

การแก้ปัญหาการจัดผังโรงงานด้วยเจนเนติกอัลกอริทึมร่วมกับอิมพัรฟ์เมนต์ฮีริสติก

Genetic Algorithms with Improvement Heuristic For Plant Layout Problems

ปารเมศ ชูติมา และ ชนะ เขียงกมลสิงห์

ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

โทรศัพท์ 218-6847 โทรสาร 251-3969

E-Mail:CPARAMES@Chula.ac.th

บทคัดย่อ

งานวิจัยฉบับนี้มีวัตถุประสงค์เพื่อเสนอวิธีการประยุกต์ใช้เจนเนติกอัลกอริทึมร่วมกับอิมพัรฟ์เมนต์ฮีริสติก (Improvement Heuristic) แบบ SDPI ในการแก้ปัญหาการจัดผังโรงงาน เนื่องจากการหาคำตอบของปัญหาการจัดผังโรงงานด้วยวิธีการทางฮิวริสติกแต่เพียงอย่างเดียวอาจได้คำตอบที่ไม่ค่อยดีและการหาคำตอบของปัญหาการจัดผังโรงงานด้วยเจนเนติกอัลกอริทึมแต่เพียงอย่างเดียวถึงแม้จะได้คำตอบที่ดีแต่โดยมากแล้วจะต้องใช้ระยะเวลานาน เพื่อแก้ปัญหาดังกล่าวจึงได้นำวิธีการทั้งสองมาใช้ร่วมกันในการหาคำตอบ ซึ่งทำได้โดยค้นหาคำตอบเริ่มต้นด้วยวิธีการฮิวริสติกแล้วป้อนให้แก่อัลกอริทึม ซึ่งผลการทดลองแสดงให้เห็นว่าการจัดผังโรงงานด้วยวิธีการดังกล่าว ให้คำตอบที่ดีกว่าฮิวริสติกแต่เพียงอย่างเดียว และสามารถลดเวลาในการหาคำตอบได้เมื่อเปรียบเทียบกับการใช้เจนเนติกอัลกอริทึมแต่เพียงอย่างเดียว

Abstract

The objective of this research is to propose an approach to apply Genetic Algorithms (GAs) incorporation with an improvement heuristic, namely SDPI, in solving plant layout problems. It is obvious that the solution obtained from a heuristic may not be the good solution. In addition, although typical GAs can provide a good solution to plant layout problems, it may take a long time before the final solution is reached. As a result, to avoid these problems, this research attempts to combine the strength of GAs and heuristic, i.e., obtain a good solution with less time-consuming. The result indicates that the proposed approach is quite promising comparing to typical GAs and heuristic.

1. บทนำ

การจัดผังโรงงาน เป็นปัญหาที่เกี่ยวกับการจัดเรียงเครื่องจักรหรือบล็อกแพลน (Block Plan) จำนวน n เครื่องลงใน m ตำแหน่ง (โดยที่

$n \leq m$) วิธีการแก้ปัญหาการจัดผังโรงงานแบ่งออกเป็นสองวิธีใหญ่ๆ ตามลักษณะของข้อมูลที่ใช้ คือ การจัดผังโรงงานโดยใช้ข้อมูลเชิงปริมาณ และการจัดผังโรงงานโดยใช้ข้อมูลเชิงคุณภาพ ข้อมูลเชิงปริมาณหมายถึงข้อมูลที่เกี่ยวข้องกับปริมาณการไหลและค่าใช้จ่ายการไหลของวัสดุระหว่างสถานีงาน ส่วนข้อมูลเชิงคุณภาพหมายถึงข้อมูลที่ใช้แสดงความสัมพันธ์ระหว่างสถานีงานหนึ่งกับสถานีอื่นๆ ตามความต้องการของผู้ออกแบบผังโรงงาน วัตถุประสงค์ในการจัดผังโรงงานคือ เพื่อให้เกิดค่าใช้จ่ายน้อยที่สุดหรือมีความสัมพันธ์ตามความต้องการของผู้ออกแบบผังโรงงานสูงสุด งานวิจัยฉบับนี้เน้นถึงการวางผังโรงงานโดยใช้ข้อมูลเชิงปริมาณ

ค่าใช้จ่ายโดยรวม (Total Cost: TC) ของการจัดผังโรงงานสามารถแสดงในรูปสมการทางคณิตศาสตร์ซึ่งแสดงให้เห็นถึงความสัมพันธ์ของปริมาณการไหลของวัสดุ ค่าใช้จ่ายในการไหลของวัสดุระหว่างแผนก และระยะทางระหว่างแผนก เป้าหมายของการจัดผังโรงงานคือ เพื่อให้ค่าใช้จ่ายโดยรวมต่ำที่สุด ดังสมการต่อไปนี้ (Kusiak [1])

$$\text{Min } TC = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^{n-1} \sum_{l=k+1}^n f_{ik} \cdot c_{jl} \cdot x_{ij} \cdot x_{kl} \quad (1)$$

$$\begin{aligned} \text{Subject to } \sum_{j=1}^n x_{ij} &= 1, \forall i \\ \sum_{i=1}^n x_{ij} &= 1, \forall j \\ x_{ij} &= \{0,1\} \end{aligned}$$

C คือค่าใช้จ่ายโดยรวมที่เกิดจากการจัดวางผังโรงงาน

n คือจำนวนแผนกหรือตำแหน่งทั้งหมด

f_{ik} คือการไหลวัสดุจากแผนก i ไปแผนก k

$$x_{ij} = \begin{cases} 1 & \text{ถ้าแผนก } i \text{ ตั้งอยู่ตำแหน่ง } j \\ 0 & \text{ถ้าไม่เป็นเช่นนั้น} \end{cases}$$

c_{jl} คือต้นทุนการไหลวัสดุต่อหน่วยจากตำแหน่ง j ไปยังตำแหน่ง l ที่เกิดจากค่าใช้จ่ายของการไหลจากแผนก j ไปยังแผนก l

กับระยะทาง

การวัดระยะทางโดยทั่วไปแล้วมีอยู่สองแบบคือ การวัดแบบเรกติเนีย (Rectilinear) และการวัดแบบยูคลิดีเนียน (Euclidean) (Askin และ Strandridge [2])

ระยะทางแบบเรกติเนีย คือระยะทางระหว่างสถานที่ที่เป็นไปตามแนวแกน x และแกน y ดังแสดงในสมการ (2)

$$d_r = \Delta x + \Delta y \quad (2)$$

โดยที่ Δx และ Δy คือความแตกต่างของระยะทางตามแนวแกน x และ y

ระยะทางแบบยูคลิดีเนียน คือระยะทางที่สั้นที่สุดระหว่างสถานที่ดังแสดงในสมการ (3)

$$d_e = \sqrt{\Delta x^2 + \Delta y^2} \quad (3)$$

รูปแบบของปัญหาตามสมการที่ 1 เรียกว่า Quadratic Assignment Problems (QAP) ซึ่งเป็น Quadratic Integer Programming (QIP) และเป็นปัญหาที่เป็นที่รู้จักกันดีทางด้าน Optimization การหาคำตอบ QAP ทำได้หลายวิธี [2] เช่นวิธีแตกกิ่งและกำหนดขอบเขต (Branch and Bound) หรือกำหนดการพลวัต (Dynamic Programming) ซึ่งทำให้เสียเวลาในการหาคำตอบเป็นฟังก์ชันเอ็กซโปเนนเชียลเมื่อขนาดของปัญหาเพิ่มขึ้น (French [3]) แนวทางในการแก้ปัญหาประเภทนี้เพื่อให้ได้คำตอบที่ดีแม้จะไม่เป็นคำตอบที่ดีที่สุดคือ การใช้ฮิวริสติก (Heuristic) ฮิวริสติกที่ใช้ในการแก้ปัญหาประเภทนี้มีอยู่สองวิธีใหญ่ๆ คือ คอนสตรักชันฮิวริสติกและอิมพรูฟเมนต์ฮิวริสติก (Francis et al. [4])

คอนสตรักชันฮิวริสติกเป็นวิธีการจัดผังโรงงานโดยเริ่มจากการวางผังโรงงานไปทีละแผนก แล้วทำการคำนวณหาค่าใช้จ่ายที่เกิดขึ้น จากนั้นย้ายตำแหน่งการวางแล้วคำนวณหาค่าใช้จ่ายใหม่ ทำเช่นนั้นจนครบทุกตำแหน่ง แล้วเลือกตำแหน่งการวางที่ทำให้เกิดค่าใช้จ่ายน้อยที่สุด จากนั้นก็เพิ่มแผนกที่เหลือทีละแผนก แล้วทำการคำนวณใหม่อีกครั้ง ทำเช่นนั้นจนครบทุกแผนกจนได้คำตอบ

อิมพรูฟเมนต์ฮิวริสติกเป็นการจัดผังโรงงานที่จำเป็นต้องมีโครงสร้างของผังเริ่มต้นก่อน แล้วทำการปรับปรุงผังโรงงานนั้นจนได้ผังโรงงานใหม่ที่มีคำตอบดีกว่าผังโรงงานเดิม อิมพรูฟเมนต์ฮิวริสติกมีอยู่หลายวิธีเช่น คราฟท์ฮิวริสติก (CRAFT) (Armour และ Buffa [5]) หรือ Steepest Descent Pairwise Interchange Heuristic (SDPI) (Nugent et al. [6])

SDPI เป็นวิธีการหาคำตอบโดยอาศัยการสลับคู่ในโครงสร้างของผังโรงงานทีละคู่ และเลือกคู่ที่ดีที่สุด หรือคู่ที่สลับแล้วทำให้เกิดค่าใช้จ่ายที่น้อยที่สุด ทำการเก็บตำแหน่งทั้งสองไว้ จากนั้นทำการสลับคู่ที่เหลือตามขั้นตอนเดิมจนได้คำตอบครบทุกตำแหน่ง

ข้อดีของฮิวริสติกคือ ง่ายต่อการใช้งาน ไม่จำเป็นต้องมีกำหนดพารามิเตอร์ต่างๆ และมีความซับซ้อนน้อย แต่มีข้อเสียก็คือ

คำตอบที่ได้อาจไม่เป็นคำตอบที่ดีที่สุด ในบางครั้งคำตอบจะติดอยู่ใน Local optima

เจนเนติกอัลกอริทึม (GAs) (Holland [7]) เป็นวิธีการหาคำตอบอย่างหนึ่งโดยมีพื้นฐานมาจากกระบวนการคัดเลือกทางธรรมชาติในการคัดเลือกสตรีงที่มีความเหมาะสมในการอยู่รอดกับสตรีงอื่นๆ โดยมีการแลกเปลี่ยนข่าวสารอย่างสุ่ม คำตอบที่ดีเปรียบได้กับสตรีงผังโรงงานที่มีค่าใช้จ่ายที่น้อยที่สุด

มีงานวิจัยจำนวนมากที่ได้นำ GAs ไปใช้ในการทำ Optimization กับปัญหาประเภทต่างๆ เช่น ปัญหา QAP ปัญหาการจัดแผนการขนส่ง (Transportation) ปัญหาการจัดตารางการผลิต (Scheduling) (Michalewicz [8])

ตัวอย่างงานวิจัยที่นำ GAs มาประยุกต์ใช้กับปัญหาการจัดผังโรงงาน ได้แก่ Chan และ Tansri [9] ได้นำเจนเนติกอัลกอริทึมมาใช้ในการแก้ปัญหาการจัดผังโรงงาน และได้ทำการเปรียบเทียบคำตอบที่ได้จากการครอสโอเวอร์แบบ PMX (Partially Matched Crossover), OX (Order Crossover) และ CX (Cycle Crossover) ในการแก้ปัญหาการจัดเรียงผังโรงงาน โดยมีข้อกำหนดให้มีสถานีใดสถานีหนึ่งหรือหลายๆ สถานีคงที่ และการจัดผังโรงงานเป็นแบบมีพื้นที่มากกว่าจำนวนสถานี Tate และ Smith [10] ได้นำเอา GAs มาใช้ในการแก้ปัญหา QAP และเปรียบเทียบคำตอบที่ได้รับระหว่างฮิวริสติกและ GAs พบว่าคำตอบที่ได้จาก GAs เป็นคำตอบที่ดีกว่า Suresh et al. [11] ได้นำ GAs มาใช้ในการแก้ปัญหาการจัดผังโรงงานร่วมกับการครอสโอเวอร์ และได้พิจารณาถึงอัตราส่วนของวิธีการจัดเรียงสถานที่ที่เป็นไปได้และเป็นไปไม่ได้ในการพิจารณาถึงความเป็นไปได้ของคำตอบ

ข้อดีของ GAs คือ สามารถหาคำตอบที่ดีหรือใกล้เคียงกับค่าที่ดีที่สุดมากกว่าฮิวริสติก แต่มีข้อเสียคือ อาจใช้เวลาในการหาคำตอบนานและมีความจำเป็นที่จะต้องกำหนดค่าพารามิเตอร์ต่างๆ ของ GAs ให้เหมาะสม

เนื่องจากทั้งวิธีการของเจนเนติกอัลกอริทึมและฮิวริสติกต่างก็มีข้อดีและข้อเสีย งานวิจัยฉบับนี้จึงได้รวมข้อดีของวิธีการทั้งสองโดยมีวัตถุประสงค์เพื่อให้คำตอบที่ได้เป็นคำตอบที่ดีที่สุดหรือใกล้เคียง และสามารถลดระยะเวลาในการหาคำตอบลงได้ การรวมวิธีการทั้งสองเข้าด้วยกันทำได้โดยการให้คำตอบเริ่มต้นที่ได้ฮิวริสติกแก่ GAs แทนการสุ่มคำตอบเริ่มต้น เพื่อให้ได้คำตอบได้ดีที่สุดหรือใกล้เคียง และสามารถลดเวลาในการหาคำตอบของ GAs ลงได้

1.1 ข้อมูลเชิงปริมาณของผังโรงงาน

ข้อมูลของผังโรงงานตัวอย่างแบ่งออกเป็นสองส่วนคือ ข้อมูลเกี่ยวกับปริมาณการไหลและข้อมูลเกี่ยวกับค่าใช้จ่ายที่เกิดจากการไหล โดยข้อมูลทั้งสองสามารถแสดงได้ด้วยแผนภูมิจาก-ไป (From-To chart) ตัวอย่างของงานวิจัยนี้เลือกปัญหาจาก Francis [4] ซึ่งเป็นปัญหาการจัดผังโรงงานขนาด 10 สถานี ข้อมูลได้แสดงไว้ในแผนภูมิ รูปที่ 1 และ 2 ตามลำดับ

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

รูปที่ 1 แผนภูมิจาก-ไปของปริมาณการไหล

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

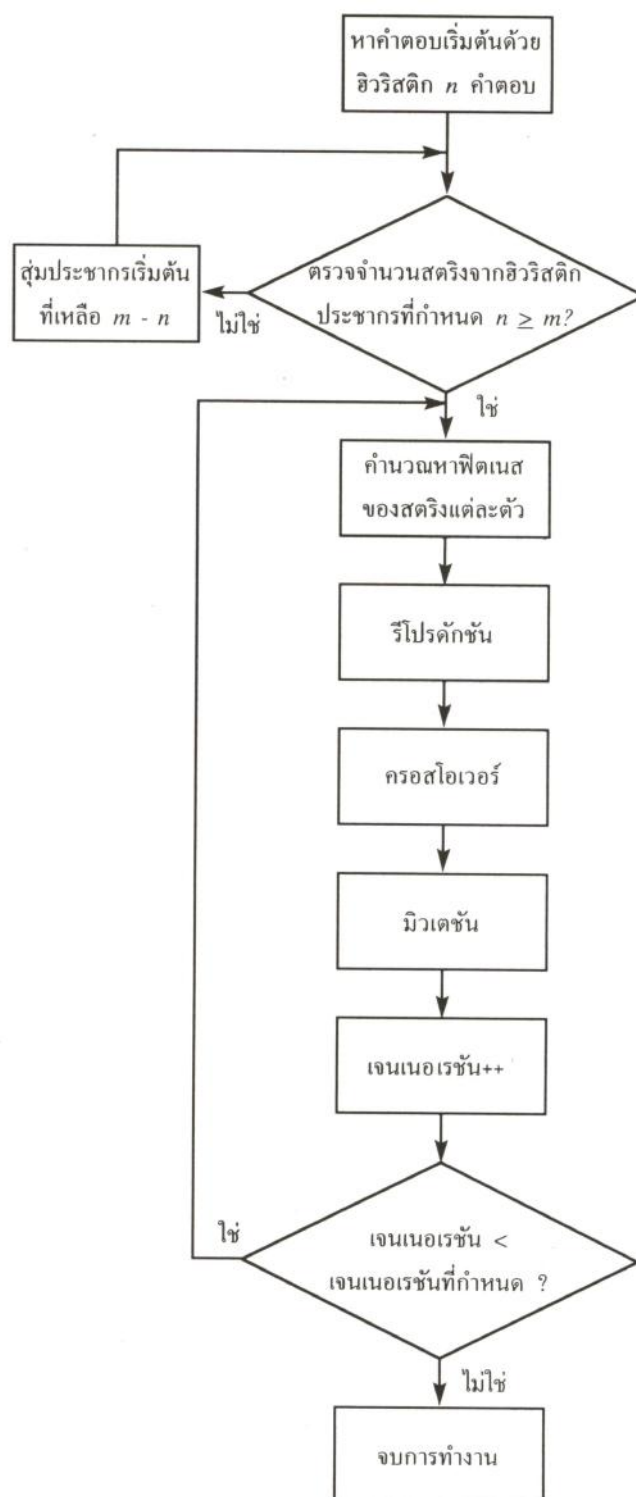
รูปที่ 2 ค่าใช้จ่ายของการไหล

1.2 เงินเนติกอัลกอริทึม

ทฤษฎีพื้นฐานเกี่ยวกับเงินเนติกอัลกอริทึมที่ใช้ในงานวิจัยนี้มีรายละเอียดดังต่อไปนี้ (รูปที่ 3)

ขั้นตอนแรกเป็นการหาสตริงคำตอบเริ่มต้นจาก SDPI จำนวน n คำตอบ แล้วนำสตริงคำตอบที่ได้มาเป็นคำตอบเริ่มต้นให้กับ GAS โดยจะต้องทำการตรวจสอบไม่ให้อาณาของสตริงคำตอบเริ่มต้นมากไปกว่าจำนวนประชากรที่กำหนดหรือประชากรในเมตติงพูล (Mating Pool) ซึ่งมีจำนวน m สตริง ถ้าน้อยกว่านั้น ให้ทำการสุ่มสตริงคำตอบเริ่มต้นที่เหลือเป็นจำนวน $m-n$ จนครบจำนวนประชากรที่กำหนด

เมื่อได้สตริงคำตอบเริ่มต้นครบทั้งหมดแล้ว จะนำสตริงทั้งหมดมาคำนวณหาค่าใช้จ่ายเพื่อที่จะนำสตริงเหล่านี้ไปเข้ากระบวนการ



รูปที่ 3 ขั้นตอนการทำงานของ GAS

การรีโพรดัคชัน ครอสโอเวอร์ และมิวเตชันต่อไป กระบวนการสุดท้ายเป็นการตรวจสอบสถานะสิ้นสุดการทำงาน โดยตรวจสอบจากจำนวนเงินเนอเรนสูงสุดที่กำหนดไว้ ถ้าเงินเนอเรนมากกว่าหรือเท่ากับเงินเนอเรนสูงสุดที่กำหนดไว้ให้หยุดทำงาน แต่ถ้าเงินเนอเรนน้อยกว่าเงินเนอเรนสูงสุดที่กำหนดไว้ ให้ทำการกระบวนการต่าง ๆ ซ้ำ

พารามิเตอร์ของ GAS ในงานวิจัยนี้มีจำนวนทั้งสิ้น 5 ค่า ได้แก่จำนวนประชากรที่กำหนด (Population) จำนวนสตริงที่เป็นค่า

ตอบเริ่มต้นจากอิวิสติก (Initial String Solution) จำนวนเงินเนอ-
เรชันค่าความน่าจะเป็นของการมีเวชัน (p_m) ค่าความน่าจะเป็นของการ
ครอสโอเวอร์ ซึ่งแบ่งเป็น 3 แบบ ได้แก่ ความน่าจะเป็นของการ
ครอสโอเวอร์แบบ PMX (pC_{pmx}) แบบ OX (pC_{ox}) และ แบบ
CX (pC_{cx})

1.3 การเข้ารหัสสตริง

การกำหนดหาระยะทางเพื่อคำนวณค่าใช้จ่ายโดยรวมจำเป็นต้องอาศัยการเข้ารหัสสตริง วิธีการเข้ารหัสสตริงกับผังโรงงานมีวิธีการ
แสดงดังตัวอย่างต่อไปนี้ สมมติให้ตำแหน่งของผังโรงงานมีขนาด
กว้าง x ยาว เป็น 2 x 5 หน่วยระยะทางหรือ 10 แผนก โดยที่ตำแหน่ง
เริ่มต้นจาก 1 จนถึง 10 และมีการวัดระยะทางแบบยูคลิดีเนียน ดังรูปที่ 4

การเข้ารหัสสตริงของผังโรงงานขนาด 10 สถานี โดยกำหนด
ให้สตริงเริ่มต้นเป็น [3 4 7 6 1 10 8 9 5 2] ค่าแต่ละค่าในสตริง
หรือยีน (Gene) คือสถานีหนึ่งๆ นั่นเอง การเข้ารหัสสตริงจะเริ่มจาก
ตำแหน่งแรกของสตริงนำไปวางไว้ในตำแหน่งที่ 1 ของผังโรงงาน และ
แผนกต่อๆ ไปก็จะนำมาวางเรียงกันต่อๆ ไปตามลำดับจากซ้ายไปขวา
และบนลงล่าง ซึ่งสามารถจัดผังโรงงานได้ดังรูปที่ 5

1	2	3	4	5
6	7	8	9	10

รูปที่ 4 ตำแหน่งของผังโรงงาน

3	4	7	6	1
10	8	9	5	2

รูปที่ 5 การจัดวางสถานีที่ 1-10

1.4 กระบวนการรีโพรดักชันและฟิตเนสฟังก์ชัน

ฟิตเนสฟังก์ชัน (Fitness function) เป็นฟังก์ชันที่ใช้กำหนด
ว่า สตริงคำตอบของการจัดผังโรงงานที่สร้างขึ้นควรที่จะถูกคัดเลือก
ในประชากรรุ่นถัดไปหรือไม่ สตริงคำตอบที่มีค่าใช้จ่ายที่น้อยกว่าควรมี
โอกาสที่จะถูกคัดเลือกได้มากกว่าสตริงคำตอบที่มีค่าใช้จ่ายที่มากกว่า
ดังสมการที่ 1

กระบวนการรีโพรดักชัน (Reproduction) คือกระบวนการ
ในการคัดเลือกสตริงไปยังเงินเนอเรชันถัดไป โดยที่สตริงที่มีค่าความ
เหมาะสมสูงหรือสตริงที่ให้ค่าใช้จ่ายต่ำกว่ามีโอกาสที่จะถูกคัดเลือกไป
ยังเงินเนอเรชันต่อไปได้มากกว่าสตริงที่ให้ค่าใช้จ่ายสูงกว่า กระบวนการ
นี้เกิดขึ้นจากกระบวนการคัดเลือกตามธรรมชาติ ตามทฤษฎี Survival
of Fittest ของ ชาร์ล ดาร์วิน วิธีการที่เลือกใช้คือวิธีการของวงล้อ
รูเล็ต (Roulette Wheel) (Goldberg [12])

ตัวอย่างของกระบวนการรีโพรดักชันแบบวงล้อรูเล็ต โดย
สมมติให้มีจำนวนประชากรเป็น 4 สตริง มีรายละเอียดดังตารางที่ 1

ให้คอลัมน์แรกของตารางที่ 1 เป็นสตริงที่สุ่มขึ้นซึ่งสามารถ
คำนวณฟิตเนสฟังก์ชันหรือค่าใช้จ่ายได้ตามที่ปรากฏในคอลัมน์ที่ 2
สตริงที่มีค่าใช้จ่ายที่น้อยที่สุดควรมีโอกาสที่จะอยู่รอดได้มากที่สุดจึง
สร้างคอลัมน์ที่ 3 และ 4 เพื่อใช้ในการคำนวณ ความน่าจะเป็นที่สตริง
จะถูกคัดเลือกไปยังเงินเนอเรชันต่อไป จะเห็นได้ว่าสตริงตัวแรกมีความ
น่าจะเป็นในการอยู่รอด 27.4% ส่วนสตริงตัวสุดท้ายมีความน่าจะเป็น
ในการอยู่รอด 22.8% เราสามารถคำนวณหาจำนวนครั้งของสตริงที่จะ
ถูกเลือกโดยเฉลี่ยได้ดังคอลัมน์ที่ 5 จำนวนได้จาก F_i/F_i และคอลัมน์
ที่ 6 เป็นจำนวนครั้งของโอกาสที่สตริงถูกคัดเลือกจริงที่ได้จากการสุ่ม
จะเห็นได้ว่า สตริงตัวแรกจากสตริงเริ่มต้นถูกคัดเลือกไปเป็นสตริงใหม่
ของเงินเนอเรชันต่อไปสองครั้ง สตริงตัวที่สองและสามจากสตริงเริ่มต้น
ถูกคัดเลือกไปเป็นสตริงใหม่ในเงินเนอเรชันต่อไปหนึ่งครั้ง และสตริงตัว
สุดท้ายไม่ได้ถูกคัดเลือกไปยังเงินเนอเรชันต่อไปเลย เมื่อพิจารณาสตริง
ที่ผ่านการคัดเลือกแล้วในคอลัมน์สุดท้ายจะเห็นได้ว่า ผลรวมและค่า

ตารางที่ 1 แสดงถึงวิธีการทำรีโพรดักชัน

สตริงเริ่มต้น (1)	ค่าใช้จ่ายโดย รวม (2)	$\sum f(x)$ $-f(x)$ (3)	$F_i=(\sum f(x)-$ $f(x)) / \sum f(x)$ (4)	ค่าเฉลี่ยของ จำนวนครั้งที่ ถูกคัดเลือก(5)	จำนวนครั้งที่ ถูกคัดเลือก จริง (6)	สตริงใหม่ ที่คัดเลือกแล้ว (7)	$f(x)$ (8)
[3 4 7 6 1 10 8 9 5 2]	282.416	1300.138	0.274	1.096	2	[3 4 7 6 1 10 8 9 5 2]	282.416
[1 8 10 7 6 2 9 4 5 3]	364.168	1218.386	0.257	1.028	1	[3 4 7 6 1 10 8 9 5 2]	282.416
[6 3 4 7 8 9 1 2 10 5]	434.344	1148.210	0.242	0.968	1	[1 8 10 7 6 2 9 4 5 3]	364.168
[5 3 9 1 2 7 8 10 6 4]	501.626	1080.928	0.228	0.912	0	[6 3 4 7 8 9 1 2 10 5]	434.344
ผลรวม $\sum f(x)$	1582.554	4747.662	1	4	4		1363.34
ค่าเฉลี่ย	395.639	1186.916	0.25	1	1		340.836

เฉลี่ยลดลงจากเดิม ผลรวมของค่าใช้จ่ายเป็น 1582.554 ลดลงเหลือ 1363.344 และค่าเฉลี่ยลดลงจาก 395.693 เป็น 340.836

1.5 การครอสโอเวอร์

การครอสโอเวอร์ (Crossover) คือ การจับคู่สตริงและการทำแลกเปลี่ยนส่วนกันเพื่อให้เกิดสตริงใหม่ และค่าความน่าจะเป็นในการครอสโอเวอร์ คือจำนวนเปอร์เซ็นต์ของสตริงที่เกิดการครอสโอเวอร์ในแต่ละเจนเนอเรชัน การครอสโอเวอร์ที่ใช้ในงานวิจัยนี้แบ่งออกเป็น 3 แบบได้แก่ การครอสโอเวอร์แบบ PMX, OX และ CX ดังมีวิธีการดังต่อไปนี้

การครอสโอเวอร์แบบ PMX: ขั้นตอนแรกคือการเลือกคู่สตริงพ่อแม่ขึ้นมาอย่างสุ่ม จากนั้นทำการเลือกตำแหน่งที่จะทำการครอสโอเวอร์อย่างสุ่มที่อยู่ในช่วง $[1, l-1]$ โดยที่ l คือความยาวของสตริง ขอบเขตของการครอสโอเวอร์อยู่ในช่วงเครื่องหมาย “|”

$$p_1 = [10 \ 9 \ 5 \ | \ 6 \ 7 \ 8 \ | \ 2 \ 4 \ 3 \ 1]$$

$$p_2 = [9 \ 8 \ 2 \ | \ 3 \ 4 \ 1 \ | \ 10 \ 6 \ 5 \ 7]$$

ในขั้นตอนต่อไปจะทำการสลับค่าระหว่างสตริงที่อยู่ในช่วง “|” นั่นคือตำแหน่งสุ่มอยู่ในช่วง $[4, 6]$ ของสตริงลูกหลานทั้งสองโดยที่ค่าที่อยู่นอกเครื่องหมาย “|” และเป็นค่าที่ซ้ำกันกับค่าที่อยู่ในช่วง “|” จะถูกกำหนดให้เป็น x

จะเห็นว่าตำแหน่งที่สองของสตริง p_2 มีค่าเป็น 8 ซึ่งซ้ำกับค่าที่อยู่ในช่วง “|” เมื่อทำการสลับแล้ว จึงเปลี่ยนค่าให้เป็น x ก่อน และตำแหน่งอื่นๆ ที่มีค่าซ้ำกันก็จะเปลี่ยนเป็น x ด้วยเช่นกัน

$$o_1 = [10 \ 9 \ 5 \ | \ 3 \ 4 \ 1 \ | \ 2 \ x \ x \ x]$$

$$o_2 = [9 \ x \ 2 \ | \ 6 \ 7 \ 8 \ | \ 10 \ x \ 5 \ x]$$

จากนั้นจึงทำการแทนค่า (map) ดังต่อไปนี้แทนค่า 3 เป็น 6 และ 6 เป็น 3 4 เป็น 7 และ 7 เป็น 4 1 เป็น 8 และ 8 เป็น 1 การแทนค่าเหล่านี้ได้มาจากค่าที่อยู่ในช่วง “|” โดยพิจารณาตำแหน่งของสตริงที่ตรงกัน

สตริงที่ได้เมื่อทำการแทนที่แล้วคือ

$$o_1 = [10 \ 9 \ 5 \ 3 \ 4 \ 1 \ 2 \ 7 \ 6 \ 8]$$

$$o_2 = [9 \ 1 \ 2 \ 6 \ 7 \ 8 \ 10 \ 3 \ 5 \ 4]$$

การครอสโอเวอร์แบบ OX : เช่นเดียวกับวิธีการของ PMX อย่างแรกคือการเลือกคู่สตริงพ่อแม่ขึ้นมาอย่างสุ่ม จากนั้นทำการเลือกตำแหน่งที่จะทำการครอสโอเวอร์อย่างสุ่ม ขอบเขตของการครอสโอเวอร์อยู่ในช่วงเครื่องหมาย “|”

$$p_1 = [9 \ 10 \ 5 \ | \ 6 \ 7 \ 8 \ | \ 2 \ 4 \ 3 \ 1]$$

$$p_2 = [9 \ 8 \ 2 \ | \ 3 \ 4 \ 1 \ | \ 10 \ 6 \ 5 \ 7]$$

สุ่มตำแหน่งที่จะทำการครอสโอเวอร์ได้อยู่ในช่วง $[4, 6]$ จากนั้นเลือกคู่แม่ที่อยู่ในช่วงการครอสโอเวอร์ ถ้าค่าที่ตรงกันให้ทำเครื่องหมาย # เครื่องหมาย # หมายความว่าปล่อยตำแหน่งนั้นไว้ว่าง สตริงลูกหลานจะเป็น

$$o_1 = [9 \ 10 \ 5 \ | \ 6 \ 7 \ 8 \ | \ 2 \ \# \ \# \]$$

$$o_2 = [9 \ \# \ 2 \ | \ 3 \ 4 \ 1 \ | \ 10 \ \# \ 5 \ \#]$$

จากนั้นจะทำการเลื่อนตำแหน่ง โดยยึดถือตำแหน่งหลังช่วง “|” เป็นจุดอ้างอิง (สำหรับ o_1) เป็น 2 และ สำหรับ o_2 เป็น 10) ได้สตริงลูกหลานดังต่อไปนี้

$$o_1 = [6 \ 7 \ 8 \ | \ \# \ \# \ \# \ | \ 2 \ 9 \ 10 \ 5]$$

$$o_2 = [3 \ 4 \ 1 \ | \ \# \ \# \ \# \ | \ 10 \ 5 \ 9 \ 2]$$

ขั้นตอนต่อไปจะเป็นการสลับตำแหน่งที่อยู่ในช่วงของการครอสโอเวอร์ของสตริงทั้งสองที่เหลือ

$$o_1 = [6 \ 7 \ 8 \ | \ 3 \ 4 \ 1 \ | \ 2 \ 9 \ 10 \ 5]$$

$$o_2 = [3 \ 4 \ 1 \ | \ 6 \ 7 \ 8 \ | \ 10 \ 5 \ 9 \ 2]$$

การครอสโอเวอร์แบบ CX: ขั้นตอนแรกเลือกคู่สตริงพ่อแม่ขึ้นมาอย่างสุ่ม พิจารณาดังต่อไปนี้

$$p_1 = [3 \ 1 \ 4 \ 5 \ 9 \ 6 \ 10 \ 2 \ 8 \ 7]$$

$$p_2 = [6 \ 10 \ 8 \ 2 \ 1 \ 7 \ 8 \ 4 \ 9 \ 5]$$

พิจารณาที่ตำแหน่งแรกของสตริง p_1 และ p_2 คงเดิมไว้

$$o_1 = [3 \ - \ - \ - \ - \ - \ - \ - \ - \]$$

$$o_2 = [6 \ - \ - \ - \ - \ - \ - \ - \ - \]$$

ค่าตำแหน่งแรกของสตริง o_2 เป็น 6 นำมาพิจารณาค่าที่เป็น 6 (ตำแหน่งที่ 6) ของสตริง o_1 จะคงเดิมไว้ ส่วนค่าที่เป็น 7 (ตำแหน่งที่ 6) ของสตริง o_2 จะคงเดิมไว้เช่นกัน จะได้เป็น

$$o_1 = [3 \ - \ - \ - \ - \ 6 \ - \ - \ - \]$$

$$o_2 = [6 \ - \ - \ - \ - \ 7 \ - \ - \ - \]$$

จากค่าที่เป็น 7 ของสตริง o_2 นำไปพิจารณาต่อในสตริง o_1 ซึ่งจะคงเดิมไว้ ได้สตริงเป็น

$$o_1 = [3 \ - \ - \ - \ - \ 6 \ - \ - \ 7]$$

$$o_2 = [6 \ - \ - \ - \ - \ 7 \ - \ - \ 5]$$

ด้วยกระบวนการเดิมจะได้สตริงเป็น

$$o_1 = [3 \ - \ - \ 5 \ - \ 6 \ - \ - \ 7]$$

$$o_2 = [6 \ - \ - \ 2 \ - \ 7 \ - \ - \ 5]$$



$$o_1 = [3 \ - \ - \ 5 \ - \ 6 \ - \ 2 \ - \ 7]$$

$$o_2 = [6 \ - \ - \ 2 \ - \ 7 \ - \ 4 \ - \ 5]$$



$$o_1 = [3 \ - \ 4 \ 5 \ - \ 6 \ - \ 2 \ - \ 7]$$

$$o_2 = [6 \ - \ 3 \ 2 \ - \ 7 \ - \ 4 \ - \ 5]$$

กระบวนการในการเลือกตัวที่จะคงเดิมไว้จะสิ้นสุดต่อเมื่อค่าในตำแหน่งใดตำแหน่งหนึ่งของ o_2 มีค่าตรงกับค่าในตำแหน่งแรกของสตริง o_1 จากนั้นจะทำการสลับค่าที่ยังคงเหลืออยู่ทั้งสองสตริงแบบตำแหน่งต่อตำแหน่ง ได้สตริงลูกหลานเป็น

$$o_1 = [3 \ 10 \ 4 \ 5 \ 1 \ 6 \ 8 \ 2 \ 9 \ 7]$$

$$o_2 = [5 \ 1 \ 3 \ 2 \ 9 \ 7 \ 10 \ 4 \ 8 \ 5]$$

1.6 การมิวเตชัน

กระบวนการมิวเตชัน (Mutation) เป็นการเปลี่ยนค่าในบางตำแหน่งของสตริงเพื่อให้เกิดคำตอบใหม่โดยเลือกตำแหน่งหรือยีนในโครโมโซม (Chromosome) หรือสตริงอย่างสุ่ม และทำการสุ่มตำแหน่งอีกหนึ่งตำแหน่งในสตริงนั้นๆ เพื่อทำการสลับค่ากับตำแหน่งที่ถูกเลือกในครั้งแรก โดยมีการกำหนดความน่าจะเป็นในการมิวเตชันเป็นจำนวนเปอร์เซ็นต์ที่จะควบคุมตำแหน่งหรือยีนที่จะผ่านการมิวเตชัน

สมมติว่าสุ่มได้ตำแหน่งในตำแหน่งที่ 8 มีค่าเป็น 6 ดังสตริงในต่อไปนี้

[1 3 2 4 8 10 7 6 | 5 | 9]

จากนั้นทำการสุ่มเลือกตำแหน่งอีกหนึ่งตำแหน่งในสตริงเดียวกัน ได้ค่าในตำแหน่งที่ 2 มีค่าเป็น 3

[1 | 3 | 2 4 8 10 7 | 6 | 5 9]

เมื่อทำการมิวเตชันแล้วจะได้สตริงใหม่เป็น

[1 6 2 4 8 10 7 3 5 9]

รายละเอียดของ GAs สามารถดูเพิ่มเติมได้จาก Goldberg [12] และ Michalewicz [8]

2. การศึกษาเบื้องต้นและการออกแบบการทดลอง

Lawrence [13] กล่าวว่า การกำหนดค่าพารามิเตอร์ของ GAs เป็นสิ่งที่สำคัญ เนื่องจากเป็นตัวกำหนดให้การหาคำตอบของ GAs ประสบความสำเร็จหรือล้มเหลวได้ และในอดีตยังไม่มียุทธศาสตร์ที่ทำการศึกษารวบรวมการกำหนดค่าพารามิเตอร์เหล่านี้ ดังนั้นในการทำวิจัยฉบับนี้จึงมีความจำเป็นที่จะต้องทำการทดลองเบื้องต้น (Pilot Study) โดยจุดประสงค์ของการทดลองครั้งนี้เพื่อหาพารามิเตอร์ที่เหมาะสมที่ทำให้การหาคำตอบเป็นไปได้อย่างรวดเร็วและได้คำตอบที่ดีหรือใกล้เคียง

การศึกษาเบื้องต้นจากปัญหาดังกล่าว มีขั้นตอนดังต่อไปนี้

1. หาจำนวนประชากรที่เหมาะสม โดยการแปรค่าพารามิเตอร์ของประชากรเป็น 10 30 50 70 90 110 และกำหนดพารามิเตอร์ของจำนวนเงินเนอเรชันเป็น 6000 เพื่อให้แน่ใจว่า GAs สามารถหาคำตอบได้ดีที่สุดและเร็วที่สุด กำหนด pC_{pmx} เป็น 0.8 (Suresh et al. [10]) และกำหนด p_m เป็น 0.1 เนื่องจากการมิวเตชันมีโอกาสเกิดขึ้นน้อยครั้ง (Goldberg [12]) จากการทดลองพบว่าจำนวนประชากรเป็น 10 ให้คำตอบที่ดีที่สุดและสามารถพบคำตอบเร็วที่สุด
2. หาจำนวนเงินเนอเรชันที่เหมาะสม โดยการแปรค่าพารามิเตอร์ของจำนวนเงินเนอเรชันเป็น 6000 10000 30000 และ 60000 โดยใช้พารามิเตอร์ของจำนวนประชากรที่ดีที่สุดตามข้อ 1 และพารามิเตอร์อื่นๆ ตามข้อ 1 จากการทดลองได้จำนวนเงินเนอเรชันเป็น 10000 (เป็น 10 เท่าของเงินเนอเรชันที่พบคำตอบ) ซึ่งทำให้แน่ใจได้คำตอบที่ดีที่สุดและทำให้แน่ใจว่าเงินเนอเรชันนี้สามารถ

ครอบคลุมคำตอบที่ดีที่สุดในการทดลองได้

3. หาค่ามิวเตชันที่เหมาะสม โดยการแปรค่าพารามิเตอร์ของการมิวเตชันเป็น 0.001 0.01 0.1 0.2....1.0 โดยใช้ค่าพารามิเตอร์ของจำนวนประชากรและจำนวนเงินเนอเรชันในข้อ 1 และ 2 โดยให้พารามิเตอร์อื่นๆ ตามข้อ 1 จากการทดลองได้ p_m เท่ากับ 0.2 ทำให้ได้คำตอบที่ดีที่สุดและสามารถพบคำตอบเร็วที่สุด
4. หาค่าการครอสโอเวอร์ที่เหมาะสม โดยการแปรค่าพารามิเตอร์ของการครอสโอเวอร์ pC_{pmx} , pC_{ox} และ pC_{cx} จาก 0.1 0.2... 1.0 โดยใช้ค่าพารามิเตอร์ของจำนวนประชากร จำนวนเงินเนอเรชัน ค่าความน่าจะเป็นของการมิวเตชันในข้อ 1, 2 และ 3 จากการทดลองได้ค่าพารามิเตอร์ของการครอสโอเวอร์ทั้งสามเป็น 0.4, 0.1 และ 0.6 ตามลำดับ ซึ่งทำให้ได้คำตอบที่ดีที่สุดและสามารถพบคำตอบเร็วที่สุด

ค่าพารามิเตอร์ที่ใช้ในการทดลองได้มาจากการศึกษาเบื้องต้น

ดังนี้

จำนวนประชากร = 10

เงินเนอเรชัน = 10000

p_m = 0.2

pC_{pmx} = 0.4

pC_{ox} = 0.1

pC_{cx} = 0.6

ขอบเขตในการค้นหา = 10^4 (8,628,800) วิธี

อัตราส่วนในการค้นหา = $(10 \times 10000) / 10!$

= 0.027 หรือ $\approx 3\%$

ขอบเขตในการค้นหา หมายถึง จำนวนคำตอบทั้งหมดที่เป็นไปได้ อัตราส่วนในการค้นหา หมายถึง อัตราส่วนระหว่าง จำนวนครั้งในการหาคำตอบของ GAs กับจำนวนคำตอบทั้งหมดที่เป็นไปได้

เกณฑ์ในการวัดสมรรถภาพ (Measure of Performance) ที่ใช้งานวิจัยนี้มี 2 เกณฑ์คือ จำนวนเงินเนอเรชันเฉลี่ยที่พบคำตอบที่ดีที่สุดและค่าเฉลี่ยของ TC เกณฑ์ตัวแรกใช้เพื่อเปรียบเทียบความเร็วในการวิ่งเข้าหาคำตอบที่ดีที่สุดของวิธีการของ GAs ที่พัฒนาขึ้นแต่ละชนิด ถ้าวิธีการใดมีจำนวนเงินเนอเรชันเฉลี่ยที่พบคำตอบที่ดีที่สุดน้อย จะหมายความว่าวิธีการนั้นเป็นวิธีการที่มีประสิทธิภาพสูงเนื่องจากโดยเฉลี่ยจะใช้เวลาในการหาคำตอบน้อย เกณฑ์ตัวที่สองเป็นเกณฑ์ที่ใช้แสดงค่าใช้จ่ายโดยรวมที่เกิดขึ้นจากการใช้วิธีการของ GAs ที่แตกต่างกัน เป็นไปได้ว่าหลังจากที่ GAs ได้ทำงานไปจนครบจำนวนเงินเนอเรชันสูงสุดที่กำหนดให้แล้ว คำตอบสุดท้ายของ GAs อาจจะยังไม่เข้าหาคำตอบที่ดีที่สุด (Optimal Solution) ก็ได้ ดังนั้น เมื่อดูจากเกณฑ์ตัวนี้เราก็ยังสามารถทราบได้ว่าวิธีการของ GAs วิธีการใดดีกว่ากัน โดยถ้าค่า TC ตอนสุดท้ายของวิธีใดน้อยกว่าจะหมายความว่าวิธีการนั้นดีกว่า ในความเป็นจริงแล้วเกณฑ์ทั้งสองควรใช้ประกอบกัน เพราะเป็นไปได้ว่ามี (เกณฑ์ที่ 1 ดี) แต่ทว่ามีค่าคำตอบสุดท้ายสูง

(เกณฑ์ที่ 2 ไม่ดี) ก็ได้เนื่องจากว่าอาจจะติดอยู่ที่ Local Optimal

3. การทดลองเพื่อปรับปรุงประสิทธิภาพของ GAs

จากค่าพารามิเตอร์ของ GAs ที่ได้มาข้างต้น เราจะทำการทดลองเพื่อค้นหาวិธีการในการหาคำตอบอย่างมีประสิทธิภาพสำหรับปัญหาการจัดผังโรงงาน ซึ่งแบ่งออกเป็น 3 การทดลองด้วยกัน

การทดลองที่ 1 : เปรียบเทียบวิธีการครอสโอเวอร์แบบ PMX, OX และ CX

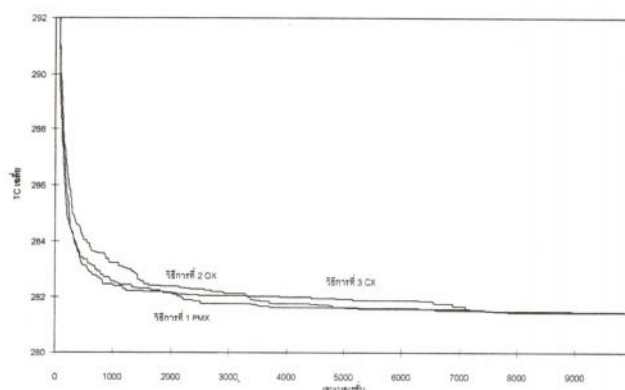
วัตถุประสงค์ เพื่อเปรียบเทียบประสิทธิภาพของการครอสโอเวอร์ทั้ง 3 แบบ และหาวิธีการครอสโอเวอร์ที่เหมาะสมสำหรับนำไปใช้ในการทดลองต่อไป

วิธีการทดลอง นำพารามิเตอร์ที่ได้จากการศึกษาเบื้องต้น (จำนวนประชากร, p_m , pC_{pmx} , pC_{ox} และ pC_{cx}) มาเปลี่ยนแปลงค่าให้อยู่ในขอบเขตใกล้เคียงกับค่าที่ได้จากการทดลองเบื้องต้น (เช่น p_m ให้มีการเปลี่ยนแปลงค่าเป็น 0.15, 0.2 และ 0.25) ทำให้ได้ผลการ Run โปรแกรมทั้งหมดจำนวน 45 การทดลอง ซึ่งได้จากการเปลี่ยนแปลงจำนวนประชากร 3 ค่า (5, 10 และ 15), p_m 3 ค่า, และ pC 5 ค่า เพื่อนำคำตอบมาหาค่าเฉลี่ยและเป็นตัวแทนสำหรับวิธีการครอสโอเวอร์แบบต่างๆ (PMX, OX และ CX) กำหนดให้จำนวนเงินเนอเรชันในการทดลองทุกครั้งมีค่าคงที่เท่ากับ 10000 เหตุผลประการหนึ่งของการทำการทดลองเช่นนี้มาจากการที่ GAs มีความไวต่อการเปลี่ยนแปลงค่าพารามิเตอร์อย่างมาก ดังนั้นการทดลองลักษณะนี้จะทำให้ทราบถึงความไวต่อการเปลี่ยนแปลงของ GAs และยังสามารถหาตัวแทนที่ดีของคำตอบของ GAs ได้อีกด้วย

ผลการทดลอง จากการวิจัยของ Chan และ Tansri [8] พบว่าการครอสโอเวอร์แบบ PMX จะมีการลู่เข้าหาคำตอบเป็นไปอย่างสม่ำเสมอ ส่วนแบบ OX มีการลู่เข้าหาคำตอบช้าที่สุด และทำงานได้ดีเมื่อมีจำนวนประชากรน้อย สำหรับการครอสโอเวอร์แบบ CX จะมีการลู่เข้าหาคำตอบอย่างรวดเร็วเมื่อมีจำนวนประชากรน้อยๆ แต่คำตอบอาจจะติดอยู่ใน Local optimal ได้ง่าย

จากการทดลองเปรียบเทียบการหาคำตอบของการครอสโอเวอร์ทั้งสามวิธี ตามรูปที่ 6 จะเห็นได้ว่า สำหรับปัญหานี้ ถึงแม้ว่าค่าเฉลี่ยของค่าใช้จ่ายโดยรวม (Average Total Cost) สุดท้ายของการครอสโอเวอร์ทุกวิธีจะมีค่าเท่ากันคือ 281.44 ที่เงินเนอเรชันที่ 10000 จะพบว่าเมื่อเปรียบเทียบระหว่างแบบ CX กับ PMX และ OX ค่าเฉลี่ยของค่าใช้จ่ายโดยรวมของแบบ CX มีค่ามากกว่า PMX และ OX เพียงเล็กน้อยในช่วงประมาณเงินเนอเรชันที่ 3000-7000 แต่จากตารางที่ 2 พบว่า การครอสโอเวอร์แบบ CX จะให้ค่าเฉลี่ยของ

จำนวนเงินเนอเรชันที่พบคำตอบน้อยที่สุด เมื่อเปรียบเทียบกับแบบ PMX และ OX (เงินเนอเรชันที่ 2110.5) ซึ่งสามารถอธิบายปรากฏการณ์นี้ได้ว่า การครอสโอเวอร์แบบ CX โดยเฉลี่ยแล้วจะทำให้เกิดการลู่เข้าหาคำตอบได้อย่างรวดเร็ว แต่อาจจะทำให้คำตอบที่ได้นี้ติดอยู่ใน Local optimal ได้ง่าย ซึ่งทำให้ค่าใช้จ่ายโดยรวมเฉลี่ยของ CX ไม่มีการเปลี่ยนแปลงเป็นระยะเวลานาน ดังนั้นเพื่อเป็นการคงไว้ซึ่งข้อดีของการครอสโอเวอร์แบบ CX (ลู่เข้าหาคำตอบเร็ว) และปรับปรุงข้อเสียเกี่ยวกับคำตอบที่ติดอยู่กับ Local optimal เป็นเวลานาน ในงานวิจัยนี้จึงเลือกทำการปรับปรุงการครอสโอเวอร์แบบ CX ในการทดลองต่อไป



รูปที่ 6 การเปรียบเทียบค่าตอบเฉลี่ยของวิธีการครอสโอเวอร์แบบ PMX, OX และ CX

ตารางที่ 2 เปรียบเทียบเงินเนอเรชันเฉลี่ยที่พบคำตอบของการครอสโอเวอร์แบบ PMX, OX และแบบ CX

วิธีการ	จำนวนเงินเนอเรชันเฉลี่ยที่พบคำตอบที่ดีที่สุด
1 การครอสโอเวอร์แบบ PMX	2370.2143
2 การครอสโอเวอร์แบบ OX	2488.7857
3 การครอสโอเวอร์แบบ CX	2110.5000

การทดลองที่ 2: เปรียบเทียบความเร็วในการลู่เข้าหาคำตอบเมื่อให้สตริงคำตอบเริ่มต้นจากอิวิริสติกแบบ SDPI แก่ GAs โดยที่สตริงคำตอบเริ่มต้นมีลักษณะเดียวกันทุกประการ แตกต่างอยู่ที่จำนวนที่ใช้เท่านั้น

วัตถุประสงค์ เพื่อให้ GAs สามารถหาคำตอบได้รวดเร็วขึ้น โดยการป้อนสตริงคำตอบเริ่มต้นที่ดีที่ได้จากอิวิริสติก SDPI แก่ประชากรเริ่มต้นของ GAs แทนที่จะใช้ประชากรเริ่มต้นเป็นแบบสุ่มทั้งหมด

วิธีการทดลอง หากค่าคำตอบเริ่มต้นที่ดีจากวิธีการ SDPI มา 1 คำตอบ จากนั้นนำมาเป็นส่วนหนึ่งประชากรของคำตอบเริ่มต้นของ GAs หมายความว่าถ้ากำหนดให้จำนวนประชากรทั้งหมดเท่ากับ 10 สดริง สมมติว่าให้สดริงคำตอบจาก SDPI มีจำนวน 1 สดริงในประชากรทั้งหมด ดังนั้นจะหาสดริงที่เหลืออีก 9 สดริงได้จากการสุ่มตามปกติของ GAs ในการทดลองนี้จะทำการแปรจำนวนสดริงคำตอบเริ่มต้นเป็นจำนวน 1 และ 5 สดริง (ดังนั้นจำนวนสดริงคำตอบเริ่มต้นที่ได้จากการสุ่มตามปกติจะเป็น 9 และ 5 ตามลำดับ) โดยที่จำนวนประชากรมีค่าคงที่เท่ากับ 10 สดริง หรืออาจจะกล่าวให้ชัดขึ้นได้ว่า วิธีการที่ 4 และ 5 เป็นความพยายามในการปรับปรุงประสิทธิภาพของการครอสโอเวอร์แบบ CX ที่ได้จากการทดลองที่ผ่านมา ซึ่งทำให้การป้อนให้สดริงของคำตอบเริ่มต้นเป็นสดริงที่ดี (ค่าใช้จ่ายต่ำ) สดริงที่นำมาเป็นสดริงคำตอบเริ่มต้นเหล่านี้เป็นสดริงที่มีลักษณะเหมือนกันทุกประการ และได้มาจากวิธีอิมพัลส์พุ่มที่อีวิวิดิค (SDPI) ซึ่งมีค่าใช้จ่ายเป็น 282.416 ซึ่งค่านี้ใกล้เคียงกับคำตอบสุดท้าย (หมายถึงคำตอบที่ได้จากการคำนวณของ GAs ตามจำนวนเงินเนอเรนซ์ที่กำหนด ซึ่งอาจจะไม่ใช่คำตอบที่ดีที่สุดหรือ Optimum Solution ก็ได้) ที่ได้จากการทดลองที่ 1 เป็นอย่างมาก (พารามิเตอร์ตัวอื่นใช้เหมือนกับการทดลองที่ 1)

ผลการทดลอง จากตารางที่ 3 พบว่า วิธีการที่ 4 และ 5 ได้จำนวนเงินเนอเรนซ์ที่พบคำตอบที่ดีที่สุดเฉลี่ยมีค่าเป็น 5264.67 และ 6114.06 ตามลำดับ ซึ่งถ้าเปรียบเทียบกับวิธีการที่ 3 (การครอสโอเวอร์แบบ CX โดยใช้คำตอบเริ่มต้นที่ได้จากการสุ่มตามปกติทั้งหมด ซึ่งมีจำนวนเงินเนอเรนซ์เฉลี่ยที่พบคำตอบที่ดีที่สุดเป็น 2110.5) จะพบว่า วิธีการที่ 4 และ 5 จะมีประสิทธิภาพด้อยกว่าวิธีการที่ 3 อย่างมาก ดังนั้นการที่เราพยายามจะให้คำตอบเริ่มต้นที่ดีแก่ GAs โดยหวังว่า GAs จะให้ได้คำตอบสุดท้ายรวดเร็วขึ้นและดีกว่าวิธีการแบบธรรมดานั้น อาจไม่ได้ผลตามที่คาดไว้เสมอไป ทั้งนี้เนื่องจากว่าคำตอบเริ่มต้นที่ดีนั้นอาจจะเป็นคำตอบที่ทำให้ GAs ติดอยู่ใน Local optimal และยากที่จะหลุดจากจุดนั้นออกมาได้ ซึ่งทำให้แทนที่จะเป็นผลดีต่อกระบวนการหาคำตอบกลับทำให้ผลลัพธ์ที่ได้แย่ลง เมื่อเทียบระหว่างวิธีการที่ 3 และ 4 จะพบว่าจำนวนเงินเนอเรนซ์เฉลี่ยที่พบคำตอบที่ดีที่สุดจะแตกต่างกันอย่างมาก (2110.5 กับ 5264.7) ซึ่งอาจเกิดจากการ run โปรแกรม GAs โดยใช้จำนวนครั้งน้อยเกินไป แต่วัตถุประสงค์ของการทดลองนี้คือการต้องการแสดงให้เห็นถึงแนวโน้มอย่างคร่าวๆ เกี่ยวกับการใช้สดริงคำตอบที่มีความหลากหลาย เมื่อเปรียบเทียบกับสดริงคำตอบแบบเดียวกันหลายคำตอบ ซึ่งจะเห็นได้ว่าการให้คำตอบเริ่มต้นที่มีความหลากหลายนั้นจะส่งผลให้ได้จำนวนเงินเนอเรนซ์เฉลี่ยที่พบคำตอบน้อยกว่า นอกจากนี้แล้วการเพิ่มจำนวนของสดริงที่ดีเข้าไปในประชากรเริ่มต้นจาก 1 เป็น 5 สดริงที่เหมือนกัน ทำให้การสุ่มหาคำตอบช้าลง (เงินเนอเรนซ์เฉลี่ยที่พบคำตอบที่ดีที่สุดเพิ่มขึ้นจาก 5264.67 เป็น 6114.06 ดังแสดงใน

ตารางที่ 3) ซึ่งอธิบายได้ว่า การเพิ่มจำนวนประชากรที่ดีซึ่งมีลักษณะเหมือนกันทุกประการเข้าไปในประชากรเริ่มต้นของ GAs นั้นจะทำให้ประสิทธิภาพในการสร้างประชากรรุ่นต่อไปที่มีค่าพัฒนาสูงขึ้นน้อยลง เนื่องจากว่าโอกาสที่ประชากรเดิมจะครอสโอเวอร์แล้วได้ประชากรใหม่ที่มีความแตกต่างจากเดิมน้อยลง ดังนั้นวิธีการแก้ไขก็คือควรที่จะเพิ่มสดริงที่ดีที่มีลักษณะแตกต่างกันให้กับประชากรเริ่มต้นของ GAs ซึ่งจะเป็นการทดลองในหัวข้อต่อไป

ตารางที่ 3 การเปรียบเทียบการให้คำตอบเริ่มต้นที่ดีแก่ประชากรของ GAs โดยที่คำตอบเริ่มต้นเป็นแบบเดียวกัน

วิธีการที่	จำนวนสดริงคำตอบเริ่มต้น (สดริงที่มีลักษณะเดียวกัน)	เงินเนอเรนซ์เฉลี่ยที่ พบคำตอบที่ดีที่สุด
4	1	5264.6667
5	5	6114.0625

การทดลองที่ 3: เปรียบเทียบการสุ่มหาคำตอบเมื่อให้คำตอบเริ่มต้นที่ดีจาก SDPI ที่แตกต่างกันแก่ประชากรเริ่มต้นของ GAs

วัตถุประสงค์ เพื่อให้ GAs สามารถหาคำตอบได้รวดเร็วขึ้นและได้คำตอบที่ดีขึ้นโดยการใช้สดริงคำตอบเริ่มต้นที่ดีหลายคำตอบจาก SDPI มาป้อนให้กับประชากรเริ่มต้นของ GAs

วิธีการทดลอง เนื่องจากในการทดลองที่ 2 พบว่าการให้คำตอบเริ่มต้นด้วยสดริงคำตอบที่ดีแบบเดียวกันไม่อาจทำให้ GAs หาคำตอบได้รวดเร็วขึ้น ในการทดลองนี้จึงทำการทดลองใหม่โดยหาคำตอบที่ดีที่แตกต่างกันจาก SDPI ขึ้นมาจำนวนหนึ่ง (5 และ 10 สดริง) ซึ่งทำให้ได้สดริงที่ดีที่อยู่ในประชากรเริ่มต้นมีความหลากหลายมากขึ้น (พารามิเตอร์อื่นใช้เหมือนการทดลองที่ 1)

ตารางที่ 4 การเปรียบเทียบการให้คำตอบเริ่มต้นที่ดีที่แตกต่างกันจากวิธีการ SDPI กับประชากรเริ่มต้นของ GAs

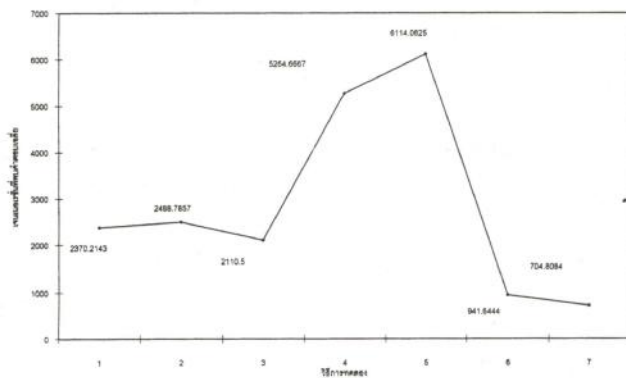
วิธีการที่	จำนวนสดริงคำตอบเริ่มต้น (สดริงลักษณะแตกต่างกัน)	เงินเนอเรนซ์เฉลี่ยที่ พบคำตอบที่ดีที่สุด
6	5	941.6444
7	10	704.1333

ผลการทดลอง ตารางที่ 4 แสดงถึงการเปรียบเทียบจำนวนเงินเนอเรนซ์เฉลี่ยที่พบคำตอบที่ดีที่สุดเมื่อให้สดริงที่ดีที่แตกต่างกันของประชากรเริ่มต้นจากวิธีการ SDPI โดยให้สดริงที่ดีนั้นมีจำนวน 5 และ 10 สดริงตามลำดับ จะเห็นได้ว่าจำนวนเงินเนอเรนซ์เฉลี่ยที่พบคำตอบที่ดีที่สุด

ของวิธีการที่ 7 จะน้อยกว่าวิธีการที่ 6 เนื่องจากมีจำนวนสตริงคำตอบเริ่มต้นที่ดีที่แตกต่างกันมากกว่า ซึ่งทำให้มีโอกาสที่จะพบคำตอบที่มีค่าฟิตเนสสูงๆ ได้ดีกว่าและรวดเร็วกว่านั่นเอง จากการทดลองที่ผ่านมาทำให้เราพบว่า การให้สตริงที่ดีแก่ประชากรเริ่มต้นของ GAs ไม่จำเป็นต้องทำให้ GAs หาคำตอบที่ดีได้อย่างรวดเร็วเสมอไป (การทดลองที่ 2) แต่การมีสตริงเริ่มต้นที่ดีที่มีความหลากหลายและมีเป็นจำนวนมาก จะทำให้โอกาสในการค้นพบคำตอบที่ดีมีมากขึ้นและรวดเร็วขึ้น

4. วิเคราะห์ผลการทดลอง

จากวิธีการทดลองทั้ง 7 วิธี ที่แตกต่างกันด้วยวิธีการให้ประชากรเริ่มต้นและการครอสโอเวอร์ดังที่กล่าวมาแล้ว สามารถนำมาสรุปได้ดังรูปที่ 7 จะเห็นได้ว่า การทดลองวิธีการที่ 7 (หาคำตอบโดย



รูปที่ 7 จำนวนเงินเนอเรนเฉลี่ยที่ GAs ค้นพบคำตอบโดยวิธีการทดลองต่างๆ

ตารางที่ 5 การวิเคราะห์ ANOVA

Source of variation	Sum of square	Degree of freedom	Mean square	F	Sig
Crossover method	5.26E+08	6	87692512	18.387	.000
Error	8.73E+08	183	4769207		
Total	1.40E+09	189			

ใช้การครอสโอเวอร์แบบ CX และให้คำตอบเริ่มต้นที่ดีที่แตกต่างกันจากฮิวริสติก SDPI จำนวน 10 สตริง) สามารถทำให้พบคำตอบที่ดีที่สุดสำหรับปัญหานี้ (คำตอบที่ดีที่สุดซึ่งได้จาก Exact Algorithm มีค่าเท่ากับ 280.85) และรวดเร็วที่สุด ส่วนวิธีการที่ทำให้พบคำตอบช้าที่สุดคือวิธีการที่ 5 (หาคำตอบโดยใช้การครอสโอเวอร์แบบ CX โดยให้คำตอบเริ่มต้นที่ดีที่มีลักษณะเหมือนกันด้วยฮิวริสติก SDPI จำนวน 5 สตริง)

4.1 การวิเคราะห์ความแปรปรวน

ในส่วนนี้จะทำการวิเคราะห์ความแตกต่างของจำนวนเงิน-

เนอเรนเฉลี่ยในการค้นพบคำตอบว่า แต่ละวิธีการที่ทำการทดลองมีผลทำให้จำนวนเงินเนอเรนเฉลี่ยในการค้นพบคำตอบแตกต่างกันอย่างมีนัยสำคัญหรือไม่ โดยกำหนดให้มีระดับนัยสำคัญเป็น 0.05

จากตารางที่ 5 สรุปได้ว่า มีวิธีการทดลองอย่างน้อยสองวิธีที่ทำให้จำนวนเงินเนอเรนเฉลี่ยที่พบคำตอบมีความแตกต่างกันที่ระดับนัยสำคัญ 0.05 ดังนั้นจึงทำการทดสอบด้วยวิธีการทดสอบแบบพหุคูณ (Multiple Range Test) ด้วยวิธี Duncan ที่ระดับความเชื่อมั่น 95% (Montgomery [14]) เพื่อจะแบ่งกลุ่มในการหาคำตอบจากเงินเนอเรนเฉลี่ยที่พบคำตอบที่เกิดขึ้นทั้งหมด

ผลการวิเคราะห์สามารถจัดกลุ่มได้เป็น 4 กลุ่มดังตารางที่ 6 โดยกลุ่มแรก (วิธีการที่ 6 และ 7) เป็นกลุ่มที่สามารถหาคำตอบได้เร็วที่สุด ถ้าพิจารณาเปรียบเทียบกับรูปที่ 8 จะเห็นได้ว่าวิธีการที่ 7 เป็นวิธีการที่ทำให้พบคำตอบได้เร็วที่สุด และกลุ่มของวิธีการที่ 4 และ 5 เป็นวิธีการที่ทำให้พบคำตอบช้าที่สุดด้วยระดับนัยสำคัญ 5%

ตารางที่ 6 การวิเคราะห์ Duncan's Multiple Range Test

วิธีการทดลอง	เงินเนอเรนที่พบคำตอบเฉลี่ย
1	2370.2143 (c)
2	2488.7857 (c)
3	2110.5 (b, c)
4	5264.6667 (d)
5	6114.0625 (d)
6	941.644 (a, b)
7	704.1333 (a)

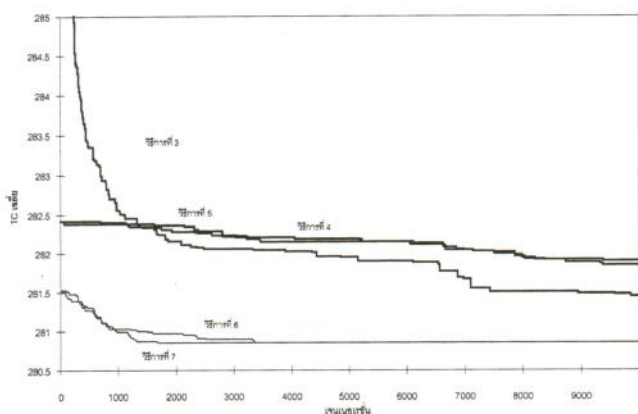
หมายเหตุ : a, b และ c แสดง Homogeneous Group

จากรูปที่ 8 มีประเด็นที่น่าสนใจอีกประเด็นหนึ่งคือ วิธีที่ 3, 4 และ 5 ถึงแม้ว่าจะ Run จนครบตามเงินเนอเรนที่กำหนดให้แล้ว (10,000 เงินเนอเรน) แต่ค่า TC เฉลี่ยมีค่าสูงกว่าที่ได้จากวิธีที่ 6 และ 7 (ที่ยังไม่ได้ Run GAs เลยแม้แต่ครั้งเดียว นั่นคือเงินเนอเรนที่ 0) เหตุการณ์เช่นนี้ไม่ใช่ว่าเป็นเรื่องที่น่าแปลกใจเลย เพราะว่า GAs เป็นกระบวนการหาคำตอบที่มีการปรับปรุงอย่างต่อเนื่อง ซึ่งโดยมากแล้วคำตอบเริ่มต้นของ GAs จะมีค่าคำตอบที่ไม่ค่อยดี เนื่องจากใช้กระบวนการสุ่มคำตอบขึ้นมาจากประชากรของคำตอบทั้งหมด ดังนั้นคำตอบเริ่มต้นของ GAs น่าจะเทียบกับค่าคำตอบของ Heuristic ไม่ได้ แต่ในขณะที่ GAs ปรับปรุงคำตอบให้ดีขึ้นเรื่อยๆ ในเงินเนอเรนต่อๆ มา ก็จะทำให้

คำตอบที่ได้จาก GAs นั้นเข้าใกล้หรือบางครั้งอาจจะดีกว่าคำตอบของ Heuristic ซึ่งคำตอบสุดท้ายของ GAs ที่ควรจะเป็นก็คือคำตอบที่ดีที่สุด (Optimal Solution) นั่นเอง การที่คำตอบสุดท้ายของ GAs วิธีที่ 3 - 5 ที่แสดงอยู่ในรูปที่ 8 นี้ดีกว่า Heuristic ก็เพราะว่าจำนวนเงินเนอเรนซ์ที่ทำการทดลองน้อยเกินไป แน่นอนว่าถ้ากำหนดให้เงินเนอเรนซ์สูงที่สุดมีค่ามากกว่านี้มากๆ คำตอบของวิธีที่ 3-5 ควรจะเป็นคำตอบเดียวกับคำตอบที่ดีที่สุด (ถ้าไม่ไปติดอยู่ใน Local optimal เสียก่อน) นอกจากนั้นแล้ววิธี Heuristic ส่วนมากจะไม่ได้คำตอบที่เป็น Optimal Solution อีกสิ่งหนึ่งที่รูปที่ 8 ไม่ได้กล่าวถึงก็คือ ในการหาคำตอบของวิธี Heuristic นั้นก็ต้องใช้เวลาเช่นกัน ถึงแม้ว่าจะใช้เวลาเมื่อเทียบกับวิธีการของ GAs ก็ตามดังนั้นการนำเอาวิธี Heuristic มาเป็นจุดเริ่มต้นสำหรับ GAs นั้นเป็นแนวคิดที่สมเหตุผล เพราะว่าจะลดเวลาการค้นหาคำตอบเริ่มต้นในช่วงแรกของ GAs ซึ่งอาจจะต้องใช้เวลาานาน นอกจากนั้นยังปรับปรุงคุณภาพของคำตอบของ Heuristic ให้ดียิ่งขึ้นและใกล้ค่า Optimal Solution มากขึ้น เมื่อจำนวนเงินเนอเรนซ์เพิ่มขึ้นอีกด้วย

5. สรุป

งานวิจัยฉบับนี้แสดงให้เห็นถึงการนำ GAs ร่วมกับอิมพัลฟ์-เมนต์อีวิริสติกแบบ SDPI ในการแก้ปัญหาการจัดผังโรงงาน เนื่องจากการหาคำตอบของปัญหาการจัดผังโรงงานด้วยวิธีการอีวิริสติกแต่เพียงอย่างเดียวอาจได้คำตอบที่ไม่ดี และการหาคำตอบของปัญหาการจัดผังโรงงานด้วย GAs แต่เพียงอย่างเดียว ถึงแม้จะได้คำตอบที่ดี แต่ต้องใช้ระยะเวลานานกว่าจะได้คำตอบเพื่อแก้ปัญหาดังกล่าวนี้ จึงได้นำวิธีการทั้งสองมาใช้ร่วมกันในการหาคำตอบของปัญหาการจัดผังโรงงาน โดยป้อนสตริงคำตอบเริ่มต้นที่ดีที่ได้จากอีวิริสติกให้แก่ประชากรเริ่มต้นของ GAs การทำเช่นนี้มีวัตถุประสงค์เพื่อให้การแก้ปัญหาในการใช้เวลานในการหาคำตอบเป็นไปได้อย่างรวดเร็วและได้คำตอบที่ดีพร้อมกัน



รูปที่ 8 การเปรียบเทียบการหาคำตอบของวิธีการต่างๆ

จากการทดลองพบว่าการจัดผังโรงงานด้วยวิธีการดังกล่าว (วิธีการที่ 6 และ 7) ให้คำตอบที่ดีกว่าอีวิริสติกแต่เพียงอย่างเดียว (SDPI) และสามารถลดเวลาในการหาคำตอบเมื่อเปรียบเทียบกับ GAs แต่เพียงอย่างเดียว สิ่งที่จะต้องคำนึงถึงให้มากเกี่ยวกับ GAs ก็คือ การให้คำตอบที่ดีแก่ประชากรเริ่มต้นของ GAs ไม่จำเป็นว่าจะทำให้ประสิทธิภาพของ GAs ดีขึ้นเสมอไป ดังแสดงในวิธีการที่ 4 และ 5 ที่มีการเพิ่มจำนวนสตริงที่ดีให้กับประชากรเริ่มต้นของ GAs แต่ทว่าสตริงเหล่านั้นเป็นสตริงลักษณะเดียวกันทั้งหมด ซึ่งทำให้โอกาสในการสร้างลูกหลานของ GAs ที่มีฟิตเนสสูงๆ น้อยลง ดังนั้นสิ่งที่พึงกระทำก็คือ การให้สตริงเริ่มต้นที่ดีที่มีความหลากหลายให้กับประชากรเริ่มต้นของ GAs มากกว่า ซึ่งเป็นการเพิ่มโอกาสให้กับการหาคำตอบที่ดีได้อย่างรวดเร็วขึ้น

แต่อย่างไรก็ตามวิจัยนี้ได้ทำการทดสอบกับปัญหาตัวอย่างเพียงแค่หนึ่งปัญหาเท่านั้น โครงสร้างของปัญหาอื่นอาจนำไปสู่ผลสรุปที่แตกต่างกันและน่าที่จะมีการทดลองต่อไป

เอกสารอ้างอิง

- [1] Kusiak, A., Intelligence Manufacturing System, Prentice-Hall Inc, 1990.
- [2] Askin, R.G., Strandridge C.R., Modeling and Analysis of Manufacturing Systems, John Wiley & Sons Inc, 1993.
- [3] French, S., *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*, Ellis Horwood Limited, 1982.
- [4] Fransis, R.L., McGinnis, L.F., White, J., *Facility Layout and Location: An Analytical approach*, 2nd ed., 1992.
- [5] Armour, G.C., and Buffa, E.S., A Heuristic Algorithm and Simulation Approach, *AIIE Transactions*, Vol. 7, No. 4, pp. 432-437, 1963.
- [6] Nugent, C.E., Vollman, R.E., and Ruml, J., An experimental comparison of techniques for the assignment of facilities to locations, *Operation Research*, Vol. 16, pp. 150-173, 1968.
- [7] Holland, J.H., *Adaptation in Natural and Artificial System*, Ann Arbor, The University of Michigan Press, 1975.
- [8] Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Programs 3rd rev and extended*, Springer Verlag Berlin Heidelberg New York, 1996.
- [9] Chan, K.C., and Tansri, H., A Study of Genetic Crossover on the Facility Layout Problems, *Computer Industrial Engineering*, Vol. 26, pp. 537-550, 1994.