



## การจัดลำดับการผลิตรถยนต์แบบมากวัตถุประสงค์บนสายการประกอบผลิตภัณฑ์ผสมแบบสองด้าน

ตรัยรัตน์ เกิดโภคทรัพย์ และ ปารเมศ ชูติมา\*

ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

\* ผู้นิพนธ์ประสานงาน โทรศัพท์ 08-0449-8844 อีเมล: parames.c@chula.ac.th DOI: 10.14416/j.kmutnb.2018.06.010

รับเมื่อ 16 ตุลาคม 2560 ตอรับเมื่อ 6 กุมภาพันธ์ 2561 เผยแพร่ออนไลน์ 21 มิถุนายน 2561

© 2018 King Mongkut's University of Technology North Bangkok. All Rights Reserved.

### บทคัดย่อ

วิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์โดยยึดหลักการจำแนก (A Multi-Objective Evolutionary Algorithm based on Decomposition; MOEA/D) เป็นเมตาฮีริสติกเชิงวิวัฒนาการที่ได้รับการพัฒนามาเพื่อแก้ปัญหาแบบมากวัตถุประสงค์ (Many-Objective Optimization Problems; MaOPs) โดยมีแนวคิดในการค้นหาคำตอบด้วยการจำแนกปัญหาออกเป็นปัญหาย่อยเพื่อหาคำตอบที่ดีที่สุดของปัญหาย่อยนั้นๆ งานวิจัยนี้ จึงเสนอ MOEA/D มาเปรียบเทียบกับวิธีการวิวัฒนาการโดยใช้ผลต่างแบบหลายวัตถุประสงค์ (Multi-Objective Differential Evolution Algorithm; MODE) ในการแก้ปัญหาการจัดลำดับการผลิตรถยนต์แบบมากวัตถุประสงค์บนสายการประกอบสองด้าน ซึ่งถูกจัดเป็นปัญหา MaOPs และปัญหาประเภทเอ็นพียาก (Non-deterministic Polynomial Hard; NP-Hard) เนื่องจากมีความซับซ้อนและจำนวนคำตอบที่มาก โดยมีวัตถุประสงค์ที่ถูกประเมินพร้อมกัน 5 วัตถุประสงค์ ได้แก่ จำนวนครั้งการเปลี่ยนแปลงสีน้อยที่สุด จำนวนรถยนต์ที่ละเมิดร่น้อยที่สุด ปริมาณงานที่ไม่เสร็จในการผลิตน้อยที่สุด เวลารอคอยงานรวมในการผลิตน้อยที่สุด และความแปรผันรวมของสัดส่วนการผลิตน้อยที่สุด ซึ่งจากการทดลองพบว่า MOEA/D มีสมรรถนะด้านการเข้าสู่คำตอบที่แท้จริงและเวลาที่ใช้ดีกว่า MODE

**คำสำคัญ:** การจัดลำดับการผลิตรถยนต์บนสายการประกอบผลิตภัณฑ์ผสมแบบสองด้าน, การจัดลำดับแบบมากวัตถุประสงค์, วิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์โดยยึดหลักการจำแนก, ปัญหาแบบมากวัตถุประสงค์

## Many-Objective Car Sequencing Problem on Mixed-model Two-sided Assembly Lines

Tirrat Kirdphoksap and Parames Chutima\*

Department of Industrial Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand

\* Corresponding Author, Tel. 08-0449-8844, E-mail: parames.c@chula.ac.th DOI: 10.14416/j.kmutnb.2018.06.010

Received 16 October 2017; Accepted 6 February 2018; Published online: 21 June 2018

© 2018 King Mongkut's University of Technology North Bangkok. All Rights Reserved.

### Abstract

A Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) is an evolutionary metaheuristic which has been developed to solve Many-Objective Optimization Problems (MaOPs). The concept of searching for answers is to decompose the original problem into subproblems for finding the optimal solution of each subproblem. This study proposes a MOEA/D compared with a Multi-Objective Differential Evolution algorithm (MODE) in order to solve many-objective car sequencing problem on two-sided assembly lines which is classified as a MaOPs and be non-deterministic polynomial hard problem (NP-Hard problem) due to complexity and a large number of solutions. Five objectives were simultaneously evaluated, including the minimal number of color changes, of vehicle violations, along with the minimum amount of unfinished production, of total idle time of the line, and of total variation in rate of production. The experiments showed that MOEA/D has better aspects of convergence performance and time consumption than MODE counterpart.

**Keywords:** Car Sequencing Problem on Mixed-model Two-sided Assembly Lines, Many-Objective Sequencing, A Multi-Objective Evolutionary Algorithm based on Decomposition; MOEA/D, Many-Objective Optimization Problems

## 1. บทนำ

ในปัจจุบันภาคอุตสาหกรรมรถยนต์เกิดการเติบโต และการแข่งขันกันอย่างมาก การพัฒนารูปแบบของรถยนต์ให้ตอบสนองต่อความต้องการของลูกค้า ซึ่งมีความหลากหลาย และเปลี่ยนแปลงตามกระแสนิยมของยุคปัจจุบัน ส่งผลให้ผู้ผลิตไม่สามารถเลือกผลิตเฉพาะรุ่นได้ ภาคอุตสาหกรรมรถยนต์จึงมีความจำเป็นที่จะต้องปรับปรุงและพัฒนาประสิทธิภาพของสายการผลิต รวมถึงนโยบายส่งเสริมการตลาดอยู่ตลอดเวลา ดังนั้นเพื่อรองรับความต้องการของลูกค้า สายการประกอบจึงต้องมีความยืดหยุ่นและสามารถปรับเปลี่ยนประเภทของผลิตภัณฑ์บนสายการผลิตเดียวกันได้ สายการประกอบจึงถูกออกแบบสำหรับหลายผลิตภัณฑ์ที่เข้ามาพร้อมๆ กัน และมีความต่อเนื่องตลอดเวลา [1] เช่นเดียวกับอุตสาหกรรมรถยนต์ เนื่องจากในแต่ละวันมีคำสั่งซื้อรถยนต์ทั้งรุ่นและสีที่แตกต่างกัน โดยที่สายการประกอบรถยนต์แบ่งออกเป็น 3 ส่วน คือ ส่วนของโครงรถยนต์ (Body) ส่วนของการพ่นสี (Paint) และส่วนของการประกอบ (Assembly) อีกทั้งรถยนต์แต่ละรุ่นมีลักษณะเฉพาะของตัวเลือก (Option) ที่แตกต่างกัน ดังนั้นการติดตั้งชุดของตัวเลือก จะต้องออกแบบมาเพื่อจัดการจำนวนรถยนต์ไม่ให้เกินกว่าจำนวนตัวของแต่ละตัวเลือกในลำดับการผลิตย่อยของแต่ละรอบการผลิต [2]

การจัดลำดับการผลิตบนสายการประกอบผลิตภัณฑ์ผสมในภาคอุตสาหกรรม ทำให้เกิดประสิทธิภาพของการผลิตสูงสุดตามวัตถุประสงค์ที่โรงงานอุตสาหกรรมต้องการ โดยที่วัตถุประสงค์จะแตกต่างกันออกไปขึ้นอยู่กับลักษณะของอุตสาหกรรมนั้นๆ อาทิเช่น เวลาในการปรับตั้งน้อยที่สุด ปริมาณงานที่ไม่เสร็จในการผลิตน้อยที่สุด ความสม่ำเสมอของการใช้ชิ้นส่วนประกอบ ต้นทุนต่อเวลาในการปรับตั้งเครื่องน้อยที่สุด อัตราความผันแปรของการผลิตน้อยที่สุด ความเสี่ยงในการหยุดของสายการประกอบมีค่าน้อยที่สุด ฯลฯ [3] ซึ่งความเป็นจริงนั้น การจัดลำดับการผลิตแบบวัตถุประสงค์เดียวมีความเป็นไปได้น้อย เนื่องจากในอุตสาหกรรมจริงมีหลายวัตถุประสงค์ที่ต้องการจะปรับปรุงประสิทธิภาพไปพร้อมๆ กัน ในข้อจำกัดความขัดแย้งกันของแต่ละวัตถุประสงค์ จึงจำเป็นต้องแก้ไขปัญหารูปของ

การจัดลำดับการผลิตแบบหลายวัตถุประสงค์ [4]

ปัญหาการจัดลำดับการผลิตแบบหลายวัตถุประสงค์ ถูกจัดเป็นปัญหาประเภทเอ็นพีฮาร์ด (Non-deterministic Polynomial Hard; NP-Hard) เนื่องจากมีความยุ่งยากซับซ้อน ใช้เวลาในการค้นหาคำตอบนาน และจำนวนคำตอบมีจำนวนมาก จึงมีการนำวิธีฮิวริสติกและเมตาฮิวริสติกเข้ามาแก้ปัญหาเพื่อหาคำตอบที่เหมาะสมและยอมรับได้ ด้วยเวลาในการค้นหาคำตอบที่น้อยลง ยกตัวอย่างเช่น เจนเนติกอัลกอริทึม (Genetic Algorithms; GA) ซึ่งเป็นเมตาฮิวริสติกที่นิยมนำมาใช้แก้ปัญหาการจัดลำดับการผลิต เพราะเจนเนติกอัลกอริทึมให้คำตอบที่ใกล้เคียงคำตอบที่ดีที่สุด (Optimal Solution) และใช้เวลาในการค้นหาคำตอบน้อย Mansouri [5] ได้เสนอวิธีแก้ปัญหาการจัดลำดับแบบหลายวัตถุประสงค์ (Multi-Objective Genetic Algorithm; MOGA) ด้วยวิธี COMSOAL ซึ่งเป็นอัลกอริทึมที่ใช้เวลาในการค้นหาคำตอบน้อย มีขั้นตอนในการดำเนินงานที่ไม่ซับซ้อนมาก และสามารถนำไปใช้ในงานจริงได้ดีกว่าเมื่อเทียบกับอัลกอริทึมอื่นๆ Chutima and Olarnvichai [6] ได้นำวิธีการบรรจบแบบขยาย (Combinatorial Optimization with Coincidence Expand; COIN-E) มาใช้ในการแก้ปัญหาการจัดลำดับการผลิตรถยนต์แบบหลายวัตถุประสงค์ พบว่า COIN-E สามารถค้นหาคำตอบที่มีประสิทธิภาพดีกว่า COIN และยังใช้เวลาในการหาคำตอบที่เร็วกว่าอัลกอริทึมอื่นๆ

ปัญหาจำนวนมากในอุตสาหกรรมจริงรวมถึงการจัดลำดับการผลิตรถยนต์แบบมากวัตถุประสงค์ จะมีจำนวนวัตถุประสงค์มากกว่า 3 วัตถุประสงค์ ซึ่งอาจขัดแย้งกัน และต้องพิจารณาพร้อมกัน เรียกปัญหานี้ว่าปัญหาแบบมากวัตถุประสงค์ (Many-Objective Optimization Problems; MaOPs) เนื่องจากเป็นปัญหาที่มีมิติของแกนเวกเตอร์หรือจำนวนวัตถุประสงค์ที่มาก ทำให้ประสิทธิภาพในการค้นหาคำตอบของวิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ (A Multi-Objective Evolutionary Algorithms; MOEAs) ลดลง และไม่สามารถแก้ปัญหา MaOPs ได้ [7]

MOEAs ส่วนใหญ่จะพิจารณาหาคำตอบโดยรวมด้วยการจัดอันดับของคำตอบที่ไม่ถูกครอบงำหรือเรียกว่า

วิธีเชิงกลุ่มที่ดีที่สุด (Pareto-based Approach) วิธีการนี้มีความซับซ้อน ใช้เวลามาก และยากที่จะเกิดสมดุลของความหลากหลายรวมถึงการลู่เข้าของคำตอบ [8] เนื่องจากจำนวนวัตถุประสงค์ที่มาก ทำให้การคัดเลือกคำตอบเกิดคำตอบที่เป็นคำตอบที่ดีที่สุด (1st frontier) จำนวนมากและอยู่รวมเป็นกระจุก ส่งผลให้ทิศทางของการพัฒนาคำตอบติดอยู่ในคำตอบที่เฉพาะกลุ่ม (Local Optima)

การแก้ปัญหา MaOPs อย่างง่ายที่สุด ทำได้ด้วยการลดจำนวนวัตถุประสงค์ลง โดยการรวมวัตถุประสงค์ที่ไม่ขัดแย้งกันไว้เป็นเป้าหมายเดียว [9] อย่างไรก็ตาม วิธีการนี้ไม่สามารถใช้กับหลายปัญหาในทางปฏิบัติได้ ด้วยเหตุนี้วิธีการแบบใหม่จึงถูกพัฒนาขึ้น เรียกว่าวิธีการเชิงวิวัฒนาการแบบมากวัตถุประสงค์ (A Many-Objective Evolutionary Algorithm; MaOEA) โดยหนึ่งในอัลกอริทึมที่โดดเด่นของ MaOEAs คือวิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์โดยยึดหลักการจำแนก (A Multi-Objective Evolutionary Algorithm based on Decomposition; MOEA/D) ซึ่งถูกพัฒนาโดย Qingfu and Hui [10] วิธีการดำเนินงานของ MOEA/D จะทำการจำแนกปัญหา MaOP ออกเป็นปัญหาย่อยและทำการแก้ปัญหาย่อยไปพร้อมกัน แต่ละปัญหาย่อยจะได้คำตอบที่ดีที่สุด

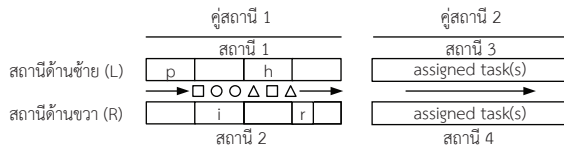
การประยุกต์ใช้อัลกอริทึมในการแก้ปัญหาแบบมากวัตถุประสงค์เป็นการประมาณค่าคำตอบ ซึ่งจำเป็นต้องเลือกคำตอบสุดท้ายที่จะนำไปดำเนินการ MaOPs ส่วนใหญ่มีจำนวนคำตอบมากและไม่รู้ขอบเขตที่แน่ชัด อีกทั้งยังใช้เวลามากในการค้นหาคำตอบ MOEA/D ซึ่งเป็นอัลกอริทึมที่โดดเด่นของ MaOEA จึงเป็นทางเลือกที่ดี เนื่องจากมีการกำหนดขอบเขตที่ชัดเจน ความซับซ้อนและเวลาในการค้นหาคำตอบน้อยกว่าอัลกอริทึมเชิงพันธุกรรมแบบการจัดลำดับที่ไม่ถูกครอบงำ II (Non-dominated Sorting Genetic Algorithm II; NSGA II) อย่างชัดเจน อีกทั้งยังคงคำตอบที่หลากหลาย โดย MOEA/D จะเห็นผลลัพธ์ที่ดีและแตกต่างกับอัลกอริทึมอื่นๆ เมื่อมีวัตถุประสงค์ตั้งแต่ 4 ถึง 15 วัตถุประสงค์ [10]

ดังนั้นในงานวิจัยนี้ จึงนำ MOEA/D มาประยุกต์ในการ

แก้ปัญหาการจัดลำดับการผลิตรถยนต์แบบมากวัตถุประสงค์ ซึ่งมีวัตถุประสงค์ในการจัดลำดับการผลิตรถยนต์จำนวนทั้งสิ้น 5 วัตถุประสงค์ ดังนี้ 1) จำนวนครั้งการเปลี่ยนแปลงสีน้อยที่สุด (Minimizing the Number of Color Change) 2) จำนวนรถยนต์ที่ละเมิดร่น้อยที่สุด (Minimizing the Number of Violations) 3) ปริมาณงานที่ไม่สำเร็จในการผลิตน้อยที่สุด (Minimizing Utility Work) 4) เวลารอคอยงานรวมในการผลิตน้อยที่สุด (Minimizing Total Idle Time) และ 5) ความแปรผันรวมของสัดส่วนการผลิตน้อยที่สุด (Minimizing the Total Production Rate Variation) โดยคัดเลือกคำตอบที่ดีที่สุดด้วยวิธีเทบปีเชฟฟ์ (Tchebycheff Approach) เพื่อให้ได้คำตอบที่ดีที่สุดของแต่ละปัญหาย่อย และนำคำตอบที่ได้จาก MOEA/D มาคัดเลือกด้วยวิธีเชิงกลุ่มที่ดีที่สุด เพื่อจัดอันดับความแข็งแกร่งกับสตรีงคำตอบและสามารถเปรียบเทียบกับอัลกอริทึมอื่นๆ ได้ ซึ่งการเปรียบเทียบประสิทธิภาพของอัลกอริทึม จะพิจารณาจากคำตอบที่แต่ละอัลกอริทึมหาได้เทียบกับกลุ่มคำตอบที่ดีที่สุดหรือเรียกว่ากลุ่มคำตอบที่แท้จริง (True-Pareto Optimal Frontier) เพื่อหาว่าอัลกอริทึมใดที่มีประสิทธิภาพในการแก้ปัญหาได้ดีกว่า

## 2. สายการประกอบผลิตภัณฑ์ผสมแบบสองด้าน

สายการประกอบแบบสองด้าน (Two-Sided Assembly Line) เป็นสายการประกอบซึ่งถูกออกแบบสำหรับผลิตภัณฑ์ที่มีขนาดใหญ่และจำนวนการผลิตมาก เช่น รถยนต์ และรถบรรทุก โดยสถานีนงานจะอยู่ทั้งสองด้านของสายการประกอบ ได้แก่ สถานีงานด้านซ้าย (L) และสถานีงานด้านขวา (R) เรียกสถานีงานที่อยู่คู่กันว่า คู่สถานีงาน (Mated Station) [11] โดยจะแบ่งประเภทชิ้นงานที่จัดสรรลงในสถานีงานเป็น 2 แบบ ดังนี้ 1) ชิ้นงานที่จัดสรรลงในสถานีงานได้เพียงด้านเดียว ได้แก่ ชิ้นงานที่จัดสรรลงในสถานีงานด้านซ้าย (L-Type) และชิ้นงานที่จัดสรรลงในสถานีงานด้านขวา (R-Type) และ 2) ชิ้นงานที่จัดสรรลงในสถานีงานได้ทั้งสองด้าน (E-Type) ข้อดีของสายการประกอบนี้คือ จะมีความยาวน้อยกว่าสายการประกอบแบบด้านเดียวและเกิดความสูญเปล่าของระยะเวลาเดินของพนักงานรวมถึงเวลาการผลิตที่น้อยลง



รูปที่ 1 สายการประกอบผลิตภัณฑ์ผสมแบบสองด้าน

สายการประกอบผลิตภัณฑ์ผสม (Mixed-model Assembly Line) เป็นสายการประกอบซึ่งผลิตผลิตภัณฑ์ตั้งแต่ 2 ชนิดขึ้นไป เช่นเดียวกับสายการประกอบแบบหลายผลิตภัณฑ์ สายการประกอบประเภทนี้จะผลิตผลิตภัณฑ์ที่มีส่วนประกอบพื้นฐานเหมือนกัน แต่แตกต่างกันในส่วน of ตัวเลือก (Option) ของแต่ละผลิตภัณฑ์ งานแต่ละชนิดสามารถเข้าสู่สายการผลิตได้อย่างอิสระ ไม่มีข้อกำหนดว่าต้องทำผลิตภัณฑ์ใดก่อนหรือหลัง และระหว่างการผลิตไม่มีการปรับตั้งสายการผลิต แสดงสายการประกอบผลิตภัณฑ์ผสมแบบสองด้าน ดังรูปที่ 1

เมื่อ  $\{p, h\}$  เป็นชิ้นงาน L-Type,  $\{i, r\}$  เป็นชิ้นงาน R-Type และรูปทรงเรขาคณิตแทนผลิตภัณฑ์รุ่นต่างๆ

### 3. ปัญหาการจัดลำดับการผลิตรถยนต์

การผลิตรถยนต์เป็นการผลิตตามคำสั่งซื้อ (Make to Order) จากลูกค้าในแต่ละวัน ดังนั้นคำสั่งซื้อจึงต้องนำมาวางแผนจัดลำดับการผลิตลงแต่ละวัน

ในงานวิจัยฉบับนี้ ได้พิจารณาฟังก์ชันวัตถุประสงค์ 5 วัตถุประสงค์ โดยให้ค่าน้ำหนักความสำคัญเท่ากันทั้งหมด ดังนี้

#### 3.1 จำนวนครั้งของการเปลี่ยนแปลงสีน้อยที่สุด

ลำดับสีของรถยนต์ที่ผลิต มีผลอย่างมากต่อเวลาและต้นทุนในการพ่นสีรถแต่ละคัน เพราะถ้าลำดับการผลิตรถยนต์มีสีเดียวติดต่อกัน ก็ไม่จำเป็นต้องทำความสะอาดหัวปืนพ่นสีและเปลี่ยนน้ำยาในการพ่น แต่หากพ่นสีเดียวติดต่อกันมากเกินไปที่กำหนด ก็มีความจำเป็นต้องล้างหัวปืนพ่นสี เนื่องจากหัวปืนจะมีการอุดตันและสีที่พ่นจะไม่มีคุณภาพ [6] แสดงสูตรคำนวณดังสมการที่ (1) เมื่อ  $d_m$  คือสัดส่วนการผลิต

รถยนต์รุ่นที่  $m$  โดยที่  $m \in \{1, 2, \dots, M\}$   $nc$  คือผลรวมสัดส่วนการผลิตหรือจำนวนรถที่ผลิตทั้งหมด  $k$  คือตำแหน่งใดๆ ในลำดับการผลิต โดยที่  $k \in [0, nc]$  และ  $\delta_{COL_{k,k+1}}$  คือการเปลี่ยนสีพ่น ณ ตำแหน่งที่  $k$  และ  $k+1$  (เท่ากับ 1 เมื่อมีการเปลี่ยนสี และเท่ากับ 0 เมื่อไม่มีการเปลี่ยนสี)

$$f_1(x) = \sum_{k=0}^{nc-1} \delta_{COL_{k,k+1}} \quad (1)$$

ข้อมูลปัญหาตัวอย่างในตารางที่ 1 และ 2 มีรถที่ผลิตทั้งหมด 9 คัน 3 รุ่น 5 สี และตัวอย่างการจัดลำดับ (จำนวนพ่นสีเดียวกันติดต่อกันสูงสุด 2 สี) ในตารางที่ 3 จะเห็นว่าตำแหน่งที่ 0 ไป 1 เปลี่ยนสีจากสี 2 ไปสี 1 ดังนั้นเกิดการเปลี่ยนสีในลำดับการผลิตที่ 1 โดยในตัวอย่างนี้เกิดจำนวนการเปลี่ยนแปลงสีทั้งหมด 7 ครั้ง

ตารางที่ 1 สีที่รถยนต์แต่ละรุ่นมี

สี	จำนวนรถยนต์สีเดียวกันติดต่อกันได้ไม่เกิน (คัน)	รุ่นของรถยนต์		
		A	B	C
1	2	1	1	0
2	2	1	1	1
3	2	0	1	0
4	2	0	1	0
5	2	0	0	1

ตารางที่ 2 สีจากการสุ่มของรุ่นรถยนต์

รุ่น	A	A	B	B	B	C	C	C	C
Priority	1	2	3	4	5	6	7	8	9
สี	1	2	2	1	4	5	2	2	5

#### 3.2 จำนวนการละเมิดตัวเลือกรวมน้อยที่สุด

รถยนต์แต่ละรุ่นมีตัวเลือกที่แตกต่างกัน ดังนั้นการจัดลำดับการผลิตจำเป็นต้องพิจารณาจำนวนรถยนต์ที่มีตัวเลือกไม่ให้เกิดติดกันเกินกว่าสัดส่วนที่กำหนด เพราะจะส่งผลกระทบต่อประสิทธิภาพของพนักงาน ทำให้ค่าเฉลี่ยเวลาที่ใช้ในการผลิตมากเกินกว่ารอบเวลาผลิต โดยพิจารณาจากอัตราส่วน  $p_o : q_o$

(เมื่อ  $p_o$  คือจำนวนสูงสุดในการเรียงรุ่นรถที่มีตัวเลือก  $o$  และ  $q_o$  คือจำนวนรถที่เรียงติดกัน) ตัวอย่างเช่น ขั้นตอนการประกอบไฟเลี้ยงกระจกมองข้างมีอัตราส่วน  $p_o : q_o$  เท่ากับ 3 ต่อ 4 หมายความว่าในการผลิตรถยนต์ทุกๆ 4 คัน จะมีรถยนต์ที่ประกอบตัวเลือกนี้ได้ไม่เกิน 3 คัน หากเกิดจะนับเป็นการละเมิดเกิดขึ้น แสดงสูตรคำนวณดังสมการที่ (2) [6]

$$f_2(x) = \sum_{o \in O} \left\{ NPV_o \cup_{i=1}^{nc} PV_{o,i}(i - q_o + 1, \dots, i) \right\} \quad (2)$$

เมื่อ  $PV_{o,i}(i - q_o + 1, \dots, i)$  คือ การนับจำนวนที่เกิดการละเมิดของตัวเลือก  $o$  ในตำแหน่งที่  $i$  โดยจะมีค่าเท่ากับ 1 เมื่อตำแหน่งที่  $i$  มีการใช้ตัวเลือก  $o$  และตำแหน่งที่  $i - q_o + 1$  ถึง  $i$  เกิดการละเมิดตัวเลือกที่อัตราส่วน  $p_o : q_o$  และเท่ากับ 0 เมื่อไม่เกิดการละเมิด,  $\cup$  คือ Union Operator และ  $NPV_o$  คือ เซตของการละเมิดที่เกิดขึ้นบนตัวเลือก  $o$

ข้อมูลปัญหาตัวอย่างในตารางที่ 4 มีทั้งหมด 4 ตัวเลือกรถยนต์ 3 รุ่น และตัวอย่างการหาจำนวนการละเมิดดังตารางที่ 3 โดยเกิดจำนวนละเมิดในตัวอย่างนี้  $o_1 = 7, o_2 = 3, o_3 = 5$  และ  $o_4 = 3$

ตารางที่ 3 ตัวอย่างลำดับการผลิต

วัน	$j - 1$	$j$									
ตำแหน่ง	-1	0	1	2	3	4	5	6	7	8	9
ลำดับ	C	C	A	B	C	B	B	C	C	C	A
สี	5	2	1	4	2	1	2	2	5	5	2
$o_1$	1	1	0	1	1	1	1	1	1	1	0
$o_2$	1	1	1	0	1	0	0	1	1	1	1
$o_3$	1	1	1	0	1	0	0	1	1	1	1
$o_4$	0	0	1	1	0	1	1	0	0	0	1

ตารางที่ 4 ตัวเลือกของรถยนต์แต่ละรุ่น

ตัวเลือก	อัตราส่วนข้อกำหนด (p/q)	รุ่นของรถยนต์		
		A	B	C
1	1/3	0	1	1
2	2/3	1	0	1
3	1/3	1	0	1
4	1/3	1	1	0

### 3.3 ปริมาณงานที่ไม่เสร็จจน้อยที่สุด

ปริมาณงานที่ไม่เสร็จในการผลิต คือผลรวมของปริมาณงานที่ไม่เสร็จทุกผลิตภัณ์ ณ คู่สถานีงาน  $n_m$  โดยเวลาที่ต้องทำงานที่ยังไม่เสร็จ จะส่งผลให้สายการประกอบต้องหยุดดำเนินงานเพื่อทำงานปัจจุบันให้เสร็จสิ้นก่อน จึงจะสามารถดำเนินงานประกอบรถยนต์ลำดับผลิตถัดไปได้ [12] โดยในส่วนแรกของการสมการฟังก์ชันวัตถุประสงค์เป็นเวลาที่เกิดขึ้นจากรอบเวลาการผลิต และส่วนที่สองเป็นเวลาในการปรับสายพานลำเลียงกลับสำหรับการผลิตในรอบถัดไป แสดงสูตรดังสมการที่ (3)

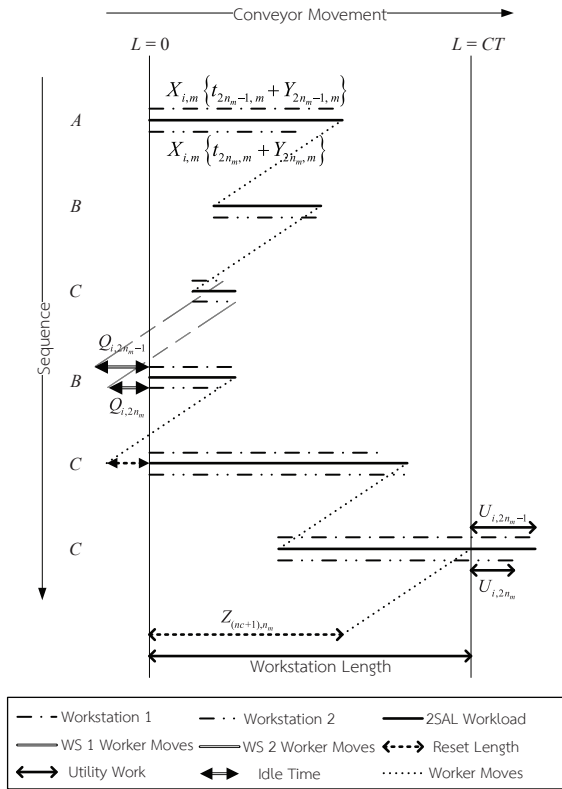
$$f_3(x) = \sum_{n_m=1}^{N_M} \left( \sum_{i=1}^{nc} U_{i,n_m} + \frac{Z_{(nc+1),n_m}}{v_c} \right) \quad (3)$$

โดยที่

$$U_{i,n_m} = \left[ \begin{array}{l} \max \left[ 0, \frac{\left( Z_{i,n_m} + v_c \sum_{m=1}^M X_{i,m} \{ t_{2n_m-1,m} + Y_{2n_m-1,m} \} - L_{n_m} \right)}{v_c} \right] \\ + \max \left[ 0, \frac{\left( Z_{i,n_m} + v_c \sum_{m=1}^M X_{i,m} \{ t_{2n_m,m} + Y_{2n_m,m} \} - L_{n_m} \right)}{v_c} \right] \end{array} \right]$$

$$Z_{(i+1),n_m} = \max \left\{ \begin{array}{l} \max \left[ 0, \min \left( Z_{i,n_m} + v_c \sum_{m=1}^M X_{i,m} \{ t_{2n_m-1,m} + Y_{2n_m-1,m} \} - \gamma v_c, L_{n_m} - \gamma v_c \right) \right] \\ \max \left[ 0, \min \left( Z_{i,n_m} + v_c \sum_{m=1}^M X_{i,m} \{ t_{2n_m,m} + Y_{2n_m,m} \} - \gamma v_c, L_{n_m} - \gamma v_c \right) \right] \end{array} \right\}$$

เมื่อ  $\gamma$  คือ ช่วงเวลาการปล่อยผลิตภัณ์เข้าสู่สายการผลิต,  $t_{j,m}$  คือ เวลาการดำเนินงานทั้งหมดของผลิตภัณ์  $m$  ที่สถานี  $j$ ,  $Y_{j,m}$  คือ เวลาเดินเปล่าทั้งหมดของผลิตภัณ์  $m$  ที่สถานี  $j$  ซึ่งขึ้นอยู่กับเวลาเสร็จสิ้นของงานลำดับก่อนหน้า,  $v_c$  คือ อัตราเร็วของสายการผลิต,  $CT$  คือ รอบเวลาการผลิต (Cycle Time),  $L_{n_m}$  คือ ความยาวของคู่สถานีงาน  $n_m$  ( $L_{n_m} = v_c \times CT$ ),  $U_{i,n_m}$  คือ ปริมาณงานที่ไม่เสร็จในลำดับผลิตที่  $i$  ในคู่สถานี  $n_m$  (หน่วยเวลา),  $X_{i,m}$  คือ ตัวแปรเลขฐานสอง เท่ากับ 1 เมื่อมีรถยนต์รุ่น  $m$  ในลำดับการผลิตที่  $i$  และเท่ากับ 0 เมื่อ



รูปที่ 2 ปริมาณงานที่ทำไม่เสร็จและเวลารอคอยงาน

ไม่มี และ  $Z_{i,n_m}$  คือ ระยะทางจากจุด  $L_{n_m}$  ที่เท่ากับ 0 ถึงจุด  $L_{n_m}$  ใดๆ ที่เริ่มงานในลำดับการผลิตที่  $i$  ในคู่สถานี  $n_m$  แสดงกราฟการคำนวณดังรูปที่ 2

### 3.4 เวลารอคอยงานรวมน้อยที่สุด

ฟังก์ชันวัตถุประสงค์นี้สนใจเวลาที่พนักงานต้องรอคอยงานถัดไปที่กำลังเข้าสู่สถานี หากเกิดเวลารอคอยมากจะส่งผลให้พนักงานมีอรรถประโยชน์ (Utilization) ที่ต่ำ โดยเวลาทั้งหมดที่รอคอยงานเข้าสู่คู่สถานีงานจะเป็นผลรวมเวลารอคอยของพนักงานในคู่สถานี  $n_m$  [13] แสดงสูตรคำนวณดังสมการที่ (4)

$$f_4(x) = \frac{1}{v_c} \sum_{n_m=1}^{N_M} \sum_{i=1}^{nc} Q_{i,n_m} \quad (4)$$

โดยที่

$$Q_{(i+1),n_m} = \left\{ \begin{aligned} & \max \left[ 0, \gamma v_c - \left( Z_{i,n_m} + v_c \sum_{m=1}^M X_{i,m} \{ t_{2n_m-1,m} + Y_{2n_m-1,m} \} \right) \right] \\ & + \max \left[ 0, \gamma v_c - \left( Z_{i,n_m} + v_c \sum_{m=1}^M X_{i,m} \{ t_{2n_m,m} + Y_{2n_m,m} \} \right) \right] \end{aligned} \right\}$$

เมื่อ  $Q_{i,n_m}$  คือระยะทางก่อนที่รถยนต์ลำดับการผลิตที่  $i$  จะเข้าสู่สถานีงาน  $n_m$  แสดงกราฟการคำนวณดังรูปที่ 2

### 3.5 ความแปรผันรวมสัดส่วนการผลิตน้อยที่สุด

ความแปรผันรวมของสัดส่วนการผลิต เกิดจากการจัดลำดับของผลิตภัณฑ์แต่ละรุ่นลงในสายการผลิต หากรุ่นใดมีการกระจายตัวเท่าๆ กัน ก็จะทำให้ความแปรผันของสัดส่วนการผลิตต่ำ แต่หากรุ่นใดมีลำดับที่เรียงติดกันหรือห่างกัน ก็จะส่งผลให้ความแปรผันของสัดส่วนการผลิตสูง วัตถุประสงค์นี้มีความสำคัญกับระบบการผลิตแบบทันเวลาพอดี (Just in Time System) เพื่อรองรับความต้องการรถยนต์แต่ละรุ่นซึ่งทราบจำนวนที่ชัดเจนในแต่ละวัน [14] แสดงสูตรคำนวณดังสมการที่ (5)

$$f_5(x) = \sum_{i=1}^{nc} \sum_{m=1}^M \left| \left( \sum_{l=1}^i \frac{X_{l,m}}{i} \right) - \frac{d_m}{nc} \right| \quad (5)$$

ส่วนแรกของสมการฟังก์ชันวัตถุประสงค์เป็นสัดส่วนของการผลิตของรถยนต์รุ่นที่  $m$  ตั้งแต่ลำดับแรกจนถึงลำดับที่ถูกผลิต (ลำดับที่  $i$ ) และส่วนที่สองเป็นสัดส่วนความต้องการรถยนต์รุ่นที่  $m$

### 4. การหาค่าเหมาะสมที่สุดที่มีหลายวัตถุประสงค์

การแก้ปัญหาที่มีหลายวัตถุประสงค์ให้เกิดประสิทธิภาพที่ดีพร้อมกันในทุกๆ ด้านนั้น เป็นไปได้ยาก เนื่องจากแต่ละวัตถุประสงค์อาจมีความขัดแย้งอย่างหลีกเลี่ยงไม่ได้ ทำให้ปัญหาที่มีหลายวัตถุประสงค์ไม่สามารถบอกได้ว่าคำตอบใดเป็นคำตอบที่ดีที่สุด [15]

ในงานวิจัยนี้เป็นการหาค่าที่เหมาะสมที่สุดสำหรับปัญหาที่มีหลายวัตถุประสงค์ โดยหาค่าวัตถุประสงค์ทุกตัว

ที่น้อยที่สุด และพิจารณาไปพร้อมกันทั้งหมดด้วยวิธีดังต่อไปนี้

1. วิธีเชิงกลุ่มที่ดีที่สุด ดังสมการที่ (6)

$$\text{Minimize} \{f_1(\bar{x}), f_2(\bar{x}), \dots, f_k(\bar{x})\} \quad (6)$$

เมื่อ  $\bar{x}_j$  คือ เวกเตอร์ตัวแปรตัดสินใจของสตรึง  $j$  โดยที่  $\bar{x}_j = \{x_1, \dots, x_{nc}\}$

$f_k(\bar{x})$  คือ ฟังก์ชันวัตถุประสงค์ที่  $k$  ของ  $\bar{x}$

ในกรณีหาค่าคำตอบที่น้อยที่สุด เวกเตอร์ของตัวแปรตัดสินใจ  $\bar{x}$  จะให้คำตอบที่ดีกว่าตัวแปรตัดสินใจ  $y$  ก็ต่อเมื่อ  $f_k(\bar{x}) \leq f_k(y)$  ทุกค่า และ  $f_k(\bar{x}) < f_k(y)$  อย่างน้อย 1 ค่า โดยที่  $k \in \{1, 2, \dots, \text{Objective}\}$

2. วิธีเทบปีเซฟฟ์ ดังสมการที่ (7)

$$\min g^e(\bar{x}_j | \bar{\lambda}_j, z^*) = \max_{1 \leq k \leq Obj} \{ \lambda_{jk} | f_k(\bar{x}_j) - z_k^* \} \quad (7)$$

เมื่อ  $g^e$  คือ ผลต่างมากที่สุดระหว่างค่าวัตถุประสงค์ของสตรึง  $j$  กับค่าเป้าหมายของค่าน้ำหนัก  $\lambda_{jk}$  ของปัญหาย่อย  $j$

$z^*$  คือ เวกเตอร์ค่าเป้าหมาย โดยที่  $z^* = \{z_1^*, \dots, z_{Obj}^*\}$

$\bar{\lambda}_j$  คือ เวกเตอร์ค่าน้ำหนักของปัญหาย่อย  $j$  โดยที่  $\bar{\lambda}_j = \{\lambda_{j,1}, \dots, \lambda_{j,Obj}\}$

การค้นหาคำตอบของเทบปีเซฟฟ์จะเลื่อนเข้าใกล้ค่าเป้าหมายทีละแกนด้วยผลต่างของค่าฟังก์ชันวัตถุประสงค์เทียบกับค่าเป้าหมาย เมื่อได้ค่าผลต่างที่มากที่สุดของคำตอบนั้นๆ แล้ว จะทำการแทนที่คำตอบที่มีผลต่างที่น้อยที่สุดเพื่อหาคำตอบที่ดีต่อไป

## 5. อัลกอริทึม MOEA/D

การจัดลำดับการผลิตรถยนต์แบบมากวัตถุประสงคบนสายการประกอบผลิตภัณฑ์ผสมแบบสองด้านจะต้องนำเข้าข้อมูลเพื่อใช้ในการแก้ปัญหา ได้แก่ จำนวนของรถยนต์แต่ละรุ่น สัดส่วนรถยนต์ ลำดับความสัมพันธ์ก่อนหลังของชิ้นงาน เวลาดำเนินงานของแต่ละชิ้นงานในแต่ละรุ่นรถยนต์ สีของรถยนต์แต่ละรุ่น สีของรถยนต์ในรอบการผลิตก่อนหน้า รุ่นของรถยนต์ในรอบการผลิตก่อนหน้า และสายการประกอบ

ผลิตภัณฑ์ผสมแบบสองด้านที่ผ่านการจัดสมดุลแล้ว

MOEA/D มีหลักแนวคิดในการปัญหาด้วยการวิวัฒนาการคำตอบที่มีไปพร้อมๆ กัน โดยเริ่มจากความสัมพันธ์ของสตรึงข้างเคียงหรือปัญหาย่อยข้างเคียง (Neighborhood) ซึ่งใช้ระยะห่างระหว่างผลรวมค่าสัมประสิทธิ์ของเวกเตอร์ ( $\bar{\lambda}_j$ ) ในแต่ละฟังก์ชันวัตถุประสงค์เป็นตัวเลือกสตรึงคำตอบที่จะนำมาสุ่มเพื่อพัฒนาคำตอบ การพัฒนาคำตอบในประชากร (Population) แต่ละตัว ใช้การครอสซิงโอเวอร์ (Crossing Over) ของวิธีเชิงพันธุกรรม (GA) เพื่อให้ได้คำตอบรุ่นใหม่ที่มีความคล้ายคลึงกับคำตอบรุ่นเดิมที่เป็นค่าที่ดี ทำให้คำตอบถูกพัฒนาต่อทุกๆ สตรึงคำตอบ

การทำงานของ MOEA/D จะนำสตรึงคำตอบจากทุกเจเนอเรชัน (Generation) มาคัดเลือกหากกลุ่มคำตอบที่ดีที่สุด โดยแต่ละเจเนอเรชันประกอบไปด้วยประชากร (สตรึงเริ่มต้น) ตามจำนวนที่กำหนด ซึ่งจะถูกทำการพัฒนาคำตอบจากการสุ่มมาสองสตรึงจากสตรึงข้างเคียงของสตรึงหลัก จากนั้นประเมินค่าฟังก์ชันวัตถุประสงค์ของทุกสตรึง หากสตรึงที่ถูกพัฒนาให้คำตอบที่ดีกว่าสตรึงหลักจะนำสตรึงที่ถูกพัฒนาไปแทนที่สตรึงหลักและประเมินสตรึงที่ถูกพัฒนาด้วยค่าน้ำหนักของทุกสตรึง หากให้คำตอบที่ดีกว่า ค่าน้ำหนักใดๆ สตรึงที่ถูกพัฒนาจะถูกนำไปแทนที่สตรึงเดิมที่ค่าน้ำหนักนั้น โดยวนสตรึงหลักตั้งแต่สตรึงที่ 1 จนถึงสตรึงสุดท้ายของประชากร เก็บสตรึงคำตอบเป็นอันจบเจเนอเรชัน แสดงขั้นตอนการทำงานดังรูปที่ 3 และตัวอย่างในลำดับถัดไป

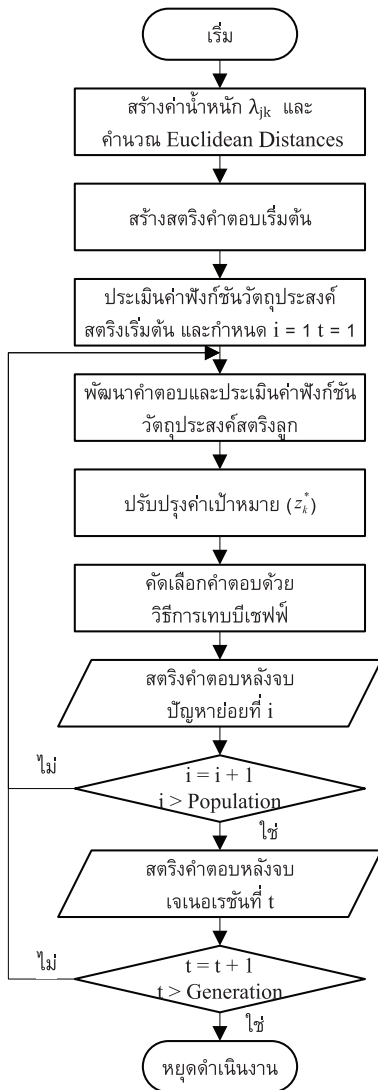
ค่าพารามิเตอร์ของตัวอย่างการประยุกต์ใช้ MOEA/D ดังนี้ 1) จำนวนประชากรเริ่มต้น 10 ตัว 2) จำนวนสตรึงข้างเคียง 3 ตัว และ 3) อัตราการเปลี่ยนแปลงคำตอบ 0.7

ตัวอย่างขั้นตอนการดำเนินงานของ MOEA/D ในการหาคำตอบของปัญหาแบบมากวัตถุประสงค โดยใช้ข้อมูลในตารางที่ 2 และ 3 มีดังนี้

### 5.1 การสร้าง $\bar{\lambda}_j$ และคำนวณ Euclidean Distances

การสร้างตารางค่าน้ำหนักด้วยวิธีการซิมเพล็กซ์แลตทิซ ดีไซน์ (Simplex Lattice Design) ให้กำหนดขนาดเท่ากับจำนวนประชากรคูณกับจำนวนฟังก์ชันวัตถุประสงค์ ( $N \times Obj$ )





รูปที่ 3 ขั้นตอนการทำงาน MOEA/D

โดยในงานวิจัยนี้ ผู้วิจัยได้สร้างตารางค่าน้ำหนักจากโปรแกรม มินิแทบ 17 (Minitab 17) ในหมวดของมิกซ์เจอร์ดีไซน์ (Mixture Design) แสดงดังตารางที่ 5

คำนวณค่า Euclidean Distances สามารถคำนวณได้จากสมการ  $d_{jl} = \sqrt{\sum_{k=1}^{Obj} (\lambda_{jk} - \lambda_{lk})^2} \forall j, l \in N$  เมื่อ *Obj* คือ จำนวนวัตถุประสงค์ และ *N* คือ จำนวนประชากร (จำนวนสตริงคำตอบเริ่มต้น)

เมื่อได้ค่า Euclidean Distances แล้ว ให้ทำการ

กำหนดสตริงข้างเคียง (ปัญหาย่อยข้างเคียง) โดยเรียงค่า  $d_{jl}$  ที่  $\forall j \in N$  จากน้อยไปมากและตัดให้เหลือเท่ากับจำนวนสตริงข้างเคียงที่กำหนดไว้ข้างต้น จะได้สตริงข้างเคียงของแต่ละปัญหาย่อย แสดงดังตารางที่ 6

ตารางที่ 5 ซิมเพล็กซ์แลตทิซดีไซน์

ปัญหาย่อย <i>j</i>	ค่าน้ำหนัก $\lambda_{jk}$ ของแต่ละวัตถุประสงค์				
	1	2	3	4	5
1	0.1	0.1	0.1	0.6	0.1
2	0.6	0.1	0.1	0.1	0.1
3	0	0	1	0	0
4	0.1	0.1	0.1	0.1	0.6
5	0.1	0.6	0.1	0.1	0.1
6	0	1	0	0	0
7	0	0	0	0	1
8	0.1	0.1	0.6	0.1	0.1
9	0	0	0	1	0
10	1	0	0	0	0

ตารางที่ 6 เซตสตริงข้างเคียงของแต่ละปัญหาย่อย

ปัญหาย่อย	สตริงข้างเคียง (ปัญหาย่อยข้างเคียง)		
1	1	9	2
2	2	10	1
3	3	8	1
4	4	7	5
5	5	6	4
6	6	5	2
7	7	4	8
8	8	3	5
9	9	1	8
10	10	2	5

### 5.2 การสร้างสตริงคำตอบเริ่มต้น

การสร้างสตริงคำตอบเริ่มต้นจะใช้วิธีสุ่มเลือกแบบอิสระ โดยสุ่มเท่ากับจำนวนประชากรที่ความยาวสตริงเท่ากับจำนวนผลิตภัณฑ์ทั้งหมด แสดงตัวอย่างสตริงจากการสุ่มดังตารางที่ 7

ตารางที่ 7 สตริงคำตอบเริ่มต้น

ปัญหา ย่อย	สตริง คำตอบ	Priority									
		7	6	4	9	8	2	1	5	3	
1	1	7	6	4	9	8	2	1	5	3	
2	2	9	5	4	8	1	3	6	7	2	
3	3	3	7	2	4	5	9	8	1	6	
4	4	8	9	6	3	1	2	5	4	7	
5	5	5	2	1	6	3	9	8	7	4	
6	6	9	6	7	3	2	8	5	4	1	
7	7	6	7	2	1	4	5	9	3	8	
8	8	7	6	5	3	4	2	8	9	1	
9	9	3	7	8	6	2	1	4	9	5	
10	10	3	9	2	8	5	6	1	7	4	

### 5.3 ประเมินค่าวัตถุประสงค์สตริงคำตอบเริ่มต้น

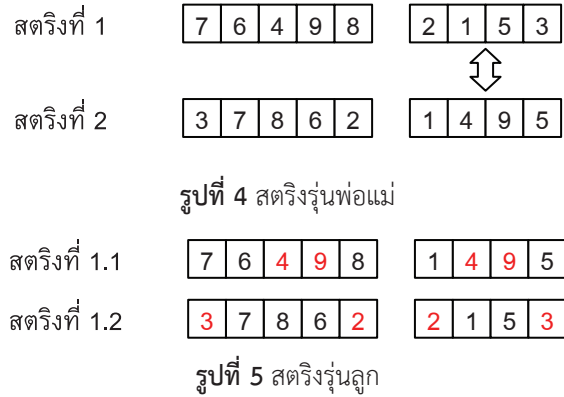
การคำนวณค่าฟังก์ชันวัตถุประสงค์ของแต่ละสตริงในตัวอย่างนี้ สามารถคำนวณได้จากสมการฟังก์ชันวัตถุประสงค์ในหัวข้อ 3 แสดงค่าฟังก์ชันวัตถุประสงค์ของแต่ละสตริงในตัวอย่างดังตารางที่ 8

ตารางที่ 8 ค่าฟังก์ชันวัตถุประสงค์สตริงคำตอบเริ่มต้น

ปัญหา ย่อย	สตริง คำตอบ	$f_i(\mathbf{x})$				
		1	2	3	4	5
1	1	7	4.67	4.00	19	4.98
2	2	8	3.67	4.00	17	3.52
3	3	7	4.67	6.33	17	3.58
4	4	7	5.00	5.67	18	4.85
5	5	8	5.33	5.67	17	4.67
6	6	6	4.33	3.67	19	5.15
7	7	6	4.67	5.67	19	4.40
8	8	7	4.67	5.67	19	4.36
9	9	6	5.00	5.33	18	3.77
10	10	7	4.00	4.33	19	2.70

### 5.4 การพัฒนาคำตอบ

การพัฒนาคำตอบจะพัฒนาไปที่ละปัญหาย่อย (Subproblem) โดยแต่ละปัญหาย่อยที่  $i$  ประกอบไปด้วย สตริงที่  $i$  และเซตของสตริงข้างเคียงในปัญหาย่อยที่  $i$  หรือ

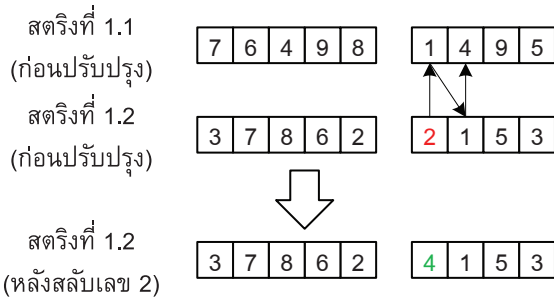


เซต  $B(i)$  ดังนั้นแต่ละเจเนอเรชันจะเกิดกระบวนการพัฒนา คำตอบเท่ากับจำนวนประชากรเริ่มต้น (จำนวนปัญหาย่อย) โดยผู้วิจัยจะแสดงตัวอย่างของการพัฒนาคำตอบในเจเนอเรชันที่ 1 (Gen 1) ปัญหาย่อยที่ 1 (Sub 1) ดังต่อไปนี้

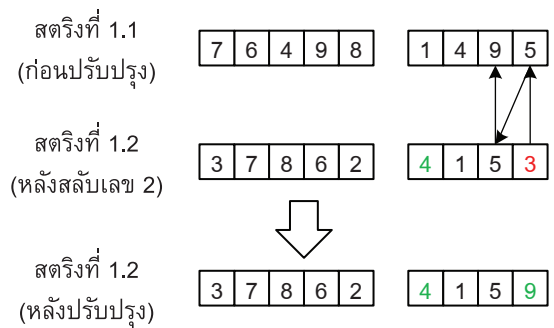
1. ทำการสุ่มเลือกสองสตริงจาก  $B(1)$  ซึ่งประกอบไปด้วย สตริงที่ 1, 9 และ 2 จะได้สตริงรุ่นพ่อแม่มาเพื่อทำการพัฒนา โดยตัวอย่างนี้สุ่มได้สตริงที่ 1 และ 9

2. การครอสโอเวอร์เป็นการนำสตริงคำตอบรุ่นพ่อแม่ มาแลกเปลี่ยนบางส่วนของสตริงซึ่งกันและกัน เพื่อให้เกิด สตริงรุ่นใหม่หรือสตริงคำตอบรุ่นลูก โดยในงานวิจัยนี้จะใช้วิธีการครอสโอเวอร์แบบพาเซิลแมพครอสโอเวอร์ 1 ตำแหน่ง (One-Point Partial-mapped Crossover; PMX) ซึ่งมีวิธีการดังต่อไปนี้

- ทำการสุ่มค่าจำนวนเต็ม  $R^1$  ในช่วง 1 ถึง  $nc-1$  เพื่อระบุขอบเขตการแลกเปลี่ยน โดยตัวอย่างนี้สุ่มได้ตำแหน่งที่ 5 แสดงสตริงพ่อแม่และรุ่นลูกดังรูปที่ 4 และ 5 ตามลำดับ (หมายเหตุ: สตริงที่ a.b คือ สตริงรุ่นลูกที่ b ในเจเนอเรชันที่ a)
- หากสตริงรุ่นลูกเป็นคำตอบที่เป็นไปไม่ได้ให้ทำการสุ่มค่า  $R^2$  ระหว่าง 0 และ 1 เพื่อกำหนดส่วนของสตริงที่ไม่เปลี่ยนแปลง หาก  $R^2$  เท่ากับ 0 สตริงส่วนแรกหรือฝั่งซ้ายจะไม่เปลี่ยนแปลง แต่หากเท่ากับ 1 สตริงส่วนหลังหรือฝั่งขวาจะไม่เปลี่ยนแปลง โดยตัวอย่างนี้สุ่มได้  $R^2$  เท่ากับ 0
- เมื่อได้ส่วนที่ไม่เปลี่ยนแปลงแล้วให้ทำการตรวจสอบหาตำแหน่งในส่วนที่เปลี่ยนแปลงที่มีเลขซ้ำกับส่วนที่ไม่เปลี่ยนแปลง



รูปที่ 6 ปรับปรุงสตริงที่ 1.2 (เลข 2)



รูปที่ 7 ปรับปรุงสตริงที่ 1.2 (เลข 3)

• ทำการปรับปรุงเลขตัวที่มีการซ้ำในสตริงทีละตัวกับคู่สตริงด้วยการสลับ (ก่อนปรับปรุง) ณ ตำแหน่งเดียวกัน หากปรับปรุงแล้วยังมีเลขซ้ำในสตริงให้สลับเลขกับคู่สตริง ณ ตำแหน่งของเลขที่ซ้ำในสตริงไปเรื่อยๆ จนกระทั่งไม่มีเลขซ้ำกัน แสดงตัวอย่างขั้นตอนการปรับปรุงสตริงที่ 1.2 และสตริงรุ่นลูก ดังรูปที่ 6 และ 7 และตารางที่ 9 ตามลำดับ

ตารางที่ 9 สตริงคำตอบรุ่นลูกใน Gen 1 Sub 1

สตริงรุ่นลูก	Priority								
1.1	7	6	4	9	8	1	2	3	5
1.2	3	7	8	6	2	4	1	5	9

3. ทำการประเมินฟังก์ชันวัตถุประสงค์ของสตริงคำตอบรุ่นลูก แสดงดังตารางที่ 10

ตารางที่ 10 ค่าวัตถุประสงค์สตริงรุ่นลูก Gen 1 Sub 1

สตริงรุ่นลูก	$f_k(v)$				
	1	2	3	4	5
1.1	7	4.67	4.00	19	4.98
1.2	6	5.33	6.33	18	3.52

เมื่อได้ค่าฟังก์ชันวัตถุประสงค์ของสตริงเริ่มต้นและสตริงรุ่นลูกแล้ว ให้ทำการจัดเก็บค่าสูงสุด ( $ff_k^{max}$ ) และต่ำสุด ( $ff_k^{min}$ ) ของแต่ละวัตถุประสงค์ในรอบนั้นๆ (หาค่าใหม่ทุกๆ รอบที่เริ่มปัญหาย่อยถัดไปให้หาค่าสูงสุดและต่ำสุดของรอบนั้นๆ ไม่มีจัดเก็บค่าเก่าของรอบก่อนหน้า) แสดงตัวอย่างดังตารางที่ 11 (ในงานวิจัยนี้ ผู้วิจัยได้บวกค่าคงที่เท่ากับ 0.0005 ให้กับค่า ( $ff_k^{max}$ ) เพื่อป้องกันกรณีส่วนหารเท่ากับ 0)

ตารางที่ 11 ค่าวัตถุประสงค์สูงสุดต่ำสุด Gen 1 Sub 1

$k$	1	2	3	4	5
$(ff_k^{max})$	8.01	5.34	6.34	19.01	5.15
$(ff_k^{min})$	6	3.67	3.67	17	2.71

เมื่อได้ค่าสูงสุดและต่ำสุดของค่าฟังก์ชันวัตถุประสงค์แล้วให้นำไปเปรียบเทียบกับค่าเป้าหมาย ( $z_k^*$ ) หาก  $z_k^* > f_k(x)$  จะกำหนดให้  $z_k^*$  เท่ากับ  $f_k(x)$  โดยที่  $k=1,2,\dots,Obj$  กำหนดให้  $z_k^*$  เริ่มต้นเป็นค่าอนันต์) แสดงดังตารางที่ 12

ตารางที่ 12 ค่าเป้าหมาย Gen 1 Sub 1

$k$	1	2	3	4	5
$z_k^*$	6	3.67	3.67	17	2.71

4. ทำการนอร์มัลไล (Normalization) ค่าฟังก์ชันวัตถุประสงค์ของสตริงข้างเคียงในปัญหาย่อยที่  $i (j \in B(i))$ , สตริงคำตอบรุ่นลูกและค่าเป้าหมายด้วยค่า  $ff_k^{max}$  และ  $ff_k^{min}$  จากสมการ  $f_k^N = \frac{(f_k - ff_k^{min})}{(ff_k^{max} - ff_k^{min})} \quad \forall k \in Obj$  แสดงผลลัพธ์ดังตารางที่ 13 (ค่าเป้าหมายที่ผ่านการนอร์มัลไลหรือ  $z_k^N$  สามารถติดลบได้ หากในรอบนั้นค่าต่ำสุดมีค่ามากกว่า  $z_k^*$ )

ตารางที่ 13 ค่าวัตถุประสงค์หลังนอร์มัลไล Gen 1 Sub 1

$j \in B(i)$	สตริง	ค่าวัตถุประสงค์ที่ $k$ หลังนอร์มัลไล				
		1	2	3	4	5
1	1	0.50	0.60	0.13	1.00	0.93
9	9	0	0.80	0.63	0.50	0.44
2	2	1.00	0	0.13	0	0.33
-	1.1	0.50	0.60	0.13	1.00	0.93
-	1.2	0	1.00	1.00	0.50	0.33
-	$z_k^N$	0	0	0	0	0

5. เปรียบเทียบสตริงรุ่นลูกด้วยวิธีการเทบปีเซฟฟ์ โดยใช้ค่าน้ำหนัก  $\lambda_k$  ของปัญหาย่อยที่  $i$  (ตัวอย่างการคำนวณนี้เป็นปัญหาย่อยที่ 1 ดังนั้น  $i = 1$  หากสตริงใดให้ค่าเทบปีเซฟฟ์ที่น้อยกว่าจะเป็นสตริงที่ดีกว่า สามารถคำนวณเทบปีเซฟฟ์ได้จากสมการที่ (7) แสดงผลลัพธ์การคำนวณดังตารางที่ 14 (หมายเหตุ:  $y$  คือสตริงรุ่นลูก)

ตารางที่ 14 ค่าเทบปีเซฟฟ์สตริงรุ่นลูก Gen 1 Sub 1

$y$	$\max\{\lambda_k   f_k^N(y) - z_k^N\}$					$g^e(y   \bar{\lambda}_j, z^N)$
1.1	0.05	0.06	0.01	0.60	0.09	0.60
1.2	0	0.10	0.10	0.30	0.03	0.30

จากตารางที่ 14 จะพบว่าสตริงที่ 1.2 ให้ค่า  $g^e(y)$  ที่ต่ำที่สุดในสตริงคำตอบรุ่นลูกที่เจเนอเรชัน 1 จึงเลือกสตริงที่ 1.2 มาดำเนินการต่อและตัดสตริงที่ 1.1 ออก

### 5.5 การหาค่าที่ดีที่สุด

การหาค่าที่ดีที่สุดด้วยวิธีการเทบปีเซฟฟ์ เพื่อให้ได้คำตอบที่ดีที่สุดของแต่ละค่าน้ำหนัก โดยจะประเมินเฉพาะสตริง  $j$  ที่อยู่ใน  $B(i)$  หรือสตริงข้างเคียงของปัญหาย่อยที่  $i$  เท่านั้น การประเมินจะไปทีละสตริง แสดงขั้นตอนการหาคำตอบที่ดีที่สุด ดังต่อไปนี้

1. ตรวจสอบเงื่อนไขก่อนประเมิน โดยต้องมีเงื่อนไขอย่างใดอย่างหนึ่งดังนี้

- สตริง  $j$  เท่ากับสตริง  $i$  ( $j = i$ )

- ค่าสุ่มระหว่าง 0 ถึง 1 น้อยกว่าหรือเท่ากับอัตราการผลิตเปลี่ยนแปลง  $O(R_j \leq O)$

2. สำหรับสตริง  $j$  ที่ผ่านเงื่อนไข ให้ทำการคำนวณค่า  $g^e(\bar{x}_j | \bar{\lambda}_j, z^N)$  ของสตริง  $j$  และค่า  $g^e(y | \bar{\lambda}_j, z^N)$  ของสตริงรุ่นลูกที่ค่าน้ำหนักเดียวกับสตริง  $j$

3. ทำการเปรียบเทียบค่า  $g^e(\bar{x}_j | \bar{\lambda}_j, z^N)$  และ  $g^e(y | \bar{\lambda}_j, z^N)$  หาก  $g^e(y) < g^e(\bar{x}_j)$  จะกำหนดให้เลขสตริงค่าฟังก์ชันวัตถุประสงค์ และค่า  $g^e(\bar{x}_j)$  ของสตริง  $j$  เท่ากับของสตริงรุ่นลูก แต่หาก  $g^e(\bar{x}_j) = g^e(y)$  จะต้องทำการเปรียบเทียบค่ามากที่รองลงมาของ  $g^e(\bar{x}_j)$  และ  $g^e(y)$  ไปทีละคู่ แสดงตัวอย่างการหาคำตอบที่ดีที่สุดเจเนอเรชันที่ 1 ปัญหาย่อยที่ 1 ดังตารางที่ 15 (หมายเหตุ: SR คือการแทนที่คำตอบ และ Con. คือเงื่อนไข)

ตารางที่ 15 การหาคำตอบที่ดีที่สุด Gen 1 Sub 1

$j \in B(1)$	$R_j$	Con.	$g^e(\bar{x}_j)$		$g^e(y)$		SR
1	0.32	pass	0.60	0.09	0.30	0.10	true
			0.06	0.05	0.10	0.03	
			0.01		0		
9	0.56	pass	0.50	0	0.50	0	true
			0	0	0	0	
			0		0		
2	0.77	fail	0.60	0.03	0.10	0.10	false
			0.01	0	0.05	0.03	
			0		0		

จากตารางที่ 15 เป็นการหาคำตอบที่ดีที่สุดเจเนอเรชันที่ 1 ปัญหาย่อยที่ 1 จะเห็นว่าสตริงที่ 1 ผ่านเงื่อนไขและมีค่า  $g^e(y | \bar{\lambda}_j, z^N)$  สูงสุดที่ดีกว่า  $g^e(\bar{x}_j | \bar{\lambda}_j, z^N)$  จึงเกิดการเปลี่ยนแปลงคำตอบในสตริงที่ 1 ส่วนสตริงที่ 2 ไม่เกิดการประเมินเนื่องจากไม่ผ่านเงื่อนไข กรณีค่าสุ่มเกินกว่าอัตราการผลิตเปลี่ยนแปลงที่กำหนด ลำดับสุดท้ายสตริงที่ 9 ผ่านเงื่อนไขแต่จะเห็นได้ว่าค่า  $g^e(\bar{x}_j | \bar{\lambda}_j, z^N)$  และ  $g^e(y | \bar{\lambda}_j, z^N)$  เมื่อประเมินแบบไล่ระดับจากมากไปน้อยแล้ว ทุกค่ามีค่าเท่ากัน จึงทำการสุ่มค่า 0 ถึง 1 อีกรอบ หากน้อยกว่าหรือเท่ากับ 0.5 จะไม่เกิดการเปลี่ยนแปลง ในกรณีตัวอย่างได้ค่าสุ่มเท่ากับ



0.67 จึงเกิดการเปลี่ยนแปลง

ชุดสตริงคำตอบหลักและค่าฟังก์ชันวัตถุประสงค์ที่ผ่านการพัฒนาคำตอบในทุกๆ ปัญหาย่อย ณ เจเนอเรชันที่ 1 แล้ว ดังตารางที่ 16 และ 17 ตามลำดับ

ตารางที่ 16 สตริงคำตอบหลักหลังจบเจเนอเรชันที่ 1

ปัญหา ย่อย	สตริง คำตอบ	Priority								
		3	7	8	6	2	4	1	5	9
1	1.2	3	7	8	6	2	4	1	5	9
2	2	9	5	4	8	1	3	6	7	2
3	1.15	3	7	8	6	2	9	5	1	4
4	1.14	3	7	8	6	2	4	1	5	9
5	1.9	3	7	8	6	2	4	1	5	9
6	6	9	6	7	3	2	8	5	4	1
7	1.14	3	7	8	6	2	4	1	5	9
8	1.15	3	7	8	6	2	9	5	1	4
9	1.2	3	7	8	6	2	4	1	5	9
10	10	3	9	2	8	5	6	1	7	4

ตารางที่ 17 ค่าฟังก์ชันวัตถุประสงค์สตริงหลังจบ Gen 1

ปัญหา ย่อย	สตริง คำตอบ	$f_k(x)$				
		1	2	3	4	5
1	1.2	6	5.33	6.33	18	3.52
2	2	8	3.67	4.00	17	3.52
3	1.15	6	4.67	4.67	19	4.01
4	1.14	6	5.33	6.33	18	3.52
5	1.9	6	5.33	6.33	18	3.52
6	6	6	4.33	3.67	19	5.15
7	1.14	6	5.33	6.33	18	3.52
8	1.15	6	4.67	4.67	19	4.01
9	1.2	6	5.33	6.33	18	3.52
10	10	7	4.00	4.33	19	2.70

## 5.6 การประเมินจำนวนคำตอบที่เกิดการพัฒนา

จำนวนคำตอบที่เกิดการพัฒนา (The Number of Solution Replace; NSR) เป็นตัวชี้วัดการพัฒนาคำตอบในแต่ละเจเนอเรชัน หากมีค่าเข้าใกล้จำนวนประชากร แสดงว่า

คำตอบมีการพัฒนาในทุกๆ ปัญหาย่อย แต่หากมีค่าเท่ากับ 0 แสดงว่าไม่เกิดการพัฒนาคำตอบเลยในเจเนอเรชันนั้นๆ

โดยกำหนดให้ “true” มีค่าเท่ากับ 1 และ “false” มีค่าเท่ากับ 0 ที่  $t = 1, \dots, T$ ,  $i = 1, \dots, N$  และ  $j = 1, \dots, N$  แสดงจำนวนคำตอบที่เกิดการพัฒนาในแต่ละเจเนอเรชัน ดังสมการที่ (8)

$$NSR_t = \sum_{j=1}^N \min\left(1, \sum_{i=1}^N SR_{i,j,t}\right) \quad (8)$$

## 5.7 การหยุดดำเนินการของอัลกอริทึม

ในแต่ละเจเนอเรชันจะประกอบไปด้วยปัญหาย่อยจำนวนเท่ากับจำนวนประชากร เมื่อจบการพัฒนาคำตอบของปัญหาย่อยที่  $i$  แล้ว จะกลับไปพัฒนาคำตอบของปัญหาย่อยที่  $i+1$  โดยใช้ชุดสตริงหลักที่พัฒนาแล้ว ด้วยวิธีในหัวข้อที่ 5.4 ไปเรื่อยๆ จนกระทั่งครบทุกปัญหาย่อย จึงจะเริ่มเจเนอเรชันถัดไปด้วยชุดสตริงหลักที่พัฒนาแล้ว โดยจะเริ่มต้นที่ปัญหาย่อยที่ 1 ในหัวข้อที่ 5.4 เช่นกัน ทำซ้ำไปเรื่อยๆ จนครบทุกเจเนอเรชัน

การหยุดดำเนินการของอัลกอริทึมจะใช้จำนวนเจเนอเรชันหรือ  $NSR_t$  เป็นตัวกำหนด เมื่ออัลกอริทึมดำเนินการครบจำนวนเจเนอเรชันที่กำหนดหรือ  $NSR_t$  เท่ากับ 0 ติดต่อกันหลายเจเนอเรชัน จะให้อัลกอริทึมหยุดการดำเนินการ

## 6. ตัวชี้วัดประสิทธิภาพอัลกอริทึม

เนื่องจากปัญหาการจัดลำดับการผลิตเป็นปัญหาประเภทเอ็นพียาก จึงไม่สามารถบอกได้ว่าคำตอบใดเป็นคำตอบที่ดีที่สุดจากการใช้อัลกอริทึม การเปรียบเทียบประสิทธิภาพจึงจะต้องพิจารณาจากคำตอบพรอนเทียร์ที่ 1 (Pareto Optimal Frontier) ของแต่ละอัลกอริทึมนำมา หากกลุ่มคำตอบที่ดีที่สุดที่หามาได้ หรือเรียกว่ากลุ่มคำตอบที่แท้จริง (True-pareto Optimal Frontier) ด้วยวิธีเชิงกลุ่มที่ดีที่สุด สมการที่ (6) เพื่อใช้ในการเปรียบเทียบสมรรถนะในการแก้ปัญหาระหว่างอัลกอริทึม

กลุ่มคำตอบที่มีความเหมาะสมที่สุดควรเป็นกลุ่มคำตอบที่มีจำนวนคำตอบมากเพียงพอ และมีคำตอบที่เป็นกลุ่มคำตอบ

ที่แท้จริงหรือใกล้เคียงที่สุด Kumar and Singh [16] ได้นำเสนอไว้ 3 ตัวชี้วัด เรียงตามลำดับความสำคัญ ได้แก่ การลู่เข้าสู่กลุ่มคำตอบที่แท้จริง (Convergence to the Pareto-optimal Set) อัตราส่วนของคำตอบที่ไม่ถูกรอบงำ (Ratio of Non-dominated Solution) และการกระจายตัวของกลุ่มคำตอบที่หาได้ (Spread Measurement)

### 6.1 Convergence to the Pareto-optimal Set

การวัดสมรรถนะอัลกอริทึมด้านการลู่เข้าสู่กลุ่มคำตอบที่แท้จริงเป็นการเปรียบเทียบระยะห่างระหว่างกลุ่มคำตอบที่ได้จากอัลกอริทึม (Obtained Pareto Optimal Solution) กับกลุ่มคำตอบที่แท้จริง แสดงสูตรคำนวณดังสมการที่ (9) และ (10)

$$Convergence_j = \frac{1}{|S^*|} \sum_{y \in S^*} \min \{d_{xy} \mid x \in S_j\} \quad (9)$$

$$d_{xy} = \sqrt{\sum_{k=1}^K \left( \frac{f_k(x) - f_k(y)}{f_k^{max} - f_k^{min}} \right)^2} \quad (10)$$

เมื่อ  $S_j$  คือ คำตอบพرونเทียร์ที่ 1 (1st Frontier) ซึ่งได้จากอัลกอริทึม  $j$ ,  $S^*$  คือกลุ่มคำตอบที่แท้จริง,  $|S^*|$  คือ จำนวนคำตอบของกลุ่มคำตอบที่แท้จริง,  $d_{xy}$  คือ ระยะห่างระหว่างคำตอบที่หาได้ ( $x$ ) กับคำตอบที่แท้จริง ( $y$ ),  $f_k^{max}$  คือ ค่าฟังก์ชันวัตถุประสงค์ที่  $k$  ที่มีค่ามากที่สุดของกลุ่มคำตอบที่แท้จริง,  $f_k^{min}$  คือ ค่าฟังก์ชันวัตถุประสงค์ที่  $k$  ที่มีค่าน้อยที่สุดของกลุ่มคำตอบที่แท้จริง,  $f_k(x)$  คือ ค่าฟังก์ชันวัตถุประสงค์ที่  $k$  ของคำตอบที่หาได้,  $f_k(y)$  คือ ค่าฟังก์ชันวัตถุประสงค์ที่  $k$  ของคำตอบที่แท้จริง

ถ้าตัวชี้วัดสมรรถนะอัลกอริทึมด้านการลู่เข้าสู่กลุ่มคำตอบที่แท้จริงของอัลกอริทึม  $j$  มีค่าเข้าใกล้ 0 แสดงว่ากลุ่มคำตอบของอัลกอริทึม  $j$  ลู่เข้าสู่กลุ่มคำตอบที่แท้จริงมากและเป็นอัลกอริทึมที่ดี

### 6.2 Spread Measurement

การวัดสมรรถนะอัลกอริทึมด้านการกระจายตัวของ

กลุ่มคำตอบที่หาได้ เป็นการวัดระยะห่างระหว่างคำตอบที่ต่อเนื่องกันภายในกลุ่มคำตอบที่หาได้ของอัลกอริทึม แสดงสูตรคำนวณดังสมการที่ (11) และ (12)

$$Spread_j = \frac{\sum_{c=1}^C d(E_c) + \sum_{i=1}^{|S_j|} |d(x_i) - \bar{d}|}{\sum_{c=1}^C d(E_c) + |S_j| \bar{d}} \quad (11)$$

$$d(x_i) = \min_{y \in S_j} \sqrt{\sum_{k=1}^K \left( \frac{f_k(x) - f_k(y)}{f_k^{max} - f_k^{min}} \right)^2} \quad (12)$$

เมื่อ  $|S_j|$  คือ จำนวนคำตอบของคำตอบพرونเทียร์ที่ 1 ซึ่งได้รับจากอัลกอริทึม,  $j d(x_i)$  คือ ระยะห่างที่น้อยที่สุดระหว่างคำตอบตำแหน่งที่  $i$  กับคำตอบตำแหน่งที่  $y$  โดยที่  $y \neq i$   $\bar{d}$  คือ ค่าเฉลี่ยของระยะห่างระหว่างคำตอบที่หาได้,  $d(E_c)$  คือ ระยะห่างระหว่างคำตอบปลายสุดของคำตอบที่จริง (Extreme Solution in the Set of True Pareto-front) กับคำตอบขอบเขตของคำตอบที่หาได้ (Boundary Solution in the Set of Obtained Solution),  $C$  คือ จำนวนคำตอบปลายของคำตอบที่แท้จริง

### 6.3 Ratio of Non-dominated Solution

การวัดสมรรถนะอัลกอริทึมด้านอัตราส่วนของคำตอบที่ไม่ถูกรอบงำ (Ratio of Non-Dominated Solution) เป็นการพิจารณาว่าคำตอบพرونเทียร์ที่ 1 ซึ่งได้จากอัลกอริทึมที่  $j$  ว่ามีจำนวนคำตอบที่ไม่ถูกรอบงำจากคำตอบของอัลกอริทึมอื่นคิดเป็นสัดส่วนเท่าใด โดยแบ่งเปรียบเทียบสมรรถนะเป็นสองด้านดังนี้

1. การเปรียบเทียบภายในอัลกอริทึม (Self-comparison) เป็นการเปรียบเทียบระหว่างจำนวนของคำตอบที่แท้จริงที่อยู่ในคำตอบพرونเทียร์ที่ 1 เพื่อพิจารณาอัตราส่วน ยิ่งมีค่าเข้าใกล้ 1 แสดงว่าคำตอบที่ 1 ของอัลกอริทึม  $j$  โดยส่วนใหญ่เป็นคำตอบที่ดีที่สุด แสดงสูตรคำนวณดังสมการที่ (13)

$$R_{NDS1,j} = \frac{|S_j - \{x \in S_j \mid \exists y \in S^* : y < x\}|}{|S_j|} \quad (13)$$

2. การเปรียบเทียบกับกลุ่มคำตอบที่แท้จริง (Pareto-

optimum Comparison) เป็นการเปรียบเทียบระหว่างจำนวนของคำตอบที่ 1 ที่อยู่ในกลุ่มคำตอบที่แท้จริง เพื่อพิจารณาอัตราส่วน ยิ่งมีค่าเข้าใกล้ 1 แสดงว่าคำตอบที่ 1 ของอัลกอริทึม  $j$  โดยส่วนใหญ่เป็นคำตอบที่ดีที่สุดของทุกๆ อัลกอริทึม แสดงสูตรดังสมการที่ (14)

$$R_{NDS2,j} = \frac{|S_j - \{x \in S_j \mid \exists y \in S^* : y \prec x\}|}{|S^*|} \quad (14)$$

เมื่อ  $y \prec x$  คือ คำตอบที่หาได้ ( $x$ ) ไม่ถูกครอบงำโดยคำตอบที่แท้จริง ( $y$ )

## 7. การทดลอง

การทดลองในงานวิจัยนี้ ใช้ปัญหามาตรฐานในการจัดลำดับของ McMullen [17] และ Mansouri [5] มาดัดแปลงเพื่อให้เข้ากับปัญหาอุตสาหกรรมจริงในปัจจุบัน โดยมีทั้งหมด 10 ปัญหา แบ่งเป็นปัญหามาตรฐานขนาดเล็ก ขนาดกลาง และขนาดใหญ่ ซึ่งมีความแตกต่างกันของจำนวนชิ้นงาน สัดส่วนความต้องการของรถยนต์แต่ละรุ่น ตัวเลือกของรถยนต์ และจำนวนรุ่นของรถยนต์ แสดงรายละเอียดของปัญหา อัตราส่วนตัวเลือก  $p_o : q_o$  สีและรุ่นรถยนต์ลำดับผลิตภัณฑ์ก่อนหน้าดังตารางที่ 18,

19 และ 20

### 7.1 การกำหนดค่าพารามิเตอร์

เนื่องจากปัญหาในงานวิจัยนี้มีหลากหลายข้อจำกัดที่แตกต่างกัน จึงจำเป็นต้องทำการทำสอบค่าพารามิเตอร์ของอัลกอริทึมให้เหมาะสมสำหรับแต่ละปัญหา เพื่อให้อัลกอริทึมมีประสิทธิภาพในการหาคำตอบของแต่ละปัญหาได้ดีที่สุด โดยจะทำการทดสอบพารามิเตอร์ของแต่ละปัญหาและแต่ละอัลกอริทึมด้วยวิธีการออกแบบการทดลอง (Design of Experiment; DOE) ซึ่งผู้วิจัยทำการแบ่งทดสอบพารามิเตอร์เป็นสองส่วน ส่วนที่หนึ่งทดสอบอัลกอริทึมละ 2 พารามิเตอร์ พารามิเตอร์ละ 3 ระดับปัจจัย และจำนวนทำซ้ำ 3 ครั้ง แสดงรายละเอียดดังตารางที่ 21 และส่วนที่สองทดสอบหาจำนวนเจนเนอเรชันที่เหมาะสมในช่วง 200 ถึง 2000 รอบ โดยใช้พารามิเตอร์จากส่วนที่หนึ่งของแต่ละปัญหาและแต่ละอัลกอริทึม

โดยที่พารามิเตอร์ 1 และ 2 ของ MOEA/D คือ จำนวนสตริงข้างเคียง ( $Nb$ ) และอัตราการเปลี่ยนแปลง ( $O$ ) ตามลำดับ และพารามิเตอร์ 1 และ 2 ของ MODE คือค่าปัจจัยขยายผลต่าง ( $F$ ) และอัตราการครอสโอเวอร์ ( $CR$ ) ตามลำดับ

ตารางที่ 18 ปัญหาที่ใช้ในงานวิจัย

ปัญหาการทดลอง (ชิ้นงาน)	จำนวนรุ่นรถยนต์	สัดส่วนผลิตภัณฑ์ (Minimum Part Set; MPS)	ความยาวสตริง	จำนวนสีรถยนต์ (สูงสุดที่ติดกัน)	จำนวนตัวเลือก	
Set 1 (16)	1.1	5	5:3:2:1:1	12	5(3)	2
	1.2	5	4:4:2:1:1	12	5(3)	2
Set 2 (24)	2.1	5	7:3:2:2:1	15	5(3)	4
	2.2	5	4:3:3:3:2	15	5(3)	4
Set 3 (65)	3.1	5	8:7:2:2:1	20	5(3)	8
	3.2	5	5:4:4:4:3	20	5(3)	8
Set 4 (70)	4.1	10	7:5:1:1:1:1:1:1:1:1	20	9(4)	10
	4.2	10	4:4:4:2:1:1:1:1:1:1	20	9(4)	10
Set 5 (148)	5.1	15	20:20:20:15:15:1:1:1:1:1:1:1:1:1:1	100	12(5)	12
	5.2	15	15:15:10:10:10:10:10:10:4:1:1:1:1:1:1	100	12(5)	12

ตารางที่ 19 สีและลำดับการผลิตก่อนหน้า

Set	สีรถยนต์ลำดับผลิตก่อน	รหัสรถยนต์ที่ผลิตลำดับก่อนหน้า (รุ่นรถยนต์)
1	1,3	4,1 (D,A)
2	2,3	2,1 (B,A)
3	4,5	1,5,1,2,4,3,4,2,2 (A,E,A,B,D,C,D,B,B)
4	1,2,8	3,1,10,7,4,7,8,4,7,3,10,8,2,5 (C,A,J,G,D,G,H,D,G,C,J,H,B,E)
5	3,6,5,10	5,11,9,6,6,4,13,2,6,2,14,1,4,6,14,8,4,5,14,1,3,12,3,8,5,14,13,15,6,8,12,6,14,7,1,3,6,1,13,14,1 (E,K,I,F,D,M,B,F,B,N,A,D,F,N,H,D,E,N,A,C,L,C,H,E,N,M,O,F,H,L,F,N,G,A,C,F,A,M,N,A,D)

ตารางที่ 20 อัตราส่วนการผลิตในลำดับการผลิตย่อย

ตัวเลือก	$p_o : q_o$				
	Set 1	Set 2	Set 3	Set 4	Set 5
$o_1$	2/3	2/3	2/3	2/3	2/3
$o_2$	1/2	1/2	1/2	1/2	1/2
$o_3$	-	1/3	1/3	1/3	17/30
$o_4$	-	1/3	1/3	1/3	1/2
$o_5$	-	-	3/5	3/5	3/5
$o_6$	-	-	4/5	4/5	39/43
$o_7$	-	-	2/10	2/10	10/15
$o_8$	-	-	3/7	10/15	6/30
$o_9$	-	-	-	-	2/15
$o_{10}$	-	-	-	-	4/9
$o_{11}$	-	-	-	-	2/5
$o_{12}$	-	-	-	-	3/7

ตารางที่ 21 การออกแบบการทดลอง

อัลกอริทึม	พารามิเตอร์ 1			พารามิเตอร์ 2		
MOEA/D	5	10	15	0.3	0.7	1.0
MODE	0.5	1.0	2.0	0.3	0.7	0.9

ในงานวิจัยนี้ ผู้วิจัยใช้โปรแกรม MATLAB 2015b ในการดำเนินงานอัลกอริทึมของแต่ละปัญหา 2000 เจเนอเรชัน

บนเครื่องคอมพิวเตอร์ Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz RAM 16 GB Windows 10 64 bit Operation System แสดงค่าพารามิเตอร์ที่เหมาะสมที่สุดในแต่ละปัญหา และแต่ละอัลกอริทึม ทดสอบที่ระดับนัยสำคัญเท่ากับ 0.05 ด้วยโปรแกรม Minitab 17 ดังตารางที่ 22

ตารางที่ 22 ค่าพารามิเตอร์ของอัลกอริทึม

Set	MOEA/D			MODE		
	Nb	O	T	F	CR	T
1.1	5	0.3	1000	2.0	0.9	800
1.2	5	0.3	1000	2.0	0.9	800
2.1	5	0.3	1200	2.0	0.7	1000
2.2	5	0.3	1200	2.0	0.7	1000
3.1	5	0.3	1400	2.0	0.7	1000
3.2	5	0.3	1400	2.0	0.7	1000
4.1	5	0.3	1400	0.5	0.7	1000
4.2	5	0.3	1400	0.5	0.7	1000
5.1	5	1.0	2000	0.5	0.9	1800
5.2	5	1.0	2000	0.5	0.9	1800

เมื่อ Nb คือ จำนวนสตริงข้างเคียง O คือ อัตราการเปลี่ยนแปลง F คือ ค่าปัจจัยขยายผลต่าง CR คือ อัตราการครอสโอเวอร์ และ T คือ จำนวนเจเนอเรชัน





## 8. ผลการทดลอง

ในงานวิจัยนี้ใช้ Simplex Lattice Design ในการสร้าง คำนวณที่ 5 Component 5 Degree of Lattice with Center Point and Axial Points ซึ่งจะได้ค่า  $\lambda_{jk}$  ทั้งหมด 131 แถว 5 หลัก ดังนั้นผู้วิจัยจึงกำหนดจำนวนประชากรที่ใช้ในการทดลองของแต่ละอัลกอริทึมเท่ากับ 131 ตัว ตามจำนวนประชากร (จำนวนปัญหาย่อย) ของ MOEA/D ซึ่งเป็นอัลกอริทึมที่มีข้อจำกัดในเรื่องของจำนวนประชากรที่ต้องเปลี่ยนแปลงตามวิธีการสร้างค่าน้ำหนัก

การแก้ปัญหาในแต่ละอัลกอริทึมแต่ละปัญหา จะใช้จำนวนเจเนอเรชันของปัญหาที่มากที่สุดคือ 2000 รอบ เพื่อให้ครอบคลุมกับทุกปัญหาการทดลอง โดยจะสร้างสตริงคำตอบ

เริ่มต้นจากการสุ่มอิสระในแต่ละอัลกอริทึมและแต่ละรอบการทำงานซ้ำ เพื่อให้เกิดความหลากหลายของการทดลอง ซึ่งงานวิจัยนี้เปรียบเทียบอัลกอริทึม 2 ตัว ได้แก่ MOEA/D และ MODE ดังนั้นผู้วิจัยจึงเลือกใช้ 2-Sample  $t$  ในการทดสอบทางสถิติ เพื่อเปรียบเทียบค่าตอบสุดท้ายในทุกอัลกอริทึมและทุกการทำงานซ้ำ แสดงผลจากการทดลองการแก้ปัญหาตัวอย่างอยู่ในรูปของค่าฟังก์ชันวัตถุประสงค์เป็นชุดคำตอบจากแต่ละอัลกอริทึมที่ผ่านการคำนวณค่าตัวชี้วัดต่างๆ แล้วจะได้ค่าเฉลี่ยตัวชี้วัดของแต่ละอัลกอริทึม (จำนวนทำซ้ำ 10 ครั้ง) ดังตารางที่ 23 ซึ่งจะพบว่า MOEA/D โดยรวมมีสมรรถนะด้านการเข้าสู่คำตอบที่แท้จริงน้อยกว่า (ดีกว่า) MODE ในทุกๆ ปัญหา แต่ในบางปัญหายังไม่สามารถระบุ

ตารางที่ 23 ผลการทดลองของการแก้ปัญหาตัวอย่าง

Algorithm	Problem Set									
	1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2	5.1	5.2
Convergence										
MOEA/D	<b>0.00718</b> (0.0047)*	0.01183 (0.00688)	<b>0.01621</b> (0.0091)*	<b>0.01639</b> (0.0070)*	<b>0.00742</b> (0.0032)*	<b>0.006374</b> (0.0024)*	<b>0.01084</b> (0.0045)*	<b>0.004841</b> (0.0022)*	0.03453 (0.03121)	0.03358 (0.01919)
MODE	0.01446 (0.00572)	0.00635 (0.00594)	0.02631 (0.00755)	0.02621 (0.00621)	0.01343 (0.00470)	0.014400 (0.00450)	0.01838 (0.00669)	0.009369 (0.00280)	0.04313 (0.02273)	0.03877 (0.02013)
Spread										
MOEA/D	<b>0.55398</b> (0.0228)*	0.5639 (0.0879)	0.5607 (0.0520)	0.5273 (0.0796)	<b>0.4540</b> (0.0457)*	<b>0.4390</b> (0.0445)*	0.5579 (0.0437)	<b>0.4663</b> (0.0443)*	<b>0.47857</b> (0.0235)*	<b>0.44827</b> (0.0210)*
MODE	0.59740 (0.03470)	<b>0.4802</b> (0.0272)*	0.5887 (0.0651)	0.4878 (0.0704)	0.5124 (0.0645)	0.4927 (0.0414)	0.5352 (0.0523)	0.5469 (0.0331)	0.55390 (0.03860)	0.5401 (0.0582)
RND51										
MOEA/D	<b>0.8611</b> (0.0746)*	0.8425 (0.1051)	0.5809 (0.2000)	<b>0.5234</b> (0.1833)*	<b>0.5130</b> (0.1576)*	<b>0.5740</b> (0.1277)*	<b>0.4557</b> (0.1759)*	<b>0.6260</b> (0.1435)*	0.3590 (0.3530)	0.2611 (0.3040)
MODE	0.7830 (0.0679)	<b>0.9389</b> (0.0599)*	0.4415 (0.1402)	0.2733 (0.1363)	0.2015 (0.1032)	0.1504 (0.1172)	0.1420 (0.1366)	0.2176 (0.1107)	0.1137 (0.2565)	0.1725 (0.2779)
RND52										
MOEA/D	<b>0.4289</b> (0.0372)*	0.4352 (0.1461)	0.1870 (0.0644)	<b>0.1473</b> (0.0544)*	<b>0.07320</b> (0.0225)*	<b>0.08077</b> (0.0180)*	<b>0.07664</b> (0.0296)*	<b>0.07455</b> (0.0171)*	0.0759 (0.0748)	0.0602 (0.0701)
MODE	0.3897 (0.0338)	<b>0.7446</b> (0.0469)*	0.1420 (0.0452)	0.0777 (0.0387)	0.02876 (0.01473)	0.02116 (0.01649)	0.02388 (0.02297)	0.02591 (0.01318)	0.0241 (0.0543)	0.0398 (0.0641)
Computational time (sec.)										
MOEA/D	<b>253.530</b> (7.100)*	<b>172.26</b> (26.37)*	<b>285.600</b> (8.760)*	<b>273.17</b> (24.29)*	<b>343.181</b> (2.471)*	<b>346.572</b> (2.246)*	<b>398.181</b> (2.745)*	<b>399.003</b> (1.114)*	1839.4 (38.0)	<b>1926.11</b> (8.82)*
MODE	322.877 (2.185)	312.86 (5.43)	365.366 (1.296)	361.35 (3.74)	420.216 (2.337)	423.110 (3.590)	488.560 (6.320)	524.190 (4.320)	1864.1 (36.8)	1974.21 (20.47)

หมายเหตุ: Mean(Std. Dev.) และ \* หมายถึง ดีกว่า ตามหลักสถิติ (แตกต่างกันมีนัยสำคัญที่ความเชื่อมั่น 95 เปอร์เซนต์)

ได้ว่าแตกต่างกันอย่างชัดเจนตามหลักทางสถิติ ส่วนด้านการกระจายตัวของคำตอบ จะเห็นว่า MOEA/D มีการกระจายตัวที่ดีกว่าในปัญหาขนาดใหญ่ แต่โดยรวมการกระจายตัวของทั้งสองอัลกอริทึมไม่มีความแตกต่างกัน ด้านอัตราส่วนที่ไม่ถูกครอบงำจะเห็นได้ว่า MOEA/D จะมีค่า RNDS1 และ RNDS2 ในปัญหาขนาดเล็กที่ไม่แน่นอน แต่ให้ค่า RNDS1 และ RNDS2 ที่ดีกว่าอย่างมีนัยสำคัญในปัญหาขนาดกลางและใหญ่ และสุดท้ายด้านเวลาที่ใช้ในการค้นหาคำตอบ จะเห็นได้ว่า MOEA/D ใช้เวลาน้อยกว่า (ดีกว่า) MODE ในปัญหาขนาดทุกขนาดอย่างชัดเจน

จากผลการทดลองสามารถสรุปได้ว่า MOEA/D มีสมรรถนะโดยรวมในการแก้ปัญหาการจัดลำดับการผลิตรถยนต์แบบมากวัตถุประสงค้ที่ดีกว่า MODE อย่างมีนัยสำคัญในด้านการลู่เข้าสู่คำตอบที่แท้จริง อัตราส่วนของคำตอบที่ไม่ถูกครอบงำ และเวลาที่ใช้ในการหาคำตอบ ส่วนด้านการกระจายตัวจะเห็นได้ว่า MOEA/D ดีกว่าอย่างชัดเจนในปัญหาขนาดกลางถึงใหญ่

## 9. สรุป

งานวิจัยนี้นำเสนอการแก้ปัญหาการจัดลำดับการผลิตรถยนต์แบบมากวัตถุประสงค้บนสายการประกอบผลิตภัณฑ์ผสมแบบสองด้าน ซึ่งเป็นสายการผลิตที่สามารถตอบสนองความต้องการของลูกค้าได้อย่างรวดเร็วตามแนวคิดการผลิตแบบทันเวลาพอดี (Just In Time) โดยมีวัตถุประสงค์ของการจัดลำดับการผลิตทั้งสิ้น 5 วัตถุประสงค์ ได้แก่ จำนวนครั้งการเปลี่ยนแปลงสีน้อยที่สุด จำนวนรถยนต์ที่ละเมิตรวมน้อยที่สุด ปริมาณงานที่ไม่เสร็จในการผลิตน้อยที่สุด เวลารอคอยงานรวมในการผลิตน้อยที่สุด และความแปรผันรวมของสัดส่วนการผลิตน้อยที่สุด พร้อมทั้งนำเสนอวิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์โดยยึดหลักการจำแนก (MOEA/D) เข้ามาประยุกต์ใช้ในการแก้ปัญหา ซึ่งพบว่า ทั้ง MOEA/D และ MODE ให้คำตอบที่มีประสิทธิภาพดีแต่ MOEA/D ใช้เวลาในการค้นหาคำตอบของปัญหาแบบมากวัตถุประสงค้ได้รวดเร็ว มีการลู่เข้าสู่คำตอบที่แท้จริงดีกว่า และมีอัตราส่วนของคำตอบที่ไม่ถูกครอบงำที่มากกว่า MODE

สำหรับแนวทางการศึกษาในอนาคตอาจปรับเปลี่ยนวิธีการพัฒนาคำตอบของ MOEA/D เพื่อลดเวลาในการค้นหาคำตอบและทำให้ MOEA/D มีทิศทางการพัฒนาคำตอบที่ดีขึ้น เนื่องจากวิธีครอสโอเวอร์จะให้คำตอบที่ใกล้เคียงกับคำตอบเดิมและไม่เปลี่ยนแปลงมาก อาจทำให้คำตอบที่ได้เป็นคำตอบที่ดีเฉพาะกลุ่ม (Local Optima) ไม่สามารถค้นเจอคำตอบที่ดีขึ้นได้

## เอกสารอ้างอิง

- [1] O. S. Akgündüz and S. Tunalı, “An adaptive genetic algorithm approach for the mixed-model assembly line sequencing problem,” *International Journal of Production Research*, vol. 48, no. 17, pp. 5157–5179, 2010.
- [2] A. Zinlou, C. Gagné, and M. Gravel, “Design of an efficient genetic algorithm to solve the industrial car sequencing problem,” in *Advances in Evolutionary Algorithms*, London: Headquarters IntechOpen Limited, 2008, pp. 377–400.
- [3] O. S. Akgündüz and S. Tunalı, “A review of the current applications of genetic algorithms in mixed-model assembly line sequencing,” *International Journal of Production Research*, vol. 49, no. 15, pp. 4483–4503, 2011.
- [4] C. J. Hyun, Y. Kim, and Y. K. Kim “A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines,” *Computers & Operations Research*, vol. 25, no. 7–8, pp. 675–690, 1998.
- [5] S.A. Mansouri, “A multi-objective genetic algorithm for mixed-model sequencing on jit assembly lines,” *European Journal of Operational Research*, vol. 167, no. 3, pp. 696–716, 2005.
- [6] P. Chutima and S. Olarnviwatchai, “A multi-objective car sequencing problem on two-



- sided assembly lines,” *Journal of Intelligent Manufacturing*, pp. 1–20, 2016.
- [7] Z. He and G. G. Yen, “Many-objective evolutionary algorithm: objective space reduction and diversity improvement,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 145–160, 2016.
- [8] L. Hui and Z. Qingfu, “A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages,” *Parallel Problem Solving from Nature IX*, vol. LNCS 4193, pp. 583–592, 2006.
- [9] D. Brockhoff and E. Zitzler, “Objective reduction in evolutionary multi-objective optimization: Theory and applications,” *Evolutionary Computation*, vol. 17, no. 2, pp. 135–166, 2009.
- [10] Z. Qingfu and L. Hui, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [11] U. Özcan and B. Toklu, “Balancing of mixed-model two-sided assembly lines,” *Computers and Industrial Engineering*, vol. 57, no. 1, pp. 217–227, 2009.
- [12] S. Olanvithchai, “Multi-objective car sequencing problem on mixed-model two-sided assembly lines,” M.S. thesis, Department of Industrial Engineering, Faculty of Engineering, Chulalongkorn University, 2013.
- [13] N. Manavizadeh, L. Tavakoli, M. Rabbani, and F. Jolai, “A multi-objective mixed-model assembly line sequencing problem in order to minimize total costs in a Make-To-Order environment, considering order priority,” *Journal of Manufacturing Systems*, vol. 32, no. 1, pp. 124–137, 2013.
- [14] A. R. Rahimi-Vahed, M. Rabbani, R. Tavakkoli-Moghaddam, S. A. Torabi, and F. Jolai, “A multi-objective scatter search for a mixed-model assembly line sequencing problem,” *Advanced Engineering Informatics*, vol. 21, no. 1, pp. 85–99, 2007.
- [15] A. Konak, D. W. Coit, and A. E. Smith, “Multi-objective optimization using genetic algorithms: A tutorial,” *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.
- [16] R. Kumar and P. K. Singh, “Pareto evolutionary algorithm hybridized with local search for biobjective TSP,” *Hybrid Evolutionary Algorithms*, vol. 75, pp. 361–398, 2007.
- [17] P. R. McMullen, “An efficient frontier approach to addressing JIT sequencing problems with setups via search heuristics,” *Computers and Industrial Engineering*, vol. 41, pp. 335–353, 2001.