บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้จะกล่าวถึงเอกสารและงานวิจัยที่เกี่ยวข้องกับการประยุกต์ใช้ GA ในการจัดเรียงกล่องลงในคอนเทนเนอร์ ซึ่งประกอบด้วย 1. ปัญหาการจัดเรียงกล่องลงในคอนเทนเนอร์ (Container packing problem: CPP), 2. วิธีการหาค่าคำตอบที่ดีที่สุด (Optimization algorithms), 3. การออกแบบการทดลอง (The design of the experiment) และการวิเคราะห์ข้อมูลทางสถิติ (The statistical analysis of the data) และ 4. ตัวอย่างงานวิจัยที่เกี่ยวข้องและบทวิจารณ์งานวิจัย

1. ปัญหาการจัดเรียงกล่องลงในคอนเทนเนอร์ (Container packing problem: CPP)

ในหัวข้อนี้จะแบ่งออกเป็น 2 ข้อย่อย คือ 1.1 ประเภทของปัญหาการจัดเรียงกล่องลงในคอนเทนเนอร์ และ 1.2 ฮิวริสติกที่ใช้ในการจัดเรียงกล่อง (Heuristic for arrangement)

1.1 ประเภทของปัญหาการจัดเรียงกล่องลงในคอนเทนเนอร์

ปัญหาการจัดเรียงกล่องลงในคอนเทนเนอร์ จัดเป็นประเภทหนึ่งของปัญหาการตัดและการบรรจุ (Cutting and packing problem) (Dyckhoff, 1990) ซึ่งปัญหาการตัดและการบรรจุนี้มีมากมายหลายแบบจนทำให้เกิดความสับสนในลักษณะโครงสร้างและรูปแบบของปัญหา ดังนั้น Dyckhoff (1990) จึงได้พัฒนาแนวทางในการแบ่งประเภทของปัญหาซนิดนี้ด้วยการสร้างรูปแบบมาตรฐานของปัญหาการตัดและการบรรจุให้แบ่งเป็นประเภทที่ชัดเจน โดยใช้สัญลักษณ์ลำดับอักษร 4 ตัวคือ (α / β / γ / δ) ซึ่ง α หมายถึง ขนาดของมิติ, β หมายถึง การกำหนดซนิด, γ หมายถึง ลักษณะของวัตถุขนาดใหญ่ และ δ หมายถึง ลักษณะของวัตถุขนาดย่อย โดยมีรายละเอียดดังตารางต่อไปนี้

ตาราง 1 แสดงสัญลักษณ์ลำดับอักษร 4 ตัวคือ (α / β / γ / δ) (Dyckhoff, 1990)

1) แสดงความหมายของสัญลักษณ์ลำดับอักษรที่ 1 (α) แทนขนาดของมิติ (Dimensionality)	
สัญลักษณ์	ความหมาย
(1)	วัตถุมิติเดียว (One-dimensional) เป็นรูปแบบของการตัดหรือการบรรจุวัตถุในแนวเดียว โดยคำนึงถึงเฉพาะช่วงความยาวในการตัดเท่านั้น เช่น การตัดท่อโลหะ เป็นต้น
(2)	วัตถุสองมิติ (Two-dimensional) เป็นรูปแบบของการตัดวัตถุรูปสี่เหลี่ยมมุมฉากขนาดต่างๆจากวัตถุรูปสี่เหลี่ยมมุมฉาก หรือการบรรจุวัตถุรูปสี่เหลี่ยมมุมฉากขนาดต่างๆลงในพื้นที่รูปสี่เหลี่ยมมุมฉาก เช่น แผ่นโลหะ หรือแผ่นพลาสติก เป็นต้น
(3)	วัตถุสามมิติ (Three-dimensional) เป็นรูปแบบของการตัดวัตถุรูปสี่เหลี่ยมลูกบาศก์ขนาดต่างๆจากวัตถุรูปสี่เหลี่ยมลูกบาศก์ หรือการบรรจุวัตถุรูปสี่เหลี่ยมลูกบาศก์ขนาดต่างๆลงในภาชนะรูปสี่เหลี่ยมลูกบาศก์ เช่น การตัดแท่งโลหะ หรือการบรรจุกล่อง เป็นต้น
(N)	มากกว่าสามมิติ (N-dimensional with N>3) ซึ่งเป็นรูปแบบของการตัดหรือการบรรจุให้อยู่ในรูปแบบของสมการทางคณิตศาสตร์ที่มีตัวแปรมากกว่าสามตัวแปร
2) แสดงความหมายของสัญลักษณ์ลำดับอักษรที่ 2 (β) แทนการกำหนดชนิด (Kind of assignment)	
(B)	กำหนดวัตถุขนาดใหญ่ทั้งหมด และเลือกวัตถุขนาดย่อย (All objects and a selection of items)
(V)	เลือกวัตถุขนาดใหญ่ และวัตถุขนาดย่อยทั้งหมด (A selection of objects and all items)
3) แสดงความหมายของสัญลักษณ์ลำดับอักษรที่ 3 (γ) แทนลักษณะของวัตถุขนาดใหญ่ (Assortment of large objects)	
(O)	วัตถุขนาดใหญ่ชิ้นเดียว (One object)
(I)	วัตถุขนาดใหญ่หลายชิ้นที่มีขนาดเท่ากัน (Identical figures)
(D)	วัตถุขนาดใหญ่หลายชิ้นที่มีขนาดแตกต่างกัน (Different figures)

ตาราง 1 (ต่อ)

4) แสดงความหมายของสัญลักษณ์ลำดับอักษรที่ 4 (δ) แทนลักษณะของวัตถุขนาดย่อย (Assortment of small items)	
(F)	**วัตถุขนาดย่อยมีจำนวนน้อย และมีขนาดที่แตกต่างกันน้อย** (Few items of different figures)
(M)	วัตถุขนาดย่อยมีจำนวนมาก และมีขนาดที่แตกต่างกันมาก (Many items of many different figures)
(R)	วัตถุขนาดย่อยมีจำนวนมาก และมีขนาดที่แตกต่างกันน้อย (Many items of relatively few different (non-congruent) figures)
(C)	วัตถุขนาดย่อยมีจำนวนมาก และมีขนาดที่ไม่แตกต่างกัน (Congruent figures)

จากตารางข้างต้นนี้ สามารถแยกลักษณะของปัญหาการตัดและการบรรจุได้ทั้งสิ้น 4 × 2 × 3 × 4 เท่ากับ 96 ลักษณะ (Dyckhoff, 1990) โดยการใช้ตารางมาตรฐานดังกล่าวสามารถที่จะบ่งบอกถึงประเภทของปัญหาได้อย่างสะดวกและชัดเจน แสดงตัวอย่างประเภทของปัญหาการตัดและการบรรจุได้ดังตาราง 2 (* เนื่องจากปัญหาหนึ่งๆมีรูปแบบการกำหนดเงื่อนไขของปัญหาไม่ตายตัวขึ้นอยู่กับนักวิจัยว่าสนใจจะกำหนดเงื่อนไขของปัญหาเป็นแบบใด ดังนั้น * คือ สามารถเลือกเงื่อนไขเป็นแบบใดก็ได้ตามสัญลักษณ์ลำดับอักษรนั้นๆ)

ตาราง 2 แสดงตัวอย่างประเภทของปัญหาการตัดและการบรรจุ (Dyckhoff, 1990)

ปัญหา	ประเภทของปัญหา
(Classical) knapsack problem	1 / B / O / *
Pallet loading problem	2 / B / O / C
More-dimensional knapsack problem	* / B / O / *
Dual-bin packing problem	1 / B / O / M
Vehicle loading problem	1 / V / I / F or 1 / V / I / M
Container loading problem	3 / V / I / * or 3 / B / O / *
(Classical) bin packing problem	1 / V / I / M

ตาราง 2 (ต่อ)

ปัญหา	ประเภทของปัญหา
Classical cutting stock problem	1 / V / I / R
2-dimensional bin packing problem	2 / V / D / M
Usual 2-dimensional cutting stock problem	2 / V / I / R
General cutting stock or trim loss problem	1 / * / * / * or 2 / * / * / * or 3 / * / * / *
Assembly line balancing problem	1 / V / I / M
Multiprocessor scheduling problem	1 / V / I / M
Memory allocation problem	1 / V / I / M
Change making problem	1 / B / O / R
Multi-period capital budgeting problem	n / B / O / *

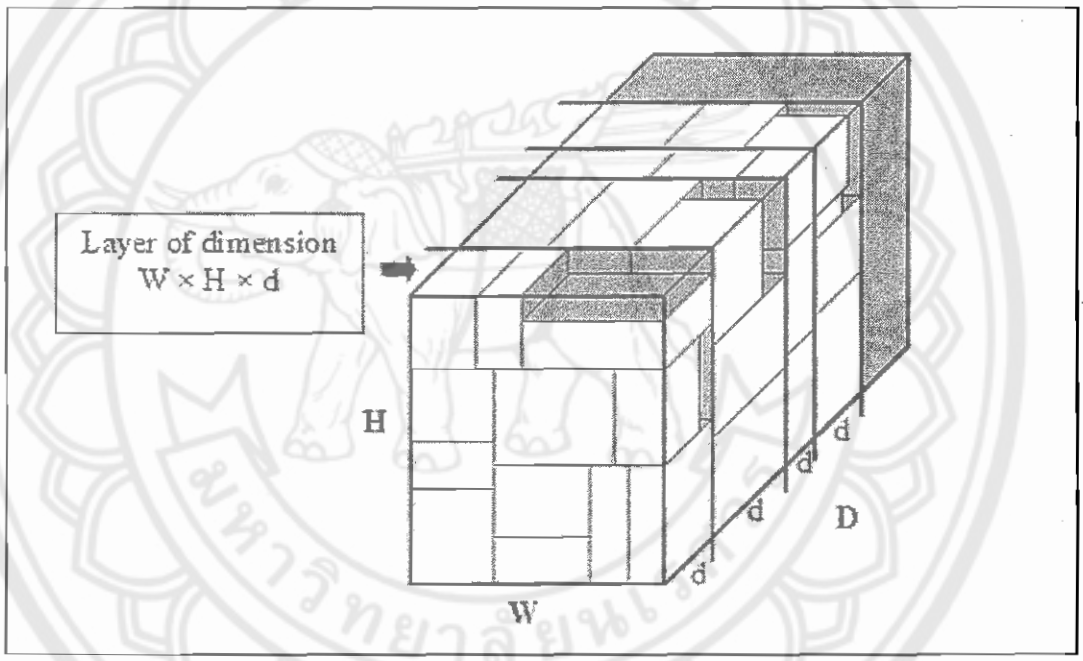
ในงานวิจัยนี้ได้จัดรูปแบบมาตรฐานของ CPP โดยใช้สัญลักษณ์ที่ Dyckhoff (1990) ได้เสนอไว้เป็นแบบ 3 / V / I / M คือ ลักษณะของปัญหาเป็นแบบวัตถุสามมิติ ซึ่งจะเลือกวัตถุขนาดใหญ่ และวัตถุขนาดย่อยทั้งหมด วัตถุเป็นวัตถุขนาดใหญ่หลายชิ้นที่มีขนาดเท่ากัน และมีวัตถุขนาดย่อยจำนวนมากที่มีขนาดแตกต่างกันมาก (ในที่นี้ให้วัตถุขนาดใหญ่ คือ คอนเทนเนอร์ และวัตถุขนาดย่อย คือ กล่อง)

1.2 **ฮิวริสติกในการจัดเรียงกล่อง** (Heuristic for arrangement)

CPP จัดเป็นปัญหาแบบ NP-hard คือ เป็นปัญหาที่ยากต่อการหาคำตอบที่เหมาะสม ยกเว้นเมื่อตัวอย่างขนาดของปัญหานั้นมีขนาดเล็กก็จะทำให้การหาคำตอบนั้นง่ายขึ้น แต่ในชีวิตจริงแล้วขนาดของปัญหานั้นมักจะมีขนาดใหญ่ ดังนั้นจึงต้องมีการประยุกต์ใช้วิธีการฮิวริสติกต่างๆ เข้ามาช่วยแก้ปัญหา โดยปกติแล้วฮิวริสติกที่ใช้ในการจัดเรียงกล่องสามารถที่จะแยกได้หลายประเภท เช่น Stack building approach, Wall- building approach, Guillotine cutting approach และ Cuboid arrangement approach (Pisinger, 2002) อธิบายวิธีการฮิวริสติกแบบต่างๆดังนี้

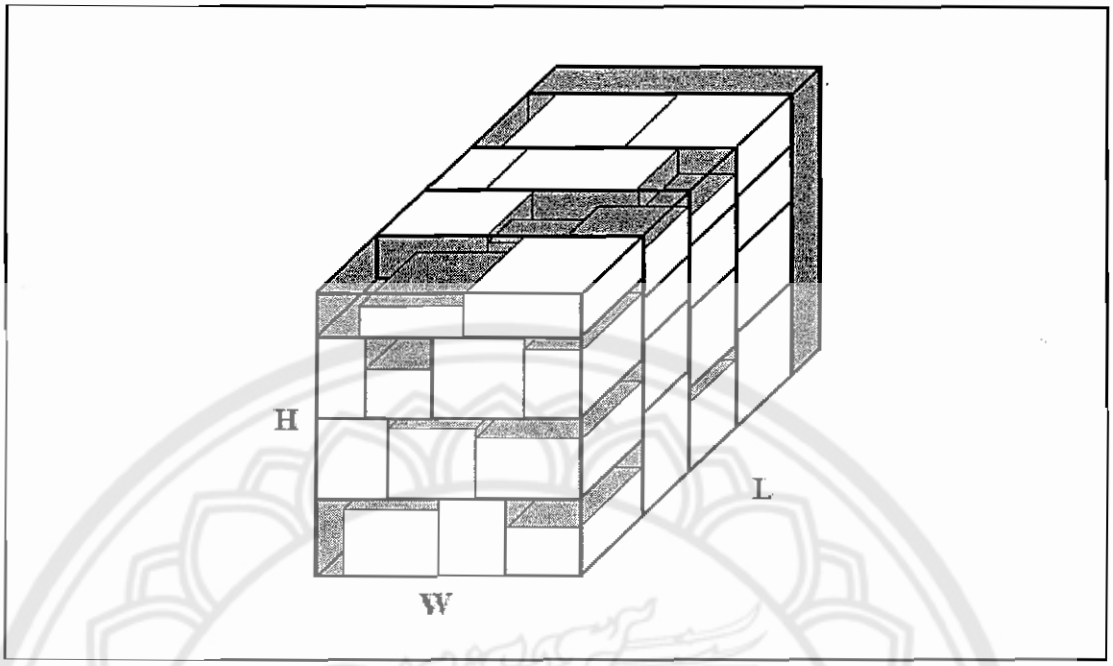
Stack building approach ถูกนำเสนอโดย Gilmore and Gomory (1965) วิธีการจะคล้ายกับ Wall-building approach แต่จะต่างกันตรงที่การบรรจุกล่องแต่ละใบจะทำการจัดเรียงไปตามพื้นของคอนเทนเนอร์ จากนั้นจะบรรจุแต่ละชั้น (Stack) ให้ขนานไปกับพื้นของคอนเทนเนอร์

Wall-building approach ถูกนำเสนอโดย George and Robinson (1980) และถูกนำไปใช้โดยนักวิจัยหลายท่านได้แก่ Bischoff and Marriott (1990), Gehring, Menscher and Meyer (1990), และ Hemminki (1994) วิธีการทำ Wall-building approach คือ นำกล่องแต่ละใบมาจัดเรียงตามแบบผนังกำแพงซึ่ง 1 กำแพงคือ 1 ชั้น จากนั้นบรรจุแต่ละชั้น (Layer) ลงในคอนเทนเนอร์ตามแนวขวางของด้านลึก (Depth: D) ของคอนเทนเนอร์ แสดงภาพประกอบดังภาพ 1



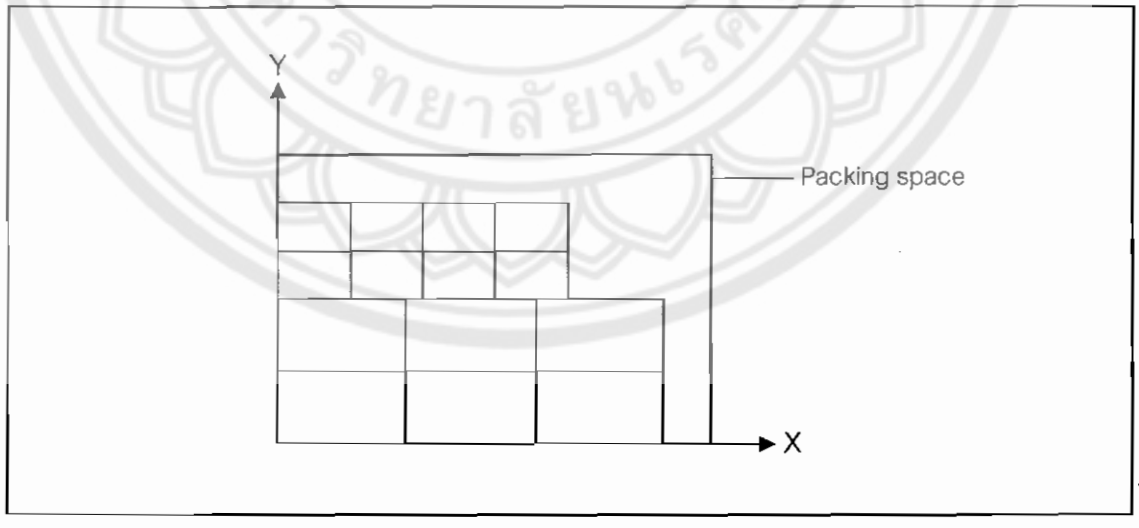
ภาพ 1 แสดงการบรรจุแบบ Wall-building approach (George & Robinson, 1980)

Guillotine cutting approach ถูกนำเสนอโดย Morabito and Arenales (1994) การบรรจุแบบ Guillotine คือการบรรจุโดยที่ปริมาตรของคอนเทนเนอร์จะถูกแบ่งออกเป็นชั้นย่อยๆ (Sub-Layer) โดยความสูง (Height) หรือความยาว (Length) ของแต่ละชั้นจะถูกกำหนดจากความสูง หรือความยาวของกล่องที่สูงหรือยาวที่สุด ที่ถูกจัดเรียงเข้าไปในชั้นนั้น แสดงภาพประกอบดังภาพ 2



ภาพ 2 แสดงการบรรจุแบบ Guillotine cutting approach

Cuboid arrangement approach ถูกนำเสนอโดย Bortfeldt and Gehring (1997) จะเป็นการบรรจุกล่องลงในคอนเทนเนอร์โดยพยายามจัดเรียงกล่องที่มีลักษณะคล้ายคลึงกันให้เป็นรูปลูกบาศก์ แสดงภาพประกอบดังภาพ 3



ภาพ 3 แสดงการบรรจุแบบ Cuboid arrangement approach (Bortfeldt & Gehring, 1997)

2. วิธีการหาค่าคำตอบที่ดีที่สุด (Optimization algorithms)

วิธีการที่นำมาใช้แก้ปัญหาในการหาค่าที่ดีที่สุด (Optimization algorithms) สามารถจำแนกได้เป็น 2 ประเภท คือ วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการทางคณิตศาสตร์ (Conventional optimization algorithms) และวิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักของการประมาณ (Approximation optimization algorithms) (Blazewicz, Domschke & Pesch, 1996a; Blazewicz et al., 1996b; Nagar, Haddock & Heragu, 1995)

2.1 วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการทางคณิตศาสตร์ (Conventional optimization algorithms: COAs)

วิธีการในกลุ่มนี้ถูกพัฒนาขึ้นมาในสมัยสงครามโลกครั้งที่สอง โดยมีจุดประสงค์ในการแก้ปัญหาทางด้านการทหารที่มีความซับซ้อน (Pongcharoen, 2001) หลังจากนั้นวิธีการเหล่านี้ได้ถูกนำไปใช้แก้ปัญหาต่างๆอย่างแพร่หลาย เช่น ปัญหาด้านการจัดตาราง (Scheduling problems) หรือปัญหาการบรรจุกล่องผลิตภัณฑ์ลงในคอนเทนเนอร์ (Container packing problem: CPP) เมื่อศึกษาจากงานวิจัยหลายๆงานทำให้ทราบว่าวิธีการที่นำมาใช้ในการหาค่าคำตอบนั้นมีอยู่หลายวิธียกตัวอย่างเช่น วิธีโปรแกรมเชิงเส้น (Integer linear programming), วิธีโปรแกรมเชิงพลวัต (Dynamic programming) และวิธีบรานช์แอนด์บาวด์ (Branch-and-bound algorithm) เป็นต้น ซึ่งมีนักวิจัยหลายท่านที่นำวิธีการในกลุ่ม COAs นี้ไปใช้แก้ปัญหาเช่น งานวิจัยของ Dowsland (1987) ซึ่งนำหลักการของ Integer programming และ Dynamic programming มาแก้ปัญหาการบรรจุลงในคอนเทนเนอร์ ซึ่งใช้เวลานานในการค้นหาผลลัพธ์ที่เหมาะสม และได้เสนอเทคนิคการตัดแบบเครื่องประหารชีวิต (Guillotine cuts) เข้ามาใช้ในการแก้ปัญหา

Haessler et al. (1990) นำการวิเคราะห์ปัญหาการบรรจุ 3 มิติ มาประยุกต์ใช้กับการบรรจุเพื่อการขนส่งโดยรถบรรทุกและการขนส่งทางรถไฟ ของสินค้าที่มีค่าความหนาแน่นต่ำ โดยความหนาแน่นของกล่องที่บรรจุจะขึ้นอยู่กับปริมาตรที่จะใช้ในการขนส่ง ขั้นตอนวิธีในการแก้ปัญหาจะเริ่มจากการใช้กฎฮิวริสติก โดยปัญหาจะถูกพิจารณาเป็นปัญหาถุงเป้ (Knapsack problem) เมื่อไม่สามารถใช้กฎฮิวริสติกหาคำตอบได้ตามเงื่อนไขข้อบังคับ ขั้นตอนวิธีเชิงคณิตศาสตร์จะถูกนำมาใช้ในการหาคำตอบแทน

Scheithauer (1991) ได้ทำการศึกษาปัญหาการบรรจุกล่องรูปสี่เหลี่ยมลงในภาชนะบรรจุรูปสี่เหลี่ยมโดยใช้ Dynamic programming ในการแก้ปัญหาขนาดเล็ก หลังจากนั้น Scheithauer and Terno (1993) ได้เสนอโมเดลในการแก้ปัญหาการบรรจุวัตถุที่มีขนาดหนึ่งมิติ

และสองมิติโดยใช้ Integer programming ร่วมกับตัวแปร 0,1 ต่อมา Scheithauer (1995) ได้ทำการศึกษาการแก้ปัญหาการบรรจุโดยใช้ Branch-and-bound algorithm พบว่าสามารถลดเวลาในการคำนวณลงได้ด้วยการตรวจสอบรูปแบบการจัดวางว่ามีลักษณะคล้ายคลึงกันหรือไม่ ถ้าพบว่ารูปแบบการจัดวางคล้ายคลึงกันจะใช้ผลลัพธ์ที่จัดเก็บไว้ในหน่วยความจำแทนการคำนวณทำให้ลดเวลาคำนวณในขั้นตอนย่อยๆลงได้

Chen, Lee & Shen (1995) ได้พิจารณาปัญหาที่เกี่ยวข้องกับการบรรจุกล่องที่มีขนาดไม่เท่ากันลงในคอนเทนเนอร์ที่มีขนาดไม่เท่ากัน จำนวนของคอนเทนเนอร์และชนิดของคอนเทนเนอร์จะถูกเลือกเพื่อต้องการให้การบรรจุเหลือปริมาตรว่างน้อยที่สุด วิธีการที่นำมาใช้ในการแก้ปัญหาคือระเบียบวิธีวิเคราะห์ซึ่งจะมีรูปแบบเป็นการกำหนดการเชิงคณิตศาสตร์

Faina (2000) ได้ศึกษาปัญหาการบรรจุลงในคอนเทนเนอร์โดยนำรูปแบบทางคณิตศาสตร์มาใช้แก้ปัญหาทำให้การหาผลลัพธ์มีประสิทธิภาพสูงและได้คำตอบที่เหมาะสม โดยเริ่มต้นสร้างแต่ละส่วนด้วยรูปแบบทางเรขาคณิต จากนั้นทำการวิเคราะห์หาตำแหน่งการวางที่เป็นไปได้แล้วทำการเปรียบเทียบหาตำแหน่งการวางที่ดีกว่า และใช้ SA มารับประกันค่าคำตอบว่าได้คำตอบที่เหมาะสม

อย่างไรก็ตามวิธีการในกลุ่มของ COAs นี้จะเหมาะสมกับปัญหาขนาดเล็กเท่านั้น (Pongcharoen, 2001; Nagar et al., 1995) เนื่องมาจากกฎเกณฑ์ในการหาค่าคำตอบมีความตายตัวจนเกินไป (Enumerative search) หากปัญหามีขนาดใหญ่ขึ้นวิธีในกลุ่มนี้จะมีความยุ่งยากและใช้เวลาในการคำนวณมาก ซึ่งต่างจากวิธีการประมาณ (Approximation optimization algorithms: AOAs) โดยจะกล่าวถึงรายละเอียดในหัวข้อถัดไป

2.2 **วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักของการประมาณ** (Approximation optimization algorithms: AOAs)

วิธีการในกลุ่มนี้จะมีรูปแบบการค้นหาแบบสุ่ม (Stochastic search) จึงมีความเหมาะสม และทำได้ดีเมื่อนำไปประยุกต์ใช้กับปัญหาขนาดใหญ่ที่มีความซับซ้อนสูงอย่างปัญหาในตระกูลของการหาค่าที่ดีที่สุดเชิงการจัด (Combinatorial optimization problems) (Pongcharoen, 2001; Nagar et al., 1995) เมื่อพิจารณาปัญหาในการหาค่าที่ดีที่สุดโดยอาศัยการจัดเรียง (Sequencing optimization problems) คือ แต่ละคำตอบจะแตกต่างกันเมื่อลำดับของทรัพยากรแตกต่างกัน เช่น งาน (Job) ในปัญหา Scheduling หรือ กล่อง (Item) ในปัญหาการจัดเรียงลงในคอนเทนเนอร์พบว่า เมื่อปัญหามีขนาดใหญ่การจะหาและนำค่าคำตอบที่เป็นไปได้ทั้งหมดมาเปรียบเทียบกันแล้วเลือกค่าคำตอบที่ดีที่สุดนั้นเป็นเรื่องยากมาก ด้วยเหตุนี้จึงทำให้มี

วิธีการหาค่าคำตอบที่ดีสุดโดยการประมาณเกิดขึ้น (Pongcharoen, 2001) ได้แก่ วิธีการเมทต้าฮิวริสติก (Metaheuristics) ซึ่งเป็นสาขาหนึ่งของ AOAs ที่ประสบความสำเร็จอย่างมากในการจัดการกับปัญหาที่มีความซับซ้อนสูงๆ โดยวิธีการในกลุ่ม Metaheuristics นี้จะมีรูปแบบของการวนซ้ำ (Iterative) เป็นลักษณะเด่นที่เหมือนกัน แต่จะแตกต่างกันตรงกลไกที่ถูกนำมาใช้ในการค้นหาและสำรวจกลุ่มของค่าคำตอบที่เป็นไปได้ให้มีประสิทธิภาพ เพื่อให้ได้มาซึ่งค่าคำตอบที่ใกล้เคียงความเป็นจริงมากที่สุด (Near optimum solution) (Osman & Laporte, 1996) ดังนั้นวิธีการที่จะกล่าวถึงต่อไปนี้ คือวิธีการต่างๆที่อยู่ในกลุ่มของ Metaheuristics ซึ่งได้แก่ ซิมูเลทเทดแอนนีลลิง (Simulated Annealing: SA), ทาบูเสิร์จ (Taboo Search: TS), นิวรอลเน็ทเวิร์ค (Neural Network: NN) และ เจเนติคอัลกอริทึม (Genetic Algorithm: GA) อธิบายรายละเอียดของแต่ละวิธีการได้ดังหัวข้อถัดไปนี้

2.2.1 Simulated Annealing (SA)

SA อาศัยหลักการทางอุณหพลศาสตร์ของกระบวนการอบเหนียว (Annealing) ซึ่งเป็นวิธีการที่เลียนแบบการลดอุณหภูมิอย่างช้าๆระหว่างการหลอมโลหะเพื่อให้โลหะนั้นอ่อนตัว แล้วค่อยๆทำให้เย็นลงเพื่อให้ได้โลหะที่เหนียวและอยู่ในสภาวะที่เหมาะสมที่สุด โดย SA ถูกพัฒนาขึ้นมาจากแนวคิดของ Kirkpatrick, Gelatt and Vecchi (1983) และ Cerny (1985) ซึ่ง SA จะอาศัยรูปแบบการค้นหาแบบสุ่ม (Stochastic search) จากกลุ่มของค่าคำตอบข้างเคียง (Neighborhood space) และมีค่าอุณหภูมิ (Temperature: t) เป็นตัวแปรควบคุม (Control parameter) ซึ่งค่า t ที่กำกับผลเฉลยเปรียบเสมือนพลังงานของสถานะ และการทำงานแบบวนซ้ำเปรียบเสมือนกับการค่อยๆลดอุณหภูมิลงเรื่อยๆ สำหรับขั้นตอนในการทำงานนั้นจะเริ่มต้นจากการสุ่มสถานะเริ่มต้น (Initial state) ซึ่งจะประกอบไปด้วยตำแหน่งของค่าคำตอบเริ่มต้นในกลุ่มของค่าคำตอบที่เป็นไปได้ทั้งหมด (Solution space) และเวลาเริ่มต้น (Initial temperature: t_0) จากนั้นก็เข้าสู่ขั้นตอนการสุ่มหาตำแหน่งของค่าคำตอบตัวถัดไปโดยจะพิจารณาจากกลุ่มค่าคำตอบข้างเคียงของตำแหน่งเริ่มต้น ถ้าค่าคำตอบที่ได้เป็นค่าคำตอบที่ดีกว่าเดิม ซึ่งพิจารณาได้จากค่าฟังก์ชันเป้าหมาย (Objective function) ก็ให้ใช้ตำแหน่งของค่าคำตอบนั้นเป็นจุดเริ่มต้นในการหาตำแหน่งอื่นต่อไป แต่ถ้าค่าคำตอบที่ได้ไม่ได้ดีกว่าค่าเดิมก็อาจจะยอมรับค่าคำตอบใหม่ที่เลวกว่าค่าคำตอบเดิมได้ ภายใต้เงื่อนไข $e^{-\Delta f/kt} > \text{Random}(0,1)$ อธิบายหลักการเลือกได้ดังนี้

If $(\Delta f < 0)$ then

$s = s'$

else

if $(e^{-\Delta f/kt} > \text{Random}(0,1))$ then $s = s'$

end if

ในเงื่อนไขแรก จะยอมรับเมื่อได้ผลเฉลยใหม่ที่ดีกว่าซึ่งในตัวอย่างโปรแกรมนี้ถือว่าต้องการค่า f ที่น้อยที่สุด และเงื่อนไขที่สองคือ ถ้าผลเฉลยใหม่ไม่ดีกว่าจะยอมรับได้ก็ต่อเมื่อ $e^{-\Delta f/kt}$ มีค่ามากกว่าค่า Random ระหว่าง 0 ถึง 1 เมื่อ $\Delta f = f(s') - f(s)$ โดย Δf คือค่าความแตกต่างระหว่างค่าคำตอบใหม่และค่าคำตอบเดิม, k คือค่าคงตัวของโบลต์ซมันน์ (Boltzmann) และ $e^{-\Delta f/kt}$ ได้มาจากพฤติกรรมของกระบวนการฟิสิกส์ระหว่างการอบเหนียว หมายความว่าในระยะแรกของการวนซ้ำค่าอุณหภูมิเริ่มต้นจะมีค่าสูงทำให้โอกาสในการยอมรับ (Probability of accepting) ค่าคำตอบที่แย่กว่านั้นมีค่าสูงตามไปด้วย รูปแบบการค้นหาก็จะเป็นไปแบบหยาบๆ เพื่อให้ขอบเขตในการค้นหานั้นสามารถแผ่ขยายออกไปได้ในวงกว้างไม่ยึดติดอยู่กับกลุ่มของค่าคำตอบเพียงกลุ่มใดกลุ่มหนึ่ง แต่เมื่อค่าอุณหภูมิลดลงเรื่อยๆจะทำให้การยอมรับค่าคำตอบที่แย่กว่านั้นเป็นไปได้ยากขึ้น อาจกล่าวได้ว่าที่อุณหภูมิต่ำๆค่าคำตอบที่แย่กว่าจะไม่สามารถถูกยอมรับได้เลยซึ่งค่าคำตอบที่จะถูกยอมรับได้จะต้องดีกว่าค่าคำตอบเดิมเท่านั้น ดังนั้นรูปแบบการค้นหาในช่วงอุณหภูมิต่ำๆนี้ จะมีขอบเขตของการค้นหาที่แคบลง แต่จะมีความละเอียดในการค้นหามากขึ้นกว่าเดิม (Osman & Laporte, 1996; Pongcharoen, 2001) และการวนซ้ำจะสิ้นสุดเมื่ออุณหภูมิเย็นลงถึงช่วงที่กำหนด หรือถึงสภาพที่ไม่ได้ผลเฉลยที่ดีกว่าเกินเวลาที่ตั้งเป็นเกณฑ์ไว้

2.2.2 Taboo Search (TS)

TS เป็นการค้นหาคำตอบที่อาศัยหลักการวนซ้ำ (Iterative Search) หลายๆ ครั้งจึงจะได้มาซึ่งค่าคำตอบที่ดีที่สุดเช่นเดียวกับ SA และ GA ซึ่ง TS ถูกพัฒนาขึ้นโดย Glover (1986) และ Hansen (1986) TS จะอาศัยหลักการหาค่าคำตอบข้างเคียง (Neighbor search) โดยการทำงานจะเริ่มจากการสุ่มสถานะเริ่มต้นซึ่งคล้ายกับการทำงานของ SA ขึ้นมา จากนั้นก็จะค้นหาค่าคำตอบข้างเคียงที่ดีที่สุดของค่าคำตอบในขณะนั้น เมื่อได้ค่าคำตอบใหม่ที่ดีกว่าแล้วก็จะถือว่าเป็น 1 รอบการทำงาน (Iteration) และให้วนซ้ำเช่นนี้ไปเรื่อยๆจนจบการทำงาน โดยที่แต่ละรอบนั้นก็จะใช้ค่าคำตอบที่ได้จากรอบที่แล้วมาเป็นค่าคำตอบเริ่มต้นในการทำงานต่อไป แต่ลักษณะเฉพาะของ TS ที่ต่างจากวิธีการอื่นคือ จะพิจารณาเฉพาะค่าคำตอบถัดไปที่ไม่เคยถูกเลือก

มาเท่านั้นโดยจะทำการค้นหาในที่ที่ไม่เคยไปมาก่อนถึงแม้ว่าจะได้ค่าคำตอบที่แย่กว่าก็ตาม ดังนั้น จึงมีการเก็บประวัติของค่าคำตอบลงในลิสต์ที่เรียกว่า "ทาบูลิสต์ (Taboo list)" ด้วยเหตุนี้เอง Taboo list จึงถูกสร้างขึ้นมาเพื่อป้องกันมิให้เกิดการค้นหาที่ซ้ำซ้อน (Cycling) โดยในลิสต์จะเก็บค่าคำตอบที่เคยถูกนำไปเป็นค่าคำตอบเริ่มต้นในการหาค่าคำตอบข้างเคียง และได้ถูกทดแทนด้วยค่าคำตอบที่ดีกว่าไปแล้ว Taboo จึงกำหนดให้ค่าคำตอบเริ่มต้นดังกล่าวนี้เป็นค่าคำตอบต้องห้าม (Forbidden solution) และจะถูกนำไปเก็บเอาไว้ใน Taboo list นั่นหมายความว่าภายใน *m* รอบการทำงาน นับตั้งแต่รอบการทำงานที่ค่าคำตอบต้องห้ามนี้ถูกนำไปใช้เป็นค่าคำตอบเริ่มต้น Taboo จะไม่ยินยอมให้ค่าคำตอบต้องห้ามนี้ถูกนำไปใช้เป็นค่าคำตอบเริ่มต้นอีก

ตัวอย่างงานวิจัยที่นำ TS ไปประยุกต์ใช้ คือ Lodi, Martello and Vigo (2002) ได้ใช้ TS แก้ปัญหา Three-dimensional bin packing โดยกำหนดให้กล่องที่นำมาทดสอบทุกๆ กล่องไม่สามารถหมุนได้ และฮิวริสติกที่ใช้ในการวางกล่อง คือ การจัดเรียงกล่องโดยแบ่งออกเป็นชั้นๆ ในชั้นแรกจะเป็นการเรียงกล่องไปตามพื้นของคอนเทนเนอร์ และจะทำการตัดชั้นโดยใช้กล่องที่สูงที่สุดเป็นเกณฑ์แล้วทำการวางในชั้นต่อไป ในการวางกล่องแต่ละใบจะนำกล่องที่ต้องการวางไปค้นหาตำแหน่งการวางโดยเปรียบเทียบกับกล่องข้างเคียงในตำแหน่งนั้น ซึ่งจะคำนึงถึงความใกล้เคียงทางด้านส่วนสูงของกล่องก่อน แล้วจึงมาคำนึงถึงส่วนกว้างของกล่องเรียกวิธีการนี้ว่า "Height first-Area second (HA)" อย่างไรก็ตามเงื่อนไขที่กำหนดขึ้นมาให้กับกล่องนั้นอาจทำให้ประสิทธิภาพในการหาคำตอบลดลง และทำให้ไม่สามารถนำไปใช้ได้อย่างกว้างขวางเท่าใดนัก

2.2.3 Neural Network (NN)

NN จะมีโครงสร้างการทำงานที่เลียนแบบพฤติกรรมการทำงานของสมองมนุษย์ โดยจะเก็บความรู้ที่ได้จากกระบวนการเรียนรู้ (Learning process) เอาไว้เพื่อใช้เป็นประสบการณ์ ในการทำงานครั้งต่อไป (Haykin, 1999) Osman and Laporte (1996) ได้กล่าวไว้ว่า NN เป็นเครื่องมือที่ทรงประสิทธิภาพมากในแขนงของวิทยาศาสตร์และวิศวกรรมศาสตร์ แต่ NN ก็มักจะไม่ประสบความสำเร็จนักเมื่อถูกนำไปประยุกต์ใช้กับปัญหาในเชิงการหาค่าที่ดีที่สุด (Optimization problem) โดย NN มักจะถูกนำไปใช้ในการทำนาย (Predict), การจัดกลุ่ม (Classify) หรือใช้ในการจดจำรูปแบบต่างๆ (Recognize pattern) ได้เป็นอย่างดี อย่างเช่น ปัญหาการเดินทางของเซลส์แมน (Traveling salesman problem: TSP) ที่ Hopfield and Tank (1985) ได้นำ NN ไปแก้ปัญหาโดยได้เสนอ NN แบบใหม่ที่เรียกว่า "Hopfield Network" ซึ่งสามารถหาค่าคำตอบที่ดีที่สุดสำหรับ TSP ได้เป็นครั้งแรก หรือแม้แต่ปัญหาการจัดตาราง (Scheduling problem) ที่ Carrasco and Pato (2004) ได้นำ NN รูปแบบต่างๆมาประยุกต์ใช้กับการจัดตาราง

เรียนตารางสอน (Class/teacher timetabling problem: CTTP) โดยเริ่มต้นจากการแสดงลักษณะของปัญหา Hard constraints และ Soft constraints ตลอดจนสมการสำหรับ Energy function ที่จะเอาไปใช้ใน NN โดยจะแบ่ง NN ออกเป็น 2 แบบ แบบแรกจะใช้ Potts mean-field บนพื้นฐานของ Continuous Potts neurons แบบที่สองจะเป็นรูปแบบที่นำเสนอขึ้นมาคือ Discrete neural network ผลจากการทดลองพบว่า รูปแบบที่เสนอนั้นมีประสิทธิภาพที่ดีกว่าทั้งในด้านเวลาและค่าคำตอบ และคำตอบที่ได้จาก Discrete neural network สามารถนำไปใช้งานได้จริง

2.2.4 Genetic Algorithm (GA)

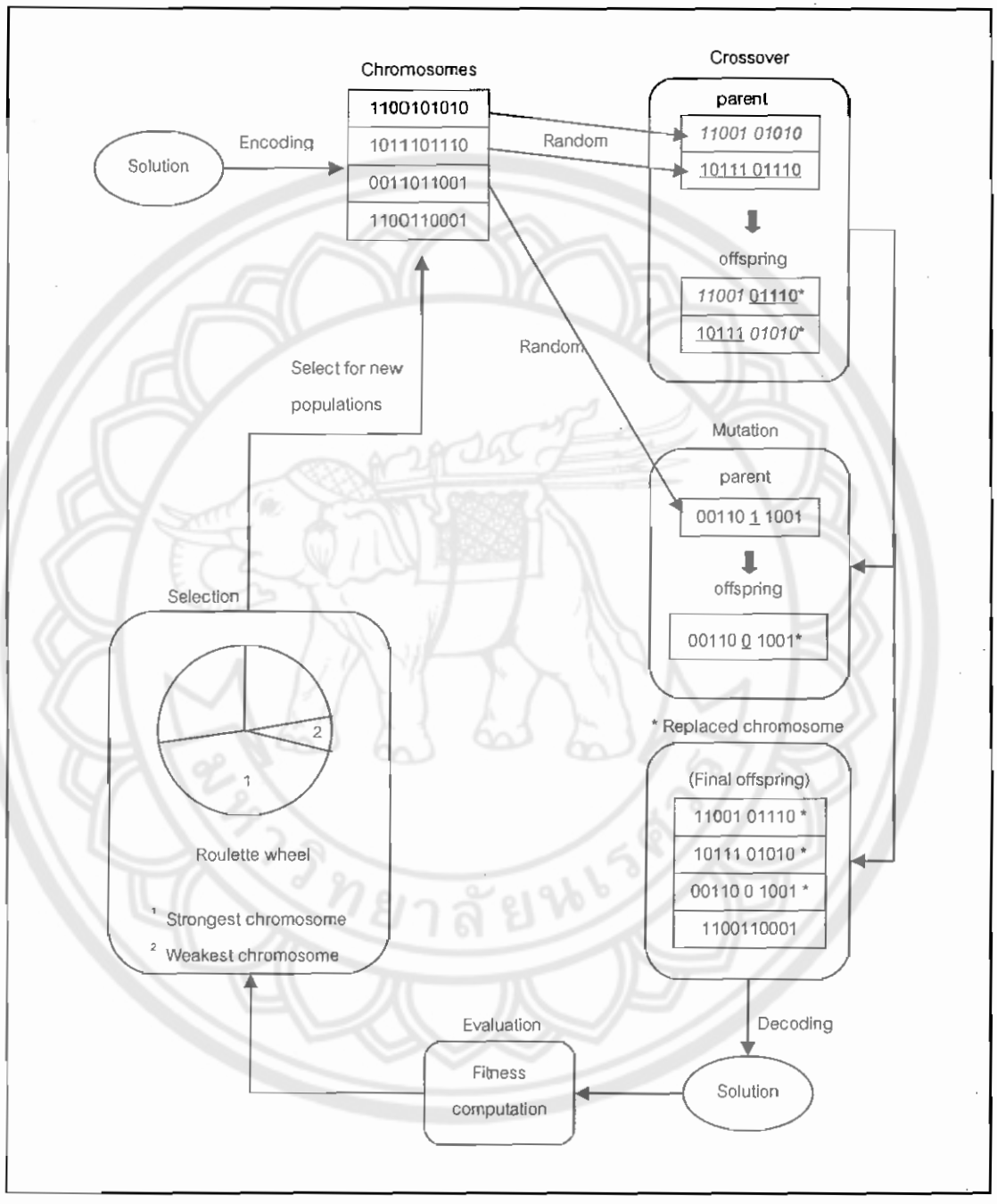
เจเนติคอัลกอริทึมถูกพัฒนามาจากทฤษฎีของ Charls Darwin ใน ค.ศ.1859 โดยเสนอแนวความคิดเกี่ยวกับการเกิดสปีชีส์ของสิ่งมีชีวิต (The origin of species) และได้เสนอหลักการของวิวัฒนาการที่ผ่านกระบวนการคัดเลือกตามธรรมชาติ ไว้ดังนี้

1. สิ่งมีชีวิตแต่ละชนิดมีแนวโน้มที่จะถ่ายทอดลักษณะของตนไปสู่ลูกหลาน
2. ธรรมชาติทำให้สิ่งมีชีวิตมีลักษณะต่างๆ กัน
3. สิ่งมีชีวิตที่มีลักษณะที่เหมาะสมที่สุด มีแนวโน้มที่จะมีลูกหลานมากกว่าสิ่งมีชีวิตที่มีลักษณะไม่เหมาะสม ซึ่งจะทำให้ประชากรของสิ่งมีชีวิตที่เหมาะสมนั้นได้อยู่รอดต่อไป
4. เมื่อระยะเวลาผ่านไปยาวนาน จะเกิดการกลายพันธุ์ (Mutation) ขึ้น และเกิดสปีชีส์ใหม่ที่มีลักษณะเหมาะสมกับระบบนิเวศน์นั้น

ต่อมาในปี ค.ศ.1975 John Holland ได้นำแนวคิดดังกล่าวนี้มาประยุกต์ใช้เป็นวิธีการแก้ปัญหาในการหาคำตอบที่เหมาะสม (Optimization problems) และใช้ชื่อว่า "เจเนติคอัลกอริทึม (Genetic Algorithm: GA)" แต่ GA ก็ยังไม่เป็นที่แพร่หลายในหมู่นักวิจัยเท่าใดนัก เพราะในขณะนั้น GA ยังเป็นแนวคิดที่ใหม่อยู่ หลังจากนั้น Goldberg (1989) ลูกศิษย์ของฮอลแลนด์ได้ใช้ GA แก้ปัญหาที่มีความยุ่งยากได้สำเร็จในปี ค.ศ.1989 และตีพิมพ์หนังสือที่อธิบายรายละเอียดต่างๆของ GA ตลอดจนวิธีการนำไปประยุกต์ใช้ ซึ่งเป็นส่วนที่ทำให้คนทั่วไปได้รู้จักและเข้าใจถึงวิธีการแก้ปัญหาที่อาศัยหลักพันธุกรรมนี้มากยิ่งขึ้นจนในที่สุดก็กลายเป็นที่นิยมของบรรดานักวิจัยอย่างแพร่หลายในเวลาต่อมา

หลักการทำงานของ GA จะมีวิธีในการหาค่าคำตอบโดยอาศัยรูปแบบกลไกการคัดสรรพันธุกรรมจากธรรมชาติซึ่งคำตอบที่ดี จะสามารถอยู่รอดและถูกถ่ายทอดไปยังรุ่นต่อๆไปได้ จุดเด่นของ GA จะต่างจากวิธีการอื่นคือ GA จะเก็บผลเฉลยเป็นเซตในขณะที่วิธีการอื่นจะเป็นการเก็บและเปลี่ยนแปลงทีละผลเฉลย (Goldberg, 1989) ในปัจจุบัน GA ได้รับการพัฒนามาอย่างต่อเนื่องจึงทำให้มีลักษณะที่แตกต่างกันออกไปตามงานวิจัยและแนวทางการพัฒนาของ

นักวิจัยนั้นๆซึ่งโดยทั่วไปจะมีโครงสร้างมาจาก GA ตัวต้นแบบที่เรียกว่า "เจเนติคอัลกอริทึมอย่างง่าย (Simple Genetic Algorithms: SGA)" โครงสร้างของ SGA สามารถแสดงได้ดังภาพ 4



ภาพ 4 แสดงโครงสร้างการทำงานของ SGA (Modified from Gen & Cheng, 1997)

ในร่างกายของสิ่งมีชีวิตจะประกอบไปด้วยเซลล์ (Cell) และในแต่ละเซลล์จะประกอบไปด้วยกลุ่มของโครโมโซม (Chromosome) และในแต่ละโครโมโซมจะประกอบไปด้วยหน่วยเล็กๆที่เรียงตัวกันกลายเป็นโครโมโซมเรียกแต่ละหน่วยนี้ว่า "ยีน (Gene)" ซึ่งในแต่ละยีนจะเหมือนกับการเข้ารหัส (Encode) ทางพันธุกรรมด้วยสารโปรตีน โดยยีนจะเป็นหน่วยควบคุมลักษณะต่างๆของสิ่งมีชีวิต เช่น สีของผิว หรือสีของตา เป็นต้น ดังนั้นเจเนติคอัลกอริทึมจึงเป็นการประยุกต์ใช้วิธีการทางพันธุกรรมศาสตร์ ซึ่งโครโมโซมแต่ละตัวก็คือคำตอบที่เป็นไปได้ของปัญหา โดยปกติแล้วโครโมโซมจะใช้สัญลักษณ์สตริง (String) และแต่ละโครโมโซมจะประกอบไปด้วยยีนหลายๆยีนซึ่งจะมีจำนวนขึ้นอยู่กับความยาวของโครโมโซมและจะแทนยีนด้วยการใช้บิต (Bits) ดังนั้นจากภาพ 4 สามารถอธิบายขั้นตอนการทำงานของ GA ดังนี้ (Gen & Cheng, 1997)

ขั้นตอนที่ 1 การสร้างโครโมโซม (Encoding chromosome)

ขั้นตอนที่ 2 กระบวนการทางพันธุกรรม (Genetic operation)

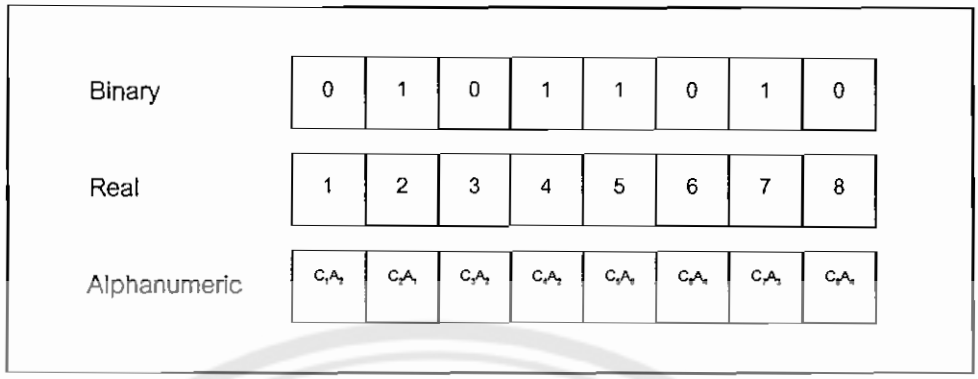
ขั้นตอนที่ 3 การคำนวณค่าความเหมาะสม (Fitness computation)

ขั้นตอนที่ 4 การคัดสรร (Selection)

ขั้นตอนที่ 5 ตรวจสอบเงื่อนไขหยุดการทำงาน

ขั้นตอนที่ 1 การสร้างโครโมโซม (Encoding chromosome)

เริ่มต้นโดยการสุ่มค่าคำตอบมาจำนวนหนึ่งแล้วแปลง (Encoding) ค่าคำตอบเหล่านี้ให้เป็น "โครโมโซม (Chromosome)" และเรียกกลุ่มของโครโมโซมเหล่านี้ว่า "ประชากร (Population)" ส่วนวิธีการ Encode นั้นจะขึ้นอยู่กับรูปแบบของปัญหาแต่ละปัญหา โดยโครโมโซมแต่ละตัวจะประกอบไปด้วยยีนจำนวนหนึ่งมาเรียงต่อกัน และสามารถแทนยีนได้ 2 รูปแบบดังนี้ (Pongcharoen, 2001) แบบที่หนึ่ง คือ แบบตัวเลขมี 2 ลักษณะคือ Binary และ Real รูปแบบของโครโมโซมที่เป็น Binary bit string จะแทนแต่ละยีนด้วย 0 กับ 1 ส่วนโครโมโซมที่ใช้แบบ Real string จะแทนแต่ละยีนด้วยเลขจำนวนจริง และแบบที่สอง คือ แบบตัวอักษรผสมตัวเลข (Alphanumeric) แสดงโครโมโซมรูปแบบต่างๆได้ดังภาพ 5



ภาพ 5 แสดงรูปแบบของโครโมโซมที่มี 8 ยีน

การทำงานในขั้นตอนนี้จะสุ่มการเรียงลำดับยีนของแต่ละโครโมโซม ทำให้โครโมโซมทั้งหมดมีการเรียงลำดับของยีนที่ต่างกัน และการกำหนดจำนวนโครโมโซมใน 1 รุ่นจะพิจารณาจากค่าที่กำหนดไว้ในตัวแปร (Parameter) "ขนาดของประชากร (Population size)" เช่น ถ้ากำหนดขนาดของประชากรเอาไว้ที่ 100 ก็หมายความว่าจะต้องมีจำนวนโครโมโซมในแต่ละรุ่นเท่ากับ 100 โครโมโซม ตัวแปรที่สำคัญอีกตัวหนึ่งของ GA คือ จำนวนรุ่นของประชากร (Number of generation) ซึ่งจะเป็นตัวกำหนดว่าโครโมโซมจะถูกพัฒนาไปทั้งหมดกี่รุ่น จากการศึกษาค้นคว้าพบว่าค่าของตัวแปรทั้ง 2 ตัวนี้มีนัยสำคัญ (Significant) ต่อประสิทธิภาพการทำงานของ GA (Pongcharoen & Promtet, 2004)

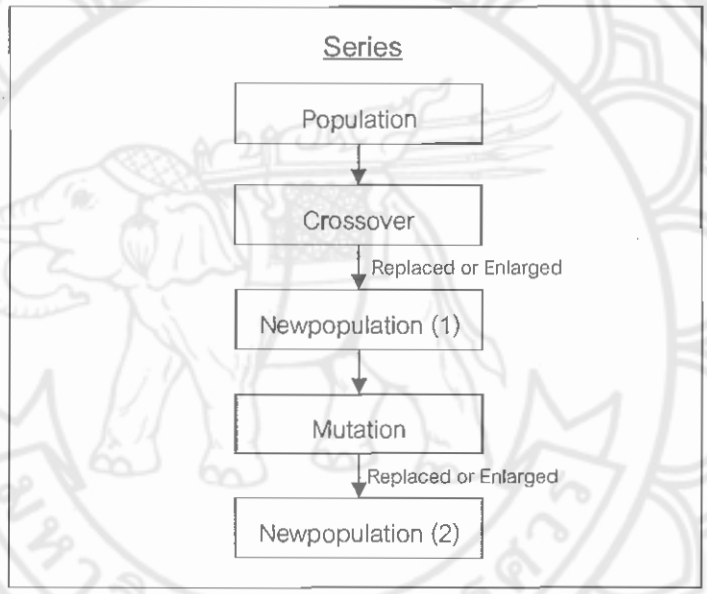
ขั้นตอนที่ 2 กระบวนการทางพันธุกรรม (Genetic operation)

หลังจากเสร็จขั้นตอนที่1 แล้วนำโครโมโซมทั้งหมดเข้าสู่กระบวนการทางพันธุกรรม (Genetic operation) ซึ่งมี 2 ขั้นตอนคือ การสลับสายพันธุ์ (Crossover) และการกลายพันธุ์ (Mutation) ซึ่ง Pongcharoen and Promtet (2004) ได้เสนอลำดับของการทำกระบวนการ GA ไว้ 2 รูปแบบ คือ แบบอนุกรม (Series) และแบบขนาน (Parallel) โดยขั้นตอนการสลับสายพันธุ์ และการกลายพันธุ์นี้จะทำให้เกิดโครโมโซมใหม่ที่เรียกว่า "โครโมโซมลูก (Offspring)" ขึ้น ซึ่งในการนำโครโมโซมลูกไปเก็บในกลุ่มของโครโมโซม หรือประชากร (Population) จะมี 2 แบบ คือ เก็บตามขนาดที่กำหนด (Regular) หรือแบบแทนที่ (Replace) และเก็บแบบขยาย (Enlarged) (Gen & Cheng, 1997) จากเนื้อหาข้างต้นสามารถแบ่งเป็น 3 ข้อย่อยคือ 1) ลำดับของการทำกระบวนการ GA (Sequence of genetic operations), 2) กระบวนการ GA (Genetic operation) และ 3) วิธีดำเนินการจัดเก็บโครโมโซมลูก อธิบายรายละเอียดของแต่ละข้อย่อยดังนี้

1 ลำดับของการทำกระบวนการ GA (Sequence of genetic operations)

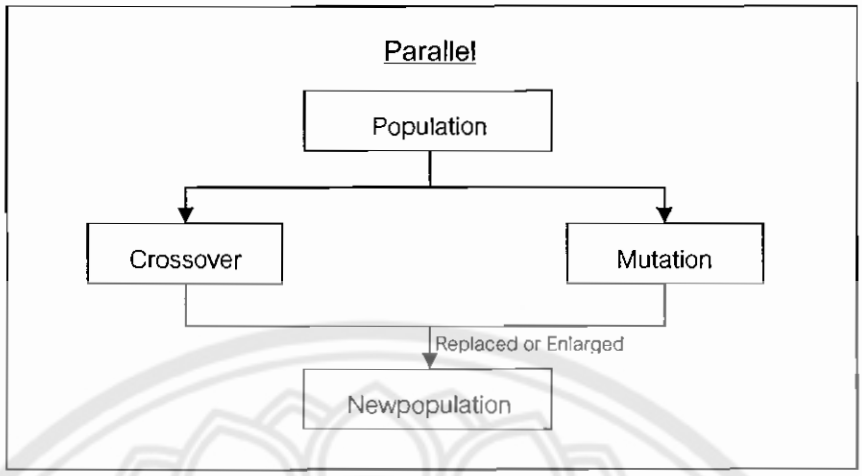
ลำดับของการทำกระบวนการ GA คือ ลำดับในการทำ การสลับสายพันธุ์ (Crossover) และการกลายพันธุ์ (Mutation) ซึ่งมี 2 รูปแบบ คือ แบบอนุกรม (Series) และแบบขนาน (Parallel) (Pongcharoen & Promtet, 2004) อธิบายรายละเอียดดังนี้

ลำดับของการทำกระบวนการ GA แบบอนุกรม คือ อันดับแรกจะทำการสลับสายพันธุ์ก่อน หลังจากนั้นจึงจะไปทำการกลายพันธุ์ แสดงลำดับของการทำกระบวนการ GA แบบอนุกรม ดังภาพ 6



ภาพ 6 แสดงลำดับของการทำกระบวนการ GA แบบอนุกรม

ลำดับของการทำกระบวนการ GA แบบขนาน คือ การทำการสลับสายพันธุ์ไปพร้อมๆ กับการทำการกลายพันธุ์ แสดงลำดับของการทำกระบวนการ GA แบบขนาน ดังภาพ 7



ภาพ 7 แสดงลำดับของการทำกระบวนการ GA แบบขนาน

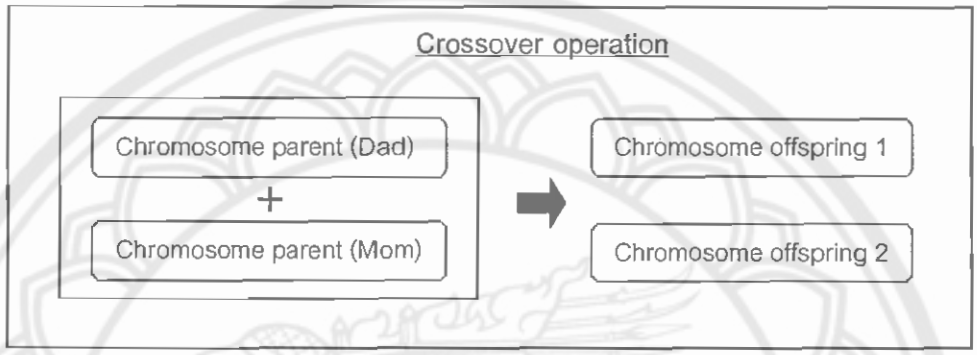
เนื้อหาจากลำดับของการทำกระบวนการ GA ที่กล่าวไว้ข้างต้น ได้มีการศึกษาโดย Pongcharoen and Promtet (2004) ถึงผลกระทบจากการใช้ลำดับของการทำกระบวนการ GA ที่ต่างกัน 2 แบบ คือ แบบอนุกรม และแบบขนาน ต่อการทำงานของ GA ในการหาผลเฉลย โดยทดสอบกับฟังก์ชั่นทางคณิตศาสตร์ และ ในวิทยานิพนธ์ของ วีณา พรหมเทศ (2548) ได้ทำการทดสอบอีกครั้ง โดยทดสอบกับปัญหาการจัดตารางสอน พบว่าลำดับของการทำกระบวนการ GA ไม่มีผลกระทบต่อการหาผลเฉลยของ GA (จากการวิเคราะห์ผลการทดลองไม่มีนัยสำคัญทางสถิติ) แต่อย่างไรก็ตามในทางปฏิบัติแล้ว Pongcharoen and Promtet (2004) และ วีณา พรหมเทศ (2548) ได้เสนอไว้ว่าควรเลือกใช้ลำดับของการทำกระบวนการ GA แบบอนุกรม เนื่องจากพบว่าให้ผลเฉลยที่ดีกว่า ซึ่งจะนำไปใช้ในวิทยานิพนธ์นี้ต่อไป

2 **กระบวนการ** GA (Genetic operations)

กระบวนการ GA ประกอบไปด้วยขั้นตอน 2 ส่วน คือ การสลับสายพันธุ์ (Crossover) และการกลายพันธุ์ (Mutation) อธิบายรายละเอียดดังนี้

การสลับสายพันธุ์ (Crossover) การสลับสายพันธุ์เป็นกลไกหลักของกระบวนการทางพันธุกรรม (Gen & Cheng, 1997) โดยจะทำการสุ่มเลือกโครโมโซมพ่อและโครโมโซมแม่ขึ้นมาสองโครโมโซม แล้วทำการสลับยีนหรือกลุ่มยีน "ระหว่าง" โครโมโซมพ่อและโครโมโซมแม่ ทำให้ได้ผลิตผลเป็นโครโมโซมลูก (Offspring) สองโครโมโซม โดยตัวแปรที่เป็นตัวกำหนดจำนวนโครโมโซมที่ต้องผ่านการสลับสายพันธุ์คือ เปอร์เซ็นต์ของโครโมโซมลูกที่เกิดจากการสลับสายพันธุ์ (Probabilities of crossover: %C) ซึ่งจะมีความสัมพันธ์โดยตรงกับขนาดของ

ประชากร เพราะจำนวนโครโมโซมที่จะถูกเลือกนั้นสามารถคำนวณได้จาก %C คูณด้วย "ขนาดของประชากร (Population size)" เช่น ถ้า %C = 0.4 และ Population size = 10 หมายความว่าจะมี โครโมโซม 4 โครโมโซมที่ถูกเลือกเป็นพ่อและแม่ มาสลับสายพันธุ์ หรือจะเกิดการสลับสายพันธุ์ 2 ครั้งต่อ 1 รุ่น เพราะการสลับสายพันธุ์ 1 ครั้ง จะได้โครโมโซมลูก 2 โครโมโซม แสดงการสลับสายพันธุ์ดังภาพ 8



ภาพ 8 แสดงการสลับสายพันธุ์

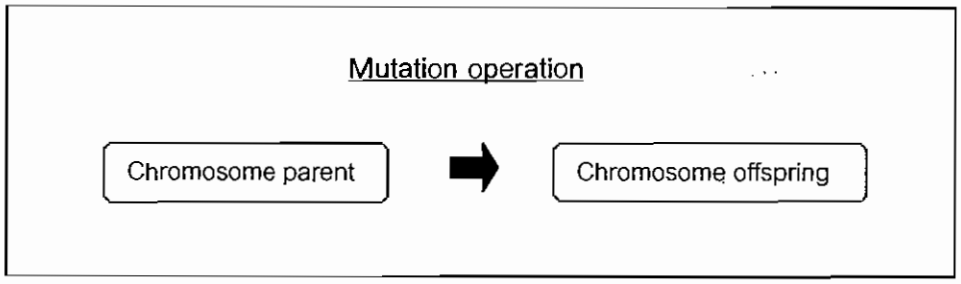
การกำหนดอัตราการสลับสายพันธุ์ ถ้าสูงเกินไปจะทำให้ ขอบเขตของคำตอบ (Solution space) ที่เป็นไปได้กว้างขึ้นและถูกค้นหาอย่างครอบคลุม และลดโอกาสที่จะได้คำตอบเป็นคำตอบเฉพาะพื้นที่ (Local solution) ซึ่งเป็นคำตอบที่ไม่ใช่คำตอบที่ดีที่สุด ในทางตรงกันข้าม ก็จะเป็นการเสียเวลาให้กับการค้นหาในพื้นที่ที่ไม่มีโอกาสจะเป็นคำตอบที่ดีที่สุด (Gen & Cheng, 1997) การกำหนดค่าที่เหมาะสมสำหรับตัวแปร %C นั้น Todd (1997) ที่แก้ปัญหา Traveling salesman ได้แนะนำเอาไว้ว่า %C นั้นควรจะกำหนดให้อยู่ระหว่าง 0.6 ถึง 0.9 ส่วนงานของ Pongcharoen et al. (2002) ที่แก้ปัญหา Scheduling ได้กำหนดเอาไว้ที่ช่วง 0.3-0.9 และงานของ Murata, Ishibuchi and Tanaka (1996) ที่แก้ปัญหา Flowshop scheduling ได้กำหนดค่าเอาไว้ที่ 1.0 จะเห็นได้ว่า จากลักษณะของปัญหาที่แตกต่างกันทำให้ค่าตัวแปร %C ของแต่ละงานวิจัยนั้นมีค่าที่ต่างกันด้วย ดังนั้นในวิทยานิพนธ์นี้จะทำการทดสอบเพื่อหาช่วงของค่า %C ที่เหมาะสมสำหรับ CPP ต่อไป

รูปแบบของการสลับสายพันธุ์มีหลายรูปแบบ และได้ถูกรวบรวมเอาไว้ในตาราง 3 โดยข้อมูลเหล่านี้ถูกนำมาจากงานของ Todd (1997) และงานของ Pongcharoen et al. (2001) ซึ่งได้ทดสอบประสิทธิภาพของวิธีการเหล่านี้เอาไว้ ส่วนขั้นตอนการทำงานของแต่ละวิธีการนั้น จะอธิบายไว้ในส่วนของภาคผนวก

ตาราง 3 แสดงการสลับสายพันธุ์ในแบบต่างๆ

การสลับสายพันธุ์ (*COP*)	เอกสารอ้างอิง
Order crossover (OX)	Davis (1985)
Partially mapped crossover (PMX)	Goldberg and Lingle (1985)
Cycling crossover (CX)	Oliver et al. (1987)
Edge recombination crossover (ERX)	Whitley et al. (1989)
Enhanced edge recombination crossover (EERX)	Starkweather et al. (1991)
Position based crossover (PBX)	Syswerda (1991)
Maximal preservation crossover (MPX)	Mühlenbein et al. (1992)
One point crossover (1PX)	Murata and Ishibuchi (1994)
Two point center crossover (2PCX)	Murata and Ishibuchi (1994)

การกลายพันธุ์ (Mutation) เมื่อประชากรถูกพัฒนาผ่านไปหลายๆรุ่น จะทำให้โครโมโซมมีลักษณะคล้ายคลึงกันมากจนแทบจะเหมือนกันทั้งหมด ซึ่งการสลับสายพันธุ์แทบจะไม่สามารถก่อให้เกิดความเปลี่ยนแปลงภายในโครโมโซมได้ จึงจำเป็นจะต้องมีกระบวนการกลายพันธุ์เพื่อเป็นการทดแทนยีนที่สูญหายไปจากประชากรในระหว่างกระบวนการถ่ายทอดพันธุกรรม และเป็นการค้นพบยีนใหม่ซึ่งไม่เคยปรากฏในประชากรเริ่มต้นมาก่อน โดยจะทำการสุ่มเลือกโครโมโซมต้นแบบมาคราวละ 1 โครโมโซมแล้วทำการสลับหรือเปลี่ยนแปลงยีน "ภายใน" ตัวโครโมโซมนั้น ซึ่งผลที่ได้จะเป็นโครโมโซมลูก 1 โครโมโซม โดยตัวแปรที่เป็นตัวกำหนดจำนวนโครโมโซมที่ต้องผ่านการกลายพันธุ์คือ เปอร์เซ็นต์ของโครโมโซมลูกที่เกิดจากการกลายพันธุ์ (Probabilities of mutation: %M) ซึ่งจะมีความสัมพันธ์โดยตรงกับขนาดของประชากรเพราะจำนวนโครโมโซมที่จะถูกเลือกนั้นสามารถคำนวณได้จาก %M คูณด้วย "ขนาดของประชากร (Population size)" เช่น ถ้า %M = 0.2 และ Population size = 10 หมายความว่าจะมีโครโมโซม 2 โครโมโซมที่ต้องถูกกลายพันธุ์ หรือจะเกิดการกลายพันธุ์ 2 ครั้งต่อ 1 รุ่น เพราะการกลายพันธุ์ 1 ครั้งจะได้โครโมโซมลูก 1 โครโมโซม แสดงการกลายพันธุ์ดังภาพ 9



ภาพ 9 แสดงการกลายพันธุ์

การกำหนดอัตราการกลายพันธุ์ถ้าต่ำเกินไปก็จะไม่ได้ใช้ประโยชน์จากการกลายพันธ์ ซึ่งอาจเป็นการเสียโอกาสในการที่จะได้คำตอบที่ดี แต่ถ้ากำหนดสูงเกินไปกระบวนการวิวัฒนาการก็จะถูกสอดแทรกด้วยการสุ่ม ซึ่งทำให้ข้อมูลที่ถ่ายทอดมาจากพ่อและแม่ไม่ได้นำมาใช้ประโยชน์ หรือไม่ได้มีการเรียนรู้จากการค้นหาในรุ่นที่ผ่านมา (Gen & Cheng, 1997) โดยในการหาคำตอบของ GA ในบางครั้งคำตอบอาจติดอยู่ใน Local optima ซึ่งเป็นคำตอบที่ยังไม่ดีที่สุด การกลายพันธุ์ด้วยอัตราส่วนที่เหมาะสมจะทำให้คำตอบสามารถหลุดออกจาก Local optima ได้ การกำหนดค่าที่เหมาะสมสำหรับตัวแปร %M นั้น Todd (1997) ได้แนะนำเอาไว้ว่า %M ควรจะอยู่ในช่วง 0.001 ถึง 0.01 ส่วนงานของ Murata et al. (1996) ได้กำหนดค่าเอาไว้ที่ 1.0 เช่นเดียวกับตัวแปร %C และงานของ Pongcharoen et al. (2002) ได้กำหนดเอาไว้ในช่วง 0.02-0.18 จากงานวิจัยที่กล่าวมาจะเห็นได้ว่า ลักษณะของปัญหาที่ต่างกันและการกำหนดค่าพารามิเตอร์อื่นๆ (เช่น %C) ที่ต่างกันก็จะทำให้ค่าตัวแปร %M ของแต่ละงานวิจัยนั้นมีค่าความเหมาะสมที่ต่างกันด้วยเช่นเดียวกับค่าตัวแปร %C ดังนั้นในวิทยานิพนธ์นี้จะทำการทดสอบเพื่อหาช่วงของค่า %M ที่เหมาะสมสำหรับ CPP ต่อไป

รูปแบบของการกลายพันธุ์มีหลายรูปแบบ และได้ถูกรวบรวมเอาไว้ในตาราง 4 โดยข้อมูลเหล่านี้ถูกนำมาจากงานของ Todd (1997) และงานของ Pongcharoen et al. (2001) ซึ่งได้ทดสอบประสิทธิภาพของวิธีการเหล่านี้เอาไว้ ส่วนขั้นตอนการทำงานของแต่ละวิธีการนั้นจะอธิบายไว้ในส่วนของภาคผนวก

ตาราง 4 แสดงการกลายพันธุ์ในแบบต่างๆ

การกลายพันธุ์ (*MOP*)	เอกสารอ้างอิง
Inverse mutation (IM)	Goldberg (1989)
Shift operation mutation (SOM)	Murata and Ishibuchi (1994)
Three operations adjacent swap mutation (3OAS)	Murata and Ishibuchi (1994)
Three operations random swap mutation (3ORS)	Murata and Ishibuchi (1994)
Two operations adjacent swap mutation (2OAS)	Murata and Ishibuchi (1994)
Two operations random swap mutation (2ORS)	Murata and Ishibuchi (1994)
Center inverse mutation (CIM)	Tralle (2000)
Enhanced two operations random swap mutation (E2ORS)	Tralle (2000)

วิธีการที่เลือกใช้ในการสลับสายพันธ์และการกลายพันธ์ เป็นปัจจัยหนึ่งที่ช่วยเพิ่มประสิทธิภาพการทำงานของ GA ซึ่งชนิดของการสลับสายพันธุ์ และการกลายพันธุ์ มีอยู่หลายชนิด แต่ในการนำมาใช้จะต้องพิจารณาว่าเหมาะสมกับปัญหานั้นหรือไม่ เพราะการสลับสายพันธุ์ หรือการกลายพันธ์บางชนิดก็ไม่สามารถนำไปประยุกต์ใช้กับปัญหาบางประเภทได้

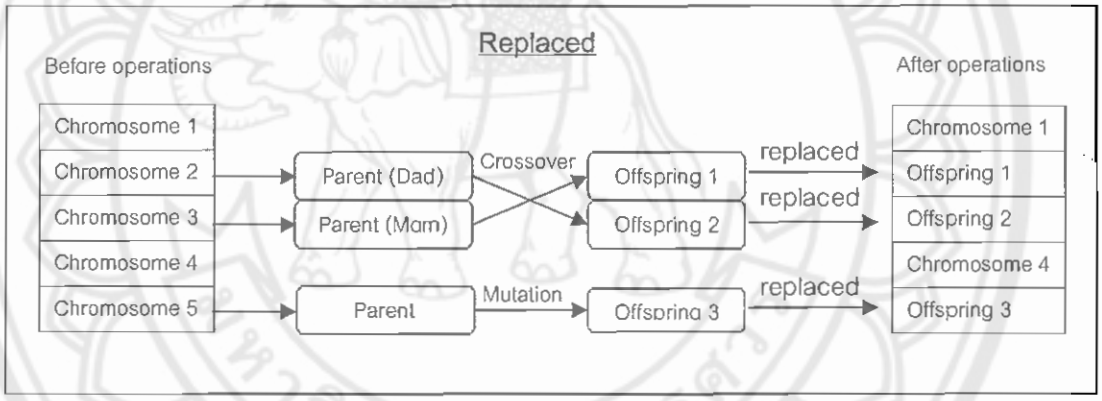
Todd (1997) ได้ทดสอบประสิทธิภาพของการสลับสายพันธุ์และการกลายพันธุ์ชนิดต่างๆกับ "ปัญหาการเดินทางของเซลส์แมน (TSP)" ที่มีจำนวนจังหวัดเท่ากับ 10, 20, 50, 100 และ 200 จังหวัด โดยที่ตัวแปรอื่นๆนั้นคงที่พบว่า การสลับสายพันธุ์และการกลายพันธุ์ที่ให้ค่าคำตอบที่ดีที่สุดคือ Enhanced edge recombination crossover (EERX) และ Two operation adjacent swap (2OAS)

Pongcharoen et al. (2001) ได้ศึกษาและทดสอบการสลับสายพันธุ์ และการกลายพันธุ์กับปัญหาการจัดตาราง (Scheduling problem) โดยได้รวมตัวแปรอื่นๆที่มีผลต่อการทำงานของ GA เข้ามาพิจารณาด้วย อันได้แก่ ขนาดของประชากร (Population size), จำนวนรุ่นของโครโมโซม (Generation), เปอร์เซ็นต์ของโครโมโซมลูกที่เกิดจากการสลับสายพันธุ์ (%C), เปอร์เซ็นต์ของโครโมโซมลูกที่เกิดจากการกลายพันธุ์ (%M) และวิเคราะห์ผลการทดลองโดยใช้หลักการทางสถิติซึ่งสรุปผลว่า การสลับสายพันธุ์ และการกลายพันธุ์ที่ดีที่สุดเป็นการใช้วิธีการของ EERX และ 2OAS เช่นเดียวกัน

3 วิธีดำเนินการจัดเก็บโครโมโซมลูก

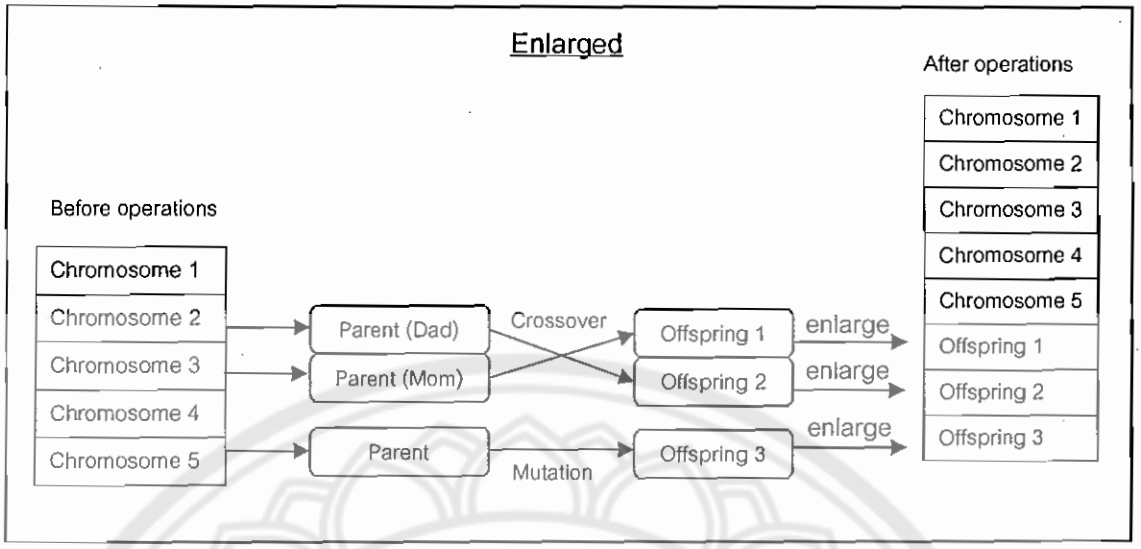
หลังจากเสร็จขั้นตอนการสลับสายพันธุ์ หรือการกลายพันธุ์แล้วจะทำให้เกิดโครโมโซมใหม่ที่เรียกว่า "โครโมโซมลูก (Offspring)" ขึ้น ซึ่งในการนำโครโมโซมลูกไปเก็บในกลุ่มของโครโมโซม หรือประชากร (Population) นั้นจะมี 2 แบบ คือ เก็บตามขนาดที่กำหนด (Regular) หรือแบบแทนที่ (Replace) และเก็บแบบขยาย (Enlarged) (Gen & Cheng, 1997) อธิบายรายละเอียดดังนี้

วิธีดำเนินการจัดเก็บโครโมโซมลูกแบบแทนที่ โครโมโซมลูกที่ได้มาจากการสลับสายพันธุ์ หรือการกลายพันธุ์ จะถูกนำไปแทนที่ (Replace) ตำแหน่งของโครโมโซมพ่อแม่ที่ถูกสุ่มเลือกขึ้นมาสลับสายพันธุ์ หรือกลายพันธุ์นั้น ซึ่งเรียกกระบวนการนี้ว่า การให้กำเนิดแบบแทนที่ (Generational replacement) ดังนั้นขนาดของประชากรรุ่นใหม่จะมีขนาดเท่ากับประชากรเดิม (Gen & Cheng, 1997) แสดงรูปแบบการจัดเก็บโครโมโซมลูกแบบแทนที่ดังภาพ 10



ภาพ 10 แสดงวิธีดำเนินการจัดเก็บโครโมโซมลูกแบบแทนที่

วิธีดำเนินการจัดเก็บโครโมโซมลูกแบบขยาย โครโมโซมลูกที่ได้มาจากการสลับสายพันธุ์ หรือการกลายพันธุ์ จะไม่ถูกนำไปแทนที่ตำแหน่งของโครโมโซมพ่อแม่ ดังนั้นขนาดของประชากรรุ่นใหม่จะมีขนาดเท่ากับ ขนาดของประชากรเดิมบวกด้วยขนาดของลูกที่เกิดขึ้นในแต่ละรุ่น ด้วยวิธีการจัดเก็บแบบขยายตัวนี้จะทำให้โครโมโซมพ่อแม่ และโครโมโซมลูกมีโอกาสเท่ากันในการที่จะผ่านไปถึงกระบวนการคัดสรรเพื่อเป็นประชากรในรุ่นถัดไป (Gen & Cheng, 1997) แสดงรูปแบบการจัดเก็บโครโมโซมลูกแบบขยายดังภาพ 11



ภาพ 11 แสดงวิธีดำเนินการจัดเก็บโครโมโซมลูกแบบขยาย

เนื้อหาวิธีดำเนินการจัดเก็บโครโมโซมลูกที่กล่าวไว้ข้างต้นได้มีการศึกษาโดย Pongcharoen and Promtet (2004) ถึงผลกระทบจากการใช้วิธีดำเนินการจัดเก็บโครโมโซมลูกที่ต่างกัน 2 แบบ คือ แบบแทนที่ และแบบขยาย ต่อการทำงานของ GA ในการหาผลเฉลย โดยทดสอบกับฟังก์ชั่นทางคณิตศาสตร์ และในวิทยานิพนธ์ของ วีณา พรหมเทศ (2548) ได้ทำการทดสอบอีกครั้ง โดยทดสอบกับปัญหาการจัดตารางสอน พบว่าวิธีดำเนินการจัดเก็บโครโมโซมลูกไม่มีผลกระทบต่อการหาผลเฉลยของ GA (จากการวิเคราะห์ผลการทดลองไม่มีนัยสำคัญทางสถิติ) แต่อย่างไรก็ตามในทางปฏิบัติแล้ว Pongcharoen and Promtet (2004) และ วีณา พรหมเทศ (2548) ได้เสนอไว้ว่าควรเลือกใช้วิธีดำเนินการจัดเก็บโครโมโซมลูกแบบแทนที่โครโมโซมพ่อแม่ (Replace) เนื่องจากพบว่าให้ผลเฉลยที่ดีกว่า ซึ่งจะนำไปใช้ในวิทยานิพนธ์นี้ต่อไป

ขั้นตอนที่ 3 การคำนวณค่าความเหมาะสม (Fitness computation)

เมื่อผ่านกระบวนการทางพันธุกรรมแล้ว โครโมโซมทั้งหมดจะถูกประเมิน (Evaluate) เพื่อคำนวณหาค่าความเหมาะสม (Fitness value) ที่จะกลายเป็นโอกาสในการอยู่รอดของแต่ละโครโมโซม (Probability of selection) โดยกระบวนการในการคำนวณค่าความเหมาะสมของโครโมโซมนั้นมี 3 ขั้นตอนคือ (Gen & Cheng, 1997)

ขั้นที่ 1 ขั้นของการถอดรหัสโครงสร้างโครโมโซมให้กลายเป็นโครงสร้างคำตอบ (Decoding chromosome) การถอดรหัสของโครโมโซมในแต่ละปัญหาจะมีวิธีการถอดรหัสที่แตกต่างกันออกไปขึ้นอยู่กับฟังก์ชันเป้าหมาย (Objective function) ที่กำหนดขึ้น

ขั้นที่ 2 นำแต่ละโครโมโซมไปคำนวณในฟังก์ชันเป้าหมาย [$f(x)$] เพื่อประเมินหาค่าคำตอบ

ขั้นที่ 3 เปลี่ยนผลลัพธ์ของสมการเป้าหมาย เป็นค่าความเหมาะสม (Fitness value) ซึ่งขึ้นอยู่กับเป้าหมายของปัญหาว่าต้องการหาค่าคำตอบที่มากที่สุด (Maximize problem) หรือต้องการหาค่าคำตอบที่น้อยที่สุด (Minimize problem) ซึ่งมีวิธีการดังนี้

สำหรับปัญหาที่ต้องการหาค่ามากสุด (Maximize problem) ค่าความเหมาะสมจะเท่ากับ ค่าผลลัพธ์ของสมการเป้าหมาย สามารถเขียนเป็นรูปสมการได้ดังนี้

$$eval(v_k) = f(x_k) \quad , k = 1, 2, \ldots, pop\_size \tag{1}$$

เมื่อ $eval(v_k)$ คือ ค่าความเหมาะสม (Fitness value) ของแต่ละโครโมโซม

$f(x_k)$ คือ ฟังก์ชันเป้าหมาย

pop_size คือ ขนาดของประกร หรือจำนวนโครโมโซมทั้งหมด

ส่วนปัญหาที่ต้องการหาค่าน้อยสุด (Minimize problem) ในกรณีนี้จำเป็นจะต้องมีการปรับค่าจากค่าคำตอบน้อยซึ่งเป็นคำตอบที่ดี ให้มีค่าความเหมาะสมมาก และค่าคำตอบมากซึ่งเป็นคำตอบที่ไม่ดี ให้มีค่าความเหมาะสมน้อย ซึ่งในวิทยานิพนธ์นี้จะใช้วิธีการกลับค่าโดยนำแนวคิดมาจาก Yang and Kuo (2003) ที่นำค่าคำตอบที่แย่ที่สุด (Worst solution: x_w) มาพิจารณา โดยนำ x_w มาเป็นตัวตั้งแล้วลบด้วยค่าคำตอบของโครโมโซมแต่ละโครโมโซม ซึ่ง วัฒนพล ชัยเนตร (2548) ได้นำมาปรับปรุงด้วยการเพิ่มพจน์ "+1" เข้ามาในสมการเพื่อให้โครโมโซมที่แย่ที่สุดมีโอกาสในการเข้าสู่กระบวนการคัดสรร เพราะในกรณีที่โครโมโซมตัวที่แย่ที่สุดถูกลบด้วยตัวของมันเองแล้วจะทำให้โอกาสในการอยู่รอดเท่ากับศูนย์ สามารถเขียนเป็นรูปสมการได้ดังนี้

$$eval(v_k) = (x_w - x_k) + 1 \tag{2}$$

เมื่อ x_w คือ โครโมโซมที่มีค่าคำตอบแย่ที่สุด

x_k คือ ค่าคำตอบของโครโมโซมที่ k โดย $k = 1, 2, \ldots, pop\_size$

<u>ตัวอย่าง 1</u> สมมุติให้ $x_1 = 13$, $x_2 = 23$ และ $x_3 = 33$ จากข้อมูลข้างต้นนี้ทำให้ทราบว่าโครโมโซมตัวที่แย่ที่สุด (x_w) คือ $x_3 = 33$ และเมื่อนำไปแทนในสมการ (2) จะทำให้ได้ค่าความเหมาะสมของโครโมโซมแต่ละตัวดังนี้

$$eval(v_1) = (33 - 13) + 1 = 20$$

$$eval(v_2) = (33 - 23) + 1 = 10$$

$$eval(v_3) = (33 - 33) + 1 = 1$$

ด้วยการคำนึงถึงความต้องการของปัญหา ว่าปัญหาต้องการจะ Maximize หรือ Minimize จึงทำให้การหาค่าความเหมาะสม (Fitness value) ของแต่ละโครโมโซมสอดคล้องกับเป้าหมายของปัญหา

ขั้นตอนที่ 4 การคัดสรร (Selection)

กลไกการคัดสรรมีแนวคิดมาจากการคัดสรรพันธุกรรมที่เหมาะสมของธรรมชาติ ซึ่งเป็นไปตามทฤษฎีของชาร์ล ดาร์วิน ในการคัดเลือกผู้มีความเหมาะสมที่จะอยู่รอด (Gen & Cheng, 1997) ในมุมมองของธรรมชาตินั้นพันธุกรรมที่เหมาะสมคือ สิ่งมีชีวิตใดๆก็ตามที่มีคุณสมบัติในการเผชิญกับอุปสรรคต่างๆไม่ว่าจะเป็นโรคระบาด หรือบรรดาผู้ล่า ที่มีผลต่อการเจริญเติบโตและการขยายเผ่าพันธุ์ (Goldberg, 1989) ส่วนในมุมมองของปัญญาประดิษฐ์อย่าง GA นั้น พันธุกรรมที่เหมาะสมคือ โครโมโซมที่ถูกประเมินจากฟังก์ชันเป้าหมาย แล้วมีค่าความเหมาะสมในการอยู่รอดสูงที่สุดนั่นเอง โดยทั่วไปแล้วกลไกการคัดสรร (Sampling mechanism) ที่ใช้ในการคัดสรรโครโมโซมแบ่งออกเป็น 3 ประเภทคือ Stochastic sampling, Deterministic sampling และ Mixed sampling (Gen & Cheng, 1997) ซึ่งจะอธิบายในหัวข้อถัดไปนี้

ประเภทที่ 1 Stochastic sampling

Stochastic sampling คือ วิธีการคัดสรรที่มีรูปแบบไม่แน่นอน รูปแบบการคัดสรรประเภทนี้ได้แก่ รูปแบบการคัดสรรด้วยวงล้อเสี่ยงทาย (Roulette wheel selection) (Goldberg, 1989) และรูปแบบการคัดสรรด้วยการสุ่มอย่างทั่วถึง (Stochastic universal sampling) (Baker, 1987) อธิบายรายละเอียดของแต่ละรูปแบบได้ดังนี้

1 รูปแบบการคัดสรรด้วยวงล้อเสี่ยงทาย (Roulette wheel selection)

รูปแบบการคัดสรรด้วยวงล้อเสี่ยงทายเป็นรูปแบบหนึ่งของวิธีการคัดสรรที่ได้รับความนิยมมาก (Goldberg, 1989) วิธีการ คือ โครโมโซมทุกตัวจะถูกจัดสรรลงบนวงล้อเสี่ยงทายโดย 1 โครโมโซมจะแทน 1 ช่อง ซึ่งจะมีจำนวนช่องบนวงล้อเท่ากับจำนวนโครโมโซมทั้งหมด

และความกว้างของแต่ละช่องก็จะถูกกำหนดโดยความน่าจะเป็นในการถูกเลือกสำหรับแต่ละโครโมโซมตามสัดส่วนค่าความเหมาะสมของโครโมโซมนั้นหากการปั่น (Spin) ของวงล้อเสี่ยงทายตกที่ช่องใดโครโมโซมที่ช่องนั้นอ้างถึงก็จะถูกเลือกให้อยู่รอดในรุ่น (Generation) ถัดไป สำหรับโครโมโซม k แทนค่าความเหมาะสม (Fitness value) ของโครโมโซม k ด้วย f_k และแทนความน่าจะเป็นในการถูกเลือก (Selection probability) ของโครโมโซม k ด้วย p_k (Gen & Cheng, 1997) สามารถหาความน่าจะเป็นในการถูกเลือกและเขียนเป็นรูปสมการได้ดังนี้

$$p_k = \frac{f_k}{\sum_{j=1}^{pop\_size} f_j} \qquad (3)$$

ตัวอย่าง 2 ข้อมูลจากตัวอย่าง 1 สามารถคำนวณความน่าจะเป็นในการถูกเลือกของโครโมโซมด้วยสมการ (3) ดังนี้

ผลรวมของค่าความเหมาะสมของทุกๆ โครโมโซม ($\sum_{j=1}^{pop\_size} f_j$) จะคำนวณได้จาก

$$\sum_{j=1}^{pop\_size} f_j = (f_1 + f_2 + f_3)$$
$$= (20 + 10 + 1) = 31$$

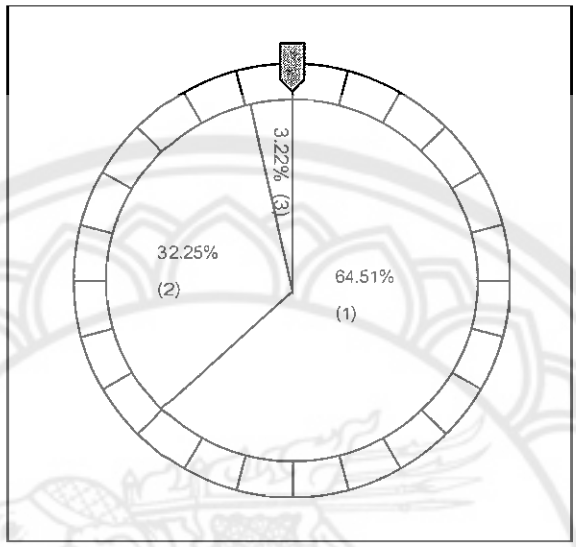
นำ f_k และ $\sum_{j=1}^{pop\_size} f_j$ แทนในสมการ (3) เพื่อคำนวณหาโอกาสในการถูกเลือกของแต่ละโครโมโซมดังนี้

$$p_1 = \frac{20}{31} = 0.6451 \text{ คิดเป็น } 64.51\%$$

$$p_2 = \frac{10}{31} = 0.3225 \text{ คิดเป็น } 32.25\%$$

$$p_3 = \frac{1}{31} = 0.0322 \text{ คิดเป็น } 3.22\%$$

สามารถสร้างวงล้อเสี่ยงทาย (Roulette wheel) ตามความน่าจะเป็นในการถูกเลือกโดยนำค่า p_k ของแต่ละโครโมโซมไปสร้างได้ดังภาพ 12



ภาพ 12 แสดงรูปแบบของวงล้อเสี่ยงทายจากตัวอย่าง 2

การทำงานของวงล้อเสี่ยงทายเพื่อคัดสรรโครโมโซมที่จะผ่านไปในรุ่นถัดไปนั้น จะเกิดขึ้นจากการหมุนวงล้อเสี่ยงทายเท่ากับจำนวนของโครโมโซมหรือค่า *pop_size* (จากตัวอย่างนี้วงล้อเสี่ยงทายจะหมุนเป็นจำนวน 3 ครั้ง) โดยที่แต่ละครั้งของการหมุนนั้น หากมาร์คเกอร์ (Marker:) ชี้ไปที่ช่องของโครโมโซมใดโครโมโซมนั้นก็จะได้ผ่านไปยังรุ่นถัดไป แล้วจึงหมุนในครั้งต่อไปจนได้ครบทุกโครโมโซม

2 รูปแบบการคัดสรรด้วยการสุ่มอย่างทั่วถึง (Stochastic universal sampling)

รูปแบบการคัดสรรด้วยการสุ่มอย่างทั่วถึง (Stochastic universal sampling) (Baker, 1987) จะมีการสร้างวงล้อเหมือนกับ Roulette wheel และหมุนวงล้อเท่ากับจำนวนประชากร (Population size) ซึ่งแทนค่าคาดหวังสำหรับแต่ละโครโมโซม k ด้วย e_k คำนวณ e_k ได้จาก $e_k = pop\_size \times p_k$ โดยสามารถคำนวณ p_k ได้จากสมการ (3) อธิบายกระบวนการทำงานของ Stochastic universal sampling ดังนี้ (Gen & Cheng, 1997)

Procedure: Stochastic Universal Sampling

```
begin
    sum ← 0;
    ptr ← rand();
        for   k ← 1  to pop_size do
            sum ← sum + e_k;
                while  (sum > ptr)  do
                    select chromosome k;
                    ptr ← ptr + 1;
                end
        end
end
```

เมื่อ *rand*() ส่งคืนค่าที่ได้จากการสุ่มเป็นเลขจำนวนจริงที่มีขอบเขตอยู่ระหว่าง [0,1]

ประเภทที่ 2 Deterministic sampling

Deterministic sampling คือ วิธีการคัดสรรที่มีรูปแบบแน่นอน ตัวอย่างรูปแบบของการคัดสรรประเภทนี้คือ Truncation selection, Block selection และ Elitist selection ซึ่งโครโมโซมทั้งหมดจะถูกจัดลำดับความเหมาะสมและเลือกตัวดีที่สุด (Gen & Cheng, 1997) อธิบายรายละเอียดของแต่ละรูปแบบดังนี้

1 Truncation selection

การคัดสรรแบบ Truncation selection จะมีการกำหนดค่าเปอร์เซ็นต์ *T* ของโครโมโซมที่ดีที่จะถูกเลือก และแต่ละโครโมโซมจะถูกเลือกไปเป็นจำนวนเท่ากับ 100/*T* ครั้ง

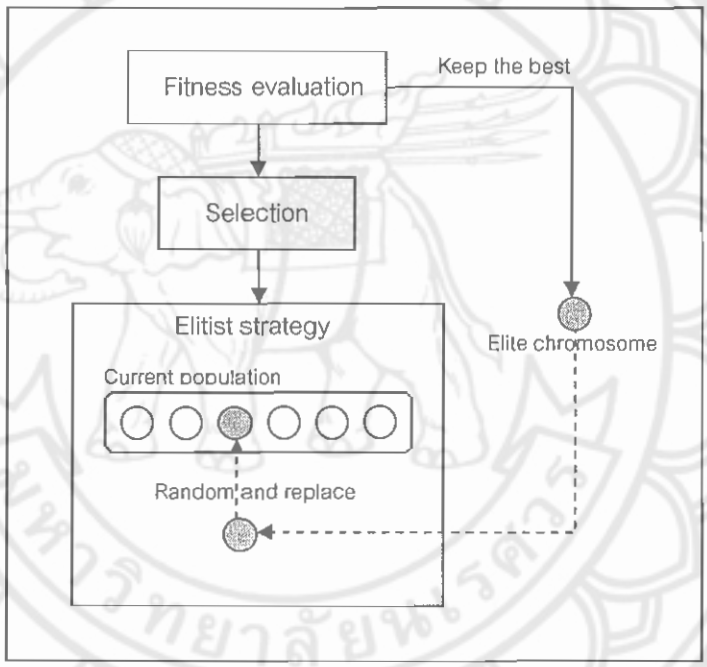
2 Block selection

การคัดสรรแบบ Block selection จะคล้ายกับ Truncation selection แต่จะต่างกันตรงที่ จะมีการกำหนดค่าเปอร์เซ็นต์ของโครโมโซมที่ดีที่จะถูกเลือกเท่ากับ *pop_size / s* และแต่ละโครโมโซมจะถูกเลือกไปเป็นจำนวนเท่ากับ *s* ครั้ง ซึ่งทั้งสองวิธีการนี้จะเหมือนกันทุกอย่างเมื่อ $s = pop\_size / T$

3 Elitist selection

Elitist selection คือ การเลือกโดยการรักษาเผ่าพันธุ์ชนชั้นดี ในแต่ละรอบการทำงานของ GA จะมีโครโมโซมตัวที่ดีที่สุดที่จะได้รับโอกาสในการอยู่รอด แต่ก็อาจจะเกิดกรณีที่

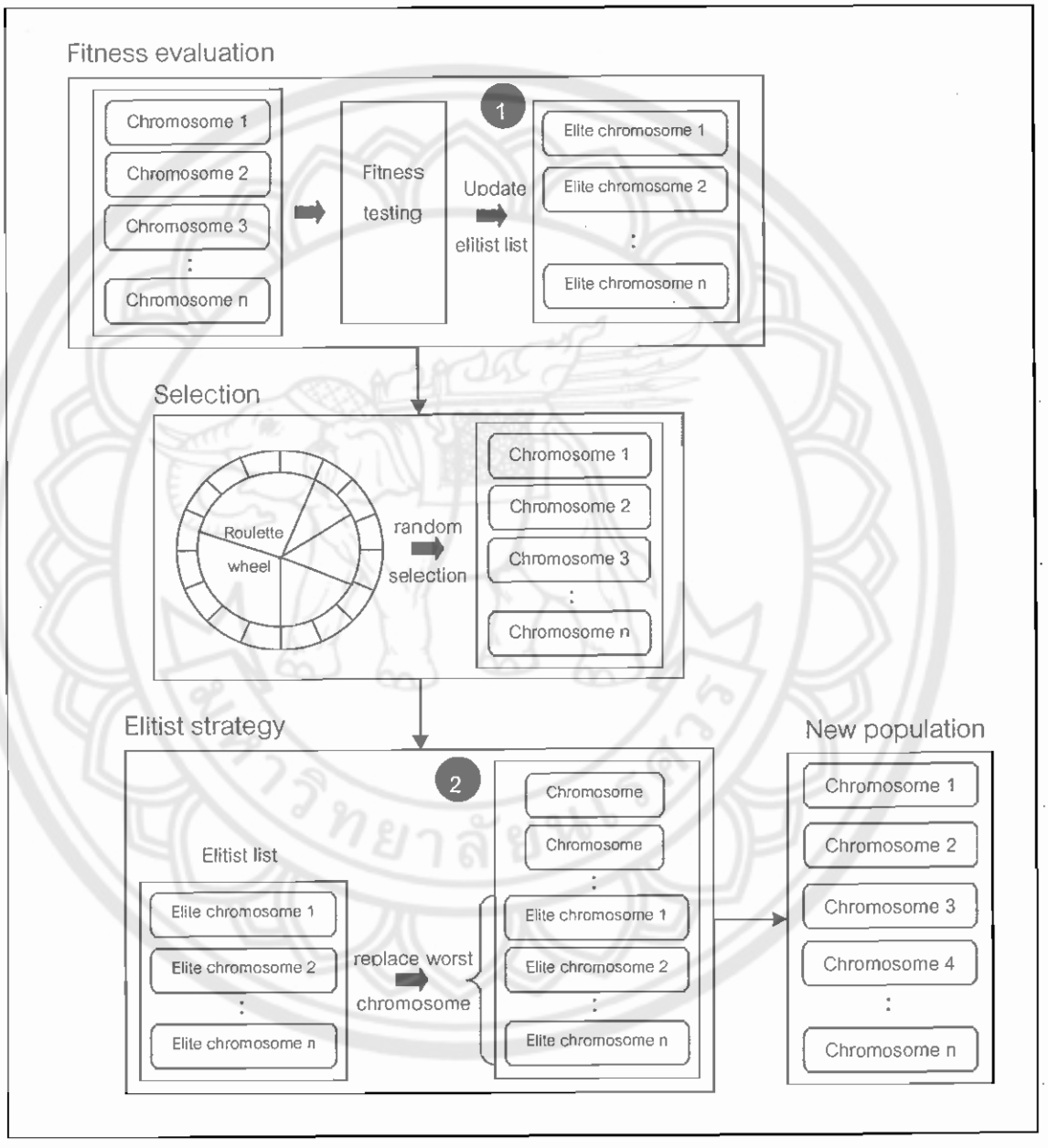
โครโมโซมตัวที่ดีที่สุดนั้นไม่สามารถผ่านไปในรุ่นถัดไปได้ ทำให้การวิวัฒนาการของโครโมโซมเกิดการไม่ต่อเนื่อง ด้วยเหตุนี้ Murata et al. (1996) จึงได้เสนอกลยุทธ์ในการปกป้องโครโมโซมตัวที่ดีที่สุด (Elite chromosomes) ให้สามารถอยู่รอดต่อไปได้ เรียกว่า กลยุทธ์การรักษาเผ่าพันธุ์ชนชั้นดี (Elitist strategy) โดยที่หลังจากโครโมโซมทั้งหมดผ่านขั้นตอนของการประเมินค่าความเหมาะสมแล้ว Elitist strategy จะเลือกเก็บโครโมโซมตัวที่ดีที่สุดเอาไว้เพื่อนำไปแทนในโครโมโซมรุ่นถัดไป โดยกลยุทธ์การรักษาเผ่าพันธุ์ชนชั้นดีของ Murata et al. (1996) นั้นจะสุ่มเลือกโครโมโซมจากประชากรในรุ่นปัจจุบัน (Current population) ขึ้นมา 1 ตัว แล้วจึงแทนโครโมโซมดังกล่าวนี้ด้วยโครโมโซมที่ดีที่สุดที่ได้จากการเก็บเอาไว้ แสดงกลยุทธ์การรักษาเผ่าพันธุ์ชนชั้นดีได้ดังภาพ 13



ภาพ 13 แสดงกลไกการทำงานของ Elitist strategy แบบ Murata et al. (1996)

ต่อมาในปี 2548 วีณา พรหมเทศ ได้ปรับปรุงแนวคิดดังกล่าวนี้ โดยเปลี่ยนจากการสุ่มเลือกโครโมโซมจากประชากรในรุ่นปัจจุบันมาเป็นการเรียงลำดับโครโมโซมแล้วจึงแทนที่โครโมโซมตัวที่แย่ที่สุดด้วยโครโมโซมพันธุ์ดีที่ได้เก็บเอาไว้ นอกจากนั้นยังได้ออกแบบให้สามารถเก็บโครโมโซมพันธุ์ดีได้หลายตัว โดยการกำหนดเปอร์เซ็นต์ของการเก็บโครโมโซมพันธุ์ดีเทียบต่อขนาดของประชากร (Percentage of keeping elite chromosomes: %E) และเก็บรักษากลุ่มของโครโมโซมพันธุ์ดีนี้เอาไว้ในลิสต์ (Elitist list) กลไกการทำงานของ Elitist selection

สามารถแบ่งได้เป็น 2 ส่วน คือ ❶ ส่วนของการเก็บโครโมโซมพันธุ์ดี และ ❷ ส่วนของการแทนที่โครโมโซมพันธุ์ดีลงในโครโมโซมปัจจุบัน แสดงกลไกการทำงานของ Elitist strategy แบบ วีณา พรหมเทศ (2548) ดังภาพ 14 และอธิบายได้ดังนี้

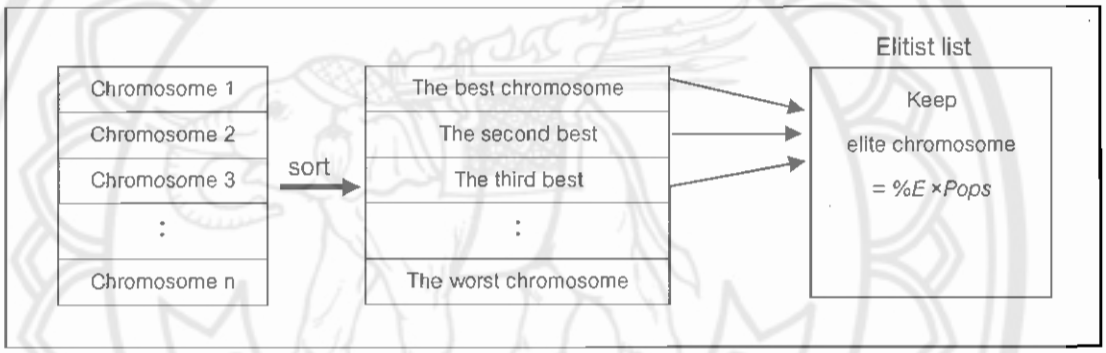


ภาพ 14 แสดงกลไกการทำงานของ Elitist selection

ส่วนแรก ❶ คือ ขั้นตอนการเก็บโครโมโซมพันธุ์ดี

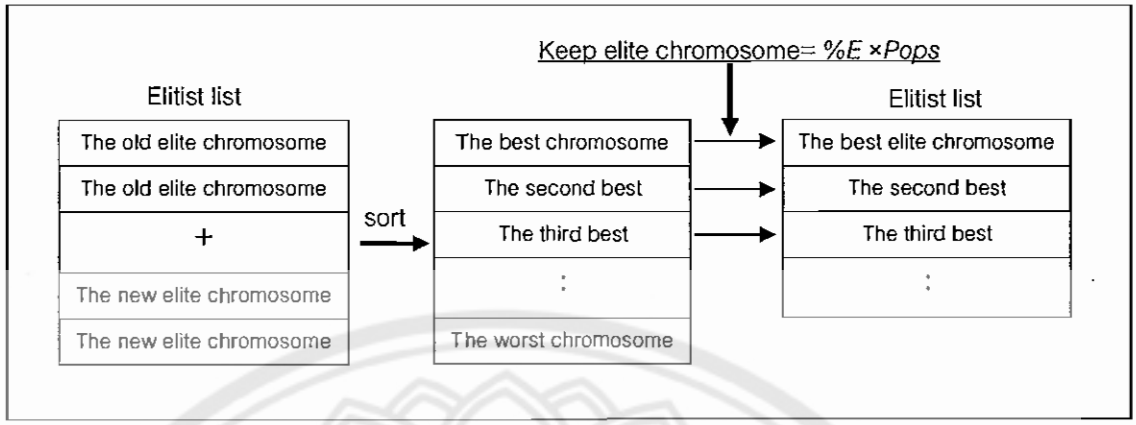
ขั้นที่ 1 กำหนดจำนวนของโครโมโซมพันธุ์ดี (Elite chromosome) ที่ต้องการจะเก็บเอาไว้ในลิสต์ ซึ่งสามารถคำนวณได้จากค่าเปอร์เซ็นต์ของการเก็บโครโมโซมพันธุ์ดีเทียบต่อขนาดของประชากร (%E ×*Pops*)

ขั้นที่ 2 หลังจากโครโมโซมทั้งหมดผ่านขั้นตอนของการประเมินค่าความเหมาะสมแล้ว นำโครโมโซมรุ่นปัจจุบันทั้งหมดมาเรียงลำดับจากค่าความเหมาะสมดีที่สุดไปยังค่าความเหมาะสมแย่ที่สุด จากนั้นทำการเก็บโครโมโซมพันธุ์ดีในรุ่นนี้โดยนับจากโครโมโซมตัวที่ดีที่สุดลงมาจนมีจำนวนโครโมโซมเท่ากับจำนวนโครโมโซมพันธุ์ดีที่ต้องการจะเก็บที่คำนวณได้จากขั้นที่ 1 (ดูภาพ 15)



ภาพ 15 แสดงวิธีการเลือกเก็บโครโมโซมพันธุ์ดี (Elite chromosome)

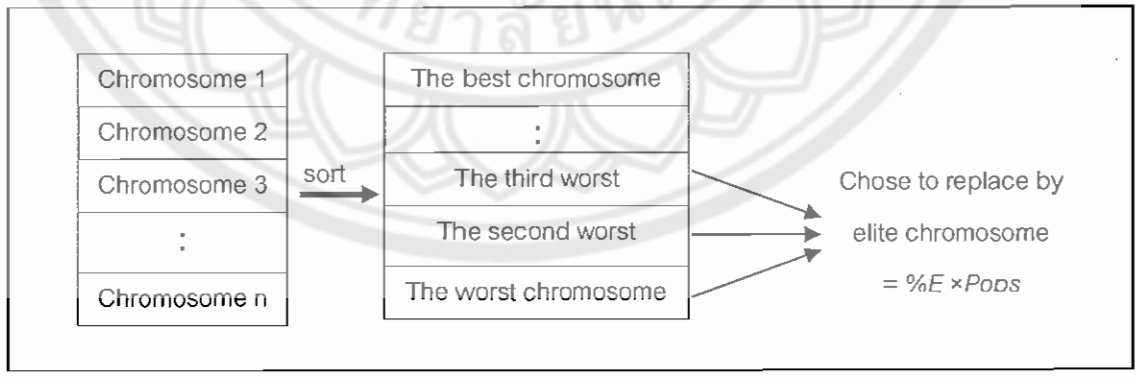
ขั้นที่ 3 ตรวจสอบลิสต์ที่ใช้เก็บโครโมโซมพันธุ์ดี (Elitist list) หากลิสต์ว่างอยู่ก็สามารถนำโครโมโซมที่ได้จากขั้นที่ 2 เก็บลงไปในลิสต์ได้เลย แต่ถ้าหากมีโครโมโซมเดิมเก็บเอาไว้ในลิสต์อยู่แล้วให้นำโครโมโซมเดิมในลิสต์ (The old elite chromosome) และโครโมโซมพันธุ์ดีรุ่นใหม่ (The new elite chromosome) มารวมกันแล้วเรียงลำดับตามค่าความเหมาะสมจากดีที่สุดไปแย่ที่สุด จากนั้นจึงเลือกโครโมโซมตัวที่ดีที่สุดและรองลงมาตามลำดับจนครบตามจำนวนที่กำหนด (ดูภาพ 16)



ภาพ 16 แสดงการจัดเก็บโครโมโซมพันธุ์ดีลงในลิสต์ (Elitis list)

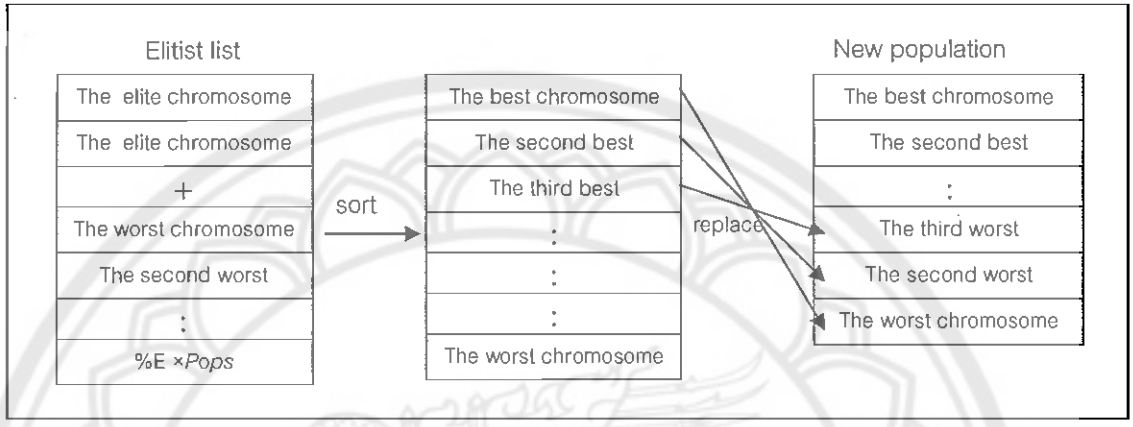
ส่วนที่สอง 2 คือ ขั้นตอนการแทนที่โครโมโซมพันธุ์ดีลงในโครโมโซมปัจจุบัน เมื่อโครโมโซมทั้งหมดเข้าสู่กระบวนการคัดสรรจนได้ประชากรรุ่นถัดไปออกมา ประชากรเหล่านี้ส่วนหนึ่งจะถูกแทนที่ด้วยโครโมโซมพันธุ์ดี (Elite chromosome) ที่ได้เก็บเอาไว้ในลิสต์โดยมีขั้นตอนดังนี้

ขั้นที่ 1 นำโครโมโซมที่ได้ผ่านกระบวนการคัดสรรแล้วมาเรียงลำดับประชากรทั้งหมดตามค่าความเหมาะสมจากดีที่สุดไปแย่ที่สุด จากนั้นเลือกโครโมโซมตัวที่แย่ที่สุดและแย่รองลงมาตามลำดับ จนได้จำนวนโครโมโซมที่จะต้องถูกแทนที่ด้วยโครโมโซมพันธุ์ดีครบตามที่ต้องการ (%E ×*Pops*) (ดูภาพ 17)



ภาพ 17 แสดงโครโมโซมที่ถูกเลือกเพื่อแทนที่ด้วยโครโมโซมพันธุ์ดี (Elite chromosome)

ขั้นที่ 2 นำโครโมโซมพันธุ์ดีที่ถูกเก็บไว้ในลิสต์ (Elitist list) กับกลุ่มของโครโมโซมตัวที่แย่ในรุ่นปัจจุบัน มาเรียงลำดับค่าความเหมาะสมจากดีที่สุดไปแย่ที่สุด จากนั้นจึงเลือกโครโมโซมกลุ่มที่ดีที่สุดไปแทนที่ในตำแหน่งของโครโมโซมกลุ่มที่แย่ในรุ่นปัจจุบัน (ดูภาพ 18)



ภาพ 18 แสดงการแทนที่โครโมโซมพันธุ์ดีลงในประชากรรุ่นปัจจุบัน

นอกจากนี้ในวิทยานิพนธ์ของ วีณา พรหมเทศ (2548) ยังได้ทำการทดลองเพื่อเปรียบเทียบแนวคิดทั้งสองแบบนี้ ซึ่งสรุปได้ว่า แนวคิดที่ได้พัฒนาขึ้นใหม่นี้ช่วยทำให้ GA เจอค่าคำตอบที่ดีกว่า Elitst strategy แบบดั้งเดิมของ Murata et al. (1996) ที่อาศัยการสุ่มเป็นหลัก แต่จำนวนของโครโมโซมพันธุ์ดี (Elite chromosome) ที่จะเก็บนั้น ยังไม่ได้ทำการศึกษาและทดสอบ ว่าควรจะกำหนดไว้ที่เท่าใด ดังนั้นในงานวิทยานิพนธ์นี้จะนำแนวคิดนี้ไปประยุกต์ใช้ และศึกษาถึงจำนวนของโครโมโซมพันธุ์ดีที่จะเก็บ ที่มีความเหมาะสมที่สุดกับ CPP ต่อไป

ประเภทที่ 3 Mixed sampling

Mixed sampling คือ วิธีการคัดสรรที่รวมเอาสองกลยุทธ์ที่กล่าวไว้ก่อนหน้านี้ คือ Stochastic และ Deterministic เข้าไว้ด้วยกัน ตัวอย่างของการคัดสรรประเภทนี้คือ Stochastic tournament selection ที่เสนอโดย Wetzel (1983) และ Tournament selection ที่เสนอโดย Goldberg, Korb and Deb (1989) อธิบายรายละเอียดแต่ละรูปแบบดังนี้ (Gen & Cheng, 1997)

1 Stochastic tournament selection

การคัดสรรแบบ Stochastic tournament selection (Gen & Cheng, 1997) วิธีนี้จะคำนวณความน่าจะเป็นในการถูกเลือกเหมือนกับการทำ Roulette wheel selection ตามสมการ (3) แต่จะมีคู่ลำดับของโครโมโซมซึ่งจะถูกเลือกโดยใช้ Roulette wheel จากนั้นจะนำ

คู่ลำดับมาเปรียบเทียบกัน ถ้าโครโมโซมใดมีค่าความเหมาะสมในการอยู่รอดสูงโครโมโซมนั้นก็จะถูกเลือกเป็นประชากรใหม่ในรุ่นถัดไป และทำซ้ำจนได้จำนวนประชากรครบตามที่กำหนดไว้

2 Tournament selection

การคัดสรรแบบ Tournament selection (Gen & Cheng, 1997) วิธีนี้จะสุ่มเลือกโครโมโซมจำนวน t โครโมโซมจากประชากรทั้งหมด เรียกจำนวนโครโมโซมในเซตนั้นว่า Tournament size และเลือกโครโมโซมตัวที่ดีที่สุดในกลุ่มนั้นให้ผ่านไปยังประชากรรุ่นถัดไป ทำซ้ำจนได้ประชากรครบตามจำนวนที่กำหนด Tournament size ที่กำหนดให้เท่ากับ 2 ($t = 2$) เรียกว่า Binary tournament โดยทั่วไปแล้ว Tournament size ไม่ได้เจาะจงว่าจะต้องมีขนาดเท่าใด ซึ่งในงานวิจัยนี้ได้กำหนด Tournament size ไว้เท่ากับ 2 เช่นกัน อธิบายกระบวนการทำงานดังนี้

Algorithm: Tournament selection

Input: The population $P(T)$ the tournament size $t \in \{1, 2, \ldots, N\}$

Output: The population after selection $P(T)'$

Tournament $(t, J_1, \ldots, J_N)$:

For i ⟵ 1 to N do

J_i' ⟵ best fit individual out of t randomly picked individuals form $\{J_1, \ldots, J_N\}$;

end

return $\{J'_1, \ldots, J'_N\}$;

ขั้นตอนที่ 5 ตรวจสอบเงื่อนไขหยุดการทำงาน

GA จะหยุดการทำงานเมื่อจำนวนรุ่นของประชากร (Number of generation) ครบที่ผู้ใช้กำหนด หรือผู้ใช้กำหนดให้หยุดการทำงานเมื่อค่าคำตอบไม่มีการเปลี่ยนแปลง

3. การออกแบบการทดลองและการวิเคราะห์ข้อมูลเชิงสถิติ

ในการทดลองนั้นถ้าผู้ทดลองต้องการหาข้อสรุปที่มีความหมายจากข้อมูลที่มีอยู่ และถ้ายิ่งปัญหาที่สนใจนั้นเกี่ยวข้องกับความผิดพลาดในการทดลอง (Experimental error) วิธีการทางสถิติเป็นวิธีการเพียงอย่างเดียวเท่านั้นที่จะสามารถนำมาใช้ในการวิเคราะห์ผลการทดลองนั้นได้ ดังนั้นสิ่งสำคัญ 2 ประการสำหรับปัญหาที่เกี่ยวกับการทดลองก็คือ การออกแบบการทดลองและการวิเคราะห์ข้อมูลทางสถิติ ซึ่งศาสตร์ทั้งสองนี้มีความเกี่ยวข้องกันอย่างมาก ทั้งนี้เพราะว่าวิธีการ

วิเคราะห์ทางสถิติที่เหมาะสมนั้น จะขึ้นกับการออกแบบการทดลองที่จะนำมาใช้ (Montgomery, 1997; ปารเมศ ชุติมา, 2545) ดังนั้นในหัวข้อนี้จึงสามารถแบ่งออกได้เป็น 2 ข้อย่อยดังนี้

3.1 การออกแบบการทดลอง

3.2 การวิเคราะห์ข้อมูลทางสถิติ

3.1 **การออกแบบการทดลอง** (The design of the experiment)

การออกแบบการทดลองทางสถิติ (Statistical design of experiment) หมายถึง กระบวนการในการวางแผนการทดลอง เพื่อจะได้มาซึ่งข้อมูลที่เหมาะสมที่สามารถนำไปใช้ในการวิเคราะห์โดยวิธีการทางสถิติทำให้ผู้วิจัยสามารถหาข้อสรุปที่สมเหตุผลได้ ส่วนคำว่า การทดลอง (Experiment) นั้นหมายถึง การทดสอบ (Test) หรือชุดของการทดสอบที่มีการเปลี่ยนแปลงกับตัวแปรขาเข้า (Input variable) ของกระบวนการหรือระบบ เพื่อสังเกตหรือบ่งชี้ถึงเหตุผลของการเปลี่ยนแปลงที่จะเกิดขึ้นกับผลตอบสนองขาออก (Output response) การทดลองส่วนมากจะเกี่ยวข้องกับปัจจัยหลายตัวและวัตถุประสงค์ของผู้ทำการทดลองก็คือ หาผลกระทบของปัจจัยเหล่านี้กับผลตอบสนองของระบบ ซึ่งจะเรียกการวางแผนและดำเนินการทดลองว่า กลยุทธ์ของการทดลอง (Strategy of experimentation) (Montgomery, 1997; ปารเมศ ชุติมา, 2545) ซึ่งมีกลยุทธ์หลายอย่างที่ผู้ทดลองสามารถนำไปใช้ได้ ดังนั้นในการวางแผนและดำเนินการทดลองใดๆ จึงจำเป็นจะต้องมีการเลือกใช้กลยุทธ์ของการทดลองให้เหมาะสม ดังนั้นในงานวิจัยนี้จึงขอกล่าวถึงเฉพาะกลยุทธ์ที่ใช้ในการศึกษาครั้งนี้เท่านั้นซึ่งแบ่งออกเป็นหัวข้อย่อยต่างๆดังนี้

3.1.1 การออกแบบการทดลองเชิงแฟกทอเรียล (Factorial designs)

3.1.2 การออกแบบการทดลองเชิงแฟกทอเรียลแบบสองระดับ (2^k factorial designs)

3.1.3 การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบสองระดับ (Two-level fractional factorial designs)

3.1.4 การออกแบบการทดลองเชิงแฟกทอเรียลแบบสามระดับ (3^k factorial designs)

3.1.5 การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบสามระดับ (Fractional replication of the 3^k factorial designs)

3.1.6 การออกแบบลาตินสแควร์ (Latin squares)

3.1.1 การออกแบบการทดลองเชิงแฟกทอเรียล (Factorial designs)

การออกแบบเชิงแฟกทอเรียล หมายถึง การทดลองที่พิจารณาถึงผลที่เกิดจากการรวมกันของระดับ (Level) ของปัจจัยทั้งหมดที่เป็นไปได้ในการทดลองนั้น การทดลองส่วนใหญ่จะเกี่ยวข้องกับการศึกษาถึงผลของปัจจัย (Factor) ตั้งแต่ 2 ปัจจัยขึ้นไป โดยในกรณีเช่นนี้การออกแบบเชิงแฟกทอเรียลจะเป็นวิธีการทดลองที่มีประสิทธิภาพสูงสุด เช่นกรณีที่มี 2 ปัจจัย คือ A และ B ถ้าปัจจัย A ประกอบด้วย a ระดับ และปัจจัย B ประกอบด้วย b ระดับ ดังนั้นในการทดลอง 1 การทำซ้ำ (Replicate) จะประกอบด้วยการทดลองร่วมปัจจัยทั้งหมด ab การทดลอง และกล่าวได้ว่าปัจจัยเหล่านี้มีการไขว้ ซึ่งกันและกัน เมื่อปัจจัยที่เกี่ยวข้องถูกนำมาจัดให้อยู่ในรูปแบบของการออกแบบเชิงแฟกทอเรียล ผลที่เกิดจากปัจจัยหนึ่ง หมายถึง การเปลี่ยนแปลงที่เกิดขึ้นกับผลตอบสนองที่เกิดจากการเปลี่ยนแปลงระดับของปัจจัยนั้นๆ ซึ่งเรียกว่า ผลหลัก (Main Effect) และในบางการทดลองอาจจะพบว่าความแตกต่างของผลตอบสนองที่เกิดขึ้นบนระดับต่างๆของปัจจัยหนึ่งจะมีค่าไม่เท่ากันที่ระดับอื่นๆ ทั้งหมดของปัจจัยอื่น ซึ่งหมายความว่า ผลตอบสนองของปัจจัยหนึ่งจะขึ้นกับระดับของปัจจัยอื่นๆ นั่นเอง และเรียกเหตุการณ์แบบนี้ว่า Interaction คือ การมีปฏิสัมพันธ์ต่อกันระหว่างปัจจัยที่เกี่ยวข้อง (Montgomery, 1997; ปารเมศ ชุติมา, 2545)

3.1.2 การออกแบบการทดลองเชิงแฟกทอเรียลแบบสองระดับ (2^k factorial designs)

การออกแบบ 2^k มีประโยชน์มากต่องานทดลองในช่วงเริ่มแรก เพื่อกรองปัจจัยที่มีอยู่เป็นจำนวนมากให้เหลือน้อยลง การออกแบบเช่นนี้จะทำให้เกิดการทดลองจำนวนน้อยที่สุดที่สามารถจะทำได้เพื่อศึกษาถึงผลของปัจจัยทั้ง k ชนิดได้อย่างบริบูรณ์ โดยการออกแบบ 2^k คือ การออกแบบในกรณีที่มีปัจจัย k ปัจจัย ซึ่งแต่ละปัจจัยประกอบด้วย 2 ระดับ และแทน 2 ระดับนี้ด้วยระดับ “สูง” หรือ "ต่ำ" ของปัจจัยหนึ่งๆ โดยใน 1 เรพลิเคตที่สมบูรณ์สำหรับการออกแบบนี้จะประกอบด้วยข้อมูลทั้งสิ้น $2 \times 2 \times \ldots \times 2 = 2^k$ ข้อมูล (Montgomery, 1997; ปารเมศ ชุติมา, 2545)

3.1.3 การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบสองระดับ (Two-level fractional factorial designs)

การออกแบบเศษส่วนเชิงแฟกทอเรียล (Fractional factorial design) จัดได้ว่าเป็นการออกแบบที่มีการใช้งานอย่างแพร่หลายในการออกแบบผลิตภัณฑ์และกระบวนการ นอกจากนั้นแล้วยังใช้ช่วยในการหาแนวทางในการปรับปรุงประสิทธิภาพของกระบวนการอีกด้วย ดังนั้นเมื่อจำนวนปัจจัยในการออกแบบ 2^k เพิ่มขึ้น จำนวนการทดลองสำหรับเรพลิเคตที่สมบูรณ์ก็

จะเพิ่มขึ้นมากเกินกว่าที่ทรัพยากรที่มีอยู่จะรองรับได้ ฉนั้นการออกแบบเศษส่วนเชิงแฟกทอเรียลจึงถูกนำมาใช้ในการกรองเพื่อหาปัจจัยที่มีผล (Screening experiment) กล่าวคือ ในการทดลองหนึ่งอาจจะมีปัจจัยมากมายที่ผู้วิจัยให้ความสนใจ ซึ่งผู้วิจัยจะใช้การออกแบบเศษส่วนเชิงแฟกทอเรียลนี้ค้นหาว่ามีปัจจัยตัวใดบ้างที่เป็นปัจจัยที่มีผลอย่างมีนัยสำคัญ โดยอาจจะใช้การออกแบบเศษส่วนแบบ 2^{k-1} (The one-half fraction of the 2^k design) ซึ่งเป็นการออกแบบครึ่งหนึ่ง (1/2) ของการทดลองแบบ 2^k หรือหากการทดลองมีปัจจัยที่เพิ่มมากขึ้นอีกก็อาจจะใช้การออกแบบเศษส่วนแบบ 2^{k-2} (The one-quarter fraction of the 2^k design) ซึ่งเป็นการออกแบบหนึ่งในสี่ (1/4) ของการทดลองแบบ 2^k ก็ได้ โดยเทคนิคที่ใช้ในการออกแบบลักษณะนี้จะเรียกว่า "การคอนฟาวด์ (Confounding)" ซึ่งเทคนิคนี้จะทำให้ข่าวสารบางอย่างเกี่ยวกับผลของปัจจัยบางตัวอยู่ในรูปที่ไม่สามารถแยกแยะได้ (Indistinguishable form) หรือในบางครั้งจะเรียกว่า ถูกคอนฟาวด์ในบล๊อก (Block) [ศึกษารายละเอียดเพิ่มเติมได้จาก Montgomery (1997) และ ปารเมศ ชุติมา (2545)]

3.1.4 การออกแบบการทดลองเชิงแฟกทอเรียลแบบสามระดับ (3^k factorial designs)

การออกแบบกรณีที่มีปัจจัยที่ต้องพิจารณา k ปัจจัย และปัจจัยทุกตัวประกอบด้วย 3 ระดับ เรียกการออกแบบนี้ว่า การออกแบบเชิงแฟกทอเรียลแบบ 3^k ซึ่งระดับทั้งสามของแต่ละปัจจัยมีค่าเป็น ต่ำ ปานกลาง และ สูง สัญลักษณ์ที่ใช้แทนระดับทั้งสามอาจใช้เป็นตัวเลข 0, 1, 2 แทน ต่ำ ปานกลาง และ สูง ตามลำดับ หรือเป็นตัวเลข -1, 0, 1 ก็ได้ ในการทดลอง 1 เรพลิเคตที่สมบูรณ์สำหรับการออกแบบเช่นนี้จะประกอบด้วยข้อมูลทั้งสิ้น $3 \times 3 \times 3 \times \ldots \times 3 = 3^k$ ข้อมูล โดยการทดลองร่วมปัจจัยในการออกแบบ 3^k จะแทนด้วยตัวเลข k ตัว ซึ่งตัวเลขตัวแรกแทนระดับของปัจจัย A, ตัวเลขตัวที่สองแทนระดับของปัจจัย B,...,และตัวเลขตัวที่ k แทนระดับของปัจจัย k (Montgomery, 1997; ปารเมศ ชุติมา, 2545)

3.1.5 การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบสามระดับ (Fractional replication of the 3^k factorial designs)

เมื่อจำนวนปัจจัยในการออกแบบ 3^k เพิ่มขึ้น จำนวนการทดลองสำหรับเรพลิเคตที่สมบูรณ์ก็จะเพิ่มขึ้นมากเกินกว่าที่ทรัพยากรที่มีอยู่จะรองรับได้ ดังนั้นการออกแบบเศษส่วนเชิงแฟกทอเรียลแบบสามระดับจะช่วยลดจำนวนการทดลองของการออกแบบเชิงแฟกทอเรียลแบบ 3^k ได้ (Montgomery, 1997) ดังนั้นในหัวข้อนี้จะขอกล่าวถึง 2 ข้อย่อยดังนี้

3.1.5.1 **การออกแบบเศษหนึ่งส่วนสามเชิงแฟกทอเรียลแบบสามระดับ** (The one-third fraction of the 3^k factorial design)

การออกแบบเชิงเศษส่วนที่มีขนาดใหญ่ที่สุดสำหรับการออกแบบ 3^k คือ การออกแบบเศษหนึ่งส่วนสามที่เหลือการรัน (Run) เพียง 3^{k-1} รันเท่านั้น ดังนั้นอาจเรียกการออกแบบนี้ได้ว่า "การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบ 3^{k-1}" (Montgomery, 1997) ซึ่งในการออกแบบ 3^{k-1} นั้นจะใช้เทคนิคของการคอนฟาวด์ในการสร้างบล๊อก (Block) ขึ้นมาเช่นเดียวกับแบบสองระดับยกตัวอย่างเช่น การออกแบบเชิงแฟกทอเรียลแบบ 3^3 นั้นจะมีการรันทั้งสิ้น 27 รัน แต่การออกแบบเศษหนึ่งส่วนสามแบบ 3^{3-1} จะช่วยลดให้เหลือการรันจริงๆเพียง 9 รัน หรือเพียงหนึ่งในสามของการรันทั้งหมดเท่านั้น สามารถเขียนเป็นสมการทางคณิตศาสตร์ได้ดังนี้

$$x_1 + 2x_2 + 2x_3 = u \pmod 3 \qquad (4)$$

เมื่อ x_1, x_2 และ x_3 แทนปัจจัยที่ 1, 2 และ 3 ตามลำดับ

จากสมการ (4) จะทำให้การรันแบ่งออกเป็นสามกลุ่มตามค่า u นั่นคือ 0, 1 และ 2 และสามารถที่จะเลือกกลุ่มใดกลุ่มหนึ่งไปใช้ก็ได้ ซึ่งจะทำให้เหลือการรันที่จำเป็นเพียง 9 รันเท่านั้น ดังตาราง 5

ตาราง 5 แสดงการออกแบบเศษหนึ่งส่วนสามแบบ 3^{3-1}

กลุ่มที่ 1 u = 0	กลุ่มที่ 2 u = 1	กลุ่มที่ 3 u = 2
000	100	200
012	112	212
101	201	001
202	002	102
021	121	221
110	210	010
122	222	022
211	011	111
220	020	120

เมื่อ 012 หมายถึง การกำหนดระดับต่ำ (0) ให้กับปัจจัยที่ 1, กำหนดระดับกลาง (1) ให้กับปัจจัยที่ 2 และกำหนดระดับสูง (2) ให้กับปัจจัยที่ 3

3.1.5.2 การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบ 3^{k-p} (Other 3^{k-p} fractional factorial design)

การทดลองในกรณีที่มีปัจจัย k เป็นจำนวนมาก และทรัพยากรที่มีอยู่ไม่สามารถรองรับได้ ดังนั้นการออกแบบที่ซับซ้อนกว่าเพื่อลดจำนวนการรันจึงถูกนำมาพิจารณา และโดยทั่วไปแล้ว มักจะอยู่ในรูปของเศษส่วน $\frac{1}{3^p}$ ของการออกแบบ 3^k หรือ การออกแบบเศษส่วนเชิงแฟกทอเรียลแบบ 3^{k-p} เมื่อ p มีค่าน้อยกว่า k ($p<k$) ซึ่งจะทำให้เหลือการรันเพียง 3^{k-p} รันเท่านั้น ตัวอย่างของการออกแบบในลักษณะนี้ได้แก่ การออกแบบ 3^{k-2} หรือการออกแบบเศษหนึ่งส่วนเก้าเชิงแฟกทอเรียลแบบสามระดับ (One-ninth fraction of 3^k factorial design), การออกแบบ 3^{k-3} หรือการออกแบบเศษหนึ่งส่วนยี่สิบเจ็ดเชิงแฟกทอเรียลแบบสามระดับ (One-twenty-seventh fraction of 3^k factorial design) เป็นต้น (Montgomery, 1997)

3.1.6 การออกแบบลาตินสแควร์ (Latin squares)

วิธีการนี้จะเป็นการบล็อกอย่างเป็นระบบในสองทิศทาง สามารถนำมาใช้เพื่อกำจัดแหล่งรบกวนของความแปรปรวนสองแหล่งได้ ดังนั้นแถวและคอลัมน์จะแสดงถึงข้อจำกัดสองประการในการทำให้เป็นเชิงสุ่ม (Montgomery, 1997; ปารเมศ ชุติมา, 2545) เช่น สมมติว่าผู้ทดลองกำลังศึกษาผลของสูตรเชื้อเพลิง 5 สูตรที่แตกต่างกันซึ่งสูตรเหล่านี้ใช้ในการขับเคลื่อนจรวด โดยผลตอบสนองจะดูจากอัตราการเผาไหม้ที่เกิดขึ้น แต่ละสูตรถูกผสมจากวัตถุดิบรุ่นเดียวกันที่มีขนาดใหญ่เพียงพอสำหรับผลิตทั้ง 5 สูตรที่จะทำการทดสอบ ยิ่งกว่านั้นสูตรจะถูกเตรียมโดยพนักงานคนละคนกัน และอาจมีความแตกต่างอย่างมากในทักษะและประสบการณ์ของพนักงาน ดังนั้นอาจดูเหมือนว่ามีสองปัจจัยรบกวนที่จะต้องเอาออกไปจากการออกแบบนั่นคือ รุ่นของวัตถุดิบและพนักงาน การออกแบบที่เหมาะสมสำหรับปัญหานี้จะประกอบด้วย การทดสอบแต่ละสูตรในแต่ละรุ่นของวัตถุดิบอย่างละหนึ่งครั้ง และสำหรับแต่ละสูตรจะต้องถูกเตรียมขึ้นโดยพนักงานแต่ละคนจาก 5 คน ผลของการออกแบบแสดงในตาราง 6 เรียกการออกแบบนี้ว่า การออกแบบลาตินสแควร์

ตาราง 6 การออกแบบลาตินสแควร์

Batches of raw material	Operators				
	1	2	3	4	5
1	A=24	B=20	C=19	D=24	E=24
2	B=17	C=24	D=30	E=27	A=36
3	C=18	D=38	E=26	A=27	B=21
4	D=26	E=31	A=26	B=23	C=22
5	E=22	A=30	B=20	C=29	D=31

จากตาราง 6 การออกแบบเช่นนี้มีลักษณะการจัดรูปเป็นแบบสี่เหลี่ยมจัตุรัส ดังนั้น 5 สูตร (Treatment) จึงถูกเขียนแทนด้วยอักษรลาติน A, B, C, D และ E ซึ่งเป็นที่มาของชื่อลาตินสแควร์ โดยจะเห็นว่าทั้งรุ่นของวัตถุดิบ (แถว) และพนักงาน (คอลัมน์) เป็นเชิงตั้งฉากกับทรีตเมนต์ [Treatment คือ ระดับของปัจจัยที่ควบคุมได้ (กิติศักดิ์ พลอยพานิชเจริญ, 2547)]

โดยทั่วไปลาตินสแควร์สำหรับปัจจัย p ตัว หรือลาตินสแควร์ขนาด $p \times p$ เป็นสี่เหลี่ยมจัตุรัส ซึ่งประกอบด้วย p แถวและ p คอลัมน์ แต่ละเซลล์ของ p^2 ประกอบด้วยตัวอักษร 1 ใน p ตัวอักษร ซึ่งประกอบกันขึ้นเป็นทรีตเมนต์ และแต่ละตัวอักษรจะปรากฏเพียงครั้งเดียวเท่านั้นในแต่ละแถวและคอลัมน์ (Montgomery, 1997; ปารเมศ ชุติมา, 2545) แสดงตัวอย่างของการออกแบบลาตินสแควร์ได้ดังภาพ 19

4 × 4	5 × 5	6 × 6
A B D C	A D B E C	A D C E B F
B C A D	D A C B E	B A E C F D
C D B A	C B E D A	C E D F A B
D A C B	B E A C D	D C F B E A
	E C D A B	F B A D C E
		E F B A D C

ภาพ 19 แสดงตัวอย่างของการออกแบบลาตินสแควร์

3.2 การวิเคราะห์ข้อมูลเชิงสถิติ (The statistical analysis of the data)

ในงานวิจัยนี้จะกล่าวถึงเฉพาะการวิเคราะห์ข้อมูลทางสถิติที่ใช้ในการศึกษาครั้งนี้เท่านั้นโดยได้แบ่งออกเป็น 3 ข้อย่อยดังนี้

3.2.1 การตรวจสอบสมมติฐาน

ในงานวิจัยบางอย่างผู้วิจัยใช้วิธีการบรรยายปรากฏการณ์นั้นไม่ได้หรือไม่เพียงพอ ดังนั้นจึงต้องมีการตรวจสอบสมมติฐาน ซึ่งสมมติฐานเป็นคำตอบที่ผู้วิจัยคาดการณ์ไว้ล่วงหน้าก่อนทำการวิจัย โดยคำตอบอาจจะถูกหรือผิดก็ได้จึงจำเป็นอย่างยิ่งที่จะต้องมีการทดสอบสมมติฐานก่อน การทดสอบสมมติฐาน คือ ขั้นตอนในการตรวจสอบคำตอบที่คาดการณ์ไว้ล่วงหน้าว่าจะตรงกับคำตอบที่ได้จากข้อมูลที่เก็บรวบรวมมาหรือไม่ (ยุทธ ไกยวรรณ์, 2546)

สมมติฐานที่ใช้กันอยู่ในงานวิจัยแบ่งออกเป็น 2 ประเภท คือ

1. สมมติฐานในการวิจัย (Research hypothesis) หรือสมมติฐานเชิงพรรณนา (Descriptive hypothesis) เป็นสมมติฐานที่ผู้วิจัยเขียนในเชิงพรรณนา หรือเขียนให้อยู่ในรูปของข้อความภาษาที่ใช้สื่อในการอธิบาย

2. สมมติฐานทางสถิติ (Statistical hypothesis) เป็นสมมติฐานที่เขียนเป็นสัญลักษณ์ทางสถิติแทนคำอธิบายหรือคำพูด เพื่อให้สามารถทดสอบด้วยวิธีการทางสถิติได้ สมมติฐานทางสถิติแบ่งออกได้ 2 แบบ คือ 1) สมมติฐานไร้นัยสำคัญ (Null hypothesis) เป็นสมมติฐานที่ไม่แสดงความแตกต่างระหว่างกลุ่มต่างๆที่ผู้วิจัยจะนำมาพิสูจน์ เขียนแทนด้วย H_0 และ 2) สมมติฐานทางเลือก (Alternative hypothesis) เป็นสมมติฐานที่แสดงความแตกต่าง หรือความไม่เท่ากันของค่าเฉลี่ยของกลุ่มตัวอย่างหรือค่าพารามิเตอร์ของประชากรที่นำมาศึกษา เป็นสมมติฐานที่มีลักษณะตรงข้ามกับ Null hypothesis เขียนแทนด้วย H_1 ซึ่ง H_1 จะต้องอยู่คู่กับ H_0 เสมอเพื่อให้เกิดความชัดเจนในการทดสอบทางสถิติ (ยุทธ ไกยวรรณ์, 2546) ในการทดสอบสมมติฐาน การที่ผู้วิจัยกำหนดขอบเขตของความคลาดเคลื่อนที่จะยอมให้เกิดขึ้นได้ในการทดสอบสมมติฐานเรียกว่า ระดับนัยสำคัญ (Level of significance) ซึ่งเป็นระดับของความผิดพลาดจากการกระทำหรือการทดลอง โดยจะกำหนดความผิดพลาดที่ 0.05 หรือ 0.01 หรือ 0.001 ทั้งนี้ขึ้นอยู่กับความสำคัญของสิ่งที่ทำการทดลอง ถ้าเกิดความคลาดเคลื่อนน้อยกว่าที่กำหนดก็จะยอมรับ H_0 ที่ตั้งไว้ ในทางตรงกันข้ามหากเกิดความคลาดเคลื่อนมากกว่าที่กำหนดจะปฏิเสธ H_0 ที่ตั้งไว้นั้นหมายความว่ายอมรับ H_1 อย่างไรก็ตามการยอมรับ H_0 หรือปฏิเสธ H_0 จะถูกต้องหรือไม่นั้นก็ขึ้นอยู่กับข้อมูลจากกลุ่มตัวอย่าง ฉนั้นการตัดสินใจบางครั้งจึงมีโอกาสผิดพลาดขึ้นได้ โดยทั่วไปการ

ตัดสินใจในการทดสอบสมมติฐานทางสถิติมี 4 ลักษณะ สามารถเขียนแสดงตารางการตัดสินใจ 4 อย่างกับสถานการณ์ที่เกิดขึ้นในการทดสอบสมมติฐานได้ดังนี้

ตาราง 7 แสดงผลการทดสอบสมมติฐาน (ยุทธ ไกยวรรณ์, 2546)

ผลการทดสอบ	ยอมรับ H_0	ไม่ยอมรับ H_0
H_0 เป็นจริง	ตัดสินใจถูกต้อง	Type I Error หรือ α Error
H_0 ไม่เป็นจริง	Type II Error หรือ β Error	ตัดสินใจถูกต้อง

จากตาราง 7 อธิบายตารางการตัดสินใจในการทดสอบสมมติฐานทางสถิติ 4 ลักษณะดังนี้ 1. ยอมรับ H_0 เมื่อสมมติฐานนั้นเป็นจริงถือว่าตัดสินใจถูกต้อง 2. ไม่ยอมรับ H_0 ทั้งที่สมมติฐานนั้นเป็นจริงถือว่าตัดสินใจเกิดความผิดพลาดแบบที่ I (Type I Error) 3. ยอมรับ H_0 ทั้งที่สมมติฐานนั้นไม่เป็นจริงถือว่าตัดสินใจเกิดความผิดพลาดแบบที่ II (Type II Error) และ 4. ไม่ยอมรับ H_0 เมื่อสมมติฐานนั้นไม่เป็นจริงถือว่าตัดสินใจถูกต้อง

ในการทดสอบสมมติฐานแต่ละครั้งจะมีการเสี่ยงต่อความผิดพลาดในการตัดสินใจทั้งแบบที่ 1 และแบบที่ 2 เสมอ ดังนั้นในการทดสอบผู้วิจัยจะต้องพยายามลดความผิดพลาดลงให้น้อยที่สุด วิธีการก็คือ เลือกการสุ่มตัวอย่างที่ถูกต้อง และรวบรวมข้อมูลมาอย่างสมบูรณ์ (ยุทธ ไกยวรรณ์, 2546)

3.2.2 **การวิเคราะห์ความแปรปรวน** (Analysis of variance: ANOVA)

ในการวิเคราะห์ความแปรปรวนจะเป็นการวิเคราะห์อัตราส่วนระหว่างความแปรปรวนระหว่างกลุ่ม (Between-group variance) และความแปรปรวนภายในกลุ่ม (Within-group variance) ความแปรปรวนระหว่างกลุ่มเป็นค่าที่เกิดจากความแตกต่างของค่าเฉลี่ยระหว่างกลุ่มต่างๆ ถ้าค่าเฉลี่ยระหว่างกลุ่มต่างๆแตกต่างกันมาก ค่าความแปรปรวนระหว่างกลุ่มก็จะมากตามไปด้วย สำหรับความแปรปรวนภายในกลุ่มเป็นค่าที่แสดงให้เห็นว่า

คะแนนแต่ละตัวที่รวบรวมมานั้นภายในแต่ละกลุ่มมีการกระจายมากหรือน้อย ค่าที่คำนวณได้เรียกว่า ความคลาดเคลื่อน (ชูศรี วงศ์รัตนะ, 2534)

การวิเคราะห์ความแปรปรวนโดยทั่วไปจะใช้เพื่อวิเคราะห์ผลจากการทดลองเชิงแฟกทอเรียลตัวอย่างเช่น การทดลองเชิงแฟกทอเรียลในกรณีที่มีปัจจัยที่จะทำการศึกษา 2 ปัจจัย คือ A และ B โดยปัจจัย A ประกอบด้วย a ระดับ และปัจจัย B ประกอบด้วย b ระดับ ซึ่งทั้งหมดนี้ถูกจัดให้อยู่ในรูปของการออกแบบเชิงแฟกทอเรียลนั่นคือ ในแต่ละเรพลิเคตของการทดลองจะประกอบด้วยการทดลองร่วมปัจจัยทั้งหมด ab การทดลอง โดยปกติจะมีจำนวนเรพลิเคตทั้งหมด n ครั้ง รูปแบบทั่วไปของการออกแบบเชิงแฟกทอเรียล 2 ปัจจัยและมีการทำซ้ำทั้งหมด n ครั้ง สามารถแสดงได้ดังตาราง 8 เมื่อกำหนดให้ y_{ijk} คือ ผลตอบสนองที่เกิดจากระดับที่ i ของปัจจัย A (เมื่อ i = 1, 2, ...,a) และระดับที่ j ของปัจจัย B (เมื่อ j = 1, 2, ..., b) สำหรับเรพลิเคตที่ k (เมื่อ k = 1, 2, ..., n) (Montgomery, 1997; ปารเมศ ชุติมา, 2545)

ตาราง 8 แสดงรูปแบบข้อมูลการทดลองเชิงแฟกทอเรียลเมื่อมี 2 ปัจจัย

		Factor B			
		1	2	...	b
Factor A	1	$y_{111}, y_{112}, \ldots, y_{11n}$	$y_{121}, y_{122}, \ldots, y_{12n}$	...	$y_{1b1}, y_{1b2}, \ldots, y_{1bn}$
	2	$y_{211}, y_{212}, \ldots, y_{21n}$	$y_{221}, y_{222}, \ldots, y_{22n}$	...	$y_{2b1}, y_{2b2}, \ldots, y_{2bn}$
	⋮	⋮	⋮	⋮	⋮
	A	$y_{a11}, y_{a12}, \ldots, y_{a1n}$	$y_{a21}, y_{a22}, \ldots, y_{a2n}$	...	$y_{ab1}, y_{ab2}, \ldots, y_{abn}$

ข้อมูลจากตาราง 8 อาจจะเขียนในรูปของแบบจำลองสถิติเชิงเส้น (Linear statistical model) ได้ดังนี้คือ

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk} \qquad (5)$$

เมื่อ i = 1, 2, ..., a

j = 1, 2, ..., b

k = 1, 2, ..., n

โดยที่ y_{ijk} หมายถึง ผลตอบสนองที่สังเกตได้เมื่อปัจจัย A อยู่ที่ระดับ i และปัจจัย B อยู่ที่ระดับ j สำหรับเรพลิเคตที่ k, μ หมายถึง ค่าเฉลี่ยทั้งหมด, τ_i หมายถึง ผลที่เกิดจากระดับที่ i ของแถว (row) ของปัจจัย A, β_j หมายถึง ผลที่เกิดจากระดับที่ j ของคอลัมน์ (column) ของปัจจัย B, $(\tau\beta)_{ij}$ หมายถึง ผลที่เกิดจากปฏิสัมพันธ์ระหว่าง τ_i กับ β_j, ε_{ijk} หมายถึง องค์ประกอบของความผิดพลาดแบบสุ่ม เนื่องจากในการทดลองนี้มีจำนวนเรพลิเคต n ครั้ง ดังนั้นจำนวนข้อมูลที่ได้จากการสังเกตทั้งหมดจึงเท่ากับ abn จำนวน (Montgomery, 1997; ปารเมศ ชุติมา, 2545)

การตรวจสอบความถูกต้องของสมมติฐานตามแบบจำลองสมการ (5) ความคลาดเคลื่อน หรือองค์ประกอบของความผิดพลาดแบบสุ่ม (ε_{ijk}) จะต้องมีการแจกแจงแบบปกติ และเป็นอิสระด้วยค่าเฉลี่ยเท่ากับ 0 และค่าความแปรปรวน σ^2 มีค่าคงตัวแต่ไม่ทราบค่า ถ้าสมมติฐานเหล่านี้เป็นจริงกระบวนการวิเคราะห์ความแปรปรวนนี้ก็จะเป็นการทดสอบสมมติฐานเกี่ยวกับการไม่มีความแตกต่างในค่าเฉลี่ยของระดับที่ถูกต้อง ในทางปฏิบัติการจะเชื่อในผลที่ได้จากการวิเคราะห์ความแปรปรวนนั้นจะต้องตรวจสอบความถูกต้องของสมมติฐานว่าเป็นจริงก่อน ซึ่งการตรวจสอบสมมติฐานขั้นต้นและความถูกต้องของแบบจำลองทำได้โดยการวิเคราะห์ส่วนตกค้าง (Residual analysis) ซึ่งจะมีการตรวจสอบสมมติฐานของความเป็นปกติโดยการทำกราฟความน่าจะเป็นแบบปกติของส่วนตกค้าง (Normal probability plot of the residuals) ถ้าหากการแจกแจงของความผิดพลาดเป็นแบบปกติรูปกราฟที่ได้จะเป็นเส้นตรง, ทำการตรวจ สอบสมมติฐานค่าความแปรปรวน σ^2 มีค่าคงตัวโดยการทำกราฟส่วนตกค้างกับค่าที่ถูกฟิต (Plot of residuals versus fitted values) ถ้าหากแบบจำลองถูกต้องและสมมติฐานมีความเหมาะสมแล้วกราฟที่ได้ไม่ควรจะมีรูปแบบหรือรูปร่างเฉพาะแต่อย่างใดทั้งสิ้น และสุดท้ายคือทำการตรวจสอบสมมติฐานของความเป็นอิสระโดยการทำกราฟส่วนตกค้างตามลำดับเวลาของการเก็บข้อมูล (Plot of residuals versus the order of the data) ถ้าส่วนตกค้างมีแนวโน้มที่จะเพิ่มขึ้นหรือลดลงตามลำดับเวลาของการเก็บข้อมูล หรือกราฟที่ได้มีการกระจายที่ปลายข้างหนึ่งมากกว่าที่ปลายอีกข้างหนึ่ง แสดงว่าสมมติฐานของความเป็นอิสระถูกละเมิด จากสมการ (5) ถ้าสมมติว่าแบบจำลองตามสมการ (5) เป็นแบบจำลองที่เหมาะสม และพจน์ของความผิดพลาด ε_{ijk} มีการกระจายแบบปกติและเป็นอิสระโดยมีค่าความแปรปรวนคงตัวเท่ากับ σ^2 วิธีการทดสอบจะทำโดยอาศัยตารางการวิเคราะห์ความแปรปรวน (ตาราง 9) ซึ่งโดยทั่วไปตาราง ANOVA จะประกอบด้วย แหล่งความแปรปรวน (Source of variation), ผลรวมกำลังสอง (Sum of square: SS), องศาแห่งความอิสระ (Degrees of freedom: DF), ค่าเฉลี่ยกำลังสอง (Mean squares: MS) และ ค่า F (F value)

ตาราง 9 ตารางการวิเคราะห์ความแปรปรวนของการทดลองเชิงแฟกทอเรียล 2 ตัวแปร แบบ Fixed effects model

Source of Variation	Sum of Square	DF	Mean Square	F
ปัจจัย A	$SS_A = \frac{1}{bn}\sum_{i=1}^{a} y_{i..}^2 - \frac{y_{...}^2}{abn}$	a-1	$MS_A = \frac{SS_A}{a-1}$	$\frac{MS_A}{MS_E}$
ปัจจัย B	$SS_B = \frac{1}{an}\sum_{j=1}^{b} y_{j..}^2 - \frac{y_{...}^2}{abn}$	b-1	$MS_B = \frac{SS_B}{b-1}$	$\frac{MS_B}{MS_E}$
AB	$SS_{AB} = \frac{1}{n}\sum_{i=1}^{a}\sum_{j=1}^{b} y_{ij}^2 - \frac{y_{...}^2}{abn} - SS_A - SS_B$	(a-1)(b-1)	$MS_{AB} = \frac{SS_{AB}}{(a-1)(b-1)}$	$\frac{MS_{AB}}{MS_E}$
ความคลาดเคลื่อน	$SS_E = SS_T - SS_A - SS_B - SS_{AB}$	ab(n-1)	$MS_E = \frac{SS_E}{ab(n-1)}$	
ผลรวม	$SS_T = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n} y_{ijk}^2 - \frac{y_{...}^2}{abn}$	abn-1		

3.2.3 **การเปรียบเทียบเชิงซ้อน** (Multiple comparison)

การทดสอบสมมติฐาน โดยวิธีการวิเคราะห์ความแปรปรวนด้วยสถิติ F-test เป็นการทดสอบสมมติฐานเกี่ยวกับความแตกต่างระหว่างค่าเฉลี่ยตั้งแต่สามกลุ่มขึ้นไป ไม่ว่าจะเป็นการวิเคราะห์ความแปรปรวนแบบทางเดียว (One-way ANOVA) หรือสองทาง (Two-way ANOVA) ก็ตามถ้าหากผลการวิเคราะห์ยอมรับ H_0 ซึ่งหมายความว่าปฏิเสธ H_1 นั่นแสดงว่าค่าเฉลี่ยของกลุ่มตัวอย่างนั้นไม่แตกต่างกัน แต่ถ้าผลวิเคราะห์ปฏิเสธ H_0 และยอมรับ H_1 แสดงว่าค่าเฉลี่ยของกลุ่มตัวอย่างนั้นแตกต่างกัน แต่ผู้วิเคราะห์ยังไม่ทราบว่าค่าเฉลี่ยของกลุ่มตัวอย่างที่แตกต่างกันนั้น คู่ใดที่แตกต่างกัน ดังนั้นจึงต้องทำการทดสอบต่อไปว่า มีคู่หรือกลุ่มใดบ้างที่แตกต่างกัน วิธีการทดสอบเพื่อให้ทราบว่าค่าเฉลี่ยของกลุ่มตัวอย่างคู่ใดบ้างที่แตกต่างกันนั้น เรียกว่า การเปรียบเทียบค่าเฉลี่ยภายหลังการวิเคราะห์ความแปรปรวน หรือการเปรียบเทียบพหุคูณ (Multiple comparison test) (ยุทธ ไกยวรรณ์, 2546)

การเปรียบเทียบพหุคูณมีหลายวิธี เช่น การทดสอบของ Duncan's Test (Duncan, 1955), Turkey's Test (Turkey, 1953) และ Newman – Keuls (Newman, 1939; Keuls, 1952) ซึ่งในวิทยานิพนธ์นี้เลือกใช้วิธีการของ Duncan เนื่องจากการเปรียบเทียบค่าเฉลี่ยแบบ Duncan มีวิธีการคล้ายคลึงกับวิธีการเปรียบเทียบแบบ T หรือ แบบ NK จะมีข้อต่างกันตรงที่ $S_{\bar{Y}}$ ของวิธีการแบบ Duncan นี้จะไม่เปิดจากตาราง Studentized Range แต่จะเปิดจากตาราง Duncan's New Multiple Range Test แทน (ยุทธ ไกยวรรณ์, 2546) โดยวิธีการจะทำการพิจารณาว่าทรีตเมนต์แต่ละคู่ที่เปรียบเทียบมีความแตกต่างอย่างมีนัยสำคัญน้อยที่สุด (Least

significant difference: LSD) เท่ากับเท่าใด แล้วทำการเปรียบเทียบความแตกต่างของค่าเฉลี่ยของสิ่งตัวอย่าง ($\overline{Y}_{i.} - \overline{Y}_{j.}$) กับค่า LSD ดังกล่าว โดยแต่ละวิธีจะมีความแตกต่างกันเพียงสถิติสำหรับการทดสอบเท่านั้น หลักการทดสอบเพื่อเปรียบเทียบเชิงซ้อนโดยวิธีการของดันแคนนี้จะต้องดำเนินการภายใต้ขนาดสิ่งตัวอย่างเท่ากันในแต่ละทรีตเมนต์ และโดยที่แต่ละค่าเฉลี่ยของทรีตเมนต์เป็นตัวสถิติซึ่งมีความคลาดเคลื่อนมาตรฐาน ดังสมการ (6)

$$S_{\overline{Y}_{i.}} = \sqrt{\frac{MS_E}{n}} \qquad (6)$$

ในการหาค่าความแตกต่างอย่างมีนัยสำคัญที่น้อยที่สุดของดันแคนนี้ จะอาศัยการกำหนดความแตกต่างให้อยู่ในรูปของพิสัย จึงอาจเรียกว่าพิสัยที่มีนัยสำคัญที่น้อยที่สุด (R_p) โดยที่

$$R_p = r_\alpha (p,f)\ S_{\overline{Y}_{i.}} \qquad (7)$$

เมื่อ r_α (p,f) คือ ค่าพิสัยที่มีนัยสำคัญ (Significant range) ซึ่งหาค่าได้จากตารางพิสัยนัยสำคัญของดันแคน (Duncan's table of significant ranges) และ p คือ จำนวนทรีตเมนต์ที่ทำการศึกษา ในขณะที่ f คือ ค่า DF ของตัวประมาณค่าความคลาดเคลื่อนมาตรฐาน หลังจากนั้นเราจะทดสอบความแตกต่างระหว่างค่าเฉลี่ยของค่าสังเกต ซึ่งจะเริ่มจากค่ามากที่สุดกับค่าน้อยที่สุด และควรจะเปรียบเทียบด้วยพิสัยนัยสำคัญต่ำสุด (R_a) ถัดมาให้หาความแตกต่างระหว่างค่ามากสุดกับค่าน้อยสุดอันดับสอง โดยการคำนวณและเปรียบเทียบด้วยพิสัยนัยสำคัญต่ำสุด R_{a-1} จากนั้นให้ทำการเปรียบเทียบในลักษณะนี้อย่างต่อเนื่องจนกระทั่งค่าเฉลี่ยของทุกตัวถูกนำไปเปรียบเทียบกับค่าเฉลี่ยที่มีขนาดใหญ่สุด และในที่สุดก็ให้คำนวณค่าความแตกต่างระหว่างค่าเฉลี่ยที่ใหญ่สุดเป็นอันดับ 2 และค่าที่น้อยที่สุด แล้วเปรียบเทียบกับพิสัยนัยสำคัญต่ำสุด R_{a-1} กระบวนการดังกล่าวนี้จะถูกดำเนินการไปจนกระทั่งความแตกต่างของ a(a-1)/2 คู่ทั้งหมดที่เป็นไปได้ของค่าเฉลี่ยได้ถูกพิจารณา ถ้าความแตกต่างของค่าสังเกตมีค่ามากกว่าพิสัยนัยสำคัญน้อยสุดที่ตรงกันจะทำให้สามารถสรุปได้ว่า ค่าเฉลี่ยคู่ที่กำลังพิจารณาอยู่นั้นมีความแตกต่างกันอย่างมีนัยสำคัญ (Montgomery, 1997; ปารเมศ ชุติมา, 2545)

4. ตัวอย่างงานวิจัยที่เกี่ยวข้องและบทวิจารณ์งานวิจัย

มีงานวิจัยหลายงานที่ศึกษา GA, CPP และปัญหาที่คล้ายคลึงกับ CPP ไว้โดยพยายามแก้ปัญหาเพื่อหาค่าคำตอบที่ดีที่สุดยกตัวอย่างงานวิจัยที่เกี่ยวข้องดังนี้

<u>ตัวอย่างของงานวิจัยที่ทำการศึกษาเกี่ยวกับกระบวนการทางพันธุกรรม (Genetic operation) และการกำหนดพารามิเตอร์ของ GA</u>

Gehring and Bortfeldt (1997) ทำการศึกษาปัญหา Container loading โดยใช้ GA และได้กำหนดค่าพารามิเตอร์ ขนาดประชากร ความน่าจะเป็นในการครอสโอเวอร์ และความน่าจะเป็นในการมิวเตชั่น เป็นค่าคงที่เท่ากับ 50, 0.5 และ 0.5 ตามลำดับ และเปรียบเทียบหา GA operator (Crossover and Mutation) ว่าตัวไหนเหมาะกับปัญหาซึ่งใช้การครอสโอเวอร์ 3 แบบ คือ Uniform order-based crossover, Partially mapped crossover, Cycle crossover และมิวเตชั่น 2 แบบ คือ Scramble sublist mutation และ Mutation by inversion ผลการทดลองพบว่า Uniform order-based crossover และ Scramble sublist mutation ให้ผลเฉลยดีที่สุด อย่างไรก็ตามพารามิเตอร์ที่ใช้ในงานวิจัยนี้ไม่ได้ทำการทดสอบเพื่อหาช่วงของค่าพารามิเตอร์ที่เหมาะสมที่สุดสำหรับปัญหา ดังนั้นในการทดลองเปรียบเทียบเพื่อหา GA operator ที่เหมาะสมจึงไม่ได้นำพารามิเตอร์ของ GA มาศึกษาร่วมด้วยแต่กำหนดไว้เป็นค่าคงที่

Todd (1997) ได้ทดสอบประสิทธิภาพของการครอสโอเวอร์ 14 ชนิด และการมิวเตชั่นอีก 5 ชนิด กับปัญหาการเดินทางของเซลส์แมน ที่มีจำนวนจังหวัดเท่ากับ 10, 20, 50, 100 และ 200 โดยกำหนดให้พารามิเตอร์อื่นๆของ GA มีขนาดคงที่ อย่างไรก็ตามพารามิเตอร์อื่นๆของ GA ที่ Todd (1997) กำหนดให้เป็นค่าคงที่นั้นอาจจะมีผลสำคัญ (Significant) ต่อค่าคำตอบ จึงทำให้ผลสรุปที่ได้มานั้นอาจจะไม่เป็นที่น่าเชื่อถือ

Kim, Kim and Cho (1998) ศึกษาปัญหา Workload smoothing in assembly lines และทดสอบหาค่า %C และ %M โดยใช้ช่วงที่ทำการทดสอบคือ 0.05, 0.10, 0.15,..., 0.95 ซึ่งช่วงที่ให้ค่าที่ดีของ %C และ %M คือ 0.70-0.90 และ 0.10-0.20 ตามลำดับ อย่างไรก็ดีงานวิจัยนี้ไม่ได้ทดสอบหาจำนวนประชากรและจำนวนรุ่นที่เหมาะสม

จากงานวิจัยที่ได้กล่าวไว้ในข้างต้น พบว่างานวิจัยแต่ละงานได้มีการใช้ค่าพารามิเตอร์ของ GA และ GA operator ที่ต่างกันออกไป บางงานได้ทดสอบก่อนการนำมาใช้แต่บางงานก็ได้กำหนดหรือนำมาใช้เลยโดยไม่ได้ทำการทดสอบก่อน ซึ่งในวิทยนิพนธ์นี้จะได้ทำการทดสอบเพื่อหาค่าพารามิเตอร์ และ GA operator แบบต่างๆที่เหมาะสมกับปัญหา CPP ก่อนที่จะนำไปใช้ต่อไป

ตัวอย่างของงานวิจัยที่ทำการศึกษาเกี่ยวกับกระบวนการคัดสรร (Selection) ใน GA

Kim et al. (1998) ศึกษาปัญหา Workload smoothing in assembly lines โดยใช้ GA และใช้ Tournament selection ซึ่งกำหนด Tournament size = 2

Li, Ip and Wang (1998) ศึกษาปัญหา Earliness and tardiness production scheduling and planning โดยใช้ GA และใช้ Roulette wheel selection มาคัดเลือกโครโมโซม

Pongcharoen (2001) ใช้ GA มาแก้ปัญหาการจัดตารางการผลิตสินค้าขนาดใหญ่ โดยเลือกใช้ Roulette wheel selection มาใช้ในกระบวนการคัดสรร

Aytug, Khouja and Vergara (2003) ได้ทำการทบทวนงานวิจัยที่ใช้ GA ในการแก้ปัญหา Production and operations management และกล่าวไว้ว่า กระบวนการคัดสรรที่นักวิจัยนิยมใช้มากที่สุด คือ Roulette wheel และ Tournament

จากตัวอย่างของงานวิจัยที่นำเสนอในข้างต้น จะเห็นได้ว่างานวิจัยแต่ละงานก็จะเลือกใช้วิธีการคัดสรรที่แตกต่างกันออกไป อย่างไรก็ดียังไม่มีงานวิจัยใดที่ทำการทดสอบกระบวนการคัดสรรว่าวิธีการใดเป็นวิธีที่เหมาะสมที่สุดสำหรับปัญหานั้นๆ ดังนั้นในงานวิทยานิพนธ์นี้จะได้ทำการศึกษาต่อไปว่า การคัดสรรแบบใดเป็นวิธีการที่ให้ค่าคำตอบที่ดีที่สุดโดยทำการทดสอบกับปัญหา CPP ต่อไป

ตัวอย่างของงานวิจัยที่ทำการศึกษาเกี่ยวกับ Elitist selection หรือกลยุทธ์การรักษาเผ่าพันธุ์ชนชั้นดี (Elitist strategy)

Murata et al. (1996) ได้เสนอกลยุทธ์ในการปกป้องโครโมโซมตัวที่ดีที่สุด (Elite chromosomes) ให้สามารถอยู่รอดต่อไปได้ เรียกว่า กลยุทธ์การรักษาเผ่าพันธุ์ชนชั้นดี (Elitist strategy) โดยหลังจากที่โครโมโซมทั้งหมดผ่านขั้นตอนของการประเมินค่าความเหมาะสมแล้ว Elitist strategy จะเลือกเก็บโครโมโซมตัวที่ดีที่สุดเอาไว้เพื่อนำไปแทนในโครโมโซมรุ่นถัดไป โดยกลยุทธ์การรักษาเผ่าพันธุ์ชนชั้นดีของ Murata et al. (1996) นั้นจะสุ่มเลือกโครโมโซมจากประชากรในรุ่นปัจจุบัน (Current population) ขึ้นมา 1 ตัว แล้วจึงแทนโครโมโซมดังกล่าวนี้ด้วยโครโมโซมชนชั้นดีที่ได้เก็บเอาไว้ อย่างไรก็ดียังไม่มีการทดสอบว่าโครโมโซมที่ถูกสุ่มออกมามีค่าที่ดีน้อยกว่าโครโมโซมที่จะแทนเข้าไป ต่อมา Li et al. (1998) นำ GA มาแก้ปัญหา Earliness and tardiness production scheduling and planning และใช้ Elitist strategy คือ ให้โครโมโซมตัวที่ดีที่สุดได้ผ่านไปในรุ่นถัดไปแต่ Li et al. (1998) นำไปใช้แค่รุ่นแรกรุ่นเดียวเท่านั้น อย่างไรก็ดีจะเห็นได้ว่าวิธีการ Elitist นี้คล้ายกับของ Murata et al. (1996) โดยไม่ได้ทำการพัฒนาขึ้นมา ในเวลา

ต่อมา Kim et al. (1998) ได้นำ GA มาแก้ปัญหาสำหรับ Workload smoothing in assembly lines และใช้ Elitist strategy ซึ่งได้พัฒนาขึ้นมาจากของ Murata et al. (1996) คือ ให้โครโมโซมตัวที่ดีที่สุดได้ผ่านไปในรุ่นถัดไป โดยนำโครโมโซมพันธุ์ดีที่เก็บเอาไว้ไปแทนที่โครโมโซมตัวที่แย่ที่สุดในรุ่นนั้น และต่อมาวีณา พรหมเทศ (2548) ได้ปรับปรุงแนวคิดของ Murata et al. (1996) โดยเปลี่ยนจากการสุ่มเลือกโครโมโซมจากประชากรในรุ่นปัจจุบันมาเป็นการเรียงลำดับโครโมโซมแล้วจึงแทนที่โครโมโซมตัวที่แย่ที่สุดด้วยโครโมโซมพันธุ์ดีที่ได้เก็บเอาไว้ นอกจากนั้นยังได้ออกแบบให้สามารถเก็บโครโมโซมพันธุ์ดีได้หลายตัว โดยกำหนดเปอร์เซ็นต์ของการเก็บโครโมโซมพันธุ์ดีเทียบต่อขนาดของประชากร (Percentage of keeping elite chromosomes: %E) และเก็บรักษากลุ่มของโครโมโซมพันธุ์ดีนี้เอาไว้ในลิสต์ (Elitist list) และได้ทำการทดลองเพื่อเปรียบเทียบแนวคิดนี้กับวิธีการของ Murata et al. (1996) ซึ่งสรุปได้ว่า แนวคิดที่พัฒนาขึ้นนี้ช่วยทำให้ GA เจอค่าคำตอบที่ดีกว่า Elitst strategy แบบดั้งเดิมที่อาศัยการสุ่มเป็นหลัก อย่างไรก็ตามงานวิจัยดังกล่าวยังไม่ได้มีการศึกษาและทดสอบถึงความเป็นไปได้ว่าควรจะกำหนดจำนวนของโครโมโซมพันธุ์ดี (Elite chromosome) ที่จะเก็บนั้นไว้ที่เท่าใด ดังนั้นในงานวิทยานิพนธ์นี้จึงจะนำแนวคิดนี้ไปประยุกต์ใช้พร้อมทั้งศึกษาและทดสอบเพื่อกำหนดหาเปอร์เซ็นต์ของการเก็บโครโมโซมชนชั้นดีเทียบต่อขนาดของประชากร (Elitist probability: %E) ที่เหมาะสมที่สุดสำหรับ CPP ต่อไป

ตัวอย่างของงานวิจัยที่ทำการศึกษาเกี่ยวกับฮิวริสติก (Heuristic) ในการวางกล่อง

George and Robinson (1980) ได้เสนอฮิวริสติกในการวางกล่องลงในคอนเทนเนอร์ โดยวิธีการที่ใช้จะถูกแบ่งออกเป็น 2 ส่วน คือ เริ่มจากการกำหนดพื้นที่ว่างที่จะนำกล่องสินค้าเข้าไปใส่ โดยเทคนิคที่ใช้ในการกำหนดพื้นที่ว่างสำหรับการบรรจุกล่องลงในคอนเทนเนอร์จะอยู่บนพื้นฐานของกฎฮิวริสติก ส่งผลให้การบรรจุจะเป็นแบบชั้นต่อชั้น เมื่อกำหนดพื้นที่ว่างแล้วส่วนถัดไปคือการจัดวางซึ่งจะมีอยู่ 4 รูปแบบคือ การวางเพื่อให้ได้ขนาดบรรจุที่เล็กที่สุด, การวางโดยพิจารณาปริมาณของกล่องในแต่ละชนิด, การวางเพื่อให้ได้ความยาวในการเรียงที่มากที่สุด และการวางโดยพิจารณาว่ากล่องเป็นกล่องเปิดหรือกล่องปิด

Gehring et al. (1990) กล่าวถึงขั้นตอนในการจัดวางกล่องที่มีขนาดแตกต่างกันลงในคอนเทนเนอร์ที่มีขนาดที่แน่นอน ซึ่งประกอบจากกฎฮิวริสติก 5 ส่วนได้แก่ การลดอย่างเป็นเชิงเส้นของปริมาณสินค้า, การจัดเรียงตามแนวตั้ง, การกำหนดชั้นตามแนวลึก, การบรรจุตามพื้นที่ว่างที่เหลืออยู่ และการสร้างทางเลือกในการเก็บ

Ngoi, Tay and Chua (1994) ได้เสนอวิธีการของระบบแสดงพื้นที่ว่าง (Spatial representation system) โดยเป็นการใช้ Single three-dimensional matrix (Spatial matrix) ซึ่งเป็นเมทริกซ์ 2 มิติ ที่แสดงรายละเอียดของภาคตัดขวาง (Cross-section) ของบรรจุภัณฑ์และพื้นที่ โดยวิธีการนี้จะมองหาพื้นที่ที่สามารถจะนำบรรจุภัณฑ์วางได้ ซึ่งจะมองหาทุกความน่าจะเป็นของการวางบรรจุภัณฑ์ต่อไป โดยทำการเปรียบเทียบกับปริมาตรที่ว่างของคอนเทนเนอร์ และพื้นที่ที่เหลือเพื่อใช้ในการวางบนบรรจุภัณฑ์ที่วางไว้แล้ว โดยในการมองหาพื้นที่ที่จะวางจะต้องมีความสอดคล้องกับบรรจุภัณฑ์ที่ยังไม่ได้จัดวางด้วย เพื่อให้ได้การวางที่เหมาะสมที่สุดในพื้นที่ที่เหลือนั้น

Bischoff et al. (1995) ศึกษาปัญหาการจัดเรียงบรรจุภัณฑ์ลงบนแพลเล็ต (Pallet loading problem) เป้าหมายของการบรรจุคือ จัดเรียงบรรจุภัณฑ์ให้ใช้ประโยชน์ของปริมาตรสูงที่สุด โดยทราบค่าขนาดของบรรจุภัณฑ์ ขนาดของแพลเล็ต และขีดจำกัดความสูงของบรรจุภัณฑ์ที่สามารถวางได้ โดยใช้ฮิวริสติกอัลกอริทึมหาคำตอบมีหลักการดังนี้ (1) บรรจุภัณฑ์จะถูกจัดเรียงในแนวราบทีละชั้น (2) พื้นที่ขนาดเล็กที่สุดจะถูกใช้จัดวางบรรจุภัณฑ์ก่อนพื้นที่ขนาดใหญ่ (3) ใช้การจัดวางแบบ 2-Blocks และรูปแบบการวางที่ให้ค่าอรรถประโยชน์การใช้พื้นที่สูงสุดในแต่ละรอบของการคำนวณจะถูกเลือกเป็นผังการจัดวาง (4) ทำซ้ำในข้อที่ 2 จนบรรจุภัณฑ์ทั้งหมดถูกจัดวาง

Bischoff and Ratcliff (1995) ศึกษาปัญหาการจัดเรียงบรรจุภัณฑ์ลงบนแพลเล็ต (Pallet loading problem) โดยจำนวนแพลเล็ตที่ใช้มีมากกว่าหนึ่งแพลเล็ต เป้าหมายในการจัดเรียงคือ ใช้จำนวนแพลเล็ตน้อยที่สุด เมื่อทราบค่าขนาดของบรรจุภัณฑ์ ขนาดของแพลเล็ต และขีดจำกัดความสูงของบรรจุภัณฑ์ที่สามารถวางบนแพลเล็ต ในการวิจัยได้ประยุกต์หลักการจัดเรียงแบบ "Single-pallet algorithm" จากงานวิจัยของ Bischoff et al. (1995) ให้สามารถจัดเรียงบรรจุภัณฑ์บนแพลเล็ตจำนวนมากกว่าหนึ่งได้ นอกจากนี้ในงานวิจัยยังได้เสนออัลกอริทึมสำหรับการจัดเรียงบรรจุภัณฑ์ลงบนหลายแพลเล็ตโดยเฉพาะ เรียกว่า "Simultaneous approach"

Gehring and Bortfeldt (1997) ศึกษา GA สำหรับแก้ปัญหา Container loading ได้เสนอแนวคิดการแก้ปัญหา 2 ขั้นตอน ดังนี้ ขั้นตอนที่ 1 สร้างเซตของกล่องแบบหอคอย โดยการจัดกล่องให้อยู่ในรูปหอคอย (Tower) ภายใต้ข้อจำกัด 5 ประการดังนี้ (1) ในการจัดวาง รูปแบบการวางตัวของกล่องสามารถหมุนได้ 90 องศาในแนวราบ แต่ไม่สามารถหมุนได้ในแนวดิ่ง (2) กล่องบางประเภทถูกกำหนดให้วางในตำแหน่งบนสุดของหอคอยเท่านั้น ไม่อนุญาตให้กล่องอื่นวางซ้อนทับลงมา (3) น้ำหนักรวมของกล่องทั้งหมดต้องไม่เกินความสามารถในการรับน้ำหนักของคอนเทนเนอร์ (4) พื้นที่ฐานของกล่องต้องถูกรองรับโดยกล่องที่เป็นฐาน คิดเป็นร้อยละไม่ต่ำกว่าค่าที่กำหนดให้ (5) การวางกล่องต้องไม่ให้จุดศูนย์ถ่วงของกล่องตกอยู่นอกพื้นที่กล่องที่รองรับ

ขั้นตอนที่ 2 การจัดเซตของกล่องแบบหอคอยลงบนพื้นของคอนเทนเนอร์ ซึ่งในขั้นตอนนี้จัดเป็นปัญหาการจัดวางแบบ 2 มิติ โดยใช้ GA มาทำการจัด

Pimpawat and Chaiyaratana (2001) ศึกษาปัญหาการบรรจุกล่องลงในคอนเทนเนอร์โดยใช้ขั้นตอนวิธีเชิงพันธุกรรมแบบวิวัฒนาการและทำงานร่วมกัน (A co-operative co-evolutionary genetic algorithm: CCGA) ซึ่งมีข้อสมมติฐานคือ (1) กล่องทุกใบมีรูปทรงเป็นรูปสี่เหลี่ยมมุมฉาก และแบ่งกล่องทั้งหมดออกเป็น 3 กลุ่ม คือกล่องขนาดใหญ่ กลาง และเล็ก โดยกล่องกลุ่มเดียวกันจะมีขนาดใกล้เคียงกัน (2) คอนเทนเนอร์มีรูปทรงเป็นรูปสี่เหลี่ยมมุมฉาก และสามารถบรรจุได้ไม่เกินปริมาตรสูงสุดที่สามารถรับได้ (3) รูปแบบการบรรจุที่สนใจจะเป็นแบบ Guillotine cutting (4) การจัดเรียงกล่องแต่ละชั้นจะเริ่มจากด้านซ้ายล่างก่อน และมีการสมมติเส้นแบ่งขั้นในแนวนอน (Horizontal boundary hyper-plane) ระหว่างสองชั้นที่ติดกันของกล่องโดยความสูงของแต่ละชั้นจะถูกกำหนดจากความสูงของกล่องที่สูงที่สุดที่ถูกจัดเรียงเข้าไปในชั้นนั้น (5) กล่องที่อยู่ในชั้นเดียวกันจะถูกจัดเรียงเข้าสู่รูปแบบด้านหน้าซ้าย (6) กระบวนการบรรจุจะเริ่มต้นจากการบรรจุกล่องที่มีขนาดใหญ่ก่อนแล้วตามด้วยกล่องขนาดกลาง และเล็กตามลำดับ ซึ่งกระบวนการจัดเรียงที่สมบูรณ์จะเกิดจากการรวมลำดับการบรรจุของกล่องทั้ง 3 กลุ่มเข้าด้วยกัน

Chien and Deng (2004) สร้างระบบช่วยสนับสนุนการตัดสินใจในการกำหนดและจัดรูปแบบกล่องลงในคอนเทนเนอร์ โดยใช้ Computational algorithm และ Graphic user interface (GUI) ซึ่งมีวิธีการบรรจุ 6 ขั้นตอนดังนี้ ขั้นที่ 1 ป้อนข้อมูล ขนาดกว้าง ยาว สูง ของกล่อง ชนิดของกล่องและจำนวนของกล่องแต่ละชนิด โดยมีเกณฑ์การเลือกกล่องตามลำดับความสำคัญ (Prioritize) 5 ข้อดังนี้ (1) เลือกกล่องที่มีมิติฐานใหญ่กว่า (2) เลือกกล่องที่มีพื้นที่ฐานใหญ่กว่า (3) เลือกกล่องที่มีมิติ x ยาวกว่า (ด้านยาว) (4) เลือกกล่องที่มีมิติ y ยาวกว่า (ด้านกว้าง) และ (5) เลือกกล่องที่มีมิติ z ยาวกว่า (ด้านสูง) ขั้นที่ 2 เลือกกล่องที่จะบรรจุซึ่งขึ้นอยู่กับลำดับตำแหน่งการเลือกตามขั้นที่ 1 ขั้นที่ 3 รวบรวมพื้นที่ว่างในคอนเทนเนอร์แล้วตรวจสอบพื้นที่ว่างทั้งหมดกับกล่องที่เลือกถ้ามีพื้นที่ว่างที่เหมาะสมไปขั้นที่ 4 แต่ถ้าไม่มีพื้นที่ว่างที่เหมาะสมให้วนกลับไปขั้นที่ 2 ขั้นที่ 4 นำกล่องที่เลือกไปใส่ในพื้นที่ว่างที่เหมาะสม โดยมีเกณฑ์การเลือกพื้นที่ว่างตามลำดับความสำคัญ (Prioritize) 3 ข้อดังนี้ (1) เลือกพื้นที่โดยอ้างอิงจุดที่แกน x น้อยที่สุด (2) เลือกพื้นที่โดยอ้างอิงจุดที่แกน y น้อยที่สุด และ (3) เลือกพื้นที่โดยอ้างอิงจุดที่แกน z น้อยที่สุด ขั้นที่ 5 บรรจุกล่องลงในพื้นที่ว่างที่เหมาะสมตามตำแหน่งที่เลือกในขั้นที่ 4 และอัพเดท (Update) ข้อมูลโดยลบกล่องที่ถูกบรรจุแล้วออกจากบัญชีของกล่องที่ยังไม่ได้บรรจุ ขั้นที่ 6 หยุดขั้นตอนการทำงานเมื่อพื้นที่ว่างทั้งหมดเล็กกว่ากล่องที่ยังไม่ได้บรรจุ หรือกล่องทั้งหมดถูกบรรจุเรียบร้อยแล้ว

จากตัวอย่างงานวิจัยที่นำเสนอไปข้างต้น จะพบว่าแต่ละงานวิจัยก็มีวิธีการฮิวริสติกที่เหมาะสมกับอัลกอริทึมที่เลือกใช้และเงื่อนไขของปัญหานั้นๆ ซึ่งในวิทยานิพนธ์นี้ได้สร้างฮิวริสติกขึ้นมาเพื่อให้เหมาะสมกับอัลกอริทึม และเรียกฮิวริสติกนี้ว่า GA smart arrange เนื่องจากต้องการประยุกต์ให้เข้ากับการทำงานของ GA และรูปแบบของโครโมโซมที่ได้ออกแบบไว้ หากนำฮิวริสติกที่ใช้วิธีการเลือกกล่องที่เหมาะสมกับพื้นที่ หรือเลือกพื้นที่ที่เหมาะสมสำหรับแต่ละกล่อง ลำดับการวางกล่องที่ใช้ GA หามาก็จะไม่มีความหมาย

ตัวอย่างข้อมูลกล่องที่นำมาใช้ในการทดลองของแต่ละงานวิจัย

George and Robinson (1980) ได้เสนอวิธีการหาคำตอบแบบประมาณในการแก้ปัญหาการจัดวางบรรจุภัณฑ์ลงในคอนเทนเนอร์ โดยปัญหาที่พิจารณาประกอบขึ้นจากกล่อง 20 ชนิด และจำนวนกล่องทั้งหมดที่ใช้คือ 800 กล่อง เงื่อนไขบังคับที่พิจารณาในปัญหานี้คือ กล่องสามารถวางทับกันได้บนด้านทุกด้านที่อยู่บนสุด อย่างไรก็ตามกล่อง 800 กล่องที่ใช้ในงานวิจัยนี้ได้เลือกมาจากกล่อง 20 ชนิด ดังนั้นจึงมีกล่องที่มีขนาดเท่ากันอยู่ซึ่งเป็นการง่ายที่จะนำกล่องที่มีขนาดเท่ากันมาวางชิดกันเพื่อลดการเกิดช่องว่างที่ไร้ประโยชน์ในการจัดเรียง

Jakobs (1996) ได้พัฒนา GA เพื่อมาใช้แก้ปัญหาการบรรจุ โดยนำไปใช้ร่วมกับกระบวนการที่ใช้ในการจัดเรียง เพื่อให้ได้ค่าคำตอบเริ่มต้นที่เรียกว่าระบบล่างซ้าย (Bottom left algorithm หรือ BL-algorithm) แต่มีข้อกำหนดให้ทุกๆกล่องมีความสูงเท่ากัน จะมีความต่างเพียงด้านยาวและด้านกว้างเท่านั้น อย่างไรก็ตามเงื่อนไขที่กำหนดขึ้นมาให้กับกล่องนั้นทำให้ไม่สามารถนำไปใช้ได้อย่างกว้างขวางเท่าใดนัก

Scheithauer and Sommerweiß (1998) ศึกษาปัญหาการจัดเรียงบรรจุภัณฑ์บนแพลเล็ตโดยเสนอแนวความคิดที่เรียกว่า GG4 และ G4 ในการจัดการพื้นที่บนแพลเล็ต โดยทำการแบ่งพื้นที่บนแพลเล็ตออกเป็นสี่เหลี่ยม 4 อัน ซึ่งในสี่เหลี่ยมแต่ละอันจะเป็นการวางของบรรจุภัณฑ์ชนิดเดียวกันที่สามารถบรรจุให้อยู่ในรูปของสี่เหลี่ยมที่สามารถวางบนแพลเล็ตได้ ดังนั้นจึงมีการจัดเรียงทั้งสิ้น 4 กลุ่ม และทำการเพิ่มชั้นขึ้นไปเรื่อยๆเพื่อใช้ในปัญหาแบบ 3 มิติ แต่บรรจุภัณฑ์ในแต่ละชั้นจะต้องมีความสูงที่เท่ากัน โดยกำหนดให้ G4 ต้องมีขนาดของบรรจุภัณฑ์เท่ากันทุกบรรจุภัณฑ์ และขนาดของบรรจุภัณฑ์สำหรับ GG4 ไม่จำเป็นต้องมีขนาดที่เท่ากัน

Miyazawa and Wakabayashi (2000) ได้ศึกษาปัญหาการบรรจุกล่องรูปทรงสี่เหลี่ยมลงในภาชนะบรรจุรูปทรงสี่เหลี่ยมในระบบ 3 มิติ โดยเสนอปัญหาการบรรจุ 3 มิติแบบออร์โธโกนัลแซดโอเรียนท์ (Orthogonal z-oriented three-dimensional packing problem: TPP^Z) ซึ่งได้แบ่ง

ปัญหาออกเป็นสองกลุ่ม กลุ่มแรกทำการแก้ปัญหาโดยวัตถุสามารถหมุนได้ 90 องศารอบแกน Z และกลุ่มที่สองแก้ปัญหาโดยวัตถุไม่สามารถหมุนได้ อย่างไรก็ตามวัตถุที่ใช้ทดสอบในงานวิจัยนี้ก็ไม่สามารถหมุนได้ทุกรูปแบบ

Pimpawat and Chaiyaratana (2001) ศึกษาปัญหาการบรรจุกล่องลงในคอนเทนเนอร์ และทดสอบโดยใช้ modified CCGA และ standard GA สรุปว่า modified CCGA มีประสิทธิภาพสูงกว่า standard GA เนื่องมาจาก modified CCGA ใช้กลไกการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบวิวัฒนาการและทำงานร่วมกันแบบปรับปรุง และใช้กฎฮิวริสติกในการแบ่งลำดับการบรรจุทั้งหมดออกเป็นลำดับการบรรจุย่อยสามลำดับ ซึ่งปัญหาที่พิจารณาประกอบขึ้นจากกล่อง 12 ชนิด และจำนวนกล่องทั้งหมดที่ใช้คือ 205 กล่อง โดยแบ่งกล่องทั้งหมดออกเป็น 3 กลุ่ม คือกล่องขนาดใหญ่เลือกมาจากกล่อง 4 ชนิด จำนวน 75 กล่อง กล่องขนาดกลางเลือกมาจากกล่อง 4 ชนิด จำนวน 65 กล่อง และกล่องขนาดเล็กเลือกมาจากกล่อง 4 ชนิด จำนวน 65 กล่อง อย่างไรก็ตามจำนวนกล่องที่นำมาทดลองนั้นมีจำนวนน้อย และกล่อง 205 กล่องที่ใช้ในงานวิจัยนี้ได้เลือกมาจากกล่อง 12 ชนิด ดังนั้นจึงมีกล่องที่มีขนาดเท่ากันอยู่จำนวนหนึ่งซึ่งเป็นการง่ายที่จะนำกล่องที่มีขนาดเท่ากันมาวางชิดกันเพื่อลดการเกิดช่องว่างที่ไร้ประโยชน์ในการจัดเรียง

Chien and Deng (2004) สร้างระบบช่วยสนับสนุนการตัดสินใจในการกำหนดและจัดรูปแบบของกล่องลงในคอนเทนเนอร์ ในส่วนของการทดลองได้เปรียบเทียบวิธีการที่เสนอ (Computational algorithm) กับ Greedy algorithm และการจัดเรียงของบริษัทที่นำข้อมูลมา โดยทดลองกับคอนเทนเนอร์ 11 แบบ และกล่อง 49 แบบที่ต่างกัน ซึ่งใช้จำนวนกล่องแต่ละแบบเป็นจำนวนที่กำหนดขึ้นรวมแล้ว 717 กล่อง ผลการทดลองปรากฏว่า Computational algorithm มีอัตราการใช้ประโยชน์ของพื้นที่ มากกว่าหรือเท่ากับ Greedy algorithm และดีกว่าคำตอบของบริษัท แต่ทางด้านเวลา Greedy algorithm ใช้เวลาเร็วกว่า อย่างไรก็ตามกล่อง 717 กล่องที่ใช้ในงานวิจัยนี้ได้เลือกมาจากกล่อง 49 แบบ ดังนั้นจึงมีกล่องที่มีขนาดเท่ากันอยู่ซึ่งเป็นการง่ายที่จะนำกล่องที่มีขนาดเท่ากันมาวางชิดกันเพื่อไม่ให้เกิดพื้นที่ว่างที่ไร้ประโยชน์ในการจัดเรียง

Boschetti (2004) ศึกษา Three-dimensional finite bin packing problem โดยใช้ New lower bounds ซึ่งได้กำหนดเงื่อนไขให้กล่องไม่สามารถหมุนได้ และหมุนได้เพียง 90 องศา อย่างไรก็ตามกล่องไม่สามารถหมุนได้อย่างอิสระ ดังนั้นอาจทำให้ประสิทธิภาพในการหาคำตอบลดลง และทำให้ไม่สามารถนำไปใช้ได้อย่างกว้างขวางเท่าใดนัก