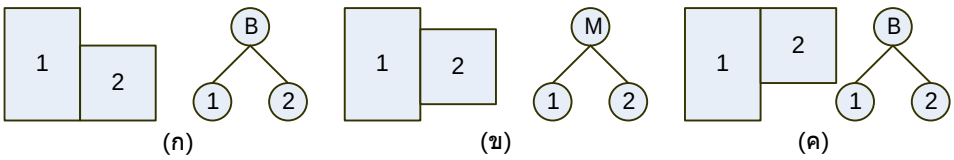


รูปที่ 4.16 แสดงค่าของตัวดำเนินการตัด

จำนวนบิตของตัวดำเนินการตัดจะเท่ากับจำนวนของโหนดภายใน (Internal Node) ซึ่งมีค่าเท่ากับ n – 1 เมื่อ n คือจำนวนหน่วยงาน

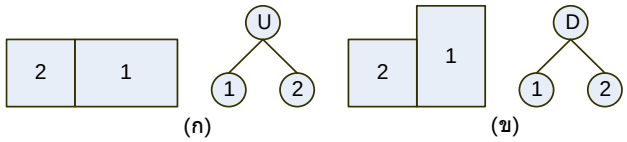
ตำแหน่งของการวาง (Allocation Position) คือตำแหน่งที่หน่วยงานที่สองวางอยู่ที่ด้านใดของหน่วยงานแรก (ที่วางอยู่ก่อน) ตำแหน่งของการวางมีอยู่ 3 ตำแหน่งคือ ข้างล่าง (B) ตรงกลาง (M) และข้างบน (T) ตาม

รูปที่ 4.17 แสดงตำแหน่งของการวางทั้งสามจำนวนบิตของตำแหน่งของการวาง จะมีค่าเท่ากับ n – 1 เมื่อ n คือจำนวนหน่วยงาน

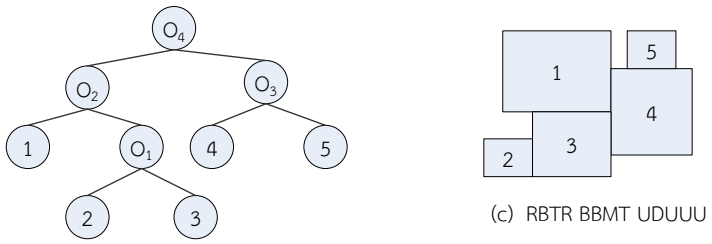


รูปที่ 4.17 ตำแหน่งของการวาง

- **การหมุน (Orientation)** จะบอกถึงลักษณะการวางของหน่วยงานว่าวางอยู่ที่องศาที่เท่าไร เนื่องจากว่าหน่วยงานที่เราพิจารณาอยู่นี้ เราสนใจที่รูปสี่เหลี่ยม ดังนั้นการหมุนของหน่วยงานจะกำหนดให้มีอยู่ 2 ตำแหน่งเท่านั้นคือ U (0 องศา) และ D (90 องศา) จำนวนบิตของการหมุนจะมีค่าเท่ากับจำนวนหน่วยงานคือ n ดังแสดงรูปที่ 4.18



รูปที่ 4.18 การหมุน



(a) Postorder expression: $123O_1O_245O_3O_4$

(b) Bit string: $Op_1Op_2Op_3Op_4A_1A_2A_3A_4Or_1Or_2Or_3Or_4Or_5$

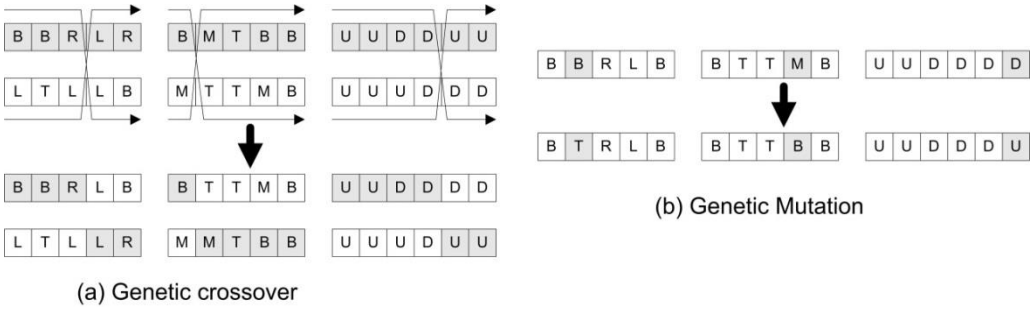
รูปที่ 4.19 การแทนค่าสไลซิ่งทรีด้วยสตริง

ตัวอย่างของการแทนค่าสไลซิ่งทรีเป็นสตริงจะแสดงไว้ในรูปที่ 4.19 ซึ่งมีจำนวนบิตที่อยู่ในสตริงทั้งหมดเท่ากับ (n – 1) + n + (n – 1) หรือ 3n – 2

การค้นหาพันธุการ (Genetic Search) ดังที่ได้กล่าวไปแล้วว่า การสร้างแผนผังเบื้องต้น (Initial Layout) นั้น จะเริ่มจากการที่ระบบสร้างตัวดำเนินการ โดยการสุ่มค่าของตัวดำเนินการให้กับเอ็มทีสไลซิ่งทรีที่สร้างจากอัลกอริธึมแม็กซิมั่มเวทแมชชิ่ง จากนั้นสไลซิ่งทรีที่ได้ก็จะถูกนำไปแทนค่าตามกระบวนการที่ได้กล่าวไปแล้ว ในเรื่องการสร้างรหัสพันธุการให้กลายเป็นโครโมโซมที่เป็นประชากรเริ่มต้น (Initial Population) ของการค้นหาพันธุการ ในการสร้างประชากรเริ่มต้น ระบบจะสร้างสตริง

ขึ้นมาชุดหนึ่ง จากนั้นระบบก็จะทำการครอสโอเวอร์และมิวเตชันในหมู่ประชากรเริ่มต้นนั้น ก็จะทำให้เกิดประชากรชุดใหม่ขึ้นมาอีกจำนวนหนึ่ง ให้ทำการหาค่าฟิตเน็ส (Fitness Value) ของสตริงแต่ละตัว แล้วคัดเลือกเอาตัวที่ดีที่สุดออกมาชุดหนึ่ง ตัวที่ได้รับการคัดเลือกนี้จะถูกนำมาทำการครอสโอเวอร์และมิวเตชัน แล้วเกิดเป็นประชากรชุดใหม่อีก และคำนวณหาค่าฟิตเน็สเพื่อคัดประชากรที่ดีที่สุดออกมาอีกหนึ่งจำนวน ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งประชากรชุดใหม่ที่ได้ออกมามีค่าฟิตเน็สไม่ดีกว่าชุดเก่า ให้หยุดกระบวนการค้นหา แล้วเลือกสตริงที่ดีที่สุดของประชากรชุดสุดท้ายมาทำการเปลี่ยนเป็นแผนผัง เราก็จะได้แผนผังที่ดีที่สุดเป็นคำตอบของการแก้ปัญหา

- ***ครอสโอเวอร์ (Crossover)*** มีกระบวนการอยู่ 2 ขั้นตอนคือ (1) การเลือกจับคู่สตริงด้วยวิธีการสุ่ม (Random) และ (2) ที่ตำแหน่ง k ใดๆ ในทั้งสามส่วนของสตริง ให้ค่า k เป็นค่าที่ได้จากการสุ่ม ดังแสดงในรูปที่ 4.20(a)
- ***มิวเตชัน (Mutation)*** สามารถเกิดขึ้นได้กับทุกบิตของสตริงโดยการสุ่ม และอัตราของการเกิดโดยปกติแล้วจะมีค่าที่น้อยมาก ประมาณ 1% ดังแสดงในรูปที่ 4.20(b)



รูปที่ 4.20 แสดงตัวอย่างของการเกิดครอสโอเวอร์และมิวเตชัน

สำหรับอ็อบเจ็กทีฟฟังก์ชัน (Objective Function) ที่ใช้สำหรับการคำนวณค่าฟิตเน็สของสตริงแต่ละตัว ใช้หลักการของพลังงานสถิต (Sirinaovakul, 1994) ซึ่งมีสมการดังนี้

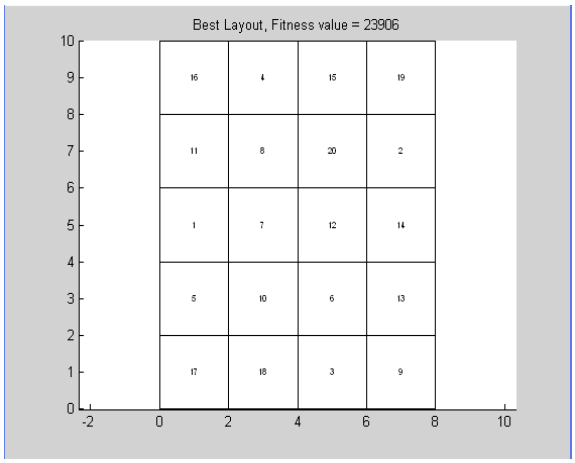
$$\square\square\square\square = \frac{1}{2}\sum_{\square=1}^{\square}\sum_{\square=\square+1}^{\square}\square_{\square\square}(\square_{\square\square}\square_{\square\square})^2$$ สมการ 4.2

เมื่อ E คือค่าใช้จ่ายของแผนผังทั้งหมด

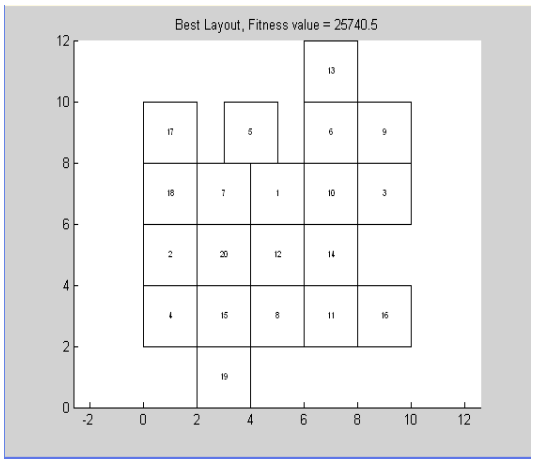
ω_{ij} คือค่าความสัมพันธ์ระหว่างหน่วยงาน i และหน่วยงาน j

f_{ij} คือความถี่ของการเดินทางระหว่างหน่วยงาน i และ j

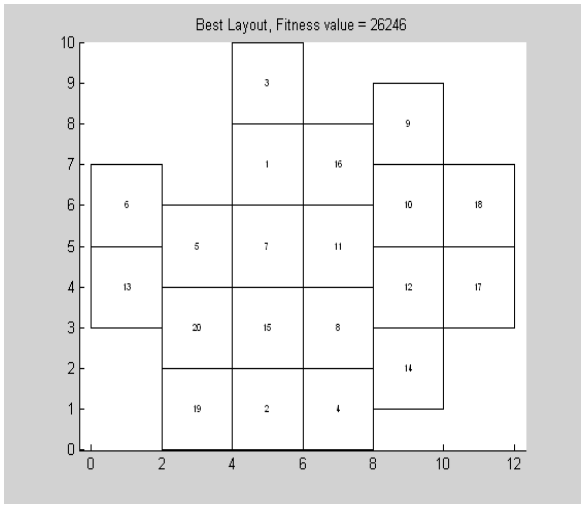
d_{ij} คือระยะห่างระหว่างหน่วยงาน i และหน่วยงาน j ซึ่งเป็นระยะทางยูคลีเดียน (Euclidian Distance) ระหว่างจุดกลาง (Centroid) ของหน่วยงาน i และหน่วยงาน j



(a) MWM



(b) DK



(c) NC

รูปที่ 4.21 ตัวอย่างผลลัพธ์ที่ได้จากการทำงานของระบบเมื่อเปรียบเทียบกับ DK และ NC

รูปที่ 4.21 เป็นภาพที่แสดงผลการสร้างแผนผังที่ได้จากระบบที่ออกแบบไว้เมื่อเปรียบเทียบกับระบบของ DK (Dunlop, 1985) และ NC (Numerical Clustering Method; (Tam, 1992)) แผนผังนี้เกิดจากการวางหน่วยงาน 20 หน่วยงานที่มีคุณสมบัติเหมือนกันหมด แต่มีวิธีการหาผังงานเริ่มต้นที่ต่างกัน ในรูปที่ 4.21(a) เป็นการหาผังงานเริ่มต้นด้วยการใช้แม็กซิมั่มเวทแมชชิ่ง รูปที่ 4.21(b) ใช้อัลกอริธึมของ DK และรูปที่ 4.21(c) เป็นของ NC เราจะเห็นได้ว่าอัลกอริธึมในรูปที่ 4.21(a) ให้แผนผังสุดท้ายออกมามีพื้นที่เสีย (Scrap Area) น้อยที่สุด เพราะเป็นรูปทรงสี่เหลี่ยมที่ไม่มีพื้นที่ว่างอยู่ภายในแผนผัง

4.6 อัลกอริธึมพันธุการขั้นสูง

ในหัวข้อนี้จะกล่าวถึงเทคนิคในขั้นสูงขึ้นของอัลกอริธึมพันธุการ ประเด็นที่เป็นหัวใจของการพัฒนาอัลกอริธึมในหัวข้อนี้ก็คือ การทำให้อัลกอริธึมมีสมรรถนะสูงขึ้น และการทำให้อัลกอริธึมนี้สามารถแก้ปัญหาเชิงตัวเลขได้

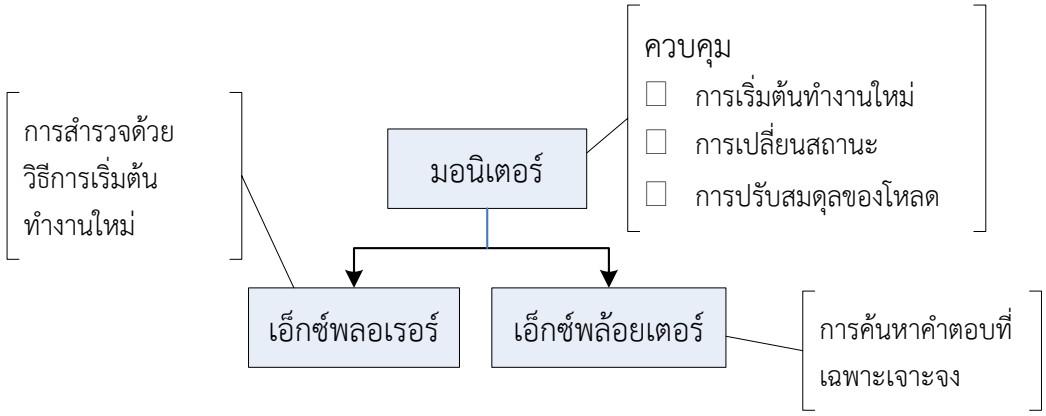
มีนักวิจัยจำนวนมากได้ค้นคิดเทคนิคใหม่ๆ ขึ้นมา เพื่อทำให้อัลกอริธึมมีสมรรถนะสูงขึ้น ด้วยการให้อัลกอริธึมพันธุการมีเอ็กซ์พลอเรชันที่ครอบคลุมพื้นที่จำนวนมากของปริภูมิปัญหา ทำให้โอกาสของการติดกับดักของค่าเหมาะที่สุดแบบโลคอล (Local optimum) น้อยลง เทคนิคหนึ่งที่มีนักวิจัยสนใจจำนวนมากคือการแบ่งกลุ่มประชากร และเทคนิคการครอสโอเวอร์หลายพ่อแม่พันธุ ดังที่เราจะได้กล่าวต่อไป

สำหรับการทำให้อัลกอริธึมสามารถแก้ปัญหาเชิงตัวเลขได้นั้น เป็นเทคนิคที่ทำให้อัลกอริธึมพันธุการสามารถแก้ปัญหาที่มีลักษณะเป็นฟังก์ชันคณิตศาสตร์ โดยเฉพาะอย่างยิ่งฟังก์ชันคณิตศาสตร์ที่มีตัวแปรเป็นค่าต่อเนื่อง (หรือเลขจำนวนจริง)

4.6.1 การทำงานแบบแบ่งกลุ่มประชากร

เพื่อเป็นการทำให้สมรรถนะของอัลกอริธึมพันธุการดีขึ้น ซูซูอิ (Tsutsui, 1997) ได้พัฒนาอัลกอริธึมของเชลแมน (Eshelman, 1991) ต่อมาเรียกว่า บีจีเอ (bGA: Bi-population GA) ขึ้น เพื่อ

ทำให้เอ็กซ์พลอเรชันของอัลกอริธึมมีประสิทธิภาพสูงขึ้นโดยการเพิ่มพื้นที่การสำรวจให้มากขึ้น ซึ่งจะทำให้เอ็กซ์พล้อยเตชันมีโอกาสทำการค้นหาคำตอบที่เหมาะที่สุดในพื้นที่ที่มากขึ้น



รูปที่ 4.22 การทำงานของ บีจีเอ (bGA)

การทำงานเริ่มต้นโดยการแบ่งประชากรออกเป็นสองชุดย่อย ชุดแรกเรียกว่าเอ็กซ์พลอเรอร์ (Explorer) เพื่อทำหน้าที่ในการสำรวจปริภูมิค้นหา (Search Space) ชุดที่สองเรียกว่าเอ็กซ์พล้อยเตอร์ (Exploiter) ทำหน้าที่ในการหาคำตอบที่ดีที่สุดตลอดการค้นหาจากคำตอบข้างเคียง โดยที่มีกลไกในการควบคุมการทำงานของประชากรย่อยทั้งสองส่วนเรียกว่ามอนิเตอร์ (Monitor) กลไกนี้ทำหน้าที่ 3 อย่าง คือการกระตุ้นให้เอ็กซ์พลอเรอร์เริ่มต้นทำงานใหม่ (Restart) กำหนดการเปลี่ยนสถานะการทำงาน และปรับความสมดุลของโหลดของประชากรย่อยทั้งสอง

หลักการทำงานของบีจีเอเริ่มต้นด้วยการให้เอ็กซ์พลอเรอร์ทำงานก่อน โดยการสร้างประชากรเริ่มต้นด้วยการสุ่ม เมื่อเอ็กซ์พลอเรอร์ทำงาน (ครอสโอเวอร์และมิวเตชัน) ได้จนถึงเงื่อนไขตามที่เราได้ตั้งไว้ จากนั้น ให้สำเนาประชากรทั้งชุดไปไว้ที่เอ็กซ์พล้อยเตอร์ สำหรับเอ็กซ์พลอเรอร์ก็จะสร้างประชากรชุดใหม่ด้วยการสุ่ม และทั้งเอ็กซ์พลอเรอร์ และเอ็กซ์พล้อยเตอร์ ก็จะทำงานคู่ขนานกันไป จนได้เงื่อนไขของการเริ่มต้นใหม่ ตัวมอนิเตอร์ก็จะกระตุ้นให้เอ็กซ์พลอเรอร์สุ่มประชากรชุดใหม่ขึ้นมาอีกชุดหนึ่ง ในขณะเดียวกัน ระบบก็จะพิจารณาว่า ถ้าหากประชากรชุดเก่าในเอ็กซ์พรอเรอร์มีคุณสมบัติดีกว่าประชากรปัจจุบันในเอ็กซ์พล้อยเตอร์ ก็ให้ย้ายประชากรในเอ็กซ์พรอเรอร์ไปแทนที่ประชากรในเอ็กซ์พล้อยเตอร์ และทิ้งประชากรเดิมที่อยู่ในเอ็กซ์พล้อยเตอร์ แต่ถ้าประชากรชุดเก่าในเอ็กซ์พลอเรอร์มี

คุณสมบัติแย่กว่าประชากรปัจจุบันในเอ็กซ์พล้อยเตอร์ ให้ทิ้งประชากรในเอ็กซ์พรอเรอร์ไป และให้เอ็กซ์พรอเรอร์ใช้ประชากรชุดใหม่ที่เพิ่งสุ่มขึ้นมาทำงานแทน จนกระทั่งได้เงื่อนไขของการเริ่มต้นใหม่ เอ็กซ์พลอเรอร์ก็จะถูกกระตุ้นให้สร้างประชกรใหม่โดยตัวมอนิเตอร์อีก การทำงานจะวนรอบเช่นนี้จนกระทั่งได้คำตอบที่ต้องการ

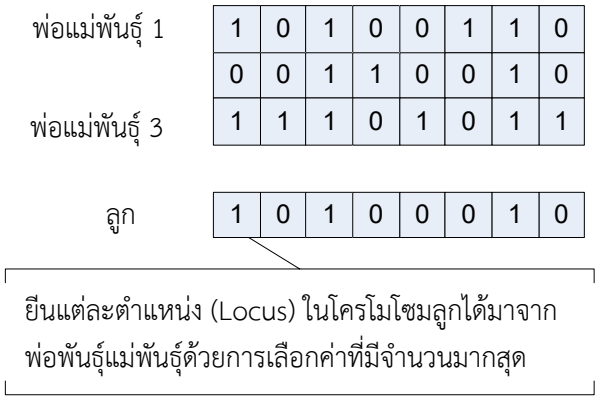
4.6.2 การครอสโอเวอร์แบบหลายพ่อแม่พันธุ์

ในทางธรรมชาติ การครอสโอเวอร์มักจะเกิดกับพ่อแม่เพียงคู่เดียว แต่ในทางคณิตศาสตร์และอัลกอริธึมแล้ว การครอสโอเวอร์ที่เกิดจากพ่อพันธุ์และแม่พันธุ์ที่มีมากกว่าสองสามารถทำได้ ไอเบนและคณะ (Eiben, 1994) ได้นำเสนออัลกอริธึมการแลกเปลี่ยนยีนแบบหลายพ่อแม่พันธุ์ (Multi-parent Recombination) โดยการใช้เทคนิคที่เรียกว่า สแกนนิ่ง (Scanning) และได้ออกแบบวิธีการสแกนนิ่งไว้ 4 วิธีและวิธีครอสโอเวอร์อีก 1 วิธีคือ

ยูนิฟอร์ม สแกนนิ่ง (Uniform Scanning) วิธีการแลกเปลี่ยนยีนแบบนี้จะเริ่มต้นจากยีนตัวแรกที่อยู่ทางซ้ายมือของทุกโครโมโซม จากนั้นจึงสแกนจากซ้ายไปขวาเพื่อสร้างโครโมโซมลูกใหม่ออกมาโดยการสร้างยีนทีละตัวจากซ้ายไปขวาเช่นกัน ที่แต่ละตำแหน่งของยีน (Locus) บนโครโมโซมลูก ให้เลือกด้วยวิธีการสุ่มว่าจะมีค่าตามยีนของโครโมโซมพ่อแม่พันธุ์ตัวไหนที่ตำแหน่งเดียวกัน ดังแสดงในรูปที่ 4.23 การสแกนแบบยูนิฟอร์ม

รูปที่ 4.23 การสแกนแบบยูนิฟอร์ม

อ็อคเคอร์เร้นซ์ เบสด์ สแกนนิ่ง (Occurrence Based Scanning) วิธีการนี้จะพิจารณาจากความถี่ของค่าที่ปรากฏในแต่ละตำแหน่ง โดยเริ่มสแกนจากซ้ายไปขวาเช่นกัน เช่น ที่ตำแหน่งแรก ค่าของโครโมโซมลูกจะพิจารณาจาก ตำแหน่งแรกของโครโมโซมพ่อแม่พันธุ์ทุกโครโมโซม ว่าค่าอะไรที่มีความถี่ของการเกิดสูงสุด ก็ให้เลือกค่านั้นมาใส่ให้กับตำแหน่งเดียวกัน (ตำแหน่งแรก) ของโครโมโซมลูก ถ้าหากว่าค่าที่ตำแหน่งนั้นมีความถี่ของการเกิดเท่ากัน ให้เลือกค่าของพ่อแม่พันธุ์ชุดแรกที่ปรากฏ มาใส่ให้กับตำแหน่งเดียวกันของโครโมโซมลูก จากนั้นก็จะพิจารณาที่ตำแหน่งถัดไปด้วยวิธีการเดียวกัน จนกระทั่งถึงตำแหน่งสุดท้าย เราจะได้ค่าใหม่ทุกตำแหน่งของโครโมโซมลูก ดังตัวอย่างตามรูปที่ 4.24



รูปที่ 4.24 การสแกนแบบอ็อคเคอร์เร้นซ์ เบสด์ สแกนนิ่ง

ฟิตเน็ส เบสด์ สแกนนิ่ง (Fitness Based Scanning) การสแกนแบบนี้เริ่มจากซ้ายไปขวาแต่การกำหนดค่าให้กับยีนแต่ละตำแหน่งของโครโมโซมลูกพิจารณาจากโครโมโซมพ่อแม่พันธุ์ทุกตัวในตำแหน่งเดียวกัน สำหรับการเลือกว่าค่าของโครโมโซมลูกจะได้รับการถ่ายทอดค่าจากพ่อแม่พันธุ์ตัวใดจะพิจารณาจากสัดส่วนของค่าฟิตเน็สของโครโมโซมแต่ละตัวหรือค่าความคาดหวัง (Expected Value) สำหรับพ่อแม่พันธุ์ i ที่มีค่าฟิตเน็ส f(i) ความน่าจะเป็นที่ค่าของพ่อแม่พันธุ์ชุดที่ i จะถูกถ่ายทอดค่าของมันไปยังโครโมโซมลูกเท่ากับ P(i) ดังนี้ (เช่นเดียวกันกับการเลือกสรรแบบวงล้อรูเล็ตต์)

$$\square(\square) = \frac{\square(\square)}{\sum \square(\square)}$$

ดังนั้น ค่าความคาดหวัง (Expected value: E(i)) ที่ค่าของยีนลูกจะได้รับการถ่ายทอดค่าจากพ่อแม่พันธุ์ i เท่ากับ

□(□) = □(□) ∗ □□□□□□□□□□□□□□□□□□

อะแด็บติ่ง สแกนนิ่ง สำหรับการแทนค่ายีนที่ต่างกัน (Adapting Scanning to Different Representation Types) วิธีการนี้จะเริ่มต้นที่ตำแหน่งหรือโลกัสแรกของทุกโครโมโซม สำหรับการหาค่าให้กับตำแหน่งแรกของโครโมโซมลูก ค่าที่อยู่ในโครโมโซมที่ซ้ำกันมากที่สุดจะได้รับการเลือกให้มาเป็นค่าของโครโมโซมลูก ถ้าไม่มีค่าซ้ำ ค่าของโครโมโซมพ่อแม่พันธุ์ตัวแรกจะได้รับเลือก สำหรับโครโมโซมที่ได้รับการเลือก ให้เลื่อนตำแหน่งของยีนที่จะพิจารณาออกไปหนึ่งตำแหน่ง สำหรับโครโมโซมที่ยังไม่ได้รับเลือก ไม่ต้องเลื่อนตำแหน่งที่จะต้องพิจารณา จากนั้นก็จะพิจารณาค่าที่จะนำมาใส่ในโครโมโซมลูกตำแหน่งถัดมา โดยเลือกจากตำแหน่งที่จะพิจารณาว่า มีค่าใดที่ซ้ำกันมากที่สุดให้เลือกค่านั้นมาเป็นค่าของโครโมโซมลูกในตำแหน่งที่ต้องการ ในการสแกนแบบนี้ เมื่อการทำงานผ่านไป ตำแหน่งที่จะพิจารณาของแต่ละโครโมโซมจะไม่เท่ากัน ซึ่งต่างจากวิธีที่ผ่านมา

จากตัวอย่างตามรูปที่ 4.25 การทำงานของอะแด็บติ่ง สแกนนิ่ง ที่รูปแรกซ้ายมือตอนบน เมื่อเริ่มต้น ตำแหน่งพิจารณาอยู่ที่ตำแหน่งแรกของพ่อแม่พันธุ์ทุกตัว ที่ตำแหน่งนี้มีค่า 2 ซ้ำกันมากสุด เลือก 2 มาเป็นค่าให้กับโครโมโซมลูกที่ตำแหน่งแรก จากนั้นโครโมโซมพ่อแม่พันธุ์ชุดที่ 2 และ 3 เท่านั้นที่เลื่อนตำแหน่งพิจารณา (ช่องตารางที่มีสีเข้มกว่า) ไปหนึ่งตำแหน่ง จากนั้น พิจารณาเฉพาะที่ตำแหน่งพิจารณา (ช่องตารางที่มีสีเข้มกว่า) เท่านั้น ว่ามีค่าของพ่อแม่พันธุ์ใดซ้ำกันมากสุด เลือกค่านั้นมาใส่ให้กับโครโมโซมลูกตำแหน่งถัดไป จะได้ค่า 3 ซึ่งซ้ำกัน 2 ตัว มาใส่ให้กับโครโมโซมลูกในตำแหน่งที่สอง แล้วทำการเลื่อนตำแหน่งพิจารณาของพ่อแม่พันธุ์ 1 และ 3 การพิจารณาจะดำเนินไปเช่นนี้จนได้ค่าของโครโมโซมลูกครบทุกตำแหน่ง

รูปที่ 4.25 การทำงานของอะแด็บติ่ง สแกนนิ่ง

แอ็ดจาเซนซี เบสด์ ครอสโอเวอร์ (Adjacency Based Crossover) วิธีการครอสโอเวอร์แบบนี้คล้ายกับวิธีการอะแด็บติ่ง สแกนนิ่ง แต่มีความแตกต่างอยู่ที่วิธีการเลื่อนตำแหน่งพิจารณา การเลื่อนตำแหน่งของวิธีนี้ ตำแหน่งพิจารณาจะถูกเลื่อนไปอยู่หลังตำแหน่งที่เก็บค่าที่ถูกเลือก และการเลื่อนตำแหน่งพิจารณาจะเลื่อนทุกโครโมโซม ความแตกต่างอีกจุดหนึ่งคือ ในการเลือกค่าแรกให้กับโครโมโซมลูก จะเลือกจากค่าแรกของโครโมโซมพ่อแม่พันธุ์ชุดแรก

รูปที่ 4.26 แอ็ดจาเซนซี เบสด์ ครอสโอเวอร์ หรือ โอบี สแกน

จากรูปที่ 4.26 เมื่อเริ่มต้น เลือกค่าแรกของโครโมโซมพ่อแม่พันธุ์ชุดแรกคือ 3 มาเป็นค่าแรกให้กับโครโมโซมลูก จากนั้นตำแหน่งพิจารณาของทุกโครโมโซมจะถูกเลื่อนไปอยู่ที่ตำแหน่งหลังตำแหน่งที่มีค่า 3 ตำแหน่งที่สองของโครโมโซมลูกจะได้ค่า 8 เพราะเป็นค่าที่อยู่หลัง 3 ที่ซ้ำกันมากที่สุด ในกรณีที่ตำแหน่งพิจารณาใหม่นั้นซ้ำกับตำแหน่งที่ได้รับการเลือกไปแล้ว ให้เลื่อนออกไปอีกจนกระทั่งพบตำแหน่งที่ไม่ซ้ำกับค่าที่เคยเลือกไปแล้ว จากรูปที่ 4.26 แถวบนรูปขวามือสุด ตำแหน่งพิจารณาจะต้องเป็นตำแหน่งหลังเลข 1 แต่เนื่องจากโครโมโซมพ่อแม่พันธุ์ชุดที่ 4 ตำแหน่งหลังเลข 1 มีค่า 3 ซึ่งเป็นค่าที่เคยถูกเลือกแล้ว ตำแหน่งพิจารณาจึงถูกเลื่อนออกไปอีกหนึ่งตำแหน่งคือตำแหน่งที่มีเลข 2

4.6.3 การเข้ารหัสแบบค่าจำนวนจริง

ผู้ที่นำเสนอวิธีการเข้ารหัสแบบค่าจำนวนจริง (Real coded) ของอัลกอริธึมพันธุการคนแรกคือ ไร้ท์ (A. Wright, 1991) วิธีการของไร้ท์คือการกำหนดขอบเขตของค่าในพารามิเตอร์ของโครโมโซม โดยที่มองว่า โครโมโซมเป็นเวคเตอร์ของพารามิเตอร์ที่มีค่าเป็นเลขจำนวนจริง และยีนของโครโมโซมก็คือพารามิเตอร์ของเวคเตอร์ สำหรับเลขจำนวนจริงที่เป็นค่าของพารามิเตอร์ในเวคเตอร์ หรือกล่าวอีกนัยหนึ่งคือคุณสมบัติที่แสดงด้วยเลขจำนวนจริงของยีนในโครโมโซม เรียกว่าแอลลีล (Allele) โดยที่ค่าของแอลลีลนี้จะมีการกำหนดขอบเขตสูงสุดและต่ำสุดดังนี้

Maximize	$f(x_1, x_2, \ldots x_n)$
เมื่อ	x_i เป็นเลขจำนวนจริงที่มีค่าอยู่ระหว่าง $a_i \le x_i \le b_i$
และ	ค่า a_i และ b_i เป็นค่าคงที่
โดยที่	n เป็นจำนวนบิตของโครโมโซม

ตัวอย่างเช่นถ้าเรามีพารามิเตอร์ 2 ตัว x_1 และ x_2 ค่าของพารามิเตอร์อาจจะเป็น $0 \le x_1 \le 1$ และ $-2 \le x_2 \le 2$ โดยที่ต้องกำหนดค่าของ n ด้วยเช่น $n = 5$ ซึ่งเป็นจำนวนบิตของโครโมโซมที่เราจะใช้งานเท่ากับ 5

การเข้ารหัส ถ้ากำหนดว่า x_i มีค่าที่อยู่ระหว่างช่วงของ $a_i \le x_i \le b_i$ ดังนั้นวิธีการที่จะเข้ารหัสของค่า x_i ที่มีจำนวน n บิตให้เป็นรหัสไบนารีคือ

$$□_□ + □\frac{□_□ - □_□}{2^□} \text{ และ } □_□ + (□ + 1)\frac{□_□ - □_□}{2^□}$$

โดยที่ค่า k เป็นค่าที่อยู่ระหว่าง $0 \le k < 2^n$ ตัวอย่างเช่น ถ้า $a_i = 0$, $b_i = 4$ และ $n = 5$ ดังนั้นค่าจำนวนจริงของ 3/8 และ 1/2 คือ

$$0 + 3\frac{4-0}{2^5} = \frac{3}{8} \text{ และ } 0 + (3 + 1)\frac{4-0}{2^5} = \frac{4}{8}$$

ซึ่งค่า k ในที่นี้คือ 3 ค่าของ 3 เมื่อเป็นไบนารีของโครโมโซมที่มี 5 บิตคือ 00011 เพื่อที่จะหลีกเลี่ยงการกำหนดค่า k เป็นช่วง ดังนั้นในการคำนวณ เราจะใช้เฉพาะสูตรทางซ้ายมือคือ

$$□_□ + □\frac{□_□ - □_□}{2^□}$$

ซึ่งจะทำให้เราได้ 3/8 เป็นค่าไบนารีคือ 00011

การครอสโอเวอร์ ของการเข้ารหัสแบบค่าจำนวนจริง จะใช้วิธีการเดียวกันกับการครอสโอเวอร์ของ ไบนารีสตริง โดยมีการครอสโอเวอร์ที่จุดเดียว เช่นถ้าเรามีขอบเขตของค่าที่อยู่ระหว่าง 0 และ 1 จำนวนพารามิเตอร์ทั้งหมดมี 5 ตัว และค่าของพ่อแม่พันธุ์ทั้งสองคือ 5/32 และ 27/32 ซึ่งสามารถเขียนเป็นรหัสไบนารีได้เท่ากับ 001 01 และ 110 11 และกำหนดให้การครอสโอเวอร์เกิดขึ้นที่ตำแหน่งที่ 3 และ 4 ดังนั้นผลของการครอสโอเวอร์คือ 001 **11** และ 110 **01** ตามลำดับ ผลที่ได้เมื่อเปลี่ยนเป็นเลขจำนวนจริงคือ 7/32 และ 25/32

เมื่อพิจารณาที่ 2 บิตสุดท้ายที่เกิดครอสโอเวอร์ 01 และ 11 ซึ่งมีค่าในเลขฐานสิบเท่ากับ 1 และ 3 มีความแตกต่างเป็น 2 หรือ 2/32 ทำให้ค่า ±2/32 กลายเป็นค่าที่มีผลต่อการเปลี่ยนแปลงซึ่งไรท์เรียกค่านี้ว่า เพอร์เทอบเบชัน (Perturbation) ตารางที่ 4.14

	ค่าจริง	ค่าไบนารี	ค่าไบนารีลูก	ค่าจริงลูก	ผลต่าง
พ่อแม่พันธุ์1	5/32	001 01	001 11	7/32	+2/32
พ่อแม่พันธุ์2	27/32	110 11	110 01	25/32	-2/32

ตารางที่ 4.14 การแสดงความแตกต่างของ 2 บิตหลัง

มิวเตชัน เป็นการเปลี่ยนค่าแอลลีลในยีนที่ตำแหน่งใดๆ ในโครโมโซม กล่าวคือเป็นการเปลี่ยนค่าของพารามิเตอร์ในเวคเตอร์ของสตริง ซึ่งตำแหน่งนี้เกิดโดยการสุ่ม การเปลี่ยนจะเป็นการเปลี่ยนจาก 0 เป็น 1 หรือ 1 เป็น 0 ถ้าการเปลี่ยนเกิดจาก 0 เป็น 1 จะมีผลทำให้ค่าของโครโมโซมมีมากขึ้น เรียกว่าการเปลี่ยนทางบวก (Positive Direction) ในทางตรงกันข้าม ถ้าการเปลี่ยนเกิดจาก 1 เป็น 0 เรียกว่าเป็นการเปลี่ยนทางลบ (Negative Direction)

2-บิต (รหัสเกร์ย)	2-บิต (รหัสไบนารี)	4-บิต (รหัสเกร์ย)	4-บิต (รหัสไบนารี)
00	00	0000	0000
01	01	0001	0001
11	10	0011	0010
10	11	0010	0011
3-บิต (รหัสเกร์ย)	2-บิต (รหัสไบนารี)	0110	0100
000	000	0111	0101
001	001	0101	0110
011	010	0100	0111
010	011	1100	1000
110	100	1101	1001
111	101	1111	1010
100	110	1110	1011
101	111	1010	1100
		1011	1101
		1001	1110
		1000	1111

ตารางที่ 4.15 ตารางรหัสเกร์ย และรหัสไบนารี

การนำเสนอของไรท์เรื่องการเข้ารหัสแบบค่าจำนวนจริง นอกจากการเข้ารหัสด้วยเลขไบนารีแล้ว เขายังได้นำเสนอวิธีการเข้ารหัสด้วย รหัสเกร์ย (Gray Code) ที่เป็นเลข 2 หลักคือ 0 และ 1

เช่นเดียวกับไบนารี แต่วิธีการเรียงจะต่างกัน โดยที่รหัสเกร์ยลำดับที่ติดกันจะมีค่าที่ต่างกันเพียงบิตเดียว ดังแสดงตารางที่ 4.15 ที่แสดงการเปรียบเทียงรหัสเกร์ยกับรหัสไบนารี

การเข้ารหัสค่าจริงด้วยรหัสเกร์ยมีสูตรในการแทนค่าที่ต่างกัน สำหรับการทำคลอโอเวอร์ และมิวเตชันจะเหมือนกัน แต่ผลที่ได้จะต่างกันทั้งจำนวนและความหมาย สำหรับงานวิจัยที่ใช้วิธีการเข้ารหัสแบบจำนวนจริงนั้นยังมีอยู่อีกหลายงาน เช่น โกลเบอร์ก (Goldberg, 1991) ได้วิเคราะห์การทำงานของอัลกอริธึมแบบเลขจำนวนจริง เฮอร์เรอร์ (Herrer, 1998) ได้เสนอวิธีการทำครอสโอเวอร์ และการทำมิวเตชันของการเข้ารหัสแบบนี้ พร้อมการยกตัวอย่าง ชาน (Chang, 1998) ได้นำวิธีการเข้ารหัสแบบนี้ไปใช้ในการบริหารน้ำท่วม ศรีกานธ์ (Srikanth, 2008) ได้ประยุกต์ใช้การเข้ารหัสแบบนี้ไปใช้งานกับการควบคุมเครื่องตัด

4.7 การปรับสมรรถนะ

การปรับสมรรถนะการทำงานของอัลกอริธึมขึ้นอยู่กับองค์ประกอบหลายอย่าง เพื่อเป็นการทำความเข้าใจกลไกของการปรับสมรรถนะเบื้องต้น ในหัวข้อนี้จะขอกล่าวถึงพารามิเตอร์ต่างๆ ที่มีผลต่อการทำงาน และอธิบายถึงกลไกการลู่เข้าสู่คำตอบของอัลกอริธึมด้วย

4.7.1 พารามิเตอร์แบบสุ่ม

ในอัลกอริธึมพันธุการนี้ มีพารามิเตอร์ที่ต้องใช้วิธีการสุ่มที่น่าสนใจคือ สถิติของการเกิดครอสโอเวอร์ สถิติของการเกิดมิวเตชัน และจำนวนประชากร

สถิติของการเกิดครอสโอเวอร์ เป็นค่าที่บอกถึงความถี่ของการเกิดครอสโอเวอร์ ถ้าหากว่าไม่มีการครอสโอเวอร์ ลูกหลานในรุ่นต่อไปก็จะเหมือนกับพ่อแม่ทุกประการ ถ้าสถิติของการเกิดครอสโอเวอร์เป็น 100% ก็หมายความ ลูกหลานทั้งหมดมาจากการครอสโอเวอร์ การเกิดครอสโอเวอร์ก็เพราะความหวังที่ว่า ลูกหลานที่เกิดใหม่จะได้รับสิ่งที่ดีจากบรรพบุรุษ และในแต่ละรุ่นก็จะมีพัฒนาการในทางที่ดีขึ้น แต่อย่างไรก็ตาม ในบางอัลกอริธึมเช่น การเลือกแบบสเตดี สเทท (Steady State Selection) จะคัดพ่อแม่ที่ดีในรุ่นปัจจุบันไปเป็นประชากรในรุ่นถัดไปด้วย ดังนั้น การครอสโอเวอร์ก็ไม่เกิดขึ้นกับโครโมโซม ทุกตัว แต่อย่างไรก็ตามการเลือกพ่อแม่ที่ดีในลักษณะนี้จะมีน้อยมาก เพียง 2-3 ตัวเท่านั้น

สถิติของการเกิดมิวเตชัน เป็นการบอกความถี่ของการเกิดมิวเตชัน ถ้าหากว่าไม่มีการเกิดมิวเตชัน ลูกหลานที่ได้จะมาจากการครอสโอเวอร์จะไม่มีการเปลี่ยนแปลงคุณสมบัติใดๆ ถ้ามีการมิวเตชันเกิดขึ้นที่บางส่วนของโครโมโซมจะทำให้เกิดการเปลี่ยนแปลงคุณสมบัติขึ้น การเกิดมิวเตชันก็เพื่อป้องกันการเกิดโลคอลแม็กซิมั่ม แต่การเกิดนี้ไม่ควรมีบ่อย เพราะการเกิดขึ้นบ่อยจะมีผลทำให้อัลกอริธึมนี้กลายเป็นการค้นหาแบบสุ่ม

จำนวนประชากร หมายถึงจำนวนโครโมโซมที่มีในแต่ละรุ่น ถ้าจำนวนประชากรมีน้อยเกินไปอัลกอริธึมก็จะมีจำนวนโครโมโซมสำหรับการจับคู่เพื่อการครอสโอเวอร์น้อย ทำให้ขนาดของปริภูมิปัญหาเล็ก และโอกาสการได้คำตอบที่ดีจะน้อยลง ในทางตรงกันข้าม ถ้าประชากรมีมากเกินไปอัลกอริธึมจะทำงานช้าเพราะปริภูมิปัญหาจะมีขนาดใหญ่มาก จากการทดลองหาขนาดของจำนวนประชากร ถ้ามีมากจนถึงจุดที่จำกัดจุดหนึ่งแล้ว การเพิ่มจำนวนประชากรก็เป็นสิ่งที่ไม่จำเป็น เพราะการเพิ่มนี้จะมีผลต่อคุณภาพของการค้นหาคำตอบไม่มาก

4.7.2 การลู่เข้าสู่คำตอบ

ดังที่ได้กล่าวมาแล้ว อัลกอริธึมพันธุการมีองค์ประกอบสำคัญคือ การเลือกสรร ครอสโอเวอร์และมิวเตชัน การเลือกสรรเป็นกระบวนการที่ทำหน้าที่ในการค้นหาโครโมโซมใหม่ สำหรับการเป็นสมาชิกของประชากรในรุ่นถัดไป การเลือกสรรนี้เป็นเครื่องมือของทั้งครอสโอเวอร์ และมิวเตชัน ที่ทำหน้าที่ในการสร้างโครโมโซมใหม่ สำหรับการเลือกสรร ถ้าหากเราพิจารณาว่า ในปริภูมิค้นหา (Search space) เป็นโครโมโซมที่เป็นไปได้ทั้งหมด ที่สามารถเป็นตัวเลือกในการเลือกสรรให้เป็นประชากรในรุ่นถัดไปได้ การทำครอสโอเวอร์และการทำมิวเตชัน เป็นเสมือนการระบุตำแหน่งของกลุ่มโครโมโซมสำหรับการเลือกสรรในปริภูมิค้นหา

การทำครอสโอเวอร์ที่เป็นการสลับยีนกันของโครโมโซมพ่อและแม่ เพื่อให้ได้ประชากรใหม่เปรียบเสมือนการสร้างโครโมโซมข้างเคียงให้กับการเลือกสรร สำหรับมิวเตชันที่ทำให้เกิดการเปลี่ยนแปลงของยีนในโครโมโซม เปรียบเสมือนการย้ายพื้นที่ในการเลือกสรร

โดยปกติ การแก้ปัญหาด้วยการใช้อัลกอริธึมพันธุการ จะอาศัยตัวดำเนินการครอสโอเวอร์เป็นหลัก สำหรับตัวดำเนินการมิวเตชันจะเกิดเป็นครั้งคราว ทั้งนี้เพื่อป้องกันไม่ให้การค้นหาคำตอบ

กลายเป็นการค้นหาแบบสุ่ม แต่ถ้าอัลกอริธึมเน้นการทำครอสโอเวอร์มากเกินไป โอกาสที่การลู่เข้าสู่คำตอบก่อนเวลาอันควร (Premature convergence) ก็มีสูง และทำให้เกิดปัญหาของโลคอลมินิมั่มได้ ดังนั้นการได้คำตอบที่ดีจึงต้องทำให้อัลกอริธึมลู่เข้าสู่คำตอบในเวลาที่เหมาะสม ซึ่งอัลกอริธึมที่ดีจะต้องสร้างสมดุลของการเกิดครอสโอเวอร์และมิวเตชัน

ครอสโอเวอร์สามารถเป็นได้ทั้งเอ็กซ์พลอเรชันและเอ็กซ์พล้อยเตชัน การพิจารณาครอสโอเวอร์ในแง่ของเอ็กซ์พลอเรชัน ก็เพราะการที่ตัวดำเนินการนี้ทำหน้าที่ในการแลกเปลี่ยนยีนของพ่อและแม่เพื่อให้ได้ยีนชุดใหม่ เป็นเสมือนการกำหนดพื้นที่การค้นหาแหล่งใหม่ในปริภูมิคำตอบ สำหรับการพิจารณาครอสโอเวอร์ในแง่ของเอ็กซ์พล้อยเตชัน ก็เพราะตัวดำเนินการนี้ใช้โครโมโซมชุดเดิมที่ได้จากบรรพบุรุษเดียวกัน เปรียบเสมือนหนึ่งว่า การเปลี่ยนแปลงนั้นเกิดขึ้นในพื้นที่ที่จำกัด หรือในกลุ่มประชากรที่จำกัด มิวเตชันก็เช่นกัน สามารถเป็นได้ทั้งเอ็กซ์พลอเรชันและเอ็กซ์พล้อยเตชัน มิวเตชันถูกพิจารณาให้เป็นเอ็กซ์พลอเรชัน เพราะตัวดำเนินการนี้ทำให้เกิดโครโมโซมที่มีรูปแบบใหม่โดยการสุ่ม สำหรับมิวเตชันจะถูกพิจารณาให้เป็นเอ็กซ์พล้อยเตชันเนื่องจากการเปลี่ยนแปลงนั้นเกิดขึ้นที่บางบิตในโครโมโซมเท่านั้น ผลการเปลี่ยนแปลงนี้จะมีไม่มากนัก เพราะค่าของบิตในโครโมโซมส่วนใหญ่ยังเหมือนเดิม (Eiben, 1998)

มีนักวิจัยจำนวนมากได้ค้นหาวิธีในการปรับปรุงอัลกอริธึมพันธุการ เพื่อทำให้อัลกอริธึมสามารถค้นหาคำตอบได้ดีขึ้นด้วยเวลาที่เหมาะสม โดยการสร้างความสมดุลของเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน เช่น (Eshelman, 1991) ได้เสนอวิธีการกำหนดจำนวนประชากรเป็น N และประชากรที่เวลา t ให้แทนด้วย P(t) ประชากรที่เวลา P(t+1) หาได้จากการกำหนดให้ โครโมโซมจำนวน N/2 คู่นี้สำเนามาจาก p(t) ไปเก็บไว้ที่ I(t) จากนั้นก็ใช้ตัวดำเนินการพันธุการ คือ การคอลสโอเวอร์ การมิวเตชัน เพื่อสร้างโครโมโซมรุ่นต่อมาจำนวน N ไปเก็บไว้ที่ I’(t) แล้วใช้วิธี**การเลือกสรรแบบจัดลำดับ**เพื่อเลือกโครโมโซมจำนวน N ตัวจาก 2N ของ p(t) รวมกับ I’(t) เพื่อเกิดเป็นโครโมโซมของ P(t+1) จากนั้นใช้ **การเลือกแบบอีลีทิซึม** เพื่อประกันว่าโครโมโซมที่เลือกไว้เป็นโครโมโซมที่ดีที่สุดในตลอดการค้นหาที่ผ่านมาและจะปรากฏอยู่ใน P(t+1)

4.7.3 การทดสอบกับฟังก์ชันเทียบเคียง

โดยปกติแล้ว อัลกอริธึมพันธุการเป็นอัลกอริธึมที่ออกแบบสำหรับการแก้ปัญหาแบบคอมบินาทอเรียล แต่ในการทดสอบอัลกอริธึมกับฟังก์ชันเทียบเคียงนั้น การเข้ารหัสโครโมโซมจะต้องเป็นแบบรหัสค่าจริง (Real coded GA) เท่านั้นจึงจะทำการทดสอบนี้ได้ ไดกาลาคิส (Digalakis, 2000) ได้ทดสอบอัลกอริธึมพันธุการกับฟังก์ชันเทียบเคียงจำนวน 14 ฟังก์ชันที่เป็นทั้งมัลติโมเดล/ยูนิโมเดล และนอนลิเนียสแควร์ (Nonlinear square) การทดสอบได้มีการปรับเปลี่ยนพารามิเตอร์แบบสุ่มที่สำคัญ เช่นจำนวนประชากร อัตราการครอสโอเวอร์ อัตราการเกิดมิวเตชัน และเงื่อนไขในการหยุด ผลลัพธ์ของการทดสอบบางส่วนของบางฟังก์ชันที่เป็นมัลติโมเดล/ยูนิโมเดล ออกมาดังแสดงในตารางที่ 4.16

%ครอสโอเวอร์ ฟังก์ชัน	60	65	70	75	80	85	90	95
ทรงกลม	1.25	0.850	0.600	0.610	0.644	0.727	0.737	0.733
โรเซนบร็อก	0.8	0.740	0.400	0.405	0.410	0.510	0.420	0432
เรสตริจิน	15	13.6	12.2	8.8	8.81	8.82	9.1	8.82
กรีวังค์	1.8	1.2	2	0.7	0.71	0.72	0.73	0.74

ตารางที่ 4.16 ผลของการหาค่าเหมาะที่สุดต่อการเปลี่ยนแปลงอัตราการครอสโอเวอร์

จากผลการทดสอบการทำงานของอัลกอริธึมพันธุการ ในการหาค่าเหมาะสุดของฟังก์ชันเทียบเคียง (Benchmark Functions) ดังตารางที่ 4.16 ผลที่ได้ทั้งหมดจะเป็นค่าโลคอลมินิมั่ม ไม่มีค่าใดเลยที่ได้ผลลัพธ์ที่ดีที่สุด (เท่ากับ 0) โดยเฉพาะฟังก์ชันของเรสตริจินที่ค่าเหมาะที่สุดเป็นโลคอลมินิมั่มที่ห่างจากค่าที่ดีที่สุด (=0) มาก ฟังก์ชันง่ายๆ อย่างเช่นฟังก์ชันรูปทรงกลม ซึ่งมีโกลบอลมินิมั่มและไม่มีโลคอลมินิมั่มเลย ผลลัพธ์ที่ได้ออกมาก็ไม่ค่อยดีนัก โดยเฉพาะที่ช่วงอัตราการครอสโอเวอร์ต่ำ ฟังก์ชันของโรเซนบร็อกซึ่งเป็นฟังก์ชันที่มีโกลบอลมินิมั่มกว้างจะได้ผลดีที่สุดในการทดสอบครั้งนี้

แต่อย่างไรก็ตาม การทดสอบของไดกาลาคิสครั้งนี้ได้การนำเอาอัลกอริธึมพันธุการธรรมดา ที่มีวิธีการเข้ารหัสแบบค่าจริงมาทดลองด้วยการปรับค่าพารามิเตอร์แบบสุ่ม แต่เทคนิคที่สำคัญ เช่น

การครอสโอเวอร์แบบหลายพ่อแม่พันธุ์ หรือการทำงานแบบแบ่งกลุ่มประชากร ซึ่งจะช่วยในการเพิ่มสมรรถนะการค้นหา ยังไม่ได้นำมาใช้

4.8 สรุป

อัลกอริธึมพันธุการเป็นวิธีการการหาค่าที่เหมาะที่สุดของปัญหาที่มีขนาดใหญ่ เนื่องจากปัญหาบางประเภทมีปริภูมิปัญหาที่ใหญ่มากจนกระทั่งไม่สามารถหาคำตอบที่ดีที่สุดได้ อัลกอริธึมพันธุการถือเป็นอัลกอริธึมทางเลือกอันหนึ่งที่ใช้สำหรับการแก้ปัญหาประเภทนี้

อัลกอริธึมพันธุการ เป็นวิธีการที่การดัดแปลงมาจากกฎของการเลือกพันธุ์ทางธรรมชาติ โดยการจำลองความคิดมาจากวิธีการแลกเปลี่ยนยีนของโครโมโซม ที่เป็นการทำครอสโอเวอร์ และมิวเตชันเพื่อการสร้างประชากรใหม่ ด้วยวิธีการดังกล่าว ประชากรรุ่นใหม่จะค่อยๆ พัฒนาตนเอง จนกระทั่งได้ประชากรที่ดี ในทางธรรมชาติ การเกิดประชากรใหม่แต่ละรุ่นจะใช้เวลาจำนวนมาก แต่เมื่อนำแนวคิดนี้มาพัฒนาเป็นอัลกอริธึมสำหรับคอมพิวเตอร์ วิวัฒนาการนี้จะถูกเร่งความเร็ว เพื่อการหาค่าที่เหมาะที่สุดของปัญหาที่เป็นแบบนอนโพริโนเมียลชนิดยาก (NP Hard Problem) ซึ่งปัญหาลักษณะนี้ คำตอบที่เป็นไปได้จะมีจำนวนไม่รู้จบ

การทำงานของอัลกอริธึมจะมีกลไกที่สำคัญ 2 อย่างทำงานวนกันไปในการสร้างประชากรในแต่ละรุ่นคือ การเลือกสรรและการแลกเปลี่ยนยีน แต่ก่อนที่กลไกทั้งสองจะเริ่มทำงาน สิ่งที่ผู้แก้ปัญหาจะต้องตัดสินใจในการออกแบบอัลกอริธึมคือ วิธีการเข้ารหัสโครโมโซม และการสร้างประชากรเริ่มต้น

สำหรับวิธีการเข้ารหัสโครโมโซมมีอยู่หลายรูปแบบคือ การเข้ารหัสแบบไบนารี การเข้ารหัสแบบเปอร์มิวเตชัน การเข้ารหัสแบบค่า การเข้ารหัสแบบค่าจำนวนจริง และการเข้ารหัสแบบโครงสร้างต้นไม้ การเลือกวิธีเข้ารหัสจะขึ้นอยู่กับชนิดของปัญหาที่เราต้องการจะแก้ไข การเข้ารหัสแบบไบนารีเป็นการเข้ารหัสที่นิยมมากที่สุด เพราะการนำเสนออัลกอริธึมนี้ถูกนำเสนอพร้อมกับวิธีการเข้ารหัสแบบไบนารี ซึ่งสามารถใช้ได้กับปัญหาที่ค่อนข้างกว้างขวาง การเข้ารหัสเปอร์มิวเตชันจะเหมาะกับปัญหาประเภทการเรียงลำดับ เช่นการเรียงลำดับเมืองที่จะต้องเดินทางของปัญหาการเดินทางของเซลส์แมนและการจัดลำดับการทำงานของเครื่องจักรของปัญหาการจัดตารางเป็นต้น การเข้ารหัสแบบค่า เป็นการเข้ารหัสที่คล้ายกับแบบไบนารีแต่ต่างกันตรงที่ค่าของแต่ละบิตในสตริงแทนที่จะเป็น 0 และ 1 ในการ

เข้ารหัสแบบค่านั้น ค่าของแต่ละบิตในสตริงจะเป็นค่าที่เป็นตัวเลขระบุจำนวน ตัวอักษร หรือสัญลักษณ์ก็ได้ และสุดท้ายการเข้ารหัสแบบโครงสร้างต้นไม้ จะเหมาะกับปัญหาที่โครงสร้างมีความสัมพันธ์กันแบบลำดับชั้น (Hierarchy) เช่นสมการคณิตศาสตร์ เป็นต้น

เมื่อได้วิธีการเข้ารหัสที่เหมาะสมกับปัญหาแล้ว ให้เราสุ่มค่าให้กับทุกบิตในสตริง หรือโครโมโซมออกมาหลายๆ ชุดเพื่อเป็นประชากรเริ่มต้น

การเลือกสรร เป็นวิธีการทำสำเนาสตริงหรือโครโมโซมว่าแต่ละสตริงจะถูกสำเนาไว้จำนวนเท่าไรสำหรับการแลกเปลี่ยนยีนในขั้นต่อไป การเลือกสรรมีวิธีการที่สำคัญ 2 แบบคือ การเลือกสรรแบบวงล้อลูเล็ตต์ และการเลือกสรรแบบจัดลำดับ สำหรับการเลือกสรรแบบสเตดี สเทท (Steady-State Selection) และการเลือกสรรแบบอีลีทิซึม (Elitism Selection) เป็นการสร้างความลำเอียง (Bias) ให้กับการเลือกสรร 2 แบบแรก เพื่อทำให้สตริงที่มีค่าฟิตเน็สสูงมีโอกาสได้สืบทอดคุณสมบัติในรุ่นถัดไป

การแลกเปลี่ยนยีน มีกระบวนการที่สำคัญ 2 แบบคือ การครอสโอเวอร์ และการมิวเตชัน การครอสโอเวอร์จะเกิดขึ้นทุกรอบของการสร้างประชากรใหม่ ดังนั้นประชากรทุกรุ่นจะเกิดจากการครอสโอเวอร์ สำหรับการมิวเตอชันจะเกิดขึ้นไม่มาก ในการสร้างประชากรหลายๆ รุ่นจะมีการเกิดมิวเตชันสักหนึ่งครั้ง เช่น ประชากร 100 รุ่นจะเกิดการมิวเตชัน 1 รุ่นเป็นต้น และในแต่ละครั้ง มิวเตชันจะเกิดกับโครโมโซมบางตัวของประชากรในรุ่นนั้นเท่านั้น สำหรับเทคนิคของครอสโอเวอร์และมิวเตชันจะขึ้นกับลักษณะของการเข้ารหัสโครโมโซม

เอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน ของอัลกอริธึมเป็นเรื่องที่สำคัญต่อการการค้นหาคำตอบที่เหมาะสม ทั้งคอลสโอเวอร์ และมิวเตชันมีผลต่อเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน อัลกอริธึมพันธุการที่ดีจะต้องสร้างสมดุลของ เอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน

วิธีการแลกเปลี่ยนยีน ที่อาศัยการทำครอสโอเวอร์แบบหลายพ่อแม่พันธุ์ เป็นอีกวิธีหนึ่งของการทำให้การค้นหาคำตอบทำได้ดีขึ้น เพราะจะเป็นการปรับสมดุลระหว่างเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน วิธีหนึ่ง

ตัวอย่างที่บทนี้ได้นำมาแสดงเพื่อประกอบความเข้าใจในการแก้ปัญหาด้วยอัลกอริธึมพันธุการก็คือ การออกแบบผังงานสำหรับโรงงาน (Plant Layout)

การออกแบบแผนผัง (Layout) คือการจัดวางหน่วยงานต่างๆ ที่ต้องการให้อยู่ในบริเวณที่กำหนดโดยใช้พื้นที่น้อยที่สุด ภายใต้เงื่อนไขว่าความสัมพันธ์ระหว่างหน่วยงาน (Interrelationship) ทั้งหมดจะอำนวยประโยชน์สูงสุดต่อการใช้งานบนพื้นที่ที่น้อยที่สุด ซึ่งความสัมพันธ์ระหว่างหน่วยงานนี้หมายถึงความสัมพันธ์ที่หน่วยงานมีต่อกันในลักษณะใดลักษณะหนึ่ง ที่มีผลต่อการจัดวางหน่วยงานจะต้องวางใกล้หรือไกลกัน

โดยปกติอัลกอริธึมพันธุการ เป็นเมตาฮิวริสติกที่ออกแบบมาใช้กับปัญหาที่เป็นคอมบินาทอเรียล เช่น การเดินทางของเซลส์แมน และการจัดตาราง ดังที่ได้ยกตัวอย่างมา แต่อย่างไรก็ตาม การใช้อัลกอริธึมพันธุการสำหรับการแก้ปัญหาเชิงตัวเลขก็มีเช่นกัน ดังที่ได้กล่าวมาในเรื่องของการเข้ารหัสแบบค่าจำนวนจริง ไดกาลาคิส (Digalakis, 2000) ได้ทดสอบการทำงานของอัลกอริธึมพันธุการกับฟังก์ชันเทียบเคียง แต่ผลที่ได้จากการทดลองยังไม่ดีนัก อาจกล่าวได้ว่าอัลกอริธึมพันธุการที่มีการเข้ารหัสแบบค่าจริงยังไม่สามารถหาค่าที่ดีที่สุดได้

4.9 แบบฝึกหัด

1. ตามหลักการแล้ว อัลกอริธึมพันธุการทำงานอย่างไร ให้บอกความแตกต่างระหว่างอัลกอริธึมพันธุการกับการค้นหาแบบฮิวริสติกอื่น ในลักษณะดังต่อไปนี้
   1.1. ตัวดำเนินการของอัลกอริธึมพันธุการมีอะไรบ้าง และทำงานอย่างไร
   1.2. ฟิตเน็สฟังก์ชันคืออะไร และมีหน้าที่อย่างไรในอัลกอริธึมพันธุการ
   1.3. ค่าฟิตเน็ส และค่าการประเมิน (Evaluation value) ต่างกันอย่างไร
   1.4. คำว่า “รุ่น” ในอัลกอริธึมพันธุการคืออะไร
2. ถ้าเรามีประชากรเป็นโครโมโซมที่เป็นเลขไบนารีดังนี้

   A = 0 1 1 0 1

   B = 1 1 0 0 0

   C = 0 1 1 0 0

   D = 1 1 0 0 1

   ให้หาค่าที่ดีที่สุดของประชากรชุดนี้ โดยการใช้อัลกอริธึมพันธุการ และกำหนดค่าฟิตเน็สของโครโมโซมคือค่าที่เป็นเลขฐานสิบของเลขไบนารี และให้ทดลองแก้ปัญหานี้ด้วยการเขียนในกระดาษจำนวน 5 รุ่น ด้วยวิธีการดังต่อไปนี้
   2.1. ใช้วิธีการเลือกสรรแบบวงล้อรูเล็ตต์
   2.2. ใช้วิธีการเลือกสรรแบบจัดลำดับ
3. ถ้าเรามีประชากรเป็นโครโมโซมที่เป็นเลขไบนารีดังนี้

   A = 0 1 1 0 1

   B = 1 1 0 0 0

   C = 0 1 1 0 0

   D = 1 1 0 0 1

   3.1. ให้ทดลองใช้วิธีการเข้ารหัสแบบค่าจริง หาค่าที่แท้จริงของโครโมโซมข้างต้น
   3.2. ทำการครอสโอเวอร์ระหว่าง A กับ C และ B กับ D แล้วหาค่าจำนวนจริงออกมา
   3.3. หาค่าเพอร์เทอบเบชัน (Perturbation) ของการครอสโอเวอร์

3.4. ถ้าโครโมโซมข้างต้นเป็นรหัสเกร์ย ให้หาค่าจริงของโครโมโซมทั้งหมด และหาค่าหลังจากการทำครอสโอเวอร์ ของ A กับ C และ B กับ D

4. ถ้าเรามีประชากรเป็นโครโมโซมที่เป็นเลขไบนารีดังนี้

A = 0 1 1 0 1

B = 1 1 0 0 0

C = 0 1 1 0 0

D = 1 1 0 0 1

ให้ทำการครอสโอเวอร์แบบหลายพ่อแม่พันธุ์ แล้วหาค่าใหม่ของโครโมโซมลูก ของวิธีดังต่อไปนี้

4.1. ยูนิฟอร์ม สแกนนิ่ง

4.2. อ็อคเคอร์เร้นซ์ เบสด์ สแกนนิ่ง

4.3. ฟิตเน็ส เบสด์ สแกนนิ่ง

5. ถ้าเรามีประชากรของโครโมโซมดังนี้

A: 1 2 3 4 5 6 7

B: 2 4 5 1 3 7 6

C: 3 2 4 1 6 5 7

ให้หาโครโมโซมลูกด้วยวิธีการดังต่อไปนี้

5.1. อะแด็บติ่ง สแกนนิ่ง

5.2. แอ็ดจาเซนซี เบสด์ ครอสโอเวอร์

6. เมตริกซ์ค่าใช้จ่ายในการทำงานจาก (Brassard, 1996)

	1	2	3	4
A	11	12	18	40
B	14	15	13	22
C	11	17	19	23
D	17	14	20	28

6.1. ให้ออกแบบการเข้ารหัสโครโมโซมของปัญหาข้างต้น

6.2. ให้กำหนดขอบเขตของค่าในแต่ละบิตของสตริงที่ได้ออกแบบมา

6.3. ให้สร้างประชากรเริ่มต้นของปัญหาดังกล่าวพร้อมกับระบุวิธีการ และให้ใช้วิธีการเลือกสรรแบบวงล้อลูเล็ตต์ในการทำสำเนาสตริงจากประชากรเริ่มต้นนี้

6.4. อธิบายพร้อมแสดงตัวอย่างของการครอสโอเวอร์ของปัญหานี้

7. เอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชันของอัลกอริธึมพันธุการคืออะไร การทำครอสโอเวอร์ และมิวเตชันของอัลกอริธึมมีผลอย่างไรต่อเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน และอธิบายวิธีการควบคุมเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชัน ของอัลกอริธึมพันธุการมาหนึ่งวิธี
8. บีจีเอ (bGA) มีหลักการทำงานอย่างไร และมีผลอย่างไรต่อเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชันของอัลกอริธึมพันธุการ
9. ถ้านักศึกษาต้องการออกแบบอัลกอริธึมพันธุการสำหรับการหาเส้นทางที่สั้นที่สุดของเซลส์แมน โดยที่มีจำนวนเมืองที่จะเดินทางเป็น n และมีระยะทางระหว่างเมืองเป็น d_{ij} เมื่อ i และ j มีค่า 1 ถึง n ให้นักศึกษากำหนดค่าทางสถิติของพารามิเตอร์ต่างๆ ที่จะเป็นเอง โดยอธิบายวิธีการแก้ปัญหาตามขั้นตอนดังต่อไปนี้

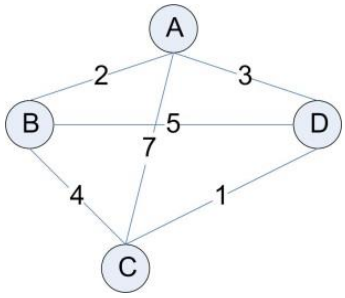
9.1. วิธีการออกแบบรหัสโครโมโซม

9.2. การกำหนดค่าในแต่ละบิตของโครโมโซมพร้อมทั้งบอกข้อจำกัด

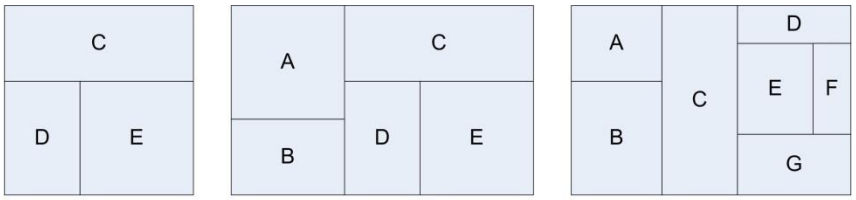
9.3. วิธีการในการเลือกสรรโครโมโซมแบบ Elitism ของปัญหานี้

9.4. วิธีการครอสโอเวอร์และมิวเตชัน

10. จากตัวอย่างการวางผังโรงงานโดยอาศัยอัลกอริธึมพันธุการดังที่แสดงไว้ข้างต้น แม็กซิมั่มเวทแมชชิ่งมีหน้าที่อะไรในการทำงานของอัลกอริธึม
11. จากกราฟข้างล่าง ให้แปลงเป็นโครงสร้างต้นไม้โดยใช้แม็กซิมั่มเวทแมชชิ่ง

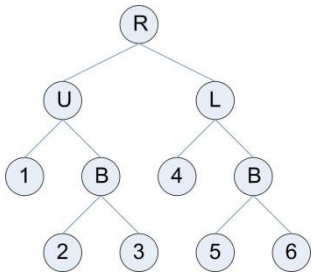


12. ให้เขียนสไลซิ่งทรีของผังงานต่อไปนี้



(1) (2) (3)

13. ให้แปลงสไลซิ่งทรีต่อไปนี้ให้เป็นผังงาน



14. การใช้สไลซิ่งทรีสำหรับการวางผังมีประโยชน์อะไรกับอัลกอริธึมพันธุการ และถ้าเราต้องการปรับวิธีการเข้ารหัสของตัวอย่างการออกแบบผังงานข้างต้นโดยใช้โครงสร้างต้นไม้แทน ให้อธิบาย และแสดงวิธีการตามหัวข้อดังต่อไปนี้

    14.1. รูปแบบของรหัสโครโมโซม และค่าในบิตต่างๆ ของโครโมโซม

    14.2. การครอสโอเวอร์และการมิวเตชันของโครโมโซม

บทที่ 5 การหาค่าเหมาะที่สุดแบบฝูงมด

Ant Colony Optimization

5.1 คำนำ

ในทศวรรษที่ผ่านมา การศึกษาพฤติกรรมของสัตว์เพื่อนำมาประยุกต์ใช้กับการทำงานของคอมพิวเตอร์เริ่มเป็นไปอย่างแพร่หลาย แม้ว่าการศึกษานี้ยังอยู่ในระยะเริ่มต้น แต่ผลการวิจัยก็เป็นที่นิยมอย่างมากและมีวิวัฒนาการย่างรวดเร็ว พฤติกรรมของสัตว์ เช่น การทำงานร่วมกันโดยเฉพาะการทำงานร่วมกันของแมลงที่เป็นสัตว์สังคมเป็นสิ่งที่แสดงให้เห็นถึงวิธีการจัดระบบการทำงานของสัตว์ทั้งฝูง ในฝูงสัตว์แต่ละฝูง สมาชิกแต่ละตัวต่างจะทำงานที่ตัวเองรับผิดชอบตามระบบที่วางไว้ในฝูง แล้วก่อให้เกิดความสำเร็จโดยรวมได้ เช่น มด ที่ทำการหาอาหาร มดแต่ละตัวต่างเดินทางออกจากรังไปยังแหล่งอาหาร เมื่อพบแหล่งอาหารที่ต้องการ มดก็จะระดมกันไปขนอาหารจากแหล่งอาหารนั้นเป็นฝูง ในระยะเริ่มแรก มดทั้งฝูงจะมีเส้นทางเดินจากรังไปยังแหล่งอาหารหลายเส้นทาง แต่เมื่อมดทำการขนอาหารไปได้ระยะเวลาหนึ่ง มดทั้งฝูงจะปรับเส้นทางเดินใหม่ให้ใช้เส้นทางที่ใกล้ที่สุดได้ โดยที่มดไม่มีหัวหน้าฝูงคอยกำกับการเดินทางของมดทุกตัว ในแง่ของการบริหารคือ มดไม่มีศูนย์กลางการควบคุมรวม (Central Control) ที่คอยกำกับการเดินของมดแต่ละตัว สัตว์ประเภทอื่นที่มีวิธีการทำงานร่วมกันในลักษณะเดียวกัน เช่น ผึ้ง ปลา และนก เป็นต้น แต่ละชนิดต่างก็ทำงานประสานกันเป็นกลุ่ม แต่ละตัวรับผิดชอบในภาระกิจของตนเองเพื่อให้ได้ผลลัพธ์ที่ดีร่วมกัน โดยไม่ต้องให้ใครเป็นผู้นำ การทำงานร่วมกันในลักษณะนี้จะเรียกว่า ปัญญาเชิงกลุ่ม (Swarm Intelligence)

5.1.1 พฤติกรรมของแมลง

การสังเกตการณ์การทำงานของแมลง ได้มีการทำมานานแล้ว นับตั้งแต่ปี 1959 นักกีฏวิทยาชื่อดังชาวฝรั่งเศสชื่อ ปิแอร์ พอล กราเซ (P.P. Grasse, 1959) ได้สังเกตเห็นพฤติกรรมของแมลงในฝูงที่ทำงานประสานกันเป็นกลุ่มและมีการสื่อสารระหว่างกันอย่างน่าอัศจรรย์ เขาเรียกการทำงานร่วมกันของแมลงในลักษณะนี้ว่า สติกเมอร์จี (Stigmergy) ซึ่งมาจากการผสมกันของคำในภาษากรีก 2 คำคือ

สติกมา (stigma หมายถึง mark, sign) และเออร์กอน (ergon แปลว่า work, action) กราเซ ได้บันทึกว่า ปลวกมีกฎง่ายๆ ในการสร้างรังดังนี้

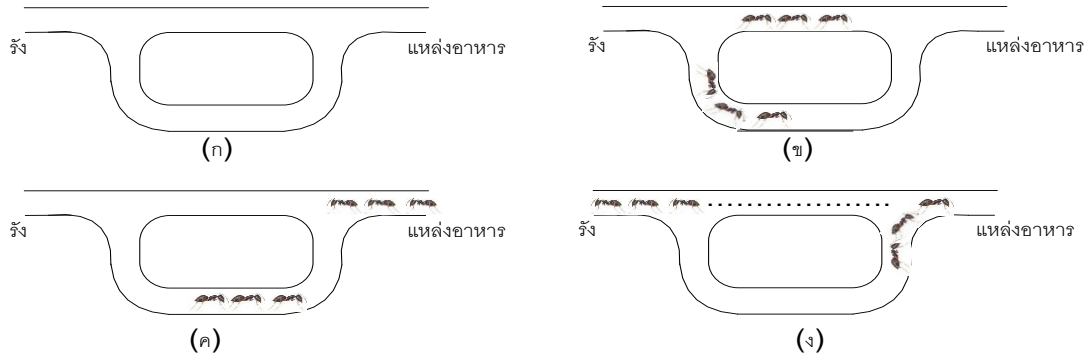
- เริ่มแรก มันจะเดินไปรอบอาณาบริเวณแบบสุ่มและคายมูลจากดินที่มันเคี้ยวผสมกับน้ำลายของมันบนเนินเตี้ยๆ ที่มันเจอ ทำให้เกิดกองดินขึ้นก่อนเล็กๆ
- กองดินที่เกิดจากน้ำลายปลวกเหล่านี้จะกระตุ้นให้ปลวกคายมูลเพิ่มลงในกองดินเดิม และขยายใหญ่ขึ้นเป็นรังปลวก จนได้ความสูงตามระดับสายพันธุ์ของปลวก
- และสุดท้าย ถ้ารังที่ถูกสร้างขึ้นอยู่ใกล้กับรังอื่นเพียงพอ พฤติกรรมอีกอย่างก็จะปรากฏขึ้นคือ ปลวกของแต่ละรังจะปีนขึ้นไปบนรัง และเริ่มที่จะสร้างรังเป็นมุมทแยงเข้าหากัน

จะเห็นได้ว่า กฎทั้งหมดนี้ได้แสดงให้เห็นถึงการใช้เชาว์ปัญญาร่วมกันของแมลงสังคม ในการสร้างรัง ปลวกไม่ได้มีการวางแผนการสร้างรังร่วมกันในตอนต้น แต่การทำงานของปลวกแต่ละตัวจะเป็นไปตามสถานการณ์ในขณะนั้น ปลวกไม่จำเป็นที่จะต้องมีความรู้เชิงภาพรวม (Global Knowledge) หรือต้องมีหน่วยความจำอื่นใดมากไปกว่าความจำที่จำเป็นในการทำงานตามหน้าที่นั้นให้เสร็จ ปลวกต้องการเฉพาะแรงกระตุ้นอย่างง่ายๆ ที่ขึ้นอยู่กับสภาพแวดล้อมในเวลานั้น กราเซ ตั้งคำว่า สติกเมอร์จี ซึ่งหมายถึง การกระตุ้นด้วยทำงาน และกระบวนการนี้ไม่ได้เกิดขึ้นเฉพาะกับปลวกเท่านั้น แต่ยังเกิดขึ้นกับ มด ผึ้ง และต่อ ในหลายๆ กิจกรรมจำนวนมากด้วย

สำหรับมดแล้ว แม้ว่าความสามารถของมดแต่ละตัวค่อนข้างจำกัด แต่มดแต่ละตัวสามารถทำงานร่วมกันเพื่อหาเส้นทางที่สั้นที่สุดจากรังไปยังแหล่งอาหารได้ มดสื่อสารกันเองโดยใช้สารเคมีที่มดผลิตขึ้นชื่อฟีโรโมน (Pheromones) เมื่อมดเดินทาง มดจะทิ้งฟีโรโมนไว้ตามทาง เพื่อเป็นสัญลักษณ์สำหรับมดตัวอื่นที่จะตามมาทีหลัง มดออกจากรังแบบไปทุกทิศทุกทาง แต่เมื่อพวกมันพบฟีโรโมนที่มดตัวอื่นทิ้งไว้ มันก็จะตัดสินใจว่าจะเดินตามหรือไม่ ถ้ามันเดินตามสัญลักษณ์นั้น มันก็จะทิ้งฟีโรโมนไว้ตามทางนั้นเพิ่ม ทำให้ความเข้มของฟีโรโมนนั้นมากขึ้น และทำให้มดตัวที่ตามมามีโอกาสที่จะเลือกเส้นทางนี้มากขึ้น ยิ่งมดใช้เส้นทางนี้มากขึ้นฟีโรโมนก็จะเข้มขึ้น โอกาสที่มดตัวถัดมาจะเลือกเส้นทางนี้ก็มากขึ้นเป็นเงาตามตัว

5.1.2 การทดลองกับสะพานคู่

ถ้าสมมุติว่ามี 2 เส้นทาง คือเส้นทางสั้นและยาว ที่จะเดินทางจากรังไปแหล่งอาหาร ในตอนเริ่มแรกฝูงมดจะเลือกใช้เส้นทางทั้งสองด้วยการสุ่ม ทำให้มดเดินบนเส้นทางทั้งสองด้วยจำนวนเท่าๆ กัน พร้อมกับทิ้งฟีโรโมนตามทางที่มดแต่ละตัวเดิน แต่มดที่เดินทางด้วยระยะทางที่สั้นกว่าจะกลับมาถึงรังก่อน ในการเดินไปยังแหล่งอาหารรอบถัดไป มดจะเลือกเส้นทางที่มีฟีโรโมนที่เข้มกว่า ทำให้มดที่มาถึงก่อน เดินทางไปยังแหล่งอาหารอีก และจะทิ้งฟีโรโมนเพิ่มตามเส้นทางที่เดินไป เนื่องจากระยะทางที่สั้นกว่า จึงทำให้เส้นทางนั้นมีโอกาสที่จะมีฟีโรโมนที่มากกว่าด้วย ซึ่งฟีโรโมนที่มากกว่านี้ก็จะดึงดูดมดตัวอื่นให้มาเดินตามเส้นทางนี้ และในที่สุดเส้นทางที่สั้นที่สุดก็จะได้รับการเลือกจากมดเกือบทั้งฝูง



รูปที่ 5.1 พฤติกรรมการเดินทางของฝูงมด

จาก

รูปที่ 5.1 แสดงการเดินทางของมดจากรังไปยังแหล่งอาหาร โดยที่กำหนดว่า เส้นทางจากรังไปยังแหล่งอาหารมีอยู่ 2 เส้นทางให้เลือกเดินคือ เส้นทางสั้น และเส้นทางยาว

รูปที่ 5.1(ก) เมื่อเริ่มต้น มดทั้งหมดจะอยู่ที่รัง และบนเส้นทางทั้ง 2 ยังไม่มีฟีโรโมน ดังนั้นเมื่อฝูงมดเริ่มเดินจากรังไปยังแหล่งอาหาร มดทั้งหมดจะถูกแบ่งออกเป็น 2 กลุ่มจำนวนเท่ากันที่จะเดินบนเส้นทางทั้ง 2 ตาม

รูปที่ 5.1(ข) มดที่เลือกเส้นทางที่สั้นกว่าจะถึงแหล่งอาหารก่อนมดที่เลือกเส้นทางที่ยาวกว่า ดัง

รูปที่ 5.1(ค) มดที่อยู่บนเส้นทางที่สั้นกว่าจะเริ่มเดินทางกลับ พร้อมกับทิ้งฟีโรโมนไว้ตามทาง ขณะที่มดที่เดินบนเส้นทางที่ยาวกว่าเพิ่งถึงแหล่งอาหาร ทำให้บนเส้นทางที่ยาวกว่ายังไม่มีฟีโรโมน ดัง

รูปที่ 5.1(ง) ในการทดสอบครั้งนี้ กำหนดว่าเมื่อมดถึงแหล่งอาหารแล้ว จะเดินทางกลับทางเดิมเสมอ และกำหนดให้มดวางฟีโรโมนเฉพาะขากลับเท่านั้น ดังนั้นมดที่จะเริ่มเดินไปยังแหล่งอาหารใหม่ก็จะเลือกเส้นทางเดินจากความแรงของกลิ่นฟีโรโมน เมื่อมีฟีโรโมนมาก กลิ่นก็แรงมาก ทำให้เส้นทางที่สั้นมีโอกาสที่จะถูกเลือกมากกว่า ยิ่งเมื่อมดเดินไปเดินมาหลายรอบ มดเกือบทั้งหมดก็จะหันมาเลือกเส้นทางที่สั้นกว่า

แต่อย่างไรก็ตาม จะมีมดบางตัวที่ไม่เดินตามกฎนี้ คือไม่เลือกเดินตามทางที่มีฟีโรโมนหนาแน่น ซึ่งในทางระบบถือว่าเป็นเรื่องที่ดีเพราะจะเป็นโอกาสที่มดตัวนี้ได้เส้นทางใหม่ที่ดีกว่า หรืออาจจะได้พบแหล่งอาหารใหม่ สิ่งที่น่าสนใจอีกประการหนึ่งคือ สำหรับฟีโรโมนที่ถูกทิ้งไว้ตามทางเดินนั้นจะระเหยไปตามกาลเวลาด้วย ระยะทางที่ยาวและต้องใช้เวลาในการเดินทางนาน ความหนาแน่นของฟีโรโมนก็จะค่อยๆ จางไปตามระยะเวลาที่ผ่านไป ทำให้เส้นทางที่ยาวไม่เป็นที่สนใจโดยอัตโนมัติ

จากผลการทดลองการหาอาหารของมดดังที่ได้กล่าวมาข้างต้น กอสส์ (S., Aron, S., Deneubourg, J.-L. and Pasteels, J.M. Goss, 1989) ได้พัฒนาโมเดลเพื่ออธิบายการทำงานของของมดแบบสะพานคู่ โดยสมมุติว่า เมื่อเวลาผ่านไป t นับตั้งแต่เริ่มต้นการทดลอง มีมดจำนวน m_1 ตัวเดินผ่านเส้นทางเส้นแรก และมีมดจำนวน m_2 ตัวเดินผ่านเส้นทางที่สอง ดังนั้นโอกาสที่มดตัวถัดไป (m+1) จะเดินผ่านทางเส้นแรกจะเท่ากับ

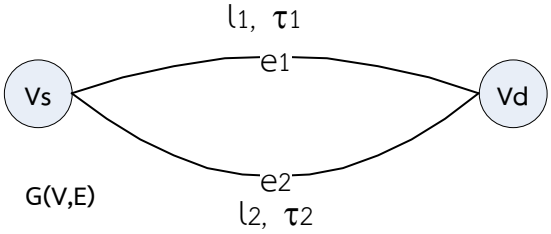
$$□_{1(□+1)} = \frac{(□_1+□)^□}{(□_1+□)^□+(□_2+□)^□} \qquad \text{สมการ 5.1}$$

จากสมการ 5.1 จะเห็นว่า โอกาสที่มดตัวที่ (m+1) จะเลือกเส้นทางแรกเท่ากับจำนวนฟีโรโมนที่อยู่บนเส้นทางแรก ซึ่งเป็นสัดส่วนกับจำนวนมดที่เดินผ่านเส้นทางแรก (m_1) หารด้วยจำนวนฟีโรโมนทั้งหมด หรือเป็นสัดส่วนกับจำนวนของมดทั้งหมด (m_1 และ m_2) ที่เดินผ่านเส้นทางทั้งสองไปแล้ว โดยที่ k และ h เป็นพารามิเตอร์ของโมเดลที่ได้จากการทดลอง พาสทีลส์ (Pasteels, 1987) ได้ทดสอบเพื่อหาค่าของ h และ k ที่เหมาะสมกับสมการ 5.1 และได้ผลออกมาเป็น $k \approx 20$ และ $h \approx 2$ ในทำนองเดียวกัน ถ้าให้ $p_{2(m+1)}$ คือโอกาสที่มดตัวที่ (m+1) จะเลือกเดินผ่านเส้นทางที่สอง ดังนั้น $p_{2(m+1)}=1-p_{1(m+1)}$

5.2 การหาค่าเหมาะที่สุดแบบฝูงมด

จากความเข้าใจการทำงานของแมลงในรูปแบบของ สติกเมอร์จี ทำให้ มาร์โค โดริโก (Dorigo, 1996) (Dorigo, 2004) ได้พัฒนาและนำเสนอวิธีการหาค่าเหมาะที่สุดแบบฝูงมด (Ant Colony Optimization ซึ่งเขียนตัวย่อเป็น ACO) ขึ้นในปี 1992 โดริโก (Dorigo, 1992) ได้พบรายละเอียดของการหาอาหารของมดว่า มดสามารถหาระยะทางที่สั้นที่สุดในการหาอาหารได้ โดยตอนเริ่มต้นของการหาอาหาร มดจะเดินสุ่มสำรวจรอบๆ รัง ในขณะที่มดเดิน มดจะทิ้งฟีโรโมนตามทางบนพื้นดิน มดทุกตัวที่เดินผ่านสามารถได้กลิ่นนี้ สำหรับการเลือกเส้นทางการเดินของมด มดจะเลือกเส้นทางที่มีกลิ่นฟีโรโมนแรง ทันทีที่มดพบแหล่งอาหาร มดจะบอกปริมาณและคุณภาพของอาหาร พร้อมกับนำส่วนหนึ่งของอาหารกลับรัง ในระหว่างที่มดเดินทางกลับรัง ปริมาณของฟีโรโมนที่มดทิ้งไว้ที่พื้น จะขึ้นอยู่กับปริมาณและคุณภาพของอาหารด้วย ร่องรอยของฟีโรโมนนี้จะเป็นตัวนำให้มดตัวอื่นๆ เดินไปแหล่งอาหาร ตามที่เราได้กล่าวมาแล้วว่า การสื่อสารกันโดยผ่านฟีโรโมนนี้เรียกว่า สติกเมอร์จี ซึ่งเป็นตัวที่ทำให้มดสามารถหาเส้นทางที่สั้นที่สุดจากรังไปยังแหล่งอาหารได้

เพื่อที่จะพัฒนาวิธีการหาเส้นทางเดินที่สั้นที่สุดจากรังไปยังแหล่งอาหารของมดเป็นอัลกอริธึม ในตอนแรกเราจะต้องสร้างโมเดลของฟีโรโมนที่ปรากฏอยู่บนเส้นทางเดิน ในรูปของกราฟ G(V,E) เมื่อ V คือเซทของโหนดที่ใช้แทนรังและแหล่งอาหาร และ E คือเซทของเส้นเชื่อมระหว่างโหนดที่ใช้แทนเส้นทางเดินของมด ตามที่แสดงในรูปที่ 5.2



รูปที่ 5.2 โมเดลการหาเส้นทางที่สั้นที่สุดจากรังไปยังแหล่งอาหารของฝูงมด

โมเดลนี้จะเป็นกราฟ G = (V,E) เมื่อ V คือโหนดที่มีอยู่ 2 โหนด ชื่อ v_s ใช้แทนรังของมด และ v_d ใช้แทนแหล่งอาหาร และ E คือลิงค์ที่มีอยู่ 2 ลิงค์ชื่อ e_1 และ e_2 ที่ใช้เชื่อมระหว่างโหนด v_s และ v_d สำหรับ e_1 กำหนดให้มีความยาวเท่ากับ l_1 และ e_2 ให้มีความยาวเท่ากับ l_2 โดยที่ $l_2 > l_1$ หรือจะกล่าว

อีกนัยหนึ่งว่า e_1 ใช้แทนระยะทางที่สั้นกว่าระหว่าง v_s และ v_d และ e_2 ใช้แทนระยะทางที่ยาวกว่าระหว่าง v_s และ v_d มดจะทิ้งฟีโรโมนบนเส้นทางที่มันเดินไป ให้ τ_i เป็นค่าของฟีโรโมนที่อยู่บนเส้นทาง e_i เมื่อ i=1,2 ตัว τ_i จะเป็นค่าที่บอกความเข้มของฟีโรโมน เรากำหนดให้ n_a คือมด เมื่อเริ่มออกจากรัง v_s มดแต่ละตัวจะเลือกเดินบนเส้นทาง e_1 หรือ e_2 ด้วยความน่าจะเป็นเท่ากับ

$$\square_{\square} = \frac{\square_{\square}}{\square_1+\square_2} \quad \square = 1,2$$ สมการ 5.2

เพื่อที่จะไปยังแหล่งอาหาร v_d จากสมการ 5.2 จะเห็นได้ว่า ถ้า τ_1 มากกว่า τ_2 โอกาสที่มดจะเลือกทางเดิน e_1 จะมากว่า และเช่นกัน ถ้า τ_2 มากกว่า τ_1 โอกาสที่มดจะเลือกทางเดิน e_2 ก็จะมากกว่า เมื่อมดเดินทางกลับจาก v_d มายัง v_s มดจะเลือกเดินกลับเส้นทางเดิมเสมอ และมดก็จะเปลี่ยนค่าของฟีโรโมนบนทางที่มันเดิน นั่นคือ เมื่อมดเดินทางบนทางเดิน e_i ค่าของฟีโรโมน τ_i ก็จะเปลี่ยนเป็น

$$\square_{\square}(\square + 1) \leftarrow \square_{\square}(\square) + \frac{\square}{\square_{\square}}$$ สมการ 5.3

ตัวแปร Q ของสมการ 5.3 เป็นพารามิเตอร์ของโมเดลที่เป็นค่าคงที่ ซึ่งแสดงถึงความแปรผันของปริมาณฟีโรโมนกับระยะทาง l_i ในอีกความหมายหนึ่งก็คือ ถ้าระยะทางยาว ปริมาณฟีโรโมนก็จะน้อย และในทางกลับกัน ถ้าระยะทางสั้น ปริมาณฟีโรโมนก็จะมาก

ในการหาอาหารของมดในแต่ละรอบ มดทุกตัวจะเริ่มต้นที่รัง v_s หลังจากนั้นมดทุกตัวจะเริ่มเดินจาก v_s ไปยังแหล่งอาหาร v_d ในระหว่างการเดินทาง มดจะมีการทิ้งฟีโรโมน และฟีโรโมนนี้ก็จะระเหยไปตามกาลเวลา สำหรับโมโดลของการระเหย สามารถสร้างเป็นสมการได้ดังนี้

$$\square_{\square}(\square + 1) \leftarrow (1 - \square) * \square_{\square}(\square) \quad \square = 1,2$$ สมการ 5.4

จากสมการ 5.4 สัญลักษณ์ ρ คือพารามิเตอร์ที่กำหนดอัตราระเหยของฟีโรโมนซึ่ง ρ จะมีค่าอยู่ระหว่าง 0 ถึง 1 หรือ $\rho \in (0,1)$ จากนั้นมดก็จะเดินทางกลับ และทิ้งฟีโรโมนไว้บนเส้นทางที่มันเดิน (ไปและกลับบนเส้นทางเดียวกัน)

ในการแปลงวิธีการหาอาหารของมดให้เป็นอัลกอริธึม เราต้องทำการปรับพฤติกรรมบางอย่างเพื่อทำให้การทำงานนั้นตรงกับลักษณะของคอมพิวเตอร์ในแง่ต่างๆ ดังนี้

- การเดินของมดจริงจะเป็นแบบอสมวาร (Asynchronous) ซึ่งหมายความว่า มดแต่ละตัวจะมีจังหวะก้าวในการเดินไม่เท่ากัน เมื่อเดินออกจากรังไปแล้ว มดบางตัวที่เดินออกไปก่อนอาจจะกลับมาทีหลัง ทั้งๆ ที่มดทั้งสองเดินทางอยู่บนเส้นทางเดียวกัน แต่ในขณะที่มดของอัลกอริธึมนี้จะมีการเดินทางเป็นแบบสมวาร (Synchronous) ซึ่งหมายความว่ามดทุกตัวจะมีจังหวะในการเดินเหมือนกันหมด นั่นคือมดจะใช้เวลาในการเดินไปเดินกลับบนเส้นทางเดียวกันด้วยเวลาที่เท่ากันหมดทุกตัว
- ในความเป็นจริง มดจะทิ้งฟีโรโมนทุกครั้งที่มันเดินทาง แต่สำหรับมดในอัลกอริธึมนี้ มดจะทิ้งฟีโรโมน เฉพาะช่วงการเดินทางกลับรังเท่านั้น
- ในระหว่างการหาอาหารของมดจริง การเลือกเส้นทางในการเดินจะขึ้นอยู่กับกลิ่นของฟีโรโมนที่อยู่บนเส้นทางนั้น เนื่องจากว่ามดจะทิ้งฟีโรโมนทุกครั้งที่มันเดิน จึงทำให้มดจริงหาเส้นทางที่สั้นที่สุดได้เร็ว แต่ในการทำงานของมดในระบบคอมพิวเตอร์ การเลือกเส้นทางจะประเมินจากปริมาณของฟีโรโมนที่ได้จากมดในช่วงที่มดเดินทางกลับรังเท่านั้น

5.2.1 เมตาฮิวริสติกของการหาค่าที่เหมาะที่สุดแบบฝูงมด

การเดินทางหาอาหารของฝูงมด สามารถนำไปใช้ในการแก้ปัญหาการหาค่าที่เหมาะที่สุดแบบคอมบินาทอเรียล (Combinatorial Optimization Problems: COP) ได้ โดริโกได้อธิบายการแก้ปัญหาโดยการนิยามปัญหาการหาค่าที่เหมาะที่สุดแบบคอมบินาทอเรียลไว้ว่า

นิยาม โมเดลของ COP สามารถนิยามได้เท่ากับ $P=(S,\Omega,f)$ โดยที่

- ค่า S เป็นปริภูมิการค้นหา (Search Space) ที่ประกอบไปด้วยเซทของตัวแปรเพื่อการตัดสินใจแบบดีสครีต (Discrete Decision Variables) X_i, i = 1,...,n;
- เซท Ω เป็นเซทของข้อจำกัด (Constraints) ในระหว่างตัวแปร
- f เป็นอ็อบเจ็คทีฟ ฟังก์ชัน เมื่อ f: $\rightarrow R_0^+$ มีค่าต่ำสุด

ตัวแปร X_i เป็นค่าที่ได้จาก $\{v_i^1,...,v_i^{|Di|}\}$ เมื่อ D_i เป็นจำนวนคำตอบของ v_i ที่เป็นค่าของตัวแปร X_i ถ้า s เป็นคำตอบที่เป็นไปได้ ดังนั้นค่า s ต้องอยู่ภายในปริภูมิการค้นหาของ S หรือ $s \in S$ และ s ต้องถูกกำหนดให้สอดคล้องกับเงื่อนไขของ Ω และ ถ้า $s^* \in S$ แล้ว s^* จะเป็นค่าที่เหมาะที่สุดแบบสากล (Global Optimal) ก็ต่อเมื่อ (if and only if) $f(s^*) \leq f(s)$ และ $\forall s \in S$.

โมเดลของ COP เมื่อนำมาใช้กับโมเดลของฟีโรโมนใน ACO ค่าของฟีโรโมนจะสอดคล้องกับองค์ประกอบของคำตอบที่เป็นไปได้ในแต่ละโมเดล ซึ่งก็คือ ค่าที่เป็นไปได้ของตัวแปรที่จะเป็นคำตอบให้กับการแก้ปัญหา ค่าฟีโรโมน τ_{ij} จะหมายถึงองค์ประกอบของคำตอบ c_{ij} ซึ่งปรากฏใน $X_i = v_i^j$ โดยที่เซทขององค์ประกอบที่ทำให้เกิดคำตอบที่เป็นไปได้ทั้งหมดคือ C ซึ่งเป็นเซทของคำตอบ c_{ij}

สำหรับการหาค่าเหมาะที่สุดแบบฝูงมดนั้น ฝูงมดสร้างคำตอบด้วยการเดินทางไปยังกราฟที่เชื่อมต่อกันทั้งหมดของ $G_C(V,E)$ เมื่อ V เป็นเซทของโหนด และ E เป็นเซทของเส้นเชื่อม กราฟนี้สามารถสร้างได้จากองค์ประกอบของคำตอบในเซท C ซึ่งมีอยู่ 2 รูปแบบคือ ชุดของโหนด และ ชุดของเส้นเชื่อม ฝูงมดจะเดินทางจากโหนดหนึ่งไปยังอีกโหนดหนึ่งโดยผ่านเส้นเชื่อม แล้วค่อยสร้างคำตอบออกมาเป็นส่วนๆ นอกจากนั้นแล้ว ฝูงมดยังทิ้งฟีโรโมนจำนวนหนึ่งบนเส้นเชื่อมในระหว่างการเดินทาง ซึ่งฟีโรโมนนี้ เป็นองค์ประกอบของคำตอบด้วย จำนวนขนาดของฟีโรโมน $\Delta\tau$ ที่ถูกทิ้งไว้จะขึ้นอยู่กับคุณภาพของคำตอบที่ค้นพบ และในเวลาเดียวกัน ฝูงมดจะใช้ฟีโรโมนเหล่านี้เป็นเครื่องนำทางในการหาคำตอบในปริภูมิการค้นหา จากลักษณะการทำงานของฝูงมดดังกล่าว โดริโก (M. Dorigo, 1992) ได้นำมาสร้างเป็นเมตาฮิวริสติก สำหรับการแก้ปัญหาเหมาะที่สุดแบบคอมบินาทอเรียล ดังนี้

อัลกอริธึม การหาค่าเหมาะที่สุดแบบฝูงมด (ACO)
ตั้งค่าพารามิเตอร์ และตั้งค่าเริ่มต้นให้กับค่าฟีโรโมน

while ยังไม่ถึงเงื่อนไขการสิ้นสุดการทำงาน
การสร้างคำตอบที่เป็นไปได้ (ด้วยวิธีการของฝูงมด)
การค้นหาคำตอบท้องถิ่น
การปรับค่าฟีโรโมน
end while

สำหรับอัลกอริธึมการหาค่าที่เหมาะที่สุดแบบฝูงมดนั้น เมื่อนำไปประยุกต์ใช้ในการแก้ปัญหาทั่วไป โดยการทำให้เป็นเมตาฮิวริสติก อัลกอริธึมนี้สามารถแบ่งการทำงานได้ออกเป็น 3 ขั้นตอน ดังกล่าวข้างต้นคือ การสร้างคำตอบที่เป็นไปได้ คือขั้นตอนการสร้างคำตอบที่เป็นไปได้ทั้งหมด ขั้นตอนต่อมาคือ การค้นหาคำตอบท้องถิ่น เป็นการค้นหาคำตอบที่ดีจากคำตอบที่เป็นไปได้ที่สร้างมาจากขั้นตอนการหาคำตอบที่เป็นไปได้ในขั้นตอนแรก โดยการพิจารณาจากค่าของฮิวริสติกฟังก์ชัน และ การปรับค่าฟีโรโมน (UpdatePheromone) เป็นขั้นตอนการปรับปรุงค่าของพารามิเตอร์ในฮิวริสติกฟังก์ชัน

5.2.2 ขั้นตอนการหาค่าเหมาะสุดแบบฝูงมด

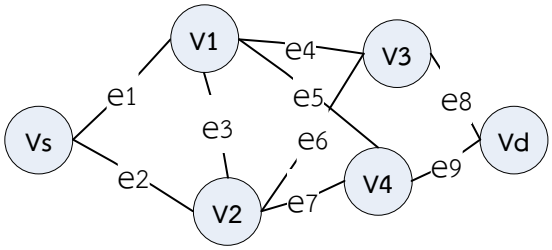
ขั้นตอนของเมตาฮิวริสติกในการหาค่าเหมาะที่สุดด้วยฝูงมดประกอบด้วย 3 ขั้นตอนใหญ่คือ

การสร้างคำตอบที่เป็นไปได้ กำหนดให้ m เป็นจำนวนมดที่สร้างเส้นทางเดินจากรังไปยังแหล่งอาหาร ให้ $C = \{c_{ij}\}$, $i = 1,\ldots,n$, $j = 1,\ldots,|D_i|$ เป็นเซทของคำตอบ (เส้นทาง/บางส่วนของเส้นทาง) ที่เป็นไปได้ทั้งหมด ในกระบวนการหาคำตอบที่เป็นไปได้นี้ การหาคำตอบจะเป็นไปแบบทีละขั้น แทนที่จะเป็นการหาเส้นทางทีเดียวจากรังไปยังแหล่งอาหาร ถ้าเราจินตนาการการเดินทางของมดบนกราฟ เมื่อมดอยู่ที่โหนดใดโหนดหนึ่ง และต้องเลือกทางเดินไปยังโหนดข้างเคียง เส้นเชื่อมที่เชื่อมจากโหนดที่มดอยู่ไปยังโหนดข้างเคียงทั้งหมดถือว่าเป็นคำตอบที่เป็นไปได้ $N(s^p)$ และ s^p เป็นเส้นทางที่มดเลือกเดิน ดังนั้นในการสร้างคำตอบที่เป็นไปได้ หรือการหาเส้นทางเดินจากรังไปยังแหล่งอาหาร เริ่มจากในช่วงแรกยังไม่มีส่วนของเส้นทางที่เป็นไปได้ใดปรากฏอยู่เลย นั่นคือ ตอนเริ่มต้น $s^p = \phi$ คือไม่มีคำตอบใดอยู่เลย ในการหาคำตอบที่เป็นไปได้ในแต่ละรอบ คำตอบบางส่วนที่เป็นไปได้ s^p ก็จะถูกเพิ่มเข้ามาด้วยการหามาจากเส้นทางบางส่วนที่เป็นไปได้ โดยกำหนดให้เส้นทางบางส่วนที่เป็นไปได้ที่มีอยู่ทั้งหมดคือ $N(s^p) \subseteq C$ แล้วนำเส้นทางบางส่วนที่ได้รับเลือกมาเพิ่มใน s^p โดยที่ยังเป็นไปตามเงื่อนไขของ Ω

กระบวนการสร้างคำตอบที่เป็นไปได้ทั้งหมด เทียบได้กับการเดินทางไปยังโหนดต่างๆ ของมดจากรังไปยังแหล่งอาหาร เพื่อสร้างออกมาเป็นกราฟ $G_C=(V,E)$

ในการเลือกคำตอบที่เป็นไปได้ s^p จาก $N(s^p)$ จะพิจารณาจากค่าของฟีโรโมนที่ปรากฏอยู่ใน s^p และกฎต่างๆ ที่ใช้ในการเลือก s^p กฎเหล่านี้จะมีความแตกต่างกันไปในแต่ละโมเดล แต่โดยทั้งหมดแล้วกฎเหล่านี้มักจะดัดแปลงมาจากพฤติกรรมของมดตามสมการ 5.1

จากกราฟรูปที่ 5.3 ในตอนแรกที่มดเริ่มเดินออกจากรัง Vs ค่าของ e1 และ e2 เป็นคำตอบบางส่วนของทั้งหมดที่เป็นไปได้ s^p ของเซท $N(s^p)$ ที่มดต้องเดินทางจาก Vs ไปยัง V1 ซึ่งเซท $N(s^p)$ เป็นเซทย่อยของ C โดยที่ C คือเซทของเส้นทางที่เป็นไปได้ทั้งหมดที่เดินจาก Vs ไปยัง Vd ผ่านโหนดต่างๆ คือ โหนด V1 โหนด V2 โหนด V3 และโหนด V4



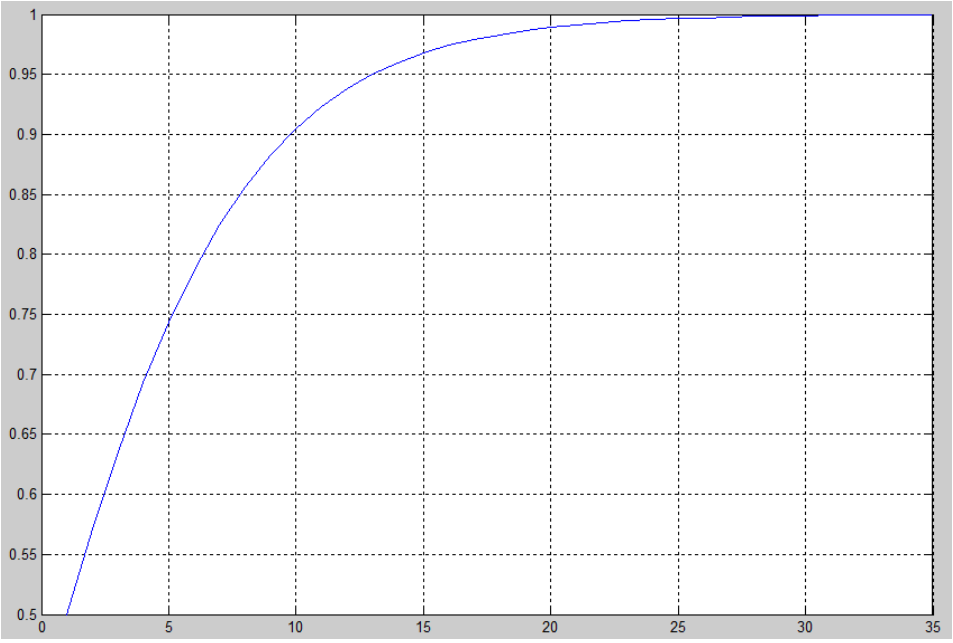
รูปที่ 5.3 กราฟ Gc(V,E) ของการเดินทางของมด

การค้นหาคำตอบท้องถิ่น เมื่อคำตอบที่เป็นไปได้ s^p ถูกสร้างออกมา อัลกอริธึมก็จะทำการปรับปรุงคำตอบที่เป็นไปได้ที่ได้มาจากการหาของมดในฝูงด้วยการค้นหาแบบท้องถิ่น (Local Search) ซึ่งกระบวนการนี้จะขึ้นอยู่กับลักษณะของปัญหาที่เราต้องการแก้ไขเป็นอย่างมาก

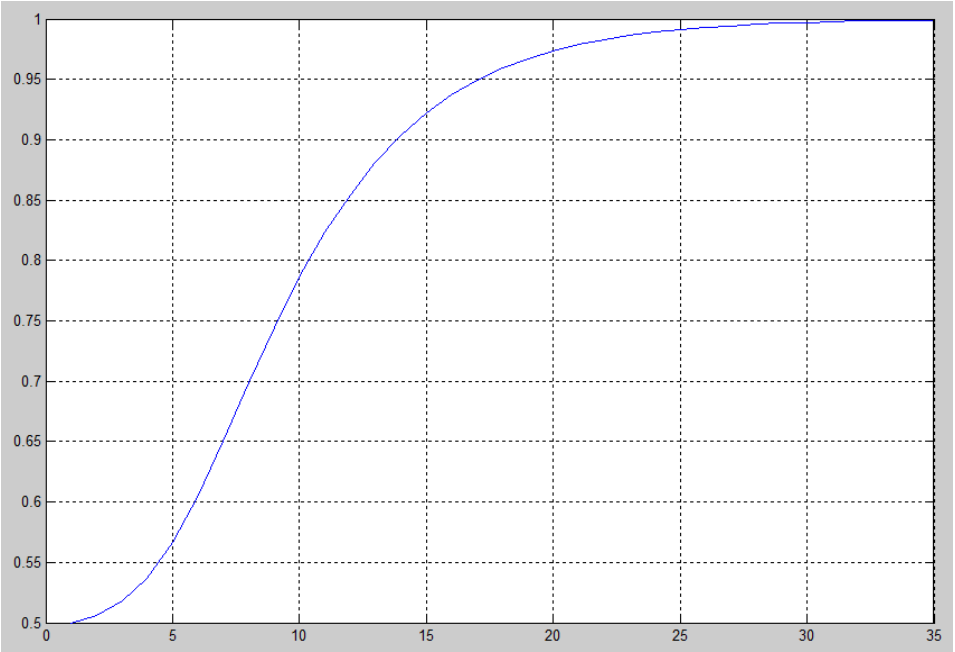
การปรับค่าฟีโรโมน การปรับปรุงค่าของฟีโรโมนเป็นการเร่งให้คำตอบที่ดีมีค่าฟีโรโมนเพิ่มเร็วขึ้น ในขณะเดียวกันก็จะเป็นการลดค่าฟีโรโมนของคำตอบที่ไม่ดี โดยปกติแล้วการปฏิบัติในขั้นตอนนี้จะแบ่งเป็น 2 ขั้นคือ การลดค่าฟีโรโมนของคำตอบที่เป็นไปได้ทุกค่า และการเพิ่มค่าฟีโรโมนให้กับคำตอบที่ดีเท่านั้น

5.2.3 การลู่เข้าหาคำตอบ

จากการพิสูจน์ของ โดริโกและสตูเซิล (Dorigo, 2004) พบว่า การแก้ปัญหาด้วยวิธีการหาค่าเหมาะที่สุดแบบฝูงมดนี้สามารถหาคำตอบได้เสมอ ด้วยการปรับค่าของฟีโรโมนตามเส้นทางที่ฝูงมดเดินผ่าน เมื่อมดเดินทางไปกลับระหว่างรังและแหล่งอาหารนี้ เมื่อถึงจุดหนึ่ง มดเกือบทั้งฝูงจะเลือกเส้นทางที่สั้นที่สุดด้วยความน่าจะเป็นเกือบเท่ากับ 1 ซึ่งเราเรียกกระบวนการที่มดค่อยๆ เปลี่ยนการตัดสินใจจากการเลือกหลายๆ เส้นทางจนเหลือเส้นทางเดียวนี้ว่า การลู่เข้าสู่คำตอบ (Convergence) ตามรูปที่ 5.4 เป็นกราฟการแสดงการลู่เข้าสู่คำตอบของการเลือกเส้นทางของฝูงมดที่เดินบนสะพานคู่ กราฟนี้แสดงการเลือกเส้นทางของฝูงมดจากการทดลองแบบสะพานคู่โดยใช้โปรแกรม Mathlab ที่เทียบอัตราส่วนของมดที่เลือกเส้นทางที่ 1 กับจำนวนรอบของการเดินทางของมด



(ก) มด
จำนวน 50
ตัว



(ข) มด
จำนวน 5 ตัว

รูปที่ 5.4 กราฟแสดงการเลือกเส้นทางเดินของมดบนทางเดินแบบสะพานคู่

รูปที่ 5.4(ก) มดจำนวน 50 ตัว รูปที่ 5.4(ข) มดจำนวน 5 ตัว การทดลองทั้งสองกำหนดให้มีฟีโรโมนตอนเริ่มต้นเท่ากัน เท่ากับ 0.5 แกน x เป็นจำนวนรอบของการเดินของฝูงมด แกน y แสดงอัตราส่วนของมดที่เลือกเส้นทางที่สั้นกว่า ในรูปที่ 5.4(ก) มดทั้ง 50 ตัวจะเลือกเส้นทางที่ 1 เมื่อมดเดินทางผ่านไปประมาณ 34 รอบ สำหรับการทดลองที่ 2 ในรูปที่ 5.4(ข) มดทั้งหมดจะเลือกสะพานที่ 1 เมื่อมดเดินทางผ่านไปประมาณ 36 รอบ จากการทดลองเมื่อเปลี่ยนจำนวนมดที่ใช้สำหรับหาคำตอบของสะพานคู่ ผลจะเป็นดังตารางที่ 5.1

จำนวนมด	2	5	10	20	50	100	150	200
จำนวนรอบ	38	36	35	34	34	33	32	32
P_i	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994

ตารางที่ 5.1 การเปรียบเทียบจำนวนมด กับจำนวนรอบที่มดจะต้องใช้ในการเดินทาง เพื่อให้มดร้อยละ 99.94 เดินจากรังไปยังแหล่งอาหารโดยใช้เส้นทางที่ 1

จากการที่เรานับจำนวนรอบที่มดเลือกเส้นทางที่ 1 ตามที่กล่าวมาข้างต้น เนื่องจากว่าเส้นทางที่ 1 เป็นเส้นทางที่เป็นคำตอบที่เราต้องการ ดังนั้นถ้ามดทั้งหมดของการทดลองเลือกเส้นทางที่ 1 เราก็มั่นใจได้ว่าเส้นทางที่ 1 เป็นคำตอบที่ถูกต้อง แต่เนื่องจากว่ามดต้องใช้การเดินไปเดินมาหลายรอบกว่าจะตัดสินใจเลือกเส้นทางที่ถูกต้อง และถ้าเราสังเกตจากกราฟตามรูปที่ 5.4 จำนวน p_i ค่อยๆ เพิ่มขึ้น เมื่อจำนวนรอบในการเดินมากขึ้น ลักษณะของการที่มดค่อยๆ เพิ่ม p_i เพื่อเลือกคำตอบที่ถูกต้อง หรือการลู่เข้าหาคำตอบ (Convergence) นี้ การวัดการลู่เข้าสู่คำตอบจะใช้จำนวนรอบของการเดินไปกลับระหว่างรังกับแหล่งอาหาร การลู่เข้าสู่คำตอบที่ดีจะมีจำนวนรอบการเดินทางไปกลับที่น้อย การวัดการลู่เข้าสู่คำตอบนี้คืออัตราการลู่เข้าสู่คำตอบ (Rate of Convergence) จากรูปที่ 5.4 และตารางที่ 5.1 แสดงให้เห็นว่าจำนวนมดที่ใช้ในการเดินทางจะมีผลต่ออัตราการลู่เข้าสู่คำตอบ จำนวนมดที่มากขึ้นจะมีผลต่ออัตราการลู่เข้าสู่คำตอบที่ดีขึ้น แต่ในขณะเดียวกัน เมื่อมีจำนวนมดมากถึงจำนวนหนึ่ง (> 150 ของตารางที่ 5.1) อัตราการลู่เข้าสู่คำตอบจะไม่ดีขึ้นอีก

ในกรณีที่ทางเดินมีมากกว่า 2 เส้นทาง จำนวนมดที่กระจายไปตามเส้นทางต่างๆ เพื่อทำการค้นหาจะทำได้ดีขึ้น สมมุติว่าในการสำรวจเส้นทาง 10 เส้นทาง แต่มีมด 10 ตัว มดทั้ง 10 ตัวนี้จะต้องทำการสำรวจเส้นทางทั้ง 10 เส้นทาง แล้วปล่อยฟีโรโมนตามเส้นทางต่างๆ จากนั้นก็จะย้อนกลับไปยังแหล่งอาหาร จนกว่าฟีโรโมนจะแสดงออกให้เห็นเส้นทางที่สั้นที่สุด ซึ่งก็จะใช้เวลายาวนานมาก ในขณะเดียวกัน ถ้ามีมด 100 ตัวช่วยกันสำรวจ การสำรวจจะทำได้เร็วขึ้น และถ้าเส้นทางทั้ง 10 มีทางแยกอีกมากมาย ก็จะทำให้การค้นหามีเส้นทางเลือกจำนวนมหาศาล แน่นอนการสำรวจบนเส้นทางได้ทุกเส้นทางจะทำให้เราได้คำตอบที่ดีที่สุดได้ แต่ถ้าทางเลือกมีจำนวนมหาศาลจนการสำรวจทุกเส้นทางทำไม่ได้ บางครั้งเราอาจจะหาคำตอบได้โดยที่บางเส้นทางยังไม่ได้รับการสำรวจเลย อย่างไรก็ตามการสำรวจจำนวนเส้นทางที่มากกว่าย่อมทำให้เรามั่นใจว่าจะได้คำตอบที่ดีกว่า แต่ก็ต้องใช้เวลาในการค้นหาคำตอบที่มากขึ้น

การกำหนดอาณาบริเวณในการค้นหาจากคำตอบที่เป็นไปได้ทั้งหมดเป็นสิ่งที่จำเป็น การที่มดทั้งหมดทำการสำรวจบนขอบเขตของคำตอบที่เป็นไปได้ จะเรียกว่าเอ็กซ์พลอเรชัน (Exploration) หรือกล่าวอีกนัยหนึ่งคือ การสำรวจหมายถึงความสามารถในการกำหนดขอบเขตของอาณาบริเวณบนปริภูมิปัญหา (Problem Space) ในการค้นหาคำตอบ เพื่อช่วยลดปัญหาโลคอลอ็อปติมั่ม (Local Optimum)

บนอาณาบริเวณที่การสำรวจได้กำหนดขอบเขตเอาไว้แล้วนี้ ถ้าการหาคำตอบที่เป็นโกลบอลอ็อปติมั่ม (Global Optimum) จะเรียกว่า เอ็กซ์พล้อยเตชัน (Exploitation) ดังนั้น เอ็กซ์พล้อยเตชัน จะหมายถึงความสามารถในการหาโกลบอลอ็อปติมั่มบนอาณาบริเวณที่การสำรวจได้กำหนดไว้

สำหรับเมตาฮิวริสติกการหาค่าเหมาะที่สุดแบบฝูงมดนี้ ขั้นตอนการสร้างคำตอบที่เป็นไปได้จะเป็นขั้นตอนเอ็กซ์พลอเรชัน ซึ่งเป็นขั้นตอนที่มดทั้งฝูงหาเส้นทางเดินจากรังไปยังแหล่งอาหาร เส้นทางนี้จะมีจำนวนมาก และทุกเส้นทางจะถูกกำกับไว้ด้วยฟีโรโมนที่บอกความสำคัญของเส้นทางนั้น สำหรับมดตัวต่อไปที่จะตัดสินใจในการเลือกทางเดิน เส้นทางเดินเหล่านี้จะมีฟีโรโมนกระจายอยู่ในรูปแบบของการแจกแจงความน่าจะเป็น (Probability Distribution) โดยที่เส้นทางที่ดีก็จะมีความน่าจะเป็นสูง รูปแบบของการแจกแจงความน่าจะเป็นนี้มีได้หลายแบบ เช่น แบบยูนิฟอร์ม (Uniform distribution) และแบบดีเจเนอเรต (Degenerate Distribution) เป็นต้น

เอ็กซ์พล้อยเตชันของการหาค่าเหมาะที่สุดแบบฝูงมดนี้คือ กระบวนการปรับค่าฟีโรโมน การปรับปรุงนี้เป็นการวางน้ำหนักของการค้นหาให้มุ่งไปสู่เส้นทางที่ดี โดยที่เส้นทางที่ดีจะมีค่าฟีโรโมนสูงขึ้นและเส้นทางที่แย่จะมีจำนวนฟีโรโมนต่ำลง

การสำรวจที่ดีไม่ได้หมายถึงความสามารถในการหาอาณาบริเวณเพื่อการค้นหาที่กว้าง แต่เป็นการกำหนดบริเวณที่เหมาะสมสำหรับเอ็กซ์พล้อยเตชันเพื่อใช้ในการหาคำตอบที่ดีที่สุด บนพื้นฐานของอัตราการลู่เข้าสู่คำตอบที่เร็วด้วย ซึ่งเป็นการสร้างสมดุลของเอ็กซ์พลอเรชันและเอ็กซ์พล้อยเตชัน

5.3 ตัวอย่างการหาค่าเหมาะที่สุดแบบฝูงมด

เพื่อที่จะทำให้เกิดความเข้าใจอย่างเป็นรูปธรรม ในการประยุกต์ใช้งานนี้จะนำเสนอตัวอย่างในการนำวิธีการหาค่าเหมาะที่สุดแบบฝูงมดไปใช้ในการแก้ปัญหาการเดินทางของเซลส์แมน การหาเส้นทางเดินรถ และการวางผังโรงงาน

5.3.1 การเดินทางของเซลส์แมน

การนำวิธีการหาอาหารของฝูงมดมาช่วยในการแก้ปัญหาการเดินทางของเซลส์แมน เป็นเรื่องแรกที่ถูกนำเสนอมาเป็นตัวอย่างเพื่อให้เข้าใจถึงวิธีการประยุกต์ใช้งาน อย่างไรก็ตาม การนำมาใช้จะต้อง

เป็นการประยุกต์ เพราะการหาค่าที่เหมาะที่สุดแบบฝูงมดดังตัวอย่างที่นำเสนอมานั้นเป็นการหาเส้นทางบนเส้นทางที่เรารู้คำตอบอยู่แล้วว่าเส้นทางไหนถูก แต่กับการแก้ปัญหาการเดินทางของเซลส์แมนเราจะไม่รู้คำตอบล่วงหน้าว่าเส้นทางไหนเป็นเส้นทางที่สั้นที่สุด

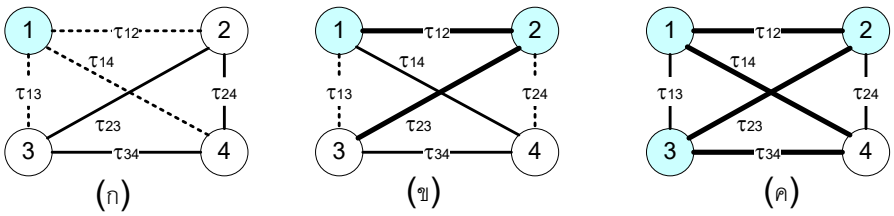
นิยามของปัญหาการเดินทางของเซลส์แมน โดยทั่วไปแล้ว ลักษณะของปัญหาการเดินทางของเซลส์แมนนั้น เซลส์แมนจะเริ่มต้นการเดินทางจากบ้านเกิดไปขายของตามเมืองต่างๆ และเดินทางกลับมายังบ้านเกิดอีกครั้ง การเดินทางนี้จะต้องเดินทางไปยังเมืองต่างๆ เพียงครั้งเดียว และต้องให้ได้ระยะทางที่สั้นที่สุดด้วย การนิยามที่เป็นทางการจะเป็นการแปลงลักษณะของปัญหาให้อยู่ในรูปของกราฟสมบูรณ์ (Complete Graph) G(V,E) โดยที่ V เป็นเซทของโหนดที่ทำหน้าที่แทนเมืองต่างๆ ที่เซลส์แมนจะต้องเดินทางไป สำหรับ E คือเซทของเส้นเชื่อมที่ใช้แทนเส้นทางเดินระหว่างเมือง สำหรับเส้นเชื่อมแต่ละเส้น e(i,j) $\in$ E เป็นค่าของระยะทางจากเมือง i ไปยังเมือง j ที่แทนด้วย d_{ij} และทั้ง i และ j เป็นสมาชิกของ V หรือ i, j $\in$ V สำหรับปัญหาเรื่องระยะทางระหว่างเมืองของการเดินทางจาก i ไป j จะเท่ากับจาก j ไป i หรือ $d_{ij} = d_{ji}$ ของทุกเส้นทางที่อยู่ใน E

เป้าหมายของการแก้ปัญหาการเดินทางของเซลส์แมนคือ การหาระยะทางที่สั้นที่สุดของวงจรฮามิลโตเนียน (Hamiltonian Circuit) ของกราฟ ซึ่งวงจรฮามิลโตเนียนเป็นเส้นทางที่เชื่อมกับทุกโหนด n ของเซท N ในกราฟ G โดยที่แต่ละโหนดจะถูกเชื่อมผ่านเพียงครั้งเดียว

ในการแก้ปัญหาการเดินทางของเซลส์แมนด้วยวิธีการหาค่าที่เหมาะที่สุดแบบฝูงมด จะเป็นการทำงานแบบวนลูปของการสร้างเส้นทางเดินทีละส่วน ร่วมกับการปรับค่าฟีโรโมน (Blum, 2005) ซึ่งมีรายละเอียดดังนี้

การสร้างคำตอบที่เป็นไปได้ (Tour construction) มดแต่ละตัวสร้างคำตอบตามลำดับดังนี้ ในขั้นตอนแรก โหนดใดโหนดหนึ่งในกราฟจะถูกเลือกออกมาเป็นเมืองเริ่มต้นด้วยการสุ่ม หลังจากนั้นมดจะสร้างเส้นทางในกราฟโดยการเคลื่อนที่จากโหนดปัจจุบันไปยังโหนดที่ยังไม่ได้ไป ในการเดินไปยังโหนดแต่ละโหนด ระยะทางของโหนดเหล่านั้นก็จะถูกนำมาพิจารณาในคำตอบที่เรากำลังสร้างขึ้น จนเมื่อเราไม่มีเมืองที่ยังไม่ได้ไปอีก มดก็จะเดินทางจากเมืองที่มันอยู่ในปัจจุบันไปยังเมืองเริ่มต้นของการเดินทาง

รูปที่ 5.5 การสร้างคำตอบในลักษณะนี้หมายความว่า มดมีหน่วยความจำ T ที่จะจดจำเมืองที่มันได้ไปมาแล้วได้ ถ้าเรากำหนดว่ามดทั้งหมดมี k ตัว มดทั้ง k ตัวจะต้องดำเนินการหาเส้นทางสำหรับตัวเอง ตามขั้นตอนดังกล่าวข้างต้น เพื่อสร้างคำตอบที่เป็นไปได้ทั้งหมดออกมา



รูปที่ 5.5 การแก้ปัญหาการเดินทางของเซลส์แมนด้วยการหาค่าที่เหมาะที่สุดแบบฝูงมด สำหรับมด 1 ตัวที่เริ่มต้นเดินทางจากเมือง (1)

สำหรับการสร้างคำตอบในแต่ละขั้นจะมีขั้นตอนดังต่อไปนี้

สมมุติว่ามดอยู่ที่โหนด v_i จากนั้นพิจารณาว่ามดจะเลือกเดินเส้นทางใด เพื่อที่จะเดินไปยังเมืองถัดไปด้วยการหาค่าทางสถิติดังต่อไปนี้

$$\square(\square_{\square\square}) = \frac{\square_{\square\square}(\square)^{\square} * \square_{\square\square}^{\square}}{\sum_{\square\square\square\square}^{\square}(\square_{\square\square}(\square)^{\square} * \square_{\square\square}^{\square}} \quad \square\square\square_{\square}^{\square} \qquad \text{สมการ 5.5}$$

สมการ 5.5 ปรับมาจากสมการ 5.2 เป็นการหาความเป็นไปได้ของการเลือกทางเดิน e_{ij} คือเดินจากโหนด v_i ไปยัง v_j เมื่อ η_{ij} เป็นค่าข้อมูลฮิวริสติก ที่ใช้สำหรับปรับน้ำหนักของค่าฟีโรโมนที่อยู่บนเส้นทางนั้น ค่านี้จะแปรผกผันกับระยะทางคือ $\eta_{ij}=1/d_{ij}$ สำหรับค่า α และ β เป็นค่าพารามิเตอร์ที่กำหนดความสัมพันธ์ของค่าฟีโรโมน (τ) และค่าข้อมูลฮิวริสติก (η) ว่าจะมีน้ำหนักที่เป็นสัดส่วนต่อกันเท่าไร สำหรับบทบาทของค่า α และ β ก็คือ เมื่อ α = 0 เมืองที่อยู่ใกล้จะมีโอกาสได้รับเลือก ถ้า β = 0 โอกาสของเส้นทางที่มีฟีโรโมนมากจะได้รับเลือก และสุดท้าย เซทของ N_i^k เป็นโหนดเพื่อนบ้านที่เป็นไปได้ทั้งหมดของมด k ในเซท N_i หรืออาจกล่าวในอีกทางหนึ่งได้ว่า N_i^k ก็คือเมืองทั้งหมดในเซท N ที่มดตัวที่ k ยังไม่ได้เดินผ่าน เมื่อพิจารณาจากโหนด v_i เมื่อเสร็จขั้นตอนนี้ เราจะได้เส้นทางที่เป็นไปได้ทั้งหมด k เส้นทาง

การปรับค่าฟีโรโมน (Pheromone update) เมื่อมดทุกตัวเสร็จสิ้นการสร้างคำตอบของมันเอง จะเกิดการปรับค่าฟีโรโมนขึ้น การปรับค่าฟีโรโมนจะมีอยู่ 2 ส่วนคือ ค่าที่ลดลงเนื่องจากการระเหยของฟีโรโมนและการเพิ่มของฟีโรโมนเนื่องจากมดได้ปล่อยฟีโรโมนออกมาเพิ่มเติมบนเส้นทางที่มดเดินผ่าน ซึ่งเป็นไปตามสมการ 5.6 ดังนี้

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \qquad \text{สมการ 5.6}$$

เมื่อ $0<\rho\leq 1$ เป็นอัตราการระเหยของฟีโรโมน ค่าอัตราการระเหยของฟีโรโมนนี้มีไว้เพื่อจำกัดการเพิ่มของฟีโรโมนบนเส้นทางของมด ซึ่งเป็นการทำให้ฟีโรโมนบนเส้นทางไม่ดีที่ได้รับการเลือกไว้ในตอนแรกให้เลือนหายไป และถ้าเส้นทางนั้นไม่ได้รับการเลือกโดยมดเลย ค่าฟีโรโมนก็จะเลือนหายไปอย่างรวดเร็ว สำหรับค่า $\Delta\tau^{k}_{ij}(t)$ เป็นจำนวนฟีโรโมนที่เพิ่มขึ้นเนื่องจากมดตัวที่ k ที่นำไปทิ้งไว้บนเส้นทางที่มันเดินผ่าน ซึ่งถูกกำหนดไว้ดังนี้

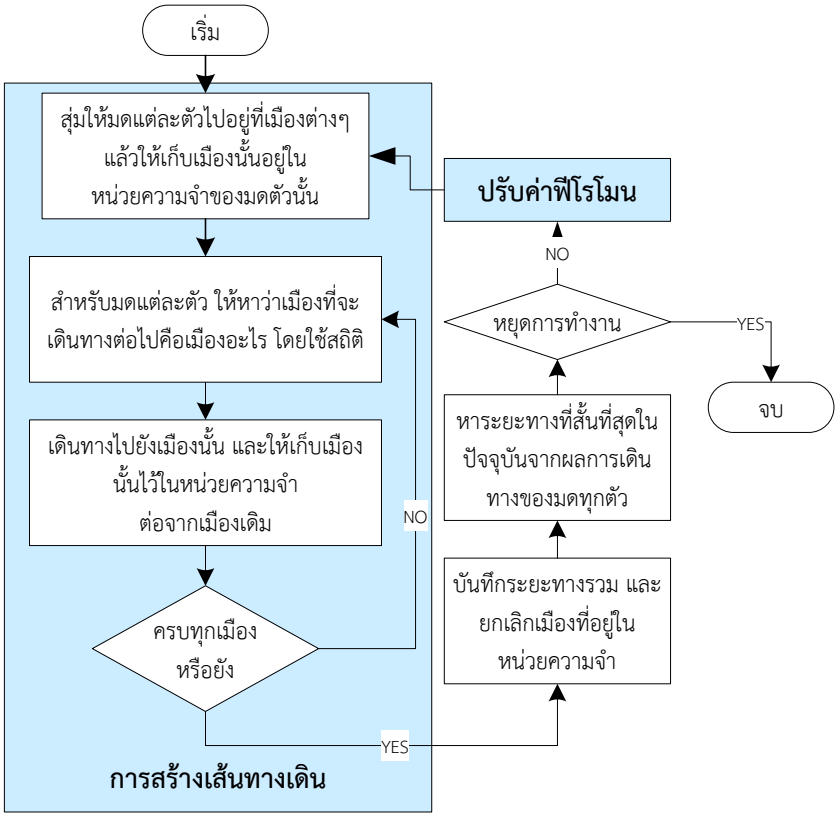
$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{1}{L^{k}}(t) & \text{ถ้า } (i,j) \in \text{เส้นทาง ที่ มด ตัว } k \\ 0 & \text{ถ้าไม่เป็นเช่นนั้น} \end{cases} \qquad \text{สมการ 5.7}$$

ค่า L^k เป็นระยะทางที่มดตัวที่ k เดินผ่าน ที่คำนวณโดยการบวกระยะทางระหว่างเมืองทั้งหมดที่มดเดินจากเมืองเริ่มต้นผ่านเมืองต่างๆ จนครบ แล้วกลับมายังเมืองเดิมอีกครั้ง สำหรับเส้นทางต่างๆ นั้นเส้นทางที่ดีจะมีจำนวนฟีโรโมนมาก และเส้นทางที่มดจำนวนมากเดินผ่านก็จะมีจำนวนฟีโรโมนมากเช่นกัน และก็มีโอกาสที่อัลกอริธึมจะเลือกเส้นทางนั้น

วิธีการแก้ปัญหาการเดินทางของเซลส์แมน อัลกอริธึมของการแก้ปัญหาการเดินทางของเซลส์แมนนั้นมีการนำเสนออยู่หลายวิธี สำหรับอัลกอริธึมที่นำเสนอเป็นเทคนิคหนึ่งที่ถูกนำเสนอโดยโดริโก (Dorigo, 2004) โดยมีรายละเอียดที่ประกอบด้วย

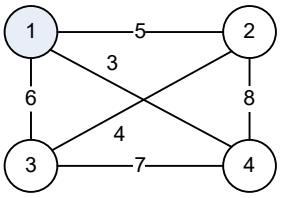
- การสร้างเส้นทางเดิน และ
- การปรับฟีโรโมน

ในส่วนของการสร้างเส้นทางเดินจะเริ่มด้วยกระบวนการสุ่มให้มดแต่ละตัวไปอยู่ที่เมืองต่างๆ จากนั้นก็เริ่มเดินทางจากเมืองที่มดอยู่ไปยังเมืองถัดโดยใช้สมการ 5.5 เพื่อการเลือกเมืองตามที่ได้เสนอไปแล้ว และเมืองที่จะเดินไปนั้นต้องเป็นเมืองที่มดตัวนั้นยังไม่ได้เดินผ่านมาก่อน จากนั้นเมื่อมดเดินจนครบทุกเมือง ก็จะทำการปรับฟีโรโมนตามเส้นทางที่มดเดินผ่าน โดยใช้สมการ 5.6 และ 5.7 เมื่อมดเดินจบรอบแล้ว ให้มดแต่ละตัวไปเริ่มต้นใหม่เพื่อเดินไปยังเมืองต่างๆ อีกรอบ ทำเช่นนี้ไปเรื่อยๆ จนได้คำตอบที่ต้องการ สำหรับรายละเอียดสามารถดูการทำงานได้จากรูปที่ 5.6



รูปที่ 5.6 การทำงานของฝูงมดในการแก้ปัญหาการเดินทางของเซลส์แมน

ตัวอย่างการแก้ปัญหา เพื่อที่จะทำความเข้าใจในการแก้ปัญหาการเดินทางของเซลส์แมนด้วยวิธีการหาค่าที่เหมาะที่สุดแบบฝูงมดดังกล่าวอย่างเป็นรูปธรรม โดยกำหนดให้มีเมืองที่จะต้องเดินทาง 4 เมืองคือ 1, 2, 3 และ 4 โดยที่แต่ละเมืองมีระยะทางดังรูปที่ 5.7



รูปที่ 5.7 ระยะทางระหว่างเมือง

การสร้างเส้นทางเดิน ถ้ากำหนดว่า ในตอนเริ่มต้นค่าฟีโรโมนของทุกเส้นทางเท่ากับ 1 และค่า α และ β เท่ากับ 1 เช่นกันและค่า $\eta = 1/d_{ij}$ สมมุติให้มดตัวที่ 1 เริ่มเดินทางจากเมือง 1 ดังนั้น i=1 และต้องการหาค่า j ว่า e_{ij} หรือเส้นทางที่จะเดินจาก i ไป j ที่ดีที่สุดคือเส้นทางใด เราเริ่มต้นดังนี้

คำนวณค่าความน่าจะเป็นของการเลือกเส้นทางระหว่างเมืองที่ 1 และ 2 หรือ $P(e_{12})$ จะได้

$$P(e_{1,2}) = \frac{\tau_{12}\frac{1}{d_{12}}}{\tau_{12}\frac{1}{d_{12}} + \tau_{13}\frac{1}{d_{13}} + \tau_{14}\frac{1}{d_{14}}} = \frac{\frac{1}{5}}{\frac{1}{5}+\frac{1}{6}+\frac{1}{3}} = 0.285$$

คำนวณค่าความไปได้ของการเลือกเส้นทางระหว่างเมืองที่ 1 และ 3 หรือ $P(e_{13})$ จะได้

$$P(e_{1,3}) = \frac{\tau_{13}\frac{1}{d_{13}}}{\tau_{12}\frac{1}{d_{12}} + \tau_{13}\frac{1}{d_{13}} + \tau_{14}\frac{1}{d_{14}}} = \frac{\frac{1}{6}}{\frac{1}{5}+\frac{1}{6}+\frac{1}{3}} = 0.238$$

คำนวณค่าความไปได้ของการเลือกเส้นทางระหว่างเมืองที่ 1 และ 4 หรือ $P(e_{14})$ จะได้

$$P(e_{1,4}) = \frac{\tau_{14}\frac{1}{d_{14}}}{\tau_{12}\frac{1}{d_{12}} + \tau_{13}\frac{1}{d_{13}} + \tau_{14}\frac{1}{d_{14}}} = \frac{\frac{1}{3}}{\frac{1}{5}+\frac{1}{6}+\frac{1}{3}} = 0.476$$

เนื่องจาก $P(e_{14})$ มีค่าสูงสุด ดังนั้น j = 4 นั่นคือเมืองที่จะเลือกเดินต่อไปคือเมือง 4 ดังนั้นมดตัวแรกจะเริ่มเดินจากเมือง 1 ไป 4 ก่อน และมดจะจำเมืองที่ได้เดินผ่านไปแล้วคือ [1,4] จากเมืองที่ 4 มด

จะเลือกไปยังเมืองถัดไป ในที่นี้ i=4 และเราต้องการหาค่า j และยังมีเมืองที่มดตัวที่ 1 ยังไม่เดินคือเมือง 2 และ 3 การเลือก j = 2 หรือ 3 นั้นหาได้ดังนี้

คำนวณค่าความไปได้ของการเลือกเส้นทางระหว่างเมืองที่ 4 และ 2 หรือ $P(e_{42})$ จะได้

$$P(e_{4,2}) = \frac{\tau_{42}\frac{1}{d_{42}}}{\tau_{42}\frac{1}{d_{42}} + \tau_{43}\frac{1}{d_{43}}} = \frac{\frac{1}{8}}{\frac{1}{8}+\frac{1}{7}} = 0.466$$

คำนวณค่าความไปได้ของการเลือกเส้นทางระหว่างเมืองที่ 4 และ 3 หรือ $P(e_{43})$ จะได้

$$P(e_{4,3}) = \frac{\tau_{43}\frac{1}{d_{43}}}{\tau_{42}\frac{1}{d_{42}} + \tau_{43}\frac{1}{d_{43}}} = \frac{\frac{1}{7}}{\frac{1}{8}+\frac{1}{7}} = 0.533$$

จากเมือง 4 เราเลือกเมือง 3 เนื่องจาก $P(e_{42}) < P(e_{43})$ ดังนั้นมดตัวแรกจะมีเมืองที่เดินผ่านมาแล้ว 3 เมืองดังนี้ [1,4,3] เหลือเมืองที่ยังไม่ได้เดินผ่านอีก 1 เมืองคือ 2 จึงเลือกเดินไปยังเมือง 2 แล้วเดินกลับไปยังเมืองที่เริ่มต้นคือเมือง 1 ดังนั้นเส้นทางเดินทั้งหมดของมดตัวแรกคือ [1,4,3,2,1] มีระยะทางรวมทั้งหมดเท่ากับ $d_{14} + d_{43} + d_{32} + d_{21}$ เท่ากับ 3 + 7 + 4 + 5 = 19

สำหรับมดตัวที่ 2 ให้เริ่มเดินจากเมืองที่ 2 และทำการเดินเช่นเดียวกับมดตัวแรกดังนี้

$$P(e_{21}) = \frac{\tau_{21}\frac{1}{d_{21}}}{\tau_{21}\frac{1}{d_{21}} + \tau_{23}\frac{1}{d_{23}} + \tau_{24}\frac{1}{d_{24}}} = \frac{\frac{1}{5}}{\frac{1}{5}+\frac{1}{4}+\frac{1}{8}} = 0.347$$

$$P(e_{23}) = \frac{\tau_{23}\frac{1}{d_{23}}}{\tau_{21}\frac{1}{d_{21}} + \tau_{23}\frac{1}{d_{23}} + \tau_{24}\frac{1}{d_{24}}} = \frac{\frac{1}{4}}{\frac{1}{5}+\frac{1}{4}+\frac{1}{8}} = 0.434$$

$$P(e_{24}) = \frac{\tau_{24}\frac{1}{d_{24}}}{\tau_{21}\frac{1}{d_{21}} + \tau_{23}\frac{1}{d_{23}} + \tau_{24}\frac{1}{d_{24}}} = \frac{\frac{1}{8}}{\frac{1}{5}+\frac{1}{4}+\frac{1}{8}} = 0.217$$

เนื่องจาก $P(e_{23})$ มีค่าสูงสุด ดังนั้น j = 3 นั่นคือเมืองที่จะเลือกเดินต่อไปคือเมือง 3 เมืองที่ได้เดินผ่านไปแล้วคือ [2,3] จากเมืองที่ 3 มดจะเลือกไปยังเมืองถัดไป คือเมือง 1 หรือ 4 ซึ่งหาได้ดังนี้

$$P(e_{31}) = \frac{\tau_{31}\frac{1}{d_{31}}}{\tau_{31}\frac{1}{d_{31}} + \tau_{34}\frac{1}{d_{34}}} = \frac{\frac{1}{6}}{\frac{1}{6}+\frac{1}{7}} = 0.538$$

$$P(e_{34}) = \frac{\tau_{34}\frac{1}{d_{34}}}{\tau_{31}\frac{1}{d_{31}} + \tau_{34}\frac{1}{d_{34}}} = \frac{\frac{1}{7}}{\frac{1}{6}+\frac{1}{7}} = 0.461$$

จากเมือง 3 เราเลือกเมือง 1 เนื่องจาก $P(e_{31}) > P(e_{34})$ ดังนั้นมดตัวตัวที่สองเดินผ่านมาแล้ว 3 เมืองดังนี้ [2,3,1] เหลือเมืองที่ยังไม่ได้เดินผ่านอีก 1 เมืองคือ 4 จึงเลือกเดินไปยังเมือง 4 แล้วเดินกลับไปยังเมืองที่เริ่มต้นคือเมือง 2 ดังนั้นเส้นทางเดินทั้งหมดของมดตัวที่สองคือ [2,3,1,4,2] มีระยะทางรวมทั้งหมดเท่ากับ $d_{23} + d_{31} + d_{14} + d_{42}$ เท่ากับ 4 + 6 + 3 + 8 = 21

สำหรับมดตัวที่ 3 ให้เริ่มต้นจากเมืองที่ 3 และมดตัวที่ 4 ให้เริ่มจากเมืองที่ 4 แล้วทำการคำนวณเช่นเดียวกับมดตัวที่ 1 และ 2 จะได้เส้นทางออกมาดังนี้

มดตัวที่ 3 เส้นทางเดินคือ [3,2,1,4,3] มีระยะทางรวมเท่ากับ 19
มดตัวที่ 4 เส้นทางเดินคือ [4,1,2,3,4] มีระยะทางรวมเท่ากับ 19

เมื่อได้เส้นทางทั้งหมดแล้ว ก็ถือว่าการสร้างเส้นทางเดินในรอบแรกเสร็จสิ้น เราจะเห็นได้ว่ามีเส้นทางเดินที่มีค่าต่ำสุดอยู่ 3 เส้นทางที่มีผลรวมเท่ากับ 19 คือ

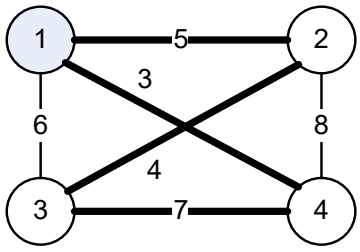
[1,4,3,2,1] มีระยะทางรวมทั้งหมดเท่ากับ $d_{14}+d_{43}+d_{32}+d_{21}$ เท่ากับ 3+7+4+5 = 19

[3,2,1,4,3] มีระยะทางรวมทั้งหมดเท่ากับ $d_{32}+d_{21}+d_{14}+d_{43}$ เท่ากับ 4+5+3+7 = 19

[4,1,2,3,4] มีระยะทางรวมทั้งหมดเท่ากับ $d_{41}+d_{12}+d_{23}+d_{34}$ เท่ากับ 3+5+4+7 = 19

จากเส้นทางเดินทั้งหมด จะเห็นได้ว่าเป็นเส้นทางเดียวกันแต่มีเมืองเริ่มต้นที่ต่างกัน ซึ่งเส้นทางทั้งหมดคือ d_{14} [หรือ d_{41}], d_{43} [หรือ d_{34}], d_{32} [หรือ d_{23}] และ d_{21}[หรือ d_{12}] ตามรูปที่ 5.8 ดังนั้นขั้นตอนต่อไปคือการปรับค่าฟีโรโมนให้กับเส้นทางทั้งหมดที่กล่าวมา

การปรับค่าฟีโรโมน จากกราฟเราจะได้เส้นทางที่เป็นคำตอบที่เป็นไปได้เบื้องต้น 3 เส้นทางดังที่กล่าวมาแล้ว เมื่อนำเส้นทางทั้งหมดมาสร้างเป็นเส้นทึบบนกราฟ จะได้ภาพดังแสดงในรูปที่ 5.8 ซึ่งเป็นการเลือกในรอบแรก ดังนั้นเราจะต้องทำการปรับค่าฟีโรโมนบนเส้นทางที่เป็นเส้นทึบดังนี้



รูปที่ 5.8 เส้นทึบเป็นเส้นทางที่ต้องมีการปรับค่าฟีโรโมน

ในการปรับค่าฟีโรโมนตามสมการ 5.6 ที่มีการกำหนดค่าเริ่มต้นของ τ=1 และค่า ρ = 0.3 ไว้ล่วงหน้า สำหรับการหาค่า $\Delta\tau$ ให้ใช้ตามสมการ 5.7 ดังแสดงไว้ข้างต้น

เมื่อแทนค่า τ_{14} =1 และค่า ρ = 0.3 ในสมการ 5.6 และ 5.7 เพื่อการปรับค่าฟีโรโมนของเส้นทาง d_{14} หลังจากที่มดทุกตัวเดินผ่านไปแล้ว 1 รอบ เราจะได้สมการของการปรับฟีโรโมนดังนี้

$$\square_{14}(\square+1) = (1-0.3) * \square_{14} + (\square)\sum_{\square=1}^{4} \frac{1}{\square^{\square}(\square)} \qquad \text{สมการ 5.8}$$

เนื่องจาก k หมายถึงมดตัวที่ 1, 2, 3, และ 4 ที่เดินบนเส้นทาง d_{14} ซึ่งรวมแล้วมี 4 ตัว และ L^k หมายถึงระยะทางที่มดตัวที่ k เดินผ่าน ซึ่งเท่ากันหมดคือ 1/3 เมื่อแทนค่าลงในสมการ 5.8 จะได้

$$\tau_{14}(t+1) = (1-0.3)*1 + 4*\frac{1}{3} = 2.033$$

การปรับค่าของฟีโรโมนของเส้นทาง d_{43} ของมดทุกตัว จะได้สมการค่าฟีโรโมนใหม่ดังนี้

$$\tau_{43}(t+1) = (1-0.3)*1 + 4*\frac{1}{7} = 1.271$$

การปรับค่าของฟีโรโมนของเส้นทาง d_{32} ของมดทุกตัว จะได้สมการค่าฟีโรโมนใหม่ดังนี้

$$\tau_{32}(t+1) = (1-0.3)*1 + 4*\frac{1}{4} = 1.7$$

การปรับค่าของฟีโรโมนของเส้นทาง d_{21} ของมดทุกตัว จะได้สมการค่าฟีโรโมนใหม่ดังนี้

$$\tau_{21}(t+1) = (1-0.3)*1 + 4*\frac{1}{5} = 1.5$$

สำหรับเส้นทางที่เหลือคือ d_{13} และ d_{24} ค่าของ $\Delta\tau$ จะเป็น 0 เนื่องจากเป็นเส้นทางที่ไม่ได้รับการเลือกให้อยู่บนเส้นทางที่ดีที่มดเดินผ่าน ตามสมการ 5.7 ดังนั้นค่าฟีโรโมนใหม่บนเส้นทางเหล่านี้จะเท่ากับ

$\tau_{13}(t+1) = \tau_{31}(t+1) = (1-0.3) = 0.7$ และ

$\tau_{24}(t+1) = \tau_{42}(t+1) = (1-0.3) = 0.7$

การสร้างเส้นทางเดิน (รอบ 2) ค่าฟีโรโมนใหม่ของเส้นทางบนกราฟตามรูปที่ 5.8 จะเป็นดังนี้ τ_{14}[หรือτ_{14}] = 2.033, τ_{43} [หรือ τ_{34}] = 1.271, τ_{32}[หรือ τ_{23}] = 1.7 และ τ_{21} [หรือ τ_{12}]= 1.5 สำหรับเส้นทางที่เหลือคือ τ_{13}[หรือ τ_{31}] และ τ_{24} [หรือ τ_{42}] ค่าฟีโรโมนจะเท่ากับ 0.7 มดตัวที่ 1 เริ่มเดินทางจากเมือง 1 โดยที่ i=1 และต้องการหาค่า j

การหาเส้นทางของมดตัวแรก เมื่อเลือกเมืองที่ 1 เป็นเมืองเริ่มต้น โดยใช้การคำนวณจาก P(e) ที่มีสามเส้นทางคือ e_{12}, e_{13} และ e_{14} จะเป็นดังนี้

$$P(e_{1,2}) = \frac{\tau_{12}\frac{1}{d_{12}}}{\tau_{12}\frac{1}{d_{12}} + \tau_{13}\frac{1}{d_{13}} + \tau_{14}\frac{1}{d_{14}}} = \frac{(1.5)\frac{1}{5}}{(1.5)\frac{1}{5} + (0.7)\frac{1}{6} + (2.033)\frac{1}{3}} = 0.274$$

$$P(e_{1,3}) = \frac{\tau_{13}\frac{1}{d_{13}}}{\tau_{12}\frac{1}{d_{12}} + \tau_{13}\frac{1}{d_{13}} + \tau_{14}\frac{1}{d_{14}}} = \frac{(0.7)\frac{1}{6}}{(1.5)\frac{1}{5} + (0.7)\frac{1}{6} + (2.033)\frac{1}{3}} = 0.1066$$

$$P(e_{1,4}) = \frac{\tau_{14}\frac{1}{d_{14}}}{\tau_{12}\frac{1}{d_{12}} + \tau_{13}\frac{1}{d_{13}} + \tau_{14}\frac{1}{d_{14}}} = \frac{(2.033)\frac{1}{3}}{(1.5)\frac{1}{5} + (0.7)\frac{1}{6} + (2.033)\frac{1}{3}} = 0.619$$

เนื่องจาก $P(e_{14})$ มีค่าสูงสุด ดังนั้น j = 4 นั่นคือเมืองที่จะเลือกเดินต่อไปคือเมือง 4 ดังนั้นมดตัวแรกจะเริ่มเดินจากเมือง 1 ไป 4 ซึ่งมดจะจำไว้ว่าเป็นเมืองที่ได้เดินผ่านไปแล้วคือ [1,4] จากเมืองที่ 4 มดจะเลือกไปยังเมืองถัดไป ในที่นี้ i=4 และเราต้องการหาค่า j และยังมีเมืองที่มดตัวที่ 1 ยังไม่เดินคือเมือง 2 และ 3 การเลือก j = 2 หรือ 3 นั้นหาได้ดังนี้

$$P(e_{4,2}) = \frac{\tau_{42}\frac{1}{d_{42}}}{\tau_{42}\frac{1}{d_{42}} + \tau_{43}\frac{1}{d_{43}}} = \frac{(0.7)\frac{1}{8}}{(0.7)\frac{1}{8} + (1.271)\frac{1}{7}} = 0.325$$

$$P(e_{4,3}) = \frac{\tau_{43}\frac{1}{d_{43}}}{\tau_{42}\frac{1}{d_{42}} + \tau_{43}\frac{1}{d_{43}}} = \frac{(1.271)\frac{1}{7}}{(0.7)\frac{1}{8} + (1.271)\frac{1}{7}} = 0.674$$

จากเมือง 4 เราเลือกเมือง 3 เนื่องจาก $P(e_{42}) < P(e_{43})$ ดังนั้นมดตัวแรกจะมีเมืองที่เดินผ่านมาแล้ว 3 เมืองดังนี้ [1,4,3] เหลือเมืองที่ยังไม่ได้เดินผ่านอีกหนึ่งเมืองคือ 2 จึงเลือกเดินไปยังเมือง 2 แล้วเดินกลับไปยังเมืองที่เริ่มต้นคือเมือง 1 ดังนั้นเส้นทางเดินทั้งหมดของมดตัวแรกคือ [1,4,3,2,1] มีระยะทางรวมทั้งหมดเท่ากับ $d_{14} + d_{43} + d_{32} + d_{21}$ เท่ากับ 3 + 7 + 4 + 5 = 19

สำหรับมดตัวที่สองเริ่มต้นการเดินทางจากเมืองที่สองและทำการหาเส้นทางเดินเช่นเดียวกับมดตัวแรก จนกระทั่งมดทุกตัวไปเริ่มต้นที่เมืองทุกเมือง (ในที่นี้คือ 4 เมือง) แล้วเลือกเส้นทางที่ดีที่สุด จากนั้นทำการปรับค่าฟีโรโมนใหม่ ทำเวียนกันเช่นนี้เรื่อยๆ จนกระทั่งได้เส้นทางที่ดีที่สุดที่ไม่มีการเปลี่ยนแปลง ก็ให้เลิกทำงาน แล้วเลือกเส้นทางที่ดีที่สุดเป็นคำตอบ

5.3.2 การหาเส้นทางเดินรถ

นิยาม การหาเส้นทางเดินรถ เป็นการหาเส้นทางที่มีระยะทางที่สั้นที่สุดหรือเส้นทางที่ทำให้เกิดค่าใช้จ่ายน้อยที่สุดของการรวมระยะทางของจำนวนรถ m คัน ที่ให้บริการกับสถานที่ของลูกค้าจำนวน n แห่ง

ในการอธิบายอย่างเป็นทางการในรูปของกราฟ G(V, A, d) เราจะนิยามปัญหาโดยให้ V คือโหนดต่างๆ ในกราฟ เมื่อเซทของ $V = \{v_0, v_1, v_2, \ldots v_n\}$ เมื่อ $v_0, v_1, v_2, \ldots v_n$ เป็นโหนดต่างๆ ในกราฟที่ใช้แทนสถานที่ต่างๆ ของลูกค้าที่รถจะวิ่งไป และ A เป็นเส้นเชื่อม (link) ระหว่างโหนดต่างๆ ในกราฟ ซึ่ง $A = \{v_i, v_j\}$ i และ j = 0... n และ $i \neq j$ หมายถึงระยะทางระหว่างสถานที่ของลูกค้า ดังนั้น v_i และ v_j จะเท่ากับ d_{ij} เมื่อ d คือระยะทางระหว่างสถานที่ต่างๆ สำหรับการเดินทางของรถทุกคันจะเริ่มต้นจากคลังสินค้า v_0 ในการขนส่งสินค้าไปยังลูกค้า ลูกค้าแต่ละคนมีความต้องการสินค้าเท่ากับ q_i (ซึ่งเป็นจำนวนเต็มบวก) และรถแต่ละคันสามารถบรรจุสินค้าได้เท่ากับ Q ในการแก้ปัญหาของการหาเส้นทางเดินรถนั้นมีเงื่อนไขดังนี้

- สถานที่ของลูกค้าคนหนึ่งๆ จะมีรถเพียงคันเดียวไปส่งของ และส่งเพียงครั้งเดียวเท่านั้น
- รถแต่ละคันจะต้องเริ่มต้นที่คลังสินค้า v_0 และเมื่อส่งของหมดแล้วต้องกลับมาที่คลังสินค้า v_0 เช่นเดิม

□ ปริมาณของสินค้าทั้งหมดที่รถแต่ละคันส่งได้จะต้องไม่เกิน Q

นอกจากนี้แล้ว ปัญหายังประกอบด้วยข้อกำหนดของระยะทางในการให้บริการของรถแต่ละคันด้วย ซึ่งกำหนดให้มีค่าสูงสุดเท่ากับ L_m และมีระยะทางของการให้บริการเท่ากับ δ (เทียบมาจากระยะเวลาของการให้บริการ) สำหรับลูกค้าแต่ละรายบนเส้นทางเดินรถ

ในการแก้ปัญหา เบลล์และแมกมูลเลน (Bell, 2004) ได้นำเสนอวิธีการไว้ 2 ขั้นตอนคือ การสร้างคำตอบที่เป็นไปได้ และการปรับค่าฟีโรโมนไว้ดังนี้

การสร้างคำตอบที่เป็นไปได้ เป็นการหาการสร้างเส้นทางเดินรถ (Route construction) ซึ่งจะเริ่มด้วยการที่มดแต่ละตัวจะแทนรถขนส่งแต่ละคัน และเส้นทางการเดินรถของรถแต่ละคันจะถูกสร้างขึ้น ด้วยการเลือกสถานที่ที่จะส่งของทีละแห่งจนกระทั่งสถานที่ต่างๆ ของลูกค้าทุกแห่งถูกส่งจนครบ ในตอนเริ่มต้น มดแต่ละตัวจะเริ่มที่คลังสินค้า และจำนวนสถานที่ที่จะไปส่งสินค้า มดจะเลือกสถานที่ที่จะไปส่งของแห่งถัดไปจากสถานที่ต่างๆ ที่อยู่ในรายการ เมื่อไปถึงสถานที่นั้น จำนวนสินค้าบนรถก็จะมีการปรับจำนวนตามสินค้าที่ส่งไปในสถานที่ที่ไปถึง แล้วก็จะไปเลือกสถานที่ที่จะส่งแห่งใหม่ รถ (หรือมด)จะกลับมายังคลังสินค้าทันทีที่สินค้าหมด หรือได้ไปส่งสินค้าจนครบทุกแห่งแล้ว ระยะทางรวม L จะถูกใช้เป็นค่าอ็อปเจ็คทีฟฟังก์ชันสำหรับการเดินทางของรถสิ้นสุด การสร้างเส้นทางเดินรถจะเริ่มจากมดตัวแรกจนกระทั่งเสร็จกระบวนการ แล้วมดตัวที่สองจึงจะเริ่มการเดินทาง เมื่อตัวที่สองเสร็จก็จะเป็นหน้าที่ของตัวที่สาม จนกระทั่งครบมดตัวที่ m หรือครบรถทุกคัน

สำหรับการเลือกสถานที่ที่จะส่งของถัดไป j มดจะให้วิธีการพิจารณาดังนี้

$$\square = \begin{cases} \arg\ \max\{(\square_{\square\square}(\square_{\square\square})^{\square}\} & \square\square\square\ \square \notin \square_{\square}, \square\square\ \square \le \square_0 \\ \square, & \square\square\square\square\square\square\square\square\square\square \end{cases}$$ สมการ 5.9

เมื่อค่า τ_{iu} คือจำนวนฟีโรโมนบนเส้นทางระหว่างเมืองปัจจุบัน i และเมืองที่เป็นไปได้ทั้งหมด u สำหรับ η_{iu} มีค่าเท่ากับ $1/d_{iu}$ และค่าพารามิเตอร์ β เป็นค่าที่ใช้ปรับน้ำหนักของระยะทางกับฟีโรโมนในอัลกอริธึม โดยให้ $\beta > 0$ สำหรับค่า q เป็นตัวแปรสุ่มแบบเอกรูป (Uniform random variable) ที่มีค่า

ระหว่าง [0,1] สำหรับ q_0 เป็นค่าพารามิเตอร์ที่กำหนดขึ้นจากการทดลอง จากการทดลองของ เบลล์และแมกมูลเลน ได้กำหนดค่าของ q_0 ไว้เท่ากับ 0.9

สถานที่ต่างๆ ที่มด (หรือรถส่งของ) ได้ไป (ส่งของ) มาแล้วจะถูกนำไปเก็บไว้ในหน่วยความจำ M_k และจะไม่ถูกเลือกในการเดินทางครั้งต่อไป ดังนั้น ในกรณีที่เมืองต่อไปที่จะเลือก เป็นเมืองที่ยังไม่ได้ไปมาแล้ว หรือ $u \notin M_k$ และค่าของ $q \le q_0$ แล้ว เมืองที่จะเลือกถัดไป หรือ j เป็นไปตามสมการ 5.9

ในกรณีที่ค่า $q > q_0$ มดจะเลือกเส้นทางเดินไปยังสถานที่ใหม่ด้วยตัวแปรสุ่ม S ซึ่งตัวแปรสุ่ม S นี้มีค่าเท่ากับ p_{ij} ซึ่งหาได้จากสูตรดังสมการ 5.10 ดังนี้

$$\square_{\square\square} = \frac{\square_{\square\square}\square_{\square\square}^{\square}}{\sum_{\square\notin\square_{\square}}(\square_{\square\square}\square_{\square\square}^{\square})} \quad \square\square\ \square \notin \square_{\square},\ \square\square\square\square\square\square\square\square\square = 0 \qquad \text{สมการ 5.10}$$

การปรับค่าฟีโรโมน การปรับค่าฟีโรโมนของเส้นทางเดินรถจะประกอบด้วย 2 ขั้นตอนใหญ่คือ การปรับค่าท้องถิ่น (Local Updating) ด้วยการลดค่าฟีโรโมนของเส้นทางเดินทุกเส้นทาง ที่เกิดจากการระเหยไปตามเวลา ด้วยสมการดังต่อไปนี้

$$\tau_{i,j} = (1-\alpha)\tau_{i,j} + (\alpha)\tau_0$$

เมื่อค่า α เป็นพารามิเตอร์ที่ควบคุมความเร็วของการระเหยของฟีโรโมน ค่า τ_0 เป็นค่าเริ่มต้นของฟีโรโมนที่กำหนดให้กับทุกเส้นทางของกราฟ G หลังจากที่มดทั้ง m ตัว สร้างเส้นทางที่เป็นไปได้ดังที่ได้กล่าวมาข้างต้น ขั้นตอนต่อไปคือการเพิ่มค่าฟีโรโมนให้กับเส้นทางที่ดีที่สุดจากเส้นทางที่มด m ตัวเลือกเอาไว้ด้วยสมการ 5.11 ดังนี้

$$\square_{\square\square} = (1-\square)\square_{\square\square} + \square(\square)^{-1} \qquad \text{สมการ 5.11}$$

ซึ่ง L คือระยะทางรวมของเส้นทางที่ดีที่สุด

5.3.3 การวางผังโรงงาน

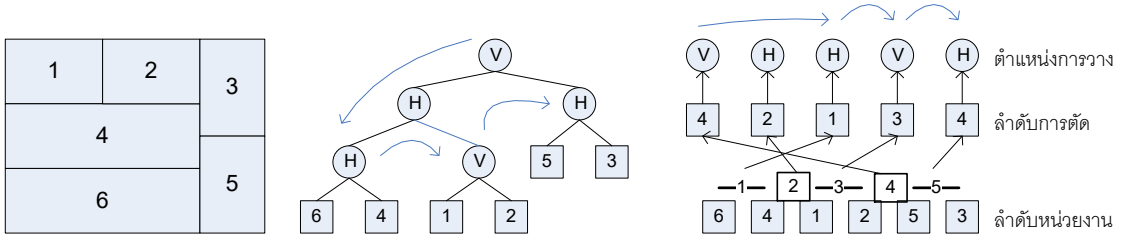
นิยาม เป็นการจัดวางหน่วยงานทั้งหลายให้อยู่ในบริเวณที่กำหนด ภายใต้เงื่อนไขของการใช้งานของหน่วยงาน และความสัมพันธ์ระหว่างหน่วยงานที่อยู่ในพื้นที่นั้น จุดประสงค์ของการจัดวางพื้นที่ก็เพื่อให้ได้ผังที่มีหน่วยงานที่เราต้องการครบถ้วน โดยที่ผังนั้นจะสามารถอำนวยความสะดวกในการทำงานได้มากที่สุดบนพื้นที่ที่น้อยที่สุด ซึ่งมีอ็อปเจ็คทีฟฟังก์ชันดังนี้

$$\min \square(\square) = \sum_{\square=1}^{\square} \sum_{\square=1}^{\square} \square_{\square\square} \square_{\square\square} \qquad \text{สมการ 5.12}$$

เมื่อ F(s) คืออ็อปเจ็คทีฟฟังก์ชันที่แสดงถึงค่าใช้จ่ายรวมที่เกิดขึ้นจากการวางผัง โดยที่คำนวณได้จากผลคูณของ f_{ij} คือค่าใช้จ่ายในการขนย้ายวัสดุจาก i ไป j และ d_{ij} เป็นระยะทางระหว่าง i และ j

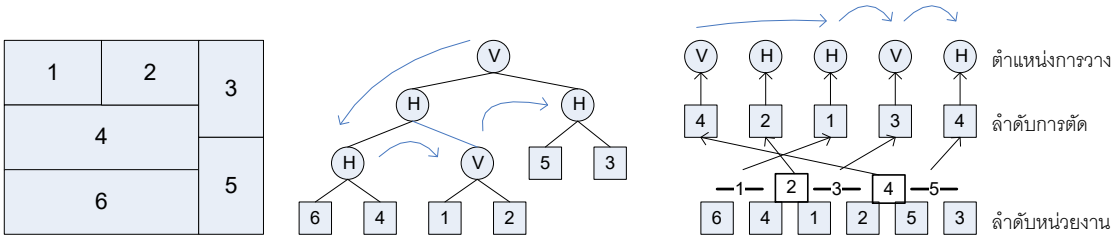
โคมารูดิน และ วัง (K.Y. Komarudin and Wong, 2010) ได้นำเสนอวิธีการปัญหานี้โดยการหาวิธีการสร้างคำตอบที่เป็นไปได้ และขบวนการปรับค่าฟีโรโมนตามขั้นตอนดังต่อไปนี้

การสร้างคำตอบที่เป็นไปได้ การสร้างคำตอบที่เป็นไปได้จะอาศัยรูปแบบของไบนารีกราฟ (Binary Graph) ในการแทนค่าผังโรงงาน สำหรับวิธีการแทนค่าเป็นไปตาม



(ก) ผังงาน (ข) การแสดงผังงานด้วยต้นไม้ (ค) การจัดลำดับของผังงานจากโครงสร้างต้นไม้

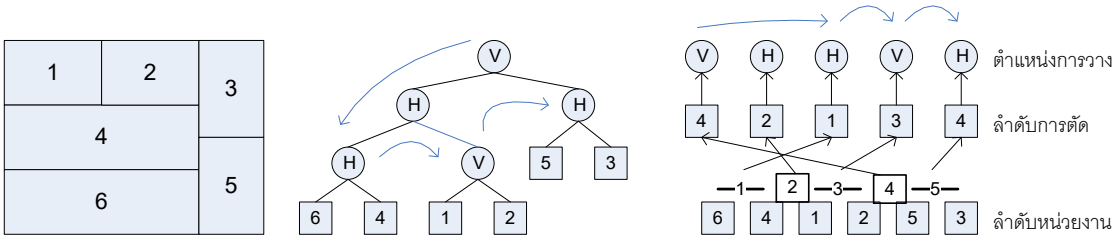
รูปที่ 5.9



(ก) ผังงาน (ข) การแสดงผังงานด้วยต้นไม้ (ค) การจัดลำดับของผังงานจากโครงสร้างต้นไม้

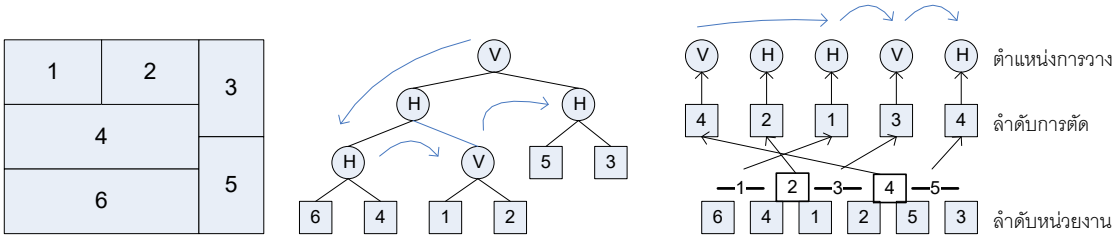
รูปที่ 5.9 แสดงการแทนค่า และการแสดงคำตอบ

จาก



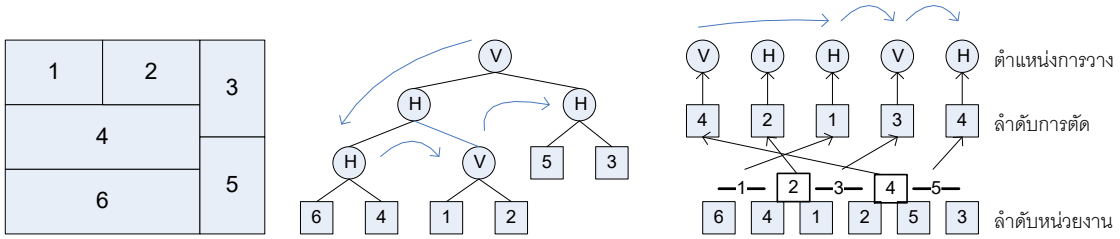
(ก) ผังงาน (ข) การแสดงผังงานด้วยต้นไม้ (ค) การจัดลำดับของผังงานจากโครงสร้างต้นไม้

รูปที่ 5.9(ก) แสดงผังงานของพื้นที่ที่เราจะทำการจัดวาง ตามรูป มีพื้นที่อยู่ 6 พื้นที่



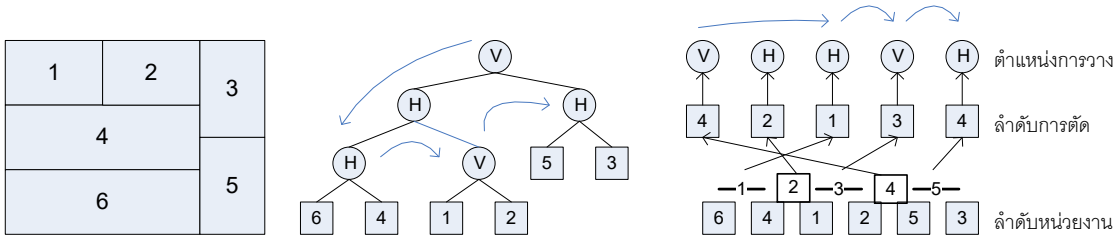
(ก) ผังงาน (ข) การแสดงผังงานด้วยต้นไม้ (ค) การจัดลำดับของผังงานจากโครงสร้างต้นไม้

รูปที่ 5.9(ข) เป็นโครงสร้างต้นไม้ที่ใช้แสดงผังงานตาม



(ก) ผังงาน (ข) การแสดงผังงานด้วยต้นไม้ (ค) การจัดลำดับของผังงานจากโครงสร้างต้นไม้

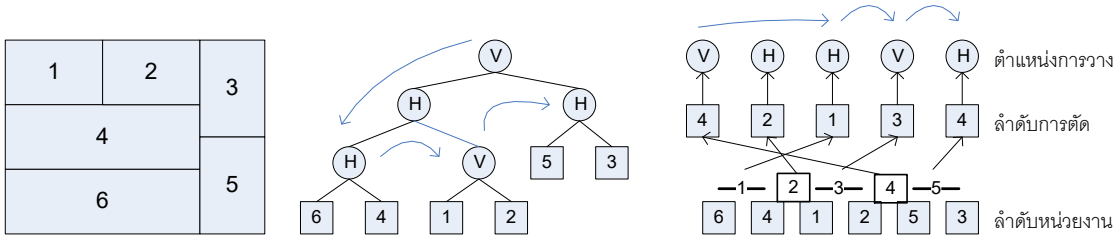
รูปที่ 5.9(ก) วิธีการแสดงคือการแทนพื้นที่ทั้งหมดด้วยโหนดปลายทาง (Terminal Node) จากนั้นหาความสัมพันธ์ของพื้นที่ (โหนดปลายทาง) เป็นคู่ๆ ตาม



(ก) ผังงาน (ข) การแสดงผังงานด้วยต้นไม้ (ค) การจัดลำดับของผังงานจากโครงสร้างต้นไม้

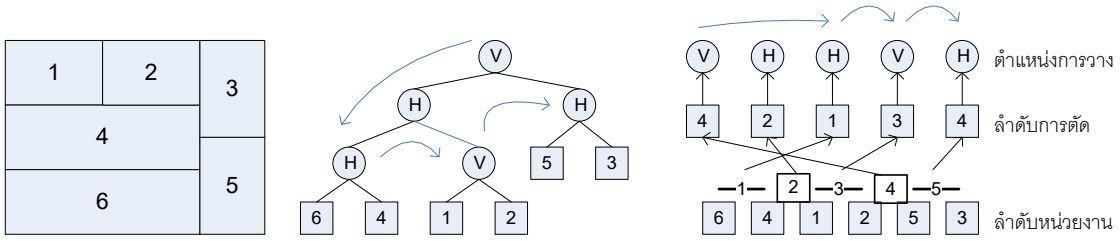
รูปที่ 5.9(ข) พื้นที่ (6) และ (4) มีความสัมพันธ์แบบ Horizontal (H) พื้นที่ (1) และ (2) มีความสัมพันธ์แบบ Vertical (V) และพื้นที่ (5) และ (3) มีความสัมพันธ์แบบ Horizontal (H) ในระดับที่สูงขึ้น พื้นที่คู่ (6 4) มีความสัมพันธ์กับคู่ (1 2) แบบ Horizontal (H) และสุดท้าย คู่ (6 4) (1 2) สัมพันธ์กับ (5 3) แบบ Vertical (V) รูปที่ 5.9(ค) เป็นการแสดงลำดับของผังงานในโครงสร้างต้นไม้ แถวล่างสุดเป็นพื้นที่ มีการจัดเรียงตามลำดับที่แสดงไว้ในโหนดปลายทางของโครงสร้างต้นไม้ เลข -1-, -3- และ -5- เป็นการจับคู่ของพื้นที่ 2 พื้นที่ เลข 2 และ 4 เป็นการจัดคู่ของพื้นที่ที่มีการจับคู่มาแล้ว ภาพเหนือขึ้นไป แสดงลำดับการตัด ที่แสดงว่าในการแบ่งพื้นที่ที่แสดงความสัมพันธ์นี้ มีวิธีการเรียงลำดับของ

ความสัมพันธ์ก่อนหลังอย่างไร และสุดท้าย โหนดที่อยู่บนสุด แสดงลักษณะของความสัมพันธ์จาก



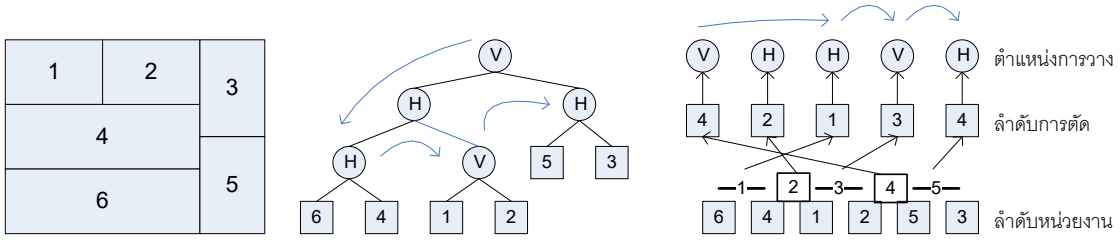
(ก) ผังงาน (ข) การแสดงผังงานด้วยต้นไม้ (ค) การจัดลำดับของผังงานจากโครงสร้างต้นไม้

รูปที่ 5.9(ค) ถ้าหากว่าเราเปลี่ยนตำแหน่งการวางตัวแรกที่เป็น V[4] ให้เป็น H ต้นไม้ของ



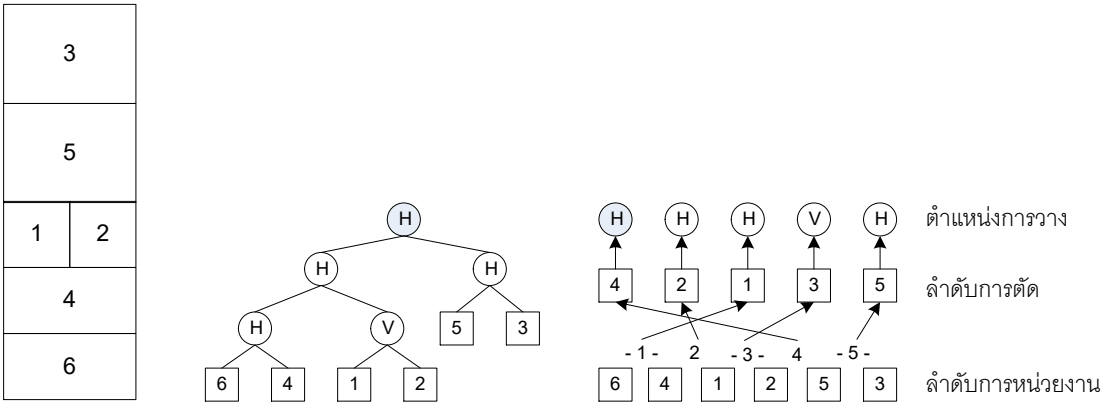
(ก) ผังงาน (ข) การแสดงผังงานด้วยต้นไม้ (ค) การจัดลำดับของผังงานจากโครงสร้างต้นไม้

รูปที่ 5.9(ข) ที่โหนดรากจะเปลี่ยนจาก V เป็น H และจะทำให้ผังงานใน



(ก) ผังงาน (ข) การแสดงผังงานด้วยต้นไม้ (ค) การจัดลำดับของผังงานจากโครงสร้างต้นไม้

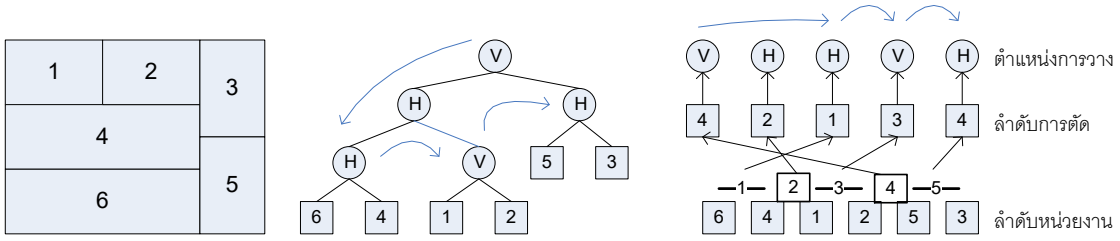
รูปที่ 5.9(ก) เปลี่ยนไปด้วย ดังรูปที่ 5.10 (ก)



(ก) ผังงาน (ข) การแทนผังงานด้วยต้นไม้ (ค) การแทนผังงานด้วยการจัดลำดับจากต้นไม้

รูปที่ 5.10 ผลของการวางผังที่เปลี่ยนไปเนื่องจากการเปลี่ยนค่าตำแหน่งการวางจาก V เป็น H

เช่นกันเมื่อมีการเปลี่ยนลำดับการตัด (Slicing sequence) ก็จะทำให้รูปร่างของการวางผังโรงงานเปลี่ยนไปด้วย และเมื่อคำนวณค่า F(s) ก็จะเปลี่ยนไป ในการวางผังนี้ เราต้องการได้ค่า F(s) ต่ำสุด ดังนั้นในการสร้างผังงานที่เป็นคำตอบที่เป็นไปได้ก็คือการเปลี่ยนตำแหน่งการวาง และลำดับการตัด ซึ่งการเปลี่ยนนี้ก็คือขบวนการค้นหาท้องถิ่น (Local search procedure) ในการสร้างผังงานทางเลือก ตัวอย่างของขบวนการนี้เช่น การสลับกิ่งของการแทนผังงานด้วยต้นไม้ตาม



(ก) ผังงาน (ข) การแสดงผังงานด้วยต้นไม้ (ค) การจัดลำดับของผังงานจากโครงสร้างต้นไม้

รูปที่ 5.9(ข) การเปลี่ยนชนิดของตำแหน่งการวาง (เช่นจาก V เป็น H หรือ H เป็น V) เป็นต้น สำหรับการตำแหน่งการวาง และลำดับการตัดใดจะได้รับเลือกให้พิจารณาจากการคำนวณตามสมการ 5.13 ดังนี้

$$\square(\square_{\square\square}|\square) = \frac{(\square_{\square\square})^{\square}(\square_{\square}\square_{\square})^{\square}}{\sum_{\square_{\square\square}\in\square(\square)}(\square_{\square\square})^{\square}(\square_{\square}\square_{\square})^{\square}} \quad \forall\square_{\square\square} \in \square(\square)$$ สมการ 5.13

เมื่อค่า $p(C_{ik}|S)$ เป็นความน่าจะเป็นที่จะวางหน่วยงาน i ไว้ที่ตำแหน่ง k ค่า C_{ik} แสดงการวางหน่วยงาน i ไว้ที่ตำแหน่ง k ค่า τ_{ik} เป็นค่าฟีโรโมนที่แสดงถึงการวางหน่วยงาน i ไว้ที่ตำแหน่ง k ค่า m_i เป็นผลรวมของการขนย้ายวัสดุ (Material flow) ที่เข้าและออกจากหน่ายงาน k ค่า α เป็นค่าที่แสดงความสัมพันธ์ของข้อมูลข่าวสารฟีโรโมน (Pheromone Information) ค่า β เป็นค่าที่แสดงความสัมพันธ์ของข้อมูลข่าวสารฮิวริสติก (Heuristic information) หรืออีกนัยหนึ่งค่า α และ β เป็นค่าที่แสดงน้ำหนักให้กับค่าของ τ และค่าของ $m\lambda$ และค่า N(s) เป็นเซทของหน่วยงาน และตำแหน่งการวางซึ่งยังไม่ได้มีการจัดวางมาก่อน สำหรับค่า λ เป็นพารามิเตอร์ของการวางหน่วยงานที่ตำแหน่ง k และมีการคำนวณจากสมการ 5.14 ดังนี้

$$\square_{\square} = \frac{\square}{2} - \left|\square_{\square} - \frac{\square}{2}\right| + \frac{\square}{2} - \left|\square_{\square} - \frac{\square}{2}\right| \qquad \text{สมการ 5.14}$$

เมื่อ L คือความยาวของหน่วยงาน ค่า W คือความกว้าง สำหรับค่า x_k และ y_k เป็นตำแหน่งของค่าบนแกน x และ y จุดมุ่งหมายของ λ ก็เพื่อทำให้การวางหน่วยงานที่มีการขนย้ายวัสดุระหว่างกันมากมาอยู่ตรงกลาง

การปรับค่าฟีโรโมน ในการปรับค่าฟีโรโมนของการแก้ปัญหานี้จะอาศัยสมการ 5.15 ดังนี้

$$\square_{\square\square} = (1 - \square)\square_{\square\square} + \square \sum_{\{\square \in \square_{\square\square\square} | \square_{\square\square} \in \square\}} \square_{\square} \frac{1}{\square(\square)} \qquad \text{สมการ 5.15}$$

เมื่อ ค่า ρ เป็นอัตราการระเหยของฟีโรโมนที่มีค่าอยู่ระหว่าง [0,1] ค่า w_s เป็นค่าคงที่ที่บอกน้ำหนักของสมการในการหาคำตอบ s ค่า F(s) เป็นอ็อปเจ็คทีฟฟังก์ชันของผังงานที่ได้

5.4 ความหลากหลายของวิธีการหาค่าเหมาะที่สุดแบบฝูงมด

วิธีการหาค่าเหมาะที่สุดแบบฝูงมดมีการดัดแปลงจากวิธีการดังกล่าวข้างต้นอีกหลายวิธีดังนี้

5.4.1 ระบบมด

วิธีตามที่อธิบายดังกล่าวข้างต้นเป็นวิธีการที่มีการใช้มากที่สุดเรียกว่า ระบบมด (Ant System) หรือ AS (M., Maniezzo, V., and Colorni, A. Dorigo, 1996) ซึ่งมีองค์ประกอบที่สำคัญ 2 อย่าง คือ การปรับปรุงฟีโรโมน และการเลือกเส้นทางระหว่างเมือง i และ j ของมดตัวที่ k

ลำดับแรกการปรับปรุงฟีโรโมนจะมีสมการ 5.16 ดังนี้

□□□(□ + 1) = (1 − □) ∗ □□□(□) + ∑□□=1 Δ□□□□(□) สมการ 5.16

เมื่อ $\tau_{i,j}$ เป็นฟีโรโมน ค่า 0<ρ≤1 เป็นอัตราการระเหยของฟีโรโมน และค่า $\Delta\tau^{k}_{ij}(t)$ เป็นจำนวนฟีโรโมนที่เพิ่มขึ้นเนื่องจากมดตัวที่ k ที่นำไปทิ้งไว้บนเส้นทางที่มันเดินผ่าน ซึ่งมันถูกกำหนดไว้ตามสมการ 5.17 ดังนี้

$$\Delta\square^{\square}_{\square\square}(\square) = \begin{cases} \frac{\square}{\square^{\square}}(\square) & \square\square(\square,\square) \in \square\square\square\square\ \square\square\ \square\square\square\ \square^{\square\square} \\ 0 & \square\square\square\square\square\square\square\square\square\square \end{cases}$$ สมการ 5.17

ค่า L^k เป็นระยะทางที่มดตัวที่ k เดินผ่านมาทั้งหมด

การเลือกเส้นทางระหว่างเมือง i และ j ของมดตัวที่ k จะหาได้จากสมการดังนี้

$$\square(\square_{\square\square}) = \frac{(\square_{\square\square}(\square))^{\propto}\square^{\square}_{\square\square}}{\sum_{\square\in\square^{\square}_{\square}}(\square_{\square\square}(\square))^{\propto}\square^{\square}_{\square\square}} \quad \square\square\ \square\square\square^{\square}_{\square},\ \square\square\square\square\square\square\square\square\square = 0$$ สมการ 5.18

โดยที่ค่า $\eta_{ij}=1/d_{ij}$ สำหรับค่า α และ β เป็นพารามิเตอร์ที่กำหนดความสัมพันธ์ของค่าฟีโรโมน (τ) และค่าข้อมูลข่าวสารฮิวริสติก (η) และ N_i^k ก็คือเมืองทั้งหมด

5.4.2 ระบบมดอีลีท

ระบบมดอีลีท (Elitist Ant System) ถูกนำเสนอครั้งแรกโดยโดริโกและคณะ (Dorigo, 1996) ความแตกต่างระหว่างระบบมดอีลีทและระบบมด อยู่ที่การปรับค่าฟีโรโมน ในระบบมดอีลีทจะใช้ค่าที่ดีที่สุดของการค้นหาที่ผ่านมา ซึ่งเป็นค่าที่ดีที่สุดแบบโกลบอลเพิ่มเข้ามาในสมการ ค่านี้ใส่เข้ามาเพื่อเพิ่มน้ำหนักให้กับเส้นทางที่เดินผ่าน ดังแสดงในสมการต่อไปนี้

$$\tau_{i,j}(t+1) = (1-\rho)*\tau_{i,j}(t) + \sum_{k=1}^{m}\Delta\tau_{ij}{}^{k}(t) + e\Delta\tau_{ij}^{bs} : i,j = 1,2,...n$$ สมการ 5.19

จากสมการข้างต้น ตัวแปร $e\Delta\tau_{ij}^{bs}$ เป็นค่าของฟีโรโมนที่วางโดยมดอีลีทบนเส้นทางที่เป็นคำตอบที่ดีตั้งแต่ตอนเริ่มต้นของการทำงาน ในขณะที่พารามิเตอร์ e เป็นน้ำหนักของเส้นทาง สำหรับค่าของ $\Delta\tau_{ij}^{bs}$ จะเป็นดังนี้

$$\Delta\tau^{bs}{}_{i,j}(t) = \begin{cases} Q/L^{bs} & if\ (i,j) \in T^{bs} \\ 0 & otherwise \end{cases} \qquad \text{สมการ 5.20}$$

จากสมการข้างต้น พารามิเตอร์ T^{bs} เป็นเส้นทางเดินที่ดีที่สุดเท่าที่หามา พารามิเตอร์ L^{bs} เป็นความยาวของเส้นทางที่ดีที่สุด ในการทำเช่นนี้ จะทำให้เอ็กซ์พลอเรชันของเมตาฮิวริสติกนี้จะมุ่งทิศของการค้นหาไปที่คำตอบที่ดีที่สุดเท่าที่หามา (Best-so-far Solution)

5.4.3 ระบบมดแบบการจัดอันดับ

ระบบมดแบบการจัดอันดับ (Rank Based Ant System: ASrank) ถูกนำเสนอครั้งแรกโดยบูลน์ไฮเมอร์และคณะ (Bullnheimer, 1997) เป็นระบบที่พัฒนาต่อเนื่องจากระบบมดอีลีท

แนวคิดคือ หลังจากมดทั้ง m ตัวได้สร้างเส้นทางเดินขึ้นมาแล้ว มดทั้งหมดจะจัดเรียงลำดับตามความยาวของระยะทางที่มดแต่ละตัวพบมา (L1 □ L2 □ ... □ Lm) จากนั้นมดที่ดีจำนวน **μ** ตัวแรกจะถูกเลือก ให้เพิ่มน้ำหนักให้กับฟีโรโมนบนเส้นทางที่พวกมันเดิน นอกจากนั้น มดที่ดีที่สุดจำนวน **ω** ตัวเท่านั้นจะถูกพิจารณา ซึ่งจะทำให้สามารถหลีกเลี่ยงอันตรายที่เกิดจากการเน้นค่าฟีโรโมนบนบางเส้นทางมากเกินไป จนทำให้เกิดโลคอลมินิมั่ม

ถ้าค่า **σ** เป็นน้ำหนักที่เพิ่มให้กับฟีโรโมนสำหรับเส้นทางที่ดีที่สุดเท่าที่หามาได้ ค่าของมันจะต้องไม่น้อยกว่าค่าที่กำหนดน้ำหนักอื่นๆ และค่าที่น้อยที่สุดจะเป็น 1 สำหรับมดลำดับที่ **μ** จะมีน้ำหนักเท่ากับ (**σ** - **μ**) และกำหนดให้ **ω** = **σ** - 1 ซึ่งหมายความว่าจำนวนมดที่พิจารณาจะมากกว่ามดอีลีท 1 ตัว จากการรวมกันของวิธีการปรับค่าฟีโรโมนแบบอีลีทและแบบการจัดอันดับ จะได้สมการออกมาดังนี้

$$\tau_{i,j}(t+1) = (1-\rho) * \tau_{i,j}(t) + \sum_{\mu=1}^{\sigma-1} \Delta\tau_{ij}{}^{bs}(t) + \sigma\Delta\tau_{ij}^{bs} : i,j = 1,2,...n \qquad \text{สมการ 5.21}$$

$$\Delta\square^{\square}_{\square\square}(\square) = \begin{cases} (\square - \square)\frac{\square}{\square_{\square}} & \square\square(\square,\square) \in \square - \square\square\square\ \square\square\square\square\ \square\square\square\ \square\square\square\square\square\square\ \square\square\ \square\square\square\square(\square,\square) \\ 0 & \square\square\square\square\square\square\square\square\square\square \end{cases}$$

สมการ 5.22

$$\Delta\square^{\square\square}_{\square\square}(\square) = \begin{cases} \frac{\square}{\square^{\square\square}} & \square\square(\square,\square) \in \square^{\square\square} \\ 0 & \square\square\square\square\square\square\square\square\square\square \end{cases}$$

สมการ 5.23

โดยที่

μ เป็นดัชนีของการจัดอันดับ (Ranking index)

$\Delta\tau_{ij}^{\mu}(t)$ การเพิ่มค่าของฟีโรโมนบนเส้นทาง (i,j) ที่เกิดจากมดอันดับที่ μ

L_{μ} เป็นเส้นทางที่ดีลำดับที่ μ

$\Delta\tau_{ij}^{bs}$ เป็นการเพิ่มค่าของฟีโรโมนบนเส้นทาง (i,j) ที่เกิดจากมดอีลีท

σ เป็นจำนวนมดอีลีท

L^{bs} เป็นเส้นทางที่ดีที่สุดเท่าที่หามา

วิธีการของระบบมดแบบการจัดอันดับ เป็นการสร้างสมดุลระหว่าง เอ็กซ์พล้อยเตชันที่การค้นหามุ่งไปสู่ทิศของเส้นทางที่ดีของมดอีลีท และเอ็กซ์พลอเรชันที่เพิ่มจำนวนมดที่ดีเข้ามาร่วมพิจารณามากขึ้น

5.4.4 ระบบมดแบบสูงสุดต่ำสุด

การหาค่าที่เหมาะที่สุดด้วยระบบมดแบบสูงสุดต่ำสุด (Max-Min Ant System) หรือ MMAS นี้นำเสนอโดย สตูเซิล และ ฮูส (Stützle, 2000) โดยกำหนดว่ามดตัวที่ดีที่สุดเท่านั้นที่จะเป็นตัวปรับปรุงฟีโรโมนบนเส้นทาง และมีการกำหนดค่าสูงสุดและต่ำของฟีโรโมนสุดด้วย ดังนั้นสมการการปรับปรุงฟีโรโมนจะเปลี่ยนเป็นสมการ 5.24 และ 5.25 ดังนี้

$$\square_{\square\square}(\square + 1) = (1 - \square) * \square_{\square\square}(\square) + \sum_{\square=1}^{\square} \Delta\square^{\square\square\square\square}_{\square\square}(\square)$$

สมการ 5.24

$$\Delta\square_{\square\square}(\square) = \begin{cases} \frac{\square}{\square_{\square\square\square\square}}(\square) & \square\square(\square,\square) \in \square\square\square\square\ \square\square\ \square\square\square\square\ \square\square\square \\ 0 & \square\square\square\square\square\square\square\square\square\square \end{cases}$$ สมการ 5.25

เมื่อค่า L^{best} เป็นระยะทางที่มดตัวที่ดีที่สุดเดินผ่าน ซึ่งขึ้นอยู่กับเป็นการเดินของมดที่รอบ (Iteration) ไหน ถ้าเป็นรอบปัจจุบันระยะทางก็จะเป็นของรอบที่ดีที่สุด L_{ib} หรือ Iteration Best ถ้าเป็นของทุกรอบเท่าที่มีการเดินมาก็จะเป็น L_{bs} หรือ Best-so-far Best

สำหรับค่าสูงสุดและต่ำสุดของฟีโรโมนกำหนดไว้ตามสมการ 5.26 ดังนี้

$$\tau_{\square\square\square} = \frac{1}{\square} * \frac{1}{\square^*}$$ สมการ 5.26

เมื่อ L^*คือความยาวที่ดีที่สุด ในกรณีที่ไม่รู้ค่าของ L^* ให้ใช้ค่า L_{best} แทน

$$\tau_{\square\square\square} = \frac{\square_{\square\square\square}(1-\square_{\square\square\square})}{\square\square_{\square\square\square}}$$ สมการ 5.27

เมื่อ $p_{dec} = \sqrt[n]{p_{best}}$ เมื่อ n เป็นจำนวนรอบที่มดเดินมาแล้ว

ดังนั้นค่าของฟีโรโมนที่ใช้ในการคำนวณของอัลกอริธึมทั้งหมดก็คือ

$$\square_{\square\square} = \begin{cases} \square_{\square\square\square} & \square\square\ \square_{\square\square} < \square_{\square\square\square} \\ \square_{\square\square} & \square\square\ \square_{\square\square\square} \le \square_{\square\square} \le \square_{\square\square\square} \\ \square_{\square\square\square} & \square\square\ \square_{\square\square} > \square_{\square\square\square} \end{cases}$$ สมการ 5.28

5.4.5 ระบบฝูงมด

ระบบฝูงมด (Ant Colony System หรือ ACS) นี้ นำเสนอโดย โดริโก และ แกมบาร์เดลล่า (M. and Gambardella, L.M. Dorigo, 1997) จะมีความแตกต่างจากระบบอื่นในเรื่องของการปรับค่าฟีโรโมนเช่นกัน คือการปรับค่าฟีโรโมนจะเกิดจากมดทุกตัวหลังจากการเดินทาง และมดแต่ละตัวจะปรับปรุงฟีโรโมนเฉพาะเส้นทาง (e_{ij}) สุดท้ายที่มันเดินตามสมการ 5.29 ดังนี้

$$\square_{\square\square}(\square+1) = (1-\square) * \square_{\square\square}(\square) + \square\square_0$$ สมการ 5.29

เมื่อ $\varphi \in (0,1)$ คือค่าโคเอฟฟิเซียนของการลดลงของฟีโรโมน และ τ_0 คือค่าฟีโรโมนเริ่มต้น และเช่นเดียวกับระบบมดแบบสูงสุดต่ำสุด ในการปรับค่าฟีโรโมนในแต่ละรอบจะดำเนินการโดยมดตัวที่ดีที่สุดเท่านั้นตามสมการดังนี้

$$\tau_{ij} = \begin{cases} (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij} & \text{ถ้า } \tau_{ij} \in \text{เส้นทางที่ดีที่สุด} \\ \tau_{ij} & \text{ถ้าไม่ใช่} \end{cases}$$

สมการ 5.30

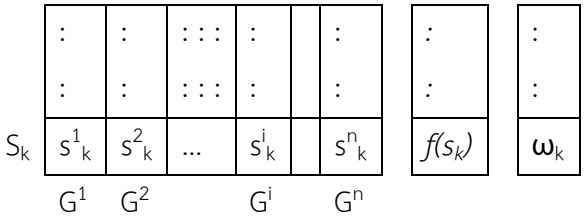
เมื่อ $\Delta\tau_{ij} = \frac{1}{\text{ความยาวเส้นทาง}}$

5.4.6 การหาค่าเหมาะที่สุดสำหรับตัวแปรที่มีค่าจำนวนจริง

การทำให้อัลกอริธึมนี้ทำงานได้ดี มีวิธีการปรับปรุงที่หลากหลายดังที่ได้กล่าวมาแล้ว แต่การทำให้การหาค่าเหมาะที่สุดแบบฝูงมดนี้สามารถแก้ปัญหาเชิงตัวเลขได้นั้น เป็นการทำให้ขอบเขตของการแก้ปัญหานั้นกว้าง และครอบคลุมมากขึ้น ในหัวข้อต่อไปนี้จะเป็นเรื่องของการหาค่าเหมาะที่สุดแบบฝูงมด สำหรับตัวแปรที่มีค่าจำนวนจริง และการทดสอบสมรรถนะ

ในการนำวิธีการหาค่าเหมาะที่สุดแบบฝูงมด มาใช้สำหรับการแก้ปัญหาเชิงตัวเลข หรือกล่าวอีกนัยหนึ่งคือ การแก้ปัญหาที่มีตัวแปรเป็นเลขจำนวนจริงนั้น อัลกอริธึมจะต้องมีการปรับ เนื่องจากอัลกอริธึมนี้ถูกออกแบบไว้สำหรับการแก้ปัญหาแบบคอมบินาทอเรียล ซึ่งมีกระบวนการเลือกเส้นทางเดินของมด อาศัยความน่าจะเป็นแบบไม่ต่อเนื่อง (Discrete Probability) เพราะการเลือกเส้นทางเดินเป็นการเลือกในลักษณะเปรียบเทียบว่าทางเลือกใดมีความน่าจะเป็นดีที่สุด ในขณะที่การแก้ปัญหาเชิงตัวเลข เป็นการกำหนดค่าที่ดีที่สุด (หรือเหมาะที่สุด) ให้กับตัวแปรที่มีค่าเป็นจำนวนจริง ดังนั้นการกำหนดค่าของตัวแปร จึงไม่สามารถใช้สมการของการหาค่าเหมาะที่สุดดังที่กล่าวมาแล้วได้

S_1	s^1_1	s^2_1	...	s^i_1		s^n_1	$f(s_1)$	ω_1
S_2	s^1_2	s^2_2	...	s^i_2		s^n_2	$f(s_2)$	ω_2
	:	:	: : :	:		:	:	:
	:	:	: : :	:		:	:	:
S_l	s^1_l	s^2_l	...	s^i_l		s^n_l	$f(s_l)$	ω_l



รูปที่ 5.11 อาร์ไคลฟส์สำหรับการเก็บคำตอบที่เป็นตัวเลือก

ในการออกแบบอัลกอริธึมสำหรับการหาค่าเหมาะที่สุดแบบฝูงมด เพื่อการแก้ปัญหาเชิงตัวเลขนั้น โซชา (Socha, 2008) ได้นำฟังก์ชันความหนาแน่นของความน่าจะเป็น (Probability Density Function) มาใช้ในรูปแบบของฟังก์ชันเกาส์เซียน (Gaussian Function) แต่เนื่องจากฟังก์ชันเกาส์เซียนเดี่ยวไม่สามารถอธิบายพื้นที่เป้าหมายที่ไม่เชื่อมต่อกันของ 2 พื้นที่ได้ ดังนั้นจึงใช้ เกาส์เซียนเคอร์เนล (Gaussian Kernel) ในการกำหนดน้ำหนักรวมของฟังก์ชันเกาส์เซียน $g^i_l(x)$ หลายๆ ฟังก์ชัน ดังสมการ 5.31 ที่หาค่า $G^i(x)$ ต่อไปนี้

$$\square^{\square}(\square) = \sum_{\square=1}^{\square} \square_{\square} \square^{\square}_{\square}(\square) = \sum_{\square=1}^{\square} \square_{\square} \frac{1}{\square^{\square}_{\square}\sqrt{2\square}} \square^{\frac{-(\square-\square^{\square}_{\square})^2}{2\square^{\square}_{\square}{}^2}} \quad \text{สมการ 5.31}$$

เมื่อ **ω** เป็นเวคเตอร์ของน้ำหนัก (Weights) ที่เกี่ยวข้องกับฟังก์ชันเกาส์เซียนแต่ละฟังก์ชัน ตัวแปร $\boldsymbol{\mu}^i$ เป็นเวคเตอร์ของค่าเฉลี่ย (Means) และตัวแปร $\boldsymbol{\sigma}^i$ เป็นเวคเตอร์ของความเบี่ยงเบนมาตรฐาน

คำตอบที่ได้จะถูกนำไปเก็บไว้ในอาร์ไคฟ์ (Archive) ในตอนเริ่มต้นคำตอบที่อยู่ในอาร์ไคฟส์นี้จะถูกกำหนดให้มีค่าเริ่มต้นในจำนวน k คำตอบ ที่สร้างขึ้นมาด้วยการสุ่ม

ในอาร์ไคฟส์จะเก็บคำตอบจำนวน K คำตอบ {*s1,s2,..sl,..sk*} และค่าที่คำนวณได้จากฟังก์ชัน {*f(s1),f(s2),..,f(sl),..f(sk)*} ที่ตัวแปรตำแหน่งที่ i ของคำตอบที่ l จะเขียนอยู่ในรูปของ s_i^l คำตอบที่อยู่ในอาร์ไคฟ์จะเก็บเรียงตามลำดับของคุณภาพ สำหรับการหาค่าต่ำสุดเราจะได้

$$f(s_1) \le f(s_2) \le \ldots \le f(s_l) \le \ldots \le f(s_k)$$

สำหรับแต่ละคำตอบ s_l มีค่าน้ำหนักเป็น **ω** ที่เป็นไปตามคุณภาพของคำตอบ ดังนั้นเราจะได้

$$\omega_1 \geq \omega_2 \geq \ldots \geq \omega_l \geq \ldots \geq \omega_k$$

เมื่อค่าน้ำหนัก ω_l ของคำตอบ s_l เป็นค่าที่ได้จากการคำนวณของสมการดังต่อไปนี้

$$\square_{\square} = \frac{1}{\square\square\sqrt{2\square}} \square^{\frac{-(\square-1)^2}{2\square^2\square^2}}$$ สมการ 5.32

ค่า ω เป็นน้ำหนักของฟังก์ชันเกาส์เซียนฟังก์ชันที่ l ที่กำหนดให้มีค่าเฉลี่ยเท่ากับ 1.0 มีค่าเบี่ยงเบนมาตรฐานเท่ากับ qk เมื่อ q เป็นพารามิเตอร์ของอัลกอริธึม ซึ่งควรจะมีค่าน้อยๆ

สำหรับเวคเตอร์ของค่าเฉลี่ย μ^i ของแต่ละ $G^i(x)$ ที่ตัวแปรที่ i คือ

$$\mu^i = \{\mu_1^i, \ldots, \mu_k^i\} = \{s_1^i, \ldots, s_k^i\}$$

ตัวแปร σ^i เป็นเวคเตอร์ของค่าความเบี่ยงมาตรฐานหาได้จากสมการ 5.33 ดังต่อไปนี้

$$\square_{\square}^{\square} = \xi \sum_{\square=1}^{\square} \frac{|\square_{\square}^{\square} - \square_{\square}^{\square}|}{\square - 1}, \square = 1,2,\ldots,\square$$ สมการ 5.33

ค่าของ $\xi > 0$ สำหรับทุกมิติของคำตอบ ซึ่งเป็นค่าที่แสดงผลเช่นเดียวกันกับการระเหยของฟีโรโมนของการหาเหมาะที่สุดแบบฝูงมดแบบธรรมดา สำหรับตัวแปร ξ ที่มีค่าสูง จะทำให้ความเร็วของการลู่เข้าสู่คำตอบลดลง

ในระหว่างการเดินทางของมดจากคำตอบหนึ่งเช่น s_l ภายในอาร์ไคฟ์ของคำตอบ ด้วยการใช้วิธีการทางสถิติ เช่นวงล้อลูเล็ตต์ ทำให้คำตอบที่ถูกจัดลำดับไว้สูงกว่า มีโอกาสที่ถูกเลือกจากมดได้สูงกว่า ความเบี่ยงเบนมาตรฐานระหว่างค่าของตัวแปร i ของ s_l จะถูกคำนวณตามสัดส่วนของตัวแปร s_l^i ที่แสดงออกมาเป็นค่าของ σ_l^i

โดยการใช้วิธีการของ บ็อกซ์-มูลเลอร์ (Box-Muller) (Box, 1958) ที่มีค่าเฉลี่ยเท่ากับ s_l^i และความเบี่ยงเบนมาตรฐานเป็น σ_l^i สำหรับตัวแปรตัวที่ i ดังสมการ 5.33 ข้างต้น ในการหาค่านี้ เมื่อหาจนครบทุกตัวแปร n ตัว เราก็จะได้ค่าของ $f(s_1)$ สำหรับมดแต่ละตัว ในท้ายที่สุดเราก็จะได้คำตอบใหม่

สำหรับมดทุกตัว ให้นำคำตอบเหล่านั้นมาเพิ่มในอาร์ไคฟ์ หลังจากรวบรวมคำตอบได้ครบ k คำตอบ คำตอบส่วนที่เหลือก็จะถูกลบออกไป

5.4.7 การทดสอบสมรรถนะ

โซชา (Socha, 2008) ได้ทำการทดสอบสมรรถนะของอัลกอริธึมของเขา โดยนำอัลกอริธึมไปใช้ในการหาค่าเหมาะที่สุดของฟังก์ชันทดสอบมาตรฐาน เพื่อเทียบเคียงกับอัลกอริธึมประเภทอื่น โดยที่เขาได้ตั้งค่าของพารามิเตอร์ให้กับอัลกอริธึม โดยที่มีฟังก์ชันที่นำมาใช้มีจำนวนมาก ในที่นี้จะนำเสนอเฉพาะบางฟังก์ชันพร้อมทั้งการกำหนดขอบเขตของการค้นหาดังต่อไปนี้

ฟังก์ชันรูปทรงกลม (Sphere model: SM)
ค่าเวคเตอร์ของตัวแปร $\in$ [-5.12, 5.12]^n, n = 6
ฟังก์ชันของโรเซนบร็อก (Rosenbrock: R_n)
ค่าเวคเตอร์ของตัวแปร $\in$ [-5, 10]^n, n = 2, 5
ฟังก์ชันของกรีวังค์ (Griewangk: GR_n)
โดยค่าเวคเตอร์ของตัวแปร $\in$ [-5.12, 5.12]^n, n = 10

ในการกำหนดเงื่อนไขของการหยุดที่เนื่องจากการพบค่าเหมาะที่สุด หรือค่าที่ดีที่สุดพบ จะกำหนดจากสมการดังต่อไปนี้

$$|f - f^*| < \epsilon_1 f + \epsilon_2$$

เมื่อค่า f เป็นค่าที่ดีที่สุดที่ค้นพบโดยอัลกอริธึม ค่าของ f* เป็นค่าที่ดีที่สุดของฟังก์ชัน ที่เราทราบอยู่แล้ว (สำหรับฟังก์ชันที่นำมาแสดงคือ 0) และค่า ϵ_1 และ ϵ_2 เป็นค่าความคลาดเคลื่อนสมบูรณ์ โดยกำหนดให้ $\epsilon_1 = \epsilon_2 = 10^{-4}$ ผลของการทดลองแสดงไว้ในตารางที่ 5.2 ในการทดสอบนี้จะให้อัลกอริธึมทำงาน 20 ครั้งต่อ 1 ฟังก์ชัน ตัวเลขในคอลัมน์ที่สอง เป็นมีเดียนของจำนวนครั้งที่อัลกอริธึมเรียกใช้ฟังก์ชันก่อนพบคำตอบที่ดีที่สุด คอลัมน์ที่สามเป็นร้อยละของการทำงานที่พบคำตอบที่ดีที่สุด ในกรณีที่ไม่ถึง 100% หมายถึงในการทำงานบางครั้งจะไม่พบคำตอบที่ดีที่สุด

ฟังก์ชัน	จำนวนครั้งที่ฟังก์ชันถูกเรียกใช้งาน (มีเดียน)	ร้อยละของการทำงานที่พบคำตอบ
โรเซนบร็อก (R2)	820	100 %
โรเซนบร็อก (R5)	2487	97 %
ทรงกลม	781	100 %
กรีวังค์ (GR10)	1390	61 %

ตารางที่ 5.2 ผลของการทดสอบสมรรถนะ

5.5 สรุป

มดแต่ละตัวที่มีความสามารถค่อนข้างจำกัด แต่มดแต่ละตัวสามารถทำงานร่วมกัน เพื่อหาเส้นทางที่สั้นที่สุดจากรังไปยังแหล่งอาหารได้ มดสื่อสารกันเองโดยใช้สารเคมีที่มดผลิตขึ้นชื่อฟีโรโมน (Pheromones) เมื่อมดเดินทาง มดจะทิ้งฟีโรโมนไว้ตามทางเพื่อเป็นสัญลักษณ์ มดตัวอื่นที่จะตามมาทีหลังก็จะใช้กลิ่นของฟีโรโมนนี้เป็นตัวตัดสินว่าจะเลือกเดินบนเส้นทางใด โดยมดโดยส่วนใหญ่จะเลือกเดินบนเส้นทางที่มีกลิ่นของฟีโรโมนมาก ลักษณะของการทำงานร่วมกันดังกล่าวข้างต้น ปิแอร์ พอล กราเซ เรียกว่า สติกเมอร์จี (Stigmergy)

ในการทดลองกับสะพานคู่ ที่มีเส้นทางหนึ่งสั้น และอีกเส้นทางหนึ่งยาว มดจะเดินจากรังมด ไปยังแหล่งอาหาร ในตอนแรก ฝูงมดจะใช้เส้นทางทั้งสองด้วยจำนวนเท่าๆ กัน พร้อมกับทิ้งฟีโรโมนตามทางที่มดแต่ละตัวเดิน เมื่อระยะเวลาผ่านไป เส้นทางที่สั้นกว่าจะมีฟีโรโมนที่มากกว่า เนื่องจากมดสามารถเดินไปและกลับได้เร็วกว่า ในที่สุด ทำให้เส้นทางที่สั้นก็จะได้รับการเลือกจากมดเกือบทั้งฝูง

แต่อย่างไรก็ตาม จะมีมดบางตัวที่ไม่เดินตามกฎนี้ คือไม่เลือกเดินตามทางที่มีฟีโรโมนหนาแน่น ซึ่งในทางระบบ ถือว่าเป็นเรื่องที่ดี เพราะจะเป็นโอกาสที่มดตัวนี้ได้เส้นทางใหม่ที่ดีกว่า หรืออาจจะได้พบแหล่งอาหารใหม่ เราเรียกว่าปรากฏการณ์นี้ว่า การเกิดมิวเตชัน (Mutation)

จากผลการทดลองของการหาอาหารของมดดังที่ได้กล่าวมาข้างต้น โดริโก ได้พัฒนาเป็นวิธีการหาค่าที่เหมาะที่สุดแบบฝูงมดขึ้น ด้วยการใช้ฝูงมดมาช่วยในการค้นหาคำตอบจากที่ดีที่สุดจากปริภูมิการค้นหา โดยอัลกอริธึมจะมีขบวนการของการทำวนซ้ำของ 3 ขั้นตอนดังต่อไปนี้

การสร้างคำตอบที่เป็นไปได้

การหาคำตอบที่ดีแบบท้องถิ่น (Local optimum)

การปรับค่าฟีโรโมน

การทำงานของขั้นตอนทั้ง 3 ข้างต้น จะวนซ้ำจนกว่าจะได้คำตอบที่ต้องการ

การหาคำตอบที่เป็นไปได้ จะเริ่มด้วยการสร้างเส้นทางเดินทั้งหมดของมด หรืออาจจะเรียกได้ว่าเป็นการสร้าง ปริภูมิปัญหา เมื่อได้ปริภูมิปัญหาแล้ว มดแต่ละตัวจะทำการหาคำตอบที่ดีที่สุดบนปริภูมิปัญหานี้ ซึ่งเราเรียกกระบวนการนี้ว่า การหาคำตอบที่ดีแบบท้องถิ่น ในการหาคำตอบที่ดีที่สุดของมดแต่ละตัวนี้ มดจะอาศัยการตัดสินใจด้วยความน่าจะเป็นตามสมการดังต่อไปนี้

$$P(e_{i,j}) = \frac{\tau_{i,j}(t)^{\alpha} * \eta_{ij}{}^{\beta}}{\sum_{l \in N_i^k} (\tau_{i,l}(t)^{\alpha} * \eta_{il}{}^{\beta})}, j \in N_i^k .$$

เมื่อ $\tau_{ij}(t)$ เป็นค่าของฟีโรโมน

η_{ij} เป็นค่าข้อมูลฮิวริสติก

α และ β เป็นค่าพารามิเตอร์ที่กำหนดความสัมพันธ์ของค่าฟีโรโมน (τ) และค่าข้อมูลฮิวริสติก (η)

N_i^k ก็คือเมืองทั้งหมดในเซท N ที่มดตัวที่ k ยังไม่ได้เดินผ่าน

เมื่อมดทุกตัวเสร็จสิ้นการสร้างคำตอบของมันเอง ค่าฟีโรโมนก็จะถูกปรับค่า ในการปรับค่าฟีโรโมนจะมีอยู่ 2 ส่วนคือ การลดลงของฟีโรโมนเนื่องจากการระเหย และการเพิ่มของฟีโรโมนเนื่องจากมดได้ปล่อยฟีโรโมนออกมาเพิ่มเติมบนเส้นทางที่มดเดินผ่าน ซึ่งเป็นไปตามสมการดังนี้

$$\tau_{i,j}(t+1) = (1-\rho) * \tau_{i,j}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k}(t)$$

เมื่อ $0<\rho\leq 1$ เป็นอัตราการระเหยของฟีโรโมน

$\Delta\tau^{k}_{ij}(t)$ เป็นจำนวนฟีโรโมนที่เพิ่มขึ้นเนื่องจากมดทั้ง k ตัวที่นำไปวางทิ้งไว้บนเส้นทางที่มันเดินผ่าน ซึ่งมันถูกกำหนดให้มีค่าดังนี้

$$\Delta\tau^{k}{}_{i,j}(t) = \begin{cases} 1/L^{k}(t) & if\ (i,j) \in tour - of\ - ant - k^{th} \\ 0 & otherwise \end{cases}$$

เมื่อ ค่า L^{k} เป็นระยะทางที่มดตัวที่ k เดินผ่าน

เมื่อมดทุกตัวได้เส้นทางที่ดีที่สุดแล้ว เส้นทางที่ได้ของมดทั้งหมดจะถูกนำมาเปรียบเทียบกันเพื่อหาเส้นทางที่ดีที่สุดอีกครั้ง ในการหาเส้นทางที่ดีที่สุดครั้งนี้เป็นการหาค่าที่เหมาะที่สุดแบบสากล เมื่อได้เส้นทางแล้ว ค่าของฟีโรโมนของเส้นทางที่ถูกเลือกก็จะถูกปรับค่า แล้วเส้นทางนี้จะถูกบันทึกไว้ว่าเป็นเส้นทางที่ดีทีสุดเท่าที่หามา (Best-so-far Solution) จากนั้นมดทุกตัวจะเริ่มทำการค้นหาเส้นทางที่ดีที่สุดอีกรอบ เส้นทางที่ดีที่สุดของมดแต่ละตัวก็จะถูกนำมาเปรียบเทียบกันอีกเพื่อหาเส้นทางที่ดีที่สุด จากนั้นก็จะมีการปรับค่าฟีโรโมน และเส้นทางที่ดีที่สุดในรอบนี้จะถูกนำไปเปรียบเทียบกับเส้นทางที่ดีที่สุดเท่าที่หามา เพื่อจะได้เส้นทางที่ดีที่สุดเท่าที่หามา ที่ดีที่สุด แล้วมดทุกตัวจะเริ่มทำการค้นหาคำตอบที่ดีที่สุดอีก และการทำงานนี้จะวนเวียนไปหลายๆ รอบจนกระทั่งพบคำตอบที่ดีที่สุด

การหาค่าเหมาะที่สุดแบบฝูงมด เป็นเมตาฮิวริสติกที่ออกแบบสำหรับใช้ในการแก้ปัญหาแบบคอมบินาทอเรียล เช่นปัญหาการเดินทางของเซลส์แมน ปัญหาการวางผังโรงงาน หรือปัญหาในการจัดตารางชนิดต่างๆ เนื่องจากวิธีการหาค่าเหมาะที่สุดแบบนี้อาศัยการเปรียบเทียบคำตอบที่เป็นไปได้ เพื่อคำตอบที่ดีที่สุด โดยมีอ็อปเจ็คทีปฟังก์ชันที่ใช้สำหรับคำนวณค่าฟิตเน็สของคำตอบที่เป็นไปได้เหล่านั้น แต่อย่างไรก็ตาม การนำการหาค่าเหมาะที่สุดแบบฝูงมดไปใช้ในการหาค่าเหมาะที่สุดเชิงตัวเลข ก็สามารถเป็นไปได้เช่นกัน

ในการหาค่าที่เหมาะที่สุดแบบฝูงมด เอ็กซ์พลอเรชันจะเกิดขึ้นเมื่อมดทำการสำรวจเส้นทางที่มดจะเดิน ในการค้นหาคำตอบที่เหมาะที่สุด การทำการสำรวจเส้นทางนี้เป็นกระบวนการในการสร้างคำตอบที่เป็นไปได้ให้กับการค้นหา สำหรับเอ็กซ์พล้อยเตชัน เป็นช่วงของการปรับฟีโรโมน ซึ่งเป็นการให้น้ำหนักกับเส้นทางที่ดีให้มากขึ้น และฟีโรโมนของเส้นทางที่ไม่ดีก็จะถูกทำให้ลดลงเรื่อยๆ ด้วยการระเหย

เนื่องจากว่า การหาค่าเหมาะที่สุดแบบฝูงมดนี้ ผลลัพธ์ที่ได้ออกมายังไม่สามารถเอาชนะวิธีการหาค่าเหมาะที่สุดแบบอื่นได้ ดังนั้นการปรับปรุงวิธีการหาค่าเหมาะที่สุดแบบฝูงมด เพื่อเพิ่มประสิทธิภาพของเอ็กซ์พลอเรชัน และเอ็กซ์พล้อยเตชันจึงเกิดขึ้น ระบบมดอีลีทเป็นระบบที่ให้ความสำคัญกับเส้นทางที่ดีที่สุดเท่าที่ค้นหามา ด้วยการเพิ่มน้ำหนักให้กับเส้นทางนั้น ดังนั้นระบบนี้จึงทำให้เอ็กซ์พล้อยเตชันของการหาคำตอบเข้มข้นขึ้น ในขณะเดียวกัน ระบบมดแบบจัดอันดับ ได้เพิ่มจำนวนมดอีลีท ด้วยการจัดอันดับของเส้นทางที่ดีที่สุด และเพิ่มเส้นทางเหล่านั้นเข้ามาในกระบวนการค้นหา ทำให้ระบบนี้นอกจากจะมีเอ็กซ์พล้อยเตชันที่เข้มข้นแล้ว การเพิ่มจำนวนมดอีลีทยังเป็นการเพิ่มเอ็กซ์พลอเรชันให้มีขอบเขตการค้นหาที่กว้างขึ้นด้วย ในระบบมดแบบสูงสุดต่ำสุด เป็นการกำหนดขอบเขตของการปรับค่าฟีโรโมนโดยมีการกำหนดค่าสูงสุดและต่ำสุด ทั้งนี้เพื่อลดความเร็วของการลู่เข้าสู่คำตอบ และเป็นการป้องกันการลู่เข้าสู่คำตอบก่อนเวลาอันควร ซึ่งจะเพิ่มโอกาสของเอ็กซ์พลอเรชัน เพื่อการค้นหาคำตอบในพื้นที่ใหม่ด้วย

อัลกอริธึมของการหาค่าเหมาะที่สุดแบบฝูงมดเท่าที่มีการนำเสนอมา ยังขาดเอ็กซ์พลอเรชันที่ดี อัลกอริธึมที่นำเสนอเกี่ยวกับการมิวเตชัน เช่นการที่มดบางตัวไม่เดินตามกลิ่นของฟีโรโมนที่เข้มที่สุด เพื่อทำให้ระบบทั้งระบบเกิดเอ็กซ์พลอเรชันที่ดียังมีไม่มากนัก ความหลากหลายของการพัฒนาอัลกอริธึมแบบนี้มักจะมุ่งเป้าไปที่การสร้างเอ็กซ์พล้อยเตชัน เช่นระบบมดอีลีท ระบบมดแบบการจัดอันดับ เป็นต้น

5.6 แบบฝึกหัด

1. ในการทำงานร่วมกันของมด คำว่า สติกเมอร์จี (Stigmergy) มีความหมายว่าอะไร แล้วให้ความหมายของคำต่อไปนี้
   1.1. Multiple interactions
   1.2. Randomness
   1.3. Positive feedback
   1.4. Negative feedback

   โดยให้นิยามคำข้างต้นด้วยคำพูดของตัวเอง แล้วอธิบายว่าคำทั้ง 4 ข้างต้นว่ามีความสัมพันธ์กันอย่างไร ในการอธิบายการทำงานของฝูงมด ในมุมของการวางฟีโรโมน และการเดินตามฟีโรโมน

2. ให้พิสูจน์ว่าถ้ามดมีความสามารถในการจำเส้นทางที่มันเดินไป แล้วกลับตามเส้นทางเดิมแล้ว มดทั้งฝูงจะสามารถลู่เข้าสู่สะพานเส้นที่สั้นทั้งหมดได้

3. จากการทดลองเรื่องสะพานคู่ที่มีทางเดินที่สั้น และยาว ถ้ามดมีความสามารถในการจำเส้นทางที่มันเดินไป แล้วกลับตามเส้นทางเดิมแล้ว ให้อธิบายการทำงานของฝูงมด เมื่อมีเส้นทางเดินเป็น 3 เส้นทางจากรังไปยังแหล่งอาหาร ให้หาค่าความน่าจะเป็น (p) ของการเลือกเส้นทาง และการปรับค่าฟีโรโมน (τ) ของมดบนเส้นทางทั้ง 3 ดังกล่าว

4. จากการทดลองเรื่องสะพานคู่ ถ้ามีมดอยู่ 2 ฝูงทำการทดลองร่วมกัน มดฝูงแรกมีจำนวน 20 ตัว และมดฝูงที่ 2 มีจำนวน 50 ตัว ท่านคิดว่ามดฝูงใดจะเลือกเส้นทางสั้นทั้งฝูงก่อนกัน เพราะอะไร

5. ให้ทดลองเขียนโปรแกรมการทดลองเรื่องสะพานคู่ แล้วทดลองหาความแตกต่างของการลู่เข้าสู่คำตอบของฝูงมดที่มีจำนวนที่ต่างกัน 5, 50, 150, 200 และ 500 ตัว โดยใช้สมการ

$$P_i = \frac{\tau_i}{\tau_1 + \tau_2}, i = 1,2,$$ และ

$$\tau_i(t+1) \leftarrow \tau_i(t) + \frac{Q}{l_i},$$

ให้อธิบายความหมายของการลู่เข้าสู่คำตอบที่เกิดจากสมการทั้งสองว่า เป็นการปรับค่าทางสถิติ ไม่ใช่มาจากจำนวนตัวมด

6. จากสมการ

$$\tau_{i,j}(t+1) = (1-\rho) * \tau_{i,j}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k}(t)$$ และ

$$\Delta\tau^{k}{}_{i,j}(t) = \begin{cases} 1/L^{k}(t) & if\ (i,j) \in tour - of\ - ant - k^{th} \\ 0 & otherwise \end{cases}$$

ที่อธิบายการปรับค่าฟีโรโมน และอธิบายว่าตัวแปร ρ มีผลต่อการเดินทางของฝูงมดอย่างไร

7. ให้เขียนโปรแกรมคอมพิวเตอร์เพื่อจำลองการทำงานของมดบนสะพานคู่ โดยใช้สมการดังต่อไปนี้

$$P(e_{i,j}) = \frac{\tau_{i,j}(t)^{\alpha} * \eta_{ij}{}^{\beta}}{\sum_{l \in N_i^k} (\tau_{i,l}(t)^{\alpha} * \eta_{il}{}^{\beta})}, j \in N_i^k.$$

$$\tau_{i,j}(t+1) = (1-\rho) * \tau_{i,j}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k}(t)$$

$$\Delta\tau^{k}{}_{i,j}(t) = \begin{cases} 1/L^{k}(t) & if\ (i,j) \in tour - of\ - ant - k^{th} \\ 0 & otherwise \end{cases}$$

กำหนดให้ η = 1 และ β = 1 แล้วหาว่าค่าของ α ที่เปลี่ยนไป มีผลอย่างไรต่อการลู่เข้าสู่คำตอบของฝูงมด

8. เมตาฮิวริสติกของการหาค่าเหมาะที่สุดแบบฝูงมดมีลักษณะของเอ็กซ์พลอเรชันและเอ็กซ์พล้อยเตชันเป็นอย่างไร

9. ให้ใช้สมการดังต่อไปนี้

$$P(e_{i,j}) = \frac{\tau_{i,j}(t)^{\alpha} * \eta_{ij}^{\beta}}{\sum_{l \in N_i^k} (\tau_{i,l}(t)^{\alpha} * \eta_{il}^{\beta})}, j \in N_i^k .$$

$$\tau_{i,j}(t+1) = (1-\rho) * \tau_{i,j}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$$

$$\Delta\tau^{k}{}_{i,j}(t) = \begin{cases} 1/L^{k}(t) & if\ (i,j) \in tour - of\ - ant - k^{th} \\ 0 & otherwise \end{cases}$$

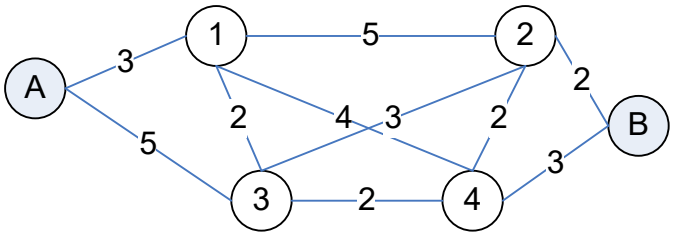
โดยกำหนดให้ η = 1 และ β = 1 และ α =1 ทดลองแก้ปัญหาการเดินทางของเซลส์แมน โดยใช้การคำนวณด้วยมือ แล้วหาคำตอบของโจทย์ดังต่อไปนี้

	A	B	C	D
A	0	5	6	10
B	-	0	9	8
C	-	-	0	7
D	-	-	-	0

9.1. ให้แก้ปัญหาข้างต้น

9.2. ให้อธิบายการลู่เข้าสู่คำตอบ

10. จากเส้นทางเดินตามรูปข้างล่าง เป็นกราฟที่แสดงถึงเส้นทางของการเดินทางจากคลังสินค้า (A) ไปยังเป้าหมายปลายทาง (B) ตัวเลขที่กำกับไว้บนเส้นระหว่างโหนดเป็นตัวเลขที่บอกระยะทาง และโหนด 1-4 เป็นทางแยก ซึ่งระยะทางและเวลาในการเดินทางผ่านทางแยกจะไม่นำมาคิด



รูปแสดงกราฟแสดงเส้นทางระหว่างคลังสินค้า (A) และเป้าหมายปลายทาง (B)

10.1. ให้ทดลองใช้วิธีการหาค่าเหมาะที่สุดแบบฝูงมดหาเส้นทางที่ดีที่สุดของเส้นทางดังกล่าว

10.2. ท่านคิดว่าขนาดของอัตราการระเหยของฟีโรโมน มีผลต่อความเร็วในการหาคำตอบของ ACO หรือไม่ ให้พิสูจน์ด้วยการคำนวณจากปัญหาข้างต้น

11. สมมุติว่าท่านเป็นนักวิจัยฝูงหุ่นยนต์ (Swarm Robotics) และท่านจะต้องคิดค้นอัลกอริธึมสำหรับหุ่นยนต์ฝูงนี้ในการหาเส้นทางที่สั้นที่สุดระหว่างคลังสินค้าและเป้าหมายปลายทาง ตามเส้นทางที่แสดงไว้ตามโจทย์ข้อ 10 สมมุติว่าเส้นทางที่เป็นไปได้ทั้งหมดถูกกำหนดค่าฟีโรโมนไว้แล้วด้วยเครื่องมือจำลอง (เช่น smart RFID tags) ที่หุ่นยนต์เหล่านี้สามารถอ่านและแก้ไขค่าฟีโรโมนได้ และค่าฟีโรโมนนี้ก็สามารถทำให้ระเหยได้ตามกาลเวลา ให้อธิบายว่าท่านจะออกแบบฝูงหุ่นยนต์เหล่านี้ให้ทำหน้าที่แบบเดียวกับฝูงมดได้อย่างไร

12. อธิบายการหาค่าเหมาะที่สุดแบบฝูงมดสำหรับแก้ปัญหาที่มีตัวแปรที่มีค่าเป็นจำนวนจริงในหัวข้อดังต่อไปนี้

12.1. เกาส์เซียนเคอร์เนล (Gaussian Kernel) และฟังก์ชันเกาส์เซียนต่างกันอย่างไร

12.2. โซซาได้นำเกาส์เซียนเคอร์เนลมาใช้เพื่ออะไร

12.3. อาร์ไคฟ์สำหรับการเก็บคำตอบที่เป็นตัวเลือกใช้สำหรับแก้ปัญหาได้อย่างไร และทำไมถึงต้องมี

12.4. เขียนอัลกอริธึมของโซซาในรูปของรหัสเทียม

13. อธิบายวิธีการที่อัลกอริธึมของโซซาหาค่าเหมาะที่สุดของฟังก์ชันรูปทรงกลม

บทที่ 6 การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค Particle Swamp Optimization

6.1 คำนำ

การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค (Particle Swamp Optimization) หรือ PSO เป็นเทคนิคทางสถิติที่อาศัยหลักการทำงานของกลุ่มประชากรเช่นเดียวกับอัลกอริธึมพันธุการ และการหาค่าเหมาะที่สุดแบบฝูงมด การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคนี้พัฒนาโดยอีเบอร์ฮาร์ดและเคนเนดี (J. and Eberhart, R. Kennedy, 1995) ที่จำลองวิธีการหาค่าเหมาะที่สุดมาจากพฤติกรรมการเดินทางของฝูงสัตว์ โดยเฉพาะฝูงนก และฝูงปลา ในการหาอาหาร

เรย์โนล์ด (Reynolds, 1987) และ แฮบเนอร์ กับ เจอร์แนนเดอร์ (Heppner, 1990) ได้จำลองรูปแบบการบินของฝูงนก พร้อมกับศึกษาพฤติกรรมของนกที่บินเป็นฝูง เขาได้ค้นพบว่านกแต่ละตัวมีการประสานงานกันในระหว่างการบิน โดยที่นกแต่ละตัวจะพยายามรักษาความห่างระหว่างกัน คือของตัวมันเอง และนกตัวที่อยู่รอบข้างมันทั้งหมด ให้เหมาะสม เพื่อที่พวกมันจะได้ไม่บินชนกันเอง แม้ว่านกทั้งฝูงจะเปลี่ยนทิศทางการบินอย่างรวดเร็วก็ตาม เมื่อฝูงนกพบสิ่งกีดขวาง พวกมันก็จะกระจายกันออกไป และกลับเข้ามารวมกลุ่มกันใหม่โดยที่ไม่มีการชนกันเอง

ก่อนหน้านั้น มีนักวิจัยชื่อ วิลสัน (Wilson, 1975) ได้กล่าวถึงประโยชน์ของนกแต่ละตัวที่จะได้เนื่องจากการบินกันเป็นฝูงเพื่อการหาอาหารร่วมกันว่า อย่างน้อยในทางทฤษฎี นกแต่ละตัวในฝูงก็จะได้ประโยชน์จากประสบการณ์ในการหาอาหารของสมาชิกในฝูง โดยเฉพาะเมื่อแหล่งอาหารอยู่อย่างกระจัดกระจาย และไม่ทราบตำแหน่งที่แน่ชัด นกตัวที่มีประสบการณ์ก็จะสามารถนำทางให้นกทุกตัวได้เข้าถึงแหล่งอาหารได้ ซึ่งโดยรวมแล้วการหาอาหารแบบเป็นฝูงนี้ อาจจะดีกว่าที่นกแต่ละตัวจะแยกกัน

ไปหาอาหาร ทำให้ข้อดีของการหาอาหารแบบเป็นฝูงของนกดังกล่าวลบความเสียเปรียบที่เกิดจากการแย่งชิงอาหารระหว่างนกแต่ละตัวในกลุ่มให้หมดไป

การหาอาหารแบบเป็นฝูงดังที่ได้กล่าวมาแล้ว ไม่ได้เกิดเฉพาะกับนกเท่านั้น แต่ยังเกิดกับสัตว์ชนิดอื่นที่อยู่กันเป็นฝูงด้วยเช่น ปลา ม้า หรือแม้กระทั่งมนุษย์ เพื่อที่จะทำให้การอธิบายดังกล่าวครอบคลุมถึงพฤติกรรมของสิ่งมีชีวิตมากชนิดขึ้น ดังนั้นในการจำลองการเคลื่อนที่ของฝูงสัตว์เหล่านี้ จึงมีการดัดแปลงให้เป็นการเคลื่อนที่ของกลุ่มอนุภาค และกำหนดให้สมาชิกแต่ละตัวที่อยู่ในกลุ่มมีชื่อเรียกว่าอนุภาค

เรโนล์ด (Reynolds, 1987) ได้สรุปโมเดลพฤติกรรมของฝูงนกในระหว่างการหาอาหารว่า นกแต่ละตัวจะรักษากฎ 3 ประการดังนี้

- การแยกกัน (Separation) นกแต่ละตัวจะแยกออกจากกันถ้าพวกมันเข้าใกล้กันเกินไป
- การจัดแนว (Alignment) นกทุกตัวจะมุ่งไปในทิศทางเฉลี่ยเดียวกับเพื่อนที่อยู่ข้างๆ กัน
- การประสานงานกัน (Cohesion) นกทุกตัวจะเคลื่อนที่ไปในตำแหน่งที่มีระยะห่างเดียวกันโดยเฉลี่ยกับเพื่อนที่อยู่ข้างๆ กัน

อีเบอร์ฮาร์ดและเคนเนดี ได้ปรับโมเดลของเรย์โนล์ด เพื่ออธิบายการบินของนกทั้งฝูงที่บินไปเกาะยังตำแหน่งที่มีอาหารอยู่ว่า

- นกแต่ละตัวจะเกาะติดกับตำแหน่งของมัน
- นกแต่ละตัวจะจำได้ว่ามันอยู่ห่างจากตำแหน่งที่มันจะเกาะเท่าไร
- นกแต่ละตัวจะแลกเปลี่ยนข้อมูลกับตัวที่อยู่ข้างๆ เกี่ยวกับระยะห่างของตำแหน่งของมัน
- ท้ายที่สุด นกทุกตัวจะบินลงไปที่ตำแหน่งของมัน

สมมุติว่า มีนกฝูงหนึ่งกำลังบินสุ่มหาอาหารอยู่ที่พื้นที่บริเวณหนึ่ง ที่ที่ซึ่งมีอาหารเพียงชิ้นเดียวในบริเวณนั้น นกทุกตัวไม่รู้ว่าอาหารอยู่ที่ไหน แต่โดยสัญชาตญาณ ทำให้พวกมันรู้ว่ามันอยู่ห่างจากอาหารเท่าไร โดยที่เมื่อพวกมันบินไปแต่ละช่วง พวกมันจะหาระยะห่างของตัวมันจากแหล่งอาหาร และ

นกทั้งฝูงจะเลือกบินตามนกตัวที่อยู่ใกล้อาหารมากที่สุด จากนั้นพวกมันก็จะบินช่วงต่อไป นกทั้งฝูงจะทำเช่นนี้ไปเรื่อยๆ จนพวกมันบินถึงแหล่งอาหาร ในการกำหนดระยะห่างของมันจากแหล่งอาหาร และการตัดสินใจเลือกบินตามนกบินตัวที่อยู่ใกล้แหล่งอาหารมากที่สุด แต่ละช่วงจะใช้เวลาน้อยมาก จนเราดูเสมือนเป็นการบินแบบต่อเนื่อง ยังมีสัตว์หลายประเภทที่มีพฤติกรรมเชิงกลุ่มแบบเดียวกับฝูงนก คือฝูงปลาที่หาอาหารในน้ำ ฝูงม้าป่า ควายป่า และฝูงช้างที่เดินทางหาอาหารในทุ่งหญ้า เป็นต้น

วิธีการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค จำลองกระบวนการดังกล่าวมาใช้ในการหาค่าที่เหมาะที่สุด สำหรับวิธีการนี้ นกที่อยู่ที่ตำแหน่งต่างๆ จะเรียกว่า “อนุภาค (Particle)” ตำแหน่งของอนุภาคแต่ละตัวก็เป็นเสมือนคำตอบที่เป็นไปได้ ตำแหน่งที่ดีที่สุดของอนุภาคทั้งกลุ่มในการเดินทางตั้งแต่ต้นจนจบ จะถือเป็นค่าที่เหมาะที่สุด อนุภาคทุกตัวจะมีค่าฟิตเน็ส ค่านี้หาได้จาก ฟิตเน็สฟังก์ชัน ฟังก์ชันนี้เปรียบเสมือนสัญชาตญาณของนก ที่ใช้สำหรับการหาค่าที่เหมาะที่สุด ค่าความเร็วในการเดินทางจะเป็นตัวกำกับระยะการเดินทางของอนุภาค อนุภาคเหล่านี้จะเดินทางไปในปริภูมิปัญหา โดยที่ในระหว่างการเดินทางนี้ อนุภาคทุกตัวจะเดินทางตามอนุภาคที่อยู่ใกล้ตำแหน่งที่เหมาะที่สุด

ในการจำลองพฤติกรรมการหาค่าเหมาะที่สุดแบบฝูงอนุภาค เพื่อใช้สำหรับการค้นหาค่าเหมาะที่สุดนี้ ตำแหน่งของอนุภาคแต่ละตัวจะถือเป็นคำตอบที่เป็นไปได้ของปัญหา และด้วยการใช้ประสบการณ์ของตัวมันเองรวมกับของเพื่อนข้างๆ นกทั้งฝูงก็จะเลือกได้ว่าจะเคลื่อนที่ไปในทิศทางใดของปริภูมิปัญหา การเคลื่อนที่แต่ละครั้ง ตำแหน่งใหม่ของอนุภาค จะเป็นคำตอบที่เป็นไปได้ที่เกิดขึ้นใหม่ และคำตอบเหล่านี้จะมีการพัฒนาให้เข้าใกล้ค่าที่เหมาะที่สุดมากขึ้น เปรียบเสมือนฝูงนกที่เคลื่อนที่เข้าหาแหล่งอาหาร ในทางปฏิบัติ ที่จุดเริ่มต้นของการทำงาน จะมีการกำหนดตำแหน่ง และความเร็วเริ่มต้นของอนุภาคทุกตัว ก่อนการค้นหาคำตอบ สำหรับการจำลองพฤติกรรม เราจะใช้ อ็อปเจ็คทีฟฟังก์ชัน เป็นตัวบอกสัญชาตญาณของนก หรือของอนุภาคในการจำลองแบบ ในขณะที่อนุภาคเคลื่อนที่ไปในปริภูมิปัญหา อนุภาคจะจดจำตำแหน่งที่ดีที่สุดที่มันได้พบมา ที่แต่ละรอบของการทำงาน อนุภาคแต่ละตัวจะเคลื่อนที่ด้วยความเร็วที่มาจากน้ำหนักรวมขององค์ประกอบ 2 ตัวคือ ความเร็วเดิม เป็นองค์ประกอบของความเร่งที่ขับอนุภาคไปยังตำแหน่งที่ดีที่สุดเท่าที่ตัวอนุภาคนั้นได้พบมา และความเร็วใหม่ที่ขับให้อนุภาคไปในตำแหน่งที่ดีที่สุดที่เพื่อนข้างๆ กับตัวมันได้พบมา เมื่อตำแหน่งที่ดีกว่าถูกค้นพบ

ตำแหน่งเหล่านี้จะเป็นแนวทางในการเคลื่อนที่ของอนุภาคทั้งกลุ่ม กระบวนการนี้จะถูกทำซ้ำจนกระทั่งคำตอบที่เราพอใจถูกค้นพบ

ในระหว่างการเคลื่อนที่ของกลุ่มอนุภาคเพื่อหาอาหาร ขอบเขตของการหาอาหารเปรียบเสมือนปริภูมิปัญหา (Problem Space) ถ้าเรากำหนดให้อนุภาคแต่ละตัวสามารถจำตำแหน่งที่ดีที่สุดของตัวมันในการเคลื่อนไหวที่ผ่านมาได้ ค่าของตำแหน่งนี้ให้เรียกว่า pbest (particle best) สำหรับการพิจารณาในระหว่างอนุภาคที่อยู่ข้างเคียงกัน ตำแหน่งที่ดีที่สุดในการเคลื่อนไหวที่ผ่านมาของกลุ่มย่อยนี้เรียกว่า lbest (local best) สำหรับค่าที่ดีที่สุดของการเคลื่อนที่ที่ผ่านมาของทั้งกลุ่ม เราจะเรียกว่า gbest (global best)

แนวคิดของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคนี้ประกอบด้วย การเปลี่ยนความเร็วของอนุภาคแต่ละตัวจากตำแหน่ง pbest และ lbest ด้วยการสร้างค่าสุ่มสองค่าเพื่อมาปรับน้ำหนักของ pbest และ lbest ในทุกขั้นตอนของการค้นหา

6.2 การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค

ดังที่ได้กล่าวมาแล้วข้างต้นว่า การหาอาหารของนกฝูงหนึ่ง ที่กำหนดว่าในบริเวณนั้นมีอาหารอยู่เพียงแห่งเดียว นกทุกตัวไม่ทราบว่าแหล่งอาหารนั้นอยู่ที่ใด นกทุกตัวจะมีวิธีการของตนเองที่จะหาว่าแหล่งอาหารอยู่ที่ใด แล้วก็จะบินไปในทิศนั้น แต่อย่างไรก็ตาม การบินของนกก็ยังรักษาความเป็นฝูงไว้ โดยที่เมื่อนกทั้งฝูงหาแหล่งอาหารแต่ละครั้ง นกทุกตัวจะคำนวณหาระยะทางที่จะเดินทางของตน แล้วนำไปเปรียบเทียบกับนกตัวอื่นๆ จากนั้นนกทั้งฝูงจะเลือกบินตามนกตัวที่มีระยะทางที่ใกล้แหล่งอาหารมากที่สุด การทำงานในรอบแรกก็จะสิ้นสุด แล้วนกทุกตัวก็จะเริ่มคำนวณหาระยะทางใหม่ และเลือกบินไปในทางที่ใกล้แหล่งอาหารที่สุดอีกรอบ การบินนี้จะเข้าใกล้แหล่งอาหารไปเรื่อยๆ นกทั้งฝูงจะทำเช่นนี้ไปจนกระทั่งพบแหล่งอาหาร ในระหว่างการคำนวณของนกทุกตัว เพื่อการตัดสินใจแต่ละครั้ง จะเรียกเป็นรอบ (Iteration) หรือ รุ่น (Generation)

6.2.1 อัลกอริธึมของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค

การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค เป็นการจำลองการหาอาหารของฝูงนก ในการหาค่าเหมาะที่สุดแบบนี้ นกแต่ละตัวในฝูงจะถูกแทนด้วยอนุภาค อนุภาคแต่ละตัวจะมีค่าฟิตเน็สที่บอกถึงระยะห่างของตัวมันจากแหล่งอาหาร ซึ่งหามาจากฟิตเน็สฟังก์ชันของการหาค่าเหมาะที่สุด และมีความเร็วซึ่งกำกับการบินของอนุภาคนั้น อนุภาคทั้งหลายจะบินผ่านปริภูมิปัญหาโดยการบินตามอนุภาคที่มีค่าฟิตเน็สที่ดีที่สุดในแต่ละช่วง

การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค จะเริ่มต้นด้วยการสุ่มหาตำแหน่งของอนุภาค (หรือคำตอบ) ขึ้นมาชุดหนึ่ง จากนั้นก็จะหาค่าเหมาะที่สุดด้วยการปรับปรุงค่าในแต่ละรอบของการตัดสินใจ โดยที่อนุภาคแต่ละตัวจะมีการปรับปรุงค่าด้วยการเปลี่ยนตำแหน่งตามค่าที่ ดีที่สุด (Best) 2 ค่า ค่าแรกคือค่าที่ดีที่สุดของการคำนวณที่ผ่านมาของอนุภาคนั้น ค่านี้จะเรียกว่า ***ค่าที่ดีที่สุดของอนุภาค หรือ pbest*** ค่าที่ดีที่สุดอีกค่าหนึ่งคือค่าที่ดีที่สุดของทั้งกลุ่มที่ผ่านมา ค่านี้จะเป็นค่าที่ดีที่สุดของทั้งหมดเท่าที่เดินทางมา และเรียกว่า ***ค่าที่ดีที่สุดของสากล หรือ gbest*** เมื่ออนุภาคที่เป็นส่วนหนึ่งของประชากรที่อยู่รอบข้างมัน เกิดค่าที่ดีที่สุดในระหว่างประชากรนั้น จะเรียกว่า ***ค่าที่ดีที่สุดของท้องถิ่น หรือ lbest*** เมื่อมีการหาค่าที่ดีที่สุด 2 ค่าคือ pbest และ gbest แล้วการปรับปรุงความเร็วและตำแหน่งก็จะเกิดขึ้นดังสมการดังต่อไปนี้

□ = □ + □$_1$□□□□()(□□□□□[] − □□□□□□□[]) + □$_2$□□□□()(□□□□□[] − □□□□□□□[]) สมการ 6.1

□□□□□□□[] = □□□□□□□[] + □[] สมการ 6.2

เมื่อ

v[] เป็นความเร็วของอนุภาค

present[] เป็นค่าที่อนุภาคคำนวณได้ในปัจจุบัน

rand() เป็นค่าสุ่มที่อยู่ระหว่าง [0,1]

สำหรับค่า pbest[] และ gbest[] เป็นค่าตามที่ได้นิยามไว้ข้างต้น เมื่อเขียนกระบวนการทำงานของอัลกอริธึมนี้เป็นรหัสเทียม (Pseudo code) จะได้ดังนี้

For อนุภาคแต่ละตัว
Do
กำหนดค่าเริ่มต้นชุดหนึ่งให้กับอนุภาคทุกตัว
End Do
While not (จำนวนรอบสูงสุด หรือบรรลุวัตถุประสงค์ของการค้นหา)
For อนุภาคแต่ละตัว
Do
คำนวณค่า ฟิตเน็ส
If ค่า ฟิตเน็ส นี้ดีกว่าค่า pbest Then ให้ค่า ฟิตเน็ส นี้แทนที่ค่า pbest
End Do
เลือกอนุภาคที่มีค่า ฟิตเน็ส ดีสุดของทั้งกลุ่มมาเปรียบเทียบกับ gbest
If ค่าที่ได้ดีกว่า gbest Then ให้ค่าที่ได้เป็นค่า gbest ใหม่
For อนุภาคแต่ละตัว
Do
คำนวณค่าความเร็วใหม่ของอนุภาคตามสมการ
v[] = v[] + $c_1$rand()(pbest[]-present[]) + c_2*rand()*(gbest[]-present[])*
ปรับปรุงค่าตำแหน่งใหม่ของอนุภาคตามสมการ
present[] = present[] + v[]
End Do
End while

วิธีการของการหาค่าเหมาะที่สุดแบบฝูงอนุภาคก็คือ การใช้หลักการเปรียบเทียบคำตอบ เพื่อหาคำตอบที่ดีที่สุดทุกครั้งที่มีการเปลี่ยนตำแหน่ง ตำแหน่งใหม่จะถูกปรับทุกครั้งตามความเร็ว v[] ที่เกิดขึ้น โดยที่การปรับตำแหน่งทุกครั้งจะเกิดคำตอบใหม่ทุกครั้ง และคำตอบนี้จะเป็นตัวบอกว่าการเปลี่ยนตำแหน่งใหม่นั้นมุ่งเข้าหาเป้าหมายที่ต้องการได้ดีแค่ไหน

กลุ่มอนุภาคนั้นมีมากกว่าการรวมกันของอนุภาค อนุภาคโดยตัวของมันเองไม่มีพลังในการแก้ปัญหา การแก้ปัญหาเกิดเมื่อแต่ละอนุภาคประสานงานกัน มีการสื่อสารระหว่างกัน ในรูปแบบที่ดี

พอที่จะหาค่าที่ดีที่สุดในระหว่างกันได้ ซึ่งผลรวมของการทำงานร่วมกันทั้งหมดนี้ ทำให้เกิดนวัตกรรมของการแก้ปัญหาขึ้น ดังที่ได้แสดงไว้ในอัลกอริธึมข้างบน

6.2.2 การทำงานของการหาค่าเหมาะที่สุดแบบฝูงอนุภาค

ในการทำงานจริงของการใช้วิธีการทำงานร่วมกันของกลุ่มอนุภาค เพื่อการหาค่าเหมาะที่สุด มีกระบวนการที่สำคัญอยู่ 2 ขั้นตอนคือ การแทนปัญหา และการกำหนดฟิตเน็สฟังก์ชัน

เนื่องจากสมการความเร็ว v[] ตามสมการ 6.1 และสมการปรับความเร็ว present[] ตามสมการ 6.2 เป็นสมการที่บอกถึงตำแหน่งเชิงตัวเลข ทำให้การแก้ปัญหาแบบนี้เป็นแบบเชิงตัวเลข ไม่ใช่การแก้ปัญหาแบบคอมบินาทอเรียล ดังนั้นฟังก์ชันที่ใช้ในการหาตำแหน่งต่างๆ ในสมการความเร็ว v[] จึงเป็นฟิตเน็สฟังก์ชันได้ในตัว และตัวแปรต่างๆ ที่เกิดขึ้นในฟังก์ชันก็คือพารามิเตอร์ของการแก้ปัญหา ดังนั้นการแทนปัญหา และการกำหนดฟิตเน็สฟังก์ชัน จึงเป็นเรื่องเดียวกันกับ การออกแบบฟังก์ชัน เพื่อการหาตำแหน่งของปัญหา และค่าของพารามิเตอร์ในฟังก์ชัน ก็คือตำแหน่งของอนุภาคในปริภูมิปัญหา

ข้อได้เปรียบของการหาค่าเหมาะที่สุดแบบฝูงอนุภาคก็คือ การหาค่าจะใช้ตัวเลขในการคำนวณ ซึ่งต่างจากการหาค่าที่เหมาะที่สุดโดยใช้อัลกอริธึมพันธุการที่ต้องใช้การเปรียบเทียบแบบคอมบินาทอเรียล เช่น $f(x) = x_1^2 + x_2^2 + x_3^2$ พารามิเตอร์ของอนุภาคก็คือ (x_1, x_2, x_3) และค่าฟิตเน็สฟังก์ชันก็คือ $f(x)$ เพื่อที่จะหาค่าดีที่สุดของ $f(x)$ (ซึ่งเราทราบอยู่แล้วว่า x ทุกตัวจะต้องเท่ากับ 0) เราจะต้องหาค่าให้กับเวคเตอร์ของ x หรือ (x_1, x_2, x_3) เนื่องจากการหาคำตอบให้กับฟังก์ชันนี้ ตัวแปรหรือพารามิเตอร์ของฟังก์ชัน มีหลายพารามิเตอร์ ดังนั้นการหาค่าเหมาะที่สุด จึงเป็นการหาค่าของพารามิเตอร์ทุกตัวหรือทุกมิติในฟังก์ชัน ดังนั้นจำนวนมิติของอนุภาคก็คือ จำนวนตัวแปรที่ปรากฏในฟิตเน็สฟังกชันนั่นเอง หลังจากนั้น เราก็จะใช้กระบวนการมาตรฐานในการหาค่าเหมาะที่สุด การค้นหาคำตอบด้วยกระบวนการทำซ้ำและการเปรียบเทียบ และการกำหนดเงื่อนไขของการหยุดการค้นหา ไม่ว่าจะเป็นการกำหนดจำนวนรอบมากที่สุด หรือปล่อยให้การค้นหาดำเนินการ จนกระทั่งได้คำตอบที่พอใจ

การทำงานของ PSO เป็นกระบวนการที่ทำงานเป็นรอบ ในแต่ละรอบของการทำงาน ความเร็วของอนุภาคแต่ละตัวจะถูกปรับปรุงบนพื้นฐานของตัวแปรที่สำคัญ 3 ตัวคือ ความเร็วปัจจุบันของอนุภาคนั้น ข้อมูลที่อนุภาคนั้นมีอยู่ และข้อมูลรวมของอนุภาคทั้งกลุ่ม หลังจากนั้น อนุภาคแต่ละตัว

จะปรับตำแหน่งของมันโดยการใช้ความเร็วใหม่ที่คำนวณได้ เพื่อที่จะทำให้รูปแบบของสมการดูง่ายขึ้น ดังนั้นสมการ 6.1 และสมการ 6.2 จึงถูกเขียนใหม่ดังต่อไปนี้

$$\square(\square + 1) = \square(\square) + \square_1 * \square_1 * \big(\square(\square) - \square(\square)\big) + \square_2 * \square_2 * (\square(\square) - \square(\square))$$ สมการ 6.3

$$\square(\square + 1) = \square(\square) + \square(\square + 1)$$ สมการ 6.4

จากสมการ 6.3 และ 6.4 ค่าของ p(t) และ g(t) จะใช้แทน p_{best} และ g_{best} ในการหาตำแหน่งใหม่ ตัวแปร v(t+1) หมายถึงความเร็วที่เวลา t+1 ค่าของ v เป็นค่าความเร็วที่มีการแทนค่าเป็นเวคเตอร์ที่มีตัวแปรหลายตัวเช่น (1.55, -1.55) ค่าความเร็วใหม่จะขึ้นอยู่กับนิพจน์ 3 ตัวตัวแรกคือ v(t) เป็นความเร็วในปัจจุบันที่เวลา t ของอนุภาค ตัวที่สองเป็นนิพจน์ $c_1*r_1*(\mathbf{p}(t)–\mathbf{x}(t))$ เมื่อ c_1 เป็นค่าคงที่ที่ใช้เป็นค่าถ่วงน้ำหนักความจำ (Cognitive weight) ค่า r_1 เป็นค่าสุ่มที่มีค่าระหว่าง 0 และ 1 [0<r1<1] ค่า p(t) เป็นค่าเวคเตอร์ของตำแหน่งที่ดีที่สุดที่อนุภาคนั้นได้เดินทางมา และค่า x(t) เป็นค่าเวคเตอร์ที่แสดงตำแหน่งปัจจุบันของอนุภาค สำหรับนิพจน์ที่สาม $c_2*r_2*(\mathbf{g}(t)–\mathbf{x}(t))$ ค่าคงที่ c_2 เป็นค่าถ่วงน้ำหนักความจำของกลุ่มอนุภาค (Global weight) สำหรับค่าคงที่ c_1 และ c_2 บางครั้งถูกเรียกว่าแฟคเตอร์การเรียนรู้ ค่า r_2 เป็นค่าสุ่มที่มีค่าระหว่าง 0 และ 1 ค่า g(t) เป็นค่าเวคเตอร์ที่แสดงตำแหน่งที่ดีที่สุดเท่าที่หาได้ของกลุ่มอนุภาค เมื่อได้ค่าความเร็วใหม่ จากนั้นก็หาค่าตำแหน่ง x(t+1)

ถ้าหากว่าเรากำหนดให้ i เป็นจำนวนอนุภาคที่มีอยู่ในกลุ่ม และ k เป็นจำนวนพารามิเตอร์ที่ปรากฏในฟิตเน็สฟังก์ชัน หรือเป็นมิติของปัญหา เราสามารถเขียนสมการ 6.3 สมการ 6.4 ใหม่ได้เป็นดังนี้

$$\square_{\square\square}(\square + 1) = \square_{\square\square}(\square) + \square_1 * \square_1 * \big(\square(\square) - \square_{\square\square}(\square)\big) + \square_2 * \square_2 * (\square(\square) - \square_{\square\square}(\square))$$ สมการ 6.5

$$\square_{\square\square}(\square + 1) = \square_{\square\square}(\square) + \square_{\square\square}(\square + 1)$$ สมการ 6.6

6.2.3 พารามิเตอร์ของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค

เพื่อที่จะทำความเข้าใจมากขึ้นเกี่ยวกับ พารามิเตอร์ของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค เราจะนิยามองค์ประกอบพื้นฐานตามสมการ 6.5 และสมการ 6.6 ดังต่อไปนี้

1. อนุภาค x(t): คือคำตอบที่เป็นไปได้ทั้งหมดที่แสดงไว้เป็น k-dimension หรือมีมิติเท่ากับ k ในรูปของเวคเตอร์ของค่าจำนวนจริง เมื่อ k เป็นจำนวนของพารามิเตอร์ที่เราต้องการหาค่าเหมาะที่สุด ที่เวลา t อนุภาคที่ i หรือ $x_{ik}(t)$ สามารถเขียนออกมาได้เป็น $x_{ik}(t) = [x_{i,1}(t), x_{i,2}(t), x_{i,3}(t) \dots x_{i,k}(t)]$
2. จำนวนประชากร คือจำนวนอนุภาค i ที่ใช้ในการหาค่าเหมาะที่สุดของ $x_{ik}(t)$ เมื่อ i เท่ากับ n อนุภาค
3. ความเร็วของกลุ่มอนุภาค (Particle velocity) v(t) เป็นค่าของความความเร็วในการเคลื่อนที่ของกลุ่มอนุภาคที่มีจำนวนมิติเท่ากับ k ของเวคเตอร์จำนวนจริง ที่เวลา t ของอนุภาคตัวที่ i^{th} ของมิติที่ k ค่าความเร็ว $v_{ik}(t)$ สามารถเขียนออกมาได้เป็น $v_{ik}(t) = [v_{i,1}(t), v_{i,2}(t), v_{i,3}(t), \dots v_{i,k}(t),]$
4. ค่าที่ดีที่สุดของแต่ละอนุภาค (Individual best) p_{best} ในขณะที่อนุภาคเคลื่อนที่ไปในปริภูมิปัญหา อนุภาคแต่ละตัวจะทำการเปรียบที่ค่าฟิตเน็สที่ดีที่สุดของตัวมันเองอยู่ตลอดเวลา ค่าฟิตเน็สนี้ เป็นค่าที่สะท้อนถึงตำแหน่งของแต่ละอนุภาคห่างจากแหล่งอาหารที่มันจะเดินทางไป ค่านี้จะเขียนเป็น p_{best} และจะมีการปรับค่าอยู่ตลอดเวลาในระหว่างการหาคำตอบที่เหมาะที่สุด
5. ค่าที่ดีที่สุดของกลุ่มอนุภาค (Global best) g_{best} เป็นตำแหน่งที่ดีที่สุดของกลุ่มอนุภาค ที่คัดมาจากค่าที่ดีที่สุดของอนุภาคทุกตัว เท่าที่ผ่านมาของการค้นหาค่าที่เหมาะที่สุด

ในการหาค่าเหมาะที่สุดแบบฝูงอนุภาค มีพารามิเตอร์จำนวนมากที่จะต้องทำความเข้าใจและมีการปรับค่าที่เหมาะสมในการทำงาน พารามิเตอร์ต่างๆ เหล่านี้เช่น จำนวนอนุภาคที่อยู่ในกลุ่ม จำนวนมิติของอนุภาค เป็นต้น เพื่อเป็นตัวอย่างในการกำหนดค่าของพารามิเตอร์เหล่านี้ ในหน้าเว็บไซต์ที่เป็นทางการของการหาค่าเหมาะที่สุดแบบฝูงอนุภาค http://www.swarmintelligence.org/ ได้กำหนดค่ามาตรฐานพารามิเตอร์ไว้ดังต่อไปนี้

จำนวนอนุภาค (Number of Particles) โดยทั่วไปแล้วจำนวนอนุภาคจะกำหนดไว้ที่ 20 – 40 สำหรับปัญหาโดยทั่วไป จำนวนอนุภาค 10 ตัวก็มากพอที่จะทำให้ได้คำตอบที่ดีแล้ว สำหรับบางปัญหาที่ยากมากๆ จำนวนอนุภาคอาจจะมากถึง 100 – 200 ตัวก็ได้

มิติของอนุภาค (Particle Dimension) จะถูกกำหนดโดยลักษณะของปัญหาที่ต้องการหาค่าที่เหมาะที่สุด หรือฟิตเน็สฟังก์ชัน

ช่วงของอนุภาค (Particle Range) เป็นการกำหนดขอบเขตของปริภูมิปัญหา ก็จะถูกกำหนดโดยลักษณะของปัญหาที่ต้องการหาค่าที่เหมาะที่สุดเช่นกัน โดยที่เราสามารถกำหนดช่วงของอนุภาคที่ต่างกันสำหรับมิติของปัญหาที่ต่างกัน

V_{max} เป็นค่าที่กำหนดการเปลี่ยนแปลงสูงสุดที่เกิดขึ้นในระหว่างการค้นหาในแต่ละรอบ ปกติแล้ว ช่วงของอนุภาคจะกำหนดให้เท่ากับค่าของ V_{max} เช่น อนุภาค (x_1, x_2, x_3) เมื่อ x_1 มีค่าอยู่ระหว่าง [-10, 10] ดังนั้น V_{max} = 20

แฟคเตอร์การเรียนรู้ (Learning Factor) ค่า c_1 และ c_2 โดยทั่วไปจะเท่ากับ 2 อย่างไรก็ตาม ค่าอื่นๆ ก็มีการใช้เช่นกัน แต่จะมีช่วงระหว่าง [0, 4]

เงื่อนไขในการหยุด (Stop Condition) จะกำหนดจากจำนวนรอบมากที่สุดของการค้นหา

การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคนี้ สามารถแบ่งเป็นแบบโกลบอลเวอร์ชัน และ โลคอลเวอร์ชัน สำหรับโกลบอลเวอร์ชัน จะทำงานเร็ว แต่ก็มีปัญหาคือการลู่เข้าสู่คำตอบก่อนเวลาอันควร สำหรับโลคอลเวอร์ชัน การทำงานจะช้าแต่การลู่เข้าสู่คำตอบก่อนเวลาอันควรจะเกิดขึ้นน้อย เราสามารถใช้โกลบอลเวอร์ชัน เพื่อหาขอบเขตของคำตอบได้เร็วขึ้น และใช้โลคอลเวอร์ชัน เพื่อปรับแต่งการค้นหาให้ละเอียดขึ้น

6.2.4 ขั้นตอนการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค

สำหรับขั้นตอนการทำงานของอัลกอริธึม PSO เริ่มต้นด้วยการให้อนุภาคแต่ละตัวในกลุ่มเป็นตัวแทนคำตอบที่เป็นไปได้ แต่ละอนุภาคมีมิติเท่ากับ k ซึ่งจำนวนมิติของอนุภาคนี้เป็นจำนวน

พารามิเตอร์ทั้งหมดที่เราต้องการหาค่าเหมาะที่สุด ดังนั้นในการหาค่าที่เหมาะที่สุดของพารามิเตอร์แต่ละตัวจะแสดงถึงมิติในปริภูมิปัญหา อัลกอริธึมของการหาค่าเหมาะที่สุดมีรายละเอียดเป็นดังนี้

ขั้นตอนที่ 1: การกำหนดค่าเริ่มต้น

กำหนดให้ค่า t = 0 และสุ่มค่าของพารามิเตอร์ของอนุภาค $x_{ik}(t)$ เมื่อ k = 1 ... k ให้กับ $x_i(t)$ เมื่อ i = 1 ... n สำหรับค่าพารามิเตอร์แต่ละตัวจะมีการกำหนดค่าสูงสุดต่ำสุด $[x_{min}, x_{max}]$ สำหรับค่าที่สุ่มออกมากับอนุภาคแต่ละตัวนี้ให้หาค่าฟิตเน็ส f(x)

ขั้นตอนที่ 2: เพิ่มค่า t = t + 1

ขั้นตอนที่ 3: ปรับค่าความเร็ว

จากค่าที่ดีที่สุดของอนุภาคทั้งกลุ่ม g_{best} และค่าที่ดีที่สุดของแต่ละอนุภาค p_{best} ปรับค่าความเร็วของอนุภาคตัวที่ i ตามสมการ 6.7 ดังต่อไปนี้

$$□_{□□}(□+1) = □_{□□}(□) + 1 * □_1 * \big(□(□) - □_{□□}(□)\big) + 1 * □_2 * (□(□) - □_{□□}(□))$$ สมการ 6.7

จากสมการข้างต้น ค่า t เป็นจำนวนรอบของการทำงาน (Iteration) ค่า i เป็นตำแหน่งของอนุภาคหรืออนุภาคตัวที่ i และค่า k เป็นตำแหน่งของพารามิเตอร์ในแต่ละอนุภาค ในส่วนข้างขวาของสมการ ส่วนที่สองเป็นนิพจน์ที่แสดงถึงความจำของแต่ละอนุภาคที่เก็บค่าที่ดีที่สุด นิพจน์ที่สามเป็นส่วนที่แสดงความจำสำหรับเก็บค่าที่ดีที่สุดของทั้งกลุ่ม

ขั้นตอนที่ 4: การปรับค่าตำแหน่ง

จากการปรับค่าความเร็ว v(t+1) ในขั้นตอนที่ 3 ในขั้นตอนนี้ปรับ v(t+1) ของขั้นตอน 3 ให้เป็น v(t) ของอนุภาค ดังนั้นค่าใหม่ของแต่ละตัวจะปรับตำแหน่งของมันเองตามสมการดังนี้

$$□_{□□}(□+1) = □_{□□}(□) + □_{□□}(□)$$ สมการ 6.8

ขั้นตอนที่ 5: การปรับค่า p_{best}

สำหรับอนุภาคแต่ละตัว หาค่าฟิตเน็ส f และปรับค่า p_{best} ถ้าค่าที่ได้ใหม่ดีกว่าค่าเดิมที่มีอยู่

ขั้นตอนที่ 6: หาค่าที่ดีที่สุดของกลุ่มอนุภาค

สำหรับอนุภาคทั้งหมด ให้ปรับค่า g_{best} ถ้าค่าที่หาได้ใหม่ดีกว่าค่าเดิมที่มีอยู่

ขั้นตอนที่ 7: เงื่อนไขในการหยุด

ให้ยุติการค้นหา ถ้าเงื่อนไขในการหยุดเป็นไปตามที่กำหนดหรือ tmax

ในการหาค่าเหมาะที่สุดด้วยวิธีการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคนั้นมีการทำได้หลายวิธี สำหรับรายละเอียดเพิ่มเติมสามารถหาดูได้จาก (W. and Liu, Y. Zhang, 2004)

6.2.5 การลู่เข้าสู่คำตอบ

มีแนวความคิดที่หลากหลายในการพิจารณา เรื่องการลู่เข้าสู่คำตอบของการหาค่าเหมาะที่สุดแบบฝูงอนุภาค นักวิจัยโดยทั่วไปมีความเชื่อว่า พฤติกรรมของฝูงอนุภาคนี้จะแปรเปลี่ยนตามพฤติกรรมที่สำคัญ 2 อย่าง อย่างแรกเป็นพฤติกรรมในการสำรวจปริภูมิการค้นหา หรือเอ็กซ์พลอเรชัน ในการหาคำตอบที่เป็นไปได้ในปริภูมิปัญหาว่าจะมีขอบเขตกว้างหรือแคบขนาดไหน และอย่างที่สองคือพฤติกรรมของการหาคำตอบหรือเอ็กซ์พล้อยเตชัน ที่เป็นการค้นหาคำตอบที่ดีที่สุดจากคำตอบที่เป็นไปได้ทั้งหมด การทำให้อัลกอริธึมลู่เข้าสู่คำตอบที่ดีได้ จะต้องเป็นการสร้างสมดุลเอ็กซ์พลอเรชัน ที่เป็นการทำให้เกิดการสำรวจที่กว้างขวางเพื่อให้ได้คำตอบที่เป็นไปได้จำนวนมากสำหรับการคัดเลือก ซึ่งช่วยให้กระบวนการหาคำตอบมีโอกาสได้คำตอบที่ดี และช่วยป้องกันไม่ให้คำตอบที่ดีต้องหลุดไป แต่ผลเสียของการมีเอ็กซ์พลอเรชันก็คือ ขอบเขตของการหาคำตอบที่เป็นไปได้จะมีขนาดใหญ่ ทำให้ ต้องใช้เวลาในการหาคำตอบนาน นั่นคือระยะเวลาของการลู่เข้าสู่คำตอบ (Convergence) จะกินเวลานาน ถ้าหากว่าเราต้องการให้ได้คำตอบที่เร็วขึ้น ขอบเขตของการสำรวจก็จะต้องเล็กลง แต่คำตอบที่ดีก็อาจจะมีโอกาสหลุดจากการสำรวจไปได้ ซึ่งเราเรียกพฤติกรรมเช่นนี้ว่า “การลู่เข้าสู่คำตอบก่อนเวลาอันควร (Premature Convergence)” มีในบางกรณีเช่นกันที่ว่า เมื่อขยายขอบเขตของการสำรวจออกไปแล้ว แต่คำตอบที่ได้ออกมา กลับไม่ดีกว่าคำตอบที่เป็นอยู่ ดังนั้นการหาอัลกอริธึมที่ดีจึงเป็นการสร้างความสมดุลระหว่างเอ็กซ์พลอเรชัน และการลู่เข้าสู่คำตอบ

อีกแนวคิดหนึ่งคือ เป็นการพิจารณาพฤติกรรมของการหาค่าเหมาะที่สุดแบบฝูงอนุภาค ที่มีต่อสมรรถนะของการหาค่าเหมาะที่สุด โดยเฉพาะอย่างยิ่งในกรณีที่ปัญหาเป็นการค้นหาในปริภูมิที่มีหลายช่วง หรือเป็นฟิตเน็สฟังก์ชันที่เกิดจากการรวมกันของหลายฟังก์ชัน ปัญหาในลักษณะนี้ จะมีผลกระทบที่เกิดจากความไม่ต่อเนื่องของปัญหา ข้อมูลที่มีขยะมาก และการแปรเปลี่ยนของเวลา ผู้ที่มีความเชื่อในลักษณะนี้จะมุ่งเน้นการพัฒนาอัลกอริธึม และพารามิเตอร์ที่ทำให้สมรรถนะดีขึ้น โดยไม่สนใจว่าพฤติกรรมของกลุ่มจะมีผลอย่างไร ไม่ว่าจะเป็นเอ็กซ์พลอเรชัน หรือเอ็กซ์พล้อยเตชัน ซึ่งการวิจัยในกลุ่มนี้จะเน้นไปที่ การทำให้อัลกอริธึมของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคนี้มีความเรียบง่าย

เพื่อที่จะเป็นการให้นิยามของ การลู่เข้าสู่คำตอบ (Convergence) อย่างเป็นทางการ การลู่เข้าสู่คำตอบของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคนี้ จะมีความหมายอย่างใดอย่างหนึ่งในสองลักษณะนี้ คือ

- การลู่เข้าสู่คำตอบหมายถึงตำแหน่งที่ดีที่สุดของ g เท่าที่หามาได้ในการหาค่าที่เหมาะที่สุด โดยไม่ต้องสนใจว่ากลุ่มอนุภาคนั้นจะมีพฤติกรรมเช่นไร
- การลู่เข้าสู่คำตอบหมายถึงการที่กลุ่มอนุภาคทั้งกลุ่มลู่ไปสู่จุดใดจุดหนึ่งในปริภูมิปัญหา ซึ่งจุดนั้นอาจจะใช่หรือไม่ใช่จุดที่เหมาะที่สุดก็ได้

ที่ผ่านมา มีความพยายามหลายอย่างในการที่จะนำวิธีการทางคณิตศาสตร์มาใช้ในการวิเคราะห์การลู่เข้าสู่คำตอบ วิธีการเหล่านี้เช่น ของเคอร์และเคนเนดี้ (M. and Kennedy, J. Clerc, 2002) และทรีเลีย (Trelea, 2003) เป็นต้น การวิเคราะห์เหล่านี้ให้ผลที่เป็นแนวทางในการเลือกพารามิเตอร์ของการหาค่าที่เหมาะที่สุดแบบกลุ่มอนุภาค ที่ทำให้เชื่อว่าการทำงานจะลู่เข้าสู่คำตอบที่ดี และเช่นกันการวิเคราะห์นี้ยังส่งผลให้เกิดการพัฒนาของอัลกอริธึมใหม่ๆ ของการหาค่าเหมาะที่สุดแบบฝูงอนุภาค อย่างไรก็ตาม การวิเคราะห์ดังกล่าวข้างต้นก็มีการวิภาควิจารณ์โดย เพนเดอร์สัน (Pedersen, 2010) ว่าการวิเคราะห์นั้นเป็นการให้สมมุติฐานที่ไม่ครอบคลุม โดยที่กำหนดว่าอนุภาคนั้นมีเพียงอนุภาคเดียวและไม่ได้นำกระบวนการทางสถิติมาใช้ โดยที่วางจุดสนใจอยู่ที่ ตำแหน่งที่ดีที่สุดของ p และ g มีค่าคงที่ตลอดระยะเวลาของกระบวนการหาคำตอบ ยิ่งไปกว่านั้นในการวิเคราะห์ของบางคนยังกำหนดให้การค้นหามีการทำซ้ำในจำนวนรอบที่ไม่จำกัด ซึ่งเป็นไปไม่ได้ในความเป็นจริง นั่นก็หมายความว่า ในการ

วิเคราะห์หาความสามารถในการลู่เข้าสู่คำตอบของอัลกอริธึมแบบนี้ ยังต้องพิจารณาจากผลลัพธ์ที่ได้ออกมาจริงจากการทดลองมากกว่า

6.2.6 การทดสอบสมรรถนะ

ทรีเลีย (Trelea, 2003) และอีเบอร์ฮาร์ท (Eberhart, 2000) ได้ทำการทดสอบสมรรถนะของอัลกอริธึมของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคด้วยการใช้ฟังก์ชันเทียบเคียงมาตรฐานจำนวน 5 ฟังก์ชัน เพื่อหาค่าเหมาะที่สุด โดยการกำหนดเงื่อนไขของการทดลองดังแสดงไว้ในตารางที่ 6.1

ฟังก์ชัน f	มิติ (n)	ช่วง [xmin,xmax]	ค่าต่ำสุดของ f	ค่าเป้าหมายของ f
ทรงกลม	30	$[-100,100]^n$	0	0.01
โรเซนบร็อก	30	$[-30, 30]^n$	0	100
เรสตริจิน	30	$[-5.12, 5.12]^n$	0	100
กรีวังค์	30	$[-600, 600]^n$	0	0.1
เชฟเฟอร์	2	$[-100, 100]^2$	0	10^{-5}

ตารางที่ 6.1 ฟังก์ชันและเงื่อนไขสำหรับการทดสอบ

ในการทดลอง มีการกำหนดจำนวนรอบของการทำงานสูงสุดเท่ากับ 10,000 รอบ การทดสอบในแต่ละฟังก์ชันจะทำซ้ำกัน 20 ครั้ง ด้วยการสุ่มค่าของพารามิเตอร์เริ่มต้น x และ v ให้อยู่ช่วงระหว่าง [xmin,xmax] ตามตารางที่ 6.1 ในแต่ละครั้งของการทดสอบ สำหรับค่าคงที่ c_1 และ c_2 กำหนดให้เท่ากับ 0.729 และ 1.494 ตามลำดับ ซึ่งเป็นค่าที่แนะนำโดยเคอร์ (Clerc, 1999) จากตารางที่ 6.2 ผลของการทดสอบสมรรถนะ แถวนอนที่จำนวนอนุภาคถูกกำกับด้วย [E] เป็นผลการทดลองของอีเบอร์ฮาร์ท ที่เหลือเป็นผลการทดลองของ ทรีเลีย ในช่องของ อัตราความสำเร็จ หมายถึงตัวเลขสัดส่วน [0,1] ที่บอกอัตราความสำเร็จของการค้นหาค่าเหมาะที่สุด สำหรับจำนวนครั้งที่ฟังก์ชันถูกประเมินจะคำนวณจาก (จำนวนอนุภาคที่ใช้ในการทดลอง) x (จำนวนรอบของการทำงานเฉลี่ย) / (อัตราความสำเร็จ)

ค่าของพารามิเตอร์: $c_1 = 0.729$, $c_2 = 1.494$

ฟังก์ชัน	จำนวนอนุภาค *(N)*	จำนวนรอบของการทำงาน				อัตราความสำเร็จ	จำนวนครั้งที่ฟังก์ชันถูกประเมิน
		เฉลี่ย	มีเดียน	ต่ำสุด	สูงสุด		
ทรงกลม	15	764	731	586	1275	1	11460
	30	395	395	330	572	1	11850
	30[E]	530	525	495	573	1	15900
	60	314	313	269	368	1	18840
โรเซนบร็อก	15	1430	729	452	9476	1	21450
	30	900	408	298	4642	1	27000
	30[E]	669	621	402	1394	1	20070
	60	611	311	219	4450	1	36660
เรสตริจิน	15	299	292	152	523	0.80	5606
	30	182	174	123	299	0.95	5747
	30[E]	213	200	161	336	1	6390
	60	166	164	119	214	1	9960
กรีวังค์	15	755	608	470	1755	0.60	18875
	30	365	361	319	455	0.90	12167
	30[E]	313	308	282	366	1	**9390**
	60	287	280	266	328	1	17220
เชฟเฟอร์	15	1203	126	91	5853	0.40	45112
	30	350	157	102	1264	0.60	17500
	30[E]	532	321	94	2046	1	15960
	60	319	119	83	2361	0.95	20147

ตารางที่ 6.2 ผลของการทดสอบสมรรถนะ

ผลการทดลองแสดงให้เห็นได้ชัดเจนว่าเมื่อเพิ่มจำนวนอนุภาคให้มากขึ้น จำนวนรอบของการทำงานของอัลกอริธึมจะลดลง และเช่นกันอัตราความสำเร็จของอัลกอริธึมในการค้นพบคำตอบก็จะสูงขึ้นอย่างมีนัยสำคัญ ในขณะเดียวกัน จำนวนครั้งที่ฟังก์ชันถูกประเมินก็จะมีจำนวนมากขึ้นในแต่ละรอบของอัลกอริธึม ทั้งนี้เนื่องมาจากแต่ละรอบของการทำงาน ฟังก์ชันจะถูกเรียกใช้เพื่อการคำนวณให้กับอนุภาคแต่ละตัว สำหรับผลของการหาค่าเหมาะที่สุดของฟังก์ชันเทียบเคียงนั้น เนื่องจากฟังก์ชันรูปทรงกลม และฟังก์ชันของโรเซนบร็อก เป็นฟังก์ชันที่ไม่ซับซ้อน ดังนั้นอัตราความสำเร็จในการค้นพบคำตอบจะสูงกว่าฟังก์ชันที่มีความซับซ้อนมากๆ เช่นเชฟเฟอร์

6.3 ความหลากหลายของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค

การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคนี้ จะแตกต่างจากการหาค่าเหมาะที่สุดแบบที่ได้กล่าวมาแล้วก่อนหน้า คือการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคถูกออกแบบมาสำหรับการแก้ปัญหาเชิงตัวเลข ดังนั้นการนำวิธีการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคเพื่อไปแก้ปัญหาคอมบินาทอเรียลจะต้องมีการดัดแปลง ดังนั้นในความหลากหลายของการปรับปรุงการทำงานของอัลกอริธึมนี้จะแยกเป็น 2 เรื่องใหญ่ คือการปรับปรุงสมรรถนะ และการทำให้อัลกอริธึมสามารถแก้ปัญหาคอบินาทอเรียวได้

6.3.1 น้ำหนักอิเนอร์เทีย

การปรับปรุงการทำงานของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคนั้น ไช และอีเบอร์อาร์ด (Shi, 1998) ได้นำเสนอวิธีการปรับปรุงประสิทธิภาพของอัลกอริธึมด้วยการเพิ่มแฟคเตอร์น้ำหนัก (Weight Factor) เข้าไปที่สมการความเร็ว ที่เรียกว่าน้ำหนักอิเนอร์เทีย (Inertia Weight) ที่มีตัวแปรเป็น ω ซึ่งการเพิ่มนี้จะทำให้สมการของการปรับปรุงค่าความเร็วของอัลกอริธึมกลายเป็น

$$□_{□□}(□+1) = □ * □_{□□}(□) + □_1 * □_1 * \left(□(□) - □_{□□}(□)\right) + □_2 * □_2 * (□(□) - □_{□□}(□))$$

สมการ 6.9

น้ำหนักอิเนอร์เทียจะเป็นตัวควบคุมขอบเขตการค้นหาของอัลกอริธึม ซึ่งจะทำให้เกิดการเปลี่ยนแปลงของความเร็ว v ในสมการ ถ้าค่าของน้ำหนักอิเนอร์เทียมีค่าสูง ทำให้ความเร็ว v ของอนุภาคสูงขึ้น เป็นผลให้อัลกอริธึมมีการเอ็กซ์พลอเรชันในปริภูมิค้นหามีมากขึ้น ในขณะเดียวกันถ้าค่า

น้ำหนักอิเนอร์เทียมีค่าน้อย ทำให้ความเร็ว v ของอนุภาคช้าลง ส่งผลให้อัลกอริธึมมีเอ็กซ์พล้อยเตชันในปริภูมิค้นหามีมากขึ้น

6.3.2 สัมประสิทธิ์การหดตัว

เคอร์ และเคนเนดี (Clerc, 2002) ปรับปรุงการทำงานของอัลกอริธึมของการหาค่าเหมาะที่สุดด้วยการเพิ่มค่า สัมประสิทธิ์การหดตัว (Constriction Coefficient) เข้ามาในสมการการปรับปรุงความเร็ว ทำให้เกิดสมการใหม่เป็น

$$v_k^{t+1} = \chi\left(v_k^t + c_1 r_1\left(p_k^t - x_k^t\right) + c_2 r_2\left(p_g^t - x_k^t\right)\right) \qquad \text{สมการ 6.10}$$

โดยที่ค่าสัมประสิทธิ์การหดตัว (χ) คำนวณจากสมการดังต่อไปนี้

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \qquad \text{สมการ 6.11}$$

โดยที่กำหนดให้ค่า $\phi = c_1 + c_2 > 4$

ค่าสัมประสิทธิ์การหดตัวที่ เคอร์ และเคนเนดี นำเสนอขึ้นมานี้มีจุดประสงค์เพื่อป้องกันการแกว่ง (Oscilate) ของการค้นหาคำตอบที่เป็นผลเนื่องมาจากความเร็วที่มีค่าสูง ซึ่งจะกำหนดให้มีค่าเท่ากับระยะทางสูงสุดหรือ V_{max}= X_{max}

ในช่วงเวลาที่ไม่ต่างกันมากนัก เคนเนดีและเมนเดส (Kennedy, 2002) ได้ทำการปรับปรุงสมการความเร็วอีกครั้งกับสมการที่มีสัมประสิทธิ์การหดตัว โดยการเปลี่ยนแปลงวิธีการปรับปรุงค่าของค่าที่ดีที่สุดแบบโกลบอล และโลคอลใหม่ ทำให้สมการการปรับปรุงความเร็วใหม่กลายเป็น

$$v_k^{t+1} = \chi(v_k^t + \textstyle\sum_{p_i \in N_k} c_i r_i\left(p_i^t - x_k^t\right)) \qquad \text{สมการ 6.12}$$

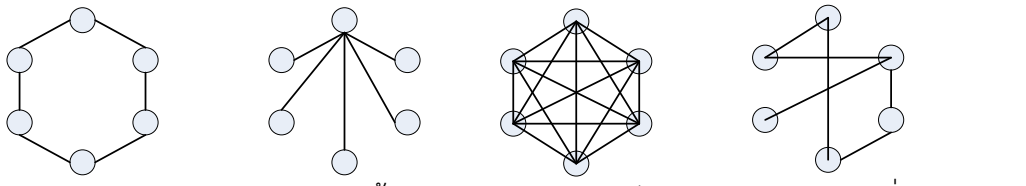
พวกเขาเรียกสมการใหม่นี้ว่า กลุ่มอนุภาคที่มีการรายงานอย่างสมบูรณ์ (Fully Informed Particle Swarm) ในสมการใหม่ค่าของ p_{best} และ g_{best} จะถูกแทนที่ด้วย p_{best} สำหรับค่าข้างเคียงทุกค่า ผลการทำงาน มีการอ้างอิงว่าทำให้ได้คำตอบที่ดีขึ้น ในขณะที่รอบของการทำงานน้อยลง แต่อย่างไรก็ตามการทำงานของสมการนี้จะต้องมีการเลือกโทโพโลยีของค่าข้างเคียงที่เหมาะสมด้วย

6.3.3 โทโพโลยีของค่าข้างเคียง

ในการนำเสนอรูปแบบการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคครั้งแรก การพิจารณาเพื่อการปรับปรุงค่าของความเร็ว เมื่อมีการนำค่าความเร็ว v ที่ได้เทียบเคียงกับค่าข้างเคียง ยังไม่มีการจำแนกรูปแบบของการเปรียบเทียบ แต่เป็นเพียงการหาค่าที่ดีที่สุดในหมู่ค่าข้างเคียงทั้งหมดเพื่อหา g_{best} ออกมาใช้ เคนเนดี (Kennedy, 1999) ได้ทำการวิเคราะห์และหารูปแบบของการเปรียบเทียบค่าเหล่านี้ และนำเสนอวิธีการออกมาเรียกว่า โทโพโลยีของค่าข้างเคียง (Neighboring Topology) และแทนที่จะใช้ค่าที่ดีที่สุดแบบโกลบอล g_{best} ดังที่เป็น เขาได้แทนค่านี้ด้วยค่าที่ดีที่สุดแบบโลคอล p_l หรือ p_l^t มาใช้แทนในสมการ ดังนี้

$$v_k^{t+1} = \chi(v_k^t + c_1 r_1(p_k^t - x_k^t) + c_2 r_2(p_l^t - x_k^t)) \qquad \text{สมการ 6.13}$$

เคนเนดี ได้ออกแบบรูปแบบของการเปรียบเทียบเป็น 4 แบบคือ แบบวงกลมที่การเปรียบเทียบค่า v จะเลือกเปรียบเทียบกับค่าอื่นอีก 2 ค่า แบบวงล้อ อนุภาคแต่ละตัวจะมีค่าข้างเคียงเป็นชุดเดียวกัน แบบสตาร์เป็นการเลือกเปรียบกับค่าข้างเคียงทั้งหมดซึ่งเป็นแบบ g_{best} เดิม และแบบสุ่ม ค่าที่นำมาเปรียบเทียบกันของแต่ละค่า v จะมีจำนวนไม่เท่ากัน ขึ้นอยู่กับการกำหนดจำนวนด้วยการสุ่ม จากผลการทดลองของเคนเนดี ยังไม่มีข้อสรุปว่าโทโพโลยีแบบใดดีที่สุด



ก. แบบวงกลม (Circle) ข. แบบวงล้อ (Wheels) ค. แบบสตาร์ (Star) ง. แบบสุ่ม (Randomly assigned edges)

รูปที่ 6.1 โทโพโลยีชนิดต่างๆ ของการเปรียบเทียบค่าข้างคียง

6.3.4 การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคแบบคอมบินาโทเรียว

ในการใช้อัลกอริธึม การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคสำหรับการแก้ปัญหาที่เป็นแบบคอมบินาโทเรียวนั้น มีผู้นำเสนออยู่หลายวิธี วิธีแรกถูกนำเสนอโดยเคนเนดี (Kennedy, 1997) ชื่อไบนารี พีเอสโอ (Binary PSO)

ไบนารี พีเอสโอ เป็นการแปลงค่าความเร็วอนุภาคที่เป็นเลขจำนวนจริงนั้นให้เป็นความน่าจะเป็นของอนุภาคว่า จะมีค่าเป็น 0 หรือ 1 โดยที่ ในแต่ละรอบของการทำงาน ความเร็วของอนุภาคจะถูกปรับปรุงเช่นเดียวกันกับอัลกอริธึมเดิม จากนั้นค่าความเร็วที่ได้ใหม่จะถูกทำให้เป็นค่าเธร์สโฮลด์โดยใช้สมการดังต่อไปนี้

$$s(v_{id}) = \frac{1}{1+\exp(-v_{id})} \qquad \text{สมการ 6.14}$$

เมื่อได้ค่า $s(v_{id})$ ค่านี้จะถูกนำไปเปรียบเทียบกับค่า r ที่ได้จากการสุ่ม ถ้าค่า r มีค่าต่ำกว่า $s(v_{id})$ ดังนั้นอนุภาคที่ i ที่ตำแหน่ง d จะมีค่าเป็น 1 มิฉะนั้นค่านี้จะเป็น 0 วิธีการนี้ทำให้การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค แก้ปัญหาฟังก์ชันที่เป็นไบนารีได้โดยไม่ต้องมีการปรับปรุง

การแก้ปัญหาคอมบินาทอเรียล ในการใช้อัลกอริธึมการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค เพื่อแก้ปัญหาแบบคอมบินาทอเรียลอีกวิธีหนึ่งคือ เทคนิคของการที่ใช้ตัวดำเนินการสลับ เป็นวิธีการที่ถูกนำเสนอโดยวัง (K., Huang, L., Zhou, C.-G., and Pang, W. Wang, 2003) เพื่อการแก้ปัญหาชนิดคอมบินาทอเรียลด้วยการ เปลี่ยนวิธีการแทนค่าปัญหา และวิธีการคำนวณของตัวดำเนินการ

การแทนค่าปัญหาของวังจะใช้สตริงเป็นฟังก์ชันของการบอกตำแหน่ง ในการดำเนินงานของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคทั่วไป ตำแหน่งของอนุภาคหนึ่งก็คือคำตอบที่เป็นไปได้หนึ่ง และการหาตำแหน่งของอนุภาคจะใช้ฟังก์ชันทางคณิตศาสตร์เป็นเครื่องมือ สำหรับวิธีนี้ การบอกตำแหน่งของอนุภาคจะใช้สตริงเป็นเครื่องมือแทน ซึ่งสตริงแต่ละชุดจะเป็นคำตอบที่เป็นไปได้คำตอบหนึ่งของการแก้ปัญหา ตัวอย่างของสตริงเหล่านี้เช่น [2 4 3 1 6 5] เป็นเส้นทางเดินที่ผ่านเมืองที่มีชื่อเป็นหมายเลข สำหรับตัวดำเนินการที่เป็น + และ – ในสมการความเร็ว และสมการการปรับความเร็ว มีวิธีการนิยามในลักษณะของตัวดำเนินการสลับดังนี้

ตัวดำเนินการสลับ (Swap Operator) ถ้ากำหนด S เป็นสตริง ดังนั้นนิยามของ S คือ S = (a_i), i = 1 ... n และให้ SO คือตัวดำเนินการสลับ ดังนั้น SO(i_1, i_2) หมายถึงการสลับตำแหน่งเมือง a_1 และ a_2 ในสตริง เช่น S = (2 4 3 1 6 5) เมื่อผ่านตัวดำเนินการสลับ SO(i_1, i_2) หรือเขียนเป็น SO(1,2) จะได้ S ใหม่เป็น (4 2 3 1 6 5) เราสามารถเขียน SO(1,2) แทนด้วย S24 ก็ได้

ลำดับของตัวดำเนินการสลับ (Swap Sequence) สำหรับตัวดำเนินการสลับ SO ที่มีการทำงานซ้อนกันหลายตัว จะเขียนเป็น SS ซึ่งหมายถึงลำดับของตัวดำเนินการสลับ เช่น SS = (SO_1, SO_2, SO_3, ... SO_n) หมายถึงลำดับของตัวดำเนินการสลับ ที่มีการจัดลำดับการทำงานตั้งแต่ SO_1, SO_2, SO_3, ... SO_n ตามลำดับ

ในกรณีที่มี SS หลายชุด กระทำต่อ S ชุดเดียวกัน ชุดของ SS เหล่านั้นสามารถนำมาเรียงต่อกันตามลำดับก่อนหลังของการกระทำที่มีต่อ S ได้ เช่น ถ้ามี SS1 ที่กระทำต่อ S ได้ผลลัพธ์ออกมา จากนั้น SS2 มาดำเนินการต่อกับผลลัพธ์นั้น ตัวดำเนินการ SS1 สามารถรวมกับ SS2 ได้เป็น SS' = SS1 $\oplus$ SS2

การสร้างลำดับของตัวดำเนินการสลับพื้นฐาน (Basic Swap Sequence หรือ BBS) ถ้า S1 และ S2 เป็นคำตอบที่เป็นไปได้ และมี SS ที่กระทำกับ S2 ได้ผลลัพธ์ออกมาเป็น S1 เราจะนิยามว่า SS = S1 – S2 หรือ S1 = S2 + SS ดังตัวอย่างต่อไปนี้

S1: (**1** 2 3 4 5)

S2: (2 3 **1** 5 4)

S1(1) = S2(3) = 1, ดังนั้นตัวดำเนินการสลับตัวแรกจะเป็น SO(1 ,3), S21 = S2 + SO(1,3) เราจะได้ผลลัพธ์ดังต่อไปนี้

S21: (1 3 2 *5* 4)

S1(2) = S21(3) = 2 ดังนั้นตัวดำเนินการสลับตัวที่สองได้แก่ SO(2,3) และ

S22 = S21 + SO(2,3) ได้ผลลัพธ์ออกมาเป็น

S22 = (1 2 3 **5** 4)

ตัวดำเนินการสลับตัวที่ 3 คือ SO(4,5) เราจะได้ S23=A สุดท้ายเราจะได้ลำดับของตัวดำเนินการสลับพื้นฐานเป็น

SS = S1 – S2
= (SO(1 3), SO(2 3), SO(4 5))

ถ้า pbest[] เป็นคำตอบที่เป็นไปได้ที่ดีที่สุดเท่าที่อนุภาคนี้หาได้ (S1 = (a_i), i = 1 ... n) และ present[] เป็นคำตอบที่ดีที่สุดที่อนุภาคนี้หาได้ในปัจจุบัน (S2 = (a_j), j = 1 ... n) ดังนั้น (pbest[]-present[]) ก็คือ SS และ (gbest[]-present[]) ก็คือ คำตอบที่ดีที่สุดของอนุภาคทั้งหมด - คำตอบที่ดีที่สุดที่อนุภาคนี้หาได้ในปัจจุบัน หรือ SS ของอนุภาคทั้งฝูง และสมการใหม่ถูกดัดแปลงเป็นดังนี้

□[] = □[]⊕□□□□()(□□□□□[] – □□□□□□□[])⊕□□□□()(□□□□□[] – □□□□□□□[])
สมการ 6.15

□□□□□□□[] = □□□□□□□[] + □[] สมการ 6.16

สำหรับเครื่องหมาย ⊕ และ – ของสมการนี้จะมีความหมายใหม่คือ เครื่องหมาย ⊕ หมายถึงการรวมกันของ SS ตั้งแต่ 2 ตัวขึ้นไป เครื่องหมาย – หมายถึงการหา SS ของคำตอบที่เป็นไปได้ 2 คำตอบ และเครื่องหมาย + หมายถึงการที่ตัวดำเนินการสลับกระทำต่อคำตอบที่เป็นไปได้ สำหรับรายละเอียดที่มากขึ้นให้ดูจากตัวอย่างการเดินทางของเซลส์แมนในหัวข้อถัดไป

แบบไฮบริด เป็นการนำเอาเมตาฮิวริสติกอื่นมาใช้ร่วมกับการหาค่าเหมาะสุดแบบกลุ่มอนุภาค จี (Q., Lan, H. and Ying, T. Qi Ji, 2010) ได้นำเอาอัลกอริธึมพันธุการมาใช้ร่วมกับการหาค่าเหมาะสุดแบบกลุ่มอนุภาค เช่นเดียวกันกับวิธีการของวัง ฉีได้ใช้สตริงมาเป็นตัวแทนสำหรับการแสดงตำแหน่งของอนุภาค ซึ่งก็คือคำตอบที่เป็นไปได้ และปรับรูปแบบของสมการใหม่เป็น

□[] = □[] + □$_1$□□□□()(□□□□□[]⊗□□□□□□□[]) +
□$_2$□□□□()(□□□□□[]⊗□□□□□□□[]) สมการ 6.17

□□□□□□□[] = □□□□□□□[]⊗□[]
สมการ 6.18

โดยที่ ⊗ หมายถึงการครอสโอเวอร์ และค่าพารามิเตอร์เช่น pbest หมายถึงค่าที่ดีที่สุดของอนุภาคแต่ละตัว present คือค่าของอนุภาคปัจจุบัน และ gbest หมายถึงค่าที่ดีที่สุดของกลุ่มอนุภาค ซึ่งพารามิเตอร์ทั้งหมดเหล่านี้จะมีค่าเป็นลำดับของการทำงาน เช่น ลำดับของเมืองในการเดินทาง ลำดับของการจัดเรียงของ ลำดับของตารางการทำงาน เป็นต้น

6.4 ตัวอย่างการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค

วิธีการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคถูกนำไปประยุกต์ใช้กับการแก้ปัญหา เป็นไปอย่างกว้างขวาง รายละเอียดสามารถหาอ่านเพิ่มเติมได้จาก โพลี (R. Poli, 2008) เพื่อที่จะเป็นตัวอย่างเพิ่มเติมสำหรับการแก้ปัญหาโดยใช้ PSO หรือการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค ในที่นี้จะนำเสนอการประยุกต์ใช้งานใน 4 เรื่องคือ การหาค่าเหมาะที่สุดของฟังก์ชัน f(x) ที่ค่าของ x มีค่าเป็นจำนวนจริง สำหรับเรื่องที่สอง และสาม เป็นการนำการหาค่าเหมาะที่สุดและกลุ่มอนุภาคเพื่อใช้สำหรับการแก้ปัญหาคอมบินาทอเรียลคือ การเดินทางของเซลส์แมน และการแก้ปัญหาการตัดสต็อกแบบมิติเดียว

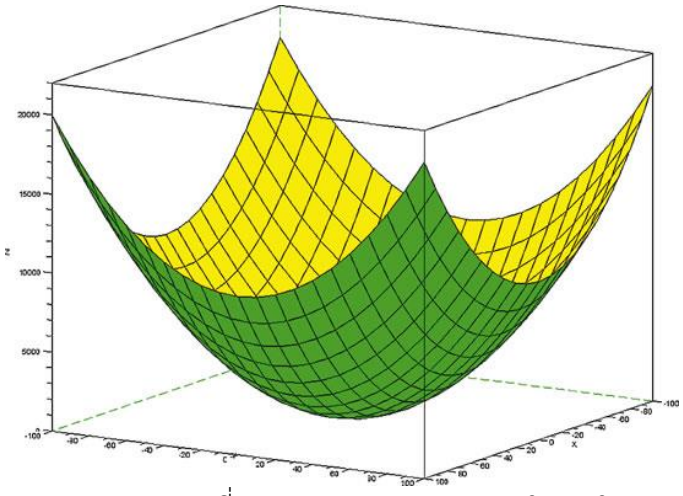
6.4.1 การหาค่าเหมาะที่สุดของฟังก์ชัน $f(x)=3+x_0^2+x_1^2$

เพื่อทำความเข้าใจการทำงานของ PSO ในตอนนี้จะทดลองหาค่าที่น้อยที่สุดของสมการ $f(x) = 3 + x_0^2 + x_1^2$ ซึ่งฟังก์ชันนี้สามารถสร้างเป็นกราฟได้ตามรูปที่ 6.2 ฐานของลูกบาศก์เป็นค่าของ x_0 และ x_1 และค่าในแกนตั้งคือค่าของฟังก์ชันของสมการ หรือ f(x) ดังนั้นที่จุดที่ผิวของกราฟมีค่าต่ำสุดเท่ากับ 3 เมื่อ $x_0 = x_1 = 0$ หรือ $f(x) = 3$

ถ้ากำหนดค่าเริ่มต้นของการทำงานดังต่อไปนี้ ให้ตำแหน่งปัจจุบันของอนุภาคอยู่ที่ **x**(t) มีเวคเตอร์ของตำแหน่งที่ $\{x_0, x_1\} = \{3.0, 4.0\}$ ค่าเวคเตอร์ความเร็วของอนุภาคเริ่มต้น **v**(t) เท่ากับ {-1.0, -1.5} กำหนดว่าค่า w = 0.7 ค่าคงที่ c1 = 1.4 ค่าคงที่ c2 = 1.4 และค่าสุ่ม r1 และ r2 เท่ากับ 0.5

และ 0.6 ตามลำดับ สุดท้ายกำหนดให้ค่าตำแหน่งที่ดีที่สุดของอนุภาคเท่าที่เดินทางมามีค่าของเวคเตอร์ **p**(t) = {2.5, 3.6} และค่าที่ดีที่สุดที่กลุ่มอนุภาคนั้นหาได้เท่าที่ผ่านมามีค่าของเวคเตอร์ **g**(t) = {2.3, 3.4} ดังนั้น ค่าความเร็วใหม่ และตำแหน่งใหม่ของอนุภาค แทนค่าทั้งหมดลงในสมการ

$$□_{□□}(□ + 1) = □ * □_{□□}(□) + □_1 * □_1 * \left(□(□) - □_{□□}(□)\right) + □_2 * □_2 * (□(□) - □_{□□}(□))$$



รูปที่ 6.2 กราฟของ $f = 3 + x_0^2 + x_1^2$

จะได้ค่าออกมาเป็น

v(t+1) = (0.7*{-1.0,-1.5}) + (1.4*0.5*({2.5,3.6} - {3.0,4.0})) + (1.4*0.6*({2.3,3.4} – {3.0,4.0}))

= {-0.70, -1.05} + {-0.35, -0.28} + {-0.59, -0.50}

= {-1.64, -1.83}

แทนค่าสมการปรับความเร็ว

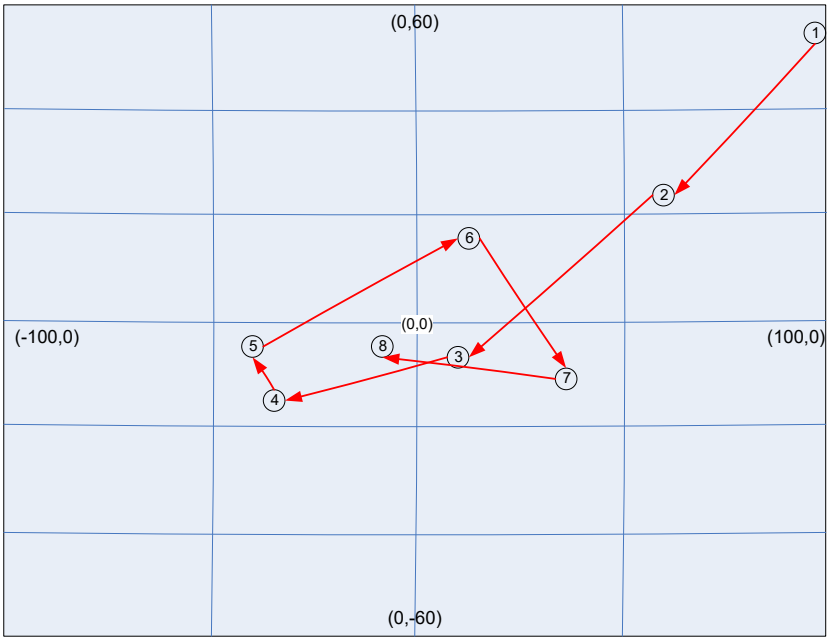
$$□_{□□}(□ + 1) = □_{□□}(□) + □_{□□}(□ + 1)$$

จะได้ค่าออกมาเป็น

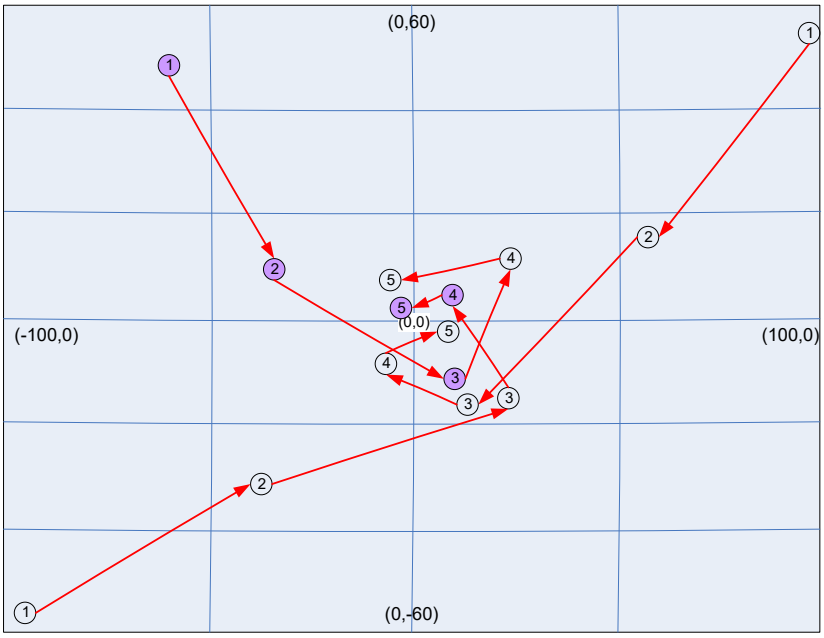
x(t+1) = {3.0, 4.0} + {-1.64, -1.83}

= {1.36, 2.17}

ตามที่เราทราบมาแล้วว่า ค่าที่เหมาะที่สุดของฟังก์ชันนี้คือ $\{x_0, x_1\}$ = (0.0, 0.0} จากผลของการคำนวณ สังเกตได้ว่าคำตอบที่ได้มีการปรับปรุงตำแหน่งของอนุภาคจาก (3.0, 4.0} เป็น {1.36, 2.17} ที่ทำให้เข้าใกล้ (0.0, 0.0) มากขึ้น ถ้าหากว่าเราทำการปรับปรุงค่าของอนุภาคนี้ โดยนำค่าตำแหน่งที่ได้ใหม่ไปปรับปรุง ด้วยการให้มีความจำที่เก็บตำแหน่งที่ดีที่สุดของกลุ่มอนุภาคไว้ (ในกรณีที่มีอนุภาคเพียงตัวเดียว ค่านี้จะไม่มี) และมีความจำที่เก็บค่าดีที่สุดของตัวอนุภาคเท่าที่ทำงานผ่านมาเอาไว้ทำเช่นนี้ไปเรื่อยๆ ตำแหน่งที่ได้ใหม่ก็จะค่อยๆ เข้าสู่จุดที่เหมาะที่สุด



รูปที่ 6.3 การเคลื่อนที่ของอนุภาคที่เข้าสู่ค่าที่เหมาะที่สุดของอนุภาคเพียงตัวเดียว (pbest)



รูปที่ 6.4 การเคลื่อนที่ของอนุภาค 3 ตัว (n = 3)

กราฟตามรูปที่ 6.3 แสดงการเคลื่อนที่ของตำแหน่งของอนุภาค เมื่อเราทำการปรับปรุงค่าของมันไปเรื่อยๆ ค่าของอนุภาคเริ่มต้นจาก x_0=100.0, x_1=80.4 และเคลื่อนที่ไปสู่ตำแหน่ง x_0=0, x_1=0 การเคลื่อนที่ของอนุภาคก่อนที่จะเข้าสู่คำตอบจะวนเป็นก้นหอยเข้าสู่ศูนย์กลาง ซึ่งเป็นเรื่องที่เกิดขึ้นโดยปกติของ PSO

ในกรณีที่มีอนุภาคจำนวน 3 ตัวทำงานร่วมกัน ค่าของ gbest จะถูกปรับปรุงให้เข้าใกล้จุดที่เหมาะที่สุดได้เร็วขึ้น เพราะ gbest จะเกิดจากการเลือกค่าที่ดีที่สุดของ pbest ทั้ง 3 อนุภาค จากรูปที่ 6.4 โหนดที่มีการแรเงา เป็นค่าของ gbest ที่เกิดจากอนุภาคทั้ง 3 ตัว และเมื่อจำนวนของอนุภาค n มากขึ้น จำนวนรอบของการทำงานเพื่อเข้าสู่จุดที่เหมาะที่สุดก็จะน้อยลงด้วย

6.4.2 การเดินทางของเซลส์แมน

ตัวอย่างของการหาค่าเหมาะที่สุดแบบฝูงอนุภาคสำหรับการแก้ปัญหาจริงนี้จะเป็นการนำการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคไปใช้ในการแก้ปัญหาการเดินทางของเซลส์แมน ที่นำเสนอโดย กังปิง วัง (K., Huang, L., Zhou, C.-G., and Pang, W. Wang, 2003) มีรายละเอียดดังต่อไปนี้

การแทนปัญหา จะเป็นการแสดงลำดับของเมืองที่เซลส์แมนจะต้องเดินทางในรูปของสตริง ในที่นี้เรากำหนดให้อยู่ในรูปของการเรียงกันตามลำดับของเมืองในวงเล็บ ดังนี้

$(a_1, a_2, a_3, \ldots a_n)$

เมื่อ a_i หมายถึงชื่อเมือง ที่มีทั้งหมด n เมือง และการเดินทางจะเริ่มจากเมือง a_1, a_2 ไปจนถึง a_n ซึ่งเป็นเมืองสุดท้าย หรืออาจจะเขียนเป็น

$S = (a_i), i = 1 \ldots n$

ตัวดำเนินการ เป็นวิธีการในการจัดลำดับเมืองเพื่อให้การเดินทางของเซลส์แมนได้ระยะทางสั้นที่สุด ตัวดำเนินการนี้จะแบ่งออกเป็น

ตัวดำเนินการสลับ (Swap Operator) ถ้ากำหนดว่า เมืองต่างๆ ที่เซลส์แมนจะต้องเดินทาง คือ $S = (a_i), i = 1 \ldots n$ และให้ SO คือตัวดำเนินการสลับ ดังนั้น $SO(i_1, i_2)$ หมายถึงการสลับตำแหน่งของเมือง a_{i1} และ a_{i2} ในลำดับของ S ถ้าให้ S’ หมายถึงคำตอบใหม่ที่เกิดจาก $SO(i_1, i_2)$ ดังนั้น

$$S' = S + SO(i_1, i_2) \qquad \text{สมการ 6.19}$$

ตัวอย่างเช่น S = (1, 3, 5, 2, 4) และ SO(1, 2) ดังนั้น

S’ = S + SO(1, 2) = (1, 3, 5, 2, 4) + (1, 2) = (3, 1, 5, 2, 4)

ลำดับของตัวดำเนินการสลับ (Swap Sequence) สำหรับตัวดำเนินการสลับ SO เมื่อมีหลายตัว และมีการดำเนินการกับ S หลายครั้ง เราจะเขียนเป็น SS ซึ่งหมายถึงลำดับของตัวดำเนินการสลับ เช่น

$SS = (SO_1, SO_2, SO_3, \ldots SO_n)$

หมายถึงลำดับของตัวดำเนินการสลับจะเป็นลำดับตั้งแต่ SO_1, SO_2, SO_3, ... SO_n และลำดับนี้จะมีความสำคัญต่อผลลัพธ์ที่จะเกิดขึ้นด้วย ถ้า S' คือคำตอบที่เกิดขึ้นใหม่ดังนั้น

$S' = S + SS = S + (SO_1, SO_2, SO_3, \ldots SO_n)$ สมการ 6.20

$= ((((S + SO_1) + SO_2) + SO_3) + \ldots SO_n)$

ซึ่งหมายถึง เมื่อตัวดำเนินการสลับ SO_1 กระทำกับ S ผลลัพธ์ที่ได้จะถูกสลับด้วย SO_2 และผลลัพธ์ที่ได้จะถูกดำเนินการต่อด้วย SO_3 เช่นนี้เป็นลำดับไปเรื่อยๆ จนกว่าจะครบ

ในบางกรณีที่ชุดของลำดับของตัวดำเนินการสลับ SS ที่ต่างกัน อาจจะให้คำตอบที่เหมือนกันได้ ในกรณีเช่นนี้ชุดลำดับของตัวดำเนินการสลับที่มีจำนวนตัวดำเนินการสลับ SO ที่น้อยที่สุดจะเรียกว่าลำดับของตัวดำเนินการสลับพื้นฐาน (Basic Swap Sequence หรือ BBS)

ในกรณีที่มี SS หลายชุด กระทำต่อ S ชุดเดียวกัน ชุดของ SS เหล่านั้นสามารถนำมาเรียงต่อกันตามลำดับก่อนหลังของการกระทำที่มีต่อ S ได้ เช่น ถ้ามี SS1 ที่กระทำต่อ S ได้ผลลัพธ์ออกมาแล้วมี SS2 มาดำเนินการต่อจากผลลัพธ์นั้น ดังนั้น SS1 สามารถรวมกับ SS2 ได้ และสามารถเขียนใหม่ได้เป็นดังนี้

$SS' = SS1 \oplus SS2$ สมการ 6.21

การสร้างลำดับของตัวดำเนินการสลับพื้นฐาน (Basic Swap Sequence หรือ BBS) สมมุติว่าเรามีคำตอบที่เป็นไปได้อยู่ 2 คำตอบ S1 และ S2 เราต้องการที่จะหาคำตอบที่เป็นลำดับของตัวดำเนินการสลับพื้นฐาน SS ที่สามารถกระทำกับ S2 เพื่อให้ได้ S1 ซึ่งเราจะนิยามว่า $SS = S1 - S2$ เราสามารถสลับเมืองใน S2 ให้เป็นไปตาม S1 จากซ้ายไปขวาเพื่อที่จะให้ได้คำตอบออกมาเป็น SS ดังนั้นสมการที่ได้ควรจะเป็น $S1 = S2 + SS$ (เครื่องหมาย + สอดคล้องกันโดยบังเอิญ) ดังตัวอย่างต่อไปนี้

S1: (**1** 2 3 4 5)

S2: (2 3 **1** 5 4)

S1(1) = S2(3) = 1, ดังนั้นตัวดำเนินการสลับตัวแรกจะเป็น SO(1 ,3), S21 = S2 + SO(1,3) เราจะได้ผลลัพธ์ดังต่อไปนี้

S21: (1 3 2 *5* 4)

S1(2) = S21(3) = 2 ดังนั้นตัวดำเนินการสลับตัวที่สองได้แก่ SO(2,3) และ

S22 = S21 + SO(2,3) ได้ผลลัพธ์ออกมาเป็น

S22 = (1 2 3 **5** 4)

ตัวดำเนินการสลับตัวที่ 3 คือ SO(4,5) เราจะได้ S23=A สุดท้ายเราจะได้ลำดับของตัวดำเนินการสลับพื้นฐานเป็น

SS = S1 – S2

= (SO(1 3), SO(2 3), SO(4 5)) สมการ 6.22

การประยุกต์ใช้งาน เมื่อนำวิธีการของตัวดำเนินการสลับ SO และลำดับของตัวดำเนินการสลับ SS และลำดับของตัวดำเนินการสลับพื้นฐาน BSS มาประยุกต์ใช้กับการหาค่าเหมาะที่สุดแบบฝูงอนุภาค จะได้สมการใหม่ดังนี้

ถ้า pbest[] เป็นคำตอบที่เป็นไปได้ที่ดีที่สุดเท่าที่อนุภาคนี้หาได้ (S1 = (a_i), i = 1 ... n) และ present[] เป็นคำตอบที่ดีที่สุดที่อนุภาคนี้หาได้ในปัจจุบัน (S2 = (a_j), j = 1 ... n) ดังนั้น (pbest[]-present[]) ก็คือ SS ตามที่อธิบายไว้ในหัวข้อ 4.2.3 และเช่นกัน (gbest[]-present[]) ก็คือ คำตอบที่ดีที่สุดของอนุภาคทั้งหมด - คำตอบที่ดีที่สุดที่อนุภาคนี้หาได้ในปัจจุบัน หรือ SS ของอนุภาคทั้งฝูง และสมการใหม่ถูกดัดแปลงโดยการนำค่า c ออกจะได้ดังนี้

v[] = v[]$\oplus$ rand()*(pbest[]-present[])$\oplus$ rand()*(gbest[]-present[]) สมการ 6.23

present[] = present[] + v[] สมการ 6.24

ค่า rand() เป็นค่าที่บอกตำแหน่งของเมืองที่จะทำการแลกเปลี่ยนหรือ $\oplus$ กันถ้าค่านี้เป็น 1 หมายความว่าไม่มีการระบุตำแหน่ง

สำหรับเครื่องหมาย $\oplus$ และ – ของสมการนี้จะมีความหมายใหม่ดังต่อไปนี้ เครื่องหมาย $\oplus$ หมายถึงการรวมกันของ SS ตั้งแต่ 2 ตัวขึ้นไปตามสมการ 6.21 เครื่องหมาย – หมายถึงการหา SS ของคำตอบที่เป็นไปได้ 2 คำตอบตามที่แสดงไว้ในสมการ 6.22 และเครื่องหมาย + หมายถึงการที่ตัวดำเนินการสลับกระทำต่อคำตอบที่เป็นไปได้ดังสมการ 6.19 และ 6.20

อัลกอริธึมการแก้ปัญหาการเดินทางของเซลส์แมน

1. การกำหนดค่าเริ่มต้น: สำหรับอนุภาคแต่ละตัวให้สุ่มคำตอบที่เป็นไปได้ และสุ่มค่าของลำดับการสลับ SS ให้กับค่าความเร่ง v[]
2. หากสิ้นสุดการค้นหาให้ไปที่ขั้นตอน 5
3. สำหรับอนุภาคแต่ละตัวที่ตำแหน่ง present[] ให้หาค่า present[] ตัวใหม่ ตามสมการ 6.24 ดังนี้
    3.1. หาค่าลำดับการสลับพื้นฐาน SS ของ S1 = (pbest[] - present[]) ตามสมการ 6.22 ดังที่กล่าวมาข้างต้น
    3.2. หาค่าลำดับการสลับพื้นฐาน SS ของ S2 = (gbest[] - present[]) ตามสมการ 6.22 ดังที่กล่าวมาข้างต้น
    3.3. หาค่า v[] ของสมการ 6.23 แล้วเปลี่ยน SS ของ v[] เป็น BSS
    3.4. หาคำตอบใหม่ตามสมการ 6.24 ซึ่งหมายถึง SS ของ v[] กระทำต่อคำตอบที่เป็นไปได้ present[] เพื่อที่จะได้ present[] ใหม่
    3.5. ปรับค่า pbest[] ถ้า pbest[] ใหม่ดีกว่า pbest[] ที่มีอยู่เดิม
4. ปรับค่า gbest[] ถ้าค่า gbest[] ใหม่ที่ดีกว่า gbest[] เดิม แล้วไปที่ 2
5. ร่างค่าของ gbset[] ออกมาเป็นคำตอบ

หมายเหตุ การที่เราจะทราบได้ว่า ค่า gbest[] หรือค่า pbest[] ใดดีกว่ากัน ให้ใช้ฟิตเน็สฟังก์ชันที่หาจากระยะทางรวมของเมืองทั้งหมดที่ถูกจัดลำดับไว้ในคำตอบที่เป็นไปได้ S = (a_1, a_2, a_3, ... a_n) เมื่อ a_i เป็นชื่อเมืองที่เรียงกันตามลำดับของการเดินทาง

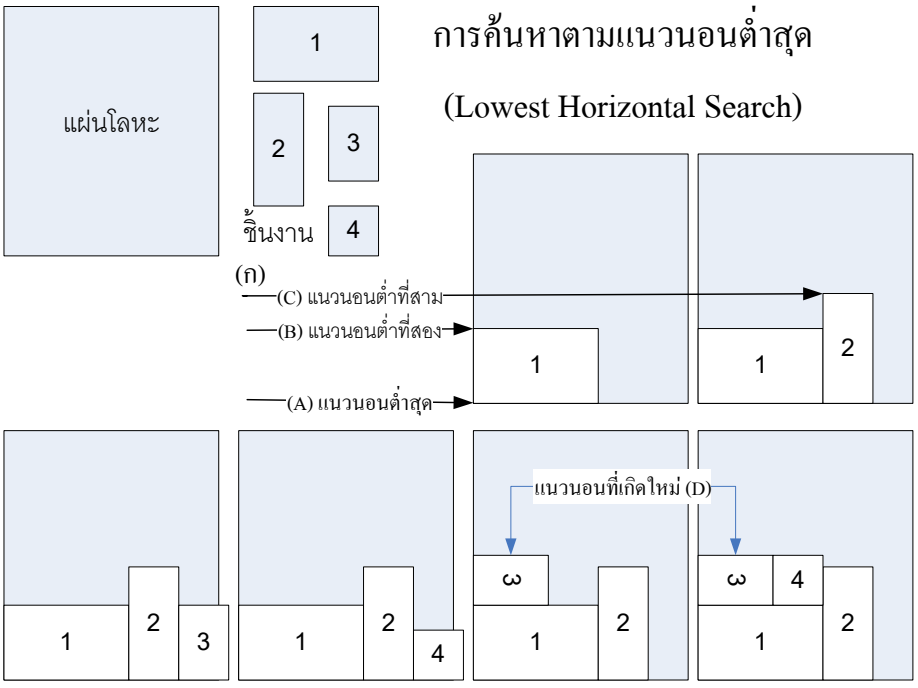
6.4.3 การตัดสต็อกแบบสองมิติ

การประยุกต์ใช้การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคสำหรับการแก้ปัญหาการตัดสต็อกแบบสองมิติ (Two Dimensional Stock Cutting) เป็นปัญหาที่มีความซับซ้อนมาก ปกติแล้ว ปัญหาของการตัดสต็อกสามารถแยกตามมิติได้หลายประเภท ตัวอย่างปัญหาของการตัดสต็อกแบบสองมิติคือการตัดผ้า ถ้าสมมุติว่าเรามีผ้าอยู่ 1 ผืน และเราต้องการตัดผ้าผืนนั้นเป็นรูปร่างต่างๆ เพื่อนำไปเย็บเป็นเสื้อ สิ่งที่เราต้องคำนึงก็คือ ทำอย่างไรที่จะตัดผ้าตามรูปร่างที่มีอยู่ทั้งหมด โดยที่ใช้พื้นที่ผ้าน้อยที่สุด และมีเศษผ้าน้อยที่สุดด้วย การตัดแบบสามมิติคือการจัดวางกล่องในตู้คอนเทนเนอร์ เป็นต้น

สำหรับในวงการอุตสาหกรรม ปัญหาการตัดสต็อกส่วนใหญ่จะเกี่ยวข้องกับการตัดแผ่นวัสดุตามรูปร่างและจำนวนที่เราต้องการจากแผ่นวัสดุที่มีขนาดใหญ่ สิ่งที่เราจะต้องคำนึงในเรื่องนี้คือ ทำอย่างไรที่เมื่อการตัดสิ้นสุดแล้ว เราจะเหลือเศษวัสดุที่ใช้งานไม่ได้จำนวนน้อยที่สุด เทคนิคในการตัดแผ่นวัสดุนี้จะมีความซับซ้อน ยากง่าย ขึ้นอยู่กับหลายปัจจัย โดยเฉพาะปัจจัยของรูปร่างของแผ่นวัสดุที่เราต้องการตัด ถ้าเราต้องการตัดวัสดุที่มีรูปร่างที่ไม่เหมือนกัน โดยเฉพาะวัสดุที่ต้องการตัดออกจากแผ่นใหญ่มีจำนวนด้านมาก รูปทรงแปลกๆ จำนวนหลายๆ ชิ้น การวางแผนการตัดเพื่อให้เหลือเศษวัสดุน้อยที่สุดก็จะทำได้ยาก ในบางอุตสาหกรรม การตัดแผ่นวัสดุจึงใช้วิธีการเลือกเอาวัสดุที่เราต้องการตัดที่มีรูปร่างเหมือนกันไปตัดในแผ่นโลหะแผ่นใหญ่เดียวกัน หลายกรณี แผ่นโลหะที่จะตัดมักจะเป็นแผ่นสี่เหลี่ยมที่มีขนาดต่างๆ กัน สำหรับวิธีการที่ใช้ในการตัดสต็อก จะเริ่มต้นด้วยการวางแผนการตัด โดยการนำรูปร่างของแผ่นโลหะที่เราต้องการตัด มาจัดวางบนแผ่นโลหะผืนใหญ่ก่อนที่จะทำการตัดจริง วิธีการนี้จะเริ่มด้วยการเลือกแผ่นโลหะที่เราต้องการตัดมาวางลงบนแผ่นโลหะผืนใหญ่ทีละแผ่นจนครบ เพื่อเป็นตัวอย่างของการแก้ปัญหาเรื่องนี้ด้วยวิธีการของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค เราจะจำกัดขอบเขตของการแก้ปัญหาให้เล็กลง โดยกำหนดให้การแก้ปัญหาการตัดสต็อกเป็นแบบสองมิติ และรูปร่างของแผ่นวัสดุที่เราต้องการตัดเป็นรูปสี่เหลี่ยม ที่นำเสนอโดยจี เลน และยิง (Qi Ji, 2010) ซึ่งมีรายละเอียดดังต่อไปนี้

อัลกอริธึมการค้นหาตามแนวนอนต่ำสุด (Lowest Horizontal Search Algorithm) อัลกอริธึมนี้จะเป็นวิธีการจัดวางชิ้นงานบนแผ่นโลหะก่อนการตัด ตามรูปที่ 6.5 เรามีโลหะที่จะถูกตัดเป็นแผ่นใหญ่ และชิ้นงานที่เราต้องการตัดออกจากแผ่นโลหะเป็นรูปสี่เหลี่ยมมี 4 ขนาด ซึ่งกำกับไว้ด้วยตัวเลขจาก 1 ถึง 4 ตามรูปที่ 6.5(ก)

การจัดวางชิ้นงานลงบนแผ่นโลหะจะเริ่มต้นด้วยการหาเส้นแนวนอนที่ต่ำสุดก่อน และเลือกชิ้นงานมาวางลงบนแนวนอนนั้น โดยพิจารณาว่า ชิ้นงานที่เลือกเข้ามาจะเอาด้านกว้าง หรือด้านนอนทาบกับแนวนอนนั้น ให้พิจารณาจาก ด้านไหนที่มีขนาดใกล้เคียงกับส่วนที่เหลือของแนวนอนมากที่สุด ถ้าหากว่าส่วนที่เหลือของแนวนอน กว้างไม่พอที่จะวางชิ้นงานใหม่ ให้ย้ายไปวางที่แนวนอนต่ำที่อยู่สูงขึ้นไป และทำเช่นเดิมไปเรื่อยๆ จนเลื่อนแนวนอนต่ำสุดขึ้นไปเรื่อยๆ จนทุกชิ้นงานถูกวางครบถ้วน



(ข) การเลือกวางชิ้นงานตามแนวนอนต่ำสุดก่อน

รูปที่ 6.5 วิธีการวางชิ้นงานบนแผ่นโลหะด้วยวิธีการค้นหาตามแนวนอนต่ำสุดก่อน

ตัวอย่างตามรูปที่ 6.5(ข) การพิจารณาแนวนอนต่ำสุดให้ดูจากพื้นที่ของแผ่นโลหะ เมื่อยังไม่มีชิ้นงานวางอยู่ แนวนอนต่ำสุดจะอยู่ที่ (A) ตามรูปที่ 6.5(ข) เราเลือกที่จะวางรูปที่ [1] ก่อน เมื่อวางแล้วจะเกิด แนวนอนอยู่ 2 แนวคือ A และ B จากนั้นนำชิ้นงาน [2] มาวางบนแนวนอนต่ำสุด ในที่นี้ก็คือ A เช่นเดิม เมื่อวางแล้วจะเกิดแนวนอนที่ 3 ที่ (C) ในการวางชิ้นงาน 2 เราสามารถวางนอน หรือวางตั้งก็ได้ ขึ้นอยู่กับความยาวของ แนวนอนต่ำสุดที่เหลือ ว่าด้านใดจะเหมาะสำหรับชิ้นงาน 2 ที่วางตั้งเพราะถ้าวางนอนแล้ว ความยาวจะเกินพื้นที่ของแผ่นโลหะ และเช่นกัน เมื่อวางชิ้นงาน 3 และ 4 บนแนวนอนต่ำสุด A แล้ว วางไม่ได้ จึงต้องเลื่อนขึ้นไปวางที่แนวนอนต่ำสุดระดับถัดขึ้นไปคือ B และก็จะเกิดแนวนอนระดับถัดไปอีกระดับเป็นระดับแนวนอน D

ในการเลือกชิ้นงานที่จะวางว่าจะวางชิ้นใดก่อนหลังนั้น เป็นการเลือกแบบเสรี คือจะเลือกวางชิ้นงานใดก่อนหรือหลังก็ได้ ไม่มีข้อจำกัดใดๆ สำหรับอัลกอริธึมของการค้นหาตามแนวนอนต่ำสุดเป็นดังนี้

ขั้นตอน 1: เลือกระดับแนวนอนต่ำสุดของพื้นที่ที่จะถูกตัด (หรือ Stock) เป็นค่าเริ่มต้น

ขั้นตอน 2: เลือกแนวนอนต่ำสุด จากระดับของแนวนอนที่มีอยู่ทั้งหมด (ตามรูปที่ 6.5(ข) คือ A B C หรือ D ถ้ามีแนวนอนต่ำสุดมากกว่า 1 ระดับ ให้เลือกระดับที่อยู่ล่างสุดก่อน จากนั้นให้พิจารณาว่าความกว้างของแนวนอนนั้นสามารถวางชิ้นงานได้หรือไม่

ถ้าได้ ให้จัดวางชิ้นงานลงบนแผ่นโลหะนั้น และทำการปรับปรุงข้อมูลของระดับของแนวนอนที่เกิดขึ้นใหม่ หรือที่หายไป (ถ้ามี)

ถ้าวางชิ้นงานไม่ได้ ให้หาชิ้นงานใหม่ แล้วเลือกด้านที่สามารถวางลงบนความกว้างที่เหลือของระดับแนวนอนต่ำสุด ถ้าหาชิ้นงานใหม่ที่จะวางลงบนแนวนอนต่ำสุดไม่ได้ ให้เลือกแนวนอนระดับต่ำสุดถัดมาเป็นระดับต่ำสุด และในขณะเดียวกันให้ปรับข้อมูลของแนวนอนระดับต่างๆ ใหม่

ขั้นตอน 3: ทำซ้ำตามขั้นตอนที่ 2 จนกระทั่งชิ้นงานที่กำลังวางอยู่นี้สามารถวางลงบนแผ่นโลหะหรือสต็อกนั้นได้

ขั้นตอน 4: ทำซ้ำตามขั้นตอนข้างต้นจนชิ้นงานทุกชิ้นถูกวางจนครบ

การปรับปรุงการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคสำหรับการแก้ปัญหาการตัดสต็อก ในการประยุกต์ใช้การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค หรือ PSO กับการแก้ปัญหาการตัดสต็อกนั้น เรื่องของการปรับปรุงค่าของอนุภาคนั้นทำได้ยาก ดังนั้น ในการแก้ปัญหานี้จึงนำวิธีการครอสโอเวอร์ และมิวเตชันของอัลกอริธึมพันธุการมาช่วยในการปรับปรุงค่าดังนี้

$$v[]=v[]+c_1 rand()*(pbest[] \otimes present[]) + c_2*rand()*(gbest[] \otimes present[]) \quad \text{สมการ 6.25}$$

$$present[] = present[] \otimes v[] \quad \text{สมการ 6.26}$$

จากสมการ 6.25 และ 6.26 เครื่องหมาย $\otimes$ หมายถึงการครอสโอเวอร์ ในที่นี้หมายถึงการครอสโอเวอร์กันระหว่างอนุภาคปัจจุบัน กับ pbest และ gbest

สำหรับการทำครอสโอเวอร์ของอนุภาคนั้น เราจะต้องเริ่มต้นจากการแทนค่าของลำดับการวางชิ้นงานออกมาเป็นโครโมโซมก่อน การแสดงลำดับการวางชิ้นงานลงบนแผ่นโลหะให้เป็นโครโมโซมนั้น เราจะแทนชื่อของชิ้นงานด้วยตัวเลขเช่น 1, 2, 3 เป็นต้น ลำดับการวางชิ้นงานจะเรียงจากซ้ายไปขวา การวางชิ้นงานจะใช้อัลกอริธึมการค้นหาตามแนวนอนต่ำสุด ตัวอย่าง เช่น A = {3,2,4,5,9,8,7,6,1,10} และ B = {9,1,10,3,2,5,4,6,7,8} ตัวเลขที่อยู่ในวงเล็บคือชื่อของชิ้นงานที่ถูกจัดลำดับการวางตามอัลกอริธึมของการค้นหาตามแนวนอนต่ำสุด และค่าฟิตเน็สของ A และ B คือ f(A) และ f(B) ที่หาได้จากการคำนวณพื้นที่ทั้งหมดดังนี้

$$f = h/l \quad \text{สมการ 6.27}$$

$$h = \sum(w_i \times h_i)/W \text{ เมื่อ } i = 1,2, \ldots n \quad \text{สมการ 6.28}$$

เมื่อ h คือค่าความสูงที่ดีที่สุดในทางทฤษฎี ซึ่งเท่ากับผลรวมของพื้นที่ทั้งหมดหารด้วยความกว้างของแผ่นโลหะที่จะถูกตัด ค่า l เป็นค่าสูงที่สุดที่เกิดขึ้นเมื่อการจัดวางชิ้นงานสิ้นสุด

สำหรับการทำครอสโอเวอร์มีขั้นตอนการทำงานเพื่อป้องกันการซ้ำกันของชิ้นงานดังนี้

ขั้นตอนที่ 1: สมมุติว่าเรามีโครโมโซม 2 ชุดคือ A = {3,2,4,5,9,8,7,6,1,10} และ B = {9,1,10,3,2,5,4,6,7,8} ให้เราหาตำแหน่งของยีนในโครโมโซมที่จะทำการครอสโอเวอร์ สมมุติว่าเป็นตำแหน่งที่ห้า

ดันนั้น A และ B จะถูกแบ่งเป็น {3,2,4,5,9 | 8,7,6,1,10} {9,1,10,3,2 | 5,4,6,7,8}

ขั้นตอนที่ 2: เปรียบเทียบส่วนแรกของโครโมโซม A และ B เราจะพบว่าชิ้นงาน 3, 2, 9 ซ้ำกัน ลบ 3, 2, 9 ออกจากส่วนหน้าของโครโมโซมทั้งสอง โครโมโซม A จะกลายเป็น p[] = {4,5} และโครโมโซม B จะกลายเป็น q[] = {1,10}

ขั้นตอนที่ 3: เปรียบเทียบส่วนหางของโครโมโซมทั้งสอง ลบส่วนที่เหมือนกันออกเช่นเดียวกับขั้นที่ 2 จะได้โครโมโซม A เป็น {1,10} และ B เป็น {5,4} ให้ทำการแทนค่าของยีนใน p[] และ q[] ตามลำดับ และทำการครอสโอเวอร์ ผลที่ได้ออกมาคือ A = {8,7,6,4,5} และ B = {1,10,6,7,8}

ขั้นตอนที่ 4: โครโมโซมที่จะนำไปใช้ในการครอสโอเวอร์จะได้เป็น

A'= {3,2,4,5,9,8,7,6,4,5}
B'= {9,1,10,3,2,1,10,6,7,8}

เมื่อทำการครอสโอเวอร์แล้วผลที่ได้จะออกมาเป็น

AA = {3,2,4,5,9,1,10,6,7,8}
BB= {9,1,10,3,2,8,7,6,4,5}

อัลกอริธึมของการหาค่าเหมาะที่สุดในการตัดสต็อก ให้จำนวนอนุภาคเท่ากับ n กำหนดให้จำนวนรอบของการทำงาน (Iteration) เป็น i เริ่มต้นที่ 1 และให้จำนวนรอบของการทำงานมากสุดเท่ากับ tmax คำตอบเริ่มต้นของอนุภาคแต่ละตัวให้สร้างด้วยการสุ่มเป็น x_i^0 (i = 1, 2, ... n) เมื่อ i เป็นตำแหน่งของอนุภาคที่มีทั้งหมดจำนวน n อนุภาค และ 0 เป็นจำนวนรอบที่คำตอบของอนุภาคที่เปลี่ยนไปตามจำนวนรอบของการหาคำตอบ ซึ่งมีค่ามากที่สุดเท่ากับ tmax และให้ x_j^t เป็นคำตอบของอนุภาคในปัจจุบันที่อนุภาคที่ j กำหนดให้ค่า $f(x_j^t)$ เป็นค่าฟิตเน็สที่เกิดจากการคำนวณตามสมการ 6.25

และสมการ 6.26 และที่รอบการทำงานปัจจุบัน(t) ของอนุภาค j นี้ มีอนุภาคที่มีคำตอบที่ดีที่สุดเป็น pbest โดยค่าฟิตเน็สเท่ากับ f(pbest) และอนุภาคที่ดีที่สุดของทั้งกลุ่มเท่าที่มีมาเป็น gbest โดยที่ค่าฟิตเน็สเท่ากับ f(gbeat) สำหรับอัลกอริธึมจะเป็นดังนี้

for(i = 0; i < tmax; i+ +)
{ for(j = 0; j < n; j + +)
{ Cross x_j'^{t} and $pbest_j$ to get x_j"^{t};
Cross x_j'^{t} and gbest to get x_j"^{t};
Mutate x_j"^{t} to get x_j^{t+1};

โดยการใช้อัลกอริธึมการค้นหาตามแนวนอนต่ำสุด ให้คำนวณค่า $f(x^{t+1})$;

if ($f(x_j^{t+1})$ > $pbest_j$)
$pbest_j$ = $f(x_j^{t+1})$; }
for(k = 0; k < n; k + +)
{ if ($f(pbest_k)$ > f(gbest))
gbest = $pbest_k$; }
}

6.5 สรุป

ในการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคนั้น เราจะสร้าง "อนุภาค" ขึ้นมาชุดหนึ่งด้วยการสุ่ม โดยที่อนุภาคแต่ละตัวจะมีตำแหน่งที่อยู่ และค่าความเร็วเบื้องต้นในปริภูมิปัญหาอยู่ด้วย จากนั้นอนุภาคก็จะเคลื่อนที่ไปในปริภูมิปัญหาตามความเร็วที่ตัวเองมี

ในการหาค่าเหมาะที่สุดนั้น แต่ละอนุภาคจะจดจำตำแหน่งที่ดีที่สุดของตัวเอง ที่เรียกว่า pbest สำหรับอนุภาคทั้งกลุ่ม ก็จะต้องจดจำตำแหน่งที่ดีที่สุดของทั้งกลุ่ม หรือ gbest ของมันด้วย

ในแต่ละรอบของการทำงาน ความเร็วของอนุภาคจะถูกปรับปรุงโดยแรงจูงใจจาก pbest และ gbest ซึ่งให้การเคลื่อนที่ของกลุ่มอนุภาคเดินไปในปริภูมิปัญหาตามทิศที่ให้คำตอบที่ดี ในขณะเดียวกัน

อนุภาคเหล่านี้ก็จะทำการสำรวจผลลัพธ์ที่ดีจากตำแหน่งข้างเคียงด้วย ด้วยการทำเช่นนี้ กลุ่มอนุภาคจะลู่เข้าไปสู่ค่าที่เหมาะที่สุด

เนื่องจาก การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค เป็นกระบวนการที่ทำงานเป็นรอบ ในแต่ละรอบของการทำงาน ความเร็วของอนุภาคแต่ละตัวจะถูกปรับปรุงบนฐานของตัวแปรสำคัญ 3 ตัวคือ ความเร็วในปัจจุบันของอนุภาคนั้น ข้อมูลที่อนุภาคนั้นมีอยู่ และข้อมูลรวมของอนุภาคทั้งกลุ่ม หลังจากนั้น อนุภาคแต่ละตัวจะปรับตำแหน่งของมันโดยการใช้ความเร็วใหม่ที่คำนวณได้ดังนี้

$v[] = v[] + c_1 rand()*(pbest[]-present[]) + c_2*rand()*(gbest[]-present[])$

$present[] = present[] + v[]$

สำหรับการปรับปรุงประสิทธิภาพ หรือความหลากหลายของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค เพื่อทำให้การทำงานของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาคมีประสิทธิภาพที่ดีขึ้นนั้น มีหลายวิธี นับตั้งแต่วิธีการกำหนดค่าเริ่มต้นของความเร็ว การเลือกค่า c_1 และ c_2 ที่เหมาะสมสำหรับการทำงานในแต่ละรอบแทนที่จะเป็นการกำหนดค่าคงที่ การปรับปรุงค่า pbest และ gbest โดยการใช้ค่าเหมาะที่สุดของกลุ่มอนุภาคย่อย (local best) แทนการใช้ของทั้งกลุ่ม (global best)

วิธีการที่ซับซ้อนขึ้นในการปรับปรุงประสิทธิภาพของการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค ก็คือการทำงานแบบผสม โดยการใช้วิธีการนี้ร่วมกับการหาค่าเหมาะที่สุดแบบอื่น เช่นการใช้ PSO ร่วมกับ GA เป็นต้น อีกวิธีหนึ่งก็คือการปรับปรุงกระบวนการค้นหา โดยเฉพาะเมื่อการค้นหาคำตอบต้องเผชิญกับการลู่เข้าสู่คำตอบก่อนเวลาอันควร ด้วยการปรับกระบวนการเคลื่อนที่ หรือพฤติกรรมของกลุ่มอนุภาค

6.6 แบบฝึกหัด

1. จากสมการของ PSO ดังกล่าวข้างล่างต่อไปนี้

   $v[] = v[] + c_1 rand()*(pbest[]-present[]) + c_2*rand()*(gbest[]-present[])$

   $present[] = present[] + v[]$

   ให้อภิปรายผลของ c_1 และ c_2 ที่มีต่อการหาค่าเหมาะที่สุดของกลุ่มอนุภาคตามเงื่อนไขดังต่อไปนี้

   1.1. ถ้าค่า $c_1 > c_2$

   1.2. ถ้าค่า $c_1 < c_2$

2. จากฟังก์ชัน $f = 3x_0^2 + 2x_1^2$ ให้นักศึกษาทดลองเขียนโปรแกรมแบบ PSO เพื่อหาค่าของ x_0 และ x_1 ที่น้อยที่สุด และให้วาดกราฟของอนุภาคที่เคลื่อนที่ไปสู่ค่าที่เหมาะที่สุดในลักษณะดังต่อไปนี้

   2.1. การเคลื่อนที่ของอนุภาคเดียว

   2.2. การเคลื่อนที่ของอนุภาค 3 ตัว

   2.3. อภิปรายความแตกต่างในการหาค่าเหมาะที่สุดของการเคลื่อนที่ของอนุภาคที่มีอนุภาคเดียวกับการเคลื่อนที่ของอนุภาค 3 ตัว

3. ถ้าท่านต้องการปรับปรุงให้ PSO สามารถทำงานได้ดีขึ้น โดยที่ไม่ต้องนำ PSO ไปทำงานร่วมกับการหาค่าเหมาะที่สุดแบบอื่น ท่านคิดว่ามีจุดใดบ้างที่สามารถจะทำได้ และจะมีผลอย่างไรต่อการหาค่าเหมาะที่สุดโดยรวม

4. จากตัวอย่างของการใช้ PSO เพื่อแก้ปัญหาการเดินทางของเซลส์แมน มีการนำ GA มาใช้ร่วมด้วย ให้อภิปรายว่า GA มีส่วนมาช่วยในการหาค่าที่เหมาะที่สุดของ PSO ได้อย่างไร

5. ในการหาค่าเหมาะที่สุดของสมการ $f = 3x_0^2 + 2x_1^2$ และการหาระยะทางที่สั้นที่สุดของการเดินทางของเซลส์แมน จากตัวอย่างที่ให้ไว้ในหนังสือ ปัญหาทั้งสองถูกแก้ด้วย PSO เหมือนกัน แต่มีวิธีการที่ต่างกัน ให้อภิปรายเหตุผลของคำถามดังต่อไปนี้
   5.1. ปัญหาทั้งสองมีลักษณะที่เหมือนหรือแตกต่างกันอย่างไร และลักษณะของปัญหาแต่ละอย่างเรียกว่าอะไร
   5.2. PSO เหมาะสำหรับการแก้ปัญหาแบบใดมากกว่ากัน เพราะอะไร
6. ให้ใช้ PSO หาค่าเหมาะที่สุดของสมการ $f(x) = \sum x_i^2$ เมื่อ $i = 1 \ldots n$
   6.1. ออกแบบอัลกอริธึม และเขียนโปรแกรม
   6.2. เขียนกราฟของการค้นหาเมื่ออนุภาคมีทั้งหมด 10 ตัว

7. จากคำอธิบายในเรื่องการเดินทางของเซลส์แมนข้างต้น ให้เขียนโปรแกรมสำหรับการแก้ปัญหาดังกล่าว และทดสอบโปรแกรมด้วยการแก้ปัญหาการเดินทางของเซลส์แมนของเมืองจำนวน m เมืองที่มีระยะห่างระหว่างเมืองเท่ากับ x_i เมื่อ $i = 1 \ldots m-1$ โดยให้กำหนดค่าของ n และ x_i ขึ้นมาเอง โดยกำหนดจำนวนอนุภาค n = 10, 20 และ 50 ในแต่ละการทดลองของจำนวนอนุภาค n (เช่น n = 10) ให้ทดลอง 10 ครั้ง หาระยะทางที่ดีที่สุด จำนวนรอบของการทำงาน (Iteration) จนได้คำตอบ ให้อภิปรายว่าเมื่อค่า n ต่างกันคำตอบและเวลาที่ได้จะเหมือนหรือต่างกันอย่างไร (สำหรับโจทย์ข้อนี้ ท่านอาจจะหาวิธีการใหม่สำหรับการแก้ปัญหาการเดินทางของเซลส์แมนเองก็ได้)

8. การใช้ PSO สำหรับการแก้ปัญหาการตัดสต็อกแบบมิติเดียว ท่านมีความเข้าใจในเรื่องต่อไปนี้อย่างไร
   8.1. การตัดสต็อกแบบมิติเดียว
   8.2. การค้นหาตามแนวนอนต่ำสุด อธิบายพร้อมยกตัวอย่าง
   8.3. โครโมโซม และการหาค่าฟิตเน็สฟังก์ชัน
   8.4. บทบาทของ GA ต่อการสร้างคำตอบที่เป็นไปได้
   8.5. ความสัมพันธ์ระหว่าง GA และ PSO

9. การทำงานของโรงงานขนาดเล็กแห่งหนึ่ง ทางโรงงานได้รับการว่าจ้างให้ทำงาน 5 ชิ้น (P1-P5) โดยที่โรงงานแห่งนี้มีเครื่องจักรอยู่ 2 เครื่อง (M1, M2) เท่านั้น และจากสถิติในการทำงานของเครื่องจักรที่ผ่านมา ทราบว่าเวลาของเครื่องจักรในการทำงานแต่ละชิ้นจะเป็นไปตามตารางดังต่อไปนี้

งาน	เครื่องจักร	เวลาที่ใช้ในการทำงาน (นาที)
P1	M1	20.00
	M2	9.55
P2	M1	51.15
	M1	20.00
P3	M1	41.84
	M2	17.95
P4	M1	80.84
	M2	40.33
P5	M1	12.20
	M2	5.20

ถ้าเราต้องการหาลำดับการทำงานของเครื่องจักรกับชิ้นงานว่า เครื่องจักรใดควรทำงานชิ้นใดก่อนหลัง เพื่อที่จะทำให้งานทั้ง 5 ชิ้นเสร็จเร็วที่สุด

9.1. ออกแบบอัลกอริธึม PSO และเขียนโปรแกรมเพื่อที่จะแก้ปัญหาดังกล่าว

9.2. เขียนกราฟของการค้นหาเมื่ออนุภาคมีทั้งหมด 10 ตัว

บทที่ 7 การหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์

Artificial Bee Colony Optimization

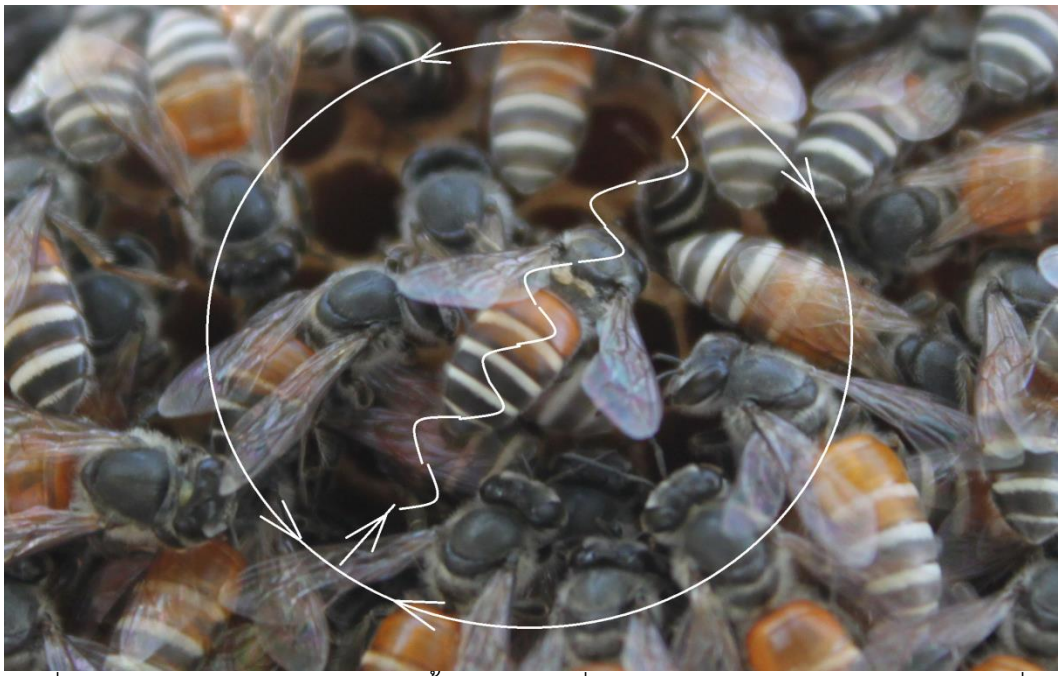
7.1 คำนำ

การศึกษาเรื่องพฤติกรรมการหาอาหารของฝูงผึ้งมีมานานแล้ว คาร์ล วอน ฟริส์ช (Frisch, 1967) เป็นคนแรกที่ได้อธิบายถึงความสามารถของผึ้งในการสื่อสารด้วยการเต้นระบำส่ายท้อง (Waggle Dance) การเต้นระบำของผึ้งนี้เป็นการบอกให้ผึ้งตัวอื่นได้รับทราบถึงระยะทาง ทิศทาง และคุณภาพของแหล่งอาหารที่ตัวมันไปพบมา ความฉลาดในการหาอาหารนี้เกิดขึ้นเนื่องจากการทำงานเป็นฝูง ที่มีการแบ่งการทำงานเป็นกลุ่ม แต่ละกลุ่มจะมีหน้าที่ที่ชัดเจน การหาอาหารของผึ้งรังหนึ่งมีองค์ประกอบสำคัญที่ทำให้เกิดกระบวนการคัดเลือกแหล่งอาหารที่ดีคือ แหล่งอาหาร ผึ้งหาอาหารที่เป็นเอ็มพลอย (Employed Foragers) และผึ้งหาอาหารที่ไม่เป็นเอ็มพลอย (Unemployed Foragers) นอกจากองค์ประกอบดังกล่าว ฝูงผึ้งยังมีโมเดลของพฤติกรรมที่สำคัญสองอย่างคือ การชักชวนกันไปยังแหล่งอาหาร และการยุติการสำรวจแหล่งอาหาร คาราโบกา (Karaboga, 2005) ได้อธิบายถึงความสัมพันธ์ขององค์ประกอบดังกล่าวร่วมกับโมเดลของพฤติกรรมการหาอาหารไว้ดังนี้

- แหล่งอาหาร คุณภาพของแหล่งอาหารขึ้นอยู่กับองค์ประกอบหลายอย่างเช่น ระยะห่างจากรัง ความเข้มข้นและจำนวนของพลังงานในแหล่งอาหาร ความยากง่ายในการนำอาหารออกจากแหล่ง เพื่อให้การบ่งบอกถึงคุณภาพของแหล่งอาหารนี้ง่ายขึ้น เราสามารถแสดงคุณสมบัติทั้งหมดดังกล่าวได้เพียงค่าเดียว (Seeley, 1995)
- ผึ้งหาอาหารที่เป็นเอ็มพลอย ผึ้งกลุ่มนี้จะเป็นผึ้งเอ็มพลอยที่ได้พบแหล่งอาหารที่มันชอบแล้ว และมีความเกี่ยวพันโดยตรงกับแหล่งอาหารที่พวกมันกำลังสำรวจอยู่ พวกมันจะนำข้อมูลข่าวสารเกี่ยวกับแหล่งอาหารที่มันสำรวจ ระยะทาง และทิศทางจากรัง และคุณภาพของแหล่งอาหาร ไปทำการแลกเปลี่ยนกับผึ้งตัวอื่นๆ (ด้วยความน่าจะเป็น p จำนวนหนึ่ง)

- ผึ้งหาอาหารที่ไม่เป็นเอ็มพลอย ผึ้งกลุ่มนี้จะทำการเสาะหาแหล่งอาหารอยู่ตลอดเวลา และยังไม่พบแหล่งอาหารที่มันประทับใจ ผึ้งกลุ่มนี้แบ่งออกเป็น 2 พวก พวกแรกคือผึ้งค้นหา (Scouts) กลุ่มนี้จะทำการสำรวจพื้นที่รอบๆ รังของมันเพื่อหาแหล่งอาหารแหล่งใหม่ และพวกที่สองคือผึ้งรับสาร (Onlookers) กลุ่มนี้จะรออยู่ที่รัง เพื่อคอยเลือกแหล่งอาหารด้วยการพิจารณาจากข่าวสารที่ได้รับจากผึ้งหาอาหารที่เป็นเอ็มพลอย ในรังหนึ่งๆ มีการประมาณกันว่า มีผึ้งหาอาหารที่ไม่ใช่เอ็มพลอยประมาณร้อยละ 5-10 ของจำนวนของผึ้งค้นหา (Seeley, 1995)

ส่วนที่สำคัญที่สุดของการทำให้ผึ้งสามารถเลือกแหล่งอาหารที่ดีคือ การระดมความรู้เกี่ยวกับแหล่งอาหารที่มาจากผึ้งแต่ละตัว ความรู้นี้ได้มาจากการแลกเปลี่ยนข้อมูลข่าวสารระหว่างกันในฝูงผึ้ง เมื่อพิจารณาพื้นที่ทั้งหมดบนรังของผึ้ง จะมีพื้นที่เป็นของส่วนกลางสำหรับผึ้งใช้ในการสื่อสารกันเกี่ยวกับแหล่งอาหารที่ผึ้งแต่ละตัวไปพบมา วิธีการที่ผึ้งใช้ในการสื่อสารกันคือการเต้นรำแบบส่ายท้อง (Waggle Dance) ซึ่งผึ้งจะส่ายท้องของมันและเดินเป็นรูปเลขแปด พื้นที่ที่ผึ้งใช้สำหรับเต้นรำนี้จะมีอยู่อย่างกระจัดกระจายไปทั่วรังผึ้ง บริเวณดังกล่าวเปรียบเสมือนเวทีเต้นรำของผึ้ง



รูปที่ 7.1 การเต้นรำแบบส่ายท้องของผึ้งเอ็มพลอยเพื่อรับสมัครสมาชิก (ภาพโดยปรีชา รอดอิ่ม)

เนื่องจากข้อมูลข่าวสารที่มีอยู่เป็นจำนวนมากในพื้นที่ของเวทีเต้นรำ ผึ้งรับสารจะติดตามข้อมูลข่าวสารนี้ผ่านการเต้นรำแบบส่ายท้อง ซึ่งการเต้นของผึ้งที่กลับมาจากแหล่งอาหารนี้จะสัมพันธ์กับคุณภาพ และระยะทางของแหล่งอาหารที่มันพบมา ถ้าแหล่งอาหารมีคุณภาพสูง การเต้นของผึ้งก็จะนาน ถ้าคุณภาพของแหล่งอาหารต่ำการเต้นก็จะสั้น สำหรับระยะทางของแหล่งอาหาร ผึ้งจะแสดงออกด้วยขนาดของวงเลขแปดที่มันเต้น โดยที่วงกว้างจะหมายถึงระยะทางไกล การตัดสินใจของผึ้งรับสารจะขึ้นอยู่กับข้อมูลที่ได้จากการเต้นเหล่านี้ และมันก็จะเลือกแหล่งอาหารที่ให้ประโยชน์สูงสุด กระบวนการที่ผึ้งเอ็มพลอยให้ข้อมูลให้กับผึ้งรับสารเพื่อการตัดสินใจนี้เรียกว่า การรับสมัครสมาชิก (Recruitment) ตามรูปที่ 7.1

สำหรับการทำงานของผึ้ง เริ่มจาก ผึ้งที่ทำหน้าที่หาอาหาร ทำหน้าที่เป็น ผึ้งหาอาหารที่ไม่เป็นเอ็มพลอย ซึ่งหมายถึงผึ้งเหล่านี้ยังไม่รู้อะไรเกี่ยวกับแหล่งอาหารที่อยู่รอบๆ รังของมัน ผึ้งเหล่านี้มีโอกาสที่เป็นไปได้ 2 อย่างที่มันจะเลือกเป็น

- เริ่มด้วยการเป็นผึ้งค้นหาและเริ่มค้นหารอบๆ รังเพื่อหาอาหาร ซึ่งการออกค้นหานี้อาจจะมาจากแรงกระตุ้นภายในตัวมันเอง หรือสิ่งบอกเหตุจากข้างนอก
- หลังจากที่เฝ้าดูการเต้นระบำส่ายท้องมาระยะหนึ่ง มันจะสมัครใจเป็นผู้ติดตามผึ้งที่ให้ข้อมูลกับมัน และบินไปยังแหล่งอาหารนั้น เมื่อบินไปถึงแหล่งอาหารมันก็จะทำการสำรวจ ซึ่งจะทำให้มันกลายเป็น ผึ้งหาอาหารที่เป็นเอ็มพลอย แล้วก็ทำการดูดเอาอาหาร (Nectar) จากแหล่งอาหารนั้น แล้วบินกลับรัง จากนั้นมันจะคายน้ำหวานนั้นไว้ในที่เก็บอาหาร แล้วมันจะทำอย่างใดอย่างหนึ่งดังต่อไปนี้
    - มันจะไม่เป็นผู้ติดตามผึ้งตัวเดิมที่ให้ข้อมูลกับมัน และละทิ้งแหล่งอาหารนี้
    - มันจะเต้นรำส่ายท้อง และรับสมัครผู้ติดตาม ก่อนที่จะกลับไปยังแหล่งอาหารเดิม
    - มันจะกลับไปยังแหล่งอาหารเดิม โดยไม่สนใจที่จะรับสมัครผึ้งตัวอื่นไปด้วย

ในการหาอาหารของผึ้ง ไม่ใช่ว่าผึ้งทุกตัวจะบินออกไปหาอาหารพร้อมกัน ผึ้งแต่ละตัวจะพิจารณาว่า จำนวนผึ้งที่ออกไปหาอาหารอยู่ในปัจจุบันมีจำนวนที่เพียงพอแล้วหรือยัง เมื่อเทียบกับ

ประชากรของมัน ถ้าไม่พอเมื่อไหร่ มันก็จะทำหน้าที่หาอาหารทันที ในการทำหน้าที่หาอาหารของผึ้งนี้ มีประเด็นที่น่าสนใจเกี่ยวกับการจัดการตัวมันเองในการทำงานคือ

- การป้อนกลับเชิงบวก (Positive feedback) คือเมื่อปริมาณอาหารของแหล่งอาหารเพิ่มขึ้น จำนวนของผึ้งรับสารที่จะมายังแหล่งอาหารนี้ก็จะเพิ่มขึ้นด้วย
- การป้อนกลับเชิงลบ (Negative feedback) ในระหว่างการหาอาหาร แหล่งอาหารที่มีคุณภาพต่ำก็จะถูกยกเลิกโดยการตัดสินใจของผึ้ง
- ความแปรปรวน (Fluctuation) ผึ้งค้นหา จะทำการค้นหาแหล่งอาหารใหม่ด้วยการสุ่ม
- การปฏิสัมพันธ์พหุคูณ (Multiple interaction) ผึ้งจะแลกเปลี่ยนข้อมูลข่าวสารเกี่ยวกับแหล่งอาหารกับเพื่อนร่วมรังของมันที่พื้นที่เต้นรำ

7.2 การหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์

อัลกอริธึมของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ (Artificial Bee Colony) หรือ เอบีซี (ABC) พัฒนามาจากกระบวนการหาอาหารของผึ้ง เพื่อใช้ในการหาคำตอบสำหรับปัญหาการหาค่าเหมาะที่สุด องค์ประกอบที่สำคัญของอัลกอริธึมนี้ก็คือ

- แหล่งอาหาร (Food Source) ของผึ้งที่ใช้แทนคำตอบที่เป็นไปได้ของปริภูมิปัญหา
- ค่าฟิตเน็ส (Fitness Value) เป็นค่าที่ใช้ในการวัดคุณค่าของแหล่งอาหาร ในการนำมาใช้กับการหาค่าที่เหมาะที่สุด จะหมายถึงคุณค่าของคำตอบที่เป็นไปได้ ซึ่งค่านี้หาได้จากอ็อปเจ็คทีฟฟังก์ชัน ที่จะนำไปใช้ในการเปรียบเทียบกับคุณค่าของคำตอบที่เป็นไปได้อื่นๆ ในปริภูมิปัญหา
- เอเยนต์ผึ้ง (Bee Agent) เป็นชุดของเอเย็นต์ที่ใช้สำหรับการคำนวณ สำหรับผึ้งในอัลกอริธึมของ ABC นั้นจะแบ่งออกได้เป็น 3 กลุ่มคือ ผึ้งเอ็มพลอย (Employed Bees) ผึ้งรับสาร (Onlooker Bees) และ ผึ้งค้นหา (Scout Bees) ในผึ้งฝูงหนึ่ง ผึ้งจะถูกแบ่งออกเป็น 2 กลุ่มเท่าๆ กัน คือ ผึ้งหาอาหารที่เป็นเอ็มพลอยหรือผึ้งเอ็มพลอย และผึ้งรับสาร ในการหาคำตอบที่เป็นไปได้แต่ละอันในปริภูมิปัญหาของผึ้งหาอาหารจะประกอบด้วยพารามิเตอร์ต่างๆ ที่บ่งถึงการหาค่าเหมาะที่สุด เช่นตำแหน่งต่างๆ ของแหล่งอาหาร ในการจำลองแบบการหาอาหารของ

ผึ้งสำหรับการหาค่าเหมาะที่สุด เรากำหนดให้จำนวนของผึ้งเอ็มพลอยเท่ากับจำนวนของแหล่งอาหาร

เมื่อเริ่มต้นของการทำงาน ผึ้งเอ็มพลอยจะทำหน้าที่ในการสำรวจแหล่งอาหาร และนำข้อมูลของแหล่งอาหารนั้นไปบอกกล่าวกับผึ้งรับสาร และผึ้งรับสารจะตัดสินใจว่าจะเลือกหรือไม่เลือกแหล่งอาหารนั้นจากข้อมูลที่ได้รับ แหล่งอาหารที่มีคุณภาพสูงก็จะมีโอกาสในการถูกเลือกสูงกว่าแหล่งอาหารที่มีคุณภาพต่ำกว่า ซึ่งคุณภาพของแหล่งอาหารนี้หาได้จากฟิตเน็สฟังก์ชัน สำหรับผึ้งเอ็มพลอยที่แหล่งอาหารของมันไม่ได้รับการยอมรับจากเพื่อนของมันที่เป็นผึ้งรับสาร ตัวมันเองที่เป็นผึ้งเอ็มพลอยอยู่จะผันไปเป็นผึ้งค้นหา และทำการสุ่มหาแหล่งอาหารแหล่งใหม่ รายละเอียดของอัลกอริธึมจะเป็นดังนี้

เริ่มแรก ตำแหน่ง ของแหล่งอาหารจะถูกสร้างขึ้นมาโดยการสุ่ม โดยที่ค่าอ็อปเจ็คทีฟฟังก์ชันที่ใช้สำหรับการคำนวณหาคุณค่าของแหล่งอาหารหาได้ดังสมการ 7.1

$$\square(\square_{\square}), \square_{\square} \in \square^{\square}, \square \in \{1,2,3, \ldots \square\square\} \qquad \text{สมการ 7.1}$$

ค่า x_i เป็นตำแหน่งของแหล่งอาหารที่เป็นเวคเตอร์ มีมิติทั้งหมดเท่ากับ D ค่า $F(x_i)$ เป็นค่าอ็อปเจ็คทีฟฟังก์ชันที่ใช้ในการคำนวณคุณค่าของแหล่งอาหาร และ SN เป็นจำนวนแหล่งอาหาร

หลังจากการกำหนดค่าเริ่มต้น ผึ้งทั้งฝูงก็จะเข้าสู่กระบวนการค้นหาตำแหน่งของแหล่งอาหาร โดยกระบวนการทั้งหมดจะเป็นการวนรอบการทำงานของกระบวนการหลัก 4 กระบวนการดังนี้

การปรับปรุงคำตอบที่เป็นไปได้ของผึ้งเอ็มพลอย
การเลือกคำตอบที่เป็นไปได้ของผึ้งรับสาร
การปรับปรุงคำตอบที่เป็นไปได้ของผึ้งรับสาร
การหลีกเลี่ยงคำตอบที่จะเข้าสู่ทางตันของผึ้งค้นหา

7.2.1 การปรับปรุงคำตอบที่เป็นไปได้ของผึ้งเอ็มพลอย

เมื่อได้ตำแหน่งของแหล่งอาหารมาแล้ว ผึ้งเอ็มพลอยจะค้นหาแหล่งอาหารเพิ่มเติม เพื่อนำมาเปรียบเทียบกับแหล่งอาหารที่มันมีอยู่ โดยการสุ่มเอาตำแหน่งของแหล่งอาหารที่อยู่ข้างเคียงกัน แล้วผึ้งเอ็มพลอยจะเลือกแหล่งอาหารที่ดีกว่าด้วยสมการ 7.2 ดังนี้

$$□_{□□} = □_{□□} + □_{□□}(□_{□□} + □_{□□}) \quad \text{สมการ 7.2}$$

จากสมการ 7.2 ค่า v_{ij} เป็นคำตอบหรือแหล่งอาหารที่เป็นไปได้แหล่งใหม่ ที่หามาจากแหล่งอาหารข้างเคียงกับแหล่งอาหารเดิม x_{ij} ซึ่งตำแหน่งของแหล่งอาหารข้างเคียงนี้ที่เลือกมาโดยการสุ่ม x_{kj} ค่า $□_{□□}$ เป็นค่าที่ได้มาจากการสุ่มที่มีช่วงระหว่าง [-1, 1] ค่านี้เป็นค่าสำหรับการปรับน้ำหนักของการเปลี่ยนแปลงค่าจากค่าเดิมเป็นค่าใหม่ของการทำงานในรอบต่อไป สำหรับค่าของ $k \in \{1,2,3 \ldots SN\}$ เป็นดัชนีที่เลือกขึ้นมาด้วยการสุ่ม และ $j \in \{1,2,3 \ldots D\}$ เป็น j เป็นมิติของแหล่งอาหาร ความแตกต่างของ x_{ij} และ x_{kj} คือตำแหน่งของแหล่งอาหารที่ต่างกันแต่มีมิติเดียวกันในแหล่งอาหาร ถ้าแหล่งอาหารใหม่ v_{ij} ดีกว่าแหล่งอาหารเดิม x_{ij} ให้แทนค่าแหล่งอาหารเดิมด้วยแหล่งอาหารใหม่

7.2.2 การเลือกคำตอบที่เป็นไปได้ของผึ้งรับสาร

เมื่อผึ้งเอ็มพลอยทั้งหลายพบแหล่งอาหารที่ต้องการแล้ว ผึ้งเหล่านี้จะกลับมายังรังของมันพร้อมทั้งนำข้อมูลของแหล่งอาหารเหล่านั้นมาบอกกับเหล่าผึ้งรับสารทั้งหมด ผึ้งรับสารจะเลือกรังเหล่านี้ด้วยความน่าจะเป็น ที่มีการกำหนดน้ำหนัก โดยที่แหล่งอาหารที่มีค่าฟิตเน็สสูงจะมีโอกาสได้รับเลือกสูง สมการของความน่าจะเป็นในการเลือกแหล่งอาหารต่างๆ จะเป็นดังนี้

$$□_{□} = \frac{□□□_{□}}{\sum_{□=1}^{□□} □□□_{□}} \quad \text{สมการ 7.3}$$

สำหรับค่า fit_i เป็นค่าฟิตเน็สของแหล่งอาหารที่ตำแหน่งที่ i ซึ่งเป็นค่าที่หามาได้จากอ็อปเจ็คทีฟ ฟังก์ชัน $F(x_i)$ ของแหล่งอาหาร i สำหรับค่าของ fit_i หาได้จากสมการดังต่อไปนี้

$$□□□ = \begin{cases} \frac{1}{1+□(□)} & □(□) \geq 0 \\ \frac{1}{□□□(□(□))} & □(□) < 0 \end{cases}$$

โดยที่ค่า $f(x)$ หมายถึงค่าที่ได้จากฟังก์ชันของแหล่งอาหารต่างๆ ของฟังก์ชัน $F(x_i)$ ของแหล่งอาหาร i

7.2.3 การปรับปรุงคำตอบที่เป็นไปได้ของผึ้งรับสาร

หลังจากที่ผึ้งรับสารเลือกแหล่งอาหารจากผึ้งเอ็มพลอยแล้ว พวกมันจะปรับปรุงคำตอบของแหล่งอาหารที่พวกมันเลือกไว้ด้วยสมการ 7.2 เช่นเดียวกันกับการปรับปรุงคำตอบของผึ้งเอ็มพลอย แหล่งอาหารเดิมจะถูกแทนที่ด้วยแหล่งอาหารใหม่ถ้าแหล่งอาหารใหม่ดีกว่า ถ้าแหล่งอาหารใหม่ไม่ดีกว่าแหล่งอาหารเดิมก็จะถูกใช้ในการเปรียบเทียบของการเลือกในรอบต่อไป

7.2.4 การหลีกเลี่ยงคำตอบที่ลู่เข้าสู่ทางตันของผึ้งค้นหา

ขั้นตอนนี้เกิดจากการที่ผึ้งเอ็มพลอยเริ่มค้นพบว่าแหล่งอาหารในบริเวณที่มันไปหาอาหารนั้นเริ่มไม่มีคุณภาพที่ดีพอ เพราะไม่ถูกเลือกจากผึ้งรับสาร ผึ้งเอ็มพลอยนี้ก็จะเปลี่ยนตัวเองให้เป็นผึ้งค้นหาโดยที่มันจะทำการบินสุ่มไปยังตำแหน่งใหม่เพื่อหาแหล่งอาหาร ตำแหน่งใหม่ของแหล่งอาหารนี้สามารถหาได้จากสมการดังต่อไปนี้

$$□_{□□} = □_{□}^{□□□} + □□□□[0,1] * (□_{□}^{□□□} - □_{□}^{□□□}) \qquad \text{สมการ 7.4}$$

จากสมการ 7.4 ค่า x_{ij} นี้ ค่า i หมายถึงคำตอบที่เป็นไปได้ และ j หมายถึงมิติของคำตอบที่เป็นไปได้ในปริภูมิปัญหา หรืออาจจะกล่าวอีกนัยหนึ่งก็คือ x_{ij} เป็นคำตอบที่ i ของมิติในแหล่งอาหารที่ j สำหรับค่า x_j^{min} เป็นขอบเขตต่ำสุดของตำแหน่งของแหล่งอาหารในมิติที่ j และ x_j^{max} เป็นขอบเขตสูงสุดของตำแหน่งของแหล่งอาหารที่มิติที่ j เมื่อพิจารณาจากแหล่งอาหาร x_{ij} ที่ผึ้งเอ็มพลอยพบมาทั้งหมด หรือกล่าวอีกนัยหนึ่งคือ x_j^{max} และ x_j^{min} เป็นค่าที่มากที่สุดและน้อยที่สุดของมิติที่ j ของแหล่งอาหารทั้งหมด

7.2.5 อัลกอริธึมของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์

จากทั้ง 4 ขั้นตอน ดังที่กล่าวมาจะทำงานวนรอบกันไป จนกระทั่งพบคำตอบ หรือจนกระทั่งจำนวนรอบเท่ากับ MCN ซึ่ง MCN คือจำนวนรอบสูงสุด (Maximum Cycle Number) ที่นำเสนอโดย คาราโบกา (Karaboga D. a., 2007) อัลกอริธึมของ ABC ที่เป็นเมตาฮิวริสติกสามารถเขียนเป็นรหัสเทียมได้ดังแสดงในรูปที่ 7.2

```
Procedure ABC_Metaheuristic
กำหนดค่าเริ่มต้น
While not(MCN) เมื่อยังไม่ถึงจำนวนรอบสูงสุด
    Update_Feasible_Solutions(Employed bees) ปรับปรุงคำตอบที่เป็นไปได้ของผึ้งเอ็มพลอย
    Select_Feasible_Solutions(Onlooker bees) เลือกคำตอบที่เป็นไปได้ของผึ้งรับสาร
    Update_Feasible_Solutions(Onlooker bees) ปรับปรุงคำตอบที่เป็นไปได้ของผึ้งรับสาร
    Avoid_Sub-Optimal_Solutions (Scout bee) หลีกเลี่ยงคำตอบที่ลู่เข้าสู่ทางตันของผึ้งค้นหา
End while
End-Procedure
```

รูปที่ 7.2 เอบีซี เมตาฮิวริสติก

7.3 ความหลากหลายของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์

การปรับปรุงอัลกอริธึมของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์นี้มีหลายวิธี จุดประสงค์สำคัญคือการทำให้อัลกอริธึมสามารถทำงานได้ดีขึ้น ซึ่งในตอนต่อไปนี้จะกล่าวถึงเฉพาะบางอัลกอริธึมที่สำคัญเท่านั้น

7.3.1 เบสโซฟา เอบีซี

อัลกอริธึมเบสโซฟา เอบีซี (Best-so-far ABC Algorithm) เป็นอัลกอริธึมที่ดัดแปลงมาจาก เอบีซี ธรรมดา นำเสนอโดย อนันต์ ธีรณี และบุญเจริญ (Banharnsakun, 2011) เพื่อเพิ่มความสามารถในกระบวนการค้นหา โดยการปรับเปลี่ยนวิธีการปรับปรุงคำตอบของแหล่งอาหาร ในอัลกอริธึม เอบีซี เดิม

ทั้งผึ้งเอ็มพลอย และผึ้งรับสารจะมีวิธีการปรับปรุงคำตอบ (แหล่งอาหาร) ที่พวกมันเลือกไว้ด้วยวิธีการเดียวกัน คือการนำเอาแหล่งอาหารที่มันมีอยู่เปรียบเทียบกับแหล่งอาหารข้างเคียงดังแสดงในสมการ 7.2 สำหรับเบสโซฟา เอบีซี ผึ้งรับสารจะใช้ข้อมูลข่าวสารจากผึ้งเอ็มพลอยทั้งหมดสำหรับการตัดสินใจเลือกแหล่งอาหาร ผึ้งรับสารนี้จะเปรียบเทียบข้อมูลข่าวสารจากแหล่งอาหารทั้งหมดที่ได้รับจากผึ้งเอ็มพลอย แล้วค่อยเลือกแหล่งอาหารที่ดีที่สุด ดังนั้นสมการ 7.2 ที่ผึ้งรับสารใช้แต่เดิมจึงต้องมีการปรับปรุง โดยที่แหล่งอาหารเดิมจะถูกปรับปรุงเป็นแหล่งอาหารใหม่ ด้วยการเปรียบเทียบแหล่งอาหารที่มันมีอยู่กับแหล่งอาหารที่ดีที่สุดเท่าที่พบมา (Best-so-far Food Source) แทนที่จะเป็นการเปรียบเทียบกับแหล่งอาหารข้างเคียงเท่านั้น ซึ่งการเปลี่ยนแปลงนี้ทำให้การปรับปรุงแหล่งอาหารของผึ้งรับสารแตกต่างจากของผึ้งเอ็มพลอย สมการใหม่ของผึ้งรับสารที่ใช้ในการปรับปรุงแหล่งอาหารจะเป็นดังนี้

$$\square_{\square\square} = \square_{\square\square} + \Phi\square_{\square}(\square_{\square\square} - \square_{\square\square}) \qquad \text{สมการ 7.5}$$

เมื่อ v_{id} = เป็นแหล่งอาหารใหม่ของผึ้งรับสารที่ตำแหน่ง i มิติ d โดยที่ d = 1, 2, 3, ... D

x_{ij} = เป็นแหล่งอาหารที่ผึ้งรับสารเลือกไว้ที่ตำแหน่ง i มิติ j

$\square$ = เป็นตัวเลขที่เกิดจากการสุ่มที่มีช่วงระหว่าง –1 ถึง 1

f_b = เป็นค่าฟิตเน็สของแหล่งอาหารที่ดีที่สุดเท่าที่พบมา

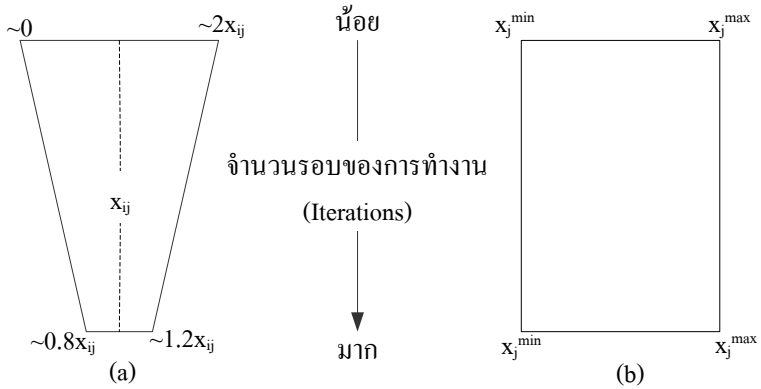
x_{bj} = เป็นแหล่งอาหารที่ดีที่สุดเท่าที่พบในมิติ j

แม้ว่าวิธีการของ เบสโซฟา เอบีซี สามารถเพิ่มความเร็วของการค้นหาคำตอบได้ดีขึ้นเมื่อเทียบกับวิธีการของ เอบีซี เดิม ในบางครั้งคำตอบที่ได้มักจะเป็นคำตอบที่ลู่เข้าสู่ทางตันของโลคอลอ็อปติมั่ม เพื่อที่จะแก้ปัญหาดังกล่าว เบสโซฟา เอบีซี ได้ทำการปรับปรุงกระบวนการหาแหล่งอาหารใหม่ของผึ้งค้นหา โดยที่ในช่วงรอบแรกๆ ของการค้นหา แหล่งอาหารที่ได้จะยังห่างจากจุดที่เหมาะที่สุด (Optimum Solution) มาก แล้วค่อยเข้าใกล้จุดที่เหมาะที่สุดในการค้นหาที่รอบมากขึ้น เมื่อแหล่งอาหารที่หาได้ใหม่จากผึ้งเอ็มพลอยมีคุณภาพไม่เปลี่ยนแปลง ไม่ว่าจะมีรอบการค้นหาเพิ่มขึ้นเท่าไร ผึ้งค้นหาจะสุ่มหาแหล่งอาหารใหม่ด้วยการเปลี่ยนวิธีการจากสมการ 7.4 ซึ่งเป็นการสุ่มจากช่วงของแหล่ง

อาหารที่ดีที่สุดและที่แย่ที่สุดที่มันเคยหามา เป็นสมการ 7.6 ที่กำหนดจากระยะห่างจากแหล่งอาหารเดิมที่มันกำลังจะยกเลิก และระยะห่างนี้จะแปรผกผันกับจำนวนรอบของการค้นหา ซึ่งหมายถึงรอบยิ่งมากระยะห่างยิ่งใกล้ ดังนี้

$$\square_{\square\square} = \square_{\square\square} + \square_{\square\square}[\square_{\square\square\square} - \frac{\square\square\square\square\square\square\square\square\square}{\square\square\square}(\square_{\square\square\square} - \square_{\square\square\square})]\square_{\square\square} \qquad \text{สมการ 7.6}$$

เมื่อค่า v_{ij} เป็นแหล่งอาหารที่ผึ้งค้นหาหาขึ้นมาใหม่ โดยการปรับจากตำแหน่งของแหล่งอาหารปัจจุบันที่มันกำลังจะยกเลิก x_{ij} และ $\square_{\square\square}$ เป็นค่าที่เกิดจากการสุ่มระหว่าง [-1, 1] ค่า $\square_{\square\square\square}$ และค่า $\square_{\square\square\square}$ เป็นค่าคงที่ที่คิดเป็นร้อยละของการปรับจากตำแหน่งของแหล่งอาหารเดิม จากการทดลองค่าที่เหมาะที่สุดของ $\square_{\square\square\square}$ และ $\square_{\square\square\square}$ คือ 1 และ 0.2 ตามลำดับ จากการทดลองด้วยค่าของค่า $\square_{\square\square\square}$ และค่า $\square_{\square\square\square}$ ในสมการที่ 7.6 ทำให้ตำแหน่งของแหล่งอาหารที่ผึ้งค้นหาสุ่มหาออกมาจะลดลงแบบเชิงเส้นจากร้อยละ 100 ถึงร้อยละ 20 ในแต่ละการทดลองดังแสดงตามรูปที่ 7.3 ขอบเขตของการค้นหาของผึ้งค้นหา (a) วิธีการ เบสโซฟา เอบีซี (b) วิธีการ เอบีซี ดั้งเดิม



รูปที่ 7.3 ขอบเขตของการค้นหาของผึ้งค้นหา (a) วิธีการ เบสโซฟา เอบีซี (b) วิธีการ เอบีซี ดั้งเดิม

อีกเรื่องหนึ่งที่ เบสโซฟา เอบีซี ได้ปรับปรุงจาก เอบีซี เดิมคือ วิธีการเปรียบเทียบคำตอบที่เป็นไปได้ เอบีซี เดิมทำการเปรียบเทียบคุณค่าของคำตอบที่เป็นไปได้โดยการใช้ค่าของฟิตเน็สตามสมการ

$$\square\square\square = \begin{cases} \dfrac{1}{1+\square(\square)} & \square(\square) \geq 0 \\ \dfrac{1}{\square\square\square(\square(\square))} & \square(\square) < 0 \end{cases}$$

จุดอ่อนของการใช้สมการฟิตเน็สข้างต้นก็คือ ถ้าค่าของ *f(x)* ที่มีจำนวนเล็กมากๆ เช่น 10^{-20} กับ 10^{-200} ค่าของฟิตเน็ส (Fitness) จะบอกความแตกต่างของค่าทั้งสองไม่ได้ เนื่องจากค่าของ *f(x)* มีจำนวนเล็กเกินไป เมื่อรวมกับ 1 แล้วไปหาร 1 ทำให้ผลคำนวณที่ได้จาก $1/(1+10^{-20})$ กับ $1/(1+10^{-200})$ ไม่มีความแตกต่างกัน ดังนั้นเพื่อป้องกันปัญหาความผิดพลาดที่เกิดจากการคำนวณของการหารด้วยตัวเลขจำนวนน้อยมากๆ ในเบสโซฟา เอบีซี จะใช้ค่าของ *f(x)* เป็นตัวเปรียบเทียบโดยตรง ซึ่งเป็นการเปรียบเทียบค่า 10^{-20} กับ 10^{-200} ทำให้การเปรียบเทียบนี้เห็นความแตกต่างของเลขทั้งสองจำนวน

การเปรียบเทียบการทำงานของเบสโซฟา เอบีซี กับ เอบีซี การเปรียบเทียบนี้ทำโดยการหาค่าที่เหมาะที่สุดของฟังก์ชัน $f(x)= 3 + x_1^2 + x_2^2$ ที่กำหนดช่วงของค่า x ไว้ดังนี้

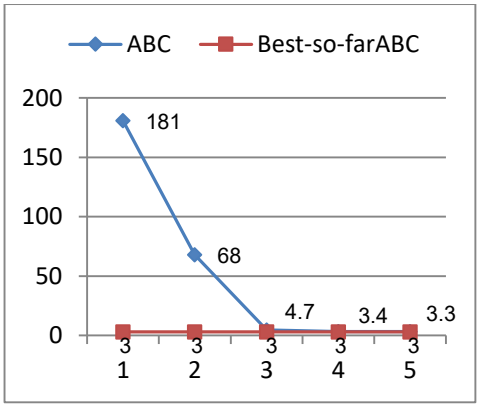
$-100 \leq x_1 \leq 100$

$-100 \leq x_2 \leq 100$

โดยที่กำหนดให้จำนวนของผึ้งเอ็มพลอย (Number of Employed Bees) = 5 และจำนวนของผึ้งค้นหา (Number of Scout Bees) = 1 ผลของการทดลองเมื่อผึ้งทั้งหมดทำงานมา 5 รอบเป็นดังตารางและดังรูปที่ 7.4 ต่อไปนี้

ABC Results	Best-so-far ABC Results
x_1 = -0.609408	x_1 = 0
x_2 = -0.0345927	x_2 = 0
f(x) = 3.37258	f(x) = 3

Itertions	f(x)= 3 + x_1^2 + x_2^2	
	ABC	Best-so-far ABC
1	181.082	3
2	68.2696	3
3	4.73936	3
4	3.433	3
5	3.37258	3



รูปที่ 7.4 การเปรียบเทียบการลู่เข้าสู่คำตอบของ ABC และ Best-so-far ABC

7.3.2 การปรับปรุงการทำงานของเอบีซีแบบอื่น

การทำวิจัยเพื่อปรับปรุง ABC ให้ทำงานได้ดีขึ้นมีอยู่อย่างแพร่หลาย วิธีที่ทำการปรับปรุงล่าสุด เช่น CABC (YAN, 2011) หรือ เคออติก เซิร์ช เอบีซี (Chaotic Search ABC) ที่ผึ้งรับสารจะใช้วิธีการเคออติก เพื่อเพิ่มความสามารถของการค้นหาในการป้องกันไม่ให้การค้นหาแบบท้องถิ่น (Local Search) นั้นติดอยู่ใน โลคอลออปติมั่ม (Local Optimum) ซึ่งเคออติกนี้ถูกนิยามไว้ดังนี้

$$\square_{\square+1} = \square * \square_{\square} * (1 - \square_{\square})$$

เมื่อ $\square$ คือค่าเคออติก แอ็ทแทร็กเตอร์ (Chaotic Attractor) ถ้าค่า $\square$ เท่ากับ 4 แล้วระบบข้างต้นจะเข้าสู่สถานะเคออติกเต็มตัว ตัวแปร x_{i+1} เป็นค่าของตัวแปร x_i ในรอบการทำงานที่ i หลังจากการเกิดลำดับเคออติก (Chaotic Sequence) แหล่งอาหารใหม่ก็จะสามารถคำนวณได้จาก

$$\square = \square_{\square\square} + \square * (2 * \square_{\square} - 1)$$

ค่า x เป็นตำแหน่งใหม่ของแหล่งอาหาร ตัวแปร x_i เป็นตัวแปรเคออติก R เป็นรัศมี (Radius) ของแหล่งอาหารใหม่ และแหล่งอาหาร x_{mi} เป็นศูนย์กลางของการค้นหา เมื่อได้แหล่งอาหารใหม่นี้แล้ว ผึ้งรับสารก็จะทำการเปรียบเทียบแหล่งอาหารที่ได้ใหม่นี้กับแหล่งอาหารเดิม แหล่งที่ดีกว่าก็จะถูกเลือก

GABC (Zhu, 2010) หรือ โกบอล เบส เอบีซี (Global Best ABC) ได้นำแนวคิดของการหาค่าเหมาะสุดแบบกลุ่มอนุภาค (PSO) มาใช้ในการหาแหล่งอาหารข้างเคียง โดยการปรับสมการการหาค่าของ v_{ij} จากเดิมในสมการ 7.2 มาเป็น

$$□_{□□} = □_{□□} + □_{□□}(□_{□□} - □_{□□}) + □_{□□}(□_{□} - □_{□□})$$

เมื่อเทอมที่สามของทางขวามือ เป็นเทอมใหม่ที่เพิ่มเข้ามาเรียกว่า gbest ตัวแปร y_j เป็นตัวแปรในมิติที่ j ของแหล่งอาหารของ gbest ตัวแปร $□_{□□}$ เป็นค่าสุ่มแบบยูนิฟอร์ม (Uniform Random Number) ที่มีค่าจาก [0,C] เมื่อ C เป็นค่าจำนวนเต็มลบ

สำหรับ IABC (Li, 2012) หรือ Improved ABC ได้ปรับสมการ 7.2 ของ ABC ใหม่เป็น

$$□_{□□} = □_{□□}□_{□□} + 2(□_{□□} - 0.5)(□_{□□} - □_{□□})□_1 + □_{□□}(□_{□} - □_{□□})□_2$$

เมื่อ v_{ij} เป็นตำแหน่งของแหล่งอาหารใหม่ และ x_{ij} เป็นตำแหน่งของแหล่งอาหารเดิม ตัวแปร w_{ij} เป็นน้ำหนักอินนิเชีย (Inertia weight) x_j เป็นมิติที่ j ของแหล่งอาหารที่ดีที่สุดเท่าที่พบมา (The best-so-far solution) $□_{□□}$ และ$□_{□□}$ เป็นตัวแปรสุ่มที่มีค่าระหว่าง [0, 1] ซึ่งความสัมพันธ์ระหว่าง w_{ij}, $□_{□□}$ และ $□_{□□}$ เป็นไปตามสมการดังต่อไปนี้

$$□_{□□} = □ = \frac{1}{1 + □^{-\frac{□□□□□□□□(□)}{□□}}}$$

$$□_2 = \begin{cases} 1 & □□□□□ \; □□□□□□□□□ \\ \dfrac{1}{1 + □^{-\frac{□□□□□□□(□)}{□□}}} & □□□□□ \; □□□□□□□□□ \end{cases}$$

เมื่อค่า ap เป็นค่าฟิตเน็สของการทำงานในรอบแรก fitness(1)

7.3.3 การลู่เข้าสู่คำตอบและการทดสอบสมรรถนะ

สำหรับอัลกอริธึม เอบีซี ผึ้งรับสาร และผึ้งเอ็มพลอย จะทำหน้าที่ในการหาคำตอบด้วยกระบวนการเอ็กซ์พล้อยเตชันในปริภูมิค้นหา ในขณะที่ผึ้งค้นหาจะทำหน้าที่เอ็กซ์พลอเรชันในปริภูมิค้นหา ดังนั้นการเพิ่มจำนวนของผึ้งค้นหาเป็นการทำให้เอ็กซ์พลอเรชันดีขึ้น ในขณะที่การเพิ่มจำนวน

ของผึ้งรับสารเป็นการเพิ่มความสามารถของเอ็กซ์พล้อยเตชัน ในพฤติกรรมจริงของผึ้ง อัตราการรับสมัครสมาชิกเพื่อให้ผึ้งรับสารเป็นสมาชิกของผึ้งหาอาหารที่เป็นเอ็มพลอย จะเป็นเสมือนตัววัดที่บอกความเร็วของฝูงผึ้งในการค้นหาหรือเอ็กซ์พลอเรชัน และการเอ็กซ์พล้อยแหล่งอาหารที่ค้นพบใหม่ เช่นกันสำหรับฝูงผึ้งประดิษฐ์อัตราการรับสมัครสมาชิกนี้ จะเป็นมาตรบอกความเร็วของการค้นพบคำตอบที่เป็นไปได้ หรือคำตอบที่ดี ของการหาค่าเหมาะที่สุด การอยู่รอดและการพัฒนาของฝูงผึ้งขึ้นอยู่กับการค้นหาคำตอบที่รวดเร็ว และสมรรถนะในการใช้ประโยชน์จากแหล่งอาหารที่ดี

การหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ ที่กระบวนการเอ็กซ์พลอเรชันจัดการโดยผึ้งค้นหานั้น จะดีสำหรับการหาค่าเหมาะที่สุดแบบโกลบอล และกระบวนการเอ็กซ์พล้อยเตชันที่จัดการโดยผึ้งรับสารและผึ้งเอ็มพลอยนี้ จะดีมากต่อการหาค่าเหมาะที่สุดแบบโลคอล ซึ่งมีการสังเกตกันว่าการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์นี้จะดีที่กระบวนการเอ็กซ์พลอเรชัน แต่จะแย่ที่กระบวนการเอ็กซ์พล้อยเตชัน

เมื่อเปรียบเทียบการทำงานของเอบีซี กับอัลกอริธึมอื่นที่เป็นแบบเมตาฮิวริสติกเหมือนกันเช่น อัลกอริธึมพันธุการ การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค คาราโบกา (Karaboga, 2009) ได้ทำการทดลองดังผลที่แสดงในตารางที่ 7.1

ตารางที่ 7.1 เป็นผลการเปรียบเทียบสมรรถนะของอัลกอริธึม 3 ประเภทได้แก่ อัลกอริธึมพันธุการ (GA) การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค (PSO) และการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ (ABC) โดยมีตัวย่อ SD คือความเบี่ยงเบนมาตรฐาน SEM (Standard Error of Mean) ค่าความคลาดเคลื่อนมาตรฐานของค่าเฉลี่ย จากตารางจะเห็นได้ชัดเจนว่า การหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์นั้นได้ผลออกมาเป็นค่าที่ดีที่สุดเกือบหมด ยกเว้นในกรณีของฟังก์ชันของโรเซนบร็อก ที่ได้ค่าใกล้เคียงค่าดีที่สุด ในขณะเดียวกันผลของฟังก์ชันนี้ได้จากการทำงานของอัลกอริธึมพันธุการนั้นได้ค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐานที่ดีกว่า แต่ค่าความคลาดเคลื่อนมาตรฐานของค่าเฉลี่ยกลับแย่กว่า ซึ่งหมายถึงผลที่ได้รับจากการทดลองบางกรณีไม่ดี ถ้าพิจารณาดูฟังก์ชันของโรเซนบร็อก ซึ่งเป็นฟังก์ชันที่มีค่าที่ดีที่สุดอยู่ในบริเวณที่กว้างมาก ฟังก์ชันในลักษณะนี้จะทำให้อัลกอริธึมเน้นทางเอ็กซ์พล้อยเตชัน เช่นอัลกอริธึมพันธุการ สามารถทำงานได้ดีด้วย สำหรับการหาค่าเหมาะสุดแบบกลุ่มอนุภาคแม้ว่าจะแย่กว่าของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ แต่ค่าที่ได้ก็ยังมีผลดีกว่าอัลกอริธึมพันธุการ ยกเว้นในกรณีของ

ฟังก์ชันของโรเซนบร็อกเท่านั้น ผลสรุปก็คือว่าการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์จะมีสมรรถนะดีที่สุด รองลงมาคือการหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค

ฟังก์ชัน	มิติ	Min		GA	PSO	ABC
ทรงกลม	30	0	ค่าเฉลี่ย	1.11E+03	0	0
			SD	74.214474	0	0
			SEM	13.549647	0	0
โรเซนบร็อก	30	0	ค่าเฉลี่ย	1.96E+05	15.088617	0.0887707
			SD	3.85E+04	24.170196	0.077390
			SEM	7029.106155	4.412854	0.014129
เรสตริจิน	30	0	ค่าเฉลี่ย	52.92259	43.9771369	0
			SD	4.564860	11.728676	0
			SEM	0.833426	2.141353	0
เชฟเฟอร์	2	0	ค่าเฉลี่ย	0.004239	0	0
			SD	0.004763	0	0
			SEM	0.000870	0	0
กรีวังค์	30	0	ค่าเฉลี่ย	10.63346	0.01739118	0
			SD	1.161455	0.020808	0
			SEM	0.212052	0.003799	0
แอคเลย์	30	0	ค่าเฉลี่ย	14.67178	0.16462236	0
			SD	0.178141	0.493867	0
			SEM	0.032524	0.090167	0

ตารางที่ 7.1 การเทียบสมรรถนะของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์และเมตาฮิวริสติกอื่น

สำหรับการเปรียบเทียบสมรรถนะของเอบซี และเบสโซฟา เอบีซีนั้น ในการทดสอบสมรรถนะที่ทำโดย อนันต์ ธีรณี และบุญเจริญ (A., Achalakul, T. and Sirinaovakul, B. Banharnsakun, 2011) ผลที่ได้ดังแสดงไว้ในตารางที่ 7.2 เป็นการทดลองเปรียบเทียบการทำงานของการหาค่าเหมาะที่สุดแบบ

ฝูงผึ้งประดิษฐ์ (หรือเอบีซี) แบบดั้งเดิมและอัลกอริธึมเบสโซฟา เอบีซี ผลที่ได้แสดงให้เห็นว่า การลู่เข้าสู่คำตอบของเบสโซฟา เอบีซี นั้นดีกว่าของ เอบีซีดั้งเดิม ในแทบทุกฟังก์ชัน ยกเว้นในกรณีที่เป็นฟังก์ชันนั้นเป็นฟังก์ชันที่ไม่ซับซ้อนมาก เช่นทรงกลม และโรเซนบร็อก ที่อัลกอริธึมทั้งสองมีจำนวนรอบของการเข้าสู่คำตอบเท่ากัน เนื่องจากว่าเบสโซฟา ใช้เวลาในการทำงานต่อรอบที่นานกว่า เพราะเบสโซฟาได้เพิ่มกระบวนการตรวสอบที่มากขึ้น ทำให้เวลาของการทำงานในภาพรวมทั้งหมดจะนานกว่าเอบีซีแบบดั้งเดิม แม้ว่าจำนวนรอบของการทำงานจะดีกว่าก็ตาม แต่อย่างไรก็ตาม สำหรับฟังก์ชันที่ยากมากๆ เช่น แอคเลย์ ที่จำนวนรอบของการทำงานของเบสโซฟาดีกว่ามาก จึงทำให้เวลาในการทำงานดีกว่าด้วย

ฟังก์ชัน	มิติ	การลู่เข้าสู่คำตอบ			ค่าเหมาะที่สุด	
		เอบีซี	เบสโซฟา		เอบีซี	เบสโซฟา
ทรงกลม	30	1000	1000	เวลา	1.109	1.546
				ค่าเฉลี่ย	2.649E−10	5.641E−302
				SD	2.136E−10	3.967E−301
กรีวังค์	30	1000	845	เวลา	2.344	2.672
				ค่าเฉลี่ย	5.369E−09	0
				SD	2.664E−08	0
เรสตริจิน	30	1000	863	เวลา	1.782	1.953
				ค่าเฉลี่ย	3.593E−07	0
				SD	3.565E−06	0
โรเซนบร็อก	30	1000	1000	เวลา	2.704	2.969
				ค่าเฉลี่ย	0.681802	3.604E−15
				SD	0.774655	3.585E−14
แอคเลย์	30	1000	109	เวลา	1.782	0.234
				ค่าเฉลี่ย	5.246E−06	0
				SD	1.702E−06	0
เชฟเฟอร์	2	2000	947	เวลา	0.672	0.468

				ค่าเฉลี่ย	7.895E−11	0
				SD	4.014E−10	0

ตารางที่ 7.2 ผลการทดสอบสมรรถนะของการหาค่าเหมาะสุดแบบฝูงผึ้ง

ในการหาค่าเหมาะที่สุดก็เช่นกัน จากการทดลองทำงานแต่ละอัลกอริธึมจำนวน 20 รอบ ค่าเฉลี่ยของค่าที่เหมาะที่สุดที่หาได้ของเบสโซฟา เอบีซีจะดีกว่าของเอบีซีดั้งเดิมมาก ที่สำคัญคือค่าความเบี่ยงเบนมาตรฐาน (SD) ไม่มีความแตกต่างจากค่าเฉลี่ยเลย คือมีค่าเท่ากับ 0 ในเกือบทุกกรณี ยกเว้นในกรณีของฟังก์ชันรูปทรงกลมเท่านั้น ที่ค่าความเบี่ยงเบนมาตรฐานมีค่าประมาณเท่ากับ 0 (3.967E−301) ในกรณีของเอบีซีแบบดั้งเดิม ในบางครั้งอัลกอริธึมไม่สามารถค้นพบค่าที่ดีที่สุดของฟังก์ชันของโรเซนบร็อกได้ เนื่องจากค่าเฉลี่ยของค่าที่เหมาะที่สุดมีค่ามากกว่า 0 มาก (0.681802) แต่อย่างไรก็ตาม การทำงานของฟังก์ชันทั้งสอง เป็นการเปรียบเทียบเพื่อให้เห็นความแตกต่างในเชิงการเทียบเคียงสมรรถนะเท่านั้น แต่สำหรับการทำงานเพื่อแก้ปัญหาจริง ผลที่ได้มีความแตกต่างกันเล็กน้อยมากจนแทบไม่เห็นนัยยะสำคัญ นอกจากกรณีของฟังก์ชันของโรเซนบร็อกเท่านั้น

การเทียงเคียงนี้ เป็นการแสดงให้เห็นผลของการทดลองระหว่างอัลกอริธึมการหาค่าเหมาะสุดแบบฝูงผึ้งแบบเดิม และเบสโซฟา เอบีซี ที่ถูกคิดขึ้นมาใหม่ ในตารางที่ 7.2 แม้ว่า การหาค่าเหมาะที่สุดจะประสบความสำเร็จทั้งหมด แต่ในแง่ของการเทียบเคียง เบสโซฟา เอบีซี จะดีกว่าในทุกด้าน ไม่ว่าจะเป็นในเรื่องของความเร็วในการลู่เข้าสู่คำตอบ และคำตอบที่ได้ ยกเว้นความเร็วของการทำงาน

7.4 ตัวอย่างการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์

เพื่อที่จะทำความเข้าใจการทำงานของวิธีการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์หรือ เอบีซี ในที่นี้จะแสดงตัวอย่างการทำงานที่แสดงให้เห็นถึงขั้นตอนการคำนวณทีละขั้น 3 ตัวอย่าง ตัวอย่างแรกเป็นการหาค่าเหมาะที่สุดของ ฟังก์ชัน $f(x) = x_1^2 + x_2^2$ ตัวอย่างที่สองเป็นเรื่องการเดินทางของเซลส์แมน และตัวอย่างที่สามเป็นตัวอย่างของการจัดตารางการผลิตด้วยเบสโซฟา เอบีซี

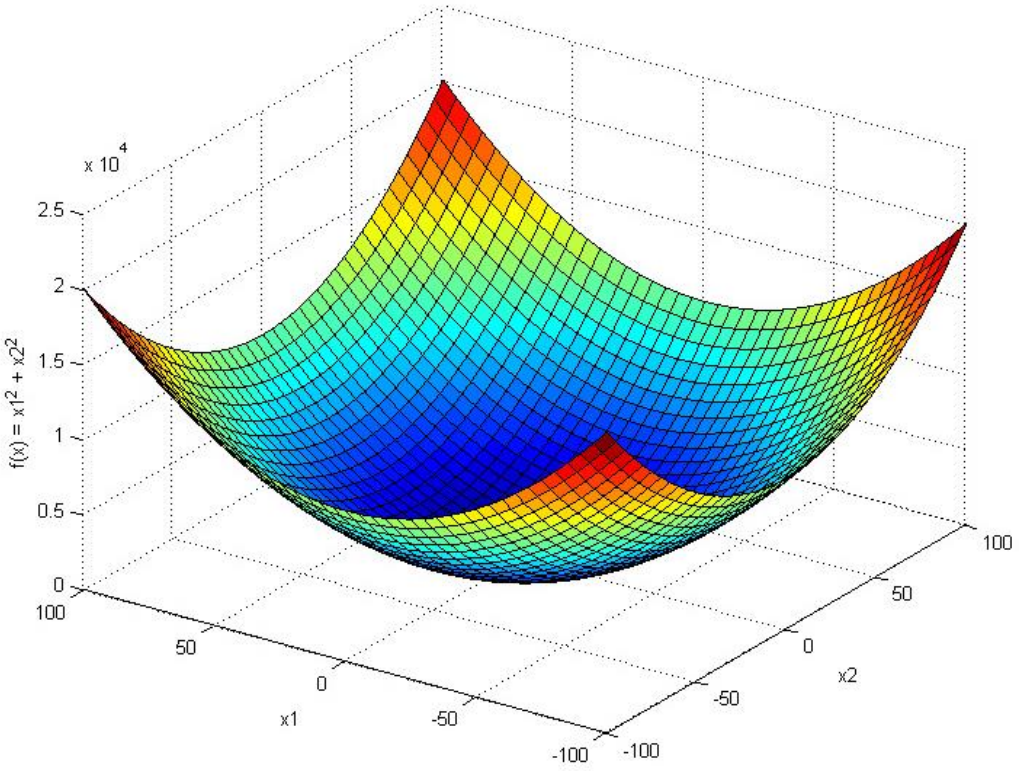
7.4.1 การหาค่าเหมาะที่สุดของ ฟังก์ชัน $f(x) = x_1^2 + x_2^2$

ตัวอย่างนี้เป็น การหาค่าเหมาะที่สุดของฟังก์ชัน $f(x) = x_1^2 + x_2^2$ ซึ่งเป็นการหาค่าเหมาะที่สุดเชิงตัวเลข โดยที่เรากำหนดขอบเขตของค่าตัวแปร x ไว้เท่ากับ $-3 \le x_1, x_2 \le 3$ ด้วยการพิจารณาจากฟังก์ชัน $f(x)$ เราสามารถทราบได้โดยทันทีว่าค่าที่เหมาะที่สุดของ x_1 และ x_2 คือ 0 ดังแสดงไว้ตามรูปที่ 7.5 การแก้ปัญหานี้ มีจุดประสงค์เพียงเพื่อทำความเข้าใจการทำงานของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ เพื่อทำให้การทำงานมีขอบเขตที่แน่นอน ในการคำนวณนี้จะกำหนดขอบเขตของพารามิเตอร์เริ่มต้นไว้ดังนี้

จำนวนของประชากรผึ้ง (Colony Size หรือ CS) = 6

จำนวนมากที่สุดของผึ้งค้นหา = 6

จำนวนมิติของปัญหา (ในที่นี้คือจำนวนตัวแปรในฟังก์ชัน: D) = 2



รูปที่ 7.5 กราฟของ $f(x) = x_1^2 + x_2^2$

เริ่มแรก เราสุ่มตำแหน่งของแหล่งอาหารขึ้นมาจำนวน 3 แหล่ง (จำนวนประชากรผึ้ง/จำนวนมิติ: CS/D) ซึ่งค่านี้ถือเป็นค่า SN สำหรับผึ้งเอ็มพลอย ด้วยการสุ่มค่าระหว่าง -3 ถึง 3 ผลของการกำหนดค่าเริ่มต้นของ x_1 และ x_2 ออกมาได้ดังนี้

x_{ij}	j = 0	j = 1
i = 0	1.3141	-3.4141
i = 1	1.1362	1.4324
i = 2	0.1021	-0.6091

ค่าที่ได้ออกมานี้เป็นค่าที่แสดงตำแหน่งอาหารที่ผึ้งเอ็มพลอยไปพบมา ในการประเมินคุณค่าของอาหารที่ได้จากแหล่งอาหารนี้ เราจะใช้การคำนวณหาค่าของแหล่งอาหารจากฟังก์ชัน

$f(x) = x_1^2 + x_2^2$ และหาค่าฟิตเน็สจาก

fitness = 1/(1+f(x)) ถ้า $f(x) \geq 1$ และ

fitness = 1/abs(f(x)) ถ้า $f(x) < 0$

ผลการคำนวณออกมาได้ดังนี้

x_{ij}	j = 0	j = 1	f(x)	Fitness
i = 0	1.3141	-3.4141	13.3829	0.0695
i = 1	1.1362	1.4324	3.3427	0.2303
i = 2	0.1021	-0.6091	0.3814	0.7239

เมื่อผึ้งเอ็มพลอยพบแหล่งอาหาร มันจะทำการเปรียบเทียบกับแหล่งอาหารข้างเคียง ด้วยการเลือกแบบสุ่ม สำหรับผึ้งเอ็มพลอยตัวแรก ค่าของ i = 0 ให้ x_{kj} เป็นแหล่งอาหารข้างเคียง (ค่าของ k เป็นค่าที่สุ่มมาจากค่าของ i) สุ่มค่า k=1 และ j=0 ดังนั้น $x_{ij} = x_{00} = 1.3141$ และ $x_{kj} = x_{10} = 1.1362$ หาค่า v_{ij} จากสมการดังนี้

$v_{ij} = x_{ij} + rand()(x_{ij} - x_{kj})$

$v_{00} = x_{00} + rand()(x_{00} - x_{10})$

$v_{00} = 1.3141 + 0.8050(1.3141-1.1362) = 1.4573$

แหล่งอาหารข้างเคียง v_{00} คือ 1.4573 -3.4141 มีค่า f(x) = 13.7798 ค่าฟิตเน็สเท่ากับ 0.0677 เมื่อเปรียบเทียบกับตำแหน่งเดิม 1.3141 -3.4141 ที่มีค่าฟิตเน็สเท่ากับ 0.0695 ปรากฏว่าค่าฟิตเน็สของแหล่งอาหารใหม่แย่กว่า ดังนั้นให้เลือกตำแหน่งเดิม

ผึ้งเอ็มพลอยตัวที่ 2 ค่าของ i = 1 ให้ x_{kj} เป็นแหล่งอาหารข้างเคียง สุ่มค่า k=2 และ j=1 ดังนั้น $x_{ij} = x_{11} = 1.4324$ และ $x_{kj} = x_{21} = -0.6091$ หาค่า v_{ij} จากสมการดังนี้

$v_{ij} = x_{ij} + rand()(x_{ij} - x_{kj})$

$v_{11} = x_{11} + rand()(x_{11} - x_{21})$

$v_{11} = 1.4324 + 0.0762 (1.4324+0.6091) = 1.5880$

ดังนั้นตำแหน่งของแหล่งอาหารข้างเคียง v_{11} คือ 1.1362 1.5880 มีค่า f(x) = 3.8127 ค่าฟิตเน็สเท่ากับ 0.2078 เมื่อเปรียบเทียบกับตำแหน่งเดิม 1.1362 1.4324 ที่มีค่าฟิตเน็สเท่ากับ 0.2303 ปรากฏว่าค่าฟิตเน็สของแหล่งอาหารใหม่แย่กว่า ดังนั้นให้เลือกตำแหน่งเดิม

ผึ้งเอ็มพลอยตัวที่ 3 ค่าของ i = 2 ให้ x_{kj} เป็นแหล่งอาหารข้างเคียง สุ่มค่า k=0 และ j=0 ดังนั้น $x_{ij} = x_{20} = 0.1021$ และ $x_{kj} = x_{00} = 1.3141$ หาค่า v_{ij} จากสมการดังนี้

$v_{ij} = x_{ij} + rand()(x_{ij} - x_{kj})$

$v_{20} = x_{20} + rand()(x_{20} - x_{00})$

$v_{20} = 0.1021+ 0.0101 (0.1021-1.3141) = 0.0899$

ตำแหน่งของแหล่งอาหารข้างเคียง v_{20} คือ 0.0899 -0.6091 มีค่า f(x) = 0.3791 ค่าฟิตเน็สเท่ากับ 0.7251 เมื่อเปรียบเทียบกับตำแหน่งเดิม 0.1021 -0.6091 ที่มีค่าฟิตเน็สเท่ากับ 0.7239

ปรากฏว่าค่าฟิตเน็สของแหล่งอาหารใหม่ดีกว่า ดังนั้นให้เลือกตำแหน่งใหม่ เมื่อทำการปรับปรุงค่าของตำแหน่งอาหารทั้งหมดจะได้ดังนี้

x_{ij}	j = 0	j = 1	f(x)	fitness
i = 0	1.3141	-3.4141	13.3829	0.0695
i = 1	1.1362	1.4324	3.3427	0.2303
i = 2	0.0899	-0.6091	0.3791	0.7251

หาค่าความน่าจะเป็นของคำตอบต่างๆ จากสมการ

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}$$

ผลของการคำนวณค่าความน่าจะเป็นออกมาได้ดังนี้

$p_1 = 0.0678$

$p_2 = 0.2247$

$p_3 = 0.7075$

ค่าความน่าจะเป็นนี้ เป็นค่าที่แสดงโอกาสของตำแหน่งแหล่งอาหารของผึ้งเอ็มพลอยที่จะถูกเลือกโดยผึ้งรับสาร โดยที่ p_1 หมายถึงโอกาสที่ตำแหน่งของแหล่งอาหารที่ผึ้งเอ็มพลอยตัวแรกหามา จะได้รับการเลือกจากผึ้งรับสาร สำหรับ p_2 หมายถึงโอกาสที่ตำแหน่งของแหล่งอาหารที่ผึ้งเอ็มพลอยตัวที่สองหามา จะได้รับการเลือกจากผึ้งรับสาร และเช่นกันสำหรับ p_3

จากความน่าจะเป็นที่คำนวณออกมาข้างต้น ค่าความน่าจะเป็น p_3 มีค่าสูงสุด ผึ้งรับสารเลือกตำแหน่งที่สาม หรือ i = 2 และหาตำแหน่งข้างเคียง เพื่อการปรับปรุงตำแหน่งอาหารใหม่จากการคำนวณโดยที่ i = 2 สำหรับค่าสุ่มได้ k = 1 และ j = 1 ดังต่อไปนี้

$v_{ij} = x_{ij} + rand()(x_{ij} - x_{kj})$

$v_{21} = x_{21} + rand()(x_{21} - x_{11})$

v_{21} = (-0.6091)+ 0.4946 ((-0.6091)-1.4324) = -1.6188

ตำแหน่งใหม่คือ 0.0899 -1.6188 ค่าของฟังก์ชันเท่ากับ 2.6287 มีค่าฟิตเน็สเท่ากับ 0.2756 เมื่อเทียบกับค่าฟิตเน็สเดิมของตำแหน่งเดิม 0.0899 -0.6091 มีค่าฟิตเน็สเท่ากับ 0.7251 แล้วมีค่าน้อยกว่า ดังนั้นให้เลือกตำแหน่งเดิม

หาตำแหน่งอาหารใหม่จากตำแหน่งของผึ้งรับสารตัวที่สองโดยที่ i = 1 เพราะตำแหน่งของผึ้งเอ็มพลอยตัวที่สองมีความน่าจะเป็นในลำดับที่สอง สุ่มค่า k = 0 และ j = 0 สุ่มค่า rand() = 0.5051 ได้ผลออกมาดังต่อไปนี้

$v_{ij} = x_{ij} + rand()(x_{ij} - x_{kj})$

$v_{10} = x_{10} + rand()(x_{10} - x_{00})$

v_{10} = (1.1362+ 0.5051* (1.1362-1.3141)) = 1.0463

ตำแหน่งใหม่คือ 1.0463 1.4324 ค่าของฟังก์ชันเท่ากับ 3.1466 มีค่าฟิตเน็สเท่ากับ 0.2412 เมื่อเทียบกับค่าฟิตเน็สเดิมของตำแหน่งเดิม 1.1362 1.4324 มีค่าฟิตเน็สเท่ากับ 0.2303 แล้วมีค่าดีกว่า ดังนั้นให้เลือกตำแหน่งใหม่ ผลของการปรับปรุงตำแหน่งทำให้ตำแหน่งของแหล่งอาหารใหม่ (คำตอบที่เป็นไปได้) ชุดใหม่มีดังนี้

x_{ij}	j = 0	j = 1	f(x)	fitness
i = 0	1.3141	-3.4141	13.3829	0.0695
i = 1	1.0463	1.4324	3.1466	0.2412
i = 2	0.0899	-0.6091	0.3791	0.7251

ในรอบถัดไป ให้ทดลองที่ i = 2 ที่มิติใหม่ที่ยังไม่ได้ทดลอง ในที่นี้คือ j = 0 (ครั้งที่แล้วทดลองที่ j = 1) สุ่มค่า k = 0 และค่า rand() = 0.0412 หาค่า v_{ij} ใหม่ดังนี้

$v_{ij} = x_{ij} + rand()(x_{ij} - x_{kj})$

$v_{20} = x_{20} + rand()(x_{20} - x_{00})$

v_{10} = (0.0899+ 0.0412* (0.0899-1.3141)) = 0.0394

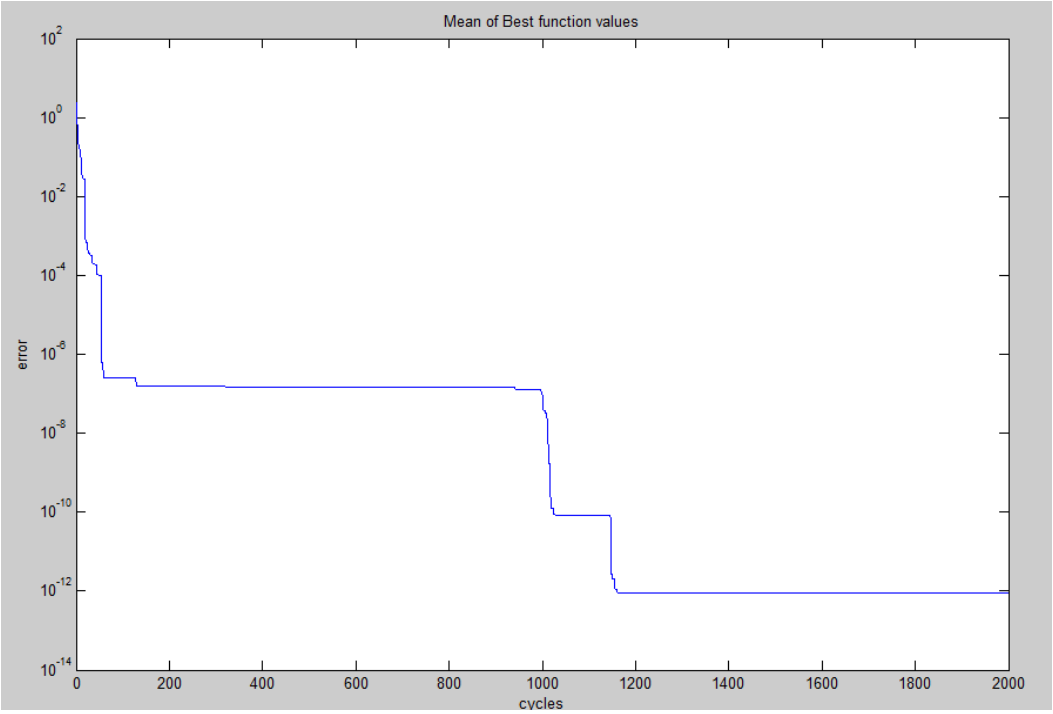
หาค่าของฟังก์ชันของตำแหน่งใหม่ ($f(x) = x_1^2 + x_2^2$) ได้เท่ากับ 0.3726 และค่าฟิตเน็สได้เท่ากับ 0.7286 เมื่อทำการเปรียบเทียบฟิตเน็สของตำแหน่งอาหารใหม่ 0.7286 จะดีกว่าของตำแหน่งเดิม 0.7251 ทำการปรับปรุงตำแหน่งของแหล่งอาหารใหม่จะได้ดังนี้

x_{ij}	j = 0	j = 1	f(x)	fitness
i = 0	1.3141	-3.4141	13.3829	0.0695
i = 1	1.0463	1.4324	3.1466	0.2412
i = 2	0.0394	-0.6091	0.3726	0.7286

ทำการบันทึกตำแหน่งที่ดีที่สุดคือ 0.0394 -0.6091 และบันทึกค่าของตำแหน่งที่ยังไม่เคยถูกปรับปรุงคือ x_{0j} = 1 ครั้ง

เริ่มรอบที่สอง การทำงานแบบเดียวกับการทำงานที่ผ่านมา โดยการหาค่าความน่าจะเป็นของตำแหน่งอาหาร x_{ij} ออกมาเป็น p_1 = 0.0669 p_2 = 0.2321 และ p_3 = 0.7010 ค่าของตำแหน่งที่ดีที่สุดคือ p_3 หรือ i =2 แล้วทำการสุ่ม k และ j ทำการหาตำแหน่งข้างเคียง เปรียบเทียบตำแหน่งของแหล่งอาหารของผึ้งทุกตัว ถ้าได้ตำแหน่งใหม่ที่ดีกว่า ให้นำเข้าไปแทนที่ จากนั้นหาค่าความน่าจะเป็น p_1 p_2 และ p_3 ของการที่แหล่งอาหารจะถูกเลือกอีก ให้ดำเนินการเช่นนี้ไปเรื่อยๆ จนกว่าจะไม่มีการเปลี่ยนแปลงเกิดขึ้น หรือจนกว่าจะได้ตำแหน่งที่เราพอใจ

สำหรับการบันทึกตำแหน่งที่ยังไม่เคยถูกปรับปรุงนี้ ก็เพื่อการตรวจสอบดูว่าตำแหน่งดังกล่าวยังเป็นตำแหน่งที่ดีอยู่หรือไม่ ในกรณีที่ตำแหน่งใด ไม่ได้รับการปรับปรุงมาเป็นเวลานาน เช่นการทำงานของผึ้งทำมา 6 รอบแล้วยังไม่ได้มีการปรับปรุง ผึ้งค้นหาก็จะออกสุ่มหาตำแหน่งใหม่ เพื่อมาใช้แทนตำแหน่งที่ยังไม่มีการปรับปรุงมาเป็นเวลานานนี้ สำหรับการใช้ 6 รอบนี้ พิจารณาจากจำนวนของตำแหน่งของแหล่งอาหาร (3 ตำแหน่ง) คูณด้วยจำนวนของมิติ (2 มิติ) ซึ่งเป็นจำนวนที่สูงที่สุด สำหรับการลู่เข้าสู่คำตอบของการหาคำตอบที่ดีที่สุดนั้นมีลักษณะดังแสดงตามรูปที่ 7.6 กราฟแสดงค่าของ *f(x) = x12 + x22* ที่เปลี่ยนไปเมื่อจำนวนรอบของการทำงานมากขึ้น



รูปที่ 7.6 กราฟแสดงค่าของ $f(x) = x_1^2 + x_2^2$ ที่เปลี่ยนไปเมื่อจำนวนรอบของการทำงานมากขึ้น

7.4.2 การเดินทางของเซลส์แมน

ตัวอย่างของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์เพื่อแก้ปัญหาการเดินทางของเซลส์แมนนี้นำมาจากงานวิจัยของอนันต์ ธีรณี และบุญเจริญ (Banharnsakun, 2010) ที่เป็นการนำ GA มาใช้ร่วมกับ ABC โดยที่กำหนดให้เส้นทางเดินถูกแสดงไว้ในรูปแบบของสตริงดังนี้

การแทนปัญหา ที่อยู่ในรูปของคำตอบจะแสดงลำดับของเมืองที่เซลส์แมนจะต้องเดินทาง ในที่นี้กำหนดให้อยู่ในรูปของเมืองที่เรียงลำดับกันในวงเล็บ ดังนี้

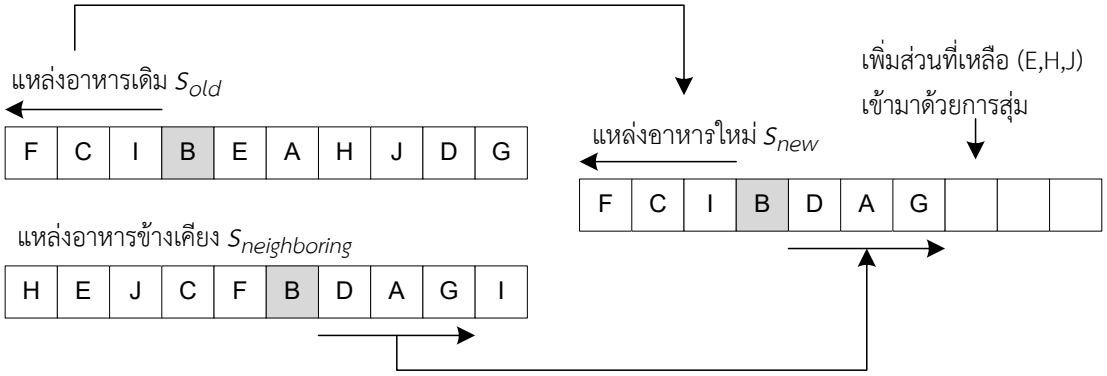
$(x_1, x_2, x_3, \ldots x_n)$

เมื่อ x_i หมายถึงชื่อเมือง ที่มีทั้งหมด n เมือง และการเดินทางจะเริ่มจากเมือง x_1, x_2 ไปจนถึง x_n ซึ่งเป็นเมืองสุดท้าย หรืออาจจะเขียนเป็น

$S = (x_i)$, $i = 1 \ldots n$

เมื่อ S หมายถึงเซทของเส้นทางทั้งหมดที่เป็นไปได้ในการเดินทางของเซลส์แมน

วิธีการ กรีดี ซับทัวร์ ครอสโอเวอร์ (Greedy Subtour Crossover Method: GSX) วิธีการ กรีดี ซับทัวร์ ครอสโอเวอร์ หรือ จีเอสเอ็กซ์ (GSX) เป็นวิธีที่ใช้ในการสร้างเส้นทางเลือกใหม่สำหรับการเดินทาง โดยอาศัยเส้นทางเดิม 2 เส้นทางมาทำการครอสโอเวอร์กัน ให้เกิดเส้นทางใหม่ ซึ่งเป็นกระบวนการเดียวกันกับอัลกอริธึมพันธุการ วิธีการมีดังนี้



รูปที่ 7.7 ตัวอย่างของวิธีการ กรีดี ซับทัวร์ ครอสโอเวอร์

สมมุติว่า A, B, C, D, E, F, G, H, I, J เป็นชื่อเมืองต่างๆ ที่เซลส์แมนจะต้องเดินทาง และเรามีเส้นทาง 2 เส้นดังนี้ s_{old} = {F,C,I,B,E,A,H,J,D,G} และ $s_{neighboring}$ = {H,E,J,C,F,B,D,A,G,I} วิธีการครอสโอเวอร์ เริ่มต้นด้วยการเลือกเมืองขึ้นมาเมืองหนึ่งด้วยการสุ่ม เพื่อใช้สำหรับการสร้างเส้นทางใหม่ s_{new}(p) สมมุติว่าเป็น B จากการสำรวจในเส้นทางแรก B เป็นเมืองที่ 4 และในเส้นทางที่ 2 เมือง B เป็นเมืองที่ 6 หรือเขียนเป็น s_{old} (p=4) = B และ $s_{neighbouring}$(p=6) = B และเส้นทางที่จะสร้างใหม่เริ่มต้นด้วย s_{new}(p=4) = {B}

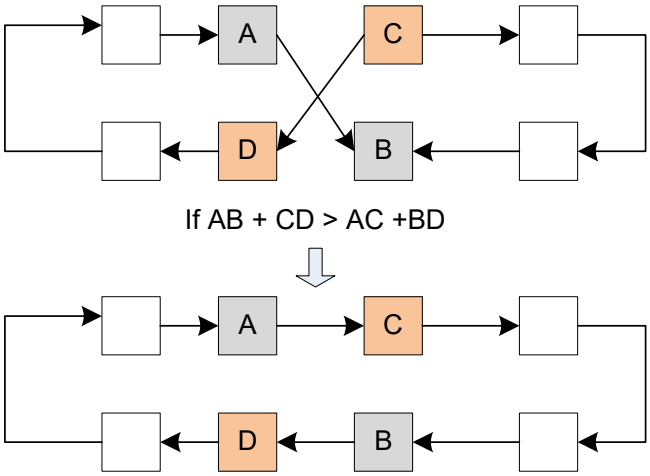
ขั้นตอนต่อไปให้ลด p ของ s_{old}(p-1=3) = I แล้วนำไปไว้หน้าของ s_{new} = {I,B} และเพิ่ม p ของ $s_{neighbouring}$(p+1=7) = D แล้วนำไปไว้หลังของ s_{new} = {I,B,D} ทำเช่นนี้ไปเรื่อยๆ เราจะได้รายการของเมืองต่างๆ ในรายการของ s_{old} ที่เพิ่มส่วนหน้าของ s_{new} คือ s_{old}(p=2) = C และ s_{old}(p=1) = F สำหรับ

รายการของ $s_{neighbouring}$ ที่เพิ่มเข้าไปส่วนหลังของ s_{new} คือ $s_{neighbouring}(p=8)$ = A และ $s_{neighbouring}(p=9)$ = G ทำให้ได้ s_{new} ใหม่คือ $s_{new}(p)$ = {F,C,I,B,D,A,G}

สำหรับเมืองถัดไปที่จะต้องนำมาเพิ่มใน s_{new} ของ s_{old} คือ G แต่ G มีอยู่แล้วใน s_{new} จึงไม่ต้องเพิ่มเข้าไปอีก สำหรับเมืองถัดไปของ $s_{neighbouring}$ ที่ต้องนำเข้าคือ I ซึ่งเมือง I นี้ก็มีอยู่แล้วใน s_{new} เช่นกัน

สำหรับเมืองที่เหลือคือ E, H, และ J ให้นำเข้าไปไว้ใน s_{new} ด้วยการสุ่ม ทำให้เราได้ s_{new} ใหม่ทั้งหมดคือ {F,C,I,B,D,A,G,H,E,J} เมื่อการปรับปรุงค่านี้เสร็จสิ้น กระบวนการ 20pt ก็จะถูกนำมาใช้เพื่อการปรับปรุงค่า s_{new} อีกครั้ง

วิธีการ ทูอ็อป (2opt method) วิธีการทูอ็อป เป็นการปรับเส้นทางเดินของเซลส์แมนเบื้องต้น เพื่อให้ได้เส้นทางที่ดีขึ้น (สั้นลง) โดยการเปรียบเทียบ ระยะทางของเส้นทางคู่หนึ่ง จากรูปที่ 7.8 ถ้าระยะทางของเมือง A ไป B และ C ไป D มากกว่าระยะทางรวมจาก A ไป C และ B ไป D แล้วให้ย้ายเส้นทางเดิน ที่เดินจากเมือง A ไป B และ C ไป D เป็นจากเมือง A ไป C และ B ไป D แทน ซึ่งจะทำให้เกิดเส้นทางใหม่ดังรูปที่ 7.8 ภาพล่าง



รูปที่ 7.8 วิธีการ ทูอ็อป

ในการหาแหล่งอาหารใหม่ของผึ้งเอ็มพลอย ถ้าแหล่งอาหารใหม่ x_{new} มีค่าฟิตเน็ส $f(x)$ ดีกว่าแหล่งอาหารเดิม x_{old} ให้แทนค่าแหล่งอาหารใหม่นี้ลงในหน่วยความจำของผึ้งเอ็มพลอย

ขั้นตอนที่สาม ผึ้งเอ็มพลอยจะแลกเปลี่ยนข่าวสารของเส้นทางเดินใหม่นี้กับผึ้งรับสาร และโอกาสของผึ้งรับสารนี้ก็จะทำการเลือกเส้นทางที่ดีที่สุด (ค่าฟิตเน็สสูงสุด) จากเส้นทางทั้งหมดที่ได้รับข่าวสารจากกลุ่มผึ้งเอ็มพลอย ก็จะมีสูงกว่า และผึ้งรับสารนี้ก็จะทำการปรับปรุงเส้นทางให้ดีขึ้นด้วยกระบวนการ GSX และ 2pot อีก

ขั้นตอนสุดท้าย ถ้าเส้นทางเดินทั้งหมดที่ผ่านกระบวนการปรับปรุงดังกล่าวข้างต้นแล้ว ค่าฟิตเน็สไม่ดีขึ้น (หรือระยะทางไม่สั้นลง) กระบวนการทั้งหมดก็ยุติ และเส้นทางเดินใหม่ที่ได้จากผึ้งค้นหาก็จะถูกมาใช้ กระบวนการทั้งหมดดังกล่าวข้างต้นจะดำเนินการต่อไปเรื่อยๆ จนกระทั่งจำนวนรอบของการทำงานเท่ากับ MNC

วิธีการแก้ปัญหา รายละเอียดของวิธีการแก้ปัญหาการเดินทางของเซลส์แมนจะเป็นดังนี้

ขั้นแรก สร้างคำตอบเริ่มต้น ด้วยการสุ่มขึ้นมา และให้เป็นคำตอบเริ่มต้นที่อยู่ในหน่วยความจำของผึ้งเอ็มพลอย ซึ่งเปรียบเสมือนหนึ่ง เป็นข้อมูลของแหล่งอาหารที่ผึ้งเอ็มพลอยไปหามาได้ หลังจากนั้นผึ้งทั้งหมด ก็จะเริ่มทำการเลือกแหล่งอาหารที่ดีที่สุดด้วยการทำงานวนรอบกระบวนการ 3 ขั้นตอน คือ การปรับปรุงคำตอบที่เป็นไปได้ การเลือกคำตอบที่เป็นไปได้ และการหลีกเลี่ยงคำตอบที่เข้าสู่ทางตัน

Procedure Travelling Saleman Problem
Create the initial parameters
Create initial solution
While not(MCN) เมื่อยังไม่ถึงจำนวนรอบสูงสุด
UpdateFeasibleSolutions(Employed Bees) ปรับปรุงคำตอบที่เป็นไปได้ของผึ้งเอ็มพลอย
EvaluateFeasibleSolaution(Employed Bees)
SelectFeasibleSolutions(Onlooker Bees) เลือกคำตอบที่เป็นไปได้ของผึ้งรับสาร
UpdateFeasibleSolutions(Onlooker Bees) ปรับปรุงคำตอบที่เป็นไปได้ของผึ้งรับสาร
EvaluateFeasibleSolaution(Onlooker Bees)
While Solution is abandoned

```
        AvoidSub-OptimalSolutions(Scout Bees) ผึ้งค้นหาหาแหล่งอาหารใหม่
        End while
        End while
End-Procedure
```

รูปที่ 7.9 ผังงานของการหาเส้นทางที่ดีที่สุดด้วย เอบีซี

การปรับปรุงคำตอบที่เป็นไปได้ ในขั้นตอนนี้ผึ้งเอ็มพลอยจะเปรียบเทียบคุณภาพ (ค่าฟิตเน็ส *f(x)*) ของแหล่งอาหาร (หรือระยะทางของเส้นทาง) กับแหล่งอาหารข้างเคียง (ระยะทางของเส้นทางที่สร้างขึ้นด้วย GXS และ 2opt) แหล่งอาหารที่มีคุณภาพดีที่สุด (เส้นทางที่สั้นที่สุด) จะถูกเลือก

การเลือกคำตอบที่เป็นไปได้ ขั้นตอนนี้ผึ้งรับสารจะเลือกเส้นทางเส้นหนึ่ง จากเส้นทางทั้งหมด ซึ่งความน่าจะเป็นของเส้นทางที่จะถูกเลือกจะมีตามสัดส่วนตามปริมาณค่าฟิตเน็ส เมื่อได้คำตอบจากการเลือกแล้ว ผึ้งรับสารก็จะทำการปรับปรุงเส้นทางด้วยวิธีการเดียวกันกับผึ้งเอ็มพลอย

การหลีกเลี่ยงคำตอบที่เข้าสู่ทางตัน เมื่อกระบวนการดำเนินการข้างต้นดำเนินการไปจนพบว่าคำตอบที่ได้ไม่ดีขึ้นไปกว่าเดิมอีก ไม่ว่าจะดำเนินการตามกระบวนการดังกล่าวซ้ำกี่ครั้งก็ตาม เส้นทางที่ถูกค้นหาใหม่โดยผึ้งค้นหาจะถูกนำมาใช้แทนเส้นทางเดิม ซึ่งกระบวนการทั้งหมดของการทำงานสามารถแสดงออกมาเป็นผังงานดังรูปที่ 7.9

7.4.3 การจัดตารางการผลิตด้วยเบสโซฟา เอบีซี

การประยุกต์เบสโซฟา เอบีซี กับการจัดตารางการผลิตสำหรับการผลิตแบบทำตามสั่ง (Job Shop Scheduling) เป็นกระบวนการในการจัดลำดับในการทำงานของเครื่องจักรที่มีหลายเครื่อง ที่ใช้ในการผลิตชิ้นงานที่เหมือนกันจำนวนหลายชิ้น ชิ้นงานแต่ละชิ้นมีงานที่เครื่องจักรทุกตัวต้องทำเป็นส่วนๆ เครื่องจักรแต่ละตัวใช้เวลาในการทำงานแต่ละส่วนของชิ้นงานไม่เท่ากัน การแก้ปัญหาของการจัดตารางนี้ก็คือ การจัดตารางการทำงานของเครื่องจักรแต่ละตัวในการผลิตชิ้นงานทั้งหมดเพื่อที่จะทำให้เวลาในการผลิตสั้นที่สุด

เฟรนช์ (French, 1982) อธิบายปัญหาการจัดตารางการผลิตสำหรับการผลิตแบบทำตามสั่ง (Job shop scheduling problem: JSSP) ในรูปของเซทของชิ้นงานจำนวน n ชิ้น ที่เขียนแทนด้วยตัวแปร J_j เมื่อ $j = 1,2,\ldots, n$ และชิ้นงานเหล่านี้ถูกผลิตด้วยเครื่องจักรจำนวน m เครื่อง ที่เขียนแทนด้วยตัวแปร M_k เมื่อ $k = 1,2,\ldots,m.$ การทำงานของชิ้นงานที่ j^{th} ด้วยเครื่องจักรเครื่องที่ k^{th} จะเขียนแทนด้วยสัญลักษณ์ O_{jk} ที่ใช้เวลาสำหรับการทำงานเท่ากับ P_{jk} งานแต่ละชิ้นจะต้องใช้เครื่องจักรทุกตัวในการทำงาน โดยที่ลำดับของเครื่องจักรที่ใช้ในการทำงานแต่ละชิ้นจะแตกต่างกันไป เวลาที่ใช้ในการผลิตงานแต่ละชิ้นก็แตกต่างกันไปด้วย เครื่องจักรแต่ละเครื่องจะทำงานได้ทีละงาน และเมื่อเครื่องจักรเริ่มทำงานเครื่องจักรจะหยุดไม่ได้ และเวลาทั้งหมดที่ใช้ในทำงานเพื่อผลิตชิ้นงานจะเรียกว่า ช่วงกว้างของเวลาการทำงาน (Makespan) จุดประสงค์ของการจัดตารางการผลิตสำหรับการผลิตตามคำสั่งก็คือการทำให้ช่วงกว้างของเวลาการทำงานสั้นที่สุด ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$\text{Min } \square_{\square\square\square} = \max(\square_1, \square_2, \square_3, \ldots \square_{\square}) \qquad \text{สมการ 7.7}$$

$$\square_{\square} = \square_{\square} + \sum_{\square=1}^{\square}(\square_{\square\square} + \square_{\square\square(\square)}) \qquad \text{สมการ 7.8}$$

เมื่อ C_j เวลาที่ใช้ในการผลิตชิ้นงาน j

r_j เป็นเวลาปล่อย (release time) ชิ้นงาน j

w_{jk} เวลาที่ชิ้นงาน j ที่รอที่ลำดับที่ k

$p_{jm(k)}$ เป็นเวลาที่ใช้ในการผลิตชิ้นงาน j ด้วยเครื่องจักร m ที่ลำดับ k

ความยากของการหาคำตอบของการจัดตารางการผลิตสำหรับการผลิตแบบทำตามสั่งนี้ก็คือจำนวนคำตอบที่เป็นไปได้ สำหรับปัญหาที่มีขนาดใหญ่ จำนวนคำตอบที่เป็นไปได้ทั้งหมดเท่ากับ $(n!)^m$ ซึ่งเป็นจำนวนที่มากจนแทบเป็นไปไม่ได้ในการหาคำตอบที่ดีที่สุด ในการแก้ปัญหานี้ นำเสนอโดยอนันต์ บุญเจริญ และธีรณี (Banharnsakun, 2012) เป้าหมายของการแก้ปัญหานี้ก็คือการหาค่า ช่วงกว้างของเวลาการทำงานที่เหมาะที่สุด สำหรับการแก้ปัญหาการจัดตารางการผลิตสำหรับการผลิตแบบทำตามสั่งเป็นไปตามรูปที่ 7.10

Procedure Job Shop Scheduling Problem

```
Create the initial parameters
Create initial solution
        While not(MCN)
        Update_Feasible_Solutions (Employed Bees)
        Evaluate_Feasible_Solaution(Employed Bees)
        Select_Feasible_Solutions (Onlooker Bees)
        Update_Feasible_Solutions (Onlooker Bees)
        Evaluate_Feasible_Solaution(Onlooker Bees)
        While Solution is abandoned
                Perform_solution_radious_search (Scout Bees)
                Perform_Best-so-far_Local_Search
        End while
        End while
End-Procedure
```

รูปที่ 7.10 การจัดตารางการผลิตสำหรับการผลิตแบบทำตามสั่งด้วย เบสโซฟา เอบีซี

การทำงานของอัลกอริธึมที่ใช้ในการแก้ปัญหาการจัดตารางการผลิตสำหรับการผลิตแบบทำตามคำสั่ง เริ่มด้วยการกำหนดค่าพารามิเตอร์เริ่มต้น และสร้างคำตอบเริ่มต้น จากนั้นให้ทำตามขั้นตอนต่อไปนี้จนกระทั่งการวนรอบถึงจำนวนสูงสุด (MCN) ก็ทำการสร้างคำตอบที่เป็นไปได้ด้วยการปรับปรุงคำตอบของผึ้งเอ็มพลอย และทำการคำนวณค่าของคำตอบแต่ละอัน และทำการเลือกคำตอบที่เป็นไปได้ของผึ้งรับสาร จากนั้นทำการปรับปรุงและประเมินคำตอบที่เป็นไปได้ของผึ้งรับสาร ถ้าหากว่าคำตอบที่ได้ไม่ได้รับการปรับปรุงให้ดีขึ้นเป็นเวลานาน (จำนวนหลายรอบของการทำงาน) ให้ผึ้งค้นหาทำการหาคำตอบอื่นมาเปรียบเทียบ จากนั้นก็ทำการสร้างคำตอบที่เป็นไปได้ ทำการปรับปรุงคำตอบและประเมินค่าของคำตอบที่เป็นไปได้ จนกระทั่งครบจำนวนรอบสูงสุด

สำหรับขั้นตอนการแก้ปัญหามีดังต่อไปนี้

ขั้นตอนที่หนึ่ง กำหนดค่าเริ่มต้นของพารามิเตอร์ดังนี้ จำนวนผึ้งเอ็มพลอย จำนวนผึ้งรับสาร จำนวนรอบสูงสุดของการทำงาน (MCN)

ถ้ากำหนดว่า ระยะเวลาของการทำงานของชิ้นงานของเครื่องจักรแต่ละเครื่องเป็นไปตามตารางที่ 7.3 และลำดับของเครื่องจักรที่ใช้สำหรับผลิตชิ้นงานเป็นไปตามตารางที่ 7.4

	M_1	M_2	M_3
J_1	15	25	18
J_2	7	10	20
J_3	12	10	8

ตารางที่ 7.3 ตัวอย่างการทำงานของเครื่องจักรแต่ละตัวที่ใช้เวลาในการทำงานแต่ละชิ้น

	O_1	O_2	O_3
J_1	M_1	M_2	M_3
J_2	M_2	M_3	M_1
J_3	M_3	M_1	M_2

ตารางที่ 7.4 ตัวอย่างของการจัดลำดับการทำงานของเครื่องจักรที่ใช้ในการทำงานแต่ละชิ้น

กำหนดว่า ลำดับการทำงานของงานแต่ละชิ้นจะมีลำดับเรียงกันเป็นแบบสตริงดังต่อไปนี้

J_1	J_2	J_3	J_1	J_2	J_3	J_2	J_1	J_3

รูปที่ 7.11 ตัวอย่างรายการของการทำงาน 3 ชิ้นกับเครื่องจักร 3 เครื่อง

เนื่องจากจำนวนงานทั้งหมดมี n งาน และแต่ละงานจะต้องใช้เครื่องจักรจำนวน m เครื่องในการผลิต ดังนั้นความยาวของสตริงที่จะแสดงผังการผลิตของงานทั้งหมดเท่ากับ n*m จากตัวอย่างดังกล่าวข้างต้น จำนวนงานมีทั้งหมด 3 ชิ้น (j_1, j_2 และ j_3) และเครื่องจักรทั้งหมดมี 3 เครื่อง เนื่องจากงานแต่ละชิ้นจะต้องผ่านเครื่องจักรทั้งหมด 3 เครื่อง ดังนั้นความยาวของตารางทั้งหมดจึงเท่ากับ 9 ตามรูปที่ 7.11 จะเห็นว่า งานแต่ละงานจะต้องมีปรากฏในสตริง 3 ครั้ง ซึ่งหมายถึง ต้องผ่านเครื่องจักร 3 เครื่อง

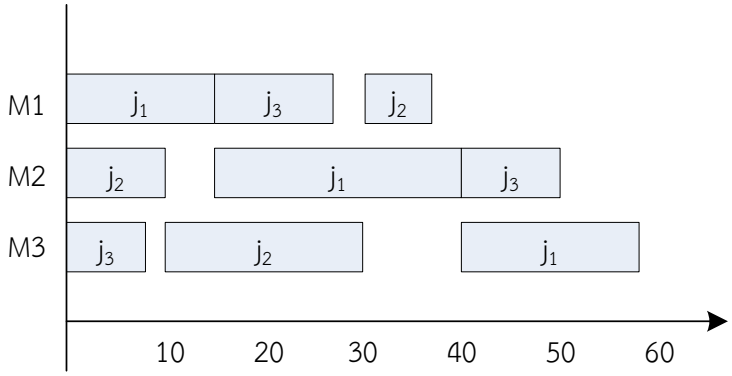
ตามรูปที่ 7.11 เป็นตัวอย่างของตารางการทำงานที่เป็นไปได้ที่เกิดจากงาน 3 ชิ้นผลิตโดยเครื่องจักร 3 เครื่อง ตารางนี้เป็นการแสดงลำดับของการผลิตงานทั้ง 3 ชิ้นว่างานแต่ชิ้นจะมีการผลิตก่อนหลังอย่างไร และเพื่อที่จะทราบว่าตารางการผลิตนี้จะใช้เวลาในการผลิตอย่างไร เราจะต้องนำตารางการผลิตที่เป็นสตริงนี้ไปเปรียบเทียบกับตารางที่ 7.3 และตารางที่ 7.4 เพื่อหาช่วงกว้างของเวลาการทำงาน (Make span) ซึ่งการหาช่วงกว้างของเวลาการทำงานจากรูปที่ 7.11 จะมีลำดับดังต่อไปนี้

ในช่องแรกของรูปที่ 7.11 งาน j_1 เป็นการผลิตครั้งที่ 1 (O_1) เมื่อพิจารณาจากตารางที่ 7.4 จะต้องใช้เครื่องจักรเครื่องที่ 1 (M_1) ตัวสตริงถัดมาเป็น j_2 เป็นการผลิตครั้งแรกเช่นกัน เมื่อเทียบกับตารางที่ 7.4 แล้วจะต้องใช้เครื่องจักร M_2 สตริงตัวที่ 3 เป็น j_3 เป็นการผลิตครั้งแรก เมื่อเทียบกับตารางที่ 7.4 จะต้องใช้เครื่องจักร M_3

สำหรับสตริงตัวที่ 4 เป็น j_1 ครั้งนี้จะเป็นการผลิตครั้งที่สอง (O_2) เนื่องจาก j_1 เคยผลิตมาก่อนหน้าแล้ว 1 ครั้ง เมื่อเทียบกับตารางที่ 7.4 แล้วจะต้องใช้เครื่องจักรเครื่องที่ 2 หรือ M_2 ถัดมาคือ j_2 เป็นการผลิตครั้งที่สอง (O_2) เมื่อเทียบกับตารางที่ 7.4 แล้วจะต้องใช้เครื่องจักร M_3 ถัดมาคือ j_3 เป็นการผลิตครั้งที่สอง (O_2) เมื่อเทียบกับตารางที่ 2 แล้วจะต้องใช้เครื่องจักรเครื่อง M_1

ถัดมาคือ j_2 เป็นการผลิตครั้งที่สาม (O_3) เมื่อเทียบกับตารางที่ 7.4 แล้วจะต้องใช้เครื่องจักร M_1 ถัดมาคือ j_1 เป็นการผลิตครั้งที่สาม (O_3) เมื่อเทียบกับตารางที่ 7.4 แล้วจะต้องใช้เครื่องจักรเครื่องที่ 3 หรือ M_3 และงานสุดท้ายคือ j_3 เป็นการผลิตครั้งที่สาม (O_3) จะต้องใช้เครื่องจักร M_2 เมื่อทำการเปรียบเทียบเสร็จจะได้ออกมาตามรูปที่ 7.12

ในการหาเวลาของการทำงานทั้งหมดหรือ ช่วงกว้างของเวลาการทำงานจะเท่ากับระยะเวลาของงานที่เสร็จหลังสุด จากรูปที่ 7.12 เมื่อเราวางงานชุดแรกลงบนตารางคือ j_1, j_2 และ j_3 แล้ว เวลาในการผลิตของ M1 จะต้องทำงานให้กับ j_1 เท่ากับ 15 นาที j_3 เท่ากับ 12 นาที และ j_2 เท่ากับ 7 นาที สำหรับเวลาในการผลิตของ M2 จะต้องทำงานให้กับ j_2 เท่ากับ 10 นาที j_1 เท่ากับ 25 นาที และ j_3 เท่ากับ 10 นาที และเวลาในการผลิตของ M3 จะต้องทำงานให้กับ j_3 เท่ากับ 8 นาที j_2 เท่ากับ 20 นาที และ j_1 เท่ากับ 18 นาที



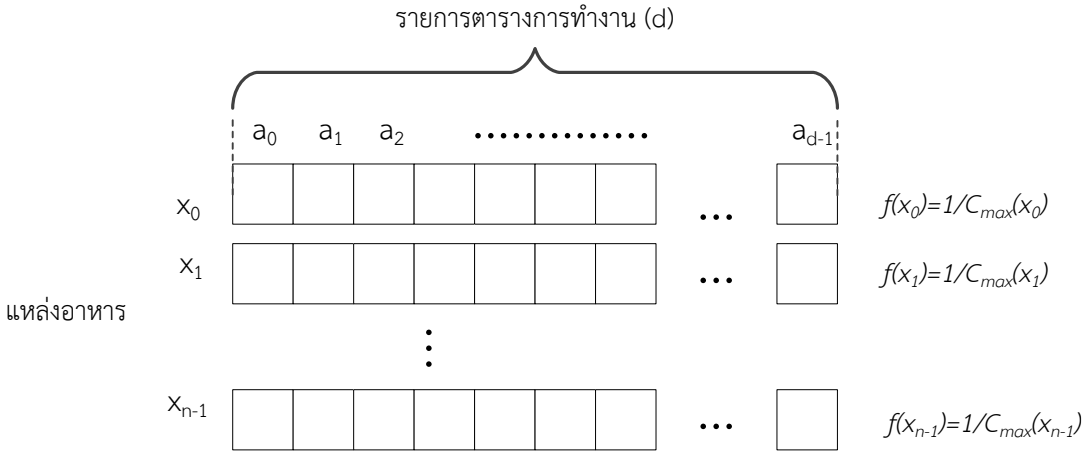
รูปที่ 7.12 ช่วงกว้างของเวลาการทำงานที่เกิดจากลำดับของงานตามรูปที่ 7.11

ในการทำงานสำหรับ j_1 ที่จะต้องใช้เครื่องจักร 3 เครื่องทำการผลิต เนื่องจาก M1 ผลิต j_1 ใช้เวลา 15 นาที ดังนั้น M2 จะเริ่มผลิต j_1 ได้ต้องนาทีที่ 15 เป็นต้นไป และเช่นกัน M2 ใช้เวลาในการผลิต j_1 25 นาที ดังนั้น M3 จะสามารถเริ่มผลิต j_1 ได้ที่นาทีที่ 15+25 = 40 เป็นต้นไป

สำหรับ j_2 ที่จะต้องใช้เครื่องจักร 3 เครื่องทำการผลิต แต่ M1 ผลิต j_2 ไม่ได้เนื่องจากทำการผลิต j_1 อยู่ ดังนั้น j_2 จึงต้องเริ่มทำการผลิตที่ M2 ใช้เวลา 10 นาที ที่เวลานี้ เครื่องจักรที่ว่างสำหรับการผลิต j_2 ต่อคือ M3 เท่านั้น M3 และจะเริ่มผลิต j_2 ได้ต้องนาทีที่ 10 เป็นต้นไป และใช้เวลา 20 นาที ดังนั้น M1 เริ่มผลิต j_2 ได้ที่นาทีที่ 10+20 = 30 เป็นต้นไป

สำหรับ j_3 ต้องใช้เครื่องจักร M3 ทำการผลิต เนื่องจาก M1 และ M2 ไม่ว่าง M3 จึงต้องทำหน้าที่ผลิต j_3 ก่อน ใช้เวลา 8 นาที ที่เวลานี้ เครื่องจักรที่ว่างสำหรับการผลิต j_3 ต่อยังไม่มี จะต้องรอจนถึงนาทีที่ 15 เครื่องจักร M1 จึงจะว่าง M1 ผลิต j_3 ใช้เวลา 10 นาที จะเสร็จที่นาทีที่ 15 + 10 = 25 ที่เวลานี้ เครื่องจักร M2 ยังไม่ว่างสำหรับการผลิต j_3 ต่อ เพราะยังทำการผลิต J_1 อยู่ ต้องรอจนกระทั่ง M2 ว่างที่เวลาที่ 40 จึงจะเริ่มทำการผลิต j_3 ได้ เมื่อคิดจำนวนเวลาสูงสุดในการผลิตงานทั้ง 3 ชิ้นนี้แล้วจะเท่ากับ 58 นาที

สำหรับค่าของฟิตเน็ส *f(x)* ของตารางจะเป็นสัดส่วนผกผันของช่วงกว้างของเวลาการทำงาน *1/c(x)* เมื่อ *c(x)* หมายถึงช่วงกว้างของเวลาการทำงานที่หาจากเวลาร่วมทั้งหมดตามรูปที่ 7.13

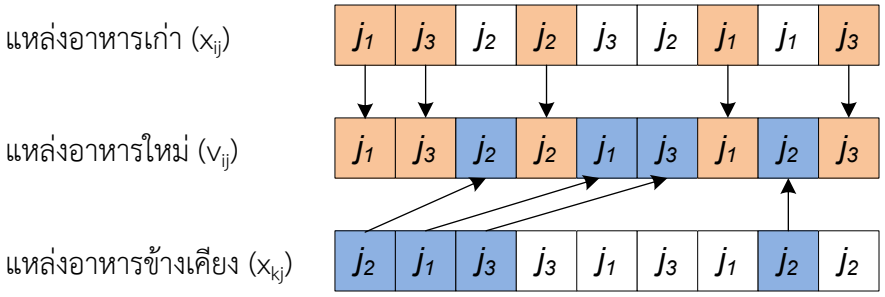


รูปที่ 7.13 การเปรียบเทียบระหว่างแหล่งอาหารและรายการของตารางการทำงาน

ขั้นตอนที่สอง เป็นการปรับปรุงตำแหน่งของแหล่งอาหารเมื่อเทียบกับแหล่งอาหารข้างเคียงตามสมการดังต่อไปนี้

$$\square_{\square\square} = \square_{\square\square} + \square(\square_{\square\square} - \square_{\square\square}) \qquad \text{สมการ 7.9}$$

สำหรับการปรับปรุงแหล่งอาหารหรือคำตอบที่เป็นไปได้นี้จะอาศัยเทคนิค โพซิชันเบสครอสโอเวอร์ (Position Base Crossover: PBX) (Kellegöz, 2008) ในการปรับปรุงแหล่งอาหารเก่า (x_{ij}) ของผึ้งเอ็มพลอยเป็นแหล่งอาหารใหม่ (v_{ij}) โดยการเปรียบเทียบกับแหล่งอาหารข้างเคียง (x_{kj}) ที่สุ่มมาจากผึ้งเอ็มพลอยตัวอื่นๆ เทคนิคของ PBX จะเป็นดังรูปที่ 7.14



รูปที่ 7.14 การแลกเปลี่ยนข้อมูลข่าวสารกับแหล่งอาหารข้างเคียงด้วยเทคนิค PBX

วิธีการของ PBX เป็นวิธีการที่อาศัยความน่าจะเป็น โดยการสุ่มเลือกการผลิตชิ้นงาน (j_i) จากแหล่งอาหารเก่า (x_{ij}) เพื่อใส่ลงในตารางของแหล่งอาหารใหม่ (v_{ij}) โดยที่ j_i แต่ตัวจะมีโอกาสของความน่าจะเป็นที่ถูกเลือกไปอยู่ในรายการของแหล่งอาหารใหม่เท่ากับ 0.5 และการผลิตชิ้นงาน j_i ที่ได้รับการเลือกแล้วจากแหล่งอาหารเก่า จะไม่ถูกเลือกจากแหล่งอาหารข้างเคียง x_{kj} จากนั้น j_i จากแหล่งอาหารข้างเคียงที่ยังไม่ได้รับการเลือกเลยจะถูกจัดให้อยู่ในตารางที่ว่างของแหล่งอาหารใหม่ โดยการเลือก j_i จากซ้ายไปขวาของแหล่งอาหารข้างเคียง และเพื่อป้องกันงาน j_i ที่จะถูกเครื่องจักร M_i ทำงานซ้ำ ดังนั้น j_i ที่ซ้ำกันจะถูกข้ามไปยัง j_i ถัดไป

สำหรับแหล่งอาหารใหม่ v_{ij} จะถูกนำมาแทน x_{ij} เมื่อค่าฟิตเน็สของ v_{ij} ดีกว่าค่าฟิตเน็สของ x_{ij} แต่ถ้าค่าที่ได้แย่กว่าให้ใช้ x_{ij} เป็นแหล่งอาหารใหม่

ขั้นตอนที่สาม สำหรับแหล่งอาหารใหม่ v_{ij} ที่ผึ้งเอ็มพลอยพบมา เมื่อนำมาบอกกับผึ้งรับสาร โอกาสที่ผึ้งรับสารจะเลือกแหล่งอาหารนั้นมีค่าเท่ากับ p_i ดังแสดงในสมการข้างล่าง

$$□_□ = \frac{□(□_□)}{\sum_{□=1}^{□□} □(□_□)}$$

ในที่นี้ค่า $f(v_i)$ เป็นค่าฟิตเน็สของแหล่งอาหารที่ i เมื่อแหล่งอาหารทั้งหมดมีเท่ากับ SN แหล่ง หลังจากผึ้งค้นหาเลือกคำตอบที่เป็นไปได้ (x_{ij}) จากผึ้งเอ็มพลอยแล้ว คำตอบที่ดี (x_{bj}) ก็จะถูกเลือกออกมาจากคำตอบที่เป็นไปได้ที่ถูกเลือกทั้งหมด (x_{ij}) คำตอบที่ดี (x_{bj}) นี้จะไปแทนที่คำตอบที่ดีที่สุดเท่าที่หามาได้ (x_{bj} ของรอบก่อนหน้า) ถ้าหากว่าคำตอบที่ดีนี้มีค่าดีกว่า และค่าฟิตเน็ส (f_b) ก็จะถูกนำไปใช้เพื่อนำทางสำหรับการค้นหาในปริภูมิปัญหาตามสมการดังต่อไปนี้

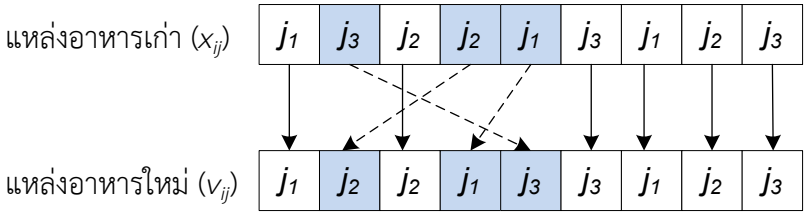
$$□_{□□} = □_{□□} + \Phi□_□(□_{□□} - □_{□□})$$

วิธีการ PBX จะใช้ในการปรับปรุงค่าของคำตอบที่เป็นไปได้ตัวเดิม (x_{ij}) ของผึ้งรับสารให้เป็นคำตอบที่เป็นไปได้ตัวใหม่ (v_{ij}) ด้วยการเปรียบเทียบกับ (x_{bj}) แทนที่จะเป็น (x_{kj}) ซึ่งเป็นแหล่งอาหาร (คำตอบที่เป็นไปได้) ข้างเคียง เช่นกัน คำตอบที่เป็นไปได้อันเดิม (x_{ij}) ที่ผึ้งรับสารจำเอาไว้จะถูกแทนที่ด้วยคำตอบที่เป็นไปได้อันใหม่ (v_{ij}) ถ้าคำตอบที่ไปได้อันใหม่ดีกว่า

ขั้นตอนที่สี่ เพื่อที่จะเป็นการเลี่ยงการลู่เข้าสู่คำตอบก่อนเวลาอันควร ผึ้งค้นหาจะไม่สนใจในคำตอบที่เป็นไปได้ตัวเดิม มันจะออกไปหาแหล่งอาหาร ซึ่งก็คือคำตอบที่เป็นไปได้ใหม่ ด้วยวิธีการออกสุ่มหาด้วยวิธีการตามสมการต่อไปนี้

$$\square_{\square\square} = \square_{\square\square} + \square_{\square\square}\left[\square_{\square\square\square} - \frac{\square\square\square\square\square\square\square\square\square\square}{\square\square\square}(\square_{\square\square\square} - \square_{\square\square\square})\right]\square_{\square\square} \qquad \text{สมการ 7.10}$$

จากสมการข้างต้น ค่าคงที่ $\square_{\square\square\square}$ และ $\square_{\square\square\square}$ เท่ากับ 1 และ 0.2 ตามลำดับ ค่า $\square_{\square\square}$ เป็นค่าสุ่ม [-1, 1] ในกรณีที่ค่า x_{ij} ไม่ได้รับการปรับปรุง ไม่ว่ารายการตารางการผลิตนี้จะผ่านกระบวนการหาอาหารของผึ้งแล้วกี่รอบก็ตาม คุณภาพของรายการตารางการผลิตไม่ดีขึ้น รายการตารางการผลิตนั้นก็จะถูกยกเลิก และถูกแทนด้วยรายการตารางการผลิตอันใหม่ (v_{ij}) ที่ผึ้งค้นหาไปหามา ในการหาแหล่งอาหารใหม่จะใช้วิธีการค้นหาแบบเรเดียสที่สอดคล้องกับสมการ 7.10 สำหรับวิธีการค้นหาเรเดียสนั้นจู่แสดงไว้ในรูปที่ 7.15 ข้างล่าง



รูปที่ 7.15 การปรับปรุงแหล่งอาหารด้วยวิธีการค้นหาเรเดียส (Radius Search) ของผึ้งค้นหา

วิธีการค้นหาเรเดียสเริ่มจากการสุ่มแหล่งอาหารมาจาก x_{ij} ที่มีอยู่แล้วมา 1 แหล่ง จากนั้นก็ทำการวิเคราะห์ว่า จำนวนมิติของแหล่งอาหารจำนวนกี่มิติจะถูกปรับปรุง ให้พิจารณาจากสมการ 7.10 ที่พิจารณาจากค่าของ $\boldsymbol{\omega}$ ในการค้นหาที่รอบแรกๆ จะพิจารณาจาก $\square_{\square\square\square}$ จากนั้นจะค่อยๆ ลดค่าลงไปจนถึง $\square_{\square\square\square}$ ในรอบแรกของการเลือกมิติที่จะถูกเปลี่ยนแปลงจะเท่ากับ 1 หรือ 100% ของมิติทั้งหมด แล้วจำนวนมิติจะลดลงเรื่อยๆ ตามจำนวนรอบของการออกหาแหล่งอาหารใหม่ของผึ้งค้นหา สำหรับวิธีการปรับปรุงแหล่งอาหารเก่า x_{ij} ให้เป็นแหล่งอาหารใหม่ v_{ij} จะเป็นดังนี้ (ดูรูปที่ 7.15)

เมื่อทราบจำนวนมิติที่ต้องการปรับปรุงแล้ว สมมุติว่าเป็น 3 มิติตามรูปที่ 7.15 เราก็จะสุ่มอีกว่ามิติใดบ้างที่จะถูกปรับปรุง ในที่นี้คือมิติที่ 2, 4 และ 5 สำหรับมิติที่ไม่ถูกปรับปรุง ให้นำค่าของมิติในแหล่งอาหารเก่า x_{ij} ไปใส่ในแหล่งอาหารใหม่ v_{ij} ที่มิติเดียวกัน สำหรับมิติที่ต้องถูกปรับปรุง ให้ย้ายค่าของมิติที่อยู่แรกสุด ไปอยู่ท้ายสุด จากนั้นนำค่าทั้งหมดไปใส่ในมิติที่ยังว่างอยู่

จากรูปที่ 7.15 มิติที่จะต้องถูกปรับปรุงคือ 2, 4 และ 5 ดังที่กล่าวมาแล้ว ซึ่งมีค่าในมิติเหล่านั้นเป็น j_3, j_2 และ j_1 ตามลำดับ ให้สลับค่าแรกไปไว้ที่ตำแหน่งหลังสุด ดังนั้นลำดับของแหล่งอาหารใหม่กลายเป็น j_2, j_1 และ j_3 แล้วนำค่าเหล่านี้ไปวางไว้ในตำแหน่งของมิติที่เหลืออยู่ (ถูกปรับปรุง) คือ 2, 4 และ 5 ตามลำดับ

ขั้นตอนที่ห้า วิธีการค้นหาแบบแวริเอเบิล เนเบอริง หรือวีเอนเอส (Variable Neighboring Search method: VNS) (Hansen, 2008) จะถูกนำมาใช้กับคำตอบที่ดีที่สุดเท่าที่พบมา (x_b) เพื่อทำให้คำตอบดีขึ้น รหัสเทียมของวีเอนเอสจะเป็นดังรูปที่ 7.16

```
Get Initial Solution, x' = x_b
Set step = 0 and p = 1
α = random integer number[1, nm]
β = random integer number [1, nm], α≠β
x' = Exchange Process(x', α,β)
While (step ≤ nm * (nm-1))
        α = random integer number[1, nm]
        β = random integer number [1, nm], α≠β
        If (p = 1) then x" = Exchange Process(x', α,β)
        Else if (p = 0) then x" = Inserting Process(x', α,β)
        If (fitness(x") ≥ fitness(x')) then x' = x"
        Else p = |p-1|
        Step = step + 1
End while
```

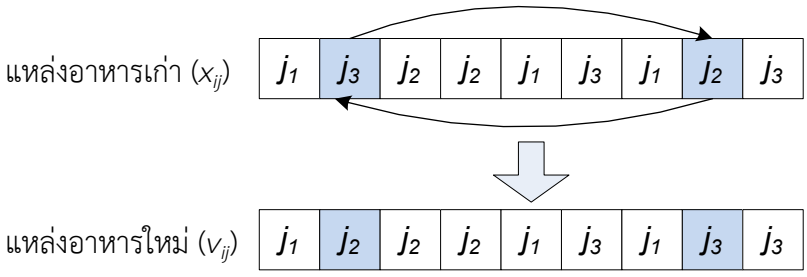
If (fitness(x') $\geq$ fitness(x_b)) then x_b = x'

End-Procedure

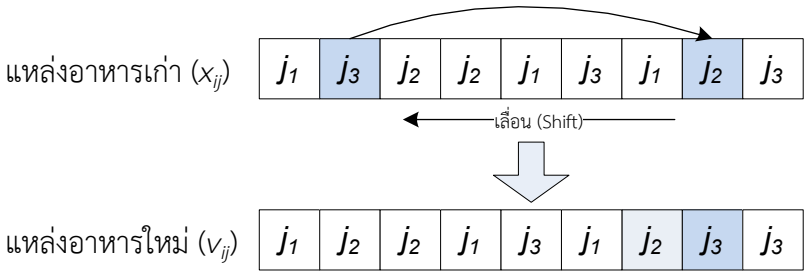
รูปที่ 7.16 วิธีการค้นหาแบบแวริเอเบิล เนเบอริง (VNS)

จากรูปที่ 7.16 ให้ α และ β เป็นเลขจำนวนเต็มที่มีค่าอยู่ระหว่าง 1 และ *nm* นิพจน์ *Exchanging _Process(x',α,β)* หมายถึงการสลับการผลิตชิ้นงาน j_i สำหรับคำตอบ *x'* ระหว่างมิติที่ α^{th} กับ β^{th} เมื่อ $\alpha \neq \beta$ นิพจน์ *Inserting_Process(x',α,β)* หมายถึงการแทรกด้วยการนำการผลิตชิ้นงานในคำตอบ *x'* ออกจากมิติที่ α^{th} และนำเข้าไปไว้ที่มิติที่ β^{th} ตัวอย่างของการทำงานตามวิธีการค้นหาแบบแวริเอเบิล เนเบอริง จะเป็นไปตามรูปที่ 7.17 และ รูปที่ 7.18 ตามลำดับ

จากรูปที่ 7.17 แสดงกระบวนการสลับ สมมุติว่ามีตัวเลขสุ่ม 2 ตัวคือ α และ β ที่มีค่าเป็น 2 และ 8 ในกรณีนี้การผลิตชิ้นงานในมิติที่ 2 และที่ 8 จะถูกสลับ ดังนั้นการผลิตชิ้นงาน j_3 ในมิติที่ 2 จะถูกสลับกับ j_2 ในมิติที่ 8 จะกลายเป็น j_2 และ j_3 ตามลำดับ



รูปที่ 7.17 กระบวนการสลับใน VNS (สำหรับ x_b ตัวใหม่)



รูปที่ 7.18 กระบวนการแทรกใน VNS (สำหรับ x_b ตัวใหม่)

จากรูปที่ 7.18 กระบวนการแทรก สมมุติว่ามีตัวเลขสุ่ม 2 ตัวคือ α และ β ที่มีค่าเป็น 2 และ 8 ในกรณีนี้การผลิตชิ้นงาน j_3 ในมิติที่ 2 จะถูกเอาออก และการผลิตชิ้นงานในส่วนที่เหลือถัดไปตั้งแต่มิติที่ 3 จนถึงตำแหน่งในมิติที่ 8 จะถูกขยับมาแทนที่ และ j_3 นี้ก็จะถูกนำไปวางไว้ในตำแหน่งที่ว่างที่มิติที่ 8

ขั้นตอนสุดท้าย เป็นการทำซ้ำขั้นตอนที่ 2 ถึง 5 จนกระทั่งจำนวนรอบเท่ากับ MCN

7.5 สรุป

อัลกอริธึมของการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ (Artificial Bee Colony) หรือ ABC พัฒนามาจากกระบวนการหาอาหารของผึ้ง โดยที่ผึ้งในกลุ่มจะถูกแบ่งออกเป็น 2 กลุ่มในตอนต้น ที่ประกอบด้วย ผึ้งเอ็มพลอย (Employed Bees) และผึ้งรับสาร (Onlooker Bees)

เมื่อเริ่มต้นของการหาอาหาร ผึ้งเอ็มพลอยจะทำหน้าที่สำรวจแหล่งอาหาร และนำข้อมูลของแหล่งอาหารนั้นไปบอกกล่าวกับผึ้งรับสาร ผึ้งรับสารจะตัดสินใจว่าจะเลือกหรือไม่เลือกแหล่งอาหารนั้นจากข้อมูลที่ได้รับ โดยพิจารณาจากคุณภาพของแหล่งอาหารซึ่งหาได้จากสมการ

$$F(x_i),\ x_i \in R^D,\ i \in \{1,2,3,\ldots SN\}$$

$$\square(\square_\square),\ \square_\square \in \square^\square,\ \square \in \{1,2,3,\ldots \square\square\}$$

โดยที่ $F(x_i)$ เป็นค่าอ็อปเจ็คทีฟ ที่ใช้บอกคุณค่าของแหล่งอาหาร ค่า x_i เป็นตำแหน่งของแหล่งอาหาร D คือมิติต่างๆ ในแหล่งอาหารนั้น และ SN เป็นจำนวนแหล่งอาหารที่ผึ้งเอ็มพลอยทำการสำรวจอยู่ เมื่อผึ้งเอ็มพลอยได้ตำแหน่งของแหล่งอาหารแล้ว มันจะทำการเปรียบเทียบกับแหล่งอาหารข้างเคียงดูว่ามีแหล่งที่ดีกว่าหรือไม่ ถ้ามามันพบแหล่งอาหารที่ดีกว่า ผึ้งเอ็มพลอยก็จะเลือกแหล่งอาหารใหม่ทันทีด้วยสมการดังนี้

$$\square_{\square\square} = \square_{\square\square} + \square_{\square\square}(\square_{\square\square} - \square_{\square\square})$$

ค่า v_{ij} เป็นคำตอบที่เป็นไปได้อันใหม่ที่ดัดแปลงมาจากคำตอบเดิม x_{ij} จากการเปรียบเทียบกับตำแหน่งข้างเคียงที่เลือกมาโดยการสุ่ม x_{kj} ค่า □□□ เป็นค่าที่ได้มาจากการสุ่มที่มีช่วงระหว่าง [-1, 1] โดยที่ค่า $k \in \{1,2,3 \ldots SN\}$ และ $j \in \{1,2,3 \ldots D\}$

สำหรับความน่าจะเป็นที่ผึ้งรับสารจะเลือกแหล่งอาหารต่างๆ ของผึ้งเอ็มพลอยจะหาได้จากสมการดังนี้

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}$$

เมื่อ p_i เป็นความน่าจะเป็นของแหล่งอาหารที่ i จะได้รับการเลือก ค่า fit_i เป็นค่าฟิตเน็สของแหล่งอาหารที่ตำแหน่งที่ i

หลังจากที่ผึ้งรับสารเลือกแหล่งอาหารจากผึ้งเอ็มพลอยแล้ว พวกมันจะปรับปรุงคำตอบของแหล่งอาหารที่พวกมันเลือกไว้ด้วยสมการเดียวกันกับการปรับปรุงคำตอบของผึ้งเอ็มพลอย แหล่งอาหารเดิมจะถูกแทนที่ด้วยแหล่งอาหารใหม่ ถ้าแหล่งอาหารใหม่ดีกว่า

สำหรับแหล่งอาหารที่ไม่มีการปรับปรุงคุณภาพเป็นเวลานาน ผึ้งเอ็มพลอยนี้ก็จะเปลี่ยนตัวเองให้เป็นผึ้งค้นหา เพื่อทำการสุ่มหาตำแหน่งของแหล่งอาหารแหล่งใหม่ ตำแหน่งใหม่ของแหล่งอาหารนี้สามารถหาได้จากสมการดังต่อไปนี้

$$□_{□□} = □_{□}^{□□□} + □□□□[0,1] * (□_{□}^{□□□} - □_{□}^{□□□})$$

ค่า x_{ij} นี้ ค่า i หมายถึงคำตอบที่เป็นไปได้ และ j หมายถึงมิติของคำตอบที่เป็นไปได้ในปริภูมิปัญหา หรืออาจจะกล่าวอีกนัยหนึ่งก็คือ x_{ij} เป็นคำตอบของผึ้งตัวที่ i ในแหล่งอาหารมิติที่ j ค่า x_j^{min} เป็นขอบเขตต่ำสุดของตำแหน่งของแหล่งอาหารในมิติที่ j และ x_j^{max} เป็นขอบเขตสูงสุดของตำแหน่งของแหล่งอาหารที่มิติที่ j

ขั้นตอนทั้งหมดข้างต้นจะทำ วนรอบกันไปจนกระทั่งพบคำตอบ หรือจนกระทั่งจำนวนรอบเท่ากับ จำนวนรอบสูงสุด (Maximum Cycle Number)

สำหรับการทำงานของเบสโซฟา เอบีซี (Best so far ABC) เป็นการปรับปรุงการทำงานของ เอบีซี ในกระบวนการทำงานที่สำคัญ 3 จุด

- จุดแรกเป็นการเปลี่ยนกระบวนการเปรียบเทียบแหล่งอาหารข้างเคียงของผึ้งรับสาร
- จุดที่สองเป็นการปรับเปลี่ยนกระบวนการทำงานของผึ้งค้นหา
- จุดสุดท้ายเป็นการปรับเปลี่ยนวิธีการเปรียบเทียบค่าฟิตเน็ส

ในจุดแรก การปรับกระบวนการเปรียบเทียบแหล่งอาหารนั้น ที่แต่เดิมจะมีวิธีการเหมือนกับผึ้งเอ็มพลอย ซึ่งเป็นการเปรียบเทียบกับแหล่งอาหารข้างเคียงแบบสุ่ม เป็นการพิจารณาเปรียบเทียบกับตำแหน่งที่ดีที่สุดเท่าที่ผึ้งเอ็มพลอยหาได้มา ซึ่งเขียนเป็นสมการใหม่ดังนี้

$$\square_{\square\square} = \square_{\square\square} + \Phi\square_{\square}(\square_{\square\square} - \square_{\square\square})$$

เมื่อ v_{id} เป็นแหล่งอาหารใหม่ของผึ้งรับสารที่ตำแหน่ง i มิติ d โดยที่ d = 1, 2, 3, ... D ตัวแปร x_{ij} เป็นแหล่งอาหารในตำแหน่งเดิม x_{bj} เป็นแหล่งอาหารที่ดีที่สุดเท่าที่พบในมิติ j ตัวแปร Φ เป็นตัวเลขที่เกิดจากการสุ่มที่มีช่วงระหว่าง –1 ถึง 1 ค่า f_b เป็นค่าฟิตเน็สของแหล่งอาหารที่ดีที่สุดเท่าที่พบ (Best-so-far Food Source)

จุดที่สองเป็นการปรับเปลี่ยนกระบวนการทำงานของผึ้งค้นหา เบสโซฟา เอบีซี ได้ปรับปรับเปลี่ยนขอบเขตของการหาแหล่งอาหารใหม่ของผึ้งค้นหา จากขอบเขตที่มีขนาดเท่ากันตลอดกระบวนการทำงานดังสมการ

$$\square_{\square\square} = \square_{\square}^{\square\square\square} + \square\square\square\square[0,1] * (\square_{\square}^{\square\square\square} - \square_{\square}^{\square\square\square})$$

เป็นขอบเขตที่แปรเปลี่ยนไปตามรอบ (Iteration) ของการค้นหา โดยที่ในรอบแรกๆ ขอบเขตการค้นหาจะกว้าง และขนาดของขอบเขตการค้นหาจะเล็กลง เมื่อจำนวนรอบของการค้นหาค่อยๆ เพิ่มขึ้นดังสมการ

$$\square_{\square\square} = \square_{\square\square} + \square_{\square\square}\left[\square_{\square\square\square} - \frac{\square\square\square\square\square\square\square\square\square}{\square\square\square}(\square_{\square\square\square} - \square_{\square\square\square})\right]\square_{\square\square}$$

จุดสุดท้ายเป็นการปรับเปลี่ยนวิธีการเปรียบเทียบค่าฟิตเน็ส ที่จากเดิม การเปรียบเทียบใช้ค่าของฟิตเน็สที่หามาได้จากสมการ

$$\square\square\square\square\square\square\square = \begin{cases} \frac{1}{1+\square(\square)}, & \square(\square) \geq 0 \\ \frac{1}{1+|\square(\square)|}, & \square(\square) < 0 \end{cases}$$

เป็นการเปรียบเทียบโดยการใช้ค่า *f(x)* ของแหล่งอาหารแต่ละตัวมาเปรียบเทียบการโดยตรง ทั้งนี้เพื่อป้องกันปัญหาที่เกิดจากค่าของ *f(x)* ที่มีจำนวนเล็กมากๆ เช่น 10^{-20} กับ 10^{-200} เมื่อใช้ค่าฟิตเน็สตามแบบเดิมจะทำให้หาค่าความแตกต่างของค่าทั้งสองไม่ได้

ในการปรับปรุงการทำงานของอัลกอริธึมเอบีซีแบบดั่งเดิมนั้น เริ่มมีการทำกันอย่างกว้างขวางมากขึ้นเช่น ซีเอบีซี (YAN, 2011) จีเอบีซี (Zhu, 2010) และ ไอเอบีซี (Li, 2011) เป็นต้น ซึ่งแนวในการพัฒนาก็จะเป็นการปรับเปลี่ยนสมการของการหาแหล่งอาหารข้างเคียงของผึ้งเอ็มพลอย การเลือกแหล่งอาหารของผึ้งรับสาร และสมการของการสร้างแหล่งอาหารใหม่ของผึ้งค้นหา โดยการเพิ่มค่าพารามิเตอร์ใหม่เข้ามาเพื่อทำให้ การค้นหาเร็วขึ้น และขอบเขตของการค้นหากว้างขึ้น

7.6 แบบฝึกหัด

1. อธิบายการหาอาหารของผึ้งตามหัวข้อดังต่อไปนี้
   1.1. ผึ้งแบ่งกลุ่มเพื่อการค้นหาแหล่งอาหารเป็นกี่กลุ่มอะไรบ้าง
   1.2. ผึ้งแต่ละกลุ่มมีหน้าที่อะไร
   1.3. การหาสมาชิกหมายถึงอะไร และใช้วิธีการอะไร
2. อธิบายขั้นตอนการทำงานที่สำคัญของอัลกอริธึมการหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์ตามหัวข้อดังต่อไปนี้
   2.1. การหาแหล่งอาหาร
   2.2. การสื่อสารให้กับผึ้งตัวอื่นทราบถึงแหล่งอาหารของตัวเองที่ไปหามา
   2.3. โอกาสที่ผึ้งตัวอื่นจะเชื่อผึ้งที่ไปพบแหล่งอาหาร แล้วบินตามไป เท่ากับเท่าไร
   2.4. การเปรียบเทียบแหล่งอาหารที่ตัวเองมีอยู่กับแหล่งอาหารข้างเคียง
   2.5. ถ้ามีผึ้งบางตัวที่ไม่ชอบแหล่งอาหารที่มีอยู่ แล้วออกไปหาแหล่งอาหารใหม่ ผึ้งเหล่านี้มีชื่อเรียกว่าอะไร
3. จากสมการ $v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj})$ ให้อธิบาย
   3.1. ค่าของ Φ_{ij} มากและน้อยมีผลต่อค่า v_{ij} ในสมการอย่างไร
   3.2. นิพจน์ $(x_{ij} - x_{kj})$ มีผลต่อค่า v_{ij} ในสมการอย่างไร
4. สมการ $x_{ij} = x_j^{min} + rand[0,1]*(x_j^{max} - x_j^{min})$ ใช้ในอัลกอริธึมอย่างไร
   4.1. ค่า *rand[0,1]* มากหรือน้อยมีผลอย่างไรต่อสมการ
   4.2. ค่า $(x_j^{max} - x_j^{min})$ หมายถึงอะไร และมีผลต่อสมการอย่างไร
5. จากสมการของ เบสโซฟา เอบีซี $v_{id} = x_{ij} + \Phi f_b(x_{ij} - x_{bj})$ ให้อธิบายว่าค่าของ f_b มีผลอย่างไรต่อสมการ
6. สมการ $$v_{ij} = x_{ij} + \phi_{ij}[\omega_{\max} - \frac{iteration}{MCN}(\omega_{\max} - \omega_{\min})]x_{ij}$$
   6.1. ใช้ในเบสโซฟา เอบีซี อย่างไร
   6.2. สมการนี้เหมือน และแตกต่างจากสมการ $x_{ij} = x_j^{min} + rand[0,1]*(x_j^{max} - x_j^{min})$ อย่างไร ในแง่ของมิติ และขอบเขตของการค้นหา

6.3. สมการดังกล่าวข้างต้นมีขอบเขตของปริภูมิปัญหากี่มิติ และสามารถเพิ่มจำนวนมิติได้หรือไม่อย่างไร

7. ให้ทดลองเขียนขั้นตอนการแก้ปัญหาของสมการ $f(x) = x_1^2 + x_2^2$ ด้วยการใช้เบสโซฟา เอบีซี
8. ให้ใช้อัลกอริธึม เอบีซี แก้ปัญหา Rosenbrock Function และทำการทดสอบในเรื่องต่อไปนี้
   8.1. ผลที่เกิดจากการเปลี่ยนแปลงจำนวนผึ้ง
   8.2. ผลของจำนวนผึ้งค้นหาที่มีต่อการค้นหาคำตอบที่เหมาะที่สุด
9. การทำงานของโรงงานขนาดเล็กแห่งหนึ่ง ทางโรงงานได้รับการว่าจ้างให้ทำงาน 5 ชิ้น (P1-P5) โดยที่โรงงานแห่งนี้มีเครื่องจักรอยู่ 2 เครื่อง (M1, M2) เท่านั้น และจากสถิติในการทำงานของเครื่องจักรที่ผ่านมา ทราบว่าเวลาของเครื่องจักรในการทำงานแต่ละชิ้นจะเป็นไปตามตารางดังต่อไปนี้

งาน	เครื่องจักร	เวลาที่ใช้ในการทำงาน (นาที)
P1	M1	20.00
	M2	9.55
P2	M1	51.15
	M1	20.00
P3	M1	41.84
	M2	17.95
P4	M1	80.84
	M2	40.33
P5	M1	12.20
	M2	5.20

ถ้าเราต้องการหาลำดับการทำงานของเครื่องจักรกับชิ้นงานว่า เครื่องจักรใดควรทำงานชิ้นใดก่อนหลัง เพื่อที่จะทำให้งานทั้ง 5 ชิ้นเสร็จเร็วที่สุด

   9.1. เขียนโปรแกรมเอบีซีเพื่อที่จะแก้ปัญหาดังกล่าว
   9.2. เขียนกราฟของการค้นหา
10. ให้ทดลองเขียนโปรแกรมการหาค่าเหมาะที่สุดแบบฝูงผึ้งเพื่อแก้ปัญหาการเดินทางของเซลส์แมน

บทที่ 8 ภูมิคุ้มกันประดิษฐ์

Artificial Immune System

8.1 คำนำ

ระบบภูมิคุ้มกันชีวภาพหรือระบบภูมิคุ้มกันของสิ่งมีชีวิต เป็นระบบป้องกันตัวเองจากเชื้อโรคและสิ่งแปลกปลอมจากภายนอกที่บุกรุกเข้ามาในร่างกาย เป็นระบบที่มีความซับซ้อนและมีวิวัฒนาการที่ยาวนาน วิวัฒนาการนี้เกิดขึ้นพร้อมกับการกำเนิดของสิ่งมีชีวิต แม้ว่ารายละเอียดการทำงานของระบบภูมิคุ้มกันนี้ยังไม่แน่ชัดว่าเป็นเช่นไร แต่จากการศึกษาของนักวิทยาศาสตร์ในเบื้องต้นทราบว่า ระบบภูมิคุ้มกันมีวิธีการป้องกันร่างกายจากสิ่งแปลกปลอมภายนอกหลายระดับ ทั้งในรูปแบบที่มีด่านเป็นชั้นคอยสกัดสิ่งแปลกปลอม และการร่วมมือกันป้องกันแปลกปลอม ซึ่งมีรายละเอียดจำนวนมาก สำหรับการนำเสนอเรื่องระบบภูมิคุ้มกันในหนังสือเล่มนี้ จะเน้นเฉพาะส่วนที่เกี่ยวข้องกับการนำไปประยุกต์ใช้กับระบบคอมพิวเตอร์เท่านั้น

ระบบภูมิคุ้มกันมีวิธีการหลายอย่างในการกำจัดเชื้อโรคขึ้นอยู่กับชนิดของเชื้อโรค และวิธีการที่เชื้อโรคนั้นเข้าสู่ร่างกาย ระบบภูมิคุ้มกันจะกำหนดวิธีการกำจัดเชื้อโรคตามชนิดของเชื้อที่เข้ามา สำหรับวิธีการกำจัดก็มีหลายลักษณะ เช่น การทำให้เชื้อโรคและสิ่งแปลกปลอมเหล่านั้นไร้ประสิทธิภาพในการทำร้ายร่างกาย หรือการทำลายเชื้อโรคและสิ่งแปลกปลอมนั้นโดยตรง

สำหรับระบบภูมิคุ้มกันแล้ว การทำงานของทั้งระบบเป็นการป้องกันตัวเอง ระบบมีแอนติบอดี (Antibody) เป็นองค์ประกอบหนึ่งที่คอยจับตัวกับแอนติเจน (Antigen) ซึ่งเป็นเชื้อโรคหรือสิ่งแปลกปลอมที่บุกรุกเข้ามาในร่างกาย การจับตัวของแอนติบอดีและแอนติเจนนี้เรียกว่าการจับคู่ (Matching) เมื่อเกิดการจับคู่แล้ว แอนติเจนจะถูกทำให้หมดฤทธิ์หรือถูกทำให้สูญสลายไป วิธีการทำงานของแอนติบอดีเหล่านี้จะมีความหลากหลายมาก โดยที่แอนติบอดีแต่ละตัวจะมีความสามารถใน

การจำแนกชนิดของสิ่งแปลกปลอม เพื่อกำหนดวิธีการการจัดการกับสิ่งแปลกปลอมนั้นอย่างเหมาะสม นอกจากนั้น แอนติบอดียังถูกแบ่งออกเป็นหลายกลุ่มตามความสามารถในการจับตัวกับแอนติเจน โดยที่การทำงานของแอนติบอดีแต่ละกลุ่มจะเป็นไปแบบอิสระต่อกันตามหน้าที่ มีการควบคุมแบบกระจาย (Distributed Control) และไม่มีการสั่งการจากส่วนกลาง

8.2 ระบบภูมิคุ้มกันทางชีวภาพ

แอนติบอดี หรือมีอีกชื่อหนึ่งคือ **อิมมูโนโกลบูลิน** (Immunoglobulin) เป็นโปรตีนขนาดใหญ่ที่มีรูปร่างเป็นรูปตัว Y อยู่ในระบบภูมิคุ้มกัน ที่ร่างกายของสัตว์ชั้นสูงสร้างขึ้น มีหน้าที่ตรวจจับและทำลายฤทธิ์สิ่งแปลกปลอมเช่น แบคทีเรีย และไวรัส ที่เข้ามาในร่างกาย แอนตีบอดีแต่ละชนิดจะจดจำโมเลกุลเป้าหมายที่จำเพาะของแอนติเจนได้ ซึ่ง **แอนติเจน** ก็คือสารที่จับตัวอยู่กับสิ่งแปลกปลอม หรือเป็นสารที่ไม่มีอยู่ในร่างกายอยู่แล้ว เมื่อมีสิ่งแปลกปลอมเข้าสู่ร่างกาย สารนี้ (แอนติเจน) สามารถกระตุ้นให้เกิดการตอบสนองทางภูมิคุ้มกันแบบจำเพาะ (Specific Immune Response) ได้ โดยการทำให้เกิดการสร้างแอนติบอดีที่สามารถทำปฏิกิริยากับแอนติเจนนั้นๆ โดยตรง

8.2.1 เซลล์ที่ทำหน้าที่ภูมิคุ้มกัน

เซลล์ที่มีความสำคัญในการทำหน้าที่ภูมิคุ้มกันเป็นเซลล์เม็ดเลือดขาว 2 ชนิดคือ เซลล์ที (T-cells) และ เซลล์บี (B-cells) เซลล์ทั้งสองชนิดนี้เกิดจากไขกระดูก แต่เซลล์ทีจะถูกส่งต่อไปยังต่อมไทมัส (Thymus) เพื่อการฟูมฟักก่อนที่มันจะกระจายตัวไปตามร่างกายผ่านทางหลอดโลหิตและหลอดน้ำเหลือง (Lymphatic Vessels)

เซลล์ทีมี 3 ชนิด ชนิดแรกคือเซลล์ทีเฮลเปอร์ (Helper T-cells) ซึ่งมีส่วนสำคัญในการกระตุ้นเซลล์บี ชนิดที่สองคือเซลล์ทีคิลเลอร์ (Killer T-cells) เป็นเซลล์ที่ทำหน้าที่จับตัวกับผู้บุกรุก และฉีดสารเคมีมีพิษเข้าไปทำลายผู้บุกรุก และชนิดที่สามเซลล์ทีซัพเพรสเซอร์ (Supressor T-cells) เซลล์นี้ทำหน้าที่ในการยับยั้งการทำงานของเซลล์ภูมิคุ้มที่จะจับกับเซลล์ตัวเอง (Self-cells) ซึ่งเซลล์ตัวเองนี้เป็นเซลล์ของร่างกายตัวเอง การยังยั้งการทำงานของเซลล์ภูมิคุ้มไม่ให้จับกับเซลล์ตัวเองก็เพื่อป้องกันการทำลายตัวเอง (Autoimmune)

เซลล์บีมีหน้าที่ในการผลิตและหลั่งสารแอนติบอดี สารแอนติบอดีนี้จะเป็นสารที่เฉพาะเจาะจงกับกับแอนติเจนชนิดที่มันจะจับตัว เมื่อมีการตรวจพบแอนติเจนที่จับตัวกับสิ่งแปลกปลอมที่บุกรุกเข้ามาในร่างกาย แอนติบอดีจะเข้าไปจับคู่ (Match) กับแอนติเจน และจะทำลายเซลล์สิ่งแปลกปลอมนั้น

8.2.2 กลไกของระบบภูมิคุ้มกัน

การป้องกันสิ่งแปลกปลอมที่ลุกล้ำเข้าไปในร่ายกายของมนุษย์ เป็นการทำงานที่ประสานกันในหลายระดับ ระบบภูมิคุ้มกันจะประกอบด้วยเครื่องป้องกันที่เป็นกายภาพเช่นผิวหนัง ระบบขับเหงื่อ เครื่องป้องกันทางสรีรวิทยาเช่น เอนไซม์ กรดในกระเพาะอาหาร และระบบภูมิคุ้มกัน ในระบบภูมิคุ้มกันของร่างกายทั้งหมดสามารถจำแนกออกได้เป็น 2 อย่างคือ อินเนทอิมมูนิตี (Innate Immunity) หรือภูมิคุ้มกันแบบไม่เฉพาะเจาะจง (Non-specific Immune Response) และ อะแดพทีฟอิมมูนิตี (Adaptive Immunity) หรือภูมิคุ้มกันแบบเฉพาะเจาะจง (Specific Acquired Immunity) ซึ่งภูมิคุ้มกันทั้งสองประเภทจะทำงานประสานกัน และมีอิทธิพลต่อกัน สำหรับ อะแดพทีฟอิมมูนิตี ยังสามารถแบ่งออกได้เป็น 2 ชนิดเช่นกัน คือ ฮูมูรอลอิมมูนิตี (Humoral Immunity) และ เซลล์มิดิเอ็ดอิมมูนิตี (Cell Mediated Immunity)

อินเนทอิมมูนิตี (Innate Immunity) หรือภูมิคุ้มกันแบบไม่เฉพาะเจาะจง เป็นภูมิคุ้มกันที่มีมาแต่กำเนิด เช่น ผิวหนัง ขนอ่อน และกรดในกระเพาะอาหาร ภูมิคุ้มกันนี้มีวิธีการกำจัดสิ่งแปลกปลอมออกจากร่างกายโดยวิธีการง่ายๆ การกำจัดจะมีขั้นตอนที่เริ่มจากการที่ระบบนี้เป็นเครื่องกีดขวางตามธรรมชาติอยู่แล้ว ได้แก่ผิวหนัง เยื่อเมือกซึ่งบุตามอวัยวะต่างๆ และขนอ่อน (Cilia) ถ้าสิ่งมีแปลกปลอมสามารถผ่านเครื่องกีดขวางตามธรรมชาตินี้เข้าไปได้ สิ่งแปลกปลอมนั้นก็จะถูกร่างกายกำจัดโดย การกิน (Phagocytosis) โดยเซลล์ที่ทำหน้าที่ในการกิน (Cell-eating) จะเคลื่อนตัวไปหาสิ่งแปลกปลอมแล้วประกบติด (Attachment) ต่อมาจะกลืน และทำการย่อย (Intracellular Digestion) ด้วยกลไกหลายอย่างในเซลล์ แล้วจึงขับสิ่งแปลกปลอมที่ถูกทำลายแล้วออกจากเซลล์

อะแดพทีฟอิมมูนิตี (Adaptive Immunity) หรือภูมิคุ้มกันแบบเฉพาะเจาะจง (Specific Acquired Immunity) เป็นระบบที่มีวิธีการกำจัดสิ่งแปลกปลอมด้วยกลไกที่ยุ่งยากกว่าวิธีแรก เกิดขึ้นเมื่อร่ายกายไม่สามารถใช้อินเนทอิมมูนิตีกำจัดสิ่งแปลกปลอมหรือแอนติเจนหรืออิมมูโนเจน

(Immunogen) ออกไปได้ เซลล์ที่ทำหน้ากำจัดสิ่งแปลกปลอมในระบบนี้คือ เม็ดเลือดขาวลิมโฟไซท์ (Lymphocytes) การตอบสนองดังกล่าวแบ่งออกเป็น 2 ส่วน คือ

ฮูมูรอลอิมมูนิตี (Humoral Immunity) คือการตอบสนองทางภูมิคุ้มกันโดยแอนติบอดีที่เป็นสารน้ำซึ่งเป็นส่วนประกอบอยู่กับของเหลวในร่างกาย จะทำงานร่วมกับเซลล์บีที่มีแหล่งกำเนิดจากไขกระดูก และย้ายไปที่เนื้อเยื่อน้ำเหลือง (Lymphoid Tissue) เพื่อพัฒนาเป็นเม็ดเลือดขาวบี-ลิมโฟไซต์ (B-lymphocyte) เมื่อเจริญเต็มแล้วจะถูกปล่อยออกมาสู่กระแสเลือดและเนื้อเยื่อน้ำเหลือง เข้าสู่กระแสน้ำเหลืองทั่วร่างกาย เมื่อมีแอนติเจนหรืออิมมูโนเจนเข้ามา เซลล์บีนี้จะทำการตอบสนองด้วยการเปลี่ยนแปลงตนเองไปเป็นเซลล์พลาสมาเพื่อทำหน้าที่สร้างแอนติบอดี แล้วเคลือบตัวกับแอนติเจน เมื่อแอนติเจนหรือสิ่งแปลกปลอมถูกเคลือบด้วยแอนติบอดีแล้วก็จะถูกขับออก

เซลล์มิดิเอ็ดอิมมูนิตี (Cell Mediated Immunity) เซลล์ที่ทำหน้าที่ในการตอบสนองต่อหน้าที่นี้คือเซลล์ที (T Lymphocyte) ซึ่งมีต้นกำเนิดจากไขกระดูกเช่นเดียวกับเซลล์บี โดยการฟูมฟักตนเองผ่านทางต่อมไทมัส (Thymus Gland) ให้เป็นเซลล์ที (T-cell) ที่สมบูรณ์ แล้วทำหน้าที่ในการฆ่าเซลล์ที่ติดเชื้อไวรัส และเซลล์ที่เป็นเนื้องอก

8.2.3 การทำงานของระบบภูมิคุ้มกัน

ระบบภูมิคุมกันทางชีวภาพสามารถมองได้เป็นการป้องกันแบบหลายชั้น ซึ่งการป้องกันแต่ละชั้นจะมีกลไกในการป้องกันที่แตกต่างกันไป ประกอบด้วย การตรวจจับ (Detection) การรู้จำ (Recognition) และการตอบสนอง (Response) ซึ่งชั้นของการป้องกันหลักประกอบด้วย เครื่องป้องกันทางกายภาพ (Anatomic Barrier) หรืออินเนทอิมมูนิตี และภูมิคุ้มกันแบบเฉพาะเจาะจงหรืออะแดพทีฟอิมมูนิตี สำหรับอินเนทอิมมูนิตี และอะแดพทีฟอิมมูนิตี นี้จะเชื่อมต่อกันภายในและมีอิทธิพลต่อกันและกัน เมื่ออะแดพทีฟอิมมูนิตี รู้จำการปรากฏตัวของผู้บุกรุก มันจะกระตุ้น ฮูมูรอลอิมมูนิตี และเซลล์มิดิเอ็ดอิมมูนิตี จัดการต่อผู้บุกรุกนั้นอย่างเป็นลำดับขั้น อินเนทอิมมูนิตีจะตอบโต้โดยตรงต่อผู้บุกรุก ถ้าผู้บุกรุกหรือสิ่งแปลกปลอมสามารถผ่านด่านอินเนทอิมมูนิตีนี้ไปได้ ร่างการจะกระตุ้นให้อะแดพทีฟอิมมูนิตี ออกมาตอบโต้กับสิ่งแปลกปลอมนั้น ตามชนิดของผู้ที่บุกรุกเข้ามา

จากรูปที่ 8.1 เป็นภาพที่แสดงวิธีการที่ระบบภูมิคุ้มกันแบบอะแดพทีฟอิมมูนิตีทำการตอบโต้กับสิ่งแปลกปลอม (แอนติเจน) ที่บุกรุกเข้ามาในร่างกาย ในขั้นตอนที่ I และ II แสดงการที่แอนติเจนบุกรุกเข้ามาในร่างกาย และกระตุ้นการทำงานของเซลล์ที ในขั้นตอนที่ III ขั้นตอนที่ IV เป็นขั้นตอนที่เซลล์ทีส่งข้อมูลไปกระตุ้นเซลล์บี ขั้นตอนที่ V เป็นการจับคู่กับแอนติเจน VI เป็นการสร้างแอนติบอดี และ VII เป็นการทำลายแอนติเจนโดยแอนติบอดี

แอนติบอดีจะถูกสร้างขึ้นมาในระบบของฮูมูรอลอิมมูนิตี ซึ่งมีฟังก์ชันในการรู้จำ และการจับคู่กับแอนติเจนที่แปลกปลอมเข้ามาในร่างกาย โมเลกุลของแอนติบอดีจะกระตุ้นให้เกิดการสร้างเซรุ่ม (Serum) ที่สามารถจับตัวกับบริเวณที่เหมาะสมของโมเลกุลของแอนติบอดี จากนั้นจะทำการเจาะรูที่ผิวเซลล์ของแอนติเจนและจะทำลายแอนติเจนนั้นไป

การสื่อสารในระบบภูมิคุ้มกันมีบทบาทสำคัญในการแลกเปลี่ยนและส่งผ่านข้อมูลระหว่างการต่อสู้กับผู้บุกรุก ระบบภูมิคุ้มกันทางชีวภาพมีรูปแบบการกระจายข่าวสารและการแลกเปลี่ยนข้อมูลที่สำคัญ 2 แบบ แบบแรกเป็นการกระจายข่าวสารที่เรียกว่า อิมมูดิฟฟิวชัน (Immune Diffusion) เป็นการส่งข้อความจากภูมิคุ้มกันหนึ่งไปยังภูมิคุ้มกันอื่นๆ โดยไม่มีการรับข้อมูลกลับ (Feedback) แบบที่สองของการแลกเปลี่ยนข้อมูลจะเรียกว่า อิมมูไดอะล็อก (Immune Dialogue) เป็นการส่งสัญญาณแลกเปลี่ยนข่าวสารกันในระบบภูมิคุ้มกันอย่างต่อเนื่อง ความไวของการตอบสนองของภูมิคุ้มกันขึ้นอยู่กับบริบทของการตอบโต้กันระหว่างเซลล์ภูมิคุ้มกันกับเซลล์สิ่งแปลกปลอม

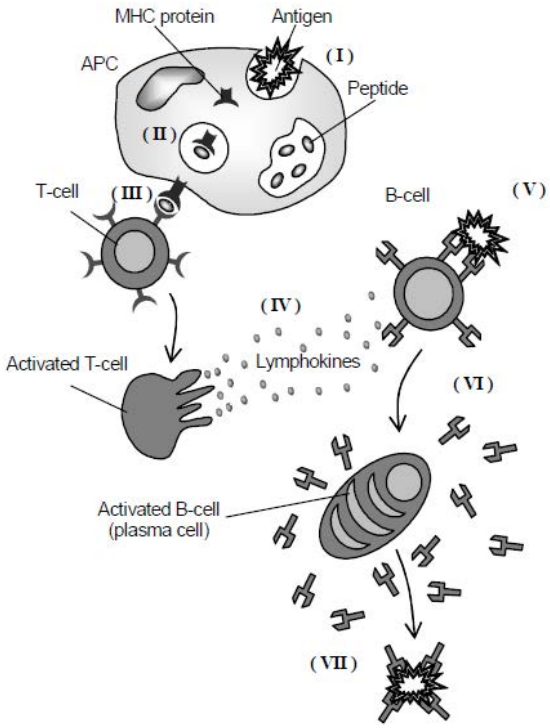


Figure -1. Pictorial representation of the essence of the acquired immune system mechanism (taken from de Castro and van Zuben (1999): I-II show the invade entering the body and activating T-Cells, which then in IV activate the B-cells, V is the antigen matching, VI the antibody production and VII the antigen's destruction.

รูปที่ 8.1 การประสานงานของ เซลล์บี เซลล์ที และ แอนติเจน จาก (U and Dasgupta, D Aickelin, 2005) ที่นำมาจาก คาสโตร (de Castro, 2000)

ร่างกายจะอยู่ภายใต้การท้าทายจากสิ่งแปลกปลอมเหล่านี้อย่างต่อเนื่อง และจะต้องปรับตัวอยู่ตลอดเวลา การส่งสัญญาณมีความสำคัญมากต่อระบบการป้องกันทางชีวภาพที่มีต่อการบุกรุกของสิ่งแปลกปลอม ด้วยการส่งสัญญาณจากภายนอกสู่ภายใน และผลของการส่งสัญญาณจะทำให้เกิดการเปลี่ยนแปลงของเซลล์ เพื่อสร้างวิธีการตอบสนองที่เหมาะสมกับผู้บุกรุก

ในแง่มุมของการประมวลข้อมูลข่าวสาร ระบบภูมิคุ้มกันเป็นระบบที่สามารถปรับตัวเองได้ทั้งในรูปแบบขนานและแบบกระจาย (Parallel and Distributed Adaptive System) ที่มีกลไกการควบคุม

แบบกระจาย ระบบภูมิคุ้มกันนี้สามารถนำมาประยุกต์ใช้กับ การถอดคุณลักษณะ (Feature Extraction) การส่งสัญญาณ (Signaling) การเรียนรู้ (Learning) ความจำ (Memory) และ การค้นคืนข่าวสารข้อมูล (Associative Retrieval) นอกจากนั้น การประมวลข้อมูลข่าวสารยังสามารถนำมาใช้กับการแก้ปัญหาเรื่องการรู้จำ (Recognition) และการจำแนกชนิด (Classification) ของแอนติเจน ในทางปฏิบัติ ระบบภูมิคุ้มกันมีความสามารถในการรู้จำรูปแบบที่เกี่ยวข้อง และจดจำรูปแบบที่เคยพบมาก่อน จากนั้นจึงใช้วิธีการคอมินาโทริกเพื่อสร้างรูปแบบการตรวจจับที่มีประสิทธิภาพ (Dasgupta, 2006)

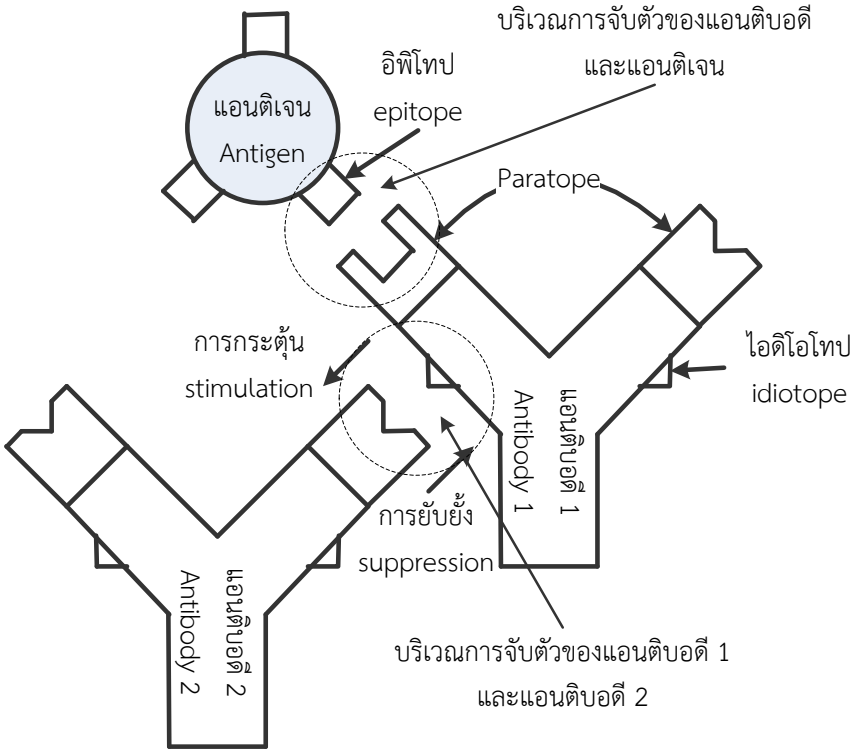
สำหรับระบบภูมิคุ้มกันประดิษฐ์นับเป็นการวิจัยที่ใหม่มาก เริ่มขึ้นเมื่อทศวรรษที่ 1990 และมีการนำไปประยุกต์ใช้ในการแก้ปัญหาหลายเรื่องเช่น ระบบความปลอดภัยของคอมพิวเตอร์ การรู้จำภาพ และการหาค่าเหมาะที่สุด ทั้งทางด้านวิทยาศาสตร์และวิศวกรรม แม้ว่าการวิจัยทางด้านนี้จะเป็นการวิจัยที่น่าสนใจสำหรับนักวิจัยทางด้านวิทยาการคอมพิวเตอร์เป็นอย่างมาก แต่การออกแบบวิธีการพื้นฐานสำหรับใช้งานยังไม่เป็นรูปร่างดีนัก สำหรับวิธีการที่นิยมได้แก่ หลักการเลือกโคลนอล (Clonal Selection Principle) อัลกอริธึมการเลือกเชิงลบ (Negative Selection Algorithm) และโมเดลโครงข่ายภูมิคุ้มกัน (Immune Network Models) (Dasgupta, 2006) (Dasgupta, et al., 2011) เป็นต้น

เซลล์ที่มีบทบาทสำคัญของอะแดพทีฟอิมมูนิตี คือ เซลล์บี และเซลล์ที ซึ่งที่ผิวของเซลล์เหล่านี้มีโมเลกุลของโปรตีนที่เรียกว่า รีเซพเตอร์ (Receptor) รีเซพเตอร์ของเซลล์บีสามารถจับคู่กับแอนติเจนที่เข้าคู่กับมันได้ การจับคู่ของเซลล์บีกับแอนติเจนนี้เป็นการที่เซลล์บีทำให้แอนติเจนนั้นหมดฤทธิ์

ตามทฤษฎีของการเลือกคลอนอล (Clonal Selection) ของระบบภูมิคุ้มกันนั้น เซลล์เม็ดเลือดขาวจะทำงานอย่างเป็นอิสระ และเมื่อมันสามารถจับคู่ได้ เซลล์บีจะแตกตัวเพิ่มจำนวนโดยการโคลนตัวเอง และปล่อยรีเซพเตอร์อิสระออกมาเรียกว่าแอนติบอดีเพื่อการจับตัวกัน การจับตัวกันนี้จะเกิดขึ้นที่บริเวณของแอนติบอดีที่เรียกว่าพาราโทป (Paratope) และบริเวณของแอนติเจนที่เรียกว่าอิพิโทป (Epitope) ที่รูปร่างสามารถเข้ากันได้ ในขณะเดียวกันเจอร์เน่ (NK Jerne, 1974) ได้ตั้งข้อสมมุติฐานว่าเซลล์เม็ดเลือดขาวจะมีปฏิกิริยาตอบสนองต่อกันในลักษณะเชื่อมโยง ทำให้ระบบภูมิคุ้มกันมีการทำงานเป็นแบบโครงข่ายสากล (Global Network) โดยปฏิกริยาของเซลล์ต่างๆ เหล่านี้จะมีทั้งการกระตุ้น

(Stimulate) และการยังยั้ง (Suppress) ด้วยกระบวนการรู้จำและการจับคู่กันในระหว่างพวกมันกันเอง ดังนั้นพาราโทปของแอนติบอดีไม่เพียงแต่จะจับคู่กับอิพิโทปของแอนติเจนเท่ากัน แต่ยังจับกับอิพิโทปของแอนติบอดีด้วยกันเองที่เรียกว่า ไอดิโอโทป (Idiotope) ด้วย

จากรูปที่ 8.2 แสดงโครงสร้างของแอนติบอดี และแสดงวิธีการที่แอนติบอดีถูกยังยั้ง เพราะแอนติบอดีอื่นที่สามารถรู้จำไอดิโอโทปของมันและต้องการจับคู่กับมัน พร้อมทั้งวิธีการที่ความเข้มข้นของการจับคู่ถูกกระตุ้นให้เพิ่มขึ้นเมื่อพวกมันรู้จำไอดิโอโทปของแอนติเจนอื่น



รูปที่ 8.2 การจับคู่กันของแอนติเจน และแอนติบอดี (Amanda Whitbrook, 2012)

8.3 หลักการของการเลือกโคลนอล

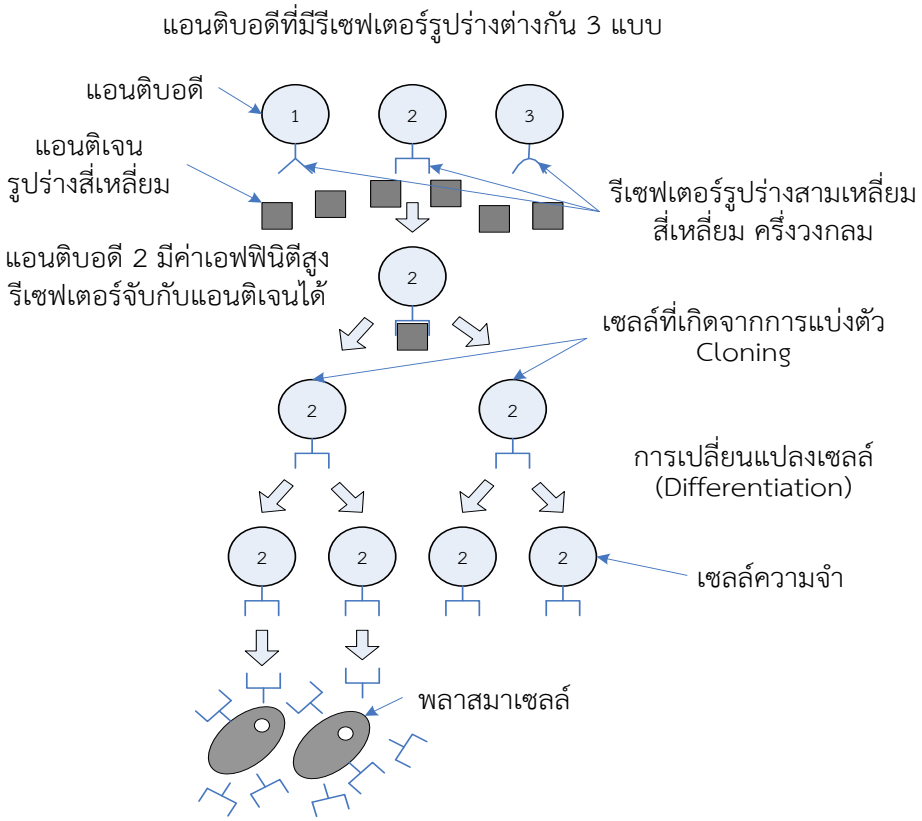
หลักการเลือกโคลนอล (Clonal Selection Principle) เป็นทฤษฎีที่ได้สร้างความเปลี่ยนแปลงเป็นอย่างมากในศาสตร์ภูมิคุ้มกันเมื่อทศวรรษที่ 1950 ผู้ที่ทำการค้นคว้าเรื่องการเลือกโคลนอลในระบบ

ภูมิคุ้มกันอย่างต่อเนื่องที่สำคัญท่านหนึ่งคือเบอร์เนท (F. Macfarlane Burnet) (Burnet, 1957) (Burnet, 1959) เบอร์เนทเป็นผู้ที่พัฒนางานวิจัยต่อจากงานของเทลเมจ (Forsdyke, 1995) (Talmage, 1959) ทำให้เขามีชื่อเสียงมาก จนกระทั่งทฤษฎีที่เขาค้นคว้านี้ถูกเรียกว่า ทฤษฎีการเลือกโคลนอลของเบอร์เนท (Burnet's Clonal Selection Theory) (Ada, 1989)

เบอร์เนทอธิบายกลไกของความจำแบบภูมิคุ้มกันวิทยา (Immunological Memory) ว่าเป็นความจำที่เกิดขึ้นเกิดจากการโคลนเม็ดเลือดขาว (Lymphocyte) เป็นการถ่ายทอดคุณลักษณะของเซลล์ต้นแบบไปยังเซลล์ที่เกิดใหม่ โดยที่เซลล์ที่เกิดใหม่สามารถแบ่งได้ 2 ชนิด ชนิดแรก เป็นชนิดที่ถูกโคลนออกมาเพื่อต่อสู้กับเชื้อโรคโดยทันที ในขณะที่ อีกชนิดหนึ่งเป็นชนิดที่ถูกสร้างให้มาอยู่ในร่างกายอย่างยาวนาน และเป็นภูมิคุ้มกันที่ใช้ในการต่อต้านแอนติเจน ในปี 1958 โนสเซล และ เลเดอร์เบอร์ก (Sir Gustav Nossal and Joshua Lederberg) (Nossal, 1958) ได้พิสูจน์ว่าเซลล์บีจะผลิตแอนติบอดีเดียวเสมอ หมายความว่าเซลบีหนึ่งๆ จะจับแอนติเจนได้เพียงชนิดเดียว ซึ่งเป็นปรากฏการณ์แรกของทฤษฎีการเลือกโคลนอล

ตามหลักการเลือกโคลนอล ระบบภูมิคุ้มกันต่อสู้กับสิ่งแปลกปลอมหรือแอนติเจนด้วยวิธีการต่างๆ 3 ขั้นตอนคือ กระบวนการเลือกโคลนอล กระบวนการแบ่งตัวอย่างรวดเร็ว (Proliferation) และกระบวนการมาชัวเรชัน (Maturation) เมื่อเชื้อโรคเข้าสู่ร่างกาย แอนติเจนของเชื้อโรคนี้จะถูกรู้จำโดยเซลล์ภูมิคุ้มกัน ด้วยการที่เซลล์ภูมิคุ้มกันมีรีเซฟเตอร์หรือพาราโทปที่สามารถตรวจจับมันได้ และเข้าไปจับคู่กับเชื้อโรคที่เรียกว่าแอฟฟินิตี (Affinity) จากนั้นเซลล์ภูมิคุ้มกันนี้จะถูกกระตุ้นให้เกิดการแบ่งตัว เซลล์ภูมิคุ้มกันที่ทำหน้าที่นี้คือเซลล์บีที่ผลิตออกมาจากไขกระดูก สำหรับเซลล์บีที่จะจับคู่กับแอนติเจนได้จะต้องมีค่า (หรือรูปร่าง) ของรีเซฟเตอร์ใกล้เคียงกับของแอนติเจน ยิ่งรูปร่างเหมือนกันมากเท่าไรค่าแอฟฟินิตีกับแอนติเจนก็จะสูงขึ้นเท่านั้น เซลล์บีที่มีค่าแอฟฟินิตีกับแอนติเจนสูงจำนวนหนึ่งจะถูกกระตุ้นให้จับแอนติเจน โดยแต่ละเซลล์จะหลั่งสารที่เป็นแอนติบอดีเพียงชนิดเดียว ที่สามารถต่อสู้กับแอนติเจนนั้นเท่านั้น จากนั้นเซลล์บีนี้จะจับตัวกับแอนติเจนด้วยรีเซฟเตอร์ และถูกกระตุ้นให้แบ่งตัวอย่างรวดเร็ว และเซลล์เหล่านี้จะเติบโตเป็นเซลล์ที่ไม่สามารถขยายตัวต่อไปอีกได้ เรียกว่าเซลล์พลาสมา (Plasma Cells) กระบวนการนี้จะเรียกว่ามาชัวเรชั่น (Maturation)

เม็ดเลือดขาว นอกจากจะเปลี่ยนแปลงตัวเองไปเป็นพลาสมาแล้ว บางส่วนยังมีการเปลี่ยนแปลงเซลล์ (Differentiation) ไปเป็นเซลล์ความจำ (Memory Cells) ที่มีอายุยืนโดยผ่านกระบวนการมาชัวเรชัน เซลล์ความจำนี้จะไหลเวียนไปตามกระแสเลือด น้ำเหลือง และเนื้อเยื่อ และเมื่อต้องเผชิญกับการกระตุ้นของแอนติเจนใหม่ มันจะเปลี่ยนตัวเองเป็นเม็ดเลือดขาวขนาดใหญ่ที่มีความสามารถในการผลิตแอนติบอดีที่มีแอฟฟินิตีสูง สำหรับการตอบโต้กับแอนติเจนชนิดใหม่ที่บุกรุกเข้ามา ตามที่แสดงไว้ในรูปที่ 8.8



รูปที่ 8.3 การเลือกโคลนอล ดัดแปลจาก คาสโตร (L.N. and Zuben, F.J.V. de Castro, 2000)

เบอร์เนท และเมดาวาร์ (Burnet and Peter Medawar) ได้ค้นพบการเกิดขึ้นของเม็ดเลือดขาวในระบบภูมิคุ้มกัน 2 ชนิดคือชนิดที่มีการต่อต้านจากเนื้อเยื่อของตัวเอง (Self-Tissue) และชนิดไม่

ต่อต้านกับเนื้อเยื่อของตัวเอง และยังได้ค้นพบว่าเนื้อเยื่อสามารถปลุกถ่ายไปยังร่างกายอื่นได้ การค้นพบนี้ทำให้เกิดความเข้าใจในเรื่องระบบภูมิคุ้มกันเป็นอย่างมาก และทำให้พวกเขาได้รับรางวัลโนเบลร่วมกันในปี 1960

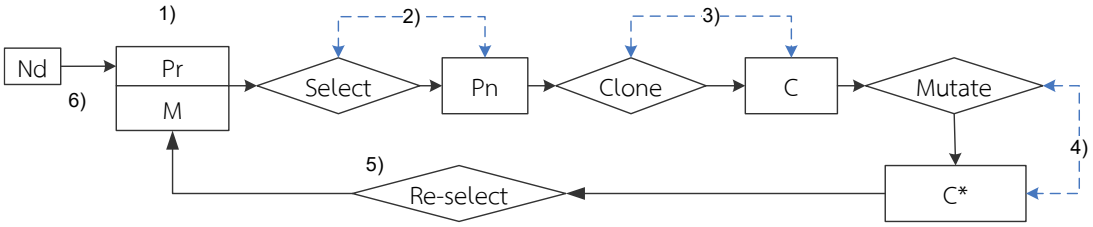
หลักการเลือกโคลนอล ยังได้อธิบายคุณลักษณะพื้นฐานของการที่ระบบภูมิคุ้มกันตอบสนองต่อการกระตุ้นของแอนติเจน ซึ่งเป็นการสร้างแนวคิดของความเข้าใจที่ว่า เซลล์ที่สามารถรู้จำแอนติเจนได้เท่านั้นจะแพร่พันธุ์ด้วยการแบ่งตัวมันเองได้ แต่เซลล์ที่ไม่สามารถรู้จำแอนติเจนถูกกำจัดออกไปในที่สุด เปรียบเสมือนกับการเลือกและไม่เลือกให้เซลล์ภูมิคุ้มกันให้คงอยู่ในร่างกายโดยธรรมชาติ ลักษณะหลักของทฤษฎีการเลือกโคลนอลมีดังนี้

- เซลล์ที่เกิดใหม่จะคัดลอกตัวเองจากพ่อแม่ของมัน ตามกลไกการโคลนตัวเองที่มีการแบ่งตัวอย่างรวดเร็ว
- เซลล์เม็ดเลือดขาวเกิดใหม่ที่มีรีเซพเตอร์เป็นแบบต่อต้านเนื้อเยื่อตัวเองจะถูกกำจัด
- การที่เซลล์มีปฏิกริยาต่อแอนติเจนทำให้เกิดขึ้นของการแบ่งตัวอย่างรวดเร็ว (Proliferation) และการเปลี่ยนแปลงเซลล์ (Differentiation)

จากการศึกษาเพิ่มเติมของเวอร์โคค์ซี (Verkoczy, 2004) และพีลันดา (Pelanda, 2006) เมื่อไม่นานมานี้พบว่า ในขั้นตอนของการคัดเลือก ที่แสดงว่าเซลล์ภูมิคุ้มกันที่มีค่าการจับคู่หรือเอฟฟินิตี (Affinity) กับแอนติเจนสูงจะถูกกระตุ้น และเซลล์ที่มีค่าแอฟฟินิตีต่ำจะไม่ถูกกระตุ้นนั้น ในบางครั้งเซลล์ที่มีค่าแอฟฟินิตีต่ำบางตัวจะมีการแก้ไข (Editing) รีเซฟเตอร์ของตัวมัน ด้วยการที่เซลล์ภูมิคุ้มกันเหล่านี้จะลบรีเซฟเตอร์ของตัวมันออก และพัฒนารีเซฟเตอร์ใหม่ขึ้นมาใหม่ ด้วยการปรับแต่งยีนจากบางส่วนของรีเซฟเตอร์เดิม

8.3.1 อัลกอริธึมการเลือกโคลนอล

อัลกอริธึมการเลือกโคลนอลถูกนำเสนอโดย คาสโตรและซูเบน (L.N. and Zuben, F.J.V. de Castro, 2000) มีขั้นตอนการทำงานแบ่งออกเป็น 6 ขั้นตอนคือ ดังรูปที่ 8.4



รูปที่ 8.4 การเลือกโคลนอลที่เสนอโดย (L.N. and Zuben, F.J.V. de Castro, 2000)

1) สร้างชุดของคำตอบที่เป็นทางเลือก (Pr) ที่ประกอบด้วยเซทย่อยของเซลล์ความจำร่วมกับประชากรที่เหลืออยู่
2) เลือกคำตอบที่ดีที่สุดจำนวน n ตัวแรก (Pn) โดยการพิจารณาจากค่าแอฟฟินิตี (ความมั่นคงในการจับตัวกับเชื้อโรคหรือแอนติเจน)
3) โคลนตัวเลือกทั้ง n ตัว (C)
4) นำตัวเลือกที่โคลนมาได้ทั้งหมดผ่านกระบวนการไฮเปอร์มิวเตชัน (Hypermutatuion) จำนวนของตัวเลือกที่ผ่านกระบวนการไฮเปอร์มิวเตชันจะเป็นสัดส่วนของแอฟฟินิตีระหว่างแอนติบอดีกับแอนติเจน และการสร้างแอนติบอดีที่เป็นแบบมาชัวเรชัน (C*)
5) การเลือก C* ชุดหนึ่งที่จะเป็นเซลล์ความจำ M ที่ใช้สำหรับการแทนที่ตัวเลือก P
6) แทนที่แอนติบอดีจำนวนหนึ่งที่มีค่าแอฟฟินิตีต่ำ ด้วยแอนติบอดีใหม่ N_d

อัลกอริธึมของคาสโตรถูกพัฒนาต่อมาโดยซู (Y., Gao, S., Dai, H., Li, F. and Tang, Z. Zhu, 2007) ด้วยการเพิ่มวิธีการปรับแต่งรีเซฟเตอร์ (Receptor Editing) โดยนำเสนอวิธีการ โซมาติกไฮเปอร์มิวเตชัน (Somatic Hypermutation) ซึ่งเป็นกระบวนการที่ใช้สำหรับการค้นหาคำตอบให้ดีขึ้น หรือเป็นการเสริมการทำงานของเอ็กซ์พล้อยเตชัน สำหรับการปรับแต่ริเซฟเตอร์จะช่วยให้การค้นหาไม่ติดกับดักของโลคอลมินิมั่ม ซึ่งเป็นการเพิ่มโอกาสใหม่ให้กับการหาคำตอบในปริภูมิคำตอบที่เรียกว่าเอ็กซ์พลอเรชัน พร้อมทั้งเสนอวิธีการทำโซมาติกไฮเปอร์มิวเตชัน และวิธีการปรับแต่รีเซฟเตอร์ โดยใช้การแสดงค่าของปัญหาด้วย เชลฟ์-สเปส โมเดลของเพอเรลสัน (Perelson, 1989) ในรูปของสตริง โดยการแสดงลำดับของยีนในเซลรีเซฟเตอร์เป็น $R = (r_1, r_2, ..., r_N)$

8.3.2 การเลือกโคลนอลสำหรับแก้ปัญหาการเดินทางของเซลส์แมน

ตัวอย่างในหัวข้อนี้เป็นอัลกอริธึมการเลือกโคลนอล ที่ใช้สำหรับการแก้ปัญหาการเดินทางของเซลส์แมน วิธีการแก้ปัญหานี้ถูกนำเสนอโดยซู (Zhu, 2007) ซึ่งมีรายละเอียดดังต่อไปนี้

initailization();
(for i=1 to NG //NG เป็นจำนวนของเจนเนอเรชัน หรือจำนวนรอบของการทำงาน
(for j = 1 to NCS //จำนวนของประชากรที่โคลน
somatic hypermutation(); //ดู โซมาติกไฮเปอร์มิวเตชัน ข้างล่าง
evaluation(); //การหาค่าแอฟฟิริเอชัน หรือระยะทางทั้งหมด
(if (Dis < curDis) then (curDis = Dis) and update(); // Dis เป็นระยะทาง และ curDis เป็นระยะทางที่ดีที่สุดจนถึงปัจจุบัน
else receptor-editing(); // ดูการปรับแต่งรีเซฟเตอร์ข้างล่าง
(if (Dis < curDis) then (curDis = Dis) and update();
else clonal-deletion(); and count++;)) // ยกเลิกตัวเลือกและเพิ่มดัชนีนับจำนวนรอบ
(if (count > Maxcount) then self-crossover() and count =0;))) // ดูเซลฟ์ ครอสโอเวอร์ข้างล่าง

โซมาติกไฮเปอร์มิวเตชันเป็นกระบวนการสลับตำแหน่งของยีน เช่นการทำโซมาติกไฮเปอร์มิวเตชันกับยีนที่ตำแหน่ง i และ j จะเป็นการสลับยีนตัวที่ i และ j ดังนี้

$r_1, r_2, ..., \boldsymbol{r_i}, r_{i+1}, ..., \boldsymbol{r_j}, r_{j+1}, ..., r_N$ // ลำดับยีนเริ่มต้น
$r_1, r_2, ..., \boldsymbol{r_j}, r_{i+1}, ..., \boldsymbol{r_i}, r_{j+1}, ..., r_N$ // ลำดับยีนจาก โซมาติกไฮเปอร์มิวเตชัน

บรรทัดแรกแสดงลำดับของยีนที่ต้องการทำโซมาติกไฮเปอร์มิวเตชันโดยมีการระบุตำแหน่งที่ $\boldsymbol{r_i}$ และ $\boldsymbol{r_j}$ เป็นตำแหน่งที่ต้องการสลับ สำหรับบรรทัดที่สองที่ตำแหน่งของ $\boldsymbol{r_i}$ ถูกแทนด้วย $\boldsymbol{r_j}$ และที่ตำแหน่งของ $\boldsymbol{r_j}$ ถูกแทนด้วย $\boldsymbol{r_i}$ ตามลำดับ

การปรับแต่งรีเซฟเตอร์ด้วยวิธีการสลับตำแหน่งจะได้ดังนี้

r_1, r_2, ..., $\mathbf{r_i}$, r_{i+1}, ..., $\mathbf{r_j}$, r_{j+1}, ..., r_N // ลำดับยีนเริ่มต้น

$\mathbf{r_i}$, r_{i+1}, ..., $\mathbf{r_j}$, => $\mathbf{r_{j-1}}$, ..., r_{i+1}, $\mathbf{r_i}$ // แสดงตำแหน่งที่สลับ

r_1, r_2, ..., $\mathbf{r_{j-1}}$, ..., r_{i+1}, $\mathbf{r_i}$, r_{j+1}, ..., r_N // เมื่อปรับแต่งรีเซฟเตอร์แล้ว

บรรทัดแรกแสดงลำดับของยีนที่ต้องการปรับแต่งรีเซฟเตอร์ ตั้งแต่ r_1 ... r_N บรรทัดที่สองแสดงลำดับของยีน $\mathbf{r_i}$, r_{i+1}, ..., $\mathbf{r_j}$ และ $\mathbf{r_{j-1}}$, ..., r_{i+1}, $\mathbf{r_i}$ ที่ต้องการปรับแต่งด้วยการสลับตำแหน่งของชุดยีนทั้ง 2 ชุดที่ระบุไว้นี้ และบรรทัดที่สามเป็นลำดับของยีนใหม่

เซลฟ์ ครอสโอเวอร์ (Self-crossover) เป็นกลไกการปรับปรุงข้อมูลของตัวเองที่การทำครอสโอเวอร์เกิดขึ้นโดยที่มีสตริงเพียงชุดเดียว แทนที่จะเป็นการครอสโอเวอร์ของสตริง 2 ตัวตามปกติ วิธีการนี้ถูกนำเสนอโดย คุนดู และพาล (Kundu, 1999) การทำงานมีขั้นตอนดังนี้

สมมุติว่า *S* = 0001 0010 0110 0101 1011 เป็นสตริงที่มีความยาวขนาด L=20 บิท

เลือกตำแหน่ง p บนสตริง S โดยการสุ่ม โดยที่ตำแหน่งของ p จะต้องอยู่ไม่เกินกว่าความยาวของสตริง L ดังนี้

$p = (0 < p < L)$

ตำแหน่ง p บนสตริง S นี้ทำให้เกิดสตริงย่อย 2 ตัว ให้ชื่อว่า *s*1 และ *s*2 จากนั้นสุ่มหาตำแหน่งบนสตริง s1 และ s2 ให้เป็นสตริงย่อย อย่างละ 2 ตัวกลายเป็น s11 และ s12 กับ s21 และ s22 แล้วทำการครอสโอเวอร์โดยการ s^1 = s11+s22 และ s^2 = s12+s21 ทำให้สตริงที่เกิดใหม่ S1 กลายเป็น $S1=s^1+s^2$ ดังตัวอย่างต่อไปนี้

กำหนดให้ S เป็นสตริงที่มีความยาวเท่ากับ 20 ดังนี้

S = 0001 0010 0110 0101 1011

สุ่มตำแหน่งบน S ได้ p = 9 ดังนั้น S จะกลายเป็นสตริงย่อย 2 ตัว s1 และ s2 ดังนี้

$s1$ = 0001 0010 0 และ $s2$ = 1100 1011 011

สุ่มหาตำแหน่งบน s1 และ s2 เป็น p1 =5 และ p2 = 4 เราจะได้ s11 และ s12 กับ s21 และ s22 ดังนี้

$s11$ = 00010; $s12$ = 0100; $s21$ = 1100; and $s22$ = 1011011:

ได้ s^1 = s11+s22 และ s^2 = s12+s21 ดังนี้

s^1 = 00010 1011011 และ s^2 = 0100 1100:

สุดท้ายจะได้สตริงที่ผ่านการทำเซลฟ์ ครอสโอเวอร์ แล้วดังนี้

$S1 = s^1 + s^2$ = 00010 1011011 0100 1100

สำหรับการนำหลักการเลือกโคลนอลมาใช้สำหรับการแก้ปัญหาของการหาค่าเหมาะที่สุดยังมีอีกจำนวนมาก โดยเฉพาะอย่างยิ่งการนำมาแก้ปัญหาการเดินทางของเซลส์แมน เช่น (Xu, 2011) (Chen, 2010) การรู้จำตัวเลข (de Castro, 2000) (de Casto, 2002) การหาค่าเหมาะที่สุดของมัลติโมเดล (Multi Model Optimization) (Lui, 2005) เป็นต้น

8.4 อัลกอริธึมการเลือกเชิงลบ

การเลือกเชิงลบ (Negative Selection) คือกลไกที่ใช้กำจัดลิมโฟไซท์หรือเม็ดเลือดขาวที่ทำปฏิกิริยาต่อเซลล์ตัวเอง (Self-cells) จุดประสงค์ของการเลือกเชิงลบคือ การทำให้เม็ดเลือดขาว ที่อยู่ในระบบภูมิคุ้มกันมีความสามารถในการตรวจจับแอนติเจนที่เป็นสิ่งแปลกปลอม โดยที่ไม่มีปฏิกิริยากับเนื้อเยื่อหรือเซลล์ตัวเอง ในระหว่างการสร้างเซลล์ที ยีนที่เป็นตัวกำหนดคุณสมบัติในการจับตัวกับแอนติเจน ที่เป็นรีเซฟเตอร์จะถูกสร้างขึ้นมาอย่างสุ่มที่ตอมไทมัส ที่นี่เซลล์ที่มีปฏิกิริยากับเซลล์ตัวเองจะถูกทำลาย สำหรับเซลล์ทีที่เหลือจะถูกฟูมฟักแล้วแพร่กระจายไปตามส่วนต่างๆ ของร่างกาย เพื่อทำหน้าที่เป็นภูมิคุ้มกันร่างกายจากสิ่งแปลกปลอม

8.4.1 อัลกอริธึมการเลือกเชิงลบ

ผู้ที่นำอัลกอริธึมการเลือกเชิงลบมาใช้งานสำหรับ การแก้ปัญหาวิศวกรรมคนแรกคือฟอร์เรสต์ (Forrest, 1994) โดยการออกแบบอัลกอริธึมครั้งนั้น ใช้สำหรับการสร้างระบบดักจับไวรัสที่บุกรุกเข้ามาในคอมพิวเตอร์ การออกแบบได้กำหนดหลักการสำคัญสามอย่างดังนี้

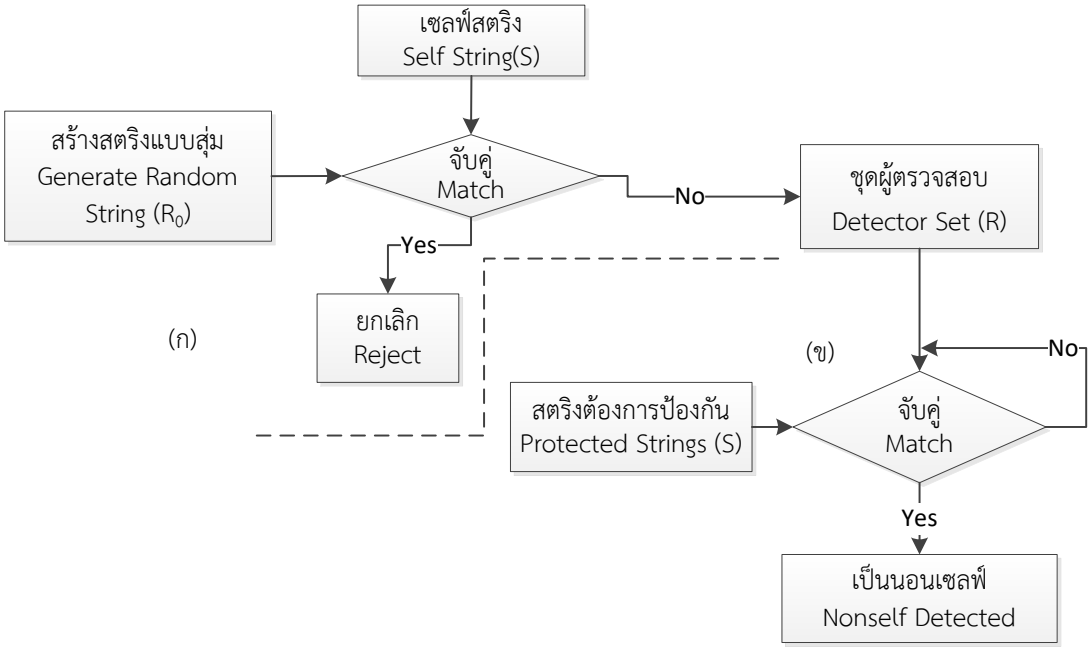
1. อัลกอริธึมสำหรับการตรวจสอบ (Detection) อัลกอริธึมนี้หนึ่งอันจะใช้ได้กับเอกสารหนึ่งฉบับเท่านั้น และการตรวจสอบไวรัสในแต่ละส่วน ไม่ว่าจะเป็นระบบซอฟแวร์แต่ละระบบ หรือคอมพิวเตอร์แต่ละตัวในระบบโครงข่าย จะเป็นไปอย่างอิสระ
2. การตรวจสอบเป็นแบบความน่าจะเป็น (Probability) หมายความว่าไวรัสที่ประสบความสำเร็จในการบุกรุกเข้าไปในระบบใดระบบหนึ่งได้จะมีความน่าจะเป็นสูงที่จะประสบความสำเร็จในการบุกรุกเข้าไปในระบบอื่นได้
3. ระบบที่มีความมั่นคงจะต้องสามารถป้องกันการบุกรุกจากสิ่งแปลกปลอมทุกชนิดได้

สำหรับการออกแบบอัลกอริธึมได้แบ่งเป็นสองส่วนคือ

1. การสร้างชุดของผู้ตรวจสอบ (Detectors) โดยให้ชุดผู้ตรวจสอบเหล่านี้คือสตริงที่ไม่สามารถจับคู่ หรือแมทช์ (Match) กับข้อมูลที่ต้องการป้องกัน ขั้นตอนนี้จะเรียกว่าการเซนเซอร์ (Censoring) ดังแสดงในรูปที่ 8.6(ก)
2. การติดตาม (Monitor) เป็นวิธีการป้องกันข้อมูลด้วยการเปรียบเทียบ (Compare) ข้อมูลที่ต้องการป้องกันทั้งหมดกับชุดผู้ตรวจสอบที่ได้จากส่วนแรก ถ้าชุดผู้ตรวจสอบนั้นเคยดักจับสิ่งแปลกปลอมได้ ดังนั้นก็เป็นไปได้ที่ชุดผู้ตรวจสอบนั้นจะดักจับสิ่งแปลกปลอมใหม่ได้อีกดังรูปที่ 8.6 (ข)

การจับคู่หรือการแมทช์ของสตริง เป็นการนำสตริง 2 ชุดมาเปรียบเทียบกัน ถ้าสตริงทั้งสองชุดเหมือนกัน จะเรียกว่าการจับคู่ (Match) ในการจับคู่ระหว่างค่าของ S ที่เป็นสตริงตัวเอง และ R_0 ซึ่งเป็นสตริงที่เกิดจากการสุ่มขึ้นมา ชุดข้อมูล R_0 นี้เปรียบเสมือนตัวไวรัสที่เป็นข้อมูลไม่ทราบชนิด เมื่อ R_0 ทำการจับคู่กับสตริง S สตริงย่อยของ R_0 ที่ไม่สามารถจับคู่กับ S ได้จะถูกยกเลิกไป สำหรับสตริงย่อยของ

R_0 ที่สามารถจับคู่กับ S ได้ จะถูกเก็บสะสมไว้ใน R เรียกว่า เรเพอร์ทัวร์ (Repertoire) ที่เป็นชุดข้อมูลที่ใช้สำหรับการตรวจสอบ



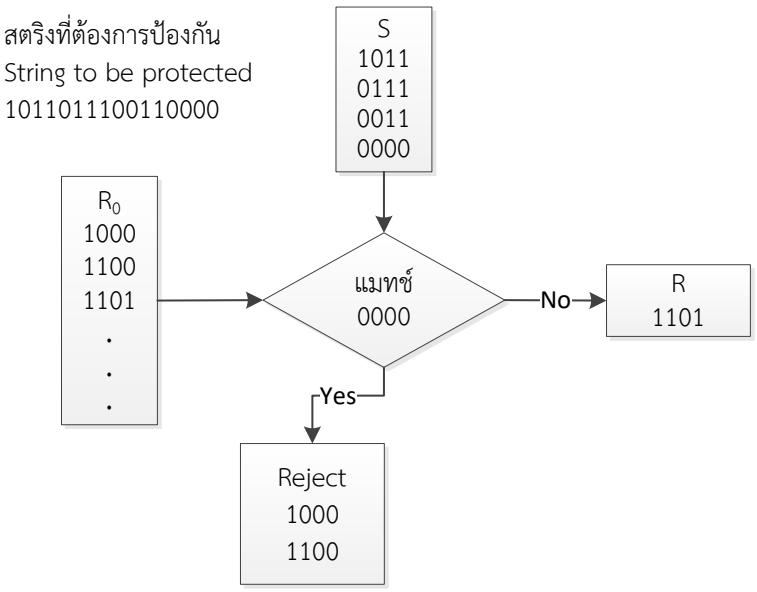
รูปที่ 8.5 การทำงานของการเลือกเชิงลบ (ก) การสร้างชุดผู้ตรวจสอบ (detectors) หรือขั้นตอนการเซนเซอร์ (Censoring) (ข) กระบวนของการติดตาม (Monitor) ดัดแปลงจากฟอร์เรสต์ (S., Perelson, S., Allen, L., and Cherukuri, R. Forrest, 1994)

ในการจับคู่กันของสตริงเพื่อสร้างชุดผู้ตรวจสอบนั้น จะอาศัยสตริงที่ได้มาจากการสุ่ม R_0 ที่รูปแบบไม่สามารถคาดการได้ล่วงหน้า เปรียบเสมือนไวรัสที่เข้ามาในระบบ ทำการเปรียบเทียบกับสตริงตัวเอง ถ้าสตริงของ R_0 ชุดใดสามารถแมทช์กับสตริงตัวเองได้ สตริงชุดนั้นจะถูกพิจารณาว่าเป็นสตริงตัวเอง สตริงตัวเองนี้จะไม่จัดอยู่ในฐานข้อมูลของตรวจสอบ สำหรับชุดสตริงที่ไม่สามารถจับคู่กับสตริงตัวเองได้จะถูกพิจารณาว่าเป็นไวรัส และถูกส่งไปเป็นชุดตรวจสอบ สำหรับการจับคู่ระหว่าง R_0 กับ S หรือสตริงตัวเองนั้น ฟอร์เรสต์ได้นำเสนอในสองรูปแบบคือ

- การจับคู่แบบสมบูรณ์

- การจับคู่บางส่วน

การจับคู่แบบสมบูรณ์นั้นสตริงย่อยของ R_0 จะต้องเหมือนกับ S ทุกหลัก เช่นถ้าเรามี S ที่มีการแบ่งเป็นสตริงย่อย 4 หลักดังต่อไปนี้ {0010 1000 1001 0000 0100 0010 1001 0011} และมี R_0 เป็น {0111, 1000, 0101, 1001} ดังนั้น R จะมี 2 ตัวคือ 0111 และ 0101 เพราะสองชุดของ R_0 นี้ไม่สามารถจับคู่กับ S ได้ สำหรับสตริง 1000 และ 1001 ที่สามารถจับคู่กับ R_0 จะถูกคัดออก ซึ่งชุดของ R_0 ที่เป็น 1000 และ 1001 นี้เป็นชุดที่มีการจับคู่แบบสมบูรณ์กับ S



รูปที่ 8.6 การจับคู่กันของสตริงที่เราต้องการป้องกัน (S) และสตริงที่มาจากการสุ่ม (R_0)

ในการจับคู่บางส่วน ตามตัวอย่างที่แสดงในรูป 8.7 โดยเริ่มจากการนำชุดของสตริงที่ต้องการการป้องกัน (S) มาแบ่งเป็นสตริงย่อยที่มีความยาวเท่ากัน (ตัวอย่างรูปที่ 8.6 ความยาวของสตริงย่อยเท่ากับ 4 หลัก) ไปแมทช์กับชุดของสตริงย่อย ที่มีจำนวนหลักเท่ากัน ที่สร้างขึ้นมาด้วยการสุ่ม (R_0) การจับคู่ครั้งนี้จะแตกต่างจากการจับคู่ครั้งแรก คือสตริงย่อยของ R_0 สองชุดแรก 1000 และ 1100 จะถูกกำจัดออกไป เพราะทั้งสองตัวแมทช์กับ 0000 อย่างน้อยจำนวน 2 หลัก สำหรับสตริงย่อย 1101 จะไม่สามารถแมทช์กับ 0000 อย่างน้อย 2 หลัก จึงถูกนำไปเก็บไว้ที่ R ในการจับคู่เช่นนี้เป็นการจับคู่ที่

กระทำเพียงบางส่วนของสตริงย่อยโดยที่มีสตริงที่เหมือนกันอย่างต่อเนื่องอย่างน้อย 2 หลัก หรือ r = 2 ซึ่ง r นี้เป็นพารามิเตอร์ที่บอกระดับของการจับคู่ดังนี้

R_0 1000

R_0 1100

S 0000

จะเห็นว่า R_0 และ S เป็นการจับคู่บางส่วน ที่มีการจับคู่กันอย่างต่อเนื่องที่ 2 หลักสุดท้ายตามที่ขีดเส้นใต้ไว้ ดังนั้น R_0 ที่เป็น 1000 และ 1100 ดังกล่าวข้างต้นจะถูกตัดออก

เนื่องจากการจับคู่กันของ R_0 และ S เป็นการจับคู่ด้วยความน่าจะเป็น ที่ขึ้นต่อจำนวนของสตริงเริ่มต้น (N_{R0}) และจำนวนของสตริงที่ต้องการป้องกัน (N_S) เพื่อที่จะทราบค่าความเป็นไปได้ของชุดตัวตรวจสอบที่จะตรวจพบการบุกรุกของสตริงไวรัส ฟอร์เรสต์ (S., Perelson, S., Allen, L., and Cherukuri, R. Forrest, 1994) ได้แสดงรายละเอียดของการหาค่าความเป็นไปได้ ดังนี้

$$\square_{\square} = \square_{\square_0} * \square \qquad \text{สมการ 8.1}$$

$$\square_{\square} = (1 - \square_{\square})^{\square_{\square}} \qquad \text{สมการ 8.2}$$

$$\square_{\square} \approx \square^{-\square_{\square}\square_{\square}} \qquad \text{สมการ 8.3}$$

$$\square_{\square} = \square_{\square_0} * \square = \frac{-\square\square\square_{\square}}{\square_{\square}} \qquad \text{สมการ 8.4}$$

$$\square_{\square_0} = \frac{-\square\square\square_{\square}}{\square_{\square} * (1-\square_{\square})^{\square_{\square}}} \qquad \text{สมการ 8.5}$$

$$\square_{\square} = \frac{1}{\square_{\square}} \qquad \text{สมการ 8.6}$$

โดยที่ตัวแปรต่างๆ มีความหมายดังต่อไปนี้

N_{R0} = จำนวนสตริงผู้ตรวจสอบ (Detector) เริ่มต้น (ก่อนทำการเซนเซอร์)

N_R = จำนวนสตริงผู้ตรวจสอบ (Detector) หลังการเซนเซอร์ (ขนาดของเรเพอร์ทัวร์)

N_S = จำนวนของสตริงตัวเอง

P_M = ความเป็นไปได้ของการจับคู่สตริงที่ได้มาจากการสุ่ม 2 ชุด

f = ความเป็นไปได้ของสตริงที่ได้จากการสุ่มไม่สามารถแมทช์กับสตริงตัวเองตัวใดเลยของ N_S

P_f = ความเป็นไปได้ที่ผู้ตรวจสอบ N_R ไม่สามารถตรวจพบผู้บุกรุกเลย

การพัฒนาอัลกอริธึมการเลือกเชิงลบนั้นมีมาอย่างต่อเนื่อง ซึ่งรายละเอียดหาดูได้จากงานของ จี และดาสกูพตา (Ji, 2007) แต่อย่างไรก็ตามอัลกอริธึมเหล่านี้ยังเรียกได้ว่า อยู่ในขั้นต้นของการพัฒนาที่ยังมีความสมบูรณ์ไม่มากนัก แม้ว่าการนำเสนออัลกอริธึมนี้ในครั้งแรกจะเกิดขึ้นตั้งแต่ปี 1994 แล้วก็ตาม แต่ก็ยังมีการตั้งข้อสงสัยในเรื่องของการนำอัลกอริธึมการเลือกเชิงลบไปประยุกต์ใช้งาน และความเป็นเอกลักษณ์ของอัลกอริธึมอยู่บ่อยครั้ง (Garrett, 2005) แต่อย่างไรก็ตาม สไตเบอร์ (Stibor, 2005) ได้ชี้ให้เห็นถึงความเป็นไปได้ในการนำการเลือกเชิงลบไปใช้ในการแก้ปัญหาเฉพาะอย่าง โดยเฉพาะทางวิศวกรรมบางอย่างได้

8.4.2 การประยุกต์ใช้งาน

การนำอัลกอริธึมการเลือกเชิงลบไปใช้งานในปัจจุบันยังมีไม่มากนัก การนำไปประยุกต์ใช้งานจะอยู่ในลักษณะของการจำลองแบบ เพื่อทดสอบการทำงานของอัลกอริธึมมากกว่าการนำอัลกอริธึมไปใช้งานจริง ในตอนนี้จะนำเสนอตัวอย่างที่ปรากฏอยู่ในบทความวิจัยเพื่อให้เห็นแนวทางของการนำอัลกอริธึมนี้ไปใช้งาน

การตรวจสอบสีผิวหนังและการจัดหมวดหมู่ (Skin Detection and Classification) การวิจัยนี้นำเสนอโดยเบนดิแอบ (Bendiab, 2010) ซึ่งเป็นการนำเสนอกรอบแนวคิดของการนำอัลกอริธึมการเลือกเชิงลบมาใช้ในการตรวจสอบว่า สีของภาพดิจิตอลที่นำมาตรวจสอบเป็นสีผิวของมนุษย์หรือไม่ โดยกำหนดว่า สีผิวในภาพถ่ายของมนุษย์แต่ละพิกเซล (Pixels) เป็นเซลล์ตัวเอง แล้วนำสีที่ได้จากภาพถ่ายทุกสีทั้งหมดมาตรวจสอบกับเซลล์ตัวเอง การทำงานของระบบเป็นดังนี้

การสร้างโมเดลของการแก้ปัญหา

แอนติบอดี คือคำตอบที่เป็นตัวเลือกของปัญหาการจัดหมวดหมู่ (Classification Problem) ซึ่งแอนติบอดีนี้จะเปรียบเสมือนชุดคุณสมบัติของผิวหนัง แอนติเจนเปรียบเสมือนปัญหาที่ต้องการแก้ และข้อกำหนด (Constraints) ในการแก้ปัญหานี้ แอนติเจนจะเปรียบเสมือนชุดของพิกเซลที่เราต้องการจำแนกว่าเป็นผิวหนังหรือไม่ใช่ผิวหนังของมนุษย์ ซึ่งชุดของพิกเซลนี้จะถูกเลือกออกมาด้วยการสุ่มปริภูมิรูปร่าง (Shape Space) ของการแก้ปัญหาในระบบภูมิคุ้มกันประดิษฐ์จะคล้ายกับปริภูมิปัญหา

(Problem Space) ของการแก้ปัญหาแบบระบบการผลิต คือพื้นที่ที่แอนติเจนและแอนติบอดีมีปฏิสัมพันธ์กัน เพื่อสร้างทางเลือกต่างๆ จำนวนมาก สำหรับการพิจารณาว่า ข้อมูลที่กำลังพิจารณาอยู่นั้น อันไหนเป็นเซลล์ตัวเองและอันไหนไม่ใช่จะประกอบด้วยการคำนวณเพื่อหาค่าของแอฟฟีนีตีของการจับคู่กันด้วยรีเซพเตอร์ (Receptors) สำหรับปัญหาที่เรากำลังพิจาณาก็คือการพิจารณาว่าพิกเซลไหนเป็นผิวหนัง และพิกเซลไหนไม่ใช่

นิยามปัญหา การกำหนดวิธีการหาคุณลักษณะของผิวหนังเพื่อการจัดหมวดหมู่จะแบ่งเป็นสองลักษณะคือ คุณลักษณะของสี และคุณสมบัติของพื้นผิว

คุณลักษณะของสี จะใช้โมเมนต์ของสี (Color Moment) ในการแสดงคุณลักษณะของสีในภาพ โมเมนต์ลำดับที่หนึ่ง (The First Order Moment: μc) ซึ่งจะแทนค่าสีโดยเฉลี่ย โมเมนต์ลำดับที่สอง (The Second Order: σc) จะแทนความเบี่ยงเบนมาตรฐาน และโมเมนต์ลำดับที่สาม (The Third Order Moment: θc) ใช้แทนความไม่สมมาตรของสี (Asymmetry Color) ซึ่งสมการที่ใช้ในการคำนวณจะอยู่ในตารางที่ 8.1 และสตริงที่ใช้สำหรับการแสดงค่าเพื่อการคำนวณของโมเมนต์ทั้งสามจะแสดงตามรูปที่ 8.7

	คุณลักษณะ (Feature)	สูตร (Formula)
โมเมนต์ของสี	โมเมนต์ลำดับที่หนึ่ง (The first order moment: μc)	$\mu_C = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}^{C}$
	โมเมนต์ลำดับที่สอง (The second order: σc)	$\sigma_C = \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(p_{ij}^{C} - \mu_C)^2\right]^{1/2}$
	โมเมนต์ลำดับที่สาม (The third order moment: θc)	$\theta_C = \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(p_{ij}^{C} - \mu_C)^3\right]^{1/3}$
คุณสมบัติของพื้นผิว	เอนโทรฟี่ (Entropy)	$Entropy = \sum_i\sum_j C(i,j)\log(C(i,j))$
	พลังงาน (Energy)	$Energy = \sum_i\sum_j C^2(i,j)$
	คอนทราสท์ (Contrast)	$Contrast = \sum_i\sum_j (i-j)^2 C(i,j)$
	โฮโมจินิตี (Homogeneity)	$Energy = \sum_i\sum_j \frac{C(i,j)}{1+\lvert i-j\rvert}$

ตารางที่ 8.1การแสดงค่าคุณลักษณะของสีและพื้นผิว (Texture)

μ_{cR}	μ_{cG}	μ_{cB}	σ_{cR}	σ_{cG}	σ_{cB}	θ_{cR}	θ_{cG}	θ_{cB}	ENT	E	C	H

รูปที่ 8.7 การแทนค่าแอนติบอดี และแอนติเจนของที่แสดงโมเมนต์ของสีและคุณสมบัติของพื้นผิว

คุณสมบัติของพื้นผิว (Texture Characteristics) ถูกแสดงในรูปของเอนโทรฟี่ (Entropy) พลังงาน (Energy) คอนทราสท์ (Contrast) และโฮโมจินิตี (Homogeneity) โดยที่มี C(i, j) เป็นโคออคเคอร์เร้น เมตริกซ์ (Co-occurrence Matrix) ตารางที่ 8.1 จะเป็นรายละเอียดของสมการดังที่กล่าวมาแล้ว รูปที่ 8.7 เป็นการแทนค่าของแอนติเจน และแอนติบอดี

สำหรับอัลกอริธึมของการวิเคราะห์สีผิว มีอยู่ 4 ขั้นตอนคือ การตั้งค่าเริ่มต้น การเรียนรู้ การเลือก และการจัดหมดหมู่ รายละเอียดของอัลกอริธึมจะมีดังต่อไปนี้

อัลกอริธึม

การตั้งค่าเริ่มต้น (Initialization):

- การหาลักษณะเด่น (Features Extraction) : เลือกชุดของพิกเซลและพิกเซลข้างเคียงในเทรนนิ่งเซทที่เป็นส่วนของผิวหนังของภาพ เซทข้อมูลนี้จะเป็นเซลล์ตัวเอง
- การหาพารามิเตอร์ (*Parameters Extraction) : เลือก*เธรส์โฮล์ด (Threshold) ของ r (พารามิเตอร์) ที่บอกระดับของการจับคู่ ว่า r ควรมีค่าเท่าไร

การเรียนรู้ (Learning):

- สร้างตัวตรวจสอบ (Detector) ออกมาโดยการสุ่มเพื่อเป็นตัวเลือกจำนวนหนึ่ง
- การประเมินผลแอฟฟินิตี (Affinity Evaluation): พิจารณาระยะทางยูคลีเดียน (Euclidean Distance) ระหว่างเซนเซอร์กับตัวเลือกที่ได้จากชุดของการเทรน (Training Set) ที่เป็นเซลล์ตัวเอง

การเลือก (Selection):

if ตัวเลือกของผู้ตรวจสอบแมทช์กับเซลล์ตัวเองจำนวน r ตำแหน่งอย่างต่อเนื่อง then ให้นำผู้ตรวจสอบนั้นทิ้งไป

else ตัวเลือกนั้นจะถูกนำเข้าไปรวมกับผู้ตรวจสอบอื่นในเรเฟอร์ทัวร์ ซึ่งจะเป็นชุดคำตอบที่เป็นไปได้ (Possible Solutions);

การจัดหมวดหมู่ (Classification):

- เลือกพิกเซลหนึ่งออกมาและคำนวณค่าที่ใช้วัดกับพิเซลข้างเคียงที่เหมาะสม
- จับคู่หรือแมทช์พิกเซลที่เลือกไว้กับชุดของผู้ตรวจสอบในเรเฟอร์ทัวร์

If สตริงทั้งสองแมทช์กันได้อย่างน้อย r ตำแหน่งอย่างต่อเนื่องกัน Then สตริงตัวเลือกจะเข้ากันได้กับชุดของนอนเซลฟ์ และถูกจัดให้อยู่ในพวกของนอนเซลฟ์ และให้นำออกจากประชากร

Otherwise พิกเซลจะเข้ากันได้กับชุดของเซลล์ตัวเอง และจะถูกจัดให้อยู่ในชุดของผิวหนัง;

ในการทดลอง เบนดิแอบ ได้ใช้ภาพถ่ายของคนที่มีสีผิวต่างๆ ทั้งผิวขาว ผิวเหลือง และผิวดำ ที่ถ่ายทั้งในที่ร่มและที่แจ้ง ประมาณ 300 ภาพ ป้อนให้กับระบบเพื่อทำการเรียนรู้สีและพื้นผิวของมนุษย์ สำหรับการสร้างชุดผู้ตรวจสอบ ในขั้นตอนการเซนเซอร์ (Censoring) เพื่อเก็บไว้ในเรเพอร์ทัวร์ (R) จากนั้นก็นำภาพที่ต้องการตรวจสอบ (R_0) ป้องเข้าไปในกระบวนของการติดตาม (Monitor) เพื่อการจัดหมวดหมู่ (Classification)

ในการประยุกต์ใช้งานอัลกอริธึมของการเลือกเชิงลบนั้น ยังมีงานที่น่าสนใจอีก เช่นการนำไปใช้ในการตรวจสอบความผิดพลาดของเครื่องบิน (Dasgupta, 2004) การนำไปใช้กับการหาค่าเหมาะที่สุด (Cao, 2007) และการใช้การเลือกเชิงลบสำหรับการหาขอบของภาพ (Ji, 2005) เป็นต้น

8.5 โครงข่ายภูมิคุ้มกัน

ความยากของการทำงานในระบบภูมิคุ้มกันก็คือ ในบางครั้งแอนติบอดีสามารถรู้จำและทำลายเนื้อเยื่อของอวัยวะที่มันอาศัยอยู่ ซึ่งเป็นการทำร้ายตัวเองที่เนื่องมาจากการจับคู่กันของแอนติบอดีและเซลล์ของร่างกายหรือเซลล์ตัวเอง เพื่อการป้องกันปัญหาดังกล่าว ระบบภูมิคุ้มกันจะต้องกำจัดหรือยังยั้งแอนติบอดีตัวที่จะก่อให้เกิดปัญหาทันทีที่มันถูกผลิตออกมา การทำเช่นนี้ได้ ระบบภูมิคุ้มกันจะต้อง

ทราบว่า แอนติบอดีนั้นมีระบบการรู้จำที่เป็นการรู้จำตนเอง หรือไม่ใช่เซลล์ตัวเอง (Self and Non-self Cell Recognition) เพื่อที่จะทำให้ระบบกำจัดแอนติบอดีที่มีการรู้จำแบบเซลล์ตัวเองออกได้ถูกตัว วิธีการนี้เป็นวิธีที่ร่างกายป้องกันการทำลายตัวเองด้วยการควบคุม (Farmer, 1986) อัลกอริธึมของการเลือกเชิงลบดังที่ได้กล่าวมาแล้วนั้น เป็นวิธีการกำจัดแอนติบอดีที่มีการรู้จำแบบเซลล์ตัวเองแบบหนึ่ง วิธีการควบคุมอีกวิธีหนึ่งในการป้องกันการทำลายตัวเองคือ โมเดลโครงข่ายภูมิคุ้มกัน (Immune Network Models)

8.5.1 การทำงานของโครงข่ายภูมิคุ้มกัน

ทฤษฎีนี้ถูกนำเสนอโดยเจอร์เน่ (Jerne, 1974) โดยมีสมมุติฐานว่าระบบภูมิคุ้มกันจะรักษาโครงข่ายไอดิโอไทปิก (Idiotypic network) ที่เชื่อมต่อกันภายในของเซลล์บีเพื่อการรู้จำแอนติเจน เซลล์เหล่านี้จะมีทั้งการกระตุ้น (Stimulation) และการยังยั้ง (Suppression) ซึ่งกันและกันด้วยวิธีการใดวิธีการหนึ่ง ที่ทำให้เกิดความเสถียรของโครงข่าย เมื่อเซลล์บี 2 ตัวจะเชื่อมต่อกัน ถ้าหากว่าค่าความมั่นคงของการจับคู่กันที่เรียกว่า แฟฟินิตี (Affinity) ของพวกมันมีค่ามากกว่าค่าเธรส์โฮล์ด (Threshold) ที่กำหนด ความมีเสถียรภาพก็จะเกิดขึ้น ซึ่งความแข็งแกร่งของการเชื่อมต่อกันภายในโครงข่ายจะเป็นสัดส่วนโดยตรงกับค่าความมั่นคงของการจับคู่กัน เจอร์เน่ได้รับรางวัลโนเบลในปี 1984 ในฐานะที่ได้สร้างประโยชน์ให้กับทฤษฎีโครงข่ายภูมิคุ้มกัน ซึ่งมีขั้นตอนดังนี้

1. การจับคู่กันของแอนติบอดีและแอนติเจน (Antibody and Antigen Binding) แอนติบอดีจะมีพาราโทป 2 อันที่ใช้สำหรับการเลือก (Selection) และการจับคู่ (Matching) กับโมเลกุลอื่น การเลือกและการจับคู่นี้จะเป็นกระบวนการเดียวกัน โดยที่พาราโทปนี้จะจับคู่กับอิพิโทปที่มีรูปร่างเข้ากันได้เท่านั้น เหมือนกับกุญแจที่จับคู่กับลูกกุญแจ (ดังรูปที่ 8.2) ความมั่นคงของการจับตัวกันขึ้นอยู่กับว่าพาราโทปและอิพิโทปเข้ากันได้สนิทแค่ไหน ยิ่งเข้ากันได้สนิทเท่าไรความมั่นคงของการเกาะติกก็จะมีมากเท่านั้น และการรู้จำกันก็จะดีมากขึ้นด้วย
2. การกระตุ้นเซลล์บี (B-cell Stimulation) ที่บริเวณพื้นผิวของเซลล์บีที่เป็นแอนติบอดีนั้น เมื่อแอนติบอดีของเซลล์บีจับตัวกับแอนติเจน เซลล์บีจะถูกกระตุ้น ระดับของการกระตุ้นขึ้นอยู่กับว่าเซลล์บีนั้นจับคู่กับแอนติเจนได้ดีแค่ไหน นอกจากนั้นง เซลล์ทีเฮลเปอร์ก็มีผลต่อเซลล์บีเช่นกัน พาราโทปของเซลล์ทีจะต่างจากของเซลล์บีที่สามารถรู้จำเศษ (Fraction) ของ

แอนติเจนได้ด้วย เศษของแอนติเจนนี้เกิดจากการที่พาราโทปและอิพิโทปจับกันไม่สนิท เพราะรูปร่างของโมเลกุลพาราโทปและอิพิโทปมีความแตกต่างเพียงเล็กน้อย โมเลกุลเหล่านี้เรียกว่าเอ็มเอชซี (MHC: Major Histocompatibility Complex) ในขณะที่ เซลล์ทีไหลเวียนไปตามร่างกาย เซลล์ทีเฮลเปอร์เหล่านี้จะทำการตรวจจับเซลล์ต่างๆ ในร่างกายเพื่อหาแอนติเจน เมื่อเซลล์ทีเฮลเปอร์จับคู่กับของเหลวแอนติเจนได้ เซลล์ทีนี้จะช่วยกระตุ้นเซลล์บีอีกแรงหนึ่ง ถ้าการกระตุ้นของเซลล์บีถึงจุดเธรส์โฮล์ด เซลล์บีก็จะแปลงเป็นเซลล์ตัวอ่อน (Blast Cell) เซลล์ตัวอ่อนเหล่านี้จะเกิดการแบ่งตัวอย่างรวดเร็ว ด้วยการโคลนตัวมันเอง เป็นผลให้เซลล์บีใหม่คัดลอกคุณสมบัติของพาราโทปของเซลล์บีเดิมไว้ กระบวนการนี้เปรียบเสมือนหนึ่งของกระบวนการรู้จำของเซลล์บี

3. กลไกของโครงข่ายภูมิคุ้มกัน (Immune Network Mechanism) ดังที่ได้กล่าวมาแล้วว่าระดับของการกระตุ้นนี้จะขึ้นอยู่กับความมั่นคงในการยึดเกาะกันของเซลล์บีในระบบภูมิคุ้มกัน ซึ่งการยึดเกาะกันนี้ทำให้เกิดโครงข่ายเซลล์บีที่จะรู้จำเซลล์บีตัวอื่นๆ ในระบบ โครงข่ายเหล่านี้จะเป็นแบบเซลฟ์ออกาไนซ์ การอยู่รอดของโครงข่ายขึ้นอยู่กับการจับคู่กันของพวกมันกันเอง และการมีกลไกการป้องกลับ (Feedback) ซึ่งทั้งสองดังกล่าวเป็นตัวสะท้อนความความมั่นคง ยิ่งเซลล์บีมีเซลล์ข้างเคียง (Neighborhood) มากขึ้นเท่าไร การกระตุ้นกันที่ได้จากโครงข่ายก็จะมีมากขึ้นเท่านั้น การอยู่รอดของเซลล์บีขึ้นอยู่กับความมั่นคงของการยึดเกาะกับแอนติเจนที่มันจับตัวอยู่ด้วยที่นอกจากเซลล์ข้างเคียงที่มีอยู่ในโครงข่ายภูมิคุ้มกันอยู่แล้ว เซลล์บีใหม่มีการพัฒนาวิธีการจับคู่กับแอนติเจนและจะแบ่งตัวทำให้มันสามารถอยู่รอดนานกว่าเซลล์บีเดิมได้ด้วยการทำงานตามกระบวนการข้างต้นซ้ำแล้วซ้ำอีกในลักษณะการป้อนกลับดังกล่าว ทำให้ระบบภูมิคุ้มกันเรียนรู้วิธีการจับคู่กับแอนติเจนที่ดีขึ้น

สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับโครงข่ายไอดิโอไทส์ สามารถหาอ่านเพื่อความเข้าใจเพิ่มเติมได้จาก อูเนอร์ (Uner, 2006) ไฮโทเวอร์ (Ron Hightower, 1995) และ ฮันท์ (Hunt, 1996)

8.5.2 โครงข่ายภูมิคุ้มกันประดิษฐ์สำหรับแก้ปัญหาการเดินทางของเซลส์แมน

ถ้าจะเปรียบเทียบระบบภูมิคุ้มกันกับการแก้ปัญหาการเดินทางของเซลส์แมน ตัวเชื้อโรคจะเปรียบเสมือนปัญหาที่เราต้องการแก้ไข กราฟที่แสดงโครงข่ายการเชื่อมต่อระหว่างเมืองเป็นเสมือนเส้นทางที่เซลส์แมนจะต้องเดิน โดยที่เมืองหรือโหนดในกราฟเป็นเหมือนแอนติเจน สำหรับเซลส์แมนซึ่งเป็นเสมือนกับเอเจนต์ของการแก้ปัญหาจะเทียบได้กับเซลล์บี เมื่อเซลส์แมนเดินทางไปถึงเมืองหนึ่ง และจะต้องทำการเลือกเมืองที่จะต้องเดินทางต่อไป เปรียบเหมือนการที่เซลล์บีทำการโคลนตัวเองเพื่อให้เกิดเซลล์บีตัวใหม่ (ซึ่งแตกต่างจากเซลล์แม่ได้) เป็นการสร้างทางเลือกที่เป็นไปได้ให้กับการแก้ปัญหา จากนั้นคำตอบที่เป็นไปไม่ได้หรือเป็นคำตอบที่ไม่ดีจะถูกคัดออก ด้วยการพิจารณาจากค่าแอฟฟิลิเอชัน (Affiliation) ซึ่งเป็นค่าที่แสดงถึงความมั่นคงในการจับคู่กัน เช่นเดียวกับการพิจารณาค่าฟิตเน็สของการค้นหาแบบฮิวริวติก โดยมีค่าเธรส์โฮล์ด (Threshold) เป็นตัวกำหนดขอบเขตสูงสุด ถ้าค่าแอฟฟิลิเอชันสูงกว่าค่าเธรส์โฮล์ดที่กำหนด ทางเลือกนั้นก็จะถูกตัดออก ค่าแอฟฟิลิเอชันนี้คือค่าฟิตเน็สของการหาค่าเหมาะที่สุด สำหรับการเลือกตัดค่าที่มีฟิตเน็สมากว่าค่าเธรส์โฮล์ดออกนั้น เปรียบเสมือนกระบวนการเลือกเชิงลบและบวกของภูมิคุ้มกัน (Bakhouya, 2007) และการเลือกว่าจะตัดออก หรือจะรักษาไว้ขึ้นอยู่กับว่าเราต้องการเลือกคำตอบที่มีค่าฟิตเน็สค่าสูงหรือเลือกค่าต่ำ ถ้าเลือกค่าสูง ค่าที่มากว่าเธรส์โฮล์ดก็จะไม่ถูกตัดออก ซึ่งเป็นการเลือกเชิงบวก แต่ในกรณีนี้เป็นการเลือกเชิงลบ เพราะค่าที่มากกว่าเธรส์โฮล์ดจะถูกตัดออก ในระหว่างการทำงานของระบบภูมิคุ้มกัน การสร้างทางเลือกสำหรับการแก้ปัญหาหรือการเลือกแบบโคลนอล เปรียบเสมือนเอ็กซ์พลอเรชันของการแก้ปัญหา เป็นการสำรวจปริภูมิสถานะที่กว้างขึ้น การที่กระบวนการเลือกเชิงลบที่คัดทางเลือกที่มีค่าแอฟฟิลิเอชันมากกว่าค่าเธรส์โฮล์ดออกเปรียบเสมือนเอ็กซ์พล้อยเตชัน ซึ่งเป็นการเจาะลึกเพื่อค้นหาคำตอบที่เหมาะสม

ถ้ากำหนดว่ากราฟที่แสดงเส้นทางของเมืองที่เชื่อมต่อกัน โดยที่เมืองเป็นโหนด และเส้นทางที่เชื่อมต่อระหว่างเมืองเป็นลิงค์ที่เชื่อมโหนดเข้าหากัน ดังนั้นโหนดต่างๆ ในกราฟจะเป็นแอนติเจน เซลล์บีเป็นเอเจนต์ที่เดินทางจากเมืองไปยังเมืองที่อยู่ข้างเคียง และมีความสามารถในการโคลนตัวเอง หรือทำลายตัวเองขึ้นอยู่กับเงื่อนไขของการเป็น การเลือกเชิงลบ หรือการเลือกเชิงบวก อัลกอริธึมเริ่มต้นด้วยเอเจนต์ดังกล่าว (หรือเซลส์แมน) อยู่ที่เมืองเริ่มต้น ในแต่ละรอบของอัลกอริธึม เอเจนต์นี้จะโคลนตัวมันเอง เอเจนต์ที่เกิดใหม่จะเดินทางไปยังเมืองข้างเคียง เมื่อเอเจนต์เดินทางไปถึงเมืองที่เคยเดินทางผ่าน

มาแล้ว กฎการเลือกเชิงบวกก็จะถูกกระตุ้น และเอเจนต์ก็จะฆ่าโหนดนั้น ถ้าเมืองที่เดินไปไม่ใช่เมืองที่เคยเดินทางผ่านมาแล้ว เอเจนต์หรือเซลล์บีจะโคลนตัวมันเองอีกและเดินทางไปยังเมืองข้างเคียงอีก เมื่อเอเจนต์ทั้งหมดจบการเดินทาง คือเอเจนต์กลับมายังเมืองที่มันเริ่มต้น กฎการเลือกเชิงลบก็จะถูกกระต้น เพื่อหาว่าในระหว่างกลุ่มของเซลล์บีที่เดินทางจนสิ้นสุดแล้ว เซลล์บีตัวไหนมีเส้นทางที่ดีที่สุด

การกำหนดค่าเธรส์โฮล์ดของค่าแอฟฟิลิเอชัน หรือค่าความมั่นคงของการจับคู่กัน ให้พิจารณาจากขอบเขตต่ำสุด (Lower Bound) ของการเดินทาง ขอบเขตต่ำสุดนี้จะทำหน้าที่ในการกำจัดทางเลือกที่ไม่ดีออกไป ที่มีผลเป็นอย่างยิ่งในการลดขนาดของปริภูมิคำตอบ (Solution Space) และลดขอบเขตของการค้นหาคำตอบที่เป็นไปได้

ในทางปฏิบัติของการแก้ปัญหาการเดินทางของเซลส์แมน จะเริ่มจากการเลือกหาเมืองที่ใช้สำหรับการเริ่มต้นเดินทางด้วยการสุ่ม ซึ่งเป็นที่ตั้งของเอเจนต์หรือเซลล์บีในตอนเริ่มต้น จากนั้นเซลล์บีก็จะโคลนตัวเองให้ออกเดินทางไปยังเมืองข้างเคียงทั้งหมด แล้วทำการสำรวจว่า เมืองข้างเคียงที่เซลล์บีใหม่ (ตัวโคลนของเซลล์บี) เดินทางไปถึงทั้งหมดเป็นเมืองที่เคยเดินทางผ่านมาแล้วหรือไม่ และมีระยะทางมากกว่าค่าเธรส์โฮล์ด (หรือขอบเขตต่ำสุด) หรือไม่ ถ้าเป็นจริง ให้เซลล์บีเหล่านั้นกำจัดตัวเองออกไป ด้วยกระบวนการการเลือกโคลนอล (Clonal Selection) สำหรับเซลล์บีที่เหลือให้ดำเนินการหาโหนดข้างเคียงแล้วทำการโคลนตัวเองออกไปอีก โหนดข้างเคียงนี้ หมายถึงโหนดที่มีเส้นทางที่เชื่อมต่อจากโหนดที่เซลล์บีอยู่ในปัจจุบัน ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งเมืองทุกเมืองเป็นเมืองที่เคยเดินทางผ่านมาแล้วทั้งหมด จากนั้นการเลือกเชิงบวก/ลบ (Negative/Positive Selection) ก็จะทำการเลือกเส้นทางที่ดีที่สุด

กำหนดให้ C={a, ..., z} เป็นเซทของเมืองที่เซลส์แมนจะต้องเดินทาง A={(x,y): x,y ∈ C} เป็นเซทของเส้นทางเดินระหว่างเมือง และ □(x,y) เป็นระยะทางระหว่างเมืองของเส้นทาง (x,y) ∈ C ในการหาคำตอบคือการหาเวคเตอร์ของ C={a, ..., z} ที่เป็นเส้นทางเดินที่เป็นไปได้ โดยที่แต่ละตัวในเวคเตอร์ (วงเล็บ)ของ C เป็นเมืองต่างๆ ที่เซลส์แมนจะต้องเดินผ่าน และมีค่าของ □(a, ..., z) สั้นที่สุด สำหรับรายละเอียดการทำงานที่นำเสนอโดยบาคฮัวยา (Bakhouya, 2007) จะเป็นดังนี้

//Initialization

```
Agent.visted = {null}        //รายการของเมืองที่เคยเดินผ่านมาแล้ว
Agent.notVisited = {List of all cities}          //รายการของเมืองที่ไม่เคยเดินผ่าน
Agent.curNode = startNode          //เมืองปัจจุบัน
Agent.pathCost = 0          //ระยะทางจากเมืองเริ่มต้นถึงเมืองปัจจุบัน
Complete = False //เป็น true เมื่อเซลล์บีกลับมาถึงเมืองเริ่มต้น
add agent to AgentList  //เพิ่มเซลล์บีในรายการของเซลล์บี AgentList
(while AgentList is not empty // เมื่อ AgentList ยังมีเซลล์บีอยู่
   (for each agent in AgentList do //สำหรับเซลล์บีแต่ละตัว
      (if Agent.notVisited = {null}  //ถ้ายังมีเมืองที่ยังไม่ได้ผ่านอยู่
         (if (distance(Agent.curNode, startNode) > 0)  //ถ้าไม่ใช่เมืองเริ่มต้น
         then (Agent.pathCost = Agent.pathCost + distance(Agent.curNode, startNode)))
      then (Agent.curNode = startNode);
            (Agent.complete = true);
            (add Agent to Agent.completedList);
            (remove Agent from AgentList);
            (Agent.visitedList = Agent.visitedList U curNode))
   Remove curNode from Agent.notvisited
   Agent.pathCost = Agent.pathCost + distance(Agent.curNode, startNode)
      (If (Agent.pathCost > MaxCost) then (remove Agent from AgentList))
      (If (curNode has successors still to be visted by Agent)
      Then (select a successor city A of curNode at random yet not visited);
             (Agent.curNode = A);
         (If (Agent has more than one successor)
         then (ClonedAgent = AgentCloned);
               (select a successor B of curNode other than A at random yet not visited);
               (ClonedAgent.curNode = B);
               (AgentList = AgentList U ClonedAgent))
      Else (remove Agent from AgentList))))

 (for each Agent in completeList do
```

(If (ninPathCost > Agent.pathCost) then (minPathCost = Agent.pathCost);
SaveAgent = Agent))
return SaveAgent.

การนำระบบภูมิคุ้มกันประดิษฐ์มาประยุกต์ใช้สำหรับการแก้ปัญหาการเดินทางของเซลส์แมนยังมีอีกหลายวิธี แต่ละวิธีมีกระบวนการที่แตกต่างกันเช่น (Endoh, 1998) (Lu, 2007) และ (Sun, 2004) เป็นต้น

8.5.3 โครงข่ายภูมิคุ้มกันประดิษฐ์สำหรับการวิเคราะห์ข้อมูล

เพื่อเป็นตัวอย่างแสดงการใช้งานของการทำงานดังกล่าวเพิ่มเติม ในตอนนี้จะนำเสนอวิธีการวิเคราะห์ข้อมูลโดยใช้ระบบภูมิคุ้มกันประดิษฐ์ ที่นำเสนอโดยทิมมิสและคณะ (Jon Timmis, 2000) ซึ่งมีรายละเอียดดังต่อไปนี้

1. ประชากรของระบบภูมิคุ้มกันประดิษฐ์ (AIS Population) เป็นจำนวนของเซลล์บี (ไม่นับเซลล์ที) ประชากรของระบบภูมิคุ้มกันนี้จะแบ่งออกเป็น 2 ชุด คือประชากรเริ่มต้น และประชากรที่เกิดจากการโคลน ประชากรเริ่มต้นนี้จะสร้างจากเซทย่อยของข้อมูลดิบที่ใช้ในการเทรน (Raw Rraining Data)
2. อ็อปเจ็คของเซลล์บีและการกระตุ้น (B-cell Object and Stimulation) เซลล์บีแต่ละตัวจะแทนหน่วยข้อมูล (Data Item) ชุดหนึ่ง ซึ่งใช้สำหรับการเรียนรู้และมีความสามารถในการตอบสนองต่อการกระตุ้นที่เกิดจากการจับคู่ของหน่วยข้อมูล ในทางธรรมชาติ การตอบสนองต่อการกระตุ้นนี้จะเกิดขึ้นเนื่องจากอิทธิพลของ 3 เรื่องคือ เรื่องแรก ความมั่นคงของการจับคู่กันของเซลล์บีและแอนติเจน ซึ่งสามารถกำหนดขนาดได้ด้วยสมการ 8.7 เมื่อ ps เป็นการจับคู่กันระหว่าง 2 เซลล์ pd เป็นระยะห่างของหน่วยข้อมูล 2 หน่วยที่มีการนอร์มาไลซ์ (Normalize) ค่าให้อยู่ในขอบเขตของ $0 \leq pd \leq 1$ ดังนั้นเมื่อ ค่าของ pd ลดลง จะทำให้ผลของการกระตุ้นของแอนติเจนที่มีต่อเซลล์บีมีมากขึ้น ดังนี้

$$\square\square = (1 - \square\square) \qquad \text{สมการ 8.7}$$

การกระตุ้นอย่างที่สองสำหรับเซลล์บีที่จับคู่อยู่กับเซลล์บีที่อยู่ข้างเคียง จะเป็นไปตามสมการ 8.8 เมื่อ ns เป็นจำนวนที่เซลล์บีถูกกระตุ้นด้วยเซลล์บีที่อยู่ข้างเคียงกับมัน n เป็นจำนวนลิงค์ที่ต่อกับเซลล์ปัจจุบัน และ dis_x เป็นระยะห่างของเซลล์บีข้างเคียงตัวที่ x กับเซลล์บี

$$\square\square = \sum_{\square=0}^{\square}(1 - \square\square\square_{\square}) \qquad \text{สมการ 8.8}$$

ตามที่ได้กล่าวมาแล้วว่า เซลล์บีนี้ไม่ได้ทำหน้าที่เฉพาะการกระตุ้นเท่านั้น แต่ยังทำหน้าที่ในการยับยั้งด้วย การเชื่อมต่อกับเซลล์ที่อยู่ข้างเคียงกันอย่างหลวมๆ ซึ่งการยับยั้งนี้จะถูกนำมาคิดด้วยดังสมการ 8.9 โดยที่ nn เป็นแฟคเตอร์การยังยั้ง (Suppression Factor) ที่เนื่องมาจากเซลล์ข้างเคียงของมัน ตัวแปร dis_x เป็นระยะทางของเซลล์ข้างเคียงตัวที่ x จากเซลล์บีปัจจุบัน โดยที่ ($0 < dis_x < 1$) ดังนั้น

$$\square\square = -\sum_{\square=0}^{\square}\square\square\square_{\square} \qquad \text{สมการ 8.9}$$

การกระตุ้นของเซลล์บีโดยรวมในระบบภูมิคุ้นกันประดิษฐ์ทั้งหมด สามารถรวมกันได้ดังสมการ 8.10 และสามารถเขียนใหม่ได้เป็นสมการ 8.11 ดังนี้

$$\square\square = \square\square + \square\square + \square\square \leq 1 \qquad \text{สมการ 8.10}$$

$$\square\square = 1 - \square\square + \sum_{\square=0}^{\square}(1 - \square\square\square_{\square}) - \sum_{\square=0}^{\square}\square\square\square_{\square}$$

$$\square\square = 1 + \square - \square\square - 2\sum_{\square=0}^{\square}\square\square\square_{\square} \qquad \text{สมการ 8.11}$$

เมื่อ sl ของเซลล์บีมีค่าสูงกว่าเธรส์โฮล์ด เซลล์บีก็จะทำการโคลนิ่ง

3. การโคลนิ่งเซลล์บี (B-cell Cloning) การโคลนของเซลล์บีจะทำให้เกิดเซลล์ใหม่อีกจำนวนมาก และทำให้พาราโทปมีความหลากหลายมากขึ้นในการจับแอนติเจน แต่อย่างไรก็ตามพาราโทปที่มีความสามารถในการจับกับแอนติเจนตัวเดิมก็ยังมีอยู่ ด้วยเหตุผลเช่นนี้เองที่

ทำให้ระบบภูมิคุ้มกันสามารถสร้างภูมิคุ้มกันให้กับร่างกายเพื่อป้องกันการติดเชื้อในอนาคต ในเวลาเดียวกันระบบภูมิคุ้มกันจะโคลนและทำให้เซลล์บีกลายพันธุ์ (Mutate) เพื่อสร้างความจำของเซลล์บีในการจำแนกรูปแบบที่ต่างกันกับเซลล์ที่ทำให้เกิดการโคลน ขบวนการนี้ทำให้เกิดเป็นกลุ่มของเซลล์ที่เหมือนกัน และก่อตัวเป็นคลัสเตอร์ในโครงข่ายที่สามารถจำแนกรูปแบบข้อมูลเฉพาะชนิดได้ และเซลล์ที่สามารถจับตัวอยู่กับโครงข่ายได้เท่านั้นที่จะอยู่รอด ดังนั้นอัตราการโคลนจะเป็นไปตามสมการ 8.12 โดยที่ k เป็นค่าคงที่แบบสเกล่า (Scalar) และ sl(x) คือค่า sl ของสมการ 8.11 ของเซลล์ตัวที่ x ดังนี้

$$\square_{\square} = \square(\square\square(\square))$$ สมการ 8.12

อัตราการกลายพันธุ์นี้จะคงที่ตลอดระยะของการสร้างโครงข่าย เมื่อพิจารณาว่าเมื่อไรจะเกิดการกลายพันธ์ การพิจารณาให้ใช้วิธีการของความน่าจะเป็นในการตัดสินใจ เช่น ตัวอย่างว่าอัตราการกลายพันธุ์เท่ากับ 10% หมายความว่าในข้อมูลชุดหนึ่งๆ ของการโคลนมีความน่าจะเป็น 10% ที่จะเกิดการกลายพันธุ์ และความน่าจะเป็นนี้ให้ใช้ค่าที่เกิดจากการสุ่มมาเป็นตัวกำหนด

4. การเชื่อมต่อกันของเซลล์บี การเชื่อมต่อกันระหว่างเซลล์บีที่เรียกว่า แอฟฟินิตี เป็นการเชื่อมต่อระหว่างเซลล์บี 2 ตัวในโครงข่าย แอฟฟินิตีนี้ถูกกำหนดให้เป็นระยะทางแบบยูคลีเดียน (Euclidean) ของเซลล์บี 2 ตัว ข้อมูลทั้งหมดจะถูกนอมัลไลซ์ให้มีค่าอยู่ระหว่า 1 กับ 0 จากสมการ 8.13 ตัวแปร ND เป็นจำนวนมิติของข้อมูล ตัวแปร a(n) เป็นฟิลด์ในเซลล์บี และ b(n) เป็นฟิลด์ของเซลล์บีที่จับคู่กัน

$$\square\square\square\square\square\square\square\square = \sqrt{\sum_{\square=1}^{\square\square}(\square(\square) - \square(\square))^2}$$ สมการ 8.13

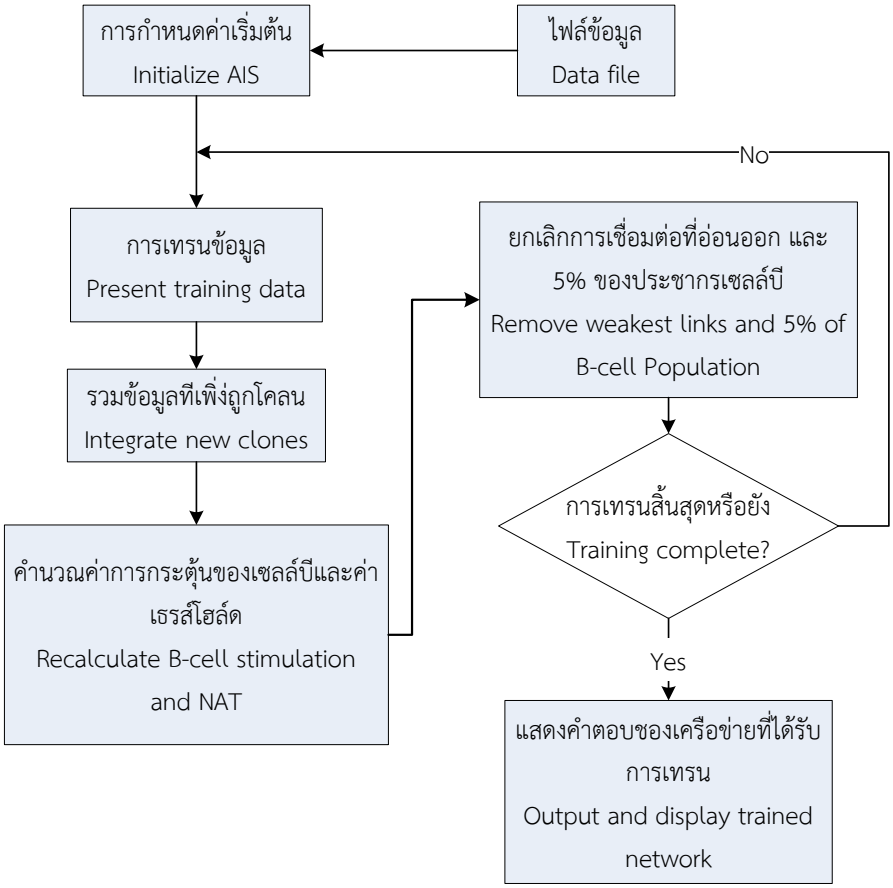
5. ค่าเธรส์โฮล์ดของแอฟฟินิตีในโครงข่าย (Network Affinity Threshold) ค่านี้จะกำหนดตามสมการ 8.14 ซึ่งเป็นค่าเฉลี่ยของค่าแอฟฟินิตีของเซลล์ทุกตัวในเครือข่าย โดยที่ค่า A

เป็นค่าคงที่สเกล่าที่ใช้ในการควบคุมการเชื่อมต่อ มีค่าอยู่ระหว่าง 0 และ 1 และตัวแปร l_i เป็นค่าแอฟฟินิตีของลิงค์ที่ i ในโครงข่ายดังนี้

$$\square\square\square = \square \frac{\sum_{\square=0}^{\square} \square\square\square(\square_{\square})}{\square}$$ สมการ 8.14

6. การสร้างโครงข่าย ในการสร้างโครงข่ายนั้น 5% ของเซลล์ที่มีการเชื่อมต่อที่อ่อนที่สุดจะถูกคัดออก

ซึ่งมีวิธีการในการแก้ปัญหาเป็นไปตามผังงานดังต่อไปนี้



รูปที่ 8.8 อัลกอริธึมของการวิเคราะห์ข้อมูล (Jon Timmis, 2000)

การใช้ระบบภูมิคุ้มกันประดิษฐ์สำหรับการหาค่าเหมาะที่สุดนั้นมีอยู่หลายวิธี แต่ละวิธีก็มีกระบวนการที่แตกต่างกัน ซึ่งจะหาอ่านรายละเอียดเพื่อความเข้าใจเพิ่มเติมได้จาก เรื่องการเดินทางของเซลส์แมน (S., Toma, N. and Yamada, K. Endoh, 1998) การจัดสรรงานให้กับหุ่นยนต์ (Task Allocation) ของ (Amanda Whitbrook, 2012) เป็นต้น

8.6 สรุป

ในการใช้งานระบบภูมิคุ้มกันประดิษฐ์นั้น (Aickelin, 2005) มีสิ่งที่ต้องคำนึงถึงคือ การเข้ารหัส (Encoding) การวัดความเหมือน (Similarity Measure) การเลือก (Selection) และมิวเตชัน (Mutation) โดยที่ก่อนการทำงานของระบบจะเริ่มขึ้น เราจะต้องตัดสินใจให้ได้ก่อนว่าจะมีวิธีการอะไรในการเข้ารหัส ซึ่งก็คือการแทนค่าของปัญหา และการวัดคุณสมบัติของการแทนค่านั้น ในระบบภูมิคุ้มกันการวัดคุณสมบัติของการแทนค่านั้นจะวัดเป็นระดับของความเหมือน จากนั้นกระบวนการของการเลือก และการมิวเตชันจะเกิดขึ้น แบบวนรอบ โดยที่สองกระบวนการนี้จะทำให้ระบบเกิดการวิวัฒนาการในแต่ละรอบของการทำงาน จนกระทั่งการวัดคุณสมบัติจะได้ค่าที่เหมาะสมที่สุดในการแก้ปัญหา หรือจนกระทั่งการทำงานถึงเงื่อนไขของการยุติ

8.6.1 การเข้ารหัส

การเข้ารหัสของระบบภูมิคุ้มกัน จะคล้ายกับการเข้ารหัสของอัลกอริธึมพันธุการ ที่วิธีการเข้ารหัสจะสัมพันธ์โดยตรงกับการหาค่าฟิตเน็สของระบบ ค่าฟิตเน็สในระบบภูมิคุ้มกันนี้ก็คือค่าความมั่นคงของการจับคู่กันหรือการแมทช์ หรือค่าแอฟฟินิตี ดังนั้นในการออกแบบวิธีการเข้ารหัสกับการหาค่าการจับคู่ของระบบจะต้องคิดไปพร้อมกัน ในระบบภูมิคุ้มกันประดิษฐ์ แอนติเจนจะหมายถึงเป้าหมายของการแก้ปัญหา เช่นข้อมูลชุดที่ถูกไวรัสโจมตี หรืออาจจะเป็นชุดข้อมูลที่เราต้องการค้นหาเป็นต้น สำหรับแอนติบอดีก็คือข้อมูลอื่นๆ ที่ไม่ใช่ข้อมูลเป้าหมาย ในการแก้ปัญหาแล้ว ในบางครั้งแอนติเจนอาจมีหลายตัวก็ได้

ในการเข้ารหัสหรือการแทนค่าทั้งแอนติเจนและแอนติบอดีนั้น ส่วนใหญ่แล้วมักจะแทนด้วยสตริงที่เป็นตัวเลข หรือตัวแปรที่บอกคุณลักษณะของแอนติเจนและแอนติบอดีนั้น โดยที่ความยาวของสตริงจะบอกจำนวนคุณลักษณะของแอนติเจนและแอนติบอดี และที่ตำแหน่งต่างๆ ในสตริงนั้นจะเป็น

ตัวแปร ค่าของตัวแปรนั้นอาจเป็นได้ทั้งเลขไบนารี หรือจำนวนจริงก็ได้ ตัวอย่างของสตริงที่เป็นเลขไบนารี เช่น 10010 เป็นต้น

ตัวอย่างของการเข้ารหัสที่ไม่ใช่เลขไบนารี เช่นในการรวบรวมข้อมูลเพื่อการทำเหมืองข้อมูล (Data Mining) สำหรับความเห็นของนักศึกษาต่อวิชาต่างๆ ทีนักศึกษาเรียนอาจมีรูปร่างดังนี้

Student ={{$subj_1$, $Score_1$},{$subj_2$, $Score_2$}...{$subj_n$, $Score_n$}}

จากสตริงที่ใช้แทนข้อมูลเกี่ยวกับการประเมินผลการสอนคือ sub_i เมื่อ i = 1, ..., n คือชื่อวิชาที่ i โดยที่มี $Score_i$ และ i = 1, ..., n คือคะแนนการประเมินที่ sub_i ได้รับจากการประเมินของนักศึกษา ที่มีคะแนนเป็นช่วง ซึ่งอาจจะเป็น 1 ถึง 5 โดยที่ 1 เป็นผลการประเมินต่ำ และ 5 เป็นผลการประเมินสูง ตัวอย่างของสตริงที่มีการประเมินผลแล้วอาจจะเป็นดังนี้

แสวง = {{cpe100, 3}, {mee110, 4}, {cve111, 5}, {gre101,3}}
ทองดี = {{cpe100, 4}, {mee111, 4}, {cve111, 4}, {lng101,3}}

8.6.2 การวัดความเหมือนและแอฟฟินิตี

การวัดความเหมือนและแอฟฟินิตีนั้นสามารถแบ่งได้ออกเป็น 2 กรณี คือกรณีที่สตริงมีค่าเป็นไบนารี และในกรณีที่สตริงมีค่าเป็นตัวเลข

ในกรณีของสตริงที่มีค่าเป็นไบนารีนั้นจะใช้การจับคู่ของสตริง โดยที่ r จะเป็นตัวบอกระดับของการจับคู่ที่ต่อเนื่องกัน ถ้าสตริงทั้ง 2 ชุดมีค่าเหมือนกันทั้งหมด การจับคู่จะเป็นการจับคู่ที่สมบูรณ์ เช่น ถ้ามีสตริงสองชุด ชุดแรกคือ 101011 และชุดที่สองคือ 101011 จะพบว่า สตริงชุดแรกและชุดที่สองเหมือนกันทุกประการ ดังนั้นสตริงทั้งสองชุดเป็นการจับคู่อย่างสมบูรณ์ ในกรณีที่สตริงทั้งสองชุดไม่ได้เป็นสตริงที่แมทช์กันอย่างสมบูรณ์ ค่า r จะบอกระดับของการจับคู่ที่ต่อเนื่องกัน เช่น สตริงชุดแรกคือ 101011 และชุดที่สองคือ 000011 สตริงทั้งสองชุดมีหลักที่ 2,4,5 และ 6 ที่เหมือนกัน แต่มีหลักที่ 4,5 และ 6 ที่เหมือนกันอย่างต่อเนื่อง สำหรับหลักที่สองไม่ต่อเนื่องกับหลักอื่น ดังนั้นสตริง 2 ชุดนี้มี r = 3 เนื่องจากสตริงทั้งสองชุดมีหลักที่เหมือนกันอย่างต่อเนื่องมากที่สุด 3 หลัก ถ้าสตริงทั้งสองชุดคือ

101011 และ 000100 ในกรณีนี้ค่า r = 1 เพราะมีหลักที่สองเพียงหลักเดียวเท่านั้นที่เหมือนกัน สำหรับกรณี 101011 และ 100011 ค่า r = 3 เช่นกัน แม้ว่า 2 หลักแรกจะเหมือนกัน แต่ก็ไม่นำมาคิดเพราะความเหมือนที่ต่อเนื่องกันสั้นกว่า 3 หลักสุดท้าย

ในกรณีที่สตริงของแอนติเจนและแอนติบอดีมีการเข้ารหัสไม่ใช้ไบนารี การวัดจะใช้การวัดระยะห่างของสตริงแบบยูคลีเดียน (Euclidian) จากตัวอย่างในการวัดระยะห่างของการประเมินการสอบดังตัวอย่างข้างต้น ถ้าใช้การวัดของเพียร์สัน (Pearson Measure) เพื่อหาความห่างของความเห็นของนักศึกษา 2 คนที่แทนค่าด้วยตัวแปร u และ v จะได้ค่าของ r ที่บอกระดับความเหมือนเป็นดังนี้

$$r = \frac{\sum_{i=1}^{n}(u_i - u_{avg})(v_i - v_{avg})}{\sqrt{\sum_{i=1}^{n}(u_i - u_{avg})^2 \sum_{i=1}^{n}(v_i - v_{avg})^2}}$$

เมื่อ n คือจำนวนวิชาที่เหมือนกันในการให้คะแนนของทั้ง u และ v จากตัวอย่างการประเมินของนักศึกษาชื่อ แสวง และทองดี ข้างต้น ค่าของ n = 2 เนื่องจากมีวิชาที่เหมือนกันในการให้คะแนนคือวิชา cpe100 และ cve111 ค่า u_i คือคะแนนการประเมินของนักศึกษา u สำหรับวิชาที่ i ที่เหมือนกับนักศึกษา v และ u_{avg} และ v_{avg} คือ เป็นค่าเฉลี่ยของการประเมินทุกวิชาของนักศึกษา u และ v คนตามลำดับ จากสตริงของการประเมินข้างต้น นักศึกษามีการประเมินทั้งหมด 4 วิชาเท่ากัน ในกรณีที่นักศึกษาทั้งสองมีวิชาที่ประเมินไม่เหมือนกันเลยค่าของ r = 0

ผลของการคำนวณค่า r จะได้ออกมามีค่าระหว่าง -1 ถึง 1 สำหรับค่า 1 หมายถึงความเห็นของ u และ v มีความเห็นสอดคล้องกัน ถ้าเป็น -1 หมายถึงความเห็นของ u และ v แตกต่างกัน ถ้าค่าออกมาเป็น 0 หมายถึงความเห็นของทั้งสองไม่มีความสัมพันธ์กัน

ในการพิจารณาเรื่องความเหมือนของสตริง ในกรณีของการเทียบความเหมือนของรีเซฟเตอร์ของแอนติบอดีและแอนติเจนนั้น ถ้าค่าของความเหมือนยิ่งสูง และค่า r ยิ่งมากก็หมายถึงการที่แอนติบอดีสามารถจับกับแอนติเจนได้ดี แต่ในกรณีของการเลือกเชิงลบ เป็นการพิจารณาการที่แอนติบอดีจะจับกับเซลล์ตัวเอง ซึ่งในกรณีนี้ค่าของความเหมือนสูง จะเป็นตัวคัดกรองว่าแอนติบอดีตัวนั้นจะต้องถูกคัดออก

8.6.3 การเลือกแบบโคลนอล และการเลือกเชิงลบ

ตามที่ได้กล่าวมาแล้วว่าการวัดความเหมือนและค่าแอฟฟินิตีนั้นเป็นเครื่องมือที่ใช้สำหรับการคัดเลือกแอนติบอดี ในการคัดเลือกแอนติบอดีในระบบภูมิคุ้มกันนั้นมีอยู่สองกรณีคือ การเลือกแบบโคลนอล และการเลือกเชิงลบ ในการเลือกแบบโคลนอล แอนติบอดีที่มีค่าของแอฟฟินิตีสูงจะถูกเลือก ซึ่งเป็นการเลือกเชิงบวก

สำหรับการเลือกเชิงบวกนั้น เมื่อพิจารณาจากระบบภูมิคุ้มกันประดิษฐ์ จากเมื่อเริ่มต้นเป้าหมายของการทำงานคือเมื่อมีแอนติเจนเข้ามาในระบบ แอนติบอดีจะเข้าไปดักจับ เมื่อเราป้อนแอนติเจนเข้ามาในระบบภูมิคุ้มกันประดิษฐ์ทีละตัว เมื่อเวลาผ่านไป แอนติบอดีที่มีค่าความเข้มข้นของแอฟฟินิตีอยู่จำนวนหนึ่ง แต่ไม่ถึงระดับที่จะจับกับแอนติเจนได้ จะถูกลดความเข้มข้นลง (Death Rate) การลดนี้เมื่อถึงจุดที่ต่ำลงจุดหนึ่ง แอนติบอดีนั้นจะถูกกำจัดออก แต่แอนติบอดีที่มีความความเข้มข้นสูงในการจับกับแอนติเจน ค่าแอฟฟินิตีนี้จะเพิ่มความเข้มข้นขึ้นเรื่อยๆ จนถึงจุดสูงสุดจุดหนึ่ง ในการเพิ่มค่าของความเข้มข้นของแอฟฟินิตีนี้ จะขึ้นอยู่กับการจับตัวกับแอนติเจน ยิ่งการจับคู่หรือการแมทช์ดี ค่าของความเข้มข้นจะสูงขึ้น ซึ่งเปรียบเสมือนการกระตุ้น (Stimulate) ให้เกิดการเพิ่มขึ้นของความเข้มข้น กระบวนการกระตุ้นนี้จะเรียกว่าการโคลน (Cloning) เมื่อจำนวนแอนติบอดีที่เพิ่มเข้ามาในระบบมีมากพอ ระบบก็จะเริ่มวนรอบ (Iterate) ของการลดความเข้มข้น และการกระตุ้นจนกระทั่งแอนติบอดีอย่างน้อยหนึ่งตัวถูกคัดออก และแอนติบอดีใหม่จะถูกเพิ่มเข้ามาจนกระทั่งระบบภูมิคุ้มกันประดิษฐ์เกิดความเสถียร เช่นการถูกคัดออกสิ้นสุดลงเป็นระยะเวลาที่นานในระยะหนึ่ง

ในทางคณิตศาสตร์ ในแต่ละรอบของการทำงาน ที่ระบบเพิ่มความเข้มข้นของแอนติบอดี จากการที่มันจับคู่หรือ แมทช์ กับแอนติเจน เมื่อการจับคู่ไม่เกิดขึ้น ค่าของความเข้มข้นจะลดลงอย่างช้าๆ ที่ ฟาร์เมอร์ (Farmer, 1986) สามารถสร้างเป็นสมการดังนี้

$$\frac{dx_i}{dt} = \left[\begin{pmatrix} antigents \\ recognised \end{pmatrix} - \begin{pmatrix} death \\ rate \end{pmatrix} \right]$$

$$= \left[k_2 \left(\sum_{j=1}^{N} m_{ji} x_i y_j \right) - k_3 x_i \right]$$

N เป็นจำนวนแอนติเจน

x_i เป็นความเข้มข้น (Concentration) ของแอนติบอดี i

y_j เป็นความเข้มข้น (Concentration) ของแอนติบอดี j

k_2 เป็นผลของการกระตุ้น (Stimulation) และ k_3 เป็นอัตราการตาย

m_{ij} เป็นฟังกชันก์การจับคู่ระหว่างแอนติบอดี i และแอนติเจน j

ในการทำงานของระบบภูมิคุ้มกันประดิษฐ์สำหรับการทำเหมืองข้อมูลต่อความเห็นของนักศึกษาเป็นดังนี้

การกำหนดค่าเริ่มต้นของการเข้ารหัสสำหรับความเห็นของนักศึกษาเพื่อการคาดการณ์ ให้เป็นแอนติเจน Ag

While (ระบบยังไม่มีขนาดไม่เต็มที่) และ (ยังมีแอนติบอดี)
(นำเข้าข้อมูลของนักศึกษาคนถัดไปในฐานะแอนติบอดี Ab
คำนวณคะแนนของความเข้มข้นการจับคู่ระหว่าง AG และ Ab
While (ระบบมีขนาดเต็มที่) และ (ระบบไม่เสถียร)
(ลดความเข้มข้นของ Ab ทั้งหมดด้วยจำนวนที่คงที่
แมทช์ค่า Ab กับ Ag และกระตุ้น (Stimulate) ถ้าจำเป็น
))
ใช้แอนติบอดีชุดสุดท้ายในการให้คำตอบ

ในตัวอย่างดังกล่าวข้างต้น ระบบจะเสถียรเมื่อการทำงานเกิดขึ้น 10 รอบโดยที่ไม่มีการเปลี่ยนขนาด ดังนั้นความเสถียรของระบบหมายถึงการมีจำนวนแอนติบอดีหรือคำตอบข้างเคียง (Neighbors) ที่ดีจำนวนเพียงพอ และการคาดการณ์ก็จะเกิดขึ้น สำหรับคำตอบข้างเคียงที่ไม่ดีจะถูกคัดออกจากระบบเมื่อการทำงานผ่านไปเพียงไม่กี่รอบ เมื่อระบบเกิดความเสถียร เราจะใช้ความเข้มข้นของแอนติบอดีในการวัดคำตอบข้างเคียงและจะหาค่าเฉลี่ยเพื่อเป็นคำตอบ

8.6.4 โซมาติกไฮเปอร์มิวเตชัน(Somatic Hypermutation)

การมิวเตชันที่เกิดขึ้นในระบบภูมิคุ้มกันประดิษฐ์นั้น จะคล้ายกับการมิวเตชันที่เกิดขึ้นในอัลกอริธึมพันธุการ เช่นการสลับบางส่วนของสตริงระหว่างสตริง 2 ชุด การเปลี่ยนค่าบางตำแหน่งในสตริงด้วยการสุ่ม หรือการสลับตำแหน่งของสตริงย่อยในสตริง ดังที่ได้กล่าวมาแล้ว นอกจากนั้นแล้ว ในการทำมิวเตชันของระบบภูมิคุ้มกันประดิษฐ์จะมีการนำแนวคิดของโซมาติก (Somatic) มาประยุกต์ใช้ เช่น การจับคู่ยิ่งมาก การทำมิวเตชันยิ่งมาก ในทางตรงกันข้าม ถ้าการพิจารณาที่การจับคู่ต่ำ การทำมิวเตชันยิ่งต่ำ

ในบางกรณี การทำมิวเตชันอาจจะไม่เหมาะสม เพราะจะมีผลทำให้เกิดการเปลี่ยนแปลงข้อมูล เช่น การทำเหมืองข้อมูลดังตัวอย่างการประเมินผลการสอนของวิชาต่างๆ โดยนักศึกษา ดังนั้นการทำมิวเตชันสำหรับการแก้ปัญหาเรื่องการทำเหมืองข้อมูลจึงไม่นิยม

แต่อย่างไรก็ตาม ในการทำมิวเตชันสำหรับปัญหาอื่นๆ นั้นมีประโยชน์มาก ตัวอย่างอัลกอริธึมการเลือกเชิงลบ ที่ใช้สำหรับการตรวจสอบการบุกรุกของไวรัสต่อระบบคอมพิวเตอร์ ในช่วงต้นของการทำงาน การทำมิวเตชันจะช่วยประหยัดเวลาในการค้นหาด้วย

สำหรับการทำเหมืองข้อมูลในบางกรณี การทำมิวเตชันอาจจะเป็นประโยชน์ เช่นการทำเพื่อแบ่งกลุ่มนักศึกษา การหาจุดศูนย์กลางของการแบ่งกลุ่มนั้น อาจจะใช้รหัสข้อมูลเทียมในการจับกลุ่ม ซึ่งการหาจุดศูนย์กลางนี้ การทำมิวเตชันจะเป็นประโยชน์มาก

8.7 แบบฝึกหัด

1. บทบาทของแอนติเจนและแอนติบอดีในระบบภูมิคุ้มกันคืออะไร และทั้งสองมีบทบาทต่อกันอย่างไร

2. อธิบายการทำงานของระบบต่อไปนี้
   2.1. อินเนทอิมมูนิตี (Immate immunity)
   2.2. อะแดพทีฟอิมมูนิตี (Adaptive immunity)

3. การตรวจจับ (Detection) การรู้จำ (Recognition) และการตอบสนอง (Response) เกิดขึ้นในระบบภูมิคุ้มกันได้อย่างไร และเป็นประโยชน์อย่างไรต่อการป้องกันโรคในร่างกายของมนุษย์

4. ในการอธิบายของเจอร์เน่ (Jerne) ที่กล่าวถึงการทำงานของระบบภูมิคุ้มกันมีลักษณะเป็นโครงข่ายสากล ให้อธิบายบทบาทของ แอนติบอดี แอนติเจน เซลล์บี รีเซฟเตอร์ (Receptor) อิพิโทป (Epitope) ทำงานร่วมกันอย่างไรในโครงข่าย ร่วมไปถึงการอธิบายว่า การกระตุ้น (Stimulation) การยับยั้ง (Suppression) การจับตัว (Affiliation) และกลไกการป้องกลับ (Feedback) ว่าเกี่ยวข้องอย่างไรกับการทำงานของระบบภูมิคุ้มกันที่เป็นโครงข่ายไอดิโอปิก (Idiopic network)

5. ให้อธิบายประโยชน์ของสิ่งต่อไปนี้ที่มีต่อระบบภูมิคุ้มกัน
   5.1. การโคลน และการมิวเตชัน
   5.2. แอฟฟินิตี
   5.3. รีเซพเตอร์ พาราโทป อิพิโทป และไอดิโอโทป
   5.4. การกระตุ้น (Stimulation)
   5.5. การยับยั้ง (Suppression)

6. การจับคู่ หรือการแมทช์ ในระบบภูมิคุ้มกันมีกี่ลักษณะ และมีรายละเอียดวิธีการเป็นเช่นไร และกระบวนการจับคู่นี้เกี่ยวข้องอย่างไรกับค่าแอฟฟินิตี

7. การนำโครงข่ายภูมิคุ้มกันประดิษฐ์มาใช้สำหรับการแก้ปัญหาทางคอมพิวเตอร์สามารถทำได้อย่างไร พร้อมทั้งบอกบทบาทของ แอนติบอดี แอนติเจน แอฟฟิลิเอชัน ที่มีต่อการแก้ปัญหา

8. ให้เขียนโปรแกรมแก้ปัญหาการเดินทางของเซลส์แมนด้วยโมเดลโครงข่ายภูมิคุ้มกันประดิษฐ์

9. ทฤษฎีการเลือกโคลนอลของเบอร์เนท อธิบายการเปลี่ยนแปลงของเซลล์ (Differentiation) การโคลน การสร้างเซลล์ความจำ (Memory Cells) และการสร้างเซลล์พลาสมา (Plasma Cells) ไว้อย่างไรในเรื่องต่อไปนี้
   9.1. การทำงานของทั้งระบบ
   9.2. การหาค่าแอฟฟินิตี มีผลอย่างไรต่อการทำงานของทั้งระบบ

10. อธิบายวิธีการที่คาสโตรและซูแบนนำวิธีการเลือกโคลนอลมาใช้สำหรับการแก้ปัญหาทางคอมพิวเตอร์

11. ให้เขียนโปรแกรมเพื่อแก้ปัญหาการเดินทางของเซลส์แมนด้วยอัลกอริธึมการเลือกโคลนอล

12. การเลือกเชิงลบในทางชีวภาพมีลักษณะเป็นอย่างไร ให้อธิบายถึงปฏิสัมพันธ์ระหว่างแอนติเจนและแอนติบอดี ของการเลือกเชิงลบ

13. ให้อธิบายกระบวนการทำงานดังต่อไปนี้ ในระบบการเลือกเชิงลบ
   13.1. การจับคู่หรือการแมทช์แบบสมบูรณ์และบางส่วน
   13.2. วิธีการทำมิวเตชันมีกี่แบบ และการทำโซมาติกไฮเปอร์มิวเตชันคืออะไร แตกต่างจากมิวเตชันอย่างไร

14. ให้อธิบายวิธีการนำการเลือกเชิงลบมาใช้งานตามวิธีการที่ฟอร์เรสต์ (Forrest) นำเสนอ

15. ให้อธิบายบทบาทของ แอนติบอดี และแอนติเจน ในการแก้ปัญหาแบบการเลือกเชิงลบ

16. ออกแบบอัลกอริธึมสำหรับการนำวิธีการเลือกเชิงลบมาใช้ในการทำระบบป้องกันไวรัสในคอมพิวเตอร์

เอกสารอ้างอิง

Ackley D. An empirical study of bit vector function optimization [Journal] // Genetic algorithm and simulated annealing. - 1987. - pp. 170-215.

Ada G. L. The conception and birth of Burnet's clonal selection theory. In Immunology 1930-1980: [Book Section] // In Essays on the History of Immunology / book auth. P. H. Mazumdar. - Toronto : Wall and Thomson, 1989.

Ahmed Z.H. Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator [Journal] // International journal of biometrics and bioinformtics. - 2010. - 6 : Vol. 3. - pp. 96-105.

Aickelin U and Dasgupta, D ARTIFICIAL IMMUNE SYSTEMS [Book Section] // Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques / book auth. Burke E.K. and Kendall, G. - [s.l.] : Springer, 2005.

Aickelin U and Dasgupta, D. ARTIFICIAL IMMUNE SYSTEMS [Book Section] // ." In Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques / book auth. by E.K. and Kendall G Burke. - [s.l.] : Springer, 2005.

Amanda Whitbrook Gabriel Gainham, Wen-Hua Chen, An Idiotypic Network Approach to Task Allocation in the Multi-Robot Domain [Conference] // The 4th International Conference on Agents and Artificial Intelligence (ICAART 2012). - Vilamoura, Algarve, Portugal : [s.n.], 2012. - pp. 5-14.

Andresen B. and Gordon, J. M. On constant thermodynamic speed for minimizing entrophy production in thermodynamic processes and simulated annealing [Journal] // Phys. Rev. E.. - 1994. - Vol. 50. - pp. 4346-4351.

Ashby W.R. Priciple of Self-Organizing Dynamic System [Journal] // Journal of General Psychology. - 1947. - 2 : Vol. 37. - pp. 125-128.

Bakhouya M., and G. Gaber An Immune Inspired-based Optimization Algorithm: Application to the Traveling Salemane Problem [Journal] // Advance Modeling and Optimization. - 2007. - 1 : Vol. 9. - pp. 105-116.

Banharnsakun A., Achalakul, T. and Sirinaovakul, B. ABC-GSX: A hybrid method for solving the Traveling Salesman Problem [Conference] // The World Congress on Nature and Biologically Inspired Computing. - 2010.

Banharnsakun A., Achalakul, T. and Sirinaovakul, B. The best-so-far selection in Artificial Bee Colony algorithm [Journal] // Applied Soft Computing. - 2011. - Vol. 11. - pp. 2888-2901.

Banharnsakun A., Sirinaovakul, B. and Achalakul, T. Job Shop Scheduling with the Best-so-far ABC [Journal]. - [s.l.] : Engineering Applications of Artificial Intelligence, 2012. - 3 : Vol. 25. - pp. 583-593.

Barr A. and Feigenbaum, E.A. The handbook of AI [Book]. - [s.l.] : Addison Wesley, 1982. - Vol. 2.

Bell J.E. and McMullen, P.R. Ant Colony Optimization Techniques for the Vehicle Routing Problem [Journal] // Advanced Engineering Informatics. - 2004. - Vol. 18. - pp. 41-48.

Bendiab E. and Kholladi, M.K. The negative selection algorithm: a supervised learning approach for skin detection and classification [Journal] // International journal of computer science and network security. - 2010. - 11 : Vol. 10. - pp. 86-92.

Blum C. and Roli, A. Metaheuristic in Combinatorial Optimization: Overview and Comceptual Comparison [Journal] // ACM Computing Surveys. - 2003. - 3 : Vol. 35. - pp. 268-308.

Blum C. Ant colony optimization: Introduction and recent trends [Journal] // Physics of Life Reviews. - 2005. - Vol. 2. - pp. 353-373.

Bonabeau E. and Theraulaz, G. Swarm Smarts [Journal] // Scientific American. - 2000. - pp. 72-79.

Bonabeau E., and Théraulaz, G. Swarm Smarts [Journal] // Scientific American. - 2000. - pp. 72-79.

Bonabeau E., Dorigo, M, and Theraulaz, G. Swarm Intelligence: From nature to artificial system [Book]. - [s.l.] : Oxford University Press, 1999.

Bonabeua and Theraulaz G Swarm Smart [Journal] // Scientific American. - 2000. - pp. 72-79.

Box G.P.E. and Muller, M.E. A note on generation of random normal deviates [Journal] // Annals of mathematical statistics. - 1958. - 2 : Vol. 29. - pp. 610-611.

Brassard G. and Bratley, P. Fundamentals of Algorithmics [Book]. - [s.l.] : Prentice-Hall Inc., 1996. - pp. 312-316.

Bryant K. Genetic Algorithms and the Traveling Salesman Problem [Report] : Thesis. - [s.l.] : Department of Mathematics, Harvey Mudd, 2000.

Bullnheimer B., Hart, R.F., and Strau, C. A New Rank Based Version of the Ant System- A Computational Study [Report] : Working Paper No. 1. - [s.l.] : Vienna

University of Economics and administration, University of Vienna and Vienna University of Technology, 1997.

Burnet F. M. A modification of Jerne's theory of antibody production using the concept of clonal selection [Journal] // Australian Journal of Science. - 1957. - Vol. 20. - pp. 67-69.

Burnet F.M. The Clonal Selection Theory of Acquired Immunity [Book]. - Cambridge : Cambridge University Press, 1959.

Camazine S Self-organizing system [Book]. - [s.l.] : Princeton University Press, 2003.

Cao X., Qiao, H., and Xu, Y. Negative selection based immune optimization [Journal] // Advance in engineering software. - 2007. - Vol. 38. - pp. 649-656.

Chang F.J. and Chen, L. Real-Coded Genetic Algorithm for Rule-Based Flood Control Reservoir Management [Journal] // Water Resources Management. - [s.l.] : Kluwer Academic Publishers, 1998. - Vol. 12. - pp. 185–198.

Chen H., Gao, S., Li, S. and Tang, Z. A probabilistic modeling clonal selection algorithm and its applicaiton to traveling salesman problems [Journal] // International journal of computer science and network security. - 2010. - Vol. 10. - pp. 57-62.

Clerc M. and Kennedy, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space [Journal] // IEEE Transactions on Evolutionary Computation. - 2002. - 1 : Vol. 6. - pp. 58-73.

Clerc M. The swarm and the queen: toward a deterministic and adaptive partical swarm optimization [Conference] // Proc. ICEC. - Washington, DC : [s.n.], 1999. - pp. 1951-1957.

Corana A., Marchesi, M., Martini, C., and Ridella, S. Minimizing multimodel functions of continuous variables with the "Simulated Annealing" algorithm [Journal] // ACM Transactions on Mathematical Software. - 1987. - 3 : Vol. 13. - pp. 262-280.

Dagli C.H. and Tatogu, M.Y. An approach to two-dimensional cutting stock problem [Journal] // International journal of production research. - 1987. - Vol. 25. - pp. 175-190.

Dasgupta D., KrishnaKumar, K., Wong, D. and Berry, M. Negative Selection Algorithm for Aircraft Fault Detection [Journal] // Lecture Notes in Computer Science. - 2004. - Vol. 3239. - pp. 1-13.

Dasgupta D., Yu S. and Nino F. Recent advances in artificial immune systems: Models and applications [Journal] // Aplied soft computing. - 2011. - Vol. 11. - pp. 1574-1587.

Dasgupta Dipankar Advance in immune system [Journal] // IEEE Computational Intellignece Magazine. - 2006. - pp. 40-49.

de Casto L.N. and Zuben, F.J.V. Learning and optimization using the clonal selection priciple [Journal] // IEEE Transaction on evolutionary computation. - 2002. - Vol. 6. - pp. 239-251.

de Castro L.N. and Zuben, F.J.V. The clonal selection algorithm with engineering applications [Conference] // GECCO, Workshop on artificial immune systems and their applications. - Las Vegas : [s.n.], 2000. - pp. 36-37.

De Jong K.A. Learning with genetic algorithms: an overview [Journal] // Machine Learning. - 1988. - 2/3 : Vol. 3. - pp. 121-138.

Digalakis J.G., and Margaritis, K.G. ON BENCHMARKING FUNCTIONS FOR GENETIC ALGORITHMS [Journal] // Intern. J. Computer Math.. - 2000. - pp. 1-27.

Doran J. and Michie, D. Experiments with the graph traverser program [Journal] // Proceedings of the Royal Society of London (A). - 1996. - Vol. 294. - pp. 235-259.

Dorigo M. and Gambardella, L.M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem [Journal] // IEEE Transactions on Evolutionary Computation. - 1997. - 1 : Vol. 1. - pp. 53-66.

Dorigo M. and Stützle, T. Ant colony optimization [Book]. - [s.l.] : The MIT Press, 2004.

Dorigo M. Learning and Natural Algorithms (in Italian). PhD thesis [Report]. - Milan, Italy : Dipartimento di Elettronica, Politecnico di Milano,, 1992.

Dorigo M., and Stützle, T. Ant Colony Optimization [Book]. - Cambridge : MIT Press, 2004.

Dorigo M., Maniezzo, V., and Colorni, A. Ant system: optimization by a colony of cooperating agents [Journal] // IEEE Transactions on Systems, Man, and Cybernetics--Part B. - 1996. - 1 : Vol. 26. - pp. 29-41.

Dunlop A. E. and Kernighan, B. W. A procedure forplacement of standard-cell VLSI circuits [Journal] // IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems. - 1985. - 1 : Vols. CAD-4. - pp. 92-98.

Eberhart R.C. and Shi, Y. Comparing inertia weights and constriction factors in particle swarm optimization [Conference] // Proc. CEC. - Sandiego, CA : [s.n.], 2000. - pp. 84-88.

Eiben A.E. and Schippers, C.A. On evolautionary exploration and exploitation [Journal] // Fundamenta informaticae. - 1998. - Vol. 35. - pp. 1-16.

Eiben A.E., Raué, P.-E., and Ruttkay, Zs. Genetic algorithms with multi-parent recombination [Conference] // The 3rd conference on parallel problem solving from nature, LNCS 866. - [s.l.] : Springer-Verlag, 1994. - pp. 78-87.

Endoh S., Toma, N. and Yamada, K. Immune algorithm for n-TSP [Conference] // IEEE International Conference on Systems, Man, and Cybernetics. - 1998. - pp. 3844-3849.

Eshelman L. J. and Schaffer, J. D. Real-coded genetic algorithms and interval-schema, Foundations of Genetic Algorithms 2 [Book]. - [s.l.] : Morgan Kaufmann Publishers, 1993. - pp. 187-202.

Eshelman L. J. The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination, Foundations of Genetic algorithms [Book]. - [s.l.] : Morgan Kaufmann Publishers, 1991.

Farley B.G. and Clark, W.A. Simulation of self organization system by digital computer [Journal] // Infromation Theory, IRE professional group on. - 1954. - 4 : Vol. 4. - pp. 76-84.

Farmer J.D., Packard, N.H. and Perelson, A.S. The immune system, adaptation, and machine learning [Journal] // Physica. - 1986. - pp. 187-204.

Farmer J.D., Packard,N.H. and Perelson, A.S. The immune system, adaptation, and machine learning [Journal] // Physica. - 1986. - D : Vol. 22. - pp. 187-204.

Foerster H. V. Self-Organizing System [Book] / ed. Yovits M.C. and Cameron, S.. - [s.l.] : Pergamon Press, 1960. - pp. 31-50.

Forrest S., Perelson, S., Allen, L., and Cherukuri, R. Self-Nonself discrimination in a computer [Journal] // IEEE Symposium on research in security and privacy. - 1994. - pp. 202-212.

Forsdyke D.R. The Origins of the Clonal Selection Theory of Immunity: A Case Study for Evaluation in Science [Journal] // FASEB Journal. - 1995. - Vol. 9. - pp. 164-166.

French S. Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shopof the Job-Shop [Book]. - [s.l.] : Wiley & Sons., 1982.

Frisch K. V. Bees; their vision, chemical senses, and language [Book]. - Ithaca : Cornell University Press, 1956.

Frisch K.V. The Dance Language and Orientation of Bees [Book]. - [s.l.] : Harvard University Press, 1967.

Gabow H. Implementation of algorithms for maximum matching on nonbipartite graphs [Report] : Ph.D. thesis. - [s.l.] : Stanford University, 1973.

Garrett S.M. How do we evaluate artificial immune system? [Journal] // Evolutionary computation. - 2005. - 2 : Vol. 13. - pp. 145-178.

Geman S. and Geman, D. Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images [Journal] // IEEE Trans. Pattern Anal. Mach. Intell., PAMI-6. - 1984. - Vol. 6. - pp. 721-741.

Gierer A. and Meinhardt, H. A theory of biological pattern formation [Journal] // Kybernetik. - 1972. - Vol. 12. - pp. 30-39.

Glansdorff P. and Prigogine, I., Thermodynamics Theory of Structure, Stability and Fluctuation [Book]. - London : Wiley-Interscience, 1971.

Glover F. Future path for integer programming and links to Artificial Intelligence [Journal] // Computer and Operation Research. - 1986. - Vol. 13. - pp. 533-549.

Goldberg D.E. Genetic Algorithm in Search, Optimization and Machine Learning [Book]. - MA : Addison-Wesley, 1998.

Goldberg D.E. Real-coded genetic algorithms, virtual alphabets, and blocking [Journal] // Complex systems. - 1991. - Vol. 5. - pp. 139-167.

Goss S., Aron, S., Deneubourg, J.-L. and Pasteels, J.M. Self-organized shortcuts in the Argentine ant [Journal] // Naturwissenschaften.. - 1989. - Vol. 76. - pp. 579-581.

Grasse P.P. La reconstruction du nid et les coordinations inter-individuelles chez Bellicosi-termes natalensis et Cubitermes sp. La theorie de la stigmergie: Essai d'interpretation des termites constructeurs [Journal] // In Ins. Soc.. - 1959. - Vol. 6. - pp. 41-83.

Grassé P.P. Termitologia, Tome II [Book]. - Paris : Fondation des Societes, Construction, 1984.

Hansen J., Sato, Mki., Kharecha, P., Beerling, D., Berner, R., Masson-Delmotte, V., Pagani, M., Raymo, M., Royer, D.L. and Zachos, J.C. Target atmospheric CO2: Where should humanity aim? [Journal] // Open Atmos. Sci. J.. - 2008. - Vol. 2. - pp. 217-231.

Hansen P., [Journal].

Hansen P., Mladenovic, N., and Moreno-Perez, J.A. Variable neightbourhood search: methods and application [Journal] // 4OP:A.Q. J. Oper. Res.. - 2008. - pp. 319-360.

Hart P., Nilsson, N. and Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths [Journal] // IEEE Trans. on Systems Science and Cybernetics. - 1968. - 2 : Vol. 4. - pp. 100-107.

Held H. and Karp, R.M. The Travelling Salesman Problem and Minimum Spanning Trees: Part II [Journal] // Math. Programming. - 1971. - 1 : Vol. 1. - pp. 6-25.

Heppner F. and Grenander, U. A stochastic nonlinear model for coordinated bird flocks [Book Section] // The Ubiquity of Chaos / ed. Krasner S.. - [s.l.] : AAAS Publications, 1990.

Herrer F., Lozano, M. and Verdegay, J.L. Tackling Real Coded Genetic Algorithm: Operators and Tools for Behavioural Analysis [Journal] // Artificial Intelligence Review. - [s.l.] : Kluwer Academic Publishers, 1998. - Vol. 12. - pp. 265–319.

Heylighen Francis The science of self organization and adativity [Report] / Center "Leo Apostel". - Belgium : Free University.

Hoffmann K. H. and Salamon, P. The optimal simulated annealing schedule for a simple model [Journal] // J. Phys. A: Math. Gen.. - 1990. - Vol. 23. - pp. 3511-3523.

Holland J. H. Adaptation in Natural and Artificial Systems [Book]. - Ann Arbor, MI : The University of Michigan Press, 1975.

Horowitz E. and Sahni, S. Computing partitions with applications to the knapsack problem [Journal] // Journal of the ACM. - 1974. - 2 : Vol. 21. - pp. 277-292.

Horowitz E. and Sahni, S. Fundamentals of Computer Algorithms [Book]. - Bethesda, MD. : Computer Science Press, Inc., 1978.

Hunt J.E. and Cooke, D.E. Learning using an artificial immune system [Journal] // Journal of network and computer applications. - 1996. - pp. 189-212.

Ingber L. Simulated annealing: Practice versus theory [Journal] // Mathl. Comput. Modelling. - 1993. - 11 : Vol. 18. - pp. 29-57.

Ingber L. Very fast simulated re-annealing [Journal] // Mathl. Comput. Modelling. - 1989. - 8 : Vol. 12. - pp. 967-973.

Jeffrey G. Emergence as a Construct: History and Issues [Journal] // Emergence: Complexity and Organization. - 1999. - 1 : Vol. 1. - pp. 49–72.

Jerne NK Towards a network theory of the immune system Annals of Immunology [Journal] // Annales d' Immunologie de L' Institut Pasteur. - 1974. - c : Vol. 125. - pp. 373-389.

Jerne NK Towards a network theory of the immune system Annals of Immunology [Journal]. - 1974. - C : Vol. 125. - pp. 373-389.

Ji Z. A Boundary-Aware Negative Selection Algorithm [Conference] // IASTED International Conference on Artificial Intelligence and Soft Computing (ASC). - Benidorm, Spain : [s.n.], 2005.

Ji Z. and Dasgupta, D. Revisiting negative selection algorithms [Journal] // Evolutionary computation. - 2007. - 2 : Vol. 15. - pp. 223-251.

Jon Timmis Mark Neal, and John Hunt An artificial immune system for data analysis [Journal] // BioSystems. - [s.l.] : Elsevier, 2000. - Vol. 55. - pp. 143-150.

Karaboga D. An idea based on honey bee swarm for numerical optimization [Report] : Technical report-TR06 / Computer Engineering Department, Engineering Faculty, Erciyes University. - Turkey : [s.n.], 2005.

Karaboga D. An Idea Based On Honey Bee Swarm for Numerical Optimization [Book Section] // Technical Report-TR06. - [s.l.] : Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

Karaboga D. and Akay, B. A compartive study of artificial bee colony algorithm [Journal] // Applied mathematics and computation. - 2009. - pp. 108-132.

Karaboga D. and Basturk, B, A Powerful And Efficient Algorithm For Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm [Journal] // Journal of Global Optimization. - [s.l.] : Springer Netherlands, 2007. - 3 : Vol. 39. - pp. 459-471.

Kellegöz T., Toklu, B. and Wilson. J.M. Comparing efficiencies of genetic crossover operators for one machine total weighted tardiness problem [Journal] // Applied Mathematics and Computation. - 2008. - pp. 590-598.

Kennedy J. and Eberhart, R. A Discrete Binary Version of the Particle Swarm Algorithm [Conference] // Proceedings of the Conference on Systems, Man, and Cybernetics. - 1997. - pp. 4104-4109.

Kennedy J. and Eberhart, R. Particle Swarm Optimization [Conference] // IEEE Int'l. Conf. on Neural Networks. - Perth, Australia : IEEE Service Center, 1995. - pp. IV:1942-1948..

Kennedy J. and Mendes, R. Population Structure and Particle Swarm Performance [Conference] // CEC '02. Proceedings of the 2002 Congress on Evolutionary Computation. - 2002. - Vol. 2. - pp. 1671-1676.

Kennedy J. Small Worlds and Mega-minds: Effects of Neighborhood Topology on Particle Swarm Performance [Conference] // Proceedings of the 1999 Congress on Evolutionary Computation. - 1999. - Vol. 3. - pp. 1931-1938.

Komarudin and Wong K.Y. Applying Ant System for solving Unequal Area Facility Layout Problems [Journal] // European Journal of Operational Research. - 2010. - 3 : Vol. 202. - pp. 730-746.

Koza J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection [Book]. - [s.l.] : MIT Press, 1992.

Kritpatrick S., Gelatt, Jr. C.D., Vecchi, M.P. Optimization by Simulated Annealing [Journal]. - 1983. - 4598 : Vol. 220. - pp. 671-680.

Kundu M.K. and Pal, N.R. Self-Crossover and Its Application to the Traveling Salesman Problem [Journal] // IEA/AIE-99. - 1999. - Vol. LNAI 1611. - pp. 326-332.

Li G., Niu, P. and Xiao, X. Development and investigation of efficient artificial bee colony algorithm for numerical function optimization [Journal] // Appl. Soft Comput. J.. - 2011.

Li G., Niu, P. and Xiao, X. Development and investigation of efficient artificial bee colony algorithm for numerical function optimization [Journal] // Appl. Soft Comput. J.. - 2012. - 1 : Vol. 12. - pp. 320-332.

Lu Jingui, Ning Fang, Dinghong Shao, and Congyan Liu. An Improved Immune-Genetic Algorithm for the Traveling Salesman Problem [Conference] // Third International Conference on Natural Computation. IEEE. - 2007. - pp. 297-301.

Luger G.F, and Stubblefield, W.A. Artificial Intelligence: Structures and Strategies for Complex Problem Solving [Book]. - [s.l.] : Pearson Addison Wesley., 1997.

Lui R.-C., Jiao, L.-C., Du, H.-F. Clonal strategy algorithm based on the immune memory [Journal] // J. Comput. Sci. & Technol.. - 2005. - Vol. 8. - pp. 728-734.

M. Dorigo M., Maniezzo, V. and Colorni, A. Ant System: Optimization by a colony of cooperating agents [Journal] // IEEE Transactions on Systems, Man, and Cybernetics – Part B. - 1996. - 1 : Vol. 26. - pp. 29-41.

M. Dorigo V. Maniezzo, and A. Colorni. Positive feedback as a search strategy. Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1991. M. Dorigo, V. Maniezzo, and A. Colorni. Positive feedback as a search strategy. Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1991. [Report] / Dipartimento di Elettronica ; Politecnico di Milano. - Milan, Italy : Dipartimento di Elettronica, Politecnico di Milano, 1991.

Metropolis N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. Equation of the state calculations by fast computing machines [Journal] // J. Chem. Phys. - 1953. - Vol. 21. - pp. 1087-1092.

Millonas M.M. Swarms, Phase Transitions, and Collective Intelligence [Book Section] // in Artificial Life III / book auth. Langton C.. - MA : Addison-Wesley., 1994.

Moon C., Kim, J., Choi, G. and Seo, Y. An efficient genetic algorithm for traveling salesman problem with precedence constraints [Journal] // European journal of operatinal research. - 2002. - Vol. 140. - pp. 606-617.

Newell A., and Ernst, G. The search for generality [Conference] // IFIP Congress. - 1965. - Vol. 65. - pp. 195-208.

Newell A., Shaw, J. C., & Simon, H. A. Report on a general problem-solving program [Conference] // The International Conference on Information Processing. - Paris : [s.n.], 1960. - pp. 256-264.

Nossal G. J. V. and Lederberg, J. Antibody production by single cells [Journal] // Nature. - 1958. - Vol. 181. - pp. 1419-1420.

Nourani Y. and Andersen, B. A comparison of simulated annealing cooling strategy [Journal] // J. Phys. A: Math. Gen.. - 1998. - Vol. 31. - pp. 8373-8385.

Nulton J. D. and Salamon, P. Statistical mechanics of combinatorial optimization [Journal] // Phys. Rev. A.. - 1988. - Vol. 37. - pp. 1351-1356.

Ólafsson S. Handbook on Simulation, Hnadbooks on in Operations research and management science VII [Book] = Metaheuristics / ed. Henderson Nelson and. - [s.l.] : Elsevier, 2006. - pp. 633-654.

Ono I, Yamamura, M. and Kobayashi, S. A genetic algorithm with characteristic preservation for function optimization [Conference] // Proceedings of the IIZUKA 96. - 1996. - pp. 511-514.

Pasteels J.M., Deneubourg, J.L., and Gross, S. Trail recuitment to newly discovery food sources from individual to collective behaviour in social insects [Book Section] // Self-organization mechanisms in ant societies (I) / book auth. Pasteels J.M., and Deneubourg, J.L.. - Birkhauser, Basel : [s.n.], 1987.

Pearl Judea Heuristics: intelligent search strategies for computer problem solving [Book]. - [s.l.] : Addison-Wesley Longman Publishing Co., Inc., 1984.

Pedersen M.E.H. and Chipperfield, A.J. Simplifying particle swarm optimization [Journal] // Applied Soft Computing. - 2010. - 2 : Vol. 10. - pp. 618-628.

Pelanda R. and Torres, R.M. Receptor editing for better or for worse [Journal] // Curr. Opin. in Immunol. - 2006. - Vol. 18. - pp. 184-190.

Perelson A.S. Immune network theory." Immunal. Rev. [Journal] // Immunal Rev.. - 1989. - Vol. 110. - pp. 5-36.

Poli R. Analysis of the publication on the applications of particle swarm optimization [Journal] // Journal of artificial intelligence and applications. - 2008. - Vol. 2008. - pp. 1-10.

Qi Ji Q., Lan, H. and Ying, T. Improved particle swarm optimization algorithm for Rectangular Cutting-Stock Problem [Conference] // International Conference on Digital Content, Multimedia Technology and its Applications (IDC). - 2010. - pp. 24-27.

Rastrigin L.A. Theoretical Foundations of Engineering Cybernatics Series [Book]. - Moscow : Nauka, 1974.

Reynolds C. W. Flocks, herds and schools: a distributed behavioral model [Journal] // . Computer Graphics, 21(4):25-34.. - 1987. - 4 : Vol. 21. - pp. 25-34.

Rich E. and Knight, K. Artificial Intelligence [Book]. - New York : McGraw-Hill, 1991.

Ron Hightower Stephanie Forrest and Alan S. Perelson The evolution of emergent organization in immune system gene libraries [Conference] // Proceedings of the 6th Conference on Genetic Algorithms. - 1995. - pp. 344-350.

Rosenbrock H.H An automatic method for finding the greastest or least value of a function [Journal] // Computer Journal. - 1960. - Vol. 3. - pp. 175-184.

Samuel A.L. Some studies in machine learning using the game of checkers [Journal] // IBM J. RES. DEVELOP. - 1959. - 1/2 : Vol. 44. - pp. 206-227.

Schaffer J.D. Multiple objective optimization with vector evaluated genetic algorithm [Conference] // The first international conference on genetic algorithm. - 1987.

Seeley T.D. The Wisdom of the Hive [Book]. - Cambridge, MA : Harvard University Press, 1995.

Serugendo G.D.M., Gleizes, M.P., and Karageorgos, A. Self-Organization and emegence in MAS: An overview [Journal] // Information. - 2006. - Vol. 30. - pp. 45-54.

Shannon C. E. Programming a Computer for Playing Chess [Journal] // Philosophical Magazine. - 1950. - Vol. 41. - pp. 256-275.

Shi C.Z., Zeng, N., Zhang, M., Liao, Y.B. and Lai, S.R. Adaptive simulated annealing algorithm for the fiber bragg grating distributed strain sensing [Journal] // Optics communications. - 2003. - pp. 167-173.

Shi Y. and Eberhart, R., A modified particle swarm optimizer [Conference] // IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on Evolutionary Computation. - 1998. - pp. 69-73..

Sirinaovakul B. and Thajchayapong, P. A knowledge base to assist a heuristic search approach to facility layout [Journal] // International Journal of Production Research. - 1994. - Vol. 32. - pp. 141-160.

Sirinaovakul B., and Limudomsuk, T. Maximum weight matching and genetic algorithm for fixed-shaped facility layout problem [Journal] // International Journal of Production Research. - 2007. - 12 : Vol. 45. - pp. 2655-2672.

Socha K. and Dorigo, M. Ant colony optimization for continuous domains [Journal] // European journal of operational reaearch. - 2008. - pp. 1155-1173.

Srikanth T. and kamala, V. A Real Coded Genetic Algorithm for Optimization of Cutting Parameters in Turning [Journal] // International Journal of Computer Science and Network Security. - 2008. - 6 : Vol. 8. - pp. 189-193.

Stibor T., Mohr,P., Timmis, J., and Eckert, C. Is negative selection appropriate for anomaly detection? [Conference] // " Conference on genetic and evolutionary computation. - Washington DC : ACM Press, 2005. - pp. 321-328.

Storn R. and Price, K. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces [Journal] // Journal of global optimization. - 1997. - Vol. 11. - pp. 341-359.

Stützle T. and Hoos, H.H. MAX MIN Ant System [Journal] // Future Generation Computer Systems. - 2000. - Vol. 16. - pp. 889-914.

Sun W.D., Xu., X.-S., Dai., H.-W., Tang, Z. and Tamura, H. An immune optimization algorithm for TSP problem [Conference] // SICE Annual conference in Sapporo. - 2004. - pp. 710-715.

Szu H. and Hartley, R. Fast Simulation Annealing [Journal] // Physics Letters A. - 1987. - Vol. 122. - pp. 157-162.

Talmage D.W. Immunological Specificity: Unique combinations of selected natural globulin provide an alternative to the classical concept [Journal] // Science. - 1959. - Vol. 129. - pp. 1643-1648.

Tam K.Y. Genetic algorithms, function optimization, and facility layout design [Journal] // European Journal of Operational Research. - 1992. - Vol. 63. - pp. 322-346.

Tang K., Li, X., Suganthan, P.N., Yang, Z. and Weise, T. Benchmark function for the CEC'2010 Special session and competition on large-scale global optimization [Conference] // 2010 IEEE world congress on computational intelligence. - Barcelona : [s.n.], 2010.

Trelea I.C. The Particle Swarm Optimization Algorithm: convergence analysis and parameter selection [Journal] // Information Processing Letters. - 2003. - 6 : Vol. 85. - pp. 317-325.

Tsutsui S., Ghosh, A., Corne, D. and Fujimoto, Y. A Real Coded Genetic Algorithm with an Explorer and an Exploiter Populations [Conference] // Proceedings of the 7th International Conference on Genetic Algorithms (ICGA-97). - 1997. - pp. 238-245.

Turing A.M. Computing Machinery and Intelligence [Journal] // Mind. - 1950. - Vol. 59. - pp. 433-460.

Uner A. and Gavalchin, J. Idiotypes [Book Section] // Encyclopedia of life science. - [s.l.] : Jonh wiley and sons, 2006.

Vanderbilt D. and Louie, S.G. A Monte Carlo simulated annealing approach to optimization over cotinuous variables [Journal] // Journal of computational physics. - 1984. - Vol. 56. - pp. 259-271.

Verkoczy L.K., Martensson, A.S. and Nemazee, D. The acope of receptor editing and its association with autoimmunity [Journal] // Curr. Opin. in Immunol. - 2004. - Vol. 16. - pp. 808-814.

Wang K., Huang, L., Zhou, C.-G., and Pang, W. Particle swarm optimization for traveling salesman problem, Proceedings of the Second International Conference on

Machine Learning and Cybernetics, 'Xi'an, 2-5 November 2003. [Conference] // Particle swarm optimization for traveling salesman problem, Proceedings of the Second International Conference on Machine Learning and Cybernetics. - Xi'an : [s.n.], 2003.

Whitley D. A genetic algorithm tutorial [Journal] // Statistics and Computing. - 1994. - Vol. 4. - pp. 65-85.

Wilson E.O. Sociobiology: The new synthesis [Book]. - Cambridge, MA : Belknap Press., 1975.

Wolf T.D. and Holvoet, T. Emergence and self-organization: A statement of similarities and differences [Conference] // International workshop on engineering: Self-organization application. - 2004. - pp. 96-110.

Wong K.Y. and See, P.C. A hybrid ant colony optimization algorithm for solving facility layout problems formulated as quadratic assignment problems [Journal] // Engineering Computations: Int J for Computer-Aided Engineering,. - 2010. - 1 : Vol. 27. - pp. 117-128.

Wright A. Genetic algorithms for real parameter optimization, Foundation of Genetic Algorithms [Book]. - [s.l.] : Morgan Kaufmann Publishers, 1991. - pp. 205-218.

Xu X. and Chen, W. Toward inproved colnal selection algorithm by increasing the diversity degree of antibody [Conference] // Information technology and artificial intelligence conference (ITAIC). - 2011. - pp. 303-344.

YAN G. and LI, C. An Effective Refinement Artificial Bee Colony Optimization Algorithm Based On Chaotic Search and Application for PID Control Tuning [Journal] // Journal of Computational Information Systems. - 2011. - 9 : Vol. 7. - pp. 3309-3316.

Zhang W. and Liu, Y. Reactive power optimization based on PSO in a practical power system [Conference] // IEEE Power Eng. Soc. General Meeting. - 2004. - Vol. 1. - pp. 239-243.

Zhu G. and Kwong, S. Gbest-guided artificial bee colony algorithm for numerical function optimization [Journal] // Applied Mathematics and Computation. - 2010. - Vol. 217. - pp. 3166-3173.

Zhu Y., Gao, S., Dai, H., Li, F. and Tang, Z. Improved clonal selecion algorithm and its application to traveling salesman problem [Journal] // International Journal of Computer Sceince and Network Security. - 2007. - 8 : Vol. 7. - pp. 109-113.

ดัชนีคำศัพท์

2opt method, 266

A* Algorithm, 60

Ackley, 25

Adapting Scanning to Different Representation Types, 147

Adaptive Immunity, 287

Adjacency Based Crossover, 148

Affinity, 293

Allocation Position, 139, 140

Alpha-Beta Cutoff, 72

Annealing, 83, 84, 87, 88, 110

Ant Colony, 19, 163, 166, 194

Ant colony optimization, 9

Ant System, 191

Artificial Bee Colony, 243, 246, 278

Artificial bee colony algorithm, 9

Asymptotic error constant, 22

Asynchronous, 168

B-cells, 286

Beam Width, 60

Best First Search, 44, 48

Best so far ABC, 279

Best-so-far ABC algorithm, 250

Binary graph, 188

Blast Cell, 307

Boltmann Constant, 87

Branch and bound, 50

Breadth First Search, 35, 49, 74

British Museum Algorithm, 44

Cell Mediated Immunity, 287

Central Control, 163

chaotic attractor, 253

Chaotic sequence, 254

Chromosome Encoding, 119

Clonal Selection, 291

Combinatorial Optimization, 2

Complete Search, 44

Constructive method, 5

Control instructions, 11, 30

Convergence, 21, 22, 110, 123, 174, 217, 218

Crossover, 116, 118, 120, 142, 265, 273

Current Generation, 120

Current State, 55, 88

Cybernetics, 8

Decentralized, 18

Depth First Search, 35, 48, 64, 74

Differentiation, 293

Downhill step, 91

Emerge, 12

Employed bees, 246, 249, 267, 269

Employed foragers, 243

Empty Slicing Tree, 138

Epitope, 291

Evaluation value, 130, 133, 158

Fitness, 116, 121, 130, 131, 132, 139, 141, 246

Fitness Based Scanning, 147

Food Source, 246

Game Playing, 68

Game Tree, 70

gbest, 210, 211, 214, 216, 217, 225, 229, 231, 232, 233, 235, 236, 237, 238, 239, 240, 254

Generator, 41, 42

Genetic Algorithm, 115, 136, 139

Genetic Search, 137, 139

Global best, 214, 254

Global Knowledge, 164

Global Optimal, 169

Goal State, 55

Greedy Algorithm, 44

Greedy Subtour, 265

Griewangk, 25

GSX, 265, 267

Heuristic, 2, 30, 35, 39, 40, 44, 55, 74, 190

Heuristic Search, 44

Hill Climbing, 54, 55, 57

Humoral Immunity, 287

Immature convergence, 31

Immune Network Mechanism, 307

Initial Layout, 136, 141

Initial State, 55

Innate Immunity, 287

Intermediate Generation, 120

Job Shop Scheduling, 268

Job shop scheduling problem, 268

Learning Factor, 215

Local Maximum, 59

Local minimum, 23, 24

Local optimum, 84, 199

Local search method, 5

Lowest Horizontal Search Algorithm, 233

Max Play, 69

Mean-End Analysis, 64

Memory Cells, 293

Metaheuristics, 2, 5, 30

MHC: Major Histocompatibility Complex, 307

Minmax, 69

Minmax Value, 69

Modulator, 26

Mutation, 118, 120, 142, 199

Neighborhood solution, 5

Nonseparable function, 27

Numerical Optimization, 3

Objective Function, 118, 142

Occurrence Based Scanning, 146

Onlooker bees, 246, 249, 267, 269

Onlookers, 20, 244

Operator, 55, 66, 118, 128, 139, 140, 223, 229

Optimization, 1, 2, 9, 19, 29, 94, 115, 118, 163, 166, 169, 207, 243

Orientation, 139, 140

Particle Dimension, 215

Particle Range, 215

Particle Swamp, 207

Particle swarm optimization, 9

Particle velocity, 214

pbest, 209, 210, 211, 214, 216, 225, 228, 229, 231, 232, 233, 235, 236, 237, 238, 239, 240

Pheromone, 8, 17, 177, 190

Pheromones, 164, 198

Plant Layout, 157

Plateau, 59

Population, 116, 141

Problem Space, 69, 74, 76, 87, 116, 174, 209

Proliferation, 293

Rank, 122

Rastrigin, 25

Rate of Convergence, 174

Recombination, 119, 120

Recursive, 64

Ridge, 59

Rosenbrock, 24, 283

Roulette Wheel, 121

Schaffer, 25

Scout bees, 246

Scouts, 20, 244

Search space, 22

Search Space, 3, 169

Selection, 116, 118, 119, 121, 122, 123, 152, 156

Self Organization, 7

Self-Organization, 7

Separable function, 24, 27

Simulated annealing, 6, 9

Simulated Annealing, 83, 86, 88

Slicing sequence, 189

Solution space, 6

Sphere, 24, 25, 26

Sphere function, 25, 26

State Space, 64

Static Board, 68

Static Board Evaluation, 69

Static Value, 69, 77

Steady State, 152

Steady-State, 123, 156

stigmergy, 17, 163

Stop Condition, 215

String Coding, 137

Swap Sequence, 224, 230, 231

Swarm Intelligence, 1, 21

Synchronous, 168

System Theory, 9

Tabu Search, 5

T-cells, 286

Tester, 41, 42, 55

Thymus, 286

Tour construction, 176

Traveling Salesman, 46

Travelling Salesman, 3

Tree Encoding, 128

Two Dimensional Stock Cutting, 233

Unemployed foragers, 243

uniform random number, 254

Uphill step, 91, 110

Upper Bound Value, 51

Value Encoding, 127

Waggle dance, 19, 243, 244

เงื่อนไขในการหยุด, 215

เจเนอเรตแอนด์เทสต์, 41

เซลฟ์ ครอสโอเวอร์, 297

เซลฟ์ออกาไนเซชัน, 2, 7

เซลล์ความจำ, 293

เซลล์ตัวอ่อน, 307

เซลล์ที, 286

เซลล์บี, 286

เซลล์มิดิเอ็ด อิมมูนิตี, 287

เพลทู, 59

เมตาฮิวริสติก, 5

เอ็กซ์พลอเรชัน, 22

เอ็กซ์พล้อยเตชัน, 22

เอ็มเอชซี, 307

แฟคเตอร์การเรียนรู้, 215

แอ็ดจาเซนซี เบสด์ ครอสโอเวอร์, 148

แอนติเจน, 286

แอนติบอดี, 285, 286

แอฟฟินิตี, 293

แอลฟา–เบต้าคัตออฟ, 72

โกลบอลมินิมั่ม, 24

โครงข่ายภูมิคุ้มกัน, 307

โซมาติกไฮเปอร์มิวเตชัน, 297

โทโพโลยีของค่าข้างเคียง, 222

โพซิชันเบสครอสโอเวอร์, 273

โลคอลแม็กซิมั่ม, 59

โลคอลมินิมั่ม, 24

ไดเวอร์สิฟิเคชัน, 22

ไบฟาร์เคชัน, 14

กระดานสถิต, 68

กระบวนการแบ่งตัวอย่างรวดเร็ว, 293

การเข้ารหัสแบบเปอร์มิวเตชัน, 126

การเข้ารหัสแบบโครงสร้างต้นไม้, 128

การเข้ารหัสแบบไบนารี, 124

การเข้ารหัสแบบค่าจำนวนจริง, 149

การเข้ารหัสโครโมโซม, 119

การเดินทางของเซลส์แมน, 3, 132, 175, 229, 264, 308

การเต้นรำแบบส่ายท้อง, 20, 244, 245

การเปลี่ยนแปลงเซลล์, 293

การเล่นเกม, 68

การเลือกคลอนอล, 291

การเลือกสรรแบบจัดลำดับ, 122

การเลือกสรรแบบวงล้อรูเล็ตต์, 121

การเลือกสรรโครโมโซม, 119

การแก้ปัญหาแบบสมบูรณ์, 3

การใช้งานของหุ่นยนต์, 66

การค้นหาแบบกว้างก่อน, 38

การค้นหาแบบที่ดีที่สุดก่อน, 48

การค้นหาแบบปีม, 60

การค้นหาแบบลึกก่อน, 35

การค้นหาแบบฮิลไคลมิ่ง, 54

การค้นหาแบบฮิวริสติก, 39

การค้นหาคำตอบท้องถิ่น, 171

การค้นหาพันธุการ, 141

การครอสโอเวอร์, 126

การครอสโอเวอร์แบบหลายพ่อแม่พันธุ์, 145

การควบคุมแบบกระจาย, 286

การจัดตารางการผลิตสำหรับการผลิตแบบทำตามสั่ง, 268

การจัดวางแพทเทิร์น, 96

การจับคู่, 285

การตอบสนองทางภูมิคุ้มกันแบบจำเพาะ, 286

การตัดสต็อกแบบสองมิติ, 233

การปฏิสัมพันธ์พหุคูณ, 246

การปรับค่าฟีโรโมน, 171

การป้อนกลับเชิงบวก, 246

การป้อนกลับเชิงลบ, 246

การมิวเตชัน, 125

การมิวเตชันของโครโมโซมที่เข้ารหัสแบบเปอร์มิวเตชัน, 127

การลู่เข้าสู่คำตอบ, 21, 22, 217

การลู่เข้าหาคำตอบ, 172

การวางผังงานโดยอาศัยอัลกอริธึมพันธุการ, 136

การสร้างสไลซิ่งทรี, 137

การสำรวจปริภูมิค้นหา, 22

การหมุนเวียนของเรย์ลี เบนนาร์ด, 10

การหาค่าเหมาะที่สุด, 1

การหาค่าเหมาะที่สุด (Optimization), 1

การหาค่าเหมาะที่สุดเชิงตัวเลข, 3

การหาค่าเหมาะที่สุดแบบกลุ่มอนุภาค, 9, 207

การหาค่าเหมาะที่สุดแบบคอมบินาทอเรียล, 2

การหาค่าเหมาะที่สุดแบบฝูงผึ้งประดิษฐ์, 243

การหาค่าเหมาะที่สุดแบบฝูงมด, 163

การหาค่าที่เหมาะที่สุดแบบคอมบินาทอเรียล, 169

การอบอ่อน, 83

ขั้นตอนการแลกเปลี่ยนยีน, 119

ความแปรปรวน, 246

ค่ามินแม็กซ์, 69

คำตอบข้างเคียง, 5

จำนวนอนุภาค, 215

ช่วงของอนุภาค, 215

ซีมูเลตเต็ดอันนีลลิ่ง, 83

ซีมูลเลเต็ด แอลนิลลิง, 9

ต่อมไทมัส, 286

ตัวดำเนินการเลือกสรร, 118

ตัวดำเนินการครอสโอเวอร์, 118

ตาบู, 5

ตารางอบอ่อน, 87

น้ำหนักอิเนอร์เทีย, 220

บรานช์แอนด์เบานด์, 50

ปริภูมิสถานะ, 64

ปัญญาเชิงกลุ่ม, 1

ปัญหา 8-Puzzle, 62

ผึ้งเอ็มพลอย, 20, 246

ผึ้งค้นหา, 20, 244, 246

ผึ้งรับสาร, 20, 244, 246

ฟังก์ชันเทียบเคียง, 23

ฟังก์ชันเป็นอิสระต่อกัน, 24

ฟังก์ชันแยกได้, 24

ฟังก์ชันของเชฟเฟอร์, 25

ฟังก์ชันของเรสตริจิน, 25

ฟังก์ชันของแอคเลย์, 25

ฟังก์ชันของโรเซนบร็อก, 24

ฟังก์ชันของกรีวังค์, 25

ฟังก์ชันยูนิโมเดล, 24

ฟังก์ชันรูปทรงกลม, 24

ฟิตเน็ส เบสด์ สแกนนิ่ง, 147

ภูมิคุ้มกันประดิษฐ์, 285

มัลติโมเดล, 24

มิติของอนุภาค, 215

มีน-เอ็นด์อะแนไลซิส, 64

ระบบมดอีลีท, 191

ริดจ์, 59

วิธีการ ทูอ็อป, 266

วิธีการค้นหาแบบโลคอล, 5

วิธีคอนสตรัคทีฟ, 5

สติกเมอร์จี, 17

สัมประสิทธิ์การหดตัว, 221

อ็อคเคอร์เร้นซ์ เบสด์ สแกนนิ่ง, 146

อะแด็บติ่ง สแกนนิ่ง สำหรับการแทนค่ายีนที่ต่างกัน, 147

อะแดพทีฟอิมมูนิตี, 287

อัลกอริธึมเบสโซฟา เอบีซี, 250

อัลกอริธึมเอ-สตาร์, 60

อัลกอริธึมกรีดี้, 44

อัลกอริธึมของเมโทรโปลิส, 86

อัลกอริธึมของสตีเพสต์-แอสเซนต์ฮิลไคลมิ่ง, 57

อัลกอริธึมของฮิลไคลมิ่งธรรมดา, 55

อัลกอริธึมพันธุการ, 9, 115

อัลกอริธึมพันธุการขั้นสูง, 144

อิเมอร์เจนซ์, 13

อินเทนสิฟิเคชัน, 22

อินเนทอิมมูนิตี, 287

อิพิโทป, 291

อิมมูโนโกลบูลิน, 286

ฮิวริสติก, 4

ฮิวริสติกฟังก์ชัน, 40
ฮูมูรอลอิมมูนิตี, 287, 288