

การพัฒนาวิธีวิวัฒนาการโดยใช้ผลต่างสำหรับปัญหาการจัดสมดุลสายการประกอบ แบบอย่างง่าย ประเภทที่ 1

Development of Differential Evolution Algorithm for Simple Assembly Line Balancing Problem Type 1

กฤต จันทรสมัย^{1*} วิศิษฐ์ จันทรชัน¹ และ พูนธนะ ศรีสระคู²

¹สาขาวิชาวิศวกรรมเครื่องกล, คณะเกษตรศาสตร์และเทคโนโลยี, มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี วิทยาเขตสุรินทร์
นอกเมือง เมืองสุรินทร์ จังหวัดสุรินทร์ 32000

²สาขาวิชาวิศวกรรมการจัดการอุตสาหกรรม, คณะเทคโนโลยีอุตสาหกรรม, มหาวิทยาลัยราชภัฏบุรีรัมย์ ในเมือง เมือง
บุรีรัมย์ จังหวัดบุรีรัมย์ 31000

Krit Chantarasamai^{1*} Visit Junchuan¹ and Poontana Sresracoo²

¹Department of Mechanical Engineering, Faculty of Agriculture and Technology, Rajamangala University of Technology
Isan Surin Campus, Nok Mueang, Mueang Surin, Surin, 32000, Thailand

²Department of Industrial Management Engineering, Faculty of Industrial Technology, Buriram Rajabhat University,
Naimuang, Muang Buriram, Buriram, 31000, Thailand

*Corresponding Author E-mail: krit.ch@rmuti.ac.th

Received: Dec 12, 2023; Revised: May 01, 2024; Accepted: May 20, 2024

บทคัดย่อ

ปัญหาการจัดสมดุลสายการประกอบแบบอย่างง่าย ประเภทที่ 1 (Simple Assembly Line Balancing Problem Type 1: SALBP-1) เป็นปัญหาที่สำคัญในอุตสาหกรรมที่นิยมใช้เป็นจำนวนมากเนื่องจากจัดกระบวนการผลิตได้ง่ายและมีประสิทธิภาพ งานวิจัยนี้จึงได้ทำการพัฒนาวิธีวิวัฒนาการโดยใช้ผลต่าง (Development of Differential Evolution Algorithm: DDE) โดยใช้ลำดับงานที่มองย้อนกลับ (Backward) เข้ามาช่วยในการจัดกระบวนการผลิตเพื่อหาจำนวนสถานีงานที่ต่ำที่สุด และวัดประสิทธิภาพของวิธีการโดยการเปรียบเทียบกับวิธีวิวิธวิธี ได้แก่ การจัดงานที่ใช้เวลามากที่สุด (Longest Operation Time: LOT), การจัดงานที่มีงานตามหลังมากที่สุด (Most Following Tasks: MFT), การจัดงานที่มีค่าน้ำหนักมากที่สุด (Ranked Positional Weight: RPW), การจัดงานที่ใช้เวลาน้อยที่สุด (Shortest Operation Time: SOT), การจัดงานที่มีงานตามหลังน้อยที่สุด (Fewest Following Tasks: FFT), วิธีอาณานิคมมด (Ant Colony Optimization: ACO), วิธีวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution: DE) และวิธีเชิงพันธุกรรมภูมิคุ้มกัน (Immune Genetic Algorithm: IGA) พบว่าวิธี DDE สามารถค้นหาคำตอบได้ดีกว่าวิธี LOT, วิธี MFT, วิธี RPW, วิธี SOT, วิธี FFT และได้คำตอบที่เท่ากันในทุกปัญหาเมื่อเปรียบเทียบกับวิธี ACO, วิธี DE และวิธี IGA แต่วิธี DDE ใช้เวลาในการค้นหาคำตอบที่รวดเร็วกว่า

คำสำคัญ: การจัดสมดุลสายการประกอบแบบอย่างง่าย, วิธีวิวัฒนาการโดยใช้ผลต่าง, วิธีวิวิธวิธี

Abstract

The Simple Assembly Line Balancing Problem Type 1 (SALBP-1) is a widely embraced method in the industry for its simplicity in organizing production processes and enhancing efficiency. Consequently, a Differential Evolution Algorithm (DDE) using a

backward task sequence was developed in this study to assist in production process management by determining the optimal number of stations. The efficacy of this method was assessed by juxtaposing it with heuristic approaches, including Longest Operation Time (LOT), Most Following Tasks (MFT), Ranked Positional Weight (RPW), Shortest Operation Time (SOT), Fewest Following Tasks (FFT), Ant Colony Optimization (ACO), Differential Evolution (DE), and Immune Genetic Algorithm (IGA). The findings reveal that DDE outperforms LOT, MFT, RPW, SOT, and FFT in discovering superior solutions and consistently matches solutions achieved by ACO, DE, and IGA methods across all problems. Notably, the DDE method exhibits a shorter time frame for solution discovery.

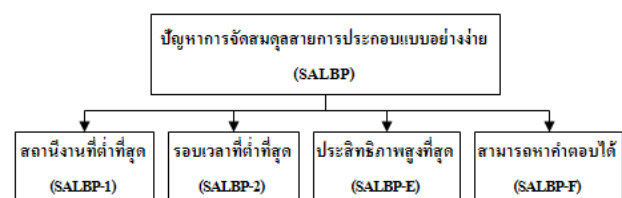
Keywords: Simple Assembly Line Balancing Problem, Differential Evolution Algorithm, Heuristic

1. บทนำ

ปัจจุบันสภาวะของธุรกิจอุตสาหกรรมทั่วโลกมีการแข่งขันกันมากขึ้น เนื่องจากมีการนำเทคโนโลยีที่ทันสมัยเข้ามาพัฒนาระบบการผลิต เพื่อให้ทำให้บริษัทของตนเองอยู่รอดและตอบสนองความต้องการของลูกค้ามากที่สุด ไม่ว่าจะเป็น การส่งมอบ วัตถุดิบ สินค้า หรือบริการ ให้ถึงมือลูกค้าทันเวลาที่ต้องการมากที่สุดหรือที่เรียกว่า ระบบทันเวลาพอดี (Just-in-time: JIT) ถ้าบริษัทไหนมีการตอบสนองต่อลูกค้าได้ดีและต่อเนื่องก็จะทำให้บริษัทนั้นคงอยู่ได้นานและมั่นคงมากยิ่งขึ้น ดังนั้นการบริหารจัดการที่ดีจึงมีความสำคัญต่อบริษัท โดยในปัจจุบันแต่ละบริษัทได้มีการพัฒนาและปรับปรุงผลิตภัณฑ์อย่างต่อเนื่องอยู่ตลอดเวลา ซึ่งเมื่อมีการพัฒนาและปรับปรุงผลิตภัณฑ์จะทำให้กระบวนการผลิตไม่ว่าจะเป็น รอบเวลาการผลิต (Cycle time) ลำดับงาน (Precedence Diagram) มีการเปลี่ยนแปลง เมื่อกระบวนการผลิตเปลี่ยนการจัดสมดุลสายการประกอบ (Assembly line Balancing) ก็ต้องมีการปรับกระบวนการประกอบใหม่เช่นกัน ซึ่งปัญหาการจัดสมดุลสายการประกอบ (Assembly line Balancing) จึงเป็นปัญหาที่สำคัญต่ออุตสาหกรรมที่มีการประกอบค่อนข้างมาก เช่น รถยนต์ แอร์ มอเตอร์ไซค์ เป็นต้น

การจัดสมดุลสายการประกอบ เป็นการวางแผนการผลิตรูปแบบหนึ่งที่มีขอบหมายงานให้กับแต่ละสถานีเพื่อให้แต่ละสถานีงานทำงานโดยใช้เวลาในการผลิตเฉลี่ยเท่า ๆ กันภายใต้เงื่อนไขต่าง ๆ อาทิ เช่น รอบเวลาการผลิต ลำดับงาน และจำนวนเครื่องจักร ทำให้การไหลของกระบวนการมีความยืดหยุ่นและขจัดความล่าช้าหรือคอขวด (Bottleneck) ส่งผลให้สามารถผลิตสินค้าได้อย่างถูกต้องและขจัดข้อผิดพลาดระหว่างการผลิต ปัญหาการจัดสมดุลสายการประกอบ

แบบอย่างง่าย (Simple Assembly Line Balancing Problem: SALBP) หรือปัญหาการจัดสมดุลสายการประกอบแบบเส้นตรง [1] สามารถแบ่งออกเป็น 4 ประเภทคือ ปัญหา SALBP-1 หาจำนวนสถานีงานที่ต่ำที่สุด กรณีทราบรอบเวลาการผลิต (c), ปัญหา SALBP-2 หารอบเวลาการผลิตที่ต่ำที่สุด กรณีทราบค่าจำนวนสถานีงาน (m), SALBP-E หาประสิทธิภาพสูงที่สุด กรณีทราบรอบเวลาการผลิต (c) และจำนวนสถานีงาน (m) และ SALBP-F เป็นปัญหาที่สามารถหาคำตอบได้ กรณีทราบรอบเวลาการผลิตและจำนวนสถานีงาน ดังรูปที่ 1 โดยในอุตสาหกรรมหลายๆ ประเภทยานิยมใช้ SALBP-1 ในรูปแบบเส้นตรง (Straight Line) ที่มีวัตถุประสงค์เพื่อหาจำนวนสถานีงานที่ต่ำที่สุด กรณีทราบรอบเวลาการผลิต เป็นจำนวนมากเนื่องจากจัดกระบวนการผลิตได้ง่ายและมีประสิทธิภาพ



รูปที่ 1 ประเภทปัญหาการจัดสมดุลสายการประกอบ

แบบเส้นตรง

ปัญหาการจัดสมดุลสายการประกอบแบบอย่างง่าย หรือ ปัญหาการจัดสมดุลสายการประกอบแบบเส้นตรง ประเภทที่ 1 เป็นปัญหาประเภทหนึ่งของการจัดสมดุลสายการประกอบที่มีความสำคัญต่อกระบวนการผลิตที่ต้องการสถานีงานที่ต่ำที่สุด โดยมีนักวิจัยหลายๆท่านได้นำเสนอวิธีวิธีวิสตติกและเมตาวิสตติกต่างๆสำหรับแก้ไขปัญหา SALBP-1 เช่น วิธี Hybrid Particle Swarm Algorithm (HPSA) ที่ทำการพัฒนาและเปรียบเทียบผลกับวิธี Particle Swarm Algorithm (PSA)

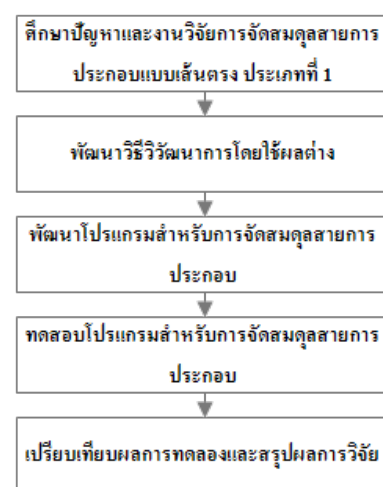
และวิธี Discrete Particle Swarm Algorithm (DPSA) พบว่าวิธี HPSA ให้ผลที่ดีกว่าวิธี PSA และวิธี DPSA [2], วิธี Ant Colony Optimization (ACO) ที่สามารถค้นหาคำตอบได้ดีเทียบเท่ากับชุดปัญหาของ Scholl [3],[4] วิธี Differential Evolution (DE) ที่ทดลองและใช้แก้ไขปัญหที่แตกต่างกัน ไม่ว่าจะเป็นการทดลองวิธี DE ในปัญหา SALBP-1 และเปรียบเทียบกับชุดปัญหา [5] การทดลองเพื่อหาสมการการปรับเปลี่ยนพิกัด (Mutation) และการแลกเปลี่ยนพิกัด (Crossover) ที่เหมาะสม [6] และการประยุกต์ใช้วิธี DE กับกรณีศึกษา [7] ซึ่งผลการทดลองพบว่าวิธี DE สามารถหาคำตอบได้ดีมีประสิทธิภาพด้วยเวลาที่รวดเร็ว, วิธี Improved Immune Algorithm ที่ปรับปรุงจากวิธีพื้นฐานและทดสอบกับชุดปัญหา ซึ่งทำให้สามารถหาคำตอบได้อย่างมีประสิทธิภาพ [8] วิธี Immune Genetic Algorithm ที่พัฒนาวิธี Immune Algorithm ร่วมกับ Genetic Algorithm และวัดประสิทธิภาพกับวิธี BB&R-BFS และ Enumeration Procedure พบว่าค้นหาคำตอบได้ดีมีประสิทธิภาพแต่ในบางปัญหายังคงใช้เวลามากในการค้นหาคำตอบ [9] วิธี Exact Methods สำหรับปัญหาสมมูลสายการประกอบแบบง่ายที่ทำการศึกษหาพารามิเตอร์ที่เหมาะสม [10] วิธี Mixed Integer Programming โดยใช้ grey theory กำหนดให้กับความต้องการและงานที่ไม่แน่นอนที่สามารถจัดงานสู่สถานีงานได้ดี [11] และวิธี COMSOAL ที่ใช้ Max-Min ของงานในการกำหนดลงสถานีงานที่สามารถลดเวลาสูญเสียได้ [12]

จากการทบทวนวรรณกรรมของการแก้ปัญหา SALBP-1 จะพบว่าการแก้ปัญหการจัดสมมูลสายการประกอบแบบอย่างง่าย ประเภทที่ 1 เพื่อหาจำนวนสถานีงานที่ต่ำที่สุดของทุกวิธีฮิวริสติกและเมตาฮิวริสติกที่กล่าวมาจะใช้ลำดับงานที่มองไปข้างหน้า (Forward) ในการจัดงานลงสถานีงานจึงส่งผลทำให้งานที่อยู่ลำดับหลังมีโอกาสน้อยกว่าสำหรับการถูกเลือกลงสถานีงาน และวิธีการวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution Algorithm: DE) เป็นวิธีเมตาฮิวริสติกที่น่าสนใจสำหรับการใช้แก้ไขปัญหานี้เนื่องจากสามารถค้นหาคำตอบได้ดี โดยใช้เวลาค้นหาคำตอบอย่างรวดเร็ว (Computation Time: CT) [1] โดยวิธีการวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution Algorithm: DE) ถูกนำเสนอ

ครั้งแรกในปี 1997 [13] และเป็นวิธีที่ถูกใช้ในงานวิจัยในปัญหาต่างๆ เช่น ปัญหาการการมอบหมายงาน (Assignment Problem) [14] ปัญหาการขนส่ง (Transportation Problem) [15] ปัญหากำหนดตำแหน่ง (Location Routing Problem) [16] เป็นต้น ดังนั้นในงานวิจัยนี้จึงได้ทำการพัฒนาวิธีการวิวัฒนาการโดยใช้ผลต่าง (Development of Differential Evolution Algorithm: DDE) ด้วยการจัดงานลงสถานีงานโดยใช้ลำดับงานที่มองย้อนกลับ (Backward) แทนการใช้ลำดับงานที่มองไปข้างหน้า (Forward) เพื่อให้งานที่อยู่ลำดับหลังสุดของการประกอบได้มีโอกาสถูกพิจารณาลงสถานีงานก่อน สำหรับปัญหา SALBP-1 ที่มีวัตถุประสงค์เพื่อหาจำนวนสถานีงานที่ต่ำที่สุด ภายใต้เงื่อนไขข้อจำกัดของรอบการผลิต และลำดับงาน จากนั้นทำการวัดประสิทธิภาพของวิธี DDE โดยการเปรียบเทียบกับวิธีฮิวริสติกและงานวิจัยอื่นๆ ที่เป็นปัญหา SALBP-1

2. วิธีการดำเนินการวิจัย

ในการวางแผนการดำเนินงานการวิจัยในการแก้ปัญหาการจัดสมมูลสายการประกอบแบบเส้นตรง ประเภทที่ 1 เพื่อหาจำนวนสถานี (Station) งานที่ต่ำที่สุด ภายใต้เงื่อนไขข้อจำกัดของรอบการผลิต (Cycle Time) และลำดับงาน (Precedence Diagram) สามารถกำหนดขั้นตอนในการดำเนินงานได้ดังรูปที่ 2



รูปที่ 2 วิธีการดำเนินการวิจัย

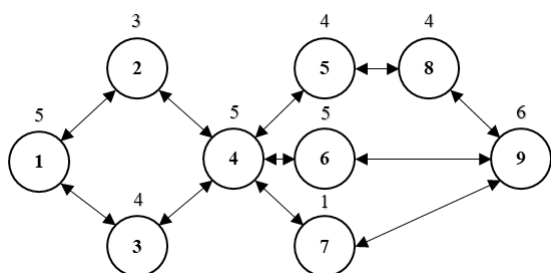
2.1 ศึกษาปัญหาและงานวิจัยการจัดสมดุลสายการประกอบแบบเส้นตรง ประเภทที่ 1

ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการจัดสมดุลสายการประกอบแบบเส้นตรง ประเภทที่ 1 กรณีทราบค่ารอบเวลาการผลิต และลำดับงาน เพื่อหาจำนวนสถานีงานที่ต่ำที่สุด โดยจะเริ่มศึกษาเอกสารงานวิจัยที่เกี่ยวข้องทั้งในประเทศและงานวิจัยต่างประเทศ เพื่อที่จะกำหนดขอบเขตวัตถุประสงค์ และเงื่อนไขที่ใช้ในการจัดสมดุลสายการประกอบของการวิจัย

2.2 พัฒนารีวิวัฒนาการโดยใช้ผลต่าง

วิธีวิวัฒนาการผลต่าง (Differential Evolution: DE) เป็นวิธีที่ดีที่สุดสำหรับแก้ปัญหาขนาดใหญ่และซับซ้อน ซึ่งวิธีนี้สามารถหาคำตอบได้อย่างรวดเร็วและมีประสิทธิภาพ [1] โดยมีกระบวนการหาคำตอบ 4 ขั้นตอนคือ การสร้างประชากรเริ่มต้น (Initial Population), การปรับเปลี่ยนพิกัด (Mutation), การแลกเปลี่ยนพิกัด (Crossover) และการคัดเลือก (Selection)

ผู้วิจัยแสดงการคำนวณโดยใช้ปัญหาของ Jaeschke จากชุดปัญหาของ Scholl [17] ที่มีจำนวน 9 งาน (Task) เวลางานแต่ละงานและเงื่อนไขของลำดับงาน (Precedence diagram) ดังรูปที่ 3 โดยลำดับงานตามทิศทางลูกศรทางขวาคืองานที่มองไปข้างหน้า (Forward) ส่วนลำดับงานตามทิศทางลูกศรทางซ้าย คืองานที่มองย้อนกลับ (Backward) สามารถแสดงดังตารางที่ 1



รูปที่ 3 ปัญหาของ Jaeschke จากชุดปัญหาของ Scholl

ตารางที่ 1 ลำดับงานของปัญหา Jaeschke

Task	Time	Target Vector	Precedence Diagram	
			Forward	Backward
1	5	0.45	-	2,3
2	3	0.65	1	4
3	4	0.12	1	4
4	5	0.78	2,3	5,6,7
5	4	0.55	4	8
6	5	0.44	4	9
7	1	0.37	4	9
8	4	0.89	5	9
9	6	0.11	6,7,8	-

2.2.1 การสร้างประชากรเริ่มต้น (Initial population)

การสร้างประชากรของคำตอบเริ่มต้นสามารถทำได้โดยการสุ่มค่าจำนวนจริงตั้งแต่ 0 ถึง 1 ให้แต่ละงาน จะได้เวกเตอร์เริ่มต้น (Target Vector) ดังตารางที่ 1 จากนั้นนำค่าเวกเตอร์เริ่มต้นมาจัดงานลงในสถานีงาน (Workstation) โดยใช้ค่าที่มากจัดลงสถานีงานก่อนค่าน้อย แต่ต้องไม่ขัดแย้งกับเงื่อนไขของลำดับงาน (Backward) และรอบเวลาการผลิต (Cycle Time: CT) เช่น การสร้างคำตอบเริ่มต้น (Vector 1) โดยกำหนดรอบเวลาการผลิตเท่ากับ 10 เงื่อนไขลำดับงานแบบย้อนกลับ (Backward) และค่าเวกเตอร์เริ่มต้นที่ได้จากการสุ่ม ตารางที่ 1 งานที่จะถูกเลือกนำมาพิจารณาลำดับแรก คืองานที่ 8 มีค่า Target Vector มากที่สุด คือ 0.89 แต่งานที่ 8 ไม่สามารถจัดลงสถานีงานได้เนื่องจากติดเงื่อนไขของงานลำดับงาน (Backward) คือจะต้องผ่านงานที่ 9 ก่อน จึงต้องพิจารณางานลำดับถัดไปที่ มีค่า Target Vector ลำดับถัดมาคืองานที่ 4 แต่งานที่ 4 ไม่สามารถจัดลงสถานีงานได้เนื่องจากติดเงื่อนไขของงานลำดับงานเช่นกัน เมื่อพิจารณางานลำดับถัดไปตามค่า Target Vector มากที่สุดลำดับถัดมา จะพบว่างานที่สามารถจัดลงสู่สถานีงานได้เป็นลำดับแรกจึงมีงานเดียว คืองานที่ 9 ที่ผ่านเงื่อนไขลำดับงานและรอบเวลาการผลิต ดังนั้นงานที่

9 จึงสามารถจัดลงสถานีงานได้ หลังจากนั้นทำการพิจารณาตามเงื่อนไขของค่า Target Vector ที่มากที่สุด ลำดับงาน และรอบเวลาการผลิต จนสามารถจัดทุกงานลงสู่สถานีงาน ดังตารางที่ 2 และเพื่อให้ได้ลำดับงานของการ

ประกอบจริง จึงต้องทำการสลับลำดับของสถานีงาน โดยทำการสลับจากสถานีงานข้างหลังมาด้านหน้า ดังตารางที่ 3 ซึ่งจะได้คำตอบที่เป็นค่าเริ่มต้นของเวกเตอร์ เพื่อไปทำการปรับเปลี่ยนพิกัด และแลกเปลี่ยนพิกัด ต่อไป

ตารางที่ 2 การจัดงานลงสถานีงานโดยใช้ลำดับงานแบบย้อนกลับ

Vector 1	Workstation	1		2			3		4	
	Task	9	8	5	6	7	4	2	3	1
	Time	6	4	4	5	1	5	3	4	5
	Target Vector	0.11	0.89	0.55	0.44	0.37	0.78	0.65	0.12	0.45

ตารางที่ 3 การสลับลำดับของสถานีงานข้างหลังมาด้านหน้า

Vector 1	Workstation	1		2		3			4	
	Task	1	3	2	4	7	6	5	8	9
	Time	5	4	3	5	1	5	4	4	6
	Target Vector	0.45	0.12	0.65	0.78	0.37	0.44	0.55	0.89	0.11

2.2.2 การปรับเปลี่ยนพิกัด (Mutation)

การปรับเปลี่ยนพิกัดเวกเตอร์ (Dimension: D) เพื่อให้ได้คำตอบใหม่ที่แปลกแตกต่างไปจากกลุ่มคำตอบเดิมแล้วจะถูกเรียกว่า มิวแทนเวกเตอร์ (Mutant Vector) โดยในงานวิจัยนี้ทำการปรับเปลี่ยนพิกัดด้วยสมการ DE/best/2 [1] ดังสมการที่ (1)

$$V_{i,G+1} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G}) \quad (2)$$

เมื่อ $V_{i,G+1}$ = มิวแทนเวกเตอร์

$X_{best,G}$ = เวกเตอร์ที่ดีที่สุดภายใต้เงื่อนไขที่กำหนด

$X_{ri,G}$ = เวกเตอร์สุ่ม (Random vector) จากประชากร G เมื่อ $i = 1-4$

F = แฟกเตอร์ในการขยายผลต่าง (Scaling Factor) คือ จำนวนจริงที่มีค่าคงที่และมีค่าระหว่าง 0-2

จากนั้นทำการปรับเปลี่ยนพิกัด (D) ด้วยสมการที่ (1) จนครบในทุกพิกัดจะได้ค่ามิวแทนเวกเตอร์ที่ 1 ดังตารางที่ 4

ตารางที่ 4 ผลการสร้างคำตอบเริ่มต้น

Vector	Dimension	1	2	3	4	5	6	7	8	9
	Task	1	3	2	4	7	6	5	8	9
Vector 1	Target Vector	0.45	0.12	0.65	0.78	0.37	0.44	0.55	0.89	0.11
Vector 2	Target Vector	0.23	0.75	0.24	0.30	0.36	0.35	0.25	0.53	0.15
Vector 3	Target Vector	0.56	0.20	0.50	0.25	0.44	0.42	0.55	0.61	0.26
Vector 4	Target Vector	0.45	0.25	0.32	0.24	0.36	0.29	0.16	0.51	0.50
Vector 5	Target Vector	0.18	0.80	0.70	0.15	0.60	0.24	0.39	0.46	0.40
	Mutant Vector 1	0.40	0.12	0.14	0.89	0.11	0.42	0.13	0.87	0.10

จากตารางที่ 4 เมื่อทำการสร้างคำตอบเริ่มต้นเพิ่ม 4 คำตอบคือ Vector 2-5 และกำหนดให้ Vector 1 คือเวกเตอร์ที่ดีที่สุด ($X_{best,G}$), ค่า $F = 0.80$ และเวกเตอร์สุ่ม (Random Vector) จำนวน 4 เวกเตอร์ (r_1, r_2, r_3, r_4) ได้เวกเตอร์ 2, 3, 4 และ 5 จากนั้นทำการคำนวณหา Mutant Vector ($V_{i,G+1}$) โดยนำเวกเตอร์สุ่มที่ได้ไปแทนสมการที่ (1) ตัวอย่างเช่น พิกัดที่ 3 $V_{i,G+1} = 0.65 + 0.8(0.24 - 0.50) + 0.8(0.32 - 0.70) = 0.14$

2.2.3 การแลกเปลี่ยนพิกัด (Crossover)

การแลกเปลี่ยนค่าในพิกัดเวกเตอร์เป็นการเพิ่มความหลากหลายในการหาคำตอบของปัญหาโดยขั้นตอนนี้จะเป็นการสร้างไทรอัลเวกเตอร์ (Trial Vector) ด้วยวิธี Exponential Crossover 1 Position [1] ดังสมการที่ (2)

$$U_{ji,G+1} = \begin{cases} V_{ji,G+1} \\ X_{ji,G} \end{cases} \quad (2)$$

เมื่อ $V_{ji,G+1}$ = มิวแทนเวกเตอร์เมื่อ $n \in \{1 \dots r\}$ และ r คือตำแหน่งแรกที่พบว่า $r_j < CR$

$X_{ji,G+1}$ = เวกเตอร์เป้าหมายในกรณีอื่น ๆ

CR = อัตราการปรับเปลี่ยนจำนวนจริงระหว่าง 0-1

r_j = การสุ่มตัวเลขจำนวนจริงที่มีค่าระหว่าง 0-1 ครั้งที่ j ; $j = 1, 2, 3, \dots, D$

จากค่ามิวแทนต์เวกเตอร์ (Mutant Vector) และเวกเตอร์เป้าหมาย (Target Vector) ของ Vector 1 ตารางที่ 4 กำหนด $CR = 0.8$ ในการแลกเปลี่ยนค่าในพิกัดเวกเตอร์จะเริ่มจากการสุ่มตัวเลขที่อยู่ระหว่าง 0-1 ให้กับทุกพิกัด (D) แล้วทำการตรวจสอบเงื่อนไขดังสมการที่ (2) พบว่าตำแหน่งแรกที่ค่า $r_j < CR$ คือ $r_5 = 0.77$ ดังนั้นค่า Trial Vector ในตำแหน่งที่ r_1-r_5 จะใช้ค่า Mutant Vector และตำแหน่งที่ r_6-r_9 จะใช้ค่า Target Vector ดังตารางที่ 5 จากนั้นนำค่าไทรอัลเวกเตอร์ที่ได้ไปจัดงานลงสถานีงานตามเงื่อนไขที่กำหนด แสดงดังตารางที่ 6

ตารางที่ 5 การไทรอัลเวกเตอร์ของเวกเตอร์เป้าหมาย 1

Vector 1	Task	1	3	2	4	7	6	5	8	9
	Target Vector	0.45	0.12	0.65	0.78	0.37	0.44	0.55	0.89	0.11
	Mutant Vector	0.40	0.12	0.14	0.89	0.11	0.42	0.13	0.87	0.10
	r_j	0.81	0.88	0.87	0.92	0.77	0.88	0.95	0.99	0.69
	Trial Vector	0.40	0.12	0.14	0.89	0.11	0.44	0.55	0.89	0.11

ตารางที่ 6 การจัดงานลงสถานีงานของเวกเตอร์เป้าหมาย 1

Vector 1	Workstation	1		2		3			4	
	Task	1	3	2	4	7	6	5	8	9
	Time	5	4	3	5	1	5	4	4	6
	Trial Vector	0.40	0.12	0.14	0.89	0.11	0.44	0.55	0.89	0.11

2.2.4 การคัดเลือก (Selection)

การคัดเลือกจะเป็นขั้นตอนของการเลือกจากเวกเตอร์ที่ให้คำตอบที่ดีที่สุดเพื่อเป็นเวกเตอร์เป้าหมายในรอบถัดไป โดยการเปรียบเทียบค่าของจำนวนสถานีงานที่ต่ำที่สุดที่ได้จากเวกเตอร์เป้าหมาย (Target Vector) กับค่าคำตอบที่ได้จากไทรอัลเวกเตอร์ (Trial Vector) ดังสมการที่ (3)

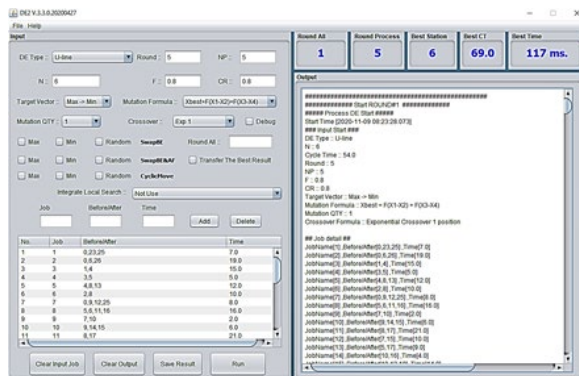
$$X_{ji,G+1} = \begin{cases} U_{ji,G+1} \\ X_{ji,G} \end{cases} \quad (3)$$

เมื่อ $U_{ji,G+1}$ = ไทรอัลเวกเตอร์ เมื่อ $(U_{ji,G+1}) < f(X_{ji,G+1})$

$X_{ji,G+1}$ = เวกเตอร์เป้าหมายในรุ่นถัดไป, $i = 1, 2, \dots, n$ ในกรณีอื่นๆ

2.3 พัฒนาโปรแกรมสำหรับการจัดสมดุลสายการประกอบ

เมื่อทำการศึกษาและพัฒนาวิธีวิวัฒนาการโดยใช้ผลต่างแล้ว ผู้วิจัยจะนำกระบวนการหาคำตอบตามการพัฒนาวิธีวิวัฒนาการโดยใช้ผลต่าง มาทำการเขียนโปรแกรมด้วยภาษา Java บน โปรแกรม NetBeans IDE 8.2 ผ่านเครื่องคอมพิวเตอร์ Intel(R) Core(TM) i5-8500 CPU@3.0 GHz, 4GB RAM ได้โปรแกรม DE2 แสดงดังรูปที่ 4



รูปที่ 4 โปรแกรม DE2

2.4 ทดสอบโปรแกรมสำหรับการจัดสมดุลสายการประกอบ

การทดสอบโปรแกรมสำหรับวิธีวิวัฒนาการโดยใช้ผลต่าง สำหรับการจัดสมดุลสายการประกอบแบบเส้นตรงประเภทที่ 1 (SALBP-1) เพื่อหาจำนวนสถานีงานที่ต่ำที่สุด (m) ผู้วิจัยได้นำชุดปัญหาของ Scholl [17] ที่มีจำนวนงานตั้งแต่ 7–70 งาน จำนวน 14 ชุดปัญหา มาใช้ทำการทดสอบ แสดงข้อมูลดังตารางที่ 7

ตารางที่ 7 แสดงข้อมูลของชุดปัญหาที่ใช้ในการทดสอบ

ชุดปัญหา	จำนวนงาน	เวลาดำเนินการต่ำสุด	เวลาดำเนินการสูงสุด	เวลารวมของทุกงาน
Mertens	7	1	6	29
Browman	8	3	17	75
Jaeschke	9	1	6	37
Jackson	11	1	7	46
Mansoor	11	2	45	185
Mitchell	21	1	13	105

ตารางที่ 7 แสดงข้อมูลของชุดปัญหาที่ใช้ในการทดสอบ(ต่อ)

ชุดปัญหา	จำนวนงาน	เวลาดำเนินการต่ำสุด	เวลาดำเนินการสูงสุด	เวลารวมของทุกงาน
Buxey	29	1	25	324
Sawyer	30	1	25	324
Lutz 1	32	100	1400	14140
Gunther	35	1	40	483
Kilbrid	45	3	55	552
Hahn	53	40	1775	14026
Warnecke	58	7	53	1548
Tonge	70	1	156	3510

2.5 การเปรียบเทียบผลการทดลองและสรุปผลการวิจัย

การเปรียบเทียบผลการทดลองของการจัดสมดุลสายการประกอบแบบเส้นตรง ประเภทที่ 1 (SALBP-1) เพื่อหาจำนวนสถานีงานที่ต่ำที่สุด (m) ในการเปรียบเทียบผลเพื่อทดสอบประสิทธิภาพของวิธีการที่พัฒนา ผู้วิจัยใช้ข้อมูลชุดปัญหาและจำนวนสถานีงานที่ต่ำที่สุด (m*) ของ Scholl [17] ดังตารางที่ 7 จำนวน 14 ชุดปัญหา มาใช้ในการหาคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่าง (DDE) เพื่อเปรียบเทียบกับวิธีวิวิธวิธี 5 วิธี ประกอบไปด้วย การจัดงานที่ใช้เวลามากที่สุด (Longest Operation Time: LOT), การจัดงานที่มีงานตามหลังมากที่สุด (Most Following Tasks: MFT), การจัดงานที่มีค่าน้ำหนักมากที่สุด (Ranked Positional Weight: RPW), การจัดงานที่ใช้เวลาน้อยที่สุด (Shortest Operation Time: SOT) และการจัดงานที่มีงานตามหลังน้อยที่สุด (Fewest Following Tasks: FFT) แล้วจากนั้นผู้วิจัยนำผลที่ได้ไปเปรียบเทียบกับงานวิจัยอื่นๆ เช่น วิธีอาณานิคมมด (Ant Colony Optimization: ACO) [3], วิธีวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution: DE) [5] และวิธีเชิงพันธุกรรมภูมิคุ้มกัน (Immune Genetic Algorithm: IGA) [9] ตามลำดับ โดยทำการเปรียบเทียบในบางชุดปัญหาของ Scholl [17] เนื่องจากในงานวิจัยแต่ละงานไม่ได้ใช้ทุกชุดปัญหาในการหาคำตอบ และเมื่อทำการทดสอบและเปรียบเทียบผลกับวิธีวิวิธวิธีเสร็จแล้ว ผู้วิจัยจะนำผลที่ได้ไปทำการสรุปผลและอภิปรายผลการวิจัยต่อไป

3. ผลการวิจัย

จากการพัฒนาวิธีวิวัฒนาการโดยใช้ผลต่าง (DDE) ในการแก้ปัญหาการจัดสมดุลสายการประกอบแบบเส้นตรงประเภทที่ 1 (SALBP-1) เพื่อหาจำนวนสถานีงานที่ต่ำที่สุด (m) โดยใช้ข้อมูลจากชุดปัญหาของ Scholl [17] มาทำการทดสอบและวัดประสิทธิภาพของโปรแกรม DE2 ผู้วิจัยจะทำการเปรียบเทียบผลกับวิธีฮิวริสติก 5 วิธี, วิธีอาณานิคมมด (ACO), วิธีวิวัฒนาการโดยใช้ผลต่าง (DE) และวิธีเชิงพันธุกรรมภูมิคุ้มกัน (IGA) สามารถแสดงผลได้ดังนี้

3.1 ผลการเปรียบเทียบกับวิธีฮิวริสติก

ผู้วิจัยนำผลที่ได้จากการหาคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่าง (DDE) โดยใช้ข้อมูลจากชุดปัญหาของ Scholl [17] จำนวน 8 ชุดปัญหา (57 ปัญหา) ไปเปรียบเทียบกับวิธีฮิวริสติก 5 วิธี ประกอบไปด้วย การจัดงานที่ใช้เวลามากที่สุด (LOT), การจัดงานที่มีงานตามหลังมากที่สุด (MFT), การจัดงานที่มีค่าน้ำหนักมากที่สุด (RPW), การจัดงานที่ใช้เวลาสั้นที่สุด (SOT) และการจัดงานที่มีงานตามหลังน้อยที่สุด (FFT) ที่หาคำตอบด้วยโปรแกรม Qm for Windows V.Student แสดงดังตารางที่ 8

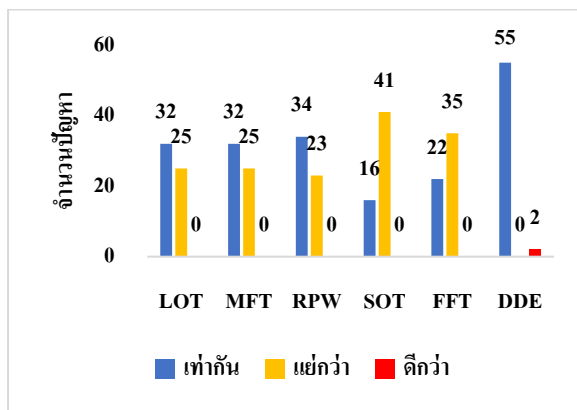
ตารางที่ 8 ผลการเปรียบเทียบกับวิธีฮิวริสติก

Problem	Task	C	m*	Heuristics					DDE	
				LOT	MFT	RPW	SOT	FFT	m	CT
Jaeschke	9	6	8	8	8	8	8	8	8	0.00
		7	7	7	7	7	7	7	7	0.00
		8	6	6	6	6	6	6	6	0.00
		10	4	4	4	4	5	4	4	0.00
		18	3	3	3	3	3	3	3	0.00
Jackson	11	7	8	8	8	8	9	8	8	0.00
		9	6	6	6	6	7	7	6	0.00
		10	5	6	6	6	6	6	5	0.00
		13	4	4	4	4	4	4	4	0.00
		14	4	4	4	4	4	4	4	0.00
		21	3	3	3	3	3	3	3	0.00
Mitchell	21	14	8	9	8	8	10	10	8	0.00
		15	8	9	8	8	10	10	8	0.00
		21	5	5	6	6	6	6	5	0.00
		26	5	5	5	5	5	5	5	0.00
		35	3	3	4	4	4	4	3	0.00
		39	3	3	3	3	3	3	3	0.00
Buxey	29	27	13	13	14	14	15	15	13	0.00
		30	12	12	12	12	14	13	12	0.00
		33	11	12	11	11	13	12	11	0.00
		36	10	10	10	10	11	10	10	0.00
		41	8	9	9	9	9	9	8	0.00
		47	7	8	8	8	8	8	7	0.00
		54	7	7	7	7	7	7	7	0.00

ตารางที่ 8 ผลการเปรียบเทียบกับวิธีฮิวริสติก (ต่อ)

Problem	Task	C	m*	Heuristics					DDE	
				LOT	MFT	RPW	SOT	FFT	m	CT
Lutz 1	32	1414	11	12	11	12	12	12	11	0.02
		1572	10	10	10	10	11	11	10	0.02
		1768	9	9	9	9	10	9	9	0.01
		2020	8	8	8	8	9	8	8	0.02
		2357	7	7	7	7	7	7	7	0.03
		2828	6	6	6	6	6	6	6	0.03
Gunther	35	41	14	15	16	16	19	18	13	0.01
		44	12	12	14	14	15	15	12	0.01
		49	11	11	12	12	12	11	11	0.00
		54	9	10	10	10	11	10	9	0.01
		61	9	9	10	9	10	10	9	0.02
		69	8	8	8	8	9	9	8	0.01
		81	7	7	7	7	8	7	6	0.01
Hahn	53	2004	8	9	8	8	8	9	8	0.03
		2338	7	7	7	7	8	8	7	0.03
		2806	6	6	6	6	6	6	6	0.04
		3507	5	5	5	5	5	5	5	0.04
		4676	4	4	4	4	4	4	4	0.03
Warnecke	53	54	31	35	33	33	36	36	31	0.01
		56	29	32	33	33	35	35	29	0.03
		58	29	30	31	30	35	32	29	0.03
		60	27	29	30	29	35	32	27	0.02
		62	27	29	30	29	35	31	27	0.03
		65	25	29	29	27	33	31	25	0.03
		68	24	27	26	26	31	28	24	0.04
Warnecke	53	71	23	26	24	24	30	26	23	0.03
		74	22	25	23	23	28	25	22	0.03
		78	21	22	22	21	25	24	21	0.02
		82	20	21	21	20	24	23	20	0.03
		86	19	20	20	20	23	22	19	0.03
		92	17	19	19	19	22	20	17	0.04
		97	17	18	17	17	21	18	17	0.03
		104	15	16	16	16	18	16	15	0.05

จากผลการหาจำนวนสถานงานที่ต่ำที่สุด (m) เมื่อเปรียบเทียบกับวิธีฮิวริสติก 5 วิธี ดังตารางที่ 8 พบว่าเมื่อเปรียบเทียบกับจำนวนสถานงานที่ต่ำที่สุด (m*) กับชุดปัญหาของ Scholl [17] วิธีวิวัฒนาการโดยใช้ผลต่าง (DDE) สามารถหาคำตอบได้ดีกว่าวิธีฮิวริสติกทั้ง 5 วิธี โดยสามารถหาคำตอบได้เทียบเท่า 55 ปัญหา และได้ดีกว่า 2 ปัญหา จากทั้งหมด 57 ปัญหา โดยใช้เวลาในการค้นหาคำตอบ (CT) เฉลี่ยทุกปัญหา 0.01 วินาที แสดงดังรูปที่ 5



รูปที่ 5 การเปรียบเทียบผลของวิธีฮิวริสติกกับวิธี DDE

3.2 ผลการเปรียบเทียบวิธีอาณานิคมมด

การเปรียบเทียบวิธีอาณานิคมมด (ACO) [3] ผู้วิจัยจะใช้ข้อมูลจากชุดปัญหาของ Scholl [17] จำนวน 5 ชุดปัญหา (43 ปัญหา) ไปทำการหาคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่าง (DDE) และเปรียบเทียบผลเฉพาะจำนวนสถานงานที่ต่ำที่สุด (m) เนื่องจากในวิธี ACO [3] ไม่แสดงผลของเวลาในการค้นหาคำตอบ (CT) แสดงดังตารางที่ 9

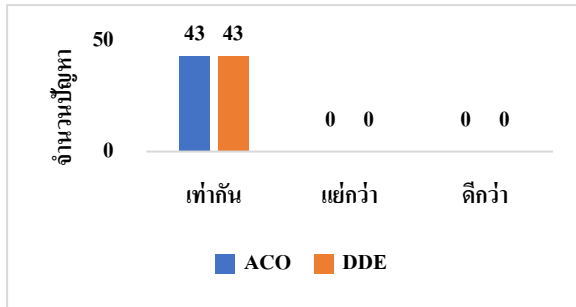
ตารางที่ 9 ผลการเปรียบเทียบกับวิธี ACO

Problem	Task	C	m*	ACO	DDE	
				m	m	CT
Mertens	7	6	6	6	6	0.00
		7	5	5	5	0.00
		8	5	5	5	0.00
		10	3	3	3	0.00
		15	2	2	2	0.00
		18	2	2	2	0.00

ตารางที่ 9 ผลการเปรียบเทียบกับวิธี ACO (ต่อ)

Problem	Task	C	m*	ACO	DDE	
				m	m	CT
Jackson	11	7	8	8	8	0.00
		9	6	6	6	0.00
		10	5	5	5	0.00
		13	4	4	4	0.00
		14	4	4	4	0.00
		21	3	3	3	0.00
Mitchell	21	14	8	8	8	0.00
		15	8	8	8	0.00
		21	5	5	5	0.00
		26	5	5	5	0.00
		35	3	3	3	0.00
		39	3	3	3	0.00
Sawyer	30	25	14	14	14	0.00
		27	13	13	13	0.00
		30	12	12	12	0.00
		33	11	11	11	0.00
		36	10	10	10	0.00
		41	8	8	8	0.02
		47	7	7	7	0.00
		54	7	7	7	0.01
		75	5	5	5	0.01
Ttong	70	160	23	23	23	0.07
		168	22	22	22	0.06
		176	21	21	21	0.13
		185	20	20	20	0.07
		195	19	19	19	0.08
		207	18	18	18	0.05
		220	17	17	17	0.05
		234	16	16	16	0.04
		251	14	14	14	0.09
		270	14	14	14	0.07
		293	13	13	13	0.06
		320	11	11	11	0.08
		364	10	10	10	0.07
		410	9	9	9	0.06
		468	8	8	8	0.06
		527	7	7	7	0.08

จากผลการหาจำนวนสถานีงานที่ต่ำที่สุด (m) ดังตารางที่ 9 เมื่อเปรียบเทียบกับวิธีอาณานิคมมด (ACO) พบว่าวิธี DDE และวิธี ACO สามารถหาคำตอบได้ดีเท่ากันทั้งหมด 43 ปัญหา ซึ่งเท่ากับกับคำตอบของจำนวนสถานีงานที่ต่ำที่สุด (m^*) ของชุดปัญหา Scholl [17] โดยวิธี DDE ใช้เวลาในการค้นหาคำตอบ (CT) เฉลี่ยทุกปัญหา 0.03 วินาที แสดงดังรูปที่ 6



รูปที่ 6 การเปรียบเทียบผลของวิธี ACO กับวิธี DDE

3.3 ผลการเปรียบเทียบวิธีวิวัฒนาการโดยใช้ผลต่าง

การเปรียบเทียบผลของจำนวนสถานีงานที่ต่ำที่สุด (m) และเวลาในการค้นหาคำตอบ (Computation time: CT) ผู้วิจัยใช้ข้อมูลจากชุดปัญหาของ Scholl [17] จำนวน 9 ชุด ปัญหา (45 ปัญหา) ในการหาคำตอบแล้วเปรียบเทียบผลของวิธีวิวัฒนาการโดยใช้ผลต่าง (DE) [5] ที่ใช้ลำดับงานไปข้างหน้า (Forward) กับวิธีวิวัฒนาการโดยใช้ผลต่าง (DDE) ใช้ลำดับงานแบบย้อนกลับ (Backward) แสดงดังตารางที่ 10

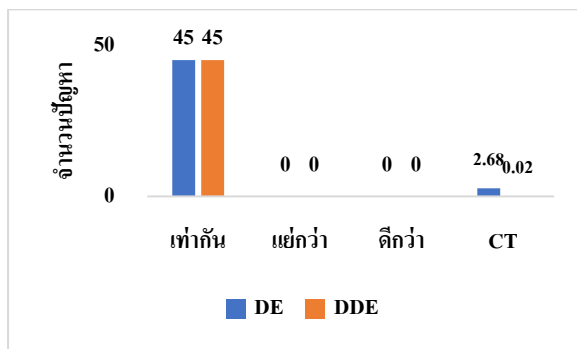
ตารางที่ 10 ผลการเปรียบเทียบกับวิธี DE

Problem	Task	C	m^*	DE		DDE	
				m	CT	m	CT
Mertens	7	6	6	6	0.00	6	0.00
		7	5	5	0.00	5	0.00
		8	5	5	0.00	5	0.00
		10	3	3	0.00	3	0.00
		15	2	2	0.19	2	0.00
		18	2	2	0.00	2	0.00
Browman	8	20	5	5	0.00	5	0.00

ตารางที่ 10 ผลการเปรียบเทียบกับวิธี DE (ต่อ)

Problem	Task	C	m^*	DE		DDE	
				m	CT	m	CT
Jaeschke	9	6	8	8	0.22	8	0.00
		7	7	7	0.00	7	0.00
		8	6	6	0.00	6	0.00
		10	4	4	0.00	4	0.00
		18	3	3	0.00	3	0.00
Jackson	11	7	8	8	0.00	8	0.00
		9	6	6	0.00	6	0.00
		10	5	5	0.16	5	0.00
		13	4	4	0.00	4	0.00
		14	4	4	0.00	4	0.00
		21	3	3	0.00	3	0.00
Mansoor	11	48	4	4	0.00	4	0.00
		62	3	3	0.15	3	0.00
		94	2	2	0.00	2	0.00
Mitchell	21	14	8	8	1.07	8	0.00
		15	8	8	0.00	8	0.00
		21	5	5	0.31	5	0.00
		26	5	5	0.00	5	0.00
		35	3	3	0.16	3	0.00
		39	3	3	0.00	3	0.00
Sawyer	30	25	14	14	0.16	14	0.00
		27	13	13	0.31	13	0.00
		30	12	12	0.00	12	0.00
		36	10	10	0.00	10	0.00
		41	8	8	5.30	8	0.02
		54	7	7	0.00	7	0.01
		75	5	5	0.00	5	0.00
Kilbrid	45	57	10	10	0.87	10	0.07
		79	7	7	44.12	7	0.12
		92	6	6	29.03	6	0.09
		110	6	6	0.00	6	0.01
		138	4	4	3.91	4	0.10
Tong	70	184	3	3	1.25	3	0.02
		176	21	21	32.53	21	0.13
		364	10	10	0.78	10	0.07
		410	9	9	0.00	9	0.06
		468	8	8	0.00	8	0.06
		527	7	7	0.00	7	0.08

จากตารางที่ 10 แสดงผลการหาจำนวนสถานีงานที่ต่ำที่สุด (m) พบว่า วิธีวิวัฒนาการโดยใช้ผลต่าง (DDE) ใช้ลำดับงานแบบย้อนกลับ (Backward) และวิธีวิวัฒนาการโดยใช้ผลต่าง (DE) ที่ใช้ลำดับงานไปข้างหน้า (Forward) สามารถหาคำตอบได้ดีเท่ากันทั้งหมด 45 ปัญหา ซึ่งเท่ากับคำตอบของจำนวนสถานีงานที่ต่ำที่สุด (m^*) ของชุดปัญหา Scholl [17] เมื่อวิเคราะห์เวลาในการค้นหาคำตอบ (CT) พบว่าวิธี DDE ใช้เวลาในการหาคำตอบโดยเฉลี่ย 0.02 วินาที ซึ่งน้อยกว่าวิธี DE ที่ใช้เวลาโดยเฉลี่ย 2.68 วินาที แสดงดังรูปที่ 7



รูปที่ 7 การเปรียบเทียบผลของวิธี DE กับวิธี DDE

3.4 ผลการเปรียบเทียบวิธีเชิงพันธุกรรมภูมิคุ้มกัน

การเปรียบเทียบผลของจำนวนสถานีงานที่ต่ำที่สุด (m) และเวลาในการค้นหาคำตอบ (Computation time: CT) ของวิธีเชิงพันธุกรรมภูมิคุ้มกัน (IGA) [9] กับวิธีวิวัฒนาการโดยใช้ผลต่าง (DDE) ผู้วิจัยใช้ข้อมูลจากชุดปัญหาของ Scholl [17] จำนวน 5 ชุดปัญหา (18 ปัญหา) ในการทดสอบประสิทธิภาพ แสดงดังตารางที่ 11

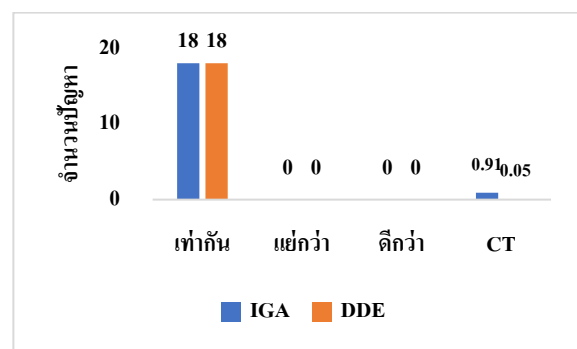
ตารางที่ 11 ผลการเปรียบเทียบกับวิธี IGA

Problem	Task	C	m^*	IGA		DDE	
				m	CT	m	CT
Mertens	7	7	5	5	0.80	5	0.00
		10	3	3	0.78	3	0.00
		18	2	2	0.73	2	0.00
Mitchell	21	14	8	8	0.81	8	0.00
		26	5	5	0.80	5	0.00
		39	3	3	0.78	3	0.00

ตารางที่ 11 ผลการเปรียบเทียบกับวิธี IGA (ต่อ)

Problem	Task	C	m^*	IGA		DDE	
				m	CT	m	CT
Sawyer	30	25	14	14	0.87	14	0.00
		30	12	12	0.86	12	0.00
		41	8	8	0.84	8	0.02
Kilbrid	45	79	7	7	0.89	7	0.12
		92	6	6	0.87	6	0.09
		138	4	4	0.86	4	0.10
Tonge	70	176	21	21	1.21	21	0.13
		251	14	14	1.12	14	0.09
		320	11	11	1.07	11	0.08
		364	10	10	1.06	10	0.07
		468	8	8	1.03	8	0.06
		527	7	7	0.98	7	0.08

การเปรียบเทียบผลของการหาจำนวนสถานีงานที่ต่ำที่สุด (m) ของวิธีเชิงพันธุกรรมภูมิคุ้มกัน (IGA) กับวิธีวิวัฒนาการโดยใช้ผลต่าง (DDE) ดังตารางที่ 11 พบว่า วิธี DDE และวิธี IGA สามารถหาคำตอบได้ดีเท่ากันทั้งหมด 18 ปัญหา และยังเท่ากับคำตอบของจำนวนสถานีงานที่ต่ำที่สุด (m^*) ในชุดปัญหา Scholl [17] เมื่อวิเคราะห์เวลาในการค้นหาคำตอบ (CT) พบว่าวิธี DDE ใช้เวลาในการหาคำตอบโดยเฉลี่ย 0.05 วินาที ซึ่งน้อยกว่าวิธี IGA ที่ใช้เวลาโดยเฉลี่ย 0.91 วินาที แสดงดังรูปที่ 8



รูปที่ 8 การเปรียบเทียบผลของวิธี IGA กับวิธี DDE

จากการพัฒนาและทดสอบวิธี DDE ด้วยโปรแกรม DE2 ในการหาจำนวนสถานีงานที่ต่ำที่สุด (m) โดยใช้ข้อมูลของชุด

ปัญหา Scholl [17] และวัดประสิทธิภาพโดยการเปรียบเทียบกับวิธีฮิวริสติก 5 วิธี (LOT, MFT, RPW, SOT, FFT), วิธี ACO, วิธี DE และวิธี IGA พบว่า วิธี DDE สามารถค้นหาคำตอบได้อย่างมีประสิทธิภาพด้วยเวลาอันรวดเร็ว ซึ่งจะเห็นได้เมื่อเปรียบเทียบกับวิธี RPW ที่เป็นวิธีที่ดีที่สุดจาก 5 วิธีฮิวริสติกที่สามารถหาคำตอบได้เทียบเท่า 34 ปัญหา และแย่กว่า 23 ปัญหา จากทั้งหมด 57 ปัญหา ซึ่งต่างจากวิธี DDE สามารถหาคำตอบได้เทียบเท่า 55 ปัญหา และดีกว่า 2 ปัญหา เมื่อเปรียบเทียบกับวิธี ACO, วิธี DE และวิธี IGA พบว่าวิธี DDE สามารถค้นหาคำตอบได้ดีและเทียบเท่าในทุกปัญหาแต่วิธี DDE ใช้เวลาในการค้นหาคำตอบที่รวดเร็วกว่า จึงสามารถสรุปได้ว่าวิธี DDE ที่ใช้ลำดับงานแบบย้อนกลับ (Backward) เพื่อให้งานที่อยู่ลำดับท้ายมีโอกาสได้พิจารณาลงสถานีงานก่อน เป็นวิธีค้นหาคำตอบที่ดีมีประสิทธิภาพ ภายในระยะเวลาอันรวดเร็ว

4. สรุปผลการวิจัยและข้อเสนอแนะ

4.1 สรุปผลการวิจัย

การพัฒนาวิธีวิวัฒนาการโดยใช้ผลต่าง (DDE) สำหรับปัญหาการจัดสมดุลสายการประกอบแบบอย่างง่าย ประเภทที่ 1 เพื่อหาจำนวนสถานีงานที่ต่ำที่สุด เป็นวิธีที่พัฒนามาจากวิธีวิวัฒนาการโดยใช้ผลต่าง (DE) แต่วิธี DDE จะใช้ลำดับงานแบบย้อนกลับ (Backward) ในการพิจารณางานลงสู่สถานีงานก่อน โดยกำหนดการปรับเปลี่ยนพิกัดด้วย $DE/best/2$ การแลกเปลี่ยนพิกัดแบบ Exponential Crossover 1 Position $F = 0.8$ และ $CR = 0.8$ ตามงานวิจัยของ Chantarasamai และ Lasunon [1] จากนั้นทำการวัดประสิทธิภาพด้วยการค้นหาคำตอบผ่านโปรแกรม DE2 โดยใช้ชุดปัญหาของ Scholl [17] 14 ชุดปัญหาที่มีจำนวนงานตั้งแต่ 7–70 งาน แล้วนำผลที่ได้ไปเปรียบเทียบกับวิธีที่ใช้ลำดับงานที่มองไปข้างหน้า (Forward) ได้แก่วิธีฮิวริสติก 5 วิธี คือการจัดงานที่ใช้เวลามากที่สุด (LOT), การจัดงานที่มีงานตามหลังมากที่สุด (MFT), การจัดงานที่มีค่าน้ำหนักมากที่สุด (RPW), การจัดงานที่ใช้เวลาสั้นที่สุด (SOT) และการจัดงานที่มีงานตามหลังน้อยที่สุด (FFT) และกับงานวิจัยอื่นๆ ได้แก่วิธีอาณานิคมมด (ACO) [3], วิธีวิวัฒนาการโดยใช้ผลต่าง (DE) [5] และวิธีเชิงพันธุกรรม

ภูมิคุ้มกัน (IGA) [9] ตามลำดับ ซึ่งพบว่าวิธี DDE สามารถค้นหาคำตอบได้ดีกว่าวิธีฮิวริสติกทั้ง 5 วิธี และสามารถค้นหาคำตอบได้ดีและเทียบเท่าในทุกปัญหาของวิธี ACO, วิธี DE และวิธี IGA วิธี DDE แต่วิธี DDE ใช้เวลาในการค้นหาคำตอบที่รวดเร็วกว่า

4.2 ข้อเสนอแนะ

จากการพัฒนาวิธี DDE สำหรับปัญหา SALBP-1 และวัดประสิทธิภาพของวิธีการกับวิธีฮิวริสติกและเมตาฮิวริสติกดังที่กล่าวมา ซึ่งก็ทำให้ได้ผลของคำตอบที่ดีและมีประสิทธิภาพ แต่ในการทดสอบในงานวิจัยนี้ใช้ขนาดปัญหาในการทดสอบที่ 7–70 งาน ดังนั้นในอนาคตควรทดสอบกับปัญหาขนาดใหญ่และมีจำนวนงานที่มากขึ้น ตลอดจนอาจนำวิธีค้นหาคำตอบแบบเฉพาะที่ (Local Search) มาช่วยในการพัฒนาคำตอบ เพื่อที่อาจจะสามารถหาคำตอบได้ดียิ่งขึ้น

5. กิตติกรรมประกาศ

ขอขอบพระคุณ สาขาวิชาวิศวกรรมเครื่องกล คณะเกษตรศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี วิทยาเขตสุรินทร์ ที่สนับสนุนอุปกรณ์ในการทำงานวิจัยครั้งนี้

ขอขอบพระคุณ สาขาวิชาวิศวกรรมการจัดการ อุตสาหกรรม คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏบุรีรัมย์ ที่สนับสนุนอุปกรณ์ในการทำงานวิจัยครั้งนี้

เอกสารอ้างอิง

- [1] K. Chantarasamai and O. U. Lasunon, "Modified Differential Evolution Algorithm for U-Shaped Assembly Line Balancing Type 2," *International Journal of Intelligent Engineering & Systems*, vol. 14, no. 4, pp. 452–462, 2021, doi: 10.22266/ijies2021.0831.39.
- [2] J. Dou, J. Li and Q. Lv, "A hybrid particle swarm algorithm for assembly line balancing problem of type 1," in *2011 IEEE International Conference on Mechatronics and Automation*, Beijing, China, 2011, pp. 1664–1669, doi: 10.1109/ICMA.2011.5986373.

- [3] M. N. I. Sulaiman, Y. H. Choo and K. E. Chong, "Ant colony optimization with look forward ant in solving assembly line balancing problem," in *2011 3rd Conference on Data Mining and Optimization (DMO)*, Putrajaya, Malaysia, 2011, pp. 115–121, doi: 10.1109/DMO.2011.5976514.
- [4] Y. guang Zhong and B. Ai, "A modified ant colony optimization algorithm for multi-objective assembly line balancing," *Soft Computing*, vol. 21, no. 22, pp. 6881–6894, 2017, doi: 10.1007/s00500-016-2240-9.
- [5] A. C. Nearchou and S. L. Omirou, "Assembly Line Balancing Using Differential Evolution Models," *Cybernetics and Systems*, vol. 48, no. 5, pp. 436–458, 2017, doi: 10.1080/01969722.2017.1319238.
- [6] P. Parawech and R. Pitakaso, "Solving an Assembly Line Balancing Problem by Differential Evolution," *Journal of Industrial Technology Ubon Ratchathani Rajabhat University*, vol. 3, no. 6, pp. 13–20, 2013.
- [7] P. Parawech, R. Paitakaso and P. Mayachearw, "Solving an Assembly Line Balancing Problem by Differential Evolution: A Case Study of a Garment Factory," *Princess Naradhiwas University Journal*, vol. 6, no. 2, pp. 92–104, 2014.
- [8] H. Y. Zhang, "An improved immune algorithm for simple assembly line balancing problem of type 1," *Journal of Algorithms & Computational Technology*, vol. 11, no. 4, pp. 317–326, 2017, doi: 10.1177/1748301817710924.
- [9] H. -y Zhang, "An immune genetic algorithm for simple assembly line balancing problem of type 1," *Assembly Automation*, vol. 39, no. 1, pp. 113–123, 2019, doi: 10.1108/AA-08-2017-101.
- [10] Z. Li, I. Kucukkoc and Q. Tang, "A comparative study of exact methods for the simple assembly line balancing problem," *Soft Computing*, vol. 24, no. 15, pp. 11459–11475, 2020, doi: 10.1007/s00500-019-04609-9.
- [11] O. A. Arik, E. Köse and J. Forrest, "Simple assembly line balancing problem of Type 1 with grey demand and grey task durations," *Grey Systems: Theory and Application*, vol. 9, no. 4, pp. 401–414, 2019, doi: 10.1108/GS-05-2019-0011.
- [12] P. SreSraco and K. Chantarasamai, "COMSOAL METHOD FOR ASSEMBLY LINE BALANCING To REDUCE IDLE TIME," *Kasem Bundit Engineering Journal*, vol. 9, no. 2, pp. 90–104, 2019.
- [13] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.
- [14] S. Kaewman, T. Srivarapongse, C. Theeraviriya and G. Jirasirilerd, "Differential Evolution Algorithm for Multilevel Assignment Problem: A Case Study in Chicken Transportation," *Mathematical and Computational Applications*, vol. 23, no. 4, 2018, Art. no. 55, doi: 10.3390/mca23040055.
- [15] R. Kamphukaew, K. Sethanan, T. Jamrus and H. K. Wang, "Differential evolution algorithms with local search for the multi-products capacitated vehicle routing problem with time windows: A case study of the ice industry," *Engineering and Applied Science Research*, vol. 45, no. 4, pp. 273–281, 2018, doi: 10.14456/easr.2018.37.
- [16] R. Akararungruangkul and S. Kaewman, "Modified Differential Evolution Algorithm Solving the Special Case of Location Routing Problem," *Mathematical and Computational Applications*, vol. 23, no. 3, 2018, Art. no. 34, doi: 10.3390/mca23030034.
- [17] *Benchmark Data Sets by Scholl (1993)*, Data Line Balancing, 2022. [Online]. Available: <https://assembly-line-balancing.de/salbp/benchmark-data-sets-1993>.