

การแก้ปัญหการจับคู่เที่ยวบินของสายการบินราคาประหยัดแบบมากวัตถุประสงค์โดยใช้วิธีเมตาฮิวริสติก

Solving Many-objective Cockpit Crew Pairing Problem of Low-cost Airline using Metaheuristic

นิชา ภฤษณพันธุ์ ปารเมศ ชุตินา*

ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

254 ถนนพญาไท แขวงวังใหม่ เขตปทุมวัน กรุงเทพมหานคร 10330

Nicha Krisanaphan Parames Chutima*

Department of Industrial Engineering, Faculty of Engineering, Chulalongkorn University

254 Phayathai Road, Wang Mai, Pathum Wan, Bangkok 10330

*Corresponding author Email: parames.c@chula.ac.th

(Received: October 25, 2019; Accepted: February 27, 2020)

บทคัดย่อ

การแก้ปัญหการจับคู่เที่ยวบินของสายการบินราคาประหยัดแบบมากวัตถุประสงค์ จัดเป็นปัญหาแบบมากวัตถุประสงค์ (Many-Objective Optimization Problems: MaOPs) และเป็นปัญหาประเภทเอ็นพี-ฮาร์ด (Non-deterministic Polynomial Hard: NP-Hard) งานวิจัยนี้เสนออัลกอริทึมที่พัฒนาเพื่อแก้ปัญหาดังกล่าว ได้แก่วิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์โดยยึดหลักการจำแนก (Multi-Objective Evolutionary Algorithm based on Decomposition: MOEA/D) และวิธีเชิงพันธุกรรมแบบการจัดลำดับที่ไม่ถูกครอบงำ III (Non-dominated Sorting Genetic Algorithm III: NSGA-III) โดยพิจารณาวัตถุประสงค์จำนวน 5 วัตถุประสงค์ พร้อมกันดังนี้ เวลาว่างระหว่างเที่ยวบินน้อยที่สุด ปรับดุลภาระงานให้เท่าเทียมกัน เส้นทางการบินสั้นน้อยที่สุด ระยะทางการบินแตกต่างกันน้อยที่สุด และจำนวนคัปบินน้อยที่สุด ผลการวิจัยพบว่า MOEA/D มีสมรรถนะด้านการเข้าสู่กลุ่มคำตอบที่แท้จริง (GD) ด้านการลู่เข้าและความหลากหลายของกลุ่มคำตอบ (IGD) ด้านการกระจายของกลุ่มคำตอบที่ได้ (Spread) ด้านอัตราส่วนของคำตอบที่ไม่ถูกครอบงำเทียบกับกลุ่มคำตอบที่อัลกอริทึมหาได้ (RNDs1) ด้านอัตราส่วนของคำตอบที่ไม่ถูกครอบงำเทียบกับกลุ่มคำตอบที่ดีที่สุด (RNDs2) และด้านเวลาที่ใช้ในการค้นหาคำตอบ (CPU Time) ดีกว่า NSGA-III

คำสำคัญ: ปัญหการจับคู่เที่ยวบิน มากวัตถุประสงค์ วิธีเชิงพันธุกรรม

ABSTRACT

Solving many-objective cockpit crew pairing problem of low-cost airline is classified as many-objective optimization problems (MaOPs) and non-deterministic polynomial hard (NP-Hard). The purpose of research is to compare the efficiency of two algorithms as follows multi-objective evolutionary algorithm based on decomposition (MOEA/D) and non-dominated sorting genetic algorithm III (NSGA-III). The objectives considered in this research are minimizing idle time, balancing workload, minimizing repeat flight leg, minimizing the difference of nautical mile between each flight code, and minimizing number of pair of cockpit crews. The experiments show that MOEA/D outperforms NSGAIII in terms of GD, IGD, Spread, RND1, RND2, and CPU Time

Keyword: cockpit crew pairing problem, many-objective, genetic algorithm

1. บทนำ

อุตสาหกรรมการบินเป็นอุตสาหกรรมที่มีขนาดใหญ่และต้นทุนสูง การลดต้นทุนจึงเป็นสิ่งที่จำเป็นในการเพิ่มโอกาสในการแข่งขัน โดยเฉพาะอย่างยิ่งสำหรับสายการบินราคาประหยัด (Low Cost Airline) มีการแข่งขันด้านราคาที่สูงขึ้น มีการตัดราคาจากสายการบินอื่น แม้รายได้จากค่าโดยสารและน้ำหนักส่วนเกินเพิ่มขึ้นเนื่องจากปริมาณการขนส่งที่เพิ่มขึ้นแต่ก็ประสบกับปัญหาขาดทุนอย่างต่อเนื่อง

ปัญหาการจัดตารางการดำเนินงานของนักบิน (Crew Scheduling) ประกอบด้วยปัญหา 2 ส่วน คือ ส่วนแรกเป็นส่วนที่เกี่ยวข้องกับปัญหาการจัดคู่นักบิน (Crew Pairing) และส่วนที่สองเป็นส่วนที่เกี่ยวข้องกับปัญหาการมอบหมายงานให้นักบิน (Crew Rostering) ซึ่งสายการบินราคาประหยัดกรณีศึกษาข้างต้นยังไม่มีเครื่องมือที่ช่วยในการจัดตารางการดำเนินงานของนักบิน

อย่างไรก็ตามตลอดระยะเวลาที่ผ่านมา สายการบินกรณีศึกษามีจำนวนผู้โดยสารที่เพิ่มสูงขึ้น ทำให้มีการเพิ่มสูงจำนวนเครื่องบินเพื่อตอบสนองต่อความต้องการของผู้โดยสาร ส่งผลให้มีจำนวนเที่ยวบินที่เพิ่มสูงขึ้น และเนื่องจากปริมาณข้อมูลที่เพิ่มขึ้นนี้ ทำให้พนักงานวางแผนจัดตารางการดำเนินงานได้ตามเที่ยวบินที่มีอยู่เท่านั้น แต่ไม่ได้คำนึงถึงค่าใช้จ่ายที่เพิ่มขึ้น ภาระงานที่เพิ่มขึ้นอย่าง

ไม่เท่าเทียม ทำให้ต้องจ้างนักบินเพิ่มจากสายการบินอื่น ส่งผลให้ค่าใช้จ่ายด้านการจัดตารางเพิ่มสูงขึ้นมาก สำหรับปัญหาการจัดตารางการปฏิบัติงาน หากสามารถจัดการให้เกิดประโยชน์สูงสุด โดยลดค่าใช้จ่าย ทั้งนี้จะต้องไม่ขัดกับข้อบังคับ ระเบียบทางการบินของรัฐ สำนักงานการบินพลเรือนแห่งประเทศไทย องค์การการบินพลเรือนระหว่างประเทศ (ICAO) และกฎของสายการบิน

ปัญหาการจัดคู่นักบินของสายการบินแบบมากวัตถุประสงค์ถูกจัดเป็นปัญหาประเภทเอ็นพี-ฮาร์ด (Non-deterministic Polynomial Hard: NP-Hard) เนื่องจากมีความยุ่งยาก ซับซ้อน และมีขนาดใหญ่ หากใช้สมการเชิงคณิตศาสตร์มาแก้ปัญหาจะใช้เวลาในการค้นหาคำตอบนาน และมีจำนวนคำตอบที่มาก ดังนั้นจึงมีการนำวิธีทางเมตาฮิวริสติกเข้ามาช่วยแก้ปัญหาแบบหลายวัตถุประสงค์ (Multi-Objective Optimization Problems: MOPs) เพื่อหาคำตอบที่เหมาะสมและเป็นที่ยอมรับได้ในเวลาที่สั้นลง ตัวอย่างเช่น ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithms: GA) เป็นวิธีการเมตาฮิวริสติกที่นิยมนำมาใช้แก้ปัญหาการจัดลำดับการผลิต เพราะเป็นวิธีการที่ให้คำตอบใกล้เคียงคำตอบที่ดีที่สุด (Optimal Solution) และใช้เวลาในการค้นหาคำตอบที่ไม่มาก

ปัจจุบันปัญหาที่เกิดขึ้นในอุตสาหกรรมการบินรวมถึงปัญหาการจัดคู่นักบินมีจำนวนวัตถุประสงค์มากกว่า 3

วัตถุประสงค์ ซึ่งอาจจะขัดแย้งกันแต่ต้องพิจารณาพร้อมกัน เรียกปัญหานี้ว่าปัญหาแบบมากวัตถุประสงค์ (Many-Objective Optimization Problems: MaOPs) เนื่องจากเป็นปัญหาที่มีมิติของแกนเวกเตอร์มาก หรือมีจำนวนวัตถุประสงค์ที่มาก ทำให้ประสิทธิภาพในการค้นหาคำตอบของวิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ (A Multi-Objective Evolutionary Algorithms: MOEAs) ลดลง และแก้ปัญหา MaOPs ได้ไม่ดีเท่าที่ควร [1]

วิธีการ MOEAs ส่วนใหญ่ จะใช้การพิจารณาคำตอบโดยรวม ด้วยการจัดอันดับของคำตอบที่ไม่ถูกรับรองหรือเรียกว่าวิธีเชิงกลุ่มที่ดีที่สุด (Pareto-based Approach) วิธีการนี้มีความซับซ้อน ใช้เวลามาก และยากที่จะเกิดสมดุลของความหลากหลายรวมถึงการลู่เข้าของคำตอบ [2] เนื่องจากการมีวัตถุประสงค์ที่มาก ทำให้เกิดการกระจุกตัวของคำตอบที่เป็นคำตอบที่ดีที่สุด (1st frontier) ส่งผลให้ทิศทางของการพัฒนาคำตอบติดอยู่ในคำตอบที่ดีเฉพาะกลุ่ม (Local Optima)

การแก้ปัญหา MaOPs อาจทำได้โดยการลดจำนวนวัตถุประสงค์ลง ด้วยการรวมวัตถุประสงค์ที่ไม่ขัดแย้งกันไว้เป็นวัตถุประสงค์เดียว [3] อย่างไรก็ตามวิธีการนี้ไม่สามารถใช้ได้กับหลายปัญหาในทางปฏิบัติ วิธีการเชิงวิวัฒนาการแบบมากวัตถุประสงค์ (A Many-Objective Evolutionary Algorithms: MaOEAs) จึงถูกพัฒนาขึ้น ซึ่งหนึ่งในอัลกอริทึมที่โดดเด่นของ MaOEAs คือ วิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์โดยยึดหลักการจำแนก (A Multi-Objective Evolutionary Algorithm based on Decomposition : MOEA/D) ถูกพัฒนาโดย Qingfu and Hui [2] วิธีการดำเนินงานของ MOEA/D คือ การที่จะทำการจำแนกปัญหา MaOPs ออกเป็นปัญหาย่อย และทำการแก้ปัญหาย่อยไปพร้อมกัน ซึ่งแต่ละปัญหาย่อยนั้นจะได้คำตอบที่ดีที่สุด

เนื่องจากการแก้ปัญหา MaOPs นอกจากจะหาคำตอบที่มีจำนวนมากและไม่รู้ขอบเขตที่ชัดเจน อีกทั้งยัง

ต้องใช้เวลาในการค้นหาคำตอบ วิธีการแก้ปัญหาเชิงวิวัฒนาการแบบหลายวัตถุประสงค์โดยยึดหลักการจำแนก (MOEA/D) ซึ่งเป็นอัลกอริทึมที่มีประสิทธิภาพสูงของ MaOEAs จึงเป็นอัลกอริทึมทางเลือก เพราะอัลกอริทึมนี้สามารถค้นหากลุ่มคำตอบที่ดีที่สุด (Pareto optimal solutions) ได้ในเจเนอเรชันเดียว และไม่เกิดการกระจุกตัวของคำตอบ เนื่องจากการค้นหาคำตอบแต่ละปัญหาย่อย ทำให้มีการกำหนดทิศทางและขอบเขตของคำตอบในแต่ละปัญหาย่อยที่ชัดเจน อีกทั้งมีความซับซ้อนและใช้เวลาน้อยกว่าอัลกอริทึมอื่น เช่น วิธีเชิงพันธุกรรมแบบการจัดลำดับที่ไม่ถูกรับรอง (Non-Dominated Sorting Genetic Algorithm II: NSGA II) และวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization: PSO) รวมถึงยังให้คำตอบที่หลากหลาย ซึ่งอัลกอริทึมนี้จะให้คำตอบที่ดีและแตกต่างกับอัลกอริทึมอื่นอย่างชัดเจน เมื่อมีวัตถุประสงค์ตั้งแต่ 4 ถึง 15 วัตถุประสงค์ [2]

ดังนั้นในงานวิจัยนี้ ผู้วิจัยได้เสนอวิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์โดยยึดหลักการจำแนก (Multi-Objective Evolutionary Algorithm based on Decomposition: MOEA/D) และวิธีการเชิงพันธุกรรมแบบการจัดลำดับที่ไม่ถูกรับรอง III (Non-dominated Sorting Genetic Algorithm III: NSGA-III) โดยมีวัตถุประสงค์ในการแก้ปัญหาการจับคู่เที่ยวบินของสายการบินราคาประหยัดจำนวน 5 วัตถุประสงค์ ดังนี้ 1) เวลาว่างระหว่างเที่ยวบินน้อยที่สุด (Minimizing idle time) 2) ปรับดุลภาระงานให้เท่าเทียมกัน (Balance workload) 3) เส้นทางการบินสั้นน้อยที่สุด (Minimizing repeat flight leg) 4) ระยะทางการบินแตกต่างกันน้อยที่สุด (Minimizing the difference of nautical mile between each flight code) 5) จำนวนคู่นักบินน้อยที่สุด (Minimizing number of pair of cockpit crews)

2. ลักษณะทั่วไปของสายการบิน

สายการบินกรณีศึกษา อยู่ภายใต้การดูแลสังกัดกระทรวงคมนาคมเป็นบริษัทให้บริการขนส่งทางอากาศภายในประเทศ และต่างประเทศ จัดเป็นสายการบินราคาประหยัด ในลักษณะการดำเนินการจุดต่อจุดโดยไม่มีบริการเชื่อมต่อ (Point-to-Point) บริษัทเป็นผู้ให้บริการเที่ยวบินที่ครอบคลุมทุกภูมิภาคภายในประเทศไทย โดยในปัจจุบันได้ให้บริการจากฐานปฏิบัติการหลักที่ท่าอากาศยานนานาชาติดอนเมือง โดยมุ่งเน้นให้บริการเที่ยวบินทั้งในเส้นทางที่มีความต้องการทางการบินหนาแน่นอย่างสายการบินหลัก และเส้นทางการบินที่มีความต้องการการบินไม่มากนักอย่างเส้นทางสายการบินรอง เพื่อให้สอดคล้องกับกลยุทธ์การแข่งขันของบริษัทว่าด้วยการให้บริการเส้นทางการบินที่ครอบคลุมมากที่สุดในประเทศไทย อย่างไรก็ตาม ตั้งแต่ปี พ.ศ. 2559 เป็นต้นมา สายการบินได้เผชิญวิกฤติขาดทุนสะสม มีการยกเลิกเที่ยวบิน การดำเนินการเที่ยวบินที่ล่าช้า เนื่องจากบริษัทมีต้นทุนค่าการบริหารสูงจากค่าใช้จ่ายด้านพนักงาน การบริหารงานการปฏิบัติหน้าที่ของนักบิน ทั้งนี้การดำเนินงานของบริษัทได้แบ่งออกเป็นหลายส่วนงาน

3. ปัญหาการจับคู่เที่ยวบิน

งานวิจัยฉบับนี้ พิจารณาฟังก์ชันวัตถุประสงค์ 5 วัตถุประสงค์ โดยให้ค่าน้ำหนักความสำคัญเท่ากันหมด ดังนี้

3.1 เวลาว่างระหว่างเที่ยวบินน้อยที่สุด

ช่วงเวลาปฏิบัติหน้าที่การบิน (Flight Duty Period ; FDP) มีผลอย่างมากต่อเวลาและต้นทุนในการจับคู่เที่ยวบิน เพราะถ้าหากเวลาว่างระหว่างเที่ยวบินนาน ก็จะทำให้เกิดความล่าช้าในเที่ยวบินถัดไป นอกจากนั้นส่งผลให้ต้องเสียค่าเช่าหุลุมจอดเครื่องบินสูงขึ้นและทางสายการบินเองอาจจะต้องชดเชยในกรณีที่เที่ยวบินล่าช้า [4] แสดงสูตรคำนวณดังสมการที่ 1

$$f_1(x) = \sum_{r=1}^R \sum_{q=1}^Q DS_{qr} X_{qr} - \sum_{r=1}^R \sum_{q=1}^Q DE_{qr} X_{qr} \quad (1)$$

เมื่อ $f_1(x)$ คือ เวลาว่างระหว่างเที่ยวบิน

DS_{qr} คือ เวลาเริ่มต้นการปฏิบัติหน้าที่การบินวันที่ r เที่ยวบินที่ q

DE_{qr} คือ เวลาสิ้นสุดการปฏิบัติหน้าที่การบินวันที่ r เที่ยวบินที่ q

$$X_{qr} = \begin{cases} 1, & \text{day } r \text{ leg } q \\ 0, & \text{otherwise} \end{cases} ; X \in \{0,1\}$$

3.2 ปรับดุลภาระงานให้เท่าเทียมกัน

การจับคู่เที่ยวบินอย่างเท่าเทียมกัน คิดจากค่าเฉลี่ยผลรวมของค่าผลต่างค่าเฉลี่ยช่วงเวลาปฏิบัติหน้าที่การบินแต่ละรหัสเที่ยวบินกับค่าเฉลี่ยช่วงเวลาปฏิบัติหน้าที่การบินที่คาดหวัง แสดงสูตรคำนวณดังสมการที่ 2 และสมการที่ 3 [5]

$$f_2(x) = \frac{1}{P} \sum_{r=1}^R T_r \quad (2)$$

$$T_r = \left(1 - \frac{1}{\sum_{q=1}^Q FDP_{qr} X_{qr}}\right) ; \forall r \in R \quad (3)$$

เมื่อ T_r คือ ค่าเฉลี่ยช่วงเวลาปฏิบัติหน้าที่การบินของแต่ละวัน

P คือ จำนวนรหัสเที่ยวบินที่ได้จากการจับคู่ทั้งหมดที่เป็นไปได้

FDP_{qr} คือ ช่วงเวลาปฏิบัติหน้าที่การบินวันที่ r เที่ยวบินที่ q

3.3 เส้นทางการบินสั้นน้อยที่สุด

เส้นทางการบินสั้นน้อยที่สุดเป็นการพิจารณาถึงการกระจายรูปแบบเที่ยวบิน (Distribution of flight pattern) โดยจัดสรรรูปแบบเที่ยวบินให้แต่ละคนมีจำนวนใกล้เคียงกันให้มากที่สุด เนื่องจากการสำรวจการปฏิบัติงานของนักบินพบว่าบางคนต้องการบินเส้นทางที่แตกต่างกันตามที่ต้องการบ้าง การที่นี้วางแผนจะจัดตารางให้รองรับความต้องการของนักบินทุกคนนั้นเป็นไปได้ยาก จึงทำการเสนอให้จัดรูปแบบเที่ยวบินแก่นักบินมีจำนวนซ้ำในแต่ละสัปดาห์ให้น้อยที่สุด ไม่ให้เกิดการบินรูปแบบเที่ยวบินซ้ำมากเกินไป ทั้งนี้ ทางสายการบินมอง

ถึงว่าการบินในเส้นทางแต่ละภูมิภาคของทุกคนเท่าเทียมกันก่อให้เกิดการทำงานที่มีทักษะได้หลากหลาย แสดงสูตรคำนวณดังสมการที่ 4 [5]

$$f_3(x) = \sum_{q=1}^Q \sum_{z=1}^Z \sum_{y=1}^Y V_{yz} X_{qr} ; \forall r \in R \quad (4)$$

$$\text{เมื่อ } V_{yz} = \begin{cases} 1, \text{flight } z \text{ destination } y \\ 0, \text{otherwise} \end{cases} ; V \in \{0,1\}$$

3.4 ระยะทางการบินแตกต่างกันน้อยที่สุด

การจับคู่เที่ยวบินอย่างเท่าเทียมกัน คิดจากค่าเฉลี่ยผลรวมของผลต่างค่าเฉลี่ยระยะทางการบินแต่ละรหัสกับค่าเฉลี่ยระยะทางการบินที่คาดหวัง แสดงสูตรคำนวณดังสมการที่ 5 และสมการที่ 6

$$f_4(x) = \frac{1}{P} \sum_{r=1}^R S_r \quad (5)$$

$$S_r = (1 - \frac{1}{\sum_{q=1}^Q U_{qr} X_{qr}}) ; \forall r \in R \quad (6)$$

เมื่อ S_r คือ ค่าเฉลี่ยระยะทางการบินของแต่ละวัน

P คือ จำนวนรหัสเที่ยวบินที่ได้จากการจับคู่ทั้งหมดที่เป็นไปได้

U_{qr} คือ ระยะทางการปฏิบัติหน้าที่การบินวันที่ r ที่เที่ยวที่ q

3.5 จำนวนคู่บินน้อยที่สุด

จำนวนคู่บินน้อยที่สุดเป็นการพิจารณาหลังจากการจับคู่เที่ยวบินครบสัปดาห์ เพื่อที่จะได้วางแผนจัดสรรคู่นักบินได้เหมาะสม แสดงสูตรคำนวณดังสมการที่ 7 และสมการที่ 8

$$f_5(x) = Nw \quad (7)$$

$$Nw = \text{roundup} \left(\frac{Nr}{7}, 0 \right) \quad (8)$$

เมื่อ Nw คือ จำนวนคู่นักบินที่ทำการบิน

Nr คือ จำนวนวันที่ทำการบิน

4. การหาค่าที่เหมาะสมที่สุดสำหรับปัญหาแบบหลายวัตถุประสงค์

งานวิจัยนี้ได้นำเทคนิคการกำหนดค่าความแข็งแรงด้วยวิธีเชิงกลุ่มที่ดีที่สุด ซึ่งเป็นการจัดอันดับของคำตอบแบบพาเรโต (Pareto Ranking Approach) และวิธีเทบิเชฟฟ์ ซึ่งเป็นการจัดอันดับของคำตอบด้วยวิธีการให้ค่าน้ำหนักของแต่ละฟังก์ชันวัตถุประสงค์เพื่อเป็นทางเลือกโดยคำตอบที่ได้จากทั้งสองวิธีนี้จะอยู่ในรูปของกลุ่มคำตอบที่ดีที่สุด และจะไม่ถูกรอบงำจากคำตอบอื่น (Non-Dominated Solution) แต่ไม่สามารถระบุได้ว่าคำตอบใดดีกว่ากัน [6] ซึ่งกลุ่มคำตอบที่ดีมีดังนี้

1. วิธีเชิงกลุ่มที่ดีที่สุด (Pareto-Based Approach)

ดังสมการที่ 9

$$\text{minimize} : \{f_1(\bar{x}), f_2(\bar{x}), \dots, f_k(\bar{x})\} \quad (9)$$

เมื่อ $\bar{x}$ คือ เวกเตอร์ตัวแปรตัดสินใจ (สตริงคำตอบ)

$f_k(x)$ คือ ฟังก์ชันวัตถุประสงค์ที่ k

ของเวกเตอร์ $\bar{x}$

กรณีหาคำตอบที่น้อยที่สุด เวกเตอร์ของตัวแปรตัดสินใจ $\bar{x}$ จะหาคำตอบที่ดีกว่าตัวแปรตัดสินใจ y ก็ต่อเมื่อ $f_k(x) \leq f_k(y)$ ทุกค่า และ $f_i(x) < f_i(y)$ อย่างน้อย 1 ค่า โดยที่ $k = 1, 2, \dots, \text{Objective}$

2. วิธีเทบิเชฟฟ์ (Tchebycheff Approach)

ดังสมการที่ 10

$$\text{minimize } g^{te}(\bar{x}_j | \bar{\lambda}_j, z^*) = \max_{1 \leq k \leq \text{obj}} \{ \lambda_{jk} | f_k(\bar{x}_j) - z_k^* | \} \quad (10)$$

เมื่อ $g^{te}(\bar{x}_j)$ คือ ผลต่างมากที่สุดระหว่างค่า

วัตถุประสงค์ของสตริง j กับค่า

เป้าหมายของเวกเตอร์ค่าน้ำหนัก $\bar{\lambda}_j$

ของปัญหาย่อยที่ j

z_k^* คือ ค่าเวกเตอร์เป้าหมาย โดยที่ $z_k^* = \{z_1^*, z_2^*, \dots, z_{obj}^*\}$ กรณีนี้นิยามค่าของคำตอบที่น้อยที่สุด $z_k^* = \min\{f_k(\bar{x})\}$ กรณีนี้นิยามค่าของคำตอบที่มากที่สุด $z_k^* = \max\{f_k(\bar{x})\}$
 $\bar{\lambda}_j$ คือ เวกเตอร์ค่าน้ำหนักของปัญหาย่อยที่ j โดยที่ $\bar{\lambda}_j = \{\lambda_{j1}, \lambda_{j2}, \dots, \lambda_{jobjective}\}$

5. อัลกอริทึม MOEA/D

วิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์โดยยึดหลักการจำแนก (A Multi-Objective Evolutionary Algorithm based on Decomposition: MOEA/D) มีหลักการในการเพิ่มจำนวนคำตอบที่ดีที่อยู่บนเส้นพารेटอ (Pareto front) ซึ่งจะนำไปสู่การหาคำตอบที่ดีสำหรับปัญหาที่มีหลายวัตถุประสงค์ได้ในเวลาอันสั้น ทั้งนี้มีแนวคิดที่จะแก้ปัญหาด้วยการวิวัฒนาการคำตอบที่มีไปพร้อมกันในทุกวัตถุประสงค์ โดยเริ่มจากความสัมพันธ์ของสตริงข้างเคียง (Neighborhood) ซึ่งใช้ระยะห่างระหว่างผลรวมค่าสัมประสิทธิ์ของเวกเตอร์ $\bar{\lambda}_j$ ด้วยวิธีระยะห่างยูคลิดีเนียน (Euclidian) ในแต่ละฟังก์ชันวัตถุประสงค์เป็นตัวเลือกสตริงคำตอบที่จะนำมาสุ่มเพื่อพัฒนาคำตอบ การพัฒนาคำตอบในประชากร (Population) แต่ละตัวใช้การครอสโอเวอร์ (Crossover) ของวิธีเชิงพันธุกรรม (Genetic Algorithm) เพื่อพัฒนาคำตอบ (Population) โดยจะให้คำตอบรุ่นลูก (Offspring) ที่มีความคล้ายคลึงกับคำตอบรุ่นเดิม (Parent) ที่เป็นค่าที่ดี ทำให้เกิดการพัฒนาทุกๆ สตริงคำตอบ [2] และทำการประเมินคำตอบที่ดีด้วยการกำหนดค่าความแข็งแรง (Fitness) แล้วคำนวณค่าเทบชีเชฟฟ์ (Tchebycheff Approach) เพื่อเลือกคำตอบที่ดีมาแทนที่ในแต่ละจุดตั้งต้น จากนั้นจะนำสตริงรุ่นลูกที่ดีที่สุดหลังจากการเทียบค่าเทบชีเชฟฟ์แล้วทำการจัดเก็บคำตอบที่ดีที่สุด (Elitist) ในแต่ละเจเนอเรชันด้วยวิธีการจัดลำดับแบบไม่ถูกครอบงำ (Non-Dominated Sorting) ขั้นตอนการทำงานของ MOEA/D ในการหาคำตอบของปัญหาหลายวัตถุประสงค์ มีดังต่อไปนี้

5.1 สร้าง λ_{jk} และคำนวณ Euclidean Distances

สร้างค่าน้ำหนัก λ_{jk} โดยใช้ซิมเพล็กซ์แลตทิซดีไซน์ (Simplex lattice design) เป็นเมทริกซ์ขนาดเท่ากับจำนวนประชากร (N) x จำนวนวัตถุประสงค์ (Obj) โดยงานวิจัยนี้ได้ใช้โปรแกรมมินิแทบ 17 (Minitab17) ดังตารางที่ 1

คำนวณระยะห่างระหว่างจุดพิคัดของเวกเตอร์น้ำหนัก λ_{jk} ทั้งหมด (Euclidean distances) ดังสมการที่ 11 และจัดเก็บสตริงข้างเคียงของแต่ละปัญหาย่อยที่ i ลงในเซต $B(i)$

$$d_{jl} = \sqrt{\sum_{k=1}^{obj} (\lambda_{jk} - \lambda_{lk})^2}; \forall j \in N, \forall l \in N \quad (11)$$

เมื่อ d_{jl} คือระยะห่างระหว่างจุด (Euclidean distances) ของค่าน้ำหนักสตริงที่ j กับสตริงที่ l โดยที่ $j = 1, 2, \dots, N$ และ $l = 1, 2, \dots, N$

เมื่อได้ค่าระยะห่างระหว่างจุดของค่าน้ำหนักแล้ว ให้ทำการกำหนดสตริงข้างเคียง โดยเรียงค่าระยะห่างของค่าน้ำหนักจากน้อยไปมากและตัดให้เหลือเท่ากับจำนวนสตริงข้างเคียงที่กำหนดไว้ข้างต้น จะได้สตริงข้างเคียงของแต่ละปัญหาย่อย ดังตารางที่ 2

ตารางที่ 1 ซิมเพล็กซ์แลตทิซดีไซน์

ปัญหาย่อย j	ค่าน้ำหนัก λ_{jk} ของแต่ละวัตถุประสงค์				
	1	2	3	4	5
1	0	0	0	0.5	0.5
2	0	1	0	0	0
3	0.5	0.5	0	0	0
4	1	0	0	0	0
5	0	0	1	0	0
6	0	0.5	0.5	0	0
7	0	0	0	0	1
8	0	0.5	0	0	0.5
9	0.5	0	0	0	0.5
10	0	0.5	0	0.5	0

ตารางที่ 1 ซิมเพล็กซ์แลตทิซดีไซน์ (ต่อ)

ปัญหา ย่อย j	ค่าน้ำหนัก λ_{jk} ของแต่ละวัตถุประสงค์				
	1	2	3	4	5
11	0	0	0.5	0.5	0
12	0	0	0.5	0	0.5
13	0.5	0	0	0.5	0
14	0	0	0	1	0
15	0.5	0	0.5	0	0

5.2 สร้างสตริงคำตอบเริ่มต้น

การสร้างสตริงคำตอบเริ่มต้นจะใช้วิธีสุ่มเลือกแบบอิสระ โดยสุ่มเท่ากับจำนวนประชากรที่ความยาวสตริงเท่ากับจำนวนเที่ยวบินผลิตภัณฑ์ทั้งหมด แสดงตัวอย่างสตริงจากการสุ่มดังตารางที่ 3

5.3 ประเมินค่าวัตถุประสงค์สตริงคำตอบเริ่มต้น

การคำนวณค่าฟังก์ชันวัตถุประสงค์ของแต่ละสตริงในตัวอย่างนี้ สามารถคำนวณได้จากสมการฟังก์ชันวัตถุประสงค์ในหัวข้อที่ 3 แสดงค่าฟังก์ชันวัตถุประสงค์ของแต่ละสตริงในตัวอย่างดังตารางที่ 4

ตารางที่ 2 เซตสตริงข้างเคียงของแต่ละปัญหาย่อย

ปัญหา ย่อย	สตริงข้างเคียง			
1	1	7	8	9
2	2	3	6	8
3	3	2	4	6
4	4	3	9	13
5	5	6	11	12
6	6	2	3	5
7	7	1	8	9
8	8	1	2	3

ตารางที่ 2 เซตสตริงข้างเคียงของแต่ละปัญหาย่อย (ต่อ)

ปัญหา ย่อย	สตริงข้างเคียง			
9	9	1	3	4
10	10	1	2	3
11	11	1	5	6
12	12	1	5	6
13	13	1	3	4
14	14	1	10	11
15	15	3	4	5

5.4 การพัฒนาคำตอบ

พัฒนาคำตอบและปรับปรุงสตริงคำตอบด้วยวิธีครอสโอเวอร์ คือ การนำสตริงคำตอบรุ่นพ่อแม่มาแลกเปลี่ยนบางส่วนของสตริงซึ่งกันและกัน เพื่อให้เกิดสตริงรุ่นใหม่หรือสตริงคำตอบรุ่นลูก โดยในงานวิจัยนี้จะใช้วิธีการครอสโอเวอร์แบบพาเซิลแมพครอสโอเวอร์ 1 ตำแหน่ง (One-Point Partially-mapped crossover: PMX) แสดงสตริงคำตอบรุ่นลูกดังตารางที่ 5

5.5 ประเมินฟังก์ชันวัตถุประสงค์สตริงคำตอบรุ่นลูก

ประเมินฟังก์ชันวัตถุประสงค์ของสตริงคำตอบรุ่นลูกดังตารางที่ 6

5.6 รวบรวมค่าวัตถุประสงค์สูงสุด และต่ำสุด

รวบรวมค่าวัตถุประสงค์สูงสุด (ff_k^{max}) และต่ำสุด (ff_k^{min}) หลังจากที่ได้ค่าฟังก์ชันวัตถุประสงค์ของทั้งสตริงคำตอบรุ่นพ่อแม่และสตริงคำตอบรุ่นลูกแล้ว โดยงานวิจัยนี้ได้มีการป้องกันส่วนหารเท่ากับ 0 ซึ่งจะให้คำตอบที่เป็นไปไม่ได้ โดยการบวกค่าคงที่ 0.0005 ให้กับค่า ff_k^{max} ดังตารางที่ 7

5.7 เปรียบเทียบค่าเป้าหมาย (z_k^*)

เมื่อได้ค่าสูงสุดและต่ำสุดของฟังก์ชันวัตถุประสงค์โดยกำหนดให้ $z_k^* = \min(f_k(x))$ ดังตารางที่ 8

5.8 นอร์มัลไลซ์ (Normalization)

นอร์มัลไลซ์ (Normalization) คำนวณค่าฟังก์ชันวัตถุประสงค์ของสตริงข้างเคียงในปัญหาย่อย สตริงคำตอบรุ่นลูก และค่าเป้าหมาย ดังตารางที่ 9

5.9 เปรียบเทียบสตริงคำตอบรุ่นลูกด้วยวิธีการเทบปีเซฟฟ์

เปรียบเทียบสตริงคำตอบรุ่นลูกด้วยวิธีการเทบปีเซฟฟ์ ซึ่งจะใช้ค่าน้ำหนัก λ_{jk} ของแต่ละปัญหาย่อย โดยตัวอย่างนี้จะแสดงเพียงปัญหาย่อยที่ 1 โดยสตริงคำตอบที่ให้ค่า $g(y|\lambda_1, z^N)$ ที่ต่ำที่สุดในสตริงคำตอบรุ่นลูกแรก จะถูกเลือกมาทำขั้นตอนต่อไป ดังตารางที่ 10

5.10 การหาค่าที่ดีที่สุด

ขั้นตอนนี้จะใช้วิธีการเทบปีเซฟฟ์ในการค้นหาคำตอบที่ดีที่สุดของแต่ละค่าน้ำหนัก โดยจะมีการประเมินทีละสตริงข้างเคียงของปัญหาย่อยเท่านั้น โดยมีลำดับการประเมินดังนี้

1. ตรวจสอบเงื่อนไขจำนวนการแทนที่สตริงคำตอบรุ่นลูกในปัญหาย่อยว่ามีค่าน้อยกว่าจำนวนการแทนที่

สูงสุดหรือไม่ ($n_r < NR$) ถ้าจำนวนการแทนที่สตริงคำตอบรุ่นลูกในปัญหาย่อยมีค่ามากกว่าหรือเท่ากับจำนวนการแทนที่สูงสุด ($n_r \geq NR$) ให้หยุดการประเมิน

2. คำนวณค่าเทบปีเซฟฟ์ของสตริงเดิม $g(\bar{x}_j|\lambda_1, z^N)$ และสตริงรุ่นลูก $g(y|\lambda_1, z^N)$ ที่ค่าน้ำหนักเดียวกัน

3. เปรียบเทียบค่าเทบปีเซฟฟ์ที่มากที่สุดของสตริงเดิม $g(\bar{x}_j|\lambda_1, z^N)$ กับสตริงรุ่นลูก $g(y|\lambda_1, z^N)$ โดยถ้าค่าเทบปีเซฟฟ์ของสตริงเดิมน้อยกว่าค่าเทบปีเซฟฟ์ของสตริงรุ่นลูก $g(\bar{x}_j|\lambda_1, z^N) < g(y|\lambda_1, z^N)$ จะแทนที่ค่าฟังก์ชันวัตถุประสงค์และค่าเทบปีเซฟฟ์ของสตริงเดิมเท่ากับค่าสตริงรุ่นลูก แต่ถ้าค่าเทบปีเซฟฟ์ของสตริงเดิมมากกว่าหรือเท่ากับค่าเทบปีเซฟฟ์ของสตริงรุ่นลูก

$g(\bar{x}_j|\lambda_1, z^N) \geq g(y|\lambda_1, z^N)$ จะเปรียบเทียบค่าต่อมา ดังตารางที่ 11

5.11 การหยุดอัลกอริทึม

หยุดการพัฒนาคำตอบ เมื่อครบจำนวนเจเนอเรชันตามที่กำหนด

ตารางที่ 3 สตริงคำตอบเริ่มต้น

ปัญหาย่อย	สตริงคำตอบ	ลำดับสตริง									
1	1	6	10	8	1	3	2	9	4	5	7
2	2	2	8	10	9	1	5	4	6	3	7
3	3	2	8	10	9	1	5	4	6	3	7
4	4	5	4	10	1	3	2	9	8	7	6
5	5	3	8	10	2	1	5	4	6	9	7
6	6	2	10	8	1	3	5	4	6	9	7
7	7	2	10	8	1	3	5	4	6	9	7
8	8	2	10	8	1	3	5	4	6	9	7
9	9	2	10	8	1	3	5	4	6	9	7
10	10	5	10	4	1	6	8	9	3	2	7
11	11	5	10	4	1	2	8	3	6	9	7
12	12	3	5	10	1	2	8	4	6	9	7
13	13	5	6	7	1	2	8	9	10	3	4
14	14	5	2	7	1	6	8	9	10	3	4
15	15	5	10	3	1	2	8	4	6	9	7

ตารางที่ 4 ค่าฟังก์ชันวัตถุประสงค์ของสตริงคำตอบเริ่มต้น

ปัญหาย่อย	สตริงคำตอบ	ฟังก์ชันวัตถุประสงค์				
		1	2	3	4	5
1	1	0.8633	0.0451	0.0476	0	2
2	2	3.4433	0.0232	0.0238	2	1
3	3	3.4433	0.0232	0.0238	2	1
4	4	0.8633	0.0451	0.0476	0	2
5	5	3.4433	0.0232	0.0238	2	1
6	6	1.7867	0.0232	0.0238	2	1
7	7	1.7867	0.0232	0.0238	2	1
8	8	1.7867	0.0232	0.0238	2	1
9	9	1.7867	0.0232	0.0238	2	1
10	10	1.5233	0.0444	0.0476	0	2
11	11	3.8533	0.0232	0.0238	1	1
12	12	1.7867	0.0232	0.0238	2	1
13	13	0.8633	0.0451	0.0476	0	2
14	14	0.8633	0.0451	0.0476	0	2
15	15	1.2033	0.0443	0.0476	1	2

ตารางที่ 5 สตริงคำตอบรุ่นลูกในเจเนอเรชันที่ 1 ปัญหาย่อยที่ 1

สตริงรุ่นลูก	ปัญหาย่อย	ลำดับสตริง									
1	1	6	10	8	1	3	2	4	5	9	7
2	2	2	10	8	1	3	5	9	4	6	7

ตารางที่ 6 ฟังก์ชันวัตถุประสงค์ของสตริงคำตอบรุ่นลูก

สตริงรุ่นลูก	1	2	3	4	5
1	0.8633	0.0451	0.0476	0	2
2	1.2033	0.0443	0.0476	1	2

ตารางที่ 7 ค่าสูงสุดต่ำสุดของสตริงคำตอบทั้งหมด

ค่าสูงสุด/ต่ำสุด	1	2	3	4	5
ff_k^{max}	3.8538	0.0456	0.0481	2.0005	2.0005
ff_k^{min}	0.8638	0.0237	0.0243	0.0005	1.0005

ตารางที่ 8 ค่าเป้าหมายของสตริงคำตอบรุ่นลูก

ค่าเป้าหมาย	1	2	3	4	5
z_k^*	0.8638	0.0237	0.0243	0.0005	1.0005

ตารางที่ 9 ค่าฟังก์ชันวัตถุประสงค์ของสตริงคำตอบหลังทำการนอร์มัลไลซ์

สตริงคำตอบ	1	2	3	4	5
1	0	0.9937	0.9958	0	0.9999
7	0.8628	0.0013	0	0.9999	0
8	0.8628	0.0013	0	0.9999	0
9	0.8628	0.0013	0	0.9999	0
สตริงรุ่นลูก1	0	0.9937	0.9958	0	0.9999
สตริงรุ่นลูก2	0.1137	0.9601	0.9956	0.4999	0.9999

ตารางที่ 10 ค่าเทบปีเซฟฟ์ของสตริงคำตอบรุ่นลูก

สตริงคำตอบรุ่นลูก	1	2	3	4	5	$g(y \lambda_1, z^N)$
1	0	0	0	0	0.4999	0.4999
2	0	0	0	0	0.9999	0.9999

ตารางที่ 11 ค่าฟังก์ชันวัตถุประสงค์ของสตริงคำตอบหลังจบเจเนอเรชันที่ 1

ปัญหาย่อย	สตริง	1	2	3	4	5
1	1	0.8633	0.0451	0.0476	0	2
2	2	3.4433	0.0232	0.0238	2	1
3	1.3	0.8633	0.0451	0.0476	0	2
4	4	0.8633	0.0451	0.0476	0	2
5	5	3.4433	0.0232	0.0238	2	1
6	6	1.7867	0.0232	0.0238	2	1
7	1.5	3.8533	0.0232	0.0238	1	1
8	8	1.7867	0.0232	0.0238	2	1
9	1.7	0.8633	0.0451	0.0476	0	2
10	10	1.5233	0.0444	0.0476	0	2
11	1.8	1.7867	0.0232	0.0238	2	1
12	1.9	1.5233	0.0444	0.0476	0	2
13	13	0.8633	0.0451	0.0476	0	2
14	1.1	1.7867	0.0232	0.0238	2	1
15	1.11	1.2033	0.0443	0.0476	1	2

6. การทดลอง

6.1 ปัญหาที่ใช้ในการทดลอง

งานวิจัยนี้ได้แบ่งปัญหาการทดลอง ออกเป็นปัญหาขนาดเล็ก ขนาดกลาง และขนาดใหญ่ รวมทั้งสิ้น 10 ปัญหา ซึ่งมีความแตกต่างกันของจำนวนคู่เที่ยวบิน ดังตารางที่ 12

6.2 การออกแบบการทดลอง

เนื่องจากปัญหามีสองข้อจำกัดที่หลากหลายแตกต่างกัน จึงจำเป็นต้องทำการกำหนดค่าพารามิเตอร์ของอัลกอริทึมให้เหมาะสมในแต่ละปัญหา เพื่อให้ได้คำตอบที่ดีที่สุด โดยจะทดสอบพารามิเตอร์ของแต่ละปัญหาและแต่ละอัลกอริทึมด้วยวิธีการออกแบบการทดลอง (Design of Experiment; DOE) ซึ่งงานวิจัยนี้จะทำการทดสอบ 2 พารามิเตอร์ พารามิเตอร์ละ 3 ระดับ และทำซ้ำ 3 ครั้ง โดยที่พารามิเตอร์ของ MOEA/D ได้แก่ จำนวนสตริงข้างเคียง (Nb) และ จำนวนการแทนที่คำตอบ (Nr) ส่วนพารามิเตอร์ของ NSGAIII ได้แก่ อัตราการรอ (Pc) และอัตราการมิวเทชัน (Pm) ดังตารางที่ 13

ตารางที่ 12 ปัญหาที่ใช้ในการทดลอง

ปัญหาการทดลอง			ความยาวสตริง(คู่เที่ยวบิน)
เล็ก	Set1	S1	10
	Set2	S2	20
	Set3	S3	30
	Set4	S4	40
กลาง	Set5	S5	50
	Set6	S6	60
	Set7	S7	70
	Set8	S8	76
ใหญ่	Set9	S9	100
	Set10	S10	141

ตารางที่ 13 การออกแบบการทดลอง

อัลกอริทึม	พารามิเตอร์	ระดับปัจจัย	
MOEA/D	Nb	2	5
	Nr	10	20
NSGAIII	Pc	0.1	0.4
	Pm	0.6	0.9

6.3 ตัวชี้วัดสมรรถนะอัลกอริทึม

ตัวชี้วัดสมรรถนะที่ใช้ในงานวิจัยนี้มีทั้งหมด 6 ตัวชี้วัด คือ

1. การเข้าสู่กลุ่มคำตอบที่แท้จริง (GD)
2. การเข้าสู่และความหลากหลายของกลุ่มคำตอบ (IGD)
3. การกระจายของกลุ่มคำตอบที่ได้ (Spread)
4. อัตราส่วนของคำตอบที่ไม่ถูกรอรับเทียบกับกลุ่มคำตอบที่อัลกอริทึมหาได้ (RNDs1)
5. อัตราส่วนของคำตอบที่ไม่ถูกรอรับเทียบกับกลุ่มคำตอบที่ดีที่สุด (RNDs2)
6. เวลาที่ใช้ในการค้นหาคำตอบ (CPU Time)

7. ผลการทดลอง

งานวิจัยนี้เปรียบเทียบอัลกอริทึม 2 ตัว ได้แก่ MOEA/D และ NSGAIII ด้วยวิธีการ 2-Sample t ในการทดสอบทางสถิติ เพื่อเปรียบเทียบคำตอบสุดท้ายที่หาได้จากแต่ละอัลกอริทึม ดังตารางที่ 14 ซึ่งจะพบว่า MOEA/D โดยรวมมีสมรรถนะด้านการเข้าสู่กลุ่มคำตอบที่แท้จริง (GD) ด้านการเข้าสู่และความหลากหลายของกลุ่มคำตอบ (IGD) ด้านการกระจายของกลุ่มคำตอบที่ได้ (Spread) ด้านอัตราส่วนของคำตอบที่ไม่ถูกรอรับเทียบกับกลุ่มคำตอบที่อัลกอริทึมหาได้ (RNDs1) ด้านอัตราส่วนของคำตอบที่ไม่ถูกรอรับเทียบกับกลุ่มคำตอบที่ดีที่สุด (RNDs2) และด้านเวลาที่ใช้ในการค้นหาคำตอบ (CPU Time) ดีกว่า NSGAIII ในปัญหาขนาดเล็กถึงกลาง ส่วนใน

ปัญหาขนาดใหญ่ยังไม่สรุปได้ว่าแตกต่างกันอย่างชัดเจน ในด้านการกระจายของกลุ่มคำตอบที่ได้ (Spread) ด้านอัตราส่วนของคำตอบที่ไม่ถูกรอรับเทียบกับกลุ่มคำตอบที่อัลกอริทึมหาได้ (RNDs1) ด้านอัตราส่วนของคำตอบที่ไม่ถูกรอรับเทียบกับกลุ่มคำตอบที่ดีที่สุด (RNDs2) แต่

โดยรวมถือว่า MOEA/D มีสมรรถนะที่ดีกว่า NSGAIII ดังตารางที่ 17

จากผลการทดลองสามารถสรุปได้ว่า MOEA/D มีสมรรถนะในการแก้ปัญหาดีกว่า NSGAIII ในทุกด้านตั้งแต่ปัญหาขนาดเล็กถึงใหญ่

ตารางที่ 14 ผลการทดลอง

Algorithm	Parameter Set									
	S1	S2	S3	S4	M1	M2	M3	M4	L1	L2
Generational Distance (GD)										
MOEA/D	0.62	0.069	0.101	0.193	0.352	0.568	0.206	0.384	0.291	0.362
NSGAIII	0.501	0.111	0.432	0.855	1.005	0.635	1.307	1.074	0.963	0.43
Inverted Generational Distance (IGD)										
MOEA/D	0.257	0.131	0.169	0.227	0.23	0.553	0.185	0.289	0.219	0.249
NSGAIII	0.759	0.177	1.46	1.332	0.805	1.103	0.935	0.851	0.761	0.515
Spread										
MOEA/D	0.032	0.06	0.01	0.015	0.018	0.018	0.066	0.104	0.03	0.073
NSGAIII	0.043	0.071	0.068	0.054	0.122	0.078	0.106	0.134	0.143	0.024
RNDs1										
MOEA/D	0.332	0.348	0.271	0.261	0.263	0.24	0.229	0.098	0.145	0.049
NSGAIII	0	0.055	0	0	0	0.099	0.057	0.057	0.068	0.126
RNDs2										
MOEA/D	0.277	0.163	0.115	0.175	0.181	0.177	0.124	0.09	0.095	0.043
NSGAIII	0	0.011	0	0	0	0.006	0.017	0.004	0.016	0.064
CPU Time										
MOEA/D	877	1490	3950	3115	4018	4148	6145	5355	9112	10873
NSGAIII	894	1524	4705	3146	4033	4150	6164	5522	9205	10990

8. สรุป

งานวิจัยนี้เสนอการแก้ปัญหาการจับคู่เที่ยวบินของสายการบินราคาประหยัดแบบมากวัตถุประสงค์ จัดเป็นปัญหาแบบมากวัตถุประสงค์ ซึ่งมีจำนวน 5 วัตถุประสงค์ ได้แก่ เวลาว่างระหว่างเที่ยวบินน้อยที่สุด ปรับลดภาระงานให้เท่าเทียมกัน เส้นทางการบินสั้นน้อยที่สุด ระยะทางการบินแตกต่างกันน้อยที่สุด และจำนวนค่านักบินน้อยที่สุด โดยเสนอวิธีการเชิงวิวัฒนาการแบบหลายวัตถุประสงค์โดยยึดหลักการจำแนก (MOEA/D) และวิธีเชิงพันธุกรรมแบบการจัดลำดับที่ไม่ถูกครอบงำ III (NSGA-III) ผลการวิจัยพบว่า MOEA/D มีสมรรถนะด้านการลู่เข้าสู่กลุ่มคำตอบที่แท้จริง ด้านการลู่เข้าและความหลากหลายของกลุ่มคำตอบ ด้านการกระจายของกลุ่มคำตอบที่ได้ ด้านอัตราส่วนของคำตอบที่ไม่ถูกครอบงำเทียบกับกลุ่มคำตอบที่อัลกอริทึมหาได้ ด้านอัตราส่วนของคำตอบที่ไม่ถูกครอบงำเทียบกับกลุ่มคำตอบที่ดีที่สุด และด้านเวลาที่ใช้ในการค้นหาคำตอบ ดีกว่า NSGAIII

แนวทางการศึกษาในอนาคตอาจปรับเปลี่ยนวิธีการพัฒนาคำตอบของ MOEA/D เพื่อลดเวลาค้นหาคำตอบและมีคำตอบที่ดีขึ้น และเพื่อที่จะสามารถประยุกต์ใช้ได้ใกล้เคียงกับปัญหาจริงมากที่สุดอาจจะนำปัญหาด้านนโยบายของสายการบินมาร่วมพิจารณาด้วย

9. กิตติกรรมประกาศ

โครงการวิจัยนี้ได้รับทุนอุดหนุนการวิจัยจากจุฬาลงกรณ์มหาวิทยาลัย

10. เอกสารอ้างอิง

- [1] 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.
- [2] Q. Zhang and H. Li MOEA/D: "A Multi-objective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no.6, pp. 712-731, 2007.
- [3] 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.
- [4] M. Deveci and N. C. Demirel, "Evolutionary algorithms for solving the airline crew pairing problem. *Computers & Industrial Engineering*, vol. 115, pp. 389-406, 2018.
- [5] K. Arayikanon and P. Chutima, "Solving cockpit crew scheduling problem of a low-cost airline using metaheuristics," *AIP Conference Proceedings*, 2018, vol. 2044, no.1.
- [6] Wattanapornprom, W., Olanvithchai, P., Chutima, P., & Chongstitvatana, P. (2009, May). Multi- objective combinatorial optimisation with coincidence algorithm. In *2009 IEEE Congress on Evolutionary Computation* (pp. 1675-1682). IEEE.