

**การปรับปรุงประสิทธิภาพการหาคำตอบที่เหมาะสมโดยวิธีการแมลงวันผลไม้
สำหรับการเดินทางของพนักงานขาย**

**Improve Performance the Discrete Fruit Fly Optimization Algorithm (FOA) for
symmetric travelling salesman problem (TSP)**

สุภาพร คำเตจา, วรัญญา สองสีใส, ศศิชา สุขกาย, นคร ไชยวงศ์ศักดิ์,
วิลาสินี ศรีสุวรรณ และ เสกสรรค์ วินยางค์กุล*

Supaporn Kamtaeja, Waranya Songseesai, Sasicha Sukkay, Nakorn Chaiwongsakda,
Wilasinee Srisuwan and Seksan Winyangkul*

¹ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏเชียงราย 80 หมู่ 9 ต.บ้านดู่ อ.เมือง จ.เชียงราย ประเทศไทย 57100

¹ Faculty of Industrial Technology, Chiang Rai Rajabhat University 80 Moo9, Ban Du, Mueang Chiang Rai,
Chiang Rai, Thailand 57100

*Corresponding author E-mail: seksan.win@crju.ac.th

บทคัดย่อ

การศึกษาการเพิ่มประสิทธิภาพในการหาคำตอบสำหรับการจัดเส้นทางปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem : TSP) ด้วยวิธีการหาค่าที่เหมาะสมของแมลงวันผลไม้ (Fruit Fly Optimization Algorithm: FOA) การแก้ปัญหการจัดเส้นทางด้วยเมตาฮิวริสติก (Metaheuristic) ปรับปรุงการหาผลเฉลยเดิมของวิธีการแมลงวันผลไม้โดยการปรับค่าพารามิเตอร์ (parameter) เพิ่มมิติ (dimension) ให้เป็น 3 มิติ (3D) และมิติที่สูงขึ้น (Hyper Dimensional) และการปรับเปลี่ยนแบบ Swap, Swop และ Insertion position แต่ละการปรับเปลี่ยนจะพิจารณาการสลับในสองตำแหน่ง (2-Opt) สามตำแหน่ง (3-Opt) และการสลับสองและสามตำแหน่ง (2-3 Opt) และการกำหนดค่าเริ่มต้นโดยใช้วิธีการ Saving โดยได้แบ่งผลการวิเคราะห์ปัญหาออกเป็น 3 ส่วน ดังนี้ 1. การพิจารณาหาตัวแปรที่เหมาะสมในการตั้งค่าพารามิเตอร์ 2. การพิจารณาที่เกิดจากการเรียนรู้ด้วยตัวเอง แบบวัตถุประสงค์เดียวโดยมีปัญหเป็นแบบ TSP 3. ผลการพิจารณาข้อมูล TSP จำนวน 9 ชุดข้อมูล โดยมีตัวแบบปัญหาเป็นแบบวัตถุประสงค์เดียว และยังมีการเปรียบเทียบกับวิธีการอื่นๆ 10 วิธีการ

จากการวิจัยพบว่าแบบมีโดเมนชั้น 3 มิติ โดยมีการปรับเปลี่ยนแบบการสลับแบบ 2-3 opt และมีการใช้ saving ในการกำหนดค่าประชากรเริ่มต้น เป็นอัลกอริทึมที่ดีที่สุด เกิดจากการปรับค่าพารามิเตอร์ให้มีโดเมนชั้นจากเดิมสองมิติ (2D) ให้เป็นสามมิติ (3D) ในพิกัด (X,Y,Z) จะเพิ่มประสิทธิภาพของ FOA โดยการเพิ่มโดเมนชั้น และมิติที่สูงขึ้น จากผลการทดสอบตัวแบบมาตรฐาน 9 ชุดข้อมูล ของปัญหา TSP โดยอัลกอริทึมที่ได้อันดับที่ 1 มีอยู่ด้วยกัน 2 อัลกอริทึม ได้แก่ 1) FOA3twothree17S ให้ผลลัพธ์ที่ดีที่สุดของค่าที่ได้คำตอบใกล้เคียงค่า optimal solution จำนวน 4 ชุดข้อมูล ดังนี้ Dantzig42, att48, st70 และ Eil76 โดยมีเปอร์เซ็นต์คลาดเคลื่อนจากผลลัพธ์ที่ดีที่สุดเป็น -2.780, 1.655, 3.659 และ 4.617 ตามลำดับ และ 2) FOA2twothree14S จำนวน 1 ชุดข้อมูล ได้แก่ berlin52 โดยมีเปอร์เซ็นต์คลาดเคลื่อนจากผลลัพธ์ที่ดีที่สุดเป็น 3.437 ซึ่งเป็นไปตามวัตถุประสงค์ เนื่องด้วยความสามารถในการเปลี่ยนแปลงค่าพารามิเตอร์มีตัวแปรที่สูงกว่าทำให้สามารถปรับตำแหน่งเพื่อเข้าหาคำคำตอบของการแก้ไขปัญหา TSP ได้ดีกว่าแบบเดิม และทำการปรับเปลี่ยนแบบ 2-Opt, 3-Opt และ 2-3 Opt โดยการนำ Saving เข้ามาช่วยในกระบวนการหาคำตอบนั้น ทำให้คำตอบเข้าใกล้ค่า Optimal solution หรือมีคำตอบใกล้เคียงกับค่าคำตอบที่ดีที่สุด และการนำ Saving เข้ามาช่วยโดยการกำหนดค่าเริ่มต้นจะสามารถช่วยลดระยะเวลาในการหาคำตอบลงได้

คำสำคัญ : การปรับปรุงประสิทธิภาพ, เมตาฮิวริสติก, วิธีการหาค่าที่เหมาะสมของแมลงวันผลไม้, ปัญหาการเดินทางของพนักงานขาย

ได้รับเมื่อ 6 พฤษภาคม 2568; แก้ไขเมื่อ 27 สิงหาคม 2568; ตอรับการตีพิมพ์เมื่อ 31 สิงหาคม 2568

Abstract

The study of optimizing solutions for the Traveling Salesman Problem (TSP) using the Fruit Fly Optimization Algorithm (FOA) involves solving routing problems with a metaheuristic approach. The research focuses on improving the original FOA solution by adjusting parameters and increasing the dimensions to three-dimensional (3D) and higher-dimensional spaces (Hyper Dimensional). Additionally, it includes modifications such as Swap, Swop, and Insertion positions. Each modification considers 2-Opt (two-position), 3-Opt (three-position), and a combination of 2-3 Opt (two and three-position). The initial configuration is set using the Saving method. The analysis of the problem is divided into three parts. 1) Identifying the optimal variables for parameter setting. 2) Self-learning optimization using a single-objective approach for solving the TSP and 3) Analysis to 9 TSP datasets using a single-objective approach while comparing results with 10 other algorithms.

The research findings indicate that the three-dimensional incorporating a modified 2-3 opt switching scheme and utilizing the savings method for initial population configuration, represents the optimal algorithm. This enhancement was achieved by extending the parameter dimensions from the original two-dimensional (2D) space to a 3D space within the (X, Y, Z). The expansion into Hyper Dimensional improves the performance of the FOA by providing greater parametric flexibility. Based on tests 9 standard TSP datasets, 2 algorithms ranked first(1th). These are 1) FOA3twothree17S, which produced the best results in terms of solutions closest to the optimal solution for 4 datasets, namely Dantzig42, att48, st70, and Eil76, with deviation percentages of -2.780, 1.655, 3.659, and 4.617, respectively. and 2) FOA2twothree14S, which achieved the best result for 1 dataset, berlin52, with a deviation percentage of 3.437. in line with the research objectives. The increased number of parameters allows the algorithm to adjust positions more effectively to approach optimal solutions compared to the FOA original. Furthermore, the integration of 2-Opt, 3-Opt, and 2-3 Opt exchanges, combined with the Savings method for initializing the population, helps guide the search toward solutions near the optimal, while also reducing computational time required to obtain high-quality solutions.

Keywords: Improve Performance, Metaheuristic, Fruit Fly Optimization Algorithm (FOA), Travelling Salesman Problem (TSP)

Received: July 6, 2025; Revised: August 27, 2025; Accepted: August 31, 2025

1. บทนำ

สภาวะการแข่งขันในโลกมีการเปลี่ยนแปลงอย่างรวดเร็วทั้งในภาคธุรกิจและภาคอุตสาหกรรมโดยปัญหาการขนส่งเป็นปัญหาหนึ่งของปัญหาด้านโลจิสติกส์ที่ได้รับความสนใจ เนื่องจากธุรกิจแทบทุกประเภทจำเป็นต้องมีการขนส่งอย่างหลีกเลี่ยงไม่ได้ เมื่อธุรกิจเกิดการขยายตัวปัญหาการจัดเส้นทางเดินรถก็就会有ความซับซ้อนมากขึ้นจากตัวแปรที่เกิดขึ้น ยิ่งธุรกิจที่มีขนาดที่ใหญ่ก็จะมีระบบงานที่มีตัวแปรที่จะต้องเลือกในการตัดสินใจที่มากขึ้น ปัญหาทางโลจิสติกส์ในการขนส่งมีหลายปัจจัยที่มีผลต่อประสิทธิภาพ ความน่าเชื่อถือและความคุ้มค่าในการย้ายสินค้าจากที่หนึ่งไปยังที่หนึ่ง การบริหารจัดการการขนส่งให้มีประสิทธิภาพจึงเป็นสิ่งสำคัญสำหรับการลดต้นทุนและเพื่อสร้างความพึงพอใจสูงสุดให้แก่ลูกค้า จึงต้องใช้เครื่องมือในการวางแผนเส้นทางและเครื่องมือในการวางแผนที่มีคุณภาพเพื่อหาเส้นทางที่มีประสิทธิภาพที่สุด เพื่อลดการใช้ระยะทาง ลดเวลาการขนส่งและลดต้นทุนการขนส่งโดยรวมทั้งหมด ปัญหาการเดินทางของพนักงานขาย (Travelling Salesman Problem

หรือ TSP) เป็นปัญหาที่พนักงานขายต้องเดินทางไปยัง N เมือง โดยต้องเริ่มจากเมืองหนึ่ง เดินทางไปยังเมืองต่าง ๆ ให้ครบทุกเมืองเพียงครั้งเดียว และกลับมายังเมืองที่เริ่มต้น โดยมีเป้าหมายคือการหาระยะทางรวมสั้นที่สุด (หรือใช้ทรัพยากรอื่น ๆ น้อยที่สุด เช่น เวลา หรือค่าใช้จ่ายในการเดินทาง) ในช่วง 20 ปีที่ผ่านมาปัญหา TSP ได้ทำหน้าที่เป็นเครื่องมือในการทดสอบ และทดลองขั้นตอนสำหรับอัลกอริทึมใหม่ๆ โดย เส้นทาง และเวลาการทำงานของเส้นทางที่แน่นอนใช้เวลานานในการคำนวณมาก ไม่สามารถใช้ได้อย่างมีประสิทธิภาพในการแก้ปัญหาที่มีขนาดใหญ่ การพัฒนาวิธีการเมต้าฮิวริสติก (Metaheuristic) ที่ได้รับการพัฒนา และปรับปรุงเพื่อแก้ปัญหา TSP มีการนำมาใช้อย่างกว้างขวางแต่ประสิทธิภาพยังไม่ค่อยดีเทียบกับวิธีการค้นหาในพื้นที่ใกล้เคียง เช่น 2-OPT, 3-OPT และวิธี Lin-Kernighan heuristics (LK) ซึ่งจะเห็นได้ว่าเมื่อนำวิธีเมต้าฮิวริสติก มารวมกับวิธีการฮิวริสติก (Heuristic) นั้นจะสามารถเพิ่มประสิทธิภาพในการหาผลเฉลยของปัญหาทางวิศวกรรมได้เป็นอย่างดี โดยการผสมผสานระหว่างเมต้าฮิวริสติกและฮิวริสติกเชิงปฏิบัติการ โดยมีเป้าหมายเพื่อใช้จุดแข็งของแต่ละวิธีร่วมกัน กล่าวคือ เมต้าฮิวริสติกมีความสามารถในการสำรวจเชิงกว้าง (exploration) เพื่อค้นหาพื้นที่คำตอบที่มีศักยภาพ ขณะที่วิธีฮิวริสติกมีประสิทธิภาพสูงในการปรับปรุงคำตอบเชิงลึก (exploitation) ให้ได้ผลลัพธ์ที่ดียิ่งขึ้น การบูรณาการทั้งสองแนวทางจึงช่วยเพิ่มความสมดุลระหว่างการสำรวจและการปรับปรุงเชิงลึก ส่งผลให้สามารถเพิ่มประสิทธิภาพในการหาคำตอบได้ดี ในการประยุกต์ใช้จริง การผสมผสานเมต้าฮิวริสติกกับวิธีฮิวริสติกสามารถสร้างผลลัพธ์ที่มีความเหมาะสมมากขึ้นสำหรับการแก้ปัญหาทางวิศวกรรม โดยเฉพาะในระบบที่มีข้อจำกัดด้านเวลา และทรัพยากร อีกทั้งยังช่วยเพิ่มความยืดหยุ่นในการปรับใช้กับปัญหาที่มีความหลากหลาย และซับซ้อน ซึ่งถือเป็นทิศทางที่มีศักยภาพสูงในการพัฒนางานวิจัยในอนาคต

การแก้ปัญหการจัดเส้นทางที่มีความซับซ้อน จำเป็นต้องใช้แบบจำลองหรือวิธีที่มีประสิทธิภาพช่วยในการแก้ไขปัญห เช่น การใช้วิธี ฮิวริสติกส์ ซึ่งเป็นวิธีการแก้ปัญหาที่เป็นปัญหาแบบปฏิบัติหรือกฎเบื้องต้น แม้ว่าค่าตัวแปรจะไม่แน่นอนแต่การคำนวณมีประสิทธิภาพ และการหาเส้นทางที่แน่นอนมักใช้เวลานานเกินไปในการคำนวณ (Lin & Kernighan, 1973; Bellman, 1962) นอกจากนี้ยังมีวิธีเมต้าฮิวริสติก ซึ่งเป็นโครงสร้างขั้นสูงของอัลกอริทึมที่ถูกพัฒนาและปรับปรุงจากวิธีฮิวริสติกส์ ออกแบบมาสำหรับการแก้ปัญหาที่มีความซับซ้อนได้รับความนิยมและถูกใช้กันอย่างกว้างขวางในปัจจุบัน เนื่องจากสามารถให้คำตอบประมาณที่มีคุณภาพเพียงพอต่อการวางแผน และช่วยลดเวลาในการคำนวณสำหรับปัญหาที่มีขนาดใหญ่หรือแก้ได้ยาก (Dorigo & Gambardella, 1997; Gendreau & Potvin, 2010; Mirjalili et al., 2014) แม้จะมีการพัฒนาอัลกอริทึมหลายรูปแบบเพื่อแก้ปัญหา TSP แต่แนวทางปัจจุบันยังคงเผชิญข้อจำกัดหลายประการ โดยอัลกอริทึมเชิง exact เช่น branch-and-bound หรือ dynamic programming สามารถให้คำตอบที่เป็น optimal solution ได้ แต่มีความซับซ้อนในการคำนวณสูงและไม่เหมาะสมกับปัญหาขนาดใหญ่เนื่องจากเวลาการประมวลผลเพิ่มขึ้น (Applegate et al., 2007; Bellman, 1962; Lawler et al., 1985) ขณะเดียวกัน

อัลกอริทึม heuristic และ metaheuristic เช่น Genetic Algorithm (GA) (Holland, 1992), Ant Colony Optimization (ACO) (Dorigo & Gambardella, 1997) หรือ Fruit Fly Optimization (FOA) แบบดั้งเดิม (Han et al., 2018) และได้กลายเป็นทางเลือกสำคัญที่ใช้กันอย่างแพร่หลายในการแก้ปัญหาเส้นทาง (Laporte, 2010; Pan, 2012; Zhang et al., 2015) แม้จะให้ผลลัพธ์เร็วขึ้น แต่มักมีความแม่นยำต่ำกว่าคำตอบ optimal และมีความไวต่อการตั้งค่าพารามิเตอร์ นอกจากนี้ วิธีการที่มีอยู่หลายแบบยังมีข้อจำกัดในการแก้ปัญหาที่ซับซ้อนหรือมีมิติสูง ทำให้ขาดความสามารถในการปรับตัวต่อการเปลี่ยนแปลงของขนาดปัญหา เงื่อนไขไดนามิก หรือความต้องการในงานโลจิสติกส์จริง ข้อจำกัดเหล่านี้เป็นแรงจูงใจสำคัญในการพัฒนาอัลกอริทึมที่สามารถสร้างสมดุลระหว่างประสิทธิภาพการคำนวณ คุณภาพของคำตอบ และความสามารถในการปรับตัวเพื่อนำไปใช้ในงานปฏิบัติจริง

อัลกอริทึมแมลงวันผลไม้ Fruit Fly Optimization Algorithm (FOA) (Han et al., 2018) เป็นอัลกอริทึมที่ถูกคิดค้นมาจากธรรมชาติ เป็นการเลียนแบบพฤติกรรมของแมลงวันผลไม้ที่เคลื่อนตัวรวมกัน อัลกอริทึมนี้อยู่ในหมวดหมู่ของอัลกอริทึมที่ได้แรงบันดาลใจจากการเคลื่อนไหวของกลุ่ม (population base) โดยที่ประชากรของตัวแทนมีการค้นหาแบบร่วมกันเพื่อหาทางออกที่ดีที่สุดในพื้นที่ของการแก้ปัญหา อัลกอริทึม FOA จำลองพฤติกรรมในการหาอาหารของแมลงวันผลไม้ที่กำลังมองหาแหล่งอาหาร และได้รับการปรับให้ใช้ในการแก้ปัญหาหลายกรณี อัลกอริทึม FOA เป็นอัลกอริทึม Swarm Intelligence Algorithm ที่ถูกนำเสนอเมื่อไม่นานมานี้ อัลกอริทึมนี้ถูกสร้างขึ้นจากพฤติกรรมทางธรรมชาติของแมลงวันผลไม้เป็นวิธีในการค้นหาทางไปยังอาหาร โดยทั่วไปแล้วแมลงวันผลไม้มีชีวิตอยู่ในภูมิภาคเขตร้อน (Pan et al., 2014) แมลงวันผลไม้จะบินไปทางพื้นที่ที่พวกมันสามารถตรวจจับกลิ่นอาหารได้ ขณะที่พวกมันเข้าใกล้กับพื้นที่อาหารของพวกมัน พวกมันจะค้นพบสารโดยใช้การมองเห็นของพวกมัน อัลกอริทึมแมลงวันผลไม้ที่ได้รับการปรับปรุง ถูกใช้กันอย่างแพร่หลายในสาขาวิทยาศาสตร์และวิศวกรรม เนื่องจากความง่ายในการคำนวณและยังเพราะว่าอัลกอริทึมนี้เข้าใจได้ง่ายนอกจากนี้ ยังสามารถหาคำตอบได้สำหรับปัญหาที่ซับซ้อนมากๆ ในเวลาที่รวดเร็ว (Wang et al., 2017) FOA ยังถูกนำมาใช้ในการแก้ปัญหาทางด้านวิศวกรรมมากมาย เช่น โปรแกรมที่ไม่เป็นเส้นตรง (Nonlinear programming problems) (Li et al., 2013; Lin, 2013; Shan et al., 2013) FOA มีความได้เปรียบมากมายเนื่องจากมีค่าพารามิเตอร์ที่น้อย และเวลารัน (Running time) ที่สั้นเมื่อเทียบกับอื่นๆ ง่ายต่อการเขียนโปรแกรมและสามารถปรับเปลี่ยนไปสู่การประยุกต์ใช้กับงานอื่นๆ แม้ FOA จะแสดงให้เห็นถึงประสิทธิภาพในการแก้ปัญหาวิศวกรรมที่ต่อเนื่อง และไม่เป็นเชิงเส้นที่ซับซ้อน เนื่องจากความเรียบง่าย ความต้องการพารามิเตอร์ที่น้อย และประสิทธิภาพในการคำนวณ (Wang et al., 2017; Li et al., 2013) แต่การนำไปประยุกต์ใช้กับปัญหาการเดินทางของพนักงานขายซึ่งเป็นปัญหาแบบไม่ต่อเนื่อง (discrete) และการจัดลำดับ (combinatorial) ซึ่งถูกกำหนดโดยการเรียงลำดับของเมือง งานวิจัยที่ผ่านมา (Lin & Kernighan,

1973; Pan et al., 2014; Wang et al., 2017) ได้ชี้ให้เห็นว่า FOA แบบดั้งเดิมมักประสบปัญหาการหาคำตอบที่ติดอยู่ในค่าเหมาะสมย่อย (premature convergence) ขาดความหลากหลายของวิธีแก้ปัญหา และไม่มีประสิทธิภาพเพียงพอเมื่อใช้กับ TSP ที่มีขนาดใหญ่หรือมีข้อจำกัดซับซ้อน อีกทั้ง FOA แบบดั้งเดิมยังคงจำกัดอยู่ในพื้นที่ค้นหาแบบ 2 มิติ (2D) ทำให้ไม่เหมาะสมกับปัญหาไม่ต่อเนื่องทำให้ความสามารถในการปรับตัวและการหาคำตอบยังด้อยกว่าวิธีการค้นหาเฉพาะที่ (local search) โดยช่องว่างการวิจัยที่สำคัญคือรูปแบบของ FOA มาตรฐานไม่สามารถนำมาใช้โดยตรงกับลักษณะที่ไม่ต่อเนื่องของ TSP รวมถึงความสามารถในการค้นหาแบบโดยรวม (global search) และการค้นหาเฉพาะที่ยังไม่ดีเท่าที่ควร

ดังนั้นผู้วิจัยจึงทำการศึกษารูปแบบของอัลกอริทึมการเพิ่มประสิทธิภาพของ FOA เพื่อให้สามารถประยุกต์ใช้ได้อย่างมีประสิทธิภาพยิ่งขึ้นกับปัญหา TSP ที่ซับซ้อน โดยการนำเสนอการปรับค่าพารามิเตอร์ (parameter) โดยการเพิ่มโดเมนชั้น (dimension) จากเดิมเป็น 2 มิติ ให้เป็น 3 มิติ (3D) และมิติที่สูงขึ้น (Hyper Dimensional) รวมถึงการปรับเปลี่ยนแบบการสลับ 2-Opt, 3-Opt และ 2-3 Opt ร่วมกับการการใช้วิธีฮิวริสติกของ Savings-algorithm สำหรับการสร้างประชากรเริ่มต้น เพื่อให้ได้ผลเฉลยที่เหมาะสมที่สุดในการค้นหา รวมถึงการเพิ่มความแม่นยำของคำตอบ และทำให้อัลกอริทึมสามารถแก้ปัญหา TSP ที่ซับซ้อนได้ดียิ่งขึ้นนำไปสู่ วัตถุประสงค์ของงานวิจัยในครั้งนี้โดยการเพิ่มมิติและปรับปรุง FOA ให้สามารถแก้ปัญหา TSP ได้อย่างมีประสิทธิภาพมากขึ้น

2. วัตถุประสงค์

2.1 เพื่อปรับปรุงประสิทธิภาพกระบวนการในการหาคำตอบ โดยพิจารณาจากค่าคำตอบเทียบกับคำตอบที่เหมาะสมที่สุด (optimal solution) ในการแก้ปัญหา TSP ให้ดียิ่งขึ้น โดยการเพิ่มโดเมนชั้น จาก 2 มิติ เป็น 3 มิติ และมิติที่สูงขึ้น

2.2 เพื่อปรับปรุงการหาผลเฉลยเดิมของอัลกอริทึมแมลงวันผลไม้ที่มีอยู่ให้มีค่าใกล้เคียงกับคำตอบที่ดีที่สุด

3. ทบทวนวรรณกรรม

3.1 ปัญหาการเดินทางของพนักงานขาย

ปัญหาการเดินทางของพนักงานขาย (TSP) เป็นหนึ่งในปัญหาคลาสสิกที่มีการศึกษาอย่างกว้างขวางในด้านการวิจัยเชิงปฏิบัติการ (Operations Research) และวิทยาการคอมพิวเตอร์ ปัญหานี้กำหนดให้นักเดินทางต้องไปเยือนเมืองหนึ่งครั้งเพียงครั้งเดียวแล้วกลับสู่จุดเริ่มต้น โดยมีเป้าหมายเพื่อลดระยะทางหรือค่าใช้จ่ายรวมให้ต่ำที่สุด (Lawler et al., 1985) ปัญหานี้จัดอยู่ในกลุ่ม NP-hard problem ซึ่งหมายความว่า การหาคำตอบที่เหมาะสมที่สุดในเชิง exact จะใช้เวลาเพิ่มขึ้นแบบเอ็กซีโพเนนเชียลเมื่อจำนวนเมืองเพิ่มขึ้น ในเชิงปฏิบัติ TSP มีความสำคัญสูงเนื่องจากสามารถประยุกต์ใช้กับการวางแผนเส้นทางการขนส่งสินค้า การจัด

เส้นทางรถเก็บขยะ การออกแบบสายการผลิต รวมถึงการจัดเส้นทางโดรนหรือหุ่นยนต์ (Applegate et al., 2007; Laporte, 2010) การแก้ปัญหา TSP อย่างมีประสิทธิภาพจึงเป็นความท้าทายที่นักวิจัยยังคงพยายามพัฒนาแนวทางใหม่ๆ อยู่เสมอ โดยวิธีการแก้ปัญหา TSP แบบดั้งเดิม ได้แก่ วิธีการ Exact Methods เช่น Branch and Bound, Integer Linear Programming, Dynamic Programming สามารถหาคำตอบที่เหมาะสมที่สุดได้แน่นอน (Bellman, 1962) อย่างไรก็ตาม วิธีเหล่านี้มีข้อจำกัดสำคัญคือ คำนวณได้ช้า ไม่สามารถขยายไปยังปัญหาที่มีจำนวนเมืองมากได้อย่างมีประสิทธิภาพ กรณีจำนวนโหนดหรือเมืองมีขนาดใหญ่ขึ้น เนื่องจากข้อจำกัดด้านเวลาและหน่วยความจำ ปัญหานี้เองจึงเป็นแรงบันดาลใจให้มีการพัฒนาวิธีเชิง Heuristic และ Metaheuristic เพื่อใช้ในการแก้ปัญหา TSP ให้ได้คำตอบที่ใกล้เคียงคำตอบที่เหมาะสมที่สุดภายในเวลาที่สมเหตุสมผล และมีประสิทธิภาพ

3.2 วิธีการเมตาฮิวริสติกสำหรับปัญหา TSP

วิธีการแบบเมตาฮิวริสติกเป็นแนวทางการแก้ปัญหาที่ได้รับความนิยมอย่างสูงสำหรับการหาคำตอบของปัญหา TSP เนื่องจากปัญหานี้ถูกจัดให้อยู่ในกลุ่ม NP-hard ซึ่งหมายความว่าไม่สามารถหาคำตอบที่เหมาะสมที่สุดได้อย่างมีประสิทธิภาพด้วยวิธีการเชิงแน่นอน (Exact Methods) เมื่อจำนวนเมืองมีขนาดใหญ่ เมตาฮิวริสติกจึงถูกนำมาประยุกต์ใช้เพื่อให้สามารถหาคำตอบที่ใกล้เคียงกับคำตอบที่เหมาะสมที่สุดภายในระยะเวลาที่เหมาะสม วิธีการเหล่านี้ไม่ได้รับประกันว่าจะได้คำตอบที่ดีที่สุดเสมอไป แต่มีข้อดีตรงที่สามารถค้นหาคำตอบที่มีคุณภาพสูงได้ภายในเวลาจำกัด และมีความยืดหยุ่นต่อการประยุกต์ใช้งานจริง โดยนักวิจัยได้พัฒนาวิธีการแบบเมตาฮิวริสติก ที่มุ่งเน้นการหาคำตอบใกล้เคียงค่าที่เหมาะสมที่สุด (near-optimal solution) ภายในเวลาที่เหมาะสม ตัวอย่างที่โดดเด่น เช่น GA อัลกอริทึมการคัดเลือกโดยธรรมชาติและพันธุกรรม (Holland, 1992), ACO จำลองพฤติกรรมการเดินทางหาอาหารของมด (Dorigo & Gambardella, 1997), Particle Swarm Optimization (PSO) จำลองพฤติกรรมเคลื่อนที่ของฝูงอนุภาค (Kennedy & Eberhart, 1995), Grey Wolf Optimizer (GWO), จำลองพฤติกรรมล่าเหยื่อของหมาป่าสีเทา (Mirjalili et al., 2014) รวมถึง FOA แม้ว่าวิธีเหล่านี้จะให้คำตอบที่มีคุณภาพดี แต่ก็ยังมีข้อจำกัด เช่น การใช้เวลาคำนวณนานเมื่อปัญหามีขนาดใหญ่ รวมถึงการจัดการกับปัญหาการจัดเส้นทางแบบ Discrete Optimization ยังค่อนข้างจำกัด และ วิธีการแบบเมตาฮิวริสติกแบบดั้งเดิมมักมีข้อเสียคือ การลู่เข้าสู่คำตอบ (convergence) ช้า และมีโอกาสติด local optimum (Pan, 2012)

3.3 การพัฒนา และปรับปรุง FOA

อัลกอริทึมแมลงวันผลไม้แบบเดิม เริ่มต้นโดยอาศัยการสุ่มตำแหน่ง การประเมินค่า (fitness) และการอัปเดตตำแหน่งที่ดีที่สุด ซึ่งมีโครงสร้างง่าย ยังคงมีข้อจำกัดบางประการ เช่น ความสามารถในการ

หลีกเลี่ยงการติดอยู่ในคำตอบเฉพาะที่ ประสิทธิภาพในการค้นหาวางกว้าง (global search ability) ที่ยังไม่สูงพอ และความเร็วในการเข้าสู่คำตอบ (convergence speed) ที่อาจไม่เหมาะสมสำหรับปัญหาที่มีความซับซ้อนสูง หรือมีมิติข้อมูลจำนวนมาก ด้วยเหตุนี้ นักวิจัยจำนวนมากจึงได้พัฒนาและปรับปรุง FOA ให้มีความสามารถที่ดีขึ้น ผ่านการประยุกต์ใช้แนวคิดใหม่ ๆ ดังแสดงในตารางที่ 1 โดยงานวิจัยที่เกี่ยวข้องกับการประยุกต์ใช้อัลกอริทึม FOA ในการแก้ปัญหา TSP สามารถแบ่งออกได้เป็น 3 แนวทางหลัก ได้แก่ 1) การปรับปรุงพารามิเตอร์ (Parameter Adjustment), 2) การเพิ่มประสิทธิภาพการค้นหา (Search Enhancement) และ 3) การผสมผสาน FOA กับเทคนิคอื่น (Hybridization with Other Methods) ดังนี้

3.3.1 การปรับปรุงอัลกอริทึม FOA โดยการปรับพารามิเตอร์

การปรับพารามิเตอร์ของ (Iskan & Gunduz, 2017) โดยการใช้ LGMS-FOA ซึ่งใช้การปรับค่าพารามิเตอร์เพื่อแก้ข้อจำกัดของ FOA แบบดั้งเดิม ผลการทดลองพบว่า LGMS-FOA สามารถเพิ่มคุณภาพการค้นหา และลดการติดอยู่ใน local optimum ได้ และ (Jiang & Yang, 2016) การเพิ่ม Elimination Mechanism และการปรับการทำงานในช่วงการมองเห็นของ FOA เพื่อเพิ่มความหลากหลายของประชากร และเร่งอัตราการเข้าสู่ (convergence rate) โดยการปรับพารามิเตอร์ช่วยลดการติดอยู่ใน local optimum และทำให้ FOA มีความเสถียรขึ้น และมีข้อจำกัดจากปรับพารามิเตอร์เพียงอย่างเดียวไม่สามารถแก้ปัญหา TSP ที่มีขนาดใหญ่และซับซ้อนได้ทั้งหมด เนื่องจากยังขาดกลไกการค้นหาที่หลากหลาย (Zhang et al., 2023) สำหรับการพัฒนา Adaptive Fruit Fly Optimization Algorithm (AFOA) เพื่อแก้ปัญหา TSP แบบ Discrete โดยเสนอการปรับพารามิเตอร์ให้เหมาะสมกับสภาพการค้นหาแบบพลวัต ส่งผลให้อัลกอริทึมสามารถปรับตัวได้ตามลักษณะของปัญหา ลดการติดอยู่ใน local optima และช่วยเพิ่มความเร็วในการหาคำตอบที่ใกล้เคียงค่าเหมาะสมที่สุด

ตารางที่ 1 การตรวจเอกสารงานวิจัยที่เกี่ยวข้องกับการพัฒนาและปรับปรุง FOA ในการแก้ปัญหา TSP

ผู้วิจัย (ปี)	วิธีการ	มิติที่ใช้	ปัญหาที่แก้	ข้อจำกัด	Research Gap
Shan et al. (2013)	FOA แบบหลายฝูง + ตัวดำเนินการ IPOX/MPX + กระบวนการสื่อสารฝูง + DOE (Taguchi)	Discrete (กำหนดการ/ลำดับงาน) 2D	SFTSP (Semiconductor Final Testing Scheduling)	เฉพาะโดเมน SFTSP; การโอนย้ายไป TSP ยังไม่ชัด	ถ่ายโอนไป TSP โดยตรง; ทดสอบกับ TSPLIB; มุมมองหลายมิติ/โครงสร้าง Hyper-D
Zheng et al. (2014)	DFOA (Discrete FOA) + crossover เฉพาะทาง + Tasting/Smelling + EXE	Discrete (Permutation space) 2D	TSP (ปรับปรุงการสำรวจ)	คุณภาพยังด้อยกว่าเมื่อไม่ผสมผสาน LS แรงๆ; เวลา กับอินสแตนซ์ ใหญ่มาก	ผสมผสาน LK/3-opt เป็นมาตรฐาน; รีสตาร์ท/หลากหลาย; ขยายมิติค้นหา

ตารางที่ 1 (ต่อ)

ผู้วิจัย (ปี)	วิธีการ	มิติที่ใช้	ปัญหาที่แก้	ข้อจำกัด	Research Gap
Jiang & Yang (2016)	Improved FOA + Elimination mechanism + ปรับช่วงการมองเห็น	Discrete (แมปลำดับเมือง) 2D	TSP บน TSPLIB 10 ชุดข้อมูล	ต้องจูนพารามิเตอร์; สเกลกับปัญหาใหญ่ยังจำกัด	ขยายมิติการค้นหา; ไฮบริดกับ local search ระดับสูง; การประเมินบนอินสแตนซ์ขนาดใหญ่
Iskan & Gunduz (2017)	FOA; LGMS-FOA (ปรับพารามิเตอร์และกลไกค้นหา)	2D (ต่อเนื่อง) + แมปสู่ Discrete สำหรับ TSP (จำกัด)	TSP และฟังก์ชันทดสอบไม่เป็นเชิงเส้น 6 ฟังก์ชัน	FOA ดั้งเดิมแก้ TSP ที่ซับซ้อนได้ไม่ดี; ไวต่อพารามิเตอร์; การแมปสู่ดิสครีตยังหยาบ	ต้องการการขยายมิติ (3D/Hyper-D); โอเพอเรเตอร์แบบดิสครีตเฉพาะทาง; ไฮบริดกับ local search
Huang et al. (2017)	Improved FOA: รักษา Osphresis; ปรับ Visual เป็น 2 โหมด (30% ดีสุด + สุ่ม)	Discrete/เมทริกซ์ลำดับเมือง (เทียบเท่า 2D)	TSP (ปรับปรุงการสำรวจ/การมองเห็น)	ยังเสี่ยงลู่เข้าเร็ว; ฟังก์ชันการเริ่มต้น; ไม่มี local search ที่ทรงพลัง	ผนวก 2-opt/3-opt/LK; กลไกปรับตัวพารามิเตอร์; มุมมองหลายมิติ
Huang & Su (2017)	FOA ปรับลดขนาดก้าว (step) + ไฮบริด GA (crossover/mutation) + Dynamic balance	Discrete (ไฮบริด GA-FOA) 2D	TSP (บาลานซ์ global/local; ป้องกัน premature)	โครงสร้างซับซ้อนขึ้น; พารามิเตอร์มาก; หลักฐานบนกรณีใหญ่มากรวมถึงยังจำกัด	เทียบกับ LKH อย่างเป็นระบบ; ขยายมิติ; กลไกควบคุมปรับตัวอัตโนมัติ
Crişan et al. (2020)	DFOA ภายใต้ข้อมูลไม่แน่นอน; ปรับตามระดับ uncertainty	Discrete 2D (Permutation) + การจัดการความไม่แน่นอน	TSP กับข้อมูลไม่สมบูรณ์/ไม่แน่นอน	ขาดการปรับปรุงประสิทธิภาพของ DFOA	การพัฒนา DFOA เพื่อเพิ่มประสิทธิภาพ
Liu et al. (2023)	Hybrid FOA + PSO	Discrete 2D	TSP	ประสิทธิภาพดีขึ้นแต่ยังจำกัดที่การ hybrid เท่านั้น, ไม่เน้น search space ขนาดใหญ่	ขาดการศึกษา FOA ในมิติที่สูงขึ้น (3D+) และยังไม่ผสาน local heuristics (เช่น 2-Opt, 3-Opt)

ตารางที่ 1 (ต่อ)

ผู้วิจัย (ปี)	วิธีการ	มิติที่ใช้	ปัญหาที่แก้	ข้อจำกัด	Research Gap
Ranjan & Kumar (2023)	Systematic Review ของ FOA และการประยุกต์	Discrete 2D	ปัญหาหลากหลาย (optimization, scheduling, TSP)	ไม่ได้พัฒนา algorithm ใหม่, ส่วนใหญ่ยังอยู่ที่ FOA แบบดั้งเดิม	ขาดการนำเสนอวิธีพัฒนา FOA ที่ผสมผสาน heuristics และรองรับ TSP ขนาดใหญ่
Zhang et al. (2023)	Adaptive FOA (ปรับพารามิเตอร์แบบพลวัต)	2D	Discrete TSP	ยังจำกัดใน 2D, ไม่รองรับการขยายสู่มิติสูง, ยังไม่บูรณาการ heuristic	ขาดการขยาย FOA เป็น 3D/Hyper Dimensional และการบูรณาการ heuristic สำหรับ local search
Almufti et al. (2025)	Comparative Analysis ของ metaheuristics (รวม FOA)	Discrete 2D	TSP (เปรียบเทียบหลาย algorithm)	วิเคราะห์เชิงเปรียบเทียบ ไม่ได้พัฒนา FOA, ข้อจำกัดชัดเจนใน TSP ที่ซับซ้อน	ขาดการปรับปรุง FOA โดยตรงเพื่อแข่งขันกับ metaheuristics อื่น ในปัญหา TSP ขนาดใหญ่

3.3.2 การเพิ่มประสิทธิภาพการค้นหา

การปรับปรุง FOA โดยแบ่งการมองเห็น (visual phase) ออกเป็นสองส่วน ส่วนแรกใช้ข้อมูลจากผลลัพธ์ที่ดีที่สุด ส่วนที่สองสร้างใหม่จากการสุ่ม เพื่อเพิ่มความหลากหลายของคำตอบ และป้องกันไม่ให้ FOA ติดกับดัก local optimum (Huang et al., 2017) รวมถึง (Zheng et al., 2014) การนำเสนอ Discrete Fruit Fly Optimization Algorithm (DFOA) โดยออกแบบตัวดำเนินการ crossover และกระบวนการ elimination (EXE) เพื่อเพิ่มประสิทธิภาพการสำรวจพื้นที่ในการหาคำตอบ (solution space) โดยเทคนิคการเสริมกลไกการค้นหา (search operators) ทำให้ FOA สามารถสำรวจ solution space ได้ลึกขึ้น ในด้านของคุณภาพคำตอบดีขึ้น แต่ยังคงมีปัญหาในการปรับให้เหมาะกับปัญหา discrete ขนาดใหญ่ และต้องใช้เวลาในการปรับค่าสำหรับการค้นหาคำตอบ (Ranjan & Kumar, 2023) ในการทบทวนเชิงระบบเกี่ยวกับ FOA และการประยุกต์ใช้งานในหลายด้าน โดยพบว่าม้งานวิจัยจำนวนมากที่มุ่งเน้นการพัฒนา FOA ในแง่การปรับปรุงกลไกการค้นหา (search operators) และการขยายพื้นที่การค้นหา ควบคู่กับการเน้นเจาะจงพื้นที่ที่มีแนวโน้มคำตอบที่ดี ซึ่งสะท้อนถึงแนวโน้มการวิจัยเพื่อเพิ่มประสิทธิภาพการค้นหา

3.3.3 การผสมผสาน FOA กับเทคนิคอื่น

การผสมผสาน FOA กับตัวดำเนินการในรูปแบบการปรับปรุงกระบวนการในการหาคำตอบ (Improved precedence operation crossover (IPOX)) และการปรับปรุงแบบหลายจุด (Modified multipoint preservative crossover (MPX)) ในการแก้ปัญหา (Shan et al., 2013) โดยการนำเสนอ อัลกอริทึมในการแก้ปัญหาการจัดตารางการทดสอบขั้นสุดท้ายของเซมิคอนดักเตอร์ (The semiconductor final testing scheduling problem (SFTSP)) ทำให้ FOA สามารถใช้กับโจทย์ combinatorial ได้ดีขึ้น (Huang & Su, 2017) ในการผสมผสาน FOA กับ GA โดยเพิ่มกลไก crossover และ mutation ส่งผลให้เกิดกลไกการสมดุลแบบพลศาสตร์ (Dynamic balance mechanism) ระหว่าง global search และ local search ลดโอกาสการมีสถานะที่อัลกอริทึมเชิงวิวัฒนาการหรือการหาค่าเหมาะสม หยุดพัฒนา และหาคำตอบไปคงอยู่ที่ค่าใดค่าหนึ่งเร็วเกินไป โดยที่คำตอบนั้นยังไม่ใช่คำตอบที่เหมาะสมที่สุด แต่เป็นเพียงคำตอบเฉพาะที่ (premature convergence) และเพิ่มความหลากหลายของประชากร (Crişan et al., 2020) การใช้ DFOA กับข้อมูลที่ไม่แน่นอน (uncertain data) โดยเน้นการปรับตัวของอัลกอริทึมให้สามารถแก้ TSP ในบริบทที่ข้อมูลไม่สมบูรณ์ การผสมผสาน FOA กับเทคนิคอื่นทำให้ FOA มีความสามารถรอบด้านมากขึ้น ทั้งการสำรวจเชิง global และการปรับปรุงเชิง local แต่เมื่อโครงสร้างอัลกอริทึมซับซ้อนขึ้น และอาจทำให้สูญเสียความง่ายในการใช้งานที่เป็นจุดแข็งของ FOA และ (Huang et al., 2017) ในการทำการวิเคราะห์ปัญหาการเดินทางของพนักงานขาย โดยใช้ FOA ร่วมกับ GA ในส่วนของการดำเนินการ crossover และ mutation ซึ่งเพิ่มความหลากหลายในประชากร และยับยั้งการเกิดปัญหา Generation of premature ได้เป็นอย่างดี (Liu et al., 2023) การประยุกต์ใช้อัลกอริทึมลูกผสม (Hybrid Metaheuristic) โดยการผสมผสาน FOA เข้ากับ PSO เพื่อนำมาแก้ปัญหา TSP ผลการทดลองแสดงให้เห็นว่าการผสมผสานดังกล่าวช่วยให้ได้ผลลัพธ์ที่ดีกว่าอัลกอริทึมเดี่ยว โดยเฉพาะในแง่ของความเร็วในการหาคำตอบและคุณภาพของเส้นทางที่ได้ และ (Almufti & Shaban, 2025) ในการศึกษาเชิงเปรียบเทียบอัลกอริทึม Metaheuristic ต่าง ๆ รวมถึง FOA สำหรับปัญหา TSP โดยพบว่า การผสมผสาน FOA กับวิธีอื่น ๆ ช่วยเพิ่มศักยภาพในการแก้ปัญหา โดยเฉพาะอย่างยิ่งเมื่อเปรียบเทียบกับ FOA แบบดั้งเดิมที่มีข้อจำกัดด้านความเร็วและความแม่นยำ

3.3.4 ช่องว่างการวิจัย และการมีส่วนร่วมของงานวิจัยนี้

จากการทบทวนพบว่า แนวโน้มการปรับปรุง FOA มี 3 ทิศทางหลัก ได้แก่ การปรับพารามิเตอร์เพื่อเพิ่มความเสถียร, การออกแบบกลไกการค้นหาใหม่เพื่อเพิ่มประสิทธิภาพการสำรวจ และการผสมผสานกับเทคนิคอื่นเพื่อเพิ่มความสามารถทั้ง global และ local search (hybridization) โดยงานวิจัยที่ผ่านมายังมีข้อจำกัดคือ ยังขาดการขยายมิติของพื้นที่ค้นหา (dimension expansion) ซึ่งอาจช่วยเพิ่มความหลากหลายและความสามารถในการหลีกเลี่ยง local optimum และยังไม่มีการใช้ Multi-Dimensional FOA (3D หรือ Hyper-dimensional FOA) ในการแก้ปัญหา TSP โดยตรง และการสร้างประชากรเริ่มต้นยังไม่ถูกพัฒนาอย่างเป็นระบบ ทำให้คุณภาพของ ผลเฉลยเริ่มต้นต่ำ และส่งผลต่อคุณภาพสุดท้าย

ดังนั้นงานวิจัยนี้จึงทำการพัฒนาเพื่อจะเติมเต็มช่องว่างดังกล่าว โดย

1. เพิ่มมิติของ FOA จาก จากเดิมเป็น 2 มิติ ให้เป็น 3 มิติ และมิติที่สูงขึ้น เพื่อเพิ่มความสามารถในการสำรวจ solution space
2. การบูรณาการกับวิธีการฮิวริสติก (2-Opt, 3-Opt, 2-3Opt และ Savings Algorithm) เข้ากับ FOA เพื่อสร้างประชากรเริ่มต้นที่มีคุณภาพ และเพิ่มประสิทธิภาพของ local search
3. ประเมินประสิทธิภาพของ FOA ที่ได้รับการปรับปรุง กับ TSP ขนาดใหญ่และซับซ้อน เพื่อแสดงให้เห็นถึงความเหนือกว่าเมื่อเทียบกับ FOA แบบดั้งเดิมและ metaheuristics อื่น ๆ

3.4 การปรับปรุงขั้นตอนของอัลกอริทึม FOA

ทางผู้วิจัยได้ทำการปรับปรุงขั้นตอนของอัลกอริทึม FOA โดยการเพิ่มมิติในกระบวนการต่างๆ ดังนี้

ขั้นตอนที่ 1 การกำหนดประชากรเริ่มต้นประชากรเริ่มต้น กำหนดตำแหน่งหลักของกลุ่มแมลงวันผลไม้ถูกสร้างขึ้นโดยสุ่มปริภูมิในพิกัด X, Y และ Z เป็นค่าเริ่มต้น โดยมีลักษณะเป็นผิวระนาบ (plan surface) หรือ ($X_{i\text{axis}}, i \in D^n$)

$$X_{\text{initial}} = \text{Ini}X_{\text{axis}} \quad (1)$$

$$Y_{\text{initial}} = \text{Ini}Y_{\text{axis}} \quad (2)$$

$$Z_{\text{initial}} = \text{Ini}Z_{\text{axis}} \quad (3)$$

ขั้นตอนที่ 2 การปรับพารามิเตอร์ จากรูปแบบของการกำหนดค่าเริ่มต้นนั้นสามารถพิจารณาปริภูมิที่มีความแตกต่างได้ในหลายสถานะ เช่น ปริภูมิปริมาตร ลักษณะเป็นปริมาตร (volume surfaces), $x \in (D^3: X Y Z)$, ปริภูมิที่สูงกว่าสองมิติ (hyper volumes), $x \in (D^N: X_1 X_2 \dots X_N)$,

$$X_i = X_{\text{axis}} + \text{Random value} \quad (4)$$

$$Y_i = Y_{\text{axis}} + \text{Random value} \quad (5)$$

$$Z_i = Z_{\text{axis}} + \text{Random value} \quad (6)$$

ขั้นตอนที่ 3 ประเมินค่าฟิตเนส (fitness) ประเมินค่าฟิตเนสของแต่ละตัวแทนในประชากร โดยฟังก์ชันนี้บ่งบอกถึงความเหมาะสมของโซลูชัน หรือสิ่งที่ต้องการให้มีค่าสูงที่สุด เพื่อใช้ในการสร้างรุ่นต่อไปเพื่อประเมินประสิทธิภาพของอัลกอริทึม หลังจากมีการปรับแต่งพารามิเตอร์ของอัลกอริทึม

$$\text{Dist}_i = \sqrt{X_i^2 + Y_i^2 + Z_i^2} \quad (7)$$

หรือ

$$\text{Dist}_i = \sqrt{X_1^2 + X_2^2 + \dots + X_i^2} \quad i=1,2,\dots,n \quad (8)$$

และขั้นตอนที่ 6 การอัปเดตตำแหน่ง ฟังก์ชันการอัปเดตตำแหน่งจะคำนวณตำแหน่งใหม่ของแมลงวันผลไม้โดยใช้ตำแหน่งเก่า ตำแหน่งของแมลงวันผลไม้ที่มีค่าฟิตเนสสูงสุด และค่าสุ่ม แมลงวันผลไม้จะบินไปยังแหล่งอาหารที่ดีที่สุด (X_{best})

$$\text{Smellbest}=\text{bestSmell} \quad (9)$$

$$\text{Xaxis}=\text{X}(\text{BestIndex}) \quad (10)$$

$$\text{Yaxis}=\text{Y}(\text{BestIndex}) \quad (11)$$

$$\text{Zaxis}=\text{Z}(\text{BestIndex}) \quad (12)$$

3.5 การปรับปรุงการการปรับค่าพารามิเตอร์ (Parameter) โดยการเพิ่มโดเมนชั้น (Dimension)

การปรับค่าพารามิเตอร์ใน FOA และโดเมนชั้น เป็นการเพิ่มจำนวนตัวแปรหรือมิติในการแทนคำตอบของปัญหา เป็นวิธีที่ช่วยให้โมเดลสามารถจำแนกความหมายหรือลักษณะของข้อมูลได้เป็นอย่างดี การเพิ่มมิติจะช่วยให้โมเดลสามารถจัดหมวดหมู่หรือจำแนกข้อมูลที่ซับซ้อนได้ดีขึ้น โดยเฉพาะเมื่อมีการเพิ่มมิติให้กับอัลกอริทึม FOA เป็น 3 มิติ หรือมิติที่สูงขึ้น จะช่วยให้โมเดลสามารถจัดการกับปัญหาที่ซับซ้อนและมีขนาดใหญ่ขึ้นได้มากขึ้น โดยอัลกอริทึม FOA จะต้องปรับการทำงานในมิติเพิ่มเติมให้เหมาะสมกับการแก้ปัญหาในมิติที่เพิ่มขึ้นด้วยการปรับค่าพารามิเตอร์และโครงสร้างของอัลกอริทึมให้เหมาะสมกับปัญหาที่เกิดขึ้น เมื่อมีการเพิ่มโดเมนชั้น จำนวนตัวแปรหรือมิติในการแทนคำตอบของปัญหาจะเพิ่มขึ้น ซึ่งจำนวนนี้จะถูกนำมาใช้ในกระบวนการคำนวณหรือการสร้างคำตอบใหม่ ช่วยเพิ่มความหลากหลายในคำตอบที่เป็นไปได้ จำเป็นต้องปรับค่าพารามิเตอร์หรือโครงสร้างของอัลกอริทึมให้เหมาะสมกับจำนวนโดเมนชั้นที่เพิ่มขึ้น

การเพิ่มมิติที่สูงขึ้นคือการนำเข้าสู่ข้อมูลหรือการทำงานในมิติที่มีความซับซ้อนมากขึ้นกว่ามิติสามที่เราคุ้นเคย การใช้มิติที่สูงขึ้นสามารถช่วยให้โมเดลหรืออัลกอริทึมสามารถจัดการกับข้อมูลที่มีความซับซ้อน และมีความใหญ่ขึ้นได้ โดยการปรับค่าพารามิเตอร์ และโครงสร้างของอัลกอริทึมให้เหมาะสมกับการทำงานในมิติที่สูงขึ้น ยกตัวอย่างเช่น การเพิ่มมิติใน FOA สามารถทำให้โมเดลสามารถจัดหมวดหมู่ข้อมูลในพื้นที่หมายเลขมากขึ้นได้ หรือช่วยให้อัลกอริทึมสามารถจัดการกับข้อมูลที่มีลักษณะซับซ้อน และมีความซับซ้อนมากขึ้นได้ดี

การทำการสลับ Swap และ Swop แบบ 2-opt และ 3-opt เป็นเทคนิคที่ใช้ในการปรับปรุงเส้นทางในปัญหา TSP เพื่อลดระยะทางทั้งหมดของเส้นทาง โดยการสลับหรือเปลี่ยนตำแหน่งของเมืองที่ถูกเยี่ยมชมในเส้นทาง แบบ 2-opt ทำการสลับเส้นทางระหว่างสองขาที่ตัดกัน เพื่อให้ได้เส้นทางใหม่ที่มีระยะทางสั้นลง และใน 3-opt จะทำการสลับเส้นทางระหว่างสามขาที่ตัดกันเพื่อให้ได้เส้นทางที่มีระยะทางสั้นลงอีก และการ Insertion position แบบ 2-opt และ 3-opt ในบทบาทของการแก้ปัญหา TSP คือตำแหน่งที่เหมาะสมที่สุดในเส้นทางที่กำลังพิจารณาเพื่อแทรกเมืองใหม่เข้าไปในเส้นทางนั้น การเลือกตำแหน่งที่เหมาะสมสามารถทำได้โดยพิจารณาระยะทางระหว่างเมืองและเมืองที่กำลังพิจารณา และเลือกตำแหน่งที่ทำให้ระยะทางรวมของเส้นทางลดลงหรือเท่าที่เป็นไปได้ ซึ่งการเลือกตำแหน่งให้เหมาะสมนี้เป็นส่วนสำคัญในการปรับปรุงเส้นทางใน TSP เพื่อให้ได้เส้นทางที่มีความสั้นที่สุดหรือเส้นทางที่มีประสิทธิภาพมากที่สุดที่เป็นไปได้ การใช้ insertion position เป็นเครื่องมือที่สำคัญในการพัฒนาวิธีการแก้ปัญหา TSP ที่มีประสิทธิภาพ และการใช้ Savings

Algorithm เข้ามาช่วยสร้างประชากรเริ่มต้น (initial population) ของ FOA เพื่อเพิ่มคุณภาพของการหาคำตอบเริ่มต้น ก่อนเข้าสู่กระบวนการค้นหาเชิง metaheuristic ของ FOA ซึ่งเป็นแนวทางที่ผสมผสานความได้เปรียบของ heuristic (ความเร็ว และการหาคำตอบของ Savings Algorithm) เข้ากับ metaheuristic (ความสามารถในการสำรวจพื้นที่คำตอบขนาดใหญ่ของ FOA) ซึ่งส่งผลโดยตรงต่อประสิทธิภาพของการค้นหาคำตอบที่เหมาะสมที่สุดในปัญหา TSP

4. วิธีดำเนินงานวิจัย

วิธีการดำเนินการวิจัยการปรับปรุงประสิทธิภาพการหาคำตอบที่เหมาะสมโดยวิธีการอัลกอริทึมแมลงวันผลไม้ (FOA) สำหรับการเดินทางของพนักงานขาย (TSP) โดยมีวัตถุประสงค์เพื่อทำการปรับปรุงประสิทธิภาพวิธีการแมลงวันผลไม้ หาคำตอบที่เหมาะสมสำหรับปัญหาการเดินทางของพนักงาน สามารถนำผลลัพธ์ที่ได้ประยุกต์ใช้ในการในงานวิจัยที่เกี่ยวข้องกับปัญหาการหาคำตอบหรือการจำลองที่ต้องการคำตอบที่เหมาะสม โดยมีวิธีการดำเนินการวิจัยดังต่อไปนี้

4.1 เครื่องมือที่ใช้ในการเก็บรวบรวมข้อมูล

1) เครื่องมือที่ใช้ในการเก็บข้อมูล

ตัวแบบปัญหาจาก ตัวแบบมาตรฐาน โดยมีปัญหาแบบ TSP จำนวน 9 ตัวแบบปัญหา และมีระยะทางที่สั้นที่สุดของแต่ละเมือง ดังแสดงในตารางที่ 2

ตารางที่ 2 ระยะทางที่สั้นที่สุดของแต่ละเมืองที่นำมาใช้ในการหาคำตอบ

No.	Instance	Optimal solution
1	Ulysses16	73.98
2	Ulysses22	75.30
3	Dantzig42	699
4	att48	33,522
5	Eil51	426
6	Berlin52	7,542
7	st70	675
8	Eil76	538
9	rat99	1,211

โดยแหล่งข้อมูลตัวแบบปัญหา TSP ซึ่งเผยแพร่โดยมหาวิทยาลัย Heidelberg นั้น ถือว่าเป็นแหล่งอ้างอิงมาตรฐานสำหรับค่าที่ดีที่สุดที่รู้จัก (best-known) ของตัวอย่างปัญหา TSP (Reinelt, 1991; University of Heidelberg, n.d.) โดยระยะทางรวมที่ได้เป็นค่าคำตอบที่เหมาะสมที่สุด (optimal solution)

2) สำหรับการแก้ปัญหาโดยใช้การปรับปรุงประสิทธิภาพการหาคำตอบที่เหมาะสมโดยวิธีการแมลงวันผลไม้ (FOA) สำหรับการเดินทางของพนักงานขายแบบวัตถุประสงค์เดียว (Single Objective) โดยพิจารณาระยะทางในการเดินทางน้อยที่สุด

3) เครื่องมือที่ใช้ในการวิเคราะห์ข้อมูล

โปรแกรม MATLAB เป็นโปรแกรมสำหรับช่วยงานคำนวณทางคณิตศาสตร์

4.2 วิธีการเก็บรวบรวมข้อมูล

1) ทำการศึกษาค้นคว้าข้อมูลโดยมีหัวข้อดังนี้

- ปัญหาการดำเนินการในการจัดเส้นทาง และการปรับปรุงประสิทธิภาพการหาคำตอบที่เหมาะสมโดยวิธีการแมลงวันผลไม้

- การปรับค่าพารามิเตอร์โดยการเพิ่มโดเมนชั้น และมิติที่สูงขึ้น

- การตรวจสอบเอกสารงานวิจัยที่เกี่ยวข้อง

2) รวบรวมข้อมูลที่เกี่ยวข้องกับงานวิจัย ได้แก่ ปัญหาการดำเนินการในการจัดเส้นทาง การแก้ปัญหาโดยใช้การปรับปรุงประสิทธิภาพการหาคำตอบที่เหมาะสมโดยวิธีการแมลงวันผลไม้ สำหรับการเดินทางของพนักงานขาย

3) ทฤษฎีที่เกี่ยวข้องกับงานวิจัยเพื่อหาวิธีช่วยแก้ปัญหาที่เกี่ยวข้อง

4) หนังสือ บทความ งานวิจัย และวารสารเชิงวิชาการที่เกี่ยวข้องกับการปรับค่าพารามิเตอร์โดยการเพิ่มโดเมนชั้น และมิติที่สูงขึ้น

4.3 การจัดทำข้อมูลและวิเคราะห์ข้อมูล

1) ทำการศึกษาและรวบรวมข้อมูลแล้วนำข้อมูลที่ได้มาวิเคราะห์ปัญหาแล้วแบ่งลำดับขั้นตอนในการดำเนินงาน

2) ศึกษาขั้นตอนวิธีการการปรับปรุงประสิทธิภาพการหาคำตอบที่เหมาะสมโดยวิธีการแมลงวันผลไม้ ได้แก่ ปัญหาการตัดสินใจโดยใช้วิธีการที่เหมาะสม การหาเส้นทางที่เหมาะสมสำหรับการเดินทางเพื่อให้ได้ระยะทางที่สั้นที่สุด

3) การคำนวณฟังก์ชันวัตถุประสงค์ และการปรับค่าพารามิเตอร์โดยใช้โปรแกรม MATLAB

4) ทำการวิเคราะห์หาผลเฉลยที่เหมาะสมในกรณีต่างๆ ได้แก่

- กรณีประยุกต์ใช้สำหรับการหาค่าเหมาะสมในการดำเนินการทางโลจิสติกส์งานวิจัยนี้มุ่งเน้นการปรับปรุงประสิทธิภาพของอัลกอริทึม FOA ในการแก้ปัญหา TSP ทั้งในแง่ ความถูกต้องของคำตอบ และความสามารถในการปรับตัวต่อปัญหาที่ซับซ้อน

- กรณีการปรับปรุงการเปลี่ยนค่าผลเฉลยที่ได้ ร่วมกับวิธีการอื่นๆในฟังก์ชันเพื่อหาค่าที่เหมาะสมสำหรับปัญหาทางวิศวกรรม ได้แก่

1. Differential Evolution Optimization (DEO)
2. Grey Wolf Optimization (GWO)
3. Simulated Annealing (SA)
4. Genetic Algorithm (GA)
5. Particle Swarm Optimization (PSO)
6. Ant Colony Optimization (ACO)
7. Teaching-Learning-Based Optimization (TLBO)
8. Fruit Fly Optimization Algorithm (FOA)
9. Firefly Algorithm (FA)
10. Whale Optimization Algorithm (WOA)

กรณีปัญหาการตัดสินใจโดยใช้วิธีการที่เหมาะสม การหาเส้นทางที่เหมาะสมสำหรับการเดินรถ เพื่อให้ได้ระยะทางรวมที่สั้นที่สุด

5) สรุปผล และเสนอแนะ

4.4 การออกแบบการทดลอง (Experimental Design)

การวิเคราะห์หาผลเฉลยถูกควบคุมตัวแปรให้อยู่ในเงื่อนไขที่เป็นธรรมชาติ โดยใช้การประเมินคงที่ 20,000 evaluations ต่อการให้ผลลัพธ์เฉลย (run) ต่อหนึ่งอัลกอริทึม ต่อหนึ่งชุดข้อมูล (ตัวแบบมาตรฐาน) ซึ่งคำนวณจาก จำนวนประชากร (Population Size) $\times$ จำนวนรอบ (Iterations) สำหรับอัลกอริทึมเชิงประชากรจะปรับ Iterations ให้สอดคล้องกับ Population Size เพื่อให้ evaluations รวมเท่ากับ 20,000 โดยการศึกษาเพิ่มเติมจะสำรวจผลของ Population Size $\in$ {40, 100, 200} โดยปรับ Iterations ให้เป็น {500, 200, 100} ตามลำดับ ภายใต้การประเมินที่คงที่ 20,000 evaluations ในแต่ละอัลกอริทึม และชุดข้อมูลมีเงื่อนไขหยุดที่ 20,000 evaluations การวัดผลประกอบด้วย คุณภาพของคำตอบได้แก่ ค่าที่ดีที่สุด ค่าเฉลี่ย ค่ามากที่สุด ค่าน้อยสุด ค่าเบี่ยงเบนมาตรฐาน และระยะเวลาในการทำงานต่อรอบ (Best, Mean, Fmin, Fmax, Std และ Time (s/round)) ในการวิจัยนี้ FOA ได้กำหนดการใช้วิธีการสลับตำแหน่ง (2-Opt, 3-Opt และ 2-3 Opt) โดยพิจารณาจากสัดส่วนเปอร์เซ็นต์ของเมืองมาเพื่อทำการปรับปรุง ได้แก่ 10%–1%, 20%–2%, 30%–3%, 40%–4% และ 50%–5% โดยพบว่ายิ่งเลือกเปอร์เซ็นต์ที่สูงขึ้นจะใช้เวลาในการคำนวณเพิ่มขึ้นอย่างมาก แม้ว่าจะมีโอกาสได้คำตอบที่ใกล้เคียงค่าที่เหมาะสมที่สุดมากขึ้นก็ตาม เพื่อสร้างสมดุลระหว่างคุณภาพของคำตอบกับเวลาในการประมวลผล เลือกใช้ค่า 20% สำหรับการสลับ 2-Opt และ 2% สำหรับการสลับ 3-Opt ซึ่งเป็นระดับที่เหมาะสมในการลดระยะเวลาการคำนวณแต่ยังคงรักษาประสิทธิภาพของการหาคำตอบได้ดี โดยดูจากผลที่ได้จากการวิเคราะห์ชี้ให้เห็นว่าการใช้สัดส่วนการสลับในระดับปานกลางสามารถช่วยเพิ่มความหลากหลายของเส้นทาง และหลีกเลี่ยงการติดอยู่ในค่าที่ไม่เหมาะสมเฉพาะที่ได้ โดยไม่ทำให้เวลาในการทำงานสูงเกินความจำเป็น รวมถึงการใช้ Savings Algorithm เพื่อสร้างประชากรเริ่มต้น

5. ผลการวิจัย

ในการศึกษาการเพิ่มประสิทธิภาพในการหาคำตอบสำหรับการจัดเส้นทางปัญหาการเดินทางของพนักงานขายโดยวิธีการอัลกอริทึมแมลงวันผลไม้ ได้แบ่งผลการวิเคราะห์ปัญหาออกเป็น

- 1) การพิจารณาหาตัวแปรที่เหมาะสมในการตั้งค่าพารามิเตอร์
- 2) การพิจารณาที่เกิดจากการเรียนรู้ด้วยตัวเอง
- 3) ผลการพิจารณาข้อมูลจำนวน 9 ชุดข้อมูลแบบวัตถุประสงค์เดียวโดยมีปัญหาคือเป็นแบบ TSP ได้แก่ Ulysses16, Ulysses22, Dantzig42, Att48, Eil51, Berlin52, ST70, Eil76 และ Rat99 พร้อมกับการเปรียบเทียบกับวิธีการอื่นๆอีก 10 อัลกอริทึมประกอบด้วย DEO , GWO, SA, GA, PSO, ACO, TLBO, FOA,FA และ WOA

5.1 การพิจารณาหาตัวแปรที่เหมาะสมในการตั้งค่าพารามิเตอร์

ในการพิจารณาวิธีการ FOA นั้นเมื่อมีการเปลี่ยนแปลงรูปแบบสามารถพิจารณาถึง การปรับปรุงในวิธีอัลกอริทึมแมลงวัน โดยสามารถแบ่งพิจารณารูปแบบของสมการออกได้เป็น 4 รูปแบบได้แก่

- 1) วิธีการ FOA แบบเดิม
- 2) วิธีการ FOA2TSP แบบ 2 มิติ ในพิกัด X และ Y (The Fruit Fly Optimization Algorithm with two-dimensional coordinates (2D) operates in a space defined by X and Y coordinates)
- 3) วิธีการ FOA3TSP แบบ 3 มิติ ในพิกัด X, Y และ Z (The Fruit Fly Optimization Algorithm with three-dimensional coordinates (3D) operates in a space defined by X, Y and Z coordinates)
- 4) วิธีการ FOA4TSP แบบมีโดเมนชั้นเป็นแบบ Hyper Dimensional (The Fruit Fly Optimization Algorithm with Hyper Dimensional)

สมการที่ใช้ในการประเมินค่าฟิตเนส (fitness) ที่มาจากการวิเคราะห์พิจารณา โดยกำหนดให้สมการมีหลายรูปแบบ ได้แก่ 2 มิติ, 3 มิติ และ Hyper Dimensional ดังแสดงในสมการที่ 13-24

สมการสำหรับโดเมนชั้นที่เป็น 2 มิติ (2D)

$$D(i,:) = (X(i,:)^2 + Y(i,:)^2)^{0.5} \quad (13)$$

$$D(i,:) = X(i,:) \times Y(i,:)^2 \quad (14)$$

$$D(i,:) = (X(i,:)^{0.5}) \times (Y(i,:)^2) \quad (15)$$

$$D(i,:) = (X(i,:)^{0.5}) + (Y(i,:)^{0.5}) \quad (16)$$

สมการสำหรับโดเมนชั้นที่เป็น 3 มิติ (3D)

$$D(i,:) = (X(i,:)^2 + Y(i,:)^2 + Z(i,:)^2)^{0.5} \quad (17)$$

$$D(i,:) = (X(i,:) \times Y(i,:) \times Z(i,:)) \quad (18)$$

$$D(i,:) = (X(i,:)^{0.5} + Y(i,:)^{0.5} + Z(i,:)^{0.5}) \quad (19)$$

$$D(i,:) = (X(i,:) \times (Y(i,:)^2) \times (Z(i,:)^{0.5})) \quad (20)$$

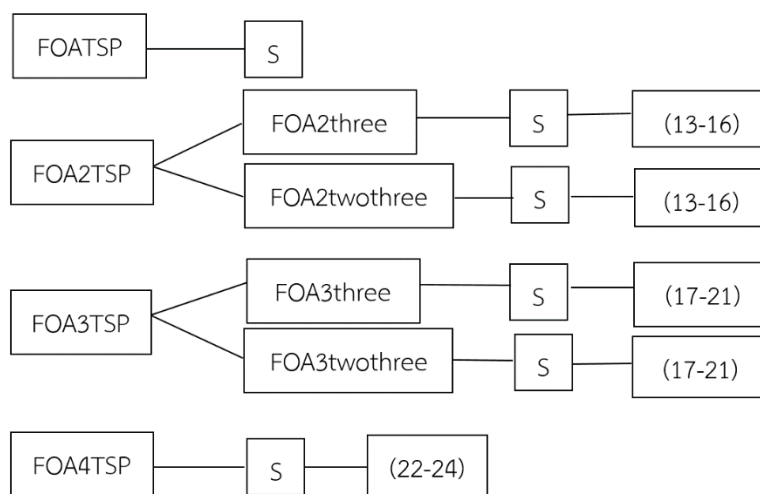
$$D(i,:) = (X(i,:) \times Y(i,:) \times (Z(i,:)^3)) \quad (21)$$

สมการสำหรับโดเมนชั้นที่เป็น Hyper Dimensional

$$D(i,:) = (X1(i,:)^2 + X2(i,:)^2 + X3(i,:)^2 + X4(i,:)^2)^{0.5} \quad (22)$$

$$D(i,:) = (X1(i,:)^{0.5} \times X2(i,:)^{0.5} \times X3(i,:)^{0.5} \times X4(i,:)^{0.5}) \quad (23)$$

$$D(I,:) = (X1(i,:) \times X2(i,:)^{0.5} \times X3(i,:)^2 \times X4(i,:)^3) \quad (24)$$



ภาพที่ 1 การกำหนดชื่อของอัลกอริทึม และรูปแบบการเลือกใช้สมการ

การกำหนดชื่อของอัลกอริทึม และรูปแบบการเลือกใช้สมการสามารถแสดงได้ดังรูปที่ 1 สามารถอธิบายได้ดังนี้

FOATSP คืออัลกอริทึม Fruit Fly Optimization Algorithm สำหรับแก้ปัญหา TSP

FOA2TS, FOA3TSP และ FOA4TSP คืออัลกอริทึม FOA แบบมีโดเมนชั้น 2 มิติ, 3 มิติ และ มิติที่สูงขึ้น ตามลำดับ โดยมีการปรับเปลี่ยน Swop, Swap และ Insertion position แบบการสลับ 2-opt

FOA2three และ FOA3three คืออัลกอริทึม FOA แบบมีโดเมนชั้น 2 มิติ และ 3 มิติ โดยมีการปรับเปลี่ยน Swop, Swap และ Insertion position แบบการสลับ 3-opt

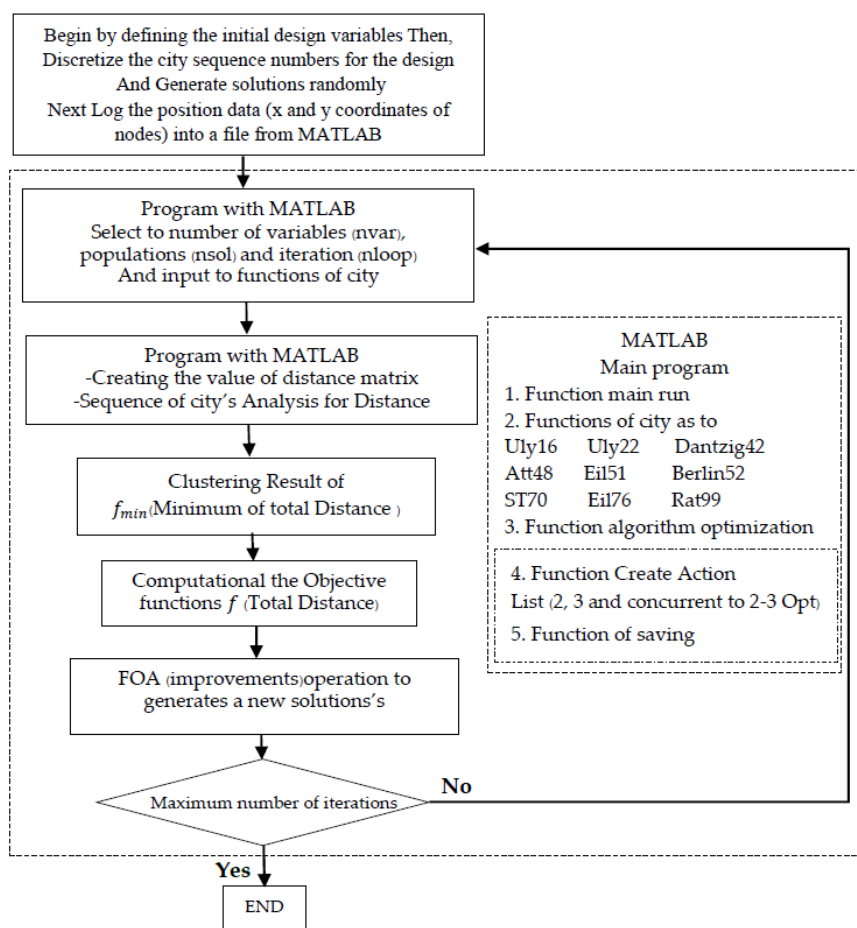
FOA2twothree และ FOA3twothree คือ อัลกอริทึม FOA แบบมีโดเมนชั้น 2 มิติ และ 3 มิติ โดยมีการปรับเปลี่ยน Swop, Swap และ Insertion position แบบการสลับ 2-3opt

S คือ การใช้ saving ในการกำหนดค่าประชากรเริ่มต้น

ตัวเลข 13-24 คือ ตัวเลขของสมการที่ใช้ในการประเมินค่าฟิตเนส (fitness) โดยมี กระบวนการในการหาค่าที่เหมาะสมของ FOA ในการหาผลเฉลยของตัวแบบปัญหาดังแสดงใน รูปที่ 2 การกำหนดค่าตัวแปรเริ่มต้นได้แก่

$n=nsol$ คือ Population size (number of fruit fly) จำนวนประชากรแมลงวันผลไม้ ได้แก่ จำนวน 40 ตัว, 100 ตัว และ 200 ตัว ในการทดลองหลายค่าประชากร (40, 100, 200) ช่วยให้สามารถวิเคราะห์ได้ว่า ขนาดประชากรเท่าใดเหมาะสมที่สุด กับลักษณะของปัญหา TSP ที่กำลังศึกษา โดยงานวิจัยที่ใช้ FOA (Pan, W. T., 2012; Wang, L., et al., 2019) และอัลกอริทึมเชิงวิวัฒนาการอื่น ๆ (เช่น PSO, GA) นิยมทดสอบหลายค่าของประชากร เช่น 20, 50, 100, 200 เพื่อหาค่า trade-off ที่เหมาะสมระหว่างคุณภาพคำตอบกับ เวลาในการประมวลผล

$nloop$ คือ Maximum number of iterations จำนวนรอบในการวนซ้ำสูงสุด ได้แก่ จำนวน 500 รอบ, 200 รอบ และ 100 รอบ



ภาพที่ 2 กระบวนการในการหาค่าที่เหมาะสมของ FOA ในการหาผลเฉลยของตัวแบบปัญหา

การเปรียบเทียบผลเฉลยที่ได้กับเวลาที่ทำการวิเคราะห์ ซึ่งจะดำเนินการในการวิเคราะห์ขนาดประชากร (Population Size) และจำนวนครั้งของการทำซ้ำ (Iterations) ที่จำนวนผลในการประเมินอยู่ที่ 20,000 ครั้ง โดยแบ่งเป็น โดยผลที่ได้จากการทดสอบกับ Ulysse16 จากการทำซ้ำ 25 รอบ มีจำนวนการสุ่ม

ในการหาคำตอบในการพิจารณาค่าเปอร์เซ็นต์ของการสลับตำแหน่ง ในรูปแบบของการปรับเปลี่ยนแบบ Swap ,Swop และ Insertion position แต่ละการปรับเปลี่ยนจะพิจารณา 2-Opt, 3-Opt และ 2-3 Opt แบ่งเป็น 10% -1%, 20%- 2% ดังแสดงผลการทดสอบในตารางที่ 3, 30%- 3%, 40%- 4% และ 50%- 5% เมื่อทำการทดสอบแล้วจึงเลือกขนาดประชากร และจำนวนครั้งของการทำซ้ำ พร้อมกับเลือกค่าเปอร์เซ็นต์ของการสลับตำแหน่งที่มีคำตอบที่ดีที่สุดและใช้เวลาเฉลี่ยนน้อยที่สุด

ผลการทดสอบขนาดประชากร และจำนวนครั้งของการทำซ้ำ ซึ่งกำหนด Population Size และ Iterations โดยวิเคราะห์จำนวนผลในการประเมินอยู่ที่ 20,000 ครั้ง ดังแสดงในตารางที่ 4 โดยผลที่ได้จากการทดสอบกับเมือง Ulysses16 จากการทำซ้ำ 25 รอบ ซึ่งผลที่ได้คือ จำนวน Population Size และ Iterations พบว่าขนาดประชากร และจำนวนครั้งของการทำซ้ำที่ 50, 400 ในอัลกอริทึม FOA2TSP มีเวลาในการหาคำตอบน้อยที่สุดที่ 62.852 มีจำนวนการสุ่มในการหาคำตอบที่ 20% ดังนั้นจึงเลือกขนาดประชากรและจำนวนครั้งของการทำซ้ำที่ 50, 400 และจำนวนการสุ่มในการหาคำตอบที่ 20% กับ 2% ในการหาคำตอบต่อไป

ตารางที่ 3 การทดสอบขนาดประชากร (Population Size) และจำนวนครั้งของการทำซ้ำ (Iterations)

เมือง Ulysse16 ตัวอย่างที่มีจำนวนการสุ่มในการหาคำตอบ 20% กับ 2%

Algorithm	Nsol	Nloop (รอบ)	Nrep (ครั้ง)	F _{min}	F _{max}	Average	FSTD	Time (s/round)
FOA2TSP	100	200	25	73.987*	87.106	78.211	4.575	74.726
FOA2three	100	200	25	75.023	109.800	86.758	10.854	56.750
FOA2twothree	100	200	25	73.987*	88.405	77.584	4.339	157.60
FOA3TSP	100	200	25	74.184	91.375	81.345	5.547	65.369
FOA3three	100	200	25	75.912	109.480	89.381	9.725	56.318
FOA3twothree	100	200	25	73.987*	90.531	79.906	5.187	159.527
FOA4TSP	100	200	25	73.999	92.815	80.639	5.493	161.104
FOA2TSP	200	100	25	73.987*	94.475	81.249	6.358	104.135
FOA2three	200	100	25	74.732	109.154	91.569	11.667	81.982
FOA2twothree	200	100	25	73.987*	90.096	78.762	5.060	155.531
FOA3TSP	200	100	25	74.614	91.611	79.859	5.874	63.161
FOA3three	200	100	25	77.702	108.550	89.515	8.931	52.402
FOA3twothree	200	100	25	74.153	89.047	78.334	3.558	155.483
FOA4TSP	200	100	25	73.987*	89.602	78.061	5.079	155.541
FOA2TSP	50	400	25	73.987*	93.084	79.734	5.387	62.852
FOA2three	50	400	25	75.089	103.890	86.460	10.821	56.111
FOA2twothree	50	400	25	74.001	89.635	78.651	4.711	156.927
FOA3TSP	50	400	25	74.184	95.147	81.822	5.647	62.218
FOA3three	50	400	25	75.038	105.671	88.836	9.828	52.548
FOA3twothree	50	400	25	74.739	94.342	80.573	5.618	158.104
FOA4TSP	50	400	25	73.987*	90.903	77.079	4.372	156.063

*Optimal solution ของ Ulysses16

ตารางที่ 4 การเปรียบเทียบค่า Fitness โดยการเปรียบเทียบขนาดประชากร (Population Size) และ จำนวนครั้งของการทำซ้ำ (Iterations)

Population Size	Iterations	number of evaluations	Time(s)	F _{min}
100	200	20,000	114.681	73.987*
200	100	20,000	113.983	73.987*
50	400	20,000	117.769	73.987*
50	400	20,000	114.349	73.987*
100	200	20,000	74.726	73.987*
200	100	20,000	104.135	73.987*
50	400	20,000	62.852	73.987*
100	200	20,000	87.165	73.987*
50	400	20,000	87.838	73.987*
100	200	20,000	110.117	73.987*

*Optimal solution ของ Ulysses16

5.2 การพิจารณาที่เกิดจากการเรียนรู้ด้วยตัวเอง

การพิจารณาที่เกิดจากการเรียนรู้ด้วยตัวเอง ได้แก่ การปรับเปลี่ยน (Swap) ได้แก่ การสลับ 2-Opt และการสลับ 3-Opt การสลับตำแหน่ง (Swop) ได้แก่ การสลับ 2-Opt และการสลับ 3-Opt และการแทรกเวกเตอร์ในข้อมูลใหม่ ได้แก่ การสลับ 2-Opt และการสลับ 3-Opt ด้วยเทคนิคของอัลกอริทึมแมลงวันในการพิจารณาค่าเปอร์เซ็นต์ของการสลับตำแหน่ง ในรูปแบบของการปรับเปลี่ยนแบบ Swap, Swop และ Insertion position แต่ละการปรับเปลี่ยนจะพิจารณา การสลับ 2-Opt การสลับ 3-Opt และการสลับ 2-3 Opt โดยในเมือง Ulysses16, Ulysses22, Dantzig42 และ Att48 จะเลือกใช้เปอร์เซ็นต์ในการสลับ 2-Opt อยู่ที่ 20% การสลับ 3-Opt อยู่ที่ 2% และในเมือง Eil51 และ Berlin52 จะเลือกใช้เปอร์เซ็นต์ในการสลับ 2-Opt อยู่ที่ 10% การสลับ 3-Opt อยู่ที่ 1% และในเมือง ST70, Eil76 และ Rat99 จะเลือกใช้เปอร์เซ็นต์ในการสลับ 2-Opt อยู่ที่ 10% และการสลับ 3-Opt อยู่ที่ 0.4% โดยเลือกค่าที่ได้จากการสุ่มจากผลของการสลับที่ได้เนื่องจากในแต่ละเมืองใช้เวลาในการประมวลผลที่แตกต่างกันออกไป ในเมืองที่มากขึ้นจะใช้เวลามากขึ้นตามจึงจะเป็นที่จะต้องลดเปอร์เซ็นต์ให้เหมาะสม และทั้งนี้ยังขึ้นอยู่กับหน่วยการประมวลผลกลาง (CPU) และขนาดของหน่วยความจำ (RAM) ของคอมพิวเตอร์ที่ใช้ประมวลผลด้วย

รวมถึงการเปรียบเทียบอัลกอริทึมที่ทำการปรับปรุงแล้ว กับอัลกอริทึมที่มีการปรับปรุงแล้ว และเพิ่มการนำวิธีการ Saving เข้ามาช่วยโดยการกำหนดค่าเริ่มต้น เมื่อเปรียบเทียบแล้วจึงทำการเลือกอัลกอริทึมที่มีค่าคำตอบเข้าใกล้ Optimal solution มากที่สุดและใช้เวลาน้อย เนื่องจากอัลกอริทึมที่ทำการปรับปรุงมีจำนวนมากจากการปรับโดเมนชั้นให้เป็น 3 มิติ และมิติที่สูงขึ้น การปรับเปลี่ยนแบบ Swop, Swap และ

Insertion position การสลับ 2-Opt การสลับ 3-Opt และการสลับ 2-3 Opt การเลือกใช้สมการในการหาค่าฟิตเนส และการนำ Saving เข้ามาช่วยโดยการกำหนดค่าเริ่มต้น

จากการทดสอบในเมือง Ulysses16, Ulysses22, DANTZIG42, และ Att48 ทั้งแบบที่มีการปรับปรุงแล้วและแบบการนำ Saving มาช่วยหาคำตอบ เพื่อใช้เปรียบเทียบพบว่า การนำ Saving เข้ามาช่วยทำให้ได้คำตอบที่ดีกว่าแบบปรับปรุงแล้ว จึงเลือกเมืองที่มีค่าน้อยของทั้งแบบที่มีการปรับปรุงแล้วจำนวน 2 อัลกอริทึมคือ FOA3twothree19 และ FOA4TSP23 และแบบที่มีการปรับปรุงแล้วและนำ Saving มาช่วยหาคำตอบจำนวน 2 อัลกอริทึมคือ FOA2twothree14S และ FOA3twothree17S เป็นอัลกอริทึมที่มีค่าน้อยที่สุดที่ 34,076.871 และใช้เวลาน้อย เพื่อใช้ในการหาคำตอบในเมืองอื่นๆต่อไป พร้อมกับเปรียบเทียบกับ FOATSPS ที่ยังไม่มีปรับปรุงการหาคำตอบ แต่มีการนำ Saving เข้ามาช่วย

5.3 ผลการพิจารณาตัวแบบปัญหา

จากการพิจารณาที่เกิดจากการเรียนรู้ด้วยตัวเองได้ทำการเลือกอัลกอริทึมจำนวน 5 อัลกอริทึมคือ FOATSPS, FOA2twothree14S, FOA3twothree17S, FOA3twothree19 และ FOA4TSP23 ที่จะใช้เป็นการเปรียบเทียบกับตัวแบบมาตรฐานจำนวน 10 ตัวแบบได้แก่ FOA, DEO, GA, TLBO, GWO, PSO, FOA, SA, FA และ WOA เพื่อประเมินประสิทธิภาพ การเปรียบเทียบกับอัลกอริทึมอื่นๆช่วยให้เห็นว่าอัลกอริทึมที่ทำการปรับปรุงนั้นมีประสิทธิภาพดีขึ้นหรือไม่และการเปรียบเทียบกับอัลกอริทึมอื่นๆ ที่เป็นที่ยอมรับเป็นการยืนยันว่าอัลกอริทึมที่ทำการปรับปรุงมีความได้เปรียบทางด้านใดบ้าง จากผลการทดลองพบว่าในเมือง Ulysses16 วิธีที่มีการปรับปรุงและใช้ Saving ให้ผลลัพธ์ใกล้เคียงกับค่าที่ดีที่สุด และวิธีที่ปรับปรุงแต่ไม่ใช้ Saving ก็ยังได้คำตอบที่ใกล้เคียงเช่นกัน ดังแสดงในตารางที่ 5 และสามารถแสดง Box plots ของอัลกอริทึม FOA ที่ทำการปรับปรุงกับวิธีการอื่น 10 วิธีการ ของเมือง Ulysses16 ได้ดังรูปที่ 3 สำหรับ Ulysses22 วิธีที่ใช้ Saving ยังคงให้ผลลัพธ์ใกล้เคียง ในขณะที่วิธีไม่ใช้ Saving ได้ค่ามากกว่าส่วน dantzig42 แบบที่ใช้ Saving ให้ค่าต่ำกว่า ขณะที่แบบไม่ใช้ Saving ให้ค่ามากกว่า และในเมือง att48, Eil51, Berlin52, ST70, Eil76 และ Rat99 วิธีที่มีการใช้ Saving ให้คำตอบที่ใกล้เคียงมากกว่าอย่างต่อเนื่องในขณะที่วิธีไม่ใช้ Saving มักได้ค่ามากกว่าจะเห็นได้ว่าหากไม่มีการใช้ Saving เมื่อจำนวนเมืองเพิ่มขึ้นคำตอบที่ได้จะเบี่ยงเบนไปมากอันเป็นผลจากความซับซ้อนที่สูงขึ้นและข้อจำกัดของอัลกอริทึม แม้จะมีการปรับปรุงวิธีก็ตามดังนั้นการใช้ Saving จึงมีบทบาทสำคัญในการช่วยให้อัลกอริทึมให้ผลลัพธ์ที่มีคุณภาพและใกล้เคียงค่าที่ถูกต้องมากยิ่งขึ้น

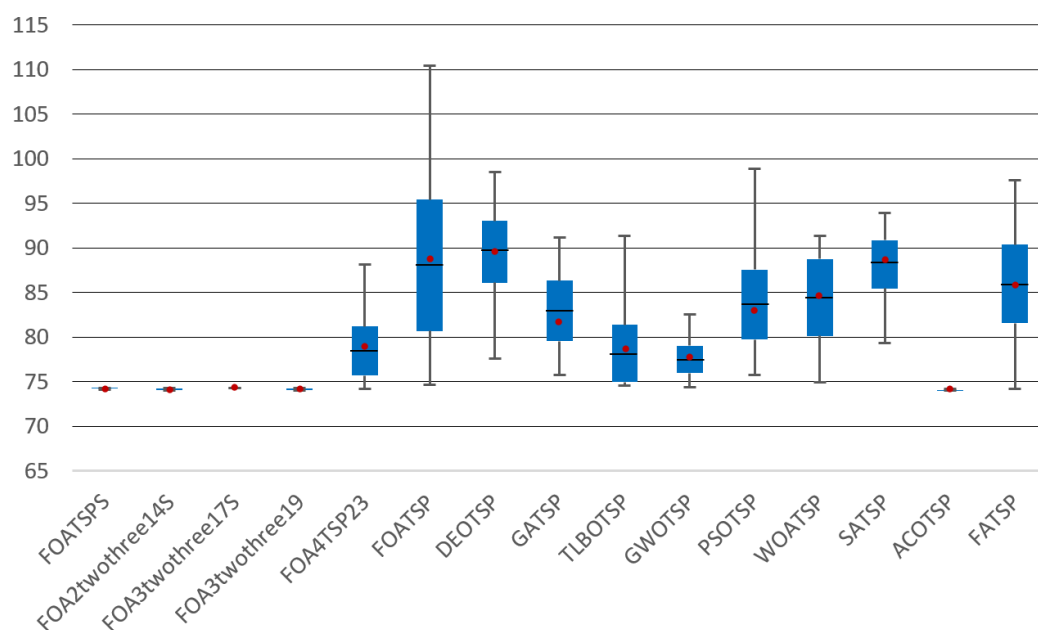
ผลสรุปจากการจัดอันดับที่ได้จากการทดสอบโดยพิจารณาจากค่าคำตอบที่ได้เทียบกับ ค่าคำตอบที่เหมาะสมที่สุด และเวลาในการประมวลผลเมื่อทำการเปรียบเทียบกันพบว่าอัลกอริทึม FOA2twothree14S และ FOA3twothree17S ได้อันดับที่ 1 ดังแสดงในตารางที่ 6 โดยอัลกอริทึม FOA2twothree14S มีโดเมนชั้น เป็นสองมิติ ในพิกัด X และ Y ที่ทำการสลับ 2-3 Opt ด้วยสมการ $D(i,:) = X(i,:) \times Y(i,:)^2$ เป็นสมการที่

เกิดจาก X คูณกับ Y^2 และมีการนำ Saving เข้ามาช่วย และอัลกอริทึม FOA3twothree17S ที่มีโดเมนชั้น 3D ในพิกัด

ตารางที่ 5 การทดสอบการปรับปรุงอัลกอริทึม FOA ของเมือง Ulysses16 มีจำนวนการสุ่มในการหาคำตอบ 20% กับ 2% พร้อมกับการเปรียบเทียบกับวิธีการอื่น 10 วิธีการ

Algorithm	Nsol	Nloop (รอบ)	Nrep (ครั้ง)	F_{min}	F_{max}	Average	FSTD
FOATSPS	50	400	25	74.108	74.307	74.291	0.055
FOA2twothree14S	50	400	25	73.999	74.307	74.147	0.096
FOA3twothree17S	50	400	25	74.307	74.307	74.307	0
FOA3twothree19	50	400	25	73.987*	74.307	74.210	2.900
FOA4TSP23	50	400	25	74.198	88.149	78.462	4.073
FOATSP	50	400	25	74.702	110.434	88.065	10.924
DEOTSP	50	400	25	77.643	98.504	89.557	5.131
GATSP	50	400	25	75.725	91.160	82.938	5.042
TLBOTSP	50	400	25	74.599	91.350	78.157	4.759
GWOTSP	50	400	25	74.356	82.515	77.498	2.270
PSOTSP	50	400	25	75.726	98.893	83.632	5.816
WOATSP	50	400	25	74.922	91.350	84.468	6.382
SATSP	50	400	25	79.320	93.964	88.183	4.023
ACOTSP	50	400	25	73.987*	74.247	74.044	0.074
FATSP	50	400	25	74.233	97.636	85.978	6.539

*Optimal solution ของ Ulysses16



ภาพที่ 3 Box plots ของอัลกอริทึม FOA ที่ทำการปรับปรุงกับวิธีการอื่น 10 วิธีการ ของเมือง Ulysses16

X, Y และ Z ทำการสลับ 2-3 Opt ด้วยสมการ $D(i,:) = (X(i,:)^2 + Y(i,:)^2 + Z(i,:)^2)^{0.5}$ เป็นสมการเกิดจากรากที่สองของ $X^2 + Y^2 + Z^2$ และมีการนำ Saving เข้ามาช่วยซึ่งเกณฑ์ในการจัดอันดับมาจากการนำค่า Fmin มาจัดลำดับ จากนั้นนำลำดับที่ได้มาหาค่าเฉลี่ยแล้วจัดอันดับอีกครั้ง อันดับที่ 2 คือ FOATSPS คืออัลกอริทึม FOA เดิมที่ไม่มีการปรับปรุงอัลกอริทึม การหาค่าฟิตเนสในรูปแบบสมการเดิมของอัลกอริทึมคือ $D(i,:) = (X(i,:)^2 + Y(i,:)^2)^{0.5}$ เป็นสมการเกิดจากรากที่สองของ X^2 บวกกับ Y^2 และมีการนำ Saving เข้ามาช่วย อันดับที่ 3 คืออัลกอริทึม ACO อันดับที่ 4 คืออัลกอริทึม TLBO อันดับที่ 5 คืออัลกอริทึม GWO และอัลกอริทึม WOA อันดับที่ 6 คืออัลกอริทึม GA อันดับที่ 7 คือ FOA4TSP23 มีโดเมนชั้นเป็นแบบ Hyper Dimensional ทำการสลับ 2-3 Opt อันดับที่ 8 คือ FOA3twothree19 มีโดเมนชั้นเป็น 3 มิติ ในพิกัด X,Y และ Z มีทำการสลับ 2-3 Opt อันดับที่ 9 คืออัลกอริทึม FOA ที่ไม่มีการปรับปรุงอัลกอริทึม อันดับที่ 10 คืออัลกอริทึม PSO อันดับที่ 11 คืออัลกอริทึม SA อันดับที่ 12 คืออัลกอริทึม DEO และอันดับที่ 13 คืออัลกอริทึม FA

ในกรณีของ FOA2twothree14S ทำให้ฟังก์ชันฟิตเนสของปัญหา TSP มีความไวต่อการเปลี่ยนแปลงของพิกัด Y มากกว่า X ซึ่งช่วยเพิ่มความแตกต่างของค่าฟิตเนสระหว่างประชากร ส่งผลให้สามารถแยกเส้นทางที่ได้ออกจากเส้นทางที่ด้อยคุณภาพได้ชัดเจนขึ้น ส่วน FOA3twothree17S เพิ่มมิติที่สามเข้ามาทำให้พื้นที่ค้นหามีความซับซ้อนและสมจริงขึ้นโดยเฉพาะในปัญหาที่มีลักษณะเชิงพื้นที่ 3D การใช้รากที่สองของระยะทางสามมิติทำให้การประเมินฟิตเนสใกล้เคียงลักษณะจริงมากกว่าแบบ 2D ปกติ จากผลการแก้ปัญหา TSP ตอบสนองต่อพารามิเตอร์มากกว่า 1 มิติ และมี non-linear scaling (เช่น ยกกำลังสอง, รากที่สอง) ช่วยให้การประเมินเส้นทางละเอียดและแม่นยำกว่ารวมถึงการใช้ 2-3 Opt ช่วยเพิ่มการปรับเส้นทางในเชิงลึกลดความเสี่ยงของการติด local minima และการใช้ Savings Algorithm ช่วยสร้างประชากรเริ่มต้นคุณภาพสูง ทำให้อัลกอริทึม FOA ที่ปรับปรุงแล้วมีศักยภาพเหนือกว่า FOA ดั้งเดิมและอัลกอริทึมเมต้าฮิวริสติกตัวอื่นๆ

ตารางที่ 6 ผลสรุปการจัดอันดับ (Rank) ที่ได้จากการทดสอบปัญหาโดยใช้วิธีเมต้าฮิวริสติก

City	Ulysses1	Ulysses22	Dantzig42	att48	Eil51	Berlin52	st70	Eil76	rat99	Mean	Rank
FOATSPS	4	2	3	3	1	3	3	3	1	2.556	2
FOA2twothree14S	3	4	2	2	2	1	2	2	3	2.333	1
FOA3twothree17S	7	3	1	1	3	2	1	1	2	2.333	1
FOA3twothree19	2	11	11	10	11	11	12	11	12	10.111	8
FOA4TSP23	5	7	13	9	9	9	10	9	11	9.111	7
FOATSP	10	10	12	11	10	10	9	12	13	10.778	9
DEOTSP	14	15	14	14	14	14	14	14	14	14.111	12
GATSP	12	9	9	7	7	5	6	6	9	7.778	6
TLBOTSP	9	5	6	5	5	6	5	5	5	5.667	4
GWOTSP	8	6	5	6	8	8	8	8	7	7.111	5
PSOTSP	13	12	8	12	13	12	11	10	8	11.000	10
WOATSP	11	8	4	8	6	7	7	7	6	7.111	5
SATSP	15	13	10	13	12	13	13	13	10	12.444	11
ACOTSP	1	1	7	4	4	4	4	4	4	3.667	3
FATSP	6	14	15	15	15	15	15	15	15	13.889	13

6. สรุปและอภิปรายผล

วิธีการแมลงวันผลไม้สำหรับปัญหา TSP ที่ใช้ในการแก้ปัญหาในการจัดเส้นทางเดิมนั้นมีค่าพารามิเตอร์แบบโดเมนชั้นเป็นสองมิติ ตำแหน่งของแมลงวันผลไม้แต่ละตัวถูกกำหนดโดยตัวแปรสองตัว โดยปกติจะเป็นพิกัด X และ Y แต่การใช้เพียง 2 มิติ ไม่เพียงพอสำหรับการแก้ปัญหาที่มีความซับซ้อนมากขึ้น เช่น ปัญหาที่มีหลายตัวแปร และซับซ้อน หรือปัจจัยที่ต้องพิจารณามากกว่าแค่การค้นหาพื้นที่ในรูปแบบ X และ Y อาจทำให้การคำนวณหาคำตอบไม่มีประสิทธิภาพเท่าที่ควร โดยเฉพาะอย่างยิ่งในปัญหาที่มีตัวแปรหลายตัวแปรที่เชื่อมโยงกัน ผู้วิจัยจึงได้ทำการปรับค่าพารามิเตอร์โดยการเพิ่มโดเมนชั้นให้เป็น 3D แบบ 3 มิติ จะเพิ่มพิกัด Z เข้ามา ทำให้พื้นที่การค้นหาซับซ้อนยิ่งขึ้น และในโดเมนชั้นที่สูงขึ้น การปรับเปลี่ยนแบบ Swop ,Swap และ Insertion position แต่ละการปรับเปลี่ยนจะพิจารณาการสลับ 2-Opt การสลับ 3-Opt และการสลับ 2-3 Opt และการนำ Saving เข้ามาช่วยโดยการกำหนดค่าเริ่มต้น ที่ทำให้เข้าใกล้ค่าคำตอบที่เหมาะสมที่สุดซึ่ง Saving จะสามารถช่วยลดระยะเวลาในการหาคำตอบลงได้

ในการวิเคราะห์การเปรียบเทียบ Algorithm ที่ดีที่สุดกับค่าคำตอบที่เหมาะสมที่สุดโดยพิจารณาจาก 9 เมือง ดังแสดงในตารางที่ 7 ที่ทำการทดสอบผลที่ได้จากการพิจารณาค่าเฉลี่ยจากการทดสอบเมือง Dantzig42 มีคำตอบลดลงจากค่าคำตอบที่เหมาะสมที่สุดคิดเป็น 2.780 % เนื่องด้วยในการทดลองใช้อัลกอริทึม FOA กับชุดข้อมูล DANTZIG42 ซึ่งมีการรายงานค่า optimal tour length เท่ากับ 699 (Reinelt, 1991; Reinelt, 1995) ผลการคำนวณของงานวิจัยนี้ได้ค่าเส้นทางที่สั้นกว่า คือ 679.56 ทั้งนี้สาเหตุสำคัญเกิดจากความแตกต่างในการคำนวณระยะทาง โดยในงานวิจัยนี้ใช้ค่า Euclidean distance แบบทศนิยมเต็มโดยไม่ปิดเศษ ขณะที่ TSPLIB กำหนดให้ใช้การปัดค่าระยะทางเป็นจำนวนเต็ม ดังนั้น แม้ผลลัพธ์ที่ได้จะดูต่ำกว่าค่าคำตอบที่เหมาะสมที่สุดที่รายงาน แต่ไม่สามารถถือว่าเป็นคำตอบที่ดีกว่า ทั้งนี้ ผลการทดลองยังสะท้อนว่าอัลกอริทึมที่พัฒนาขึ้นสามารถหาคำตอบที่ใกล้เคียงกับค่าที่ได้รับการยอมรับในวงวิชาการ

ในเมือง Ulysses16 มีคำตอบเท่ากับค่าคำตอบที่เหมาะสมที่สุดในเมือง Ulysses22, att48 และ Eil51 มีคำตอบใกล้เคียงค่าคำตอบที่เหมาะสมที่สุดคิดเป็น 0.239%, 1.655%, และ 2.691% ตามลำดับ และเมือง Berlin52, ST70, Eil76 และ Rat99 มีคำตอบมากกว่า Optimal solution คิดเป็น 3.437%, 3.659% ,4.617% และ 4.160% ตามลำดับ

ตารางที่ 7 การเปรียบเทียบ Algorithm ที่ดีที่สุด กับค่าคำตอบที่เหมาะสมที่สุดโดยวิธีการแมลงวันผลไม้

City	Optimal solution	Fitness	Optimal solution (%)	Algorithm
Ulysses16	73.98	73.98*	ได้ Optimal solution	FOA4TSP24S
Ulysses22	75.30	75.48	0.239	FOA4TSP22S
Dantzig42	699	679.56**	ลดลง -2.780	FOA3twothree17S
att48	33,522	34,076.87	1.655	FOA3twothree17S
eil51	426	437.46	2.691	FOATSPS
berlin52	7,542	7,801.23	3.437	FOA3twothree14S

ตารางที่ 7 (ต่อ)

City	Optimal solution	Fitness	Optimal solution (%)	Algorithm
st70	675	699.70	3.659	FOA3twothree17S
Eil76	538	562.843	4.617	FOA3twothree17S
rat99	1,211	1,261.38	4.160	FOATSPS

* Optimal solution, **≈ Optimal tour length(Reinelt, 1991; Reinelt, 1995)

เมื่อพิจารณาการปรับปรุงวิธีการแมลงวันผลไม้ในการหาคำตอบของปัญหา TSP จากการทดสอบตัวแบบเมืองทั้ง 9 ตัวแบบนี้ผลลัพธ์ที่ได้จาก FOA3twothree17S สามารถได้คำตอบที่ดีที่สุดจำนวน 5 ตัวแบบ ลงลงมาได้แก่ FOA4TSP24S จำนวน 1 ตัวแบบ ,FOA4TSP22S 1 ตัว และ FOATSPS 2 ตัวแบบ ซึ่งจะเห็นได้มาเมื่อมีการใช้มิติที่สูงขึ้นจากเดิม (X, Y) เป็น (X, Y, Z) และ (X1, X2, X3, X4) ทำให้รูปแบบการหาคำตอบในจำนวนประชากร และรอบที่เท่ากันเมื่อเทียบกับแบบเดิมนั้นมีความสามารถในการปรับเปลี่ยนค่าของตัวแปรได้ โดยสามารถเข้าใกล้ค่าคำตอบที่เหมาะสมที่สุดโดยเมืองที่ได้ค่าน้อยกว่าค่าคำตอบที่เหมาะสมที่สุดได้แก่เมือง Dantzig42 และเมืองที่ได้ค่าเท่ากับค่าคำตอบที่เหมาะสมที่สุดได้แก่เมือง Ulysses16 และ เมืองที่ได้ค่าใกล้เคียง ค่าคำตอบที่เหมาะสมที่สุดได้แก่เมือง Ulysses22 ,Att48, Eil51, Berlin52, ST70,Eil76 และ Rat99 ซึ่งผลการพัฒนา FOA ไปสู่มิติที่สูงขึ้น (3D และ Hyper Dimensional) แสดงให้เห็นถึงศักยภาพในการสำรวจพื้นที่คำตอบที่กว้างขึ้นซึ่งช่วยลดโอกาสในการติดอยู่ที่ local optima ได้อย่างมีนัยสำคัญอธิบายได้จากทฤษฎีการเพิ่ม degrees of freedom ของตัวแปรที่ทำให้เส้นทางการเคลื่อนที่ (trajectory) ของอัลกอริทึมไม่ถูกจำกัดอยู่ในระนาบสองมิติเดิมในขณะเดียวกันการผนวก Heuristic Methods ได้แก่ 2-Opt, 3-Opt, 2-3Opt และ Savings Algorithm ทำให้อัลกอริทึมสามารถสร้างประชากรเริ่มต้นที่มีคุณภาพและมีการปรับแต่งเชิงโครงสร้างของเส้นทางได้ดียิ่งขึ้น ส่งผลให้กระบวนการ local search มีประสิทธิภาพสูงขึ้นกว่าการใช้ FOA แบบเดิมโดยการผสานวิธีเมต้าฮิวริสติก กับความรู้เชิงฮิวริสติกที่เฉพาะเจาะจงกับปัญหา (heuristics problem-specific knowledge) เช่น 2-Opt, 3-Opt, และ Savings Algorithm ช่วยเพิ่มประสิทธิภาพของการหาคำตอบอย่างมีนัยสำคัญ (Liu et al.,2023; Zhang et al.,2023) โดยสามารถสร้างประชากรเริ่มต้นที่มีคุณภาพสูงและปรับปรุง local search ได้ดีซึ่งสอดคล้องกับการแก้ปัญหาเชิงผสมผสาน (Combinatorial Optimization) สำหรับการนำใช้กับปัญหา TSP ที่จะต้องสร้างสมดุลระหว่าง exploration (การสำรวจพื้นที่คำตอบใหม่) และ exploitation (การปรับแต่งคำตอบที่ติดอยู่แล้ว) เพื่อหลีกเลี่ยง local optima

โดยจะเห็นได้ว่าการใช้ Algorithm FOA3twothree17S ที่เป็นสามมิติมีการปรับเปลี่ยน Swap, Swop และ Insertion position แบบการสลับ 2-3 Opt มีการพิจารณาค่าเปอร์เซ็นต์ของการสลับตำแหน่ง และการนำ Saving เข้ามาช่วยในการกำหนดค่าประชากรเริ่มต้นจะเป็นทำให้เข้าใกล้ค่าคำตอบที่เหมาะสมที่สุดเนื่องด้วย ความสามารถในการสำรวจจาก FOA แบบมิติสูง และความสามารถในการเลือกคำตอบจากวิธีฮิวริสติกจึงถือเป็นการผสานจุดแข็งของ global search และ local search เข้าด้วยกัน ดังนั้นจึงสามารถอธิบายเชิง

ทฤษฎีได้ว่า FOA3twothree17S มีประสิทธิภาพสูงสุดในการแก้ปัญหา TSP ที่ซับซ้อนเมื่อเทียบกับ FOA แบบดั้งเดิม และเมตาดาอิวิลิติกอื่นๆ และอัลกอริทึม FOA4TSP24S ที่เป็นโดเมนขั้นที่สูงขึ้น ที่มีการปรับเปลี่ยน Swap, Swop และ Insertion position แบบการสลับแบบ 2-3 Opt และการพิจารณาค่าเปอร์เซ็นต์ของการสลับตำแหน่งและมีการนำ Saving เข้ามา จะเป็นทำให้ได้ค่าใกล้เคียงกับค่าคำตอบที่เหมาะสมที่สุด และอัลกอริทึม FOA4TSP22S เป็นโดเมนขั้นที่สูงขึ้นที่มีการปรับเปลี่ยน Swap, Swop และ Insertion position แบบการสลับแบบ 2-3 Opt และการพิจารณาค่าเปอร์เซ็นต์ของการสลับตำแหน่งและมีการนำ Saving เข้ามา และอัลกอริทึม FOATSPS คืออัลกอริทึม FOA เดิมที่ไม่มีการปรับปรุงอัลกอริทึมและมีการนำ Saving เข้ามา โดยทุกอัลกอริทึมจะเลือกใช้เปอร์เซ็นต์ในการสลับ 2-Opt อยู่ที่ 10%-20% และการสลับ 3-Opt อยู่ที่ 0.4% - 2%

จากวัตถุประสงค์ของการศึกษาในการเพิ่มประสิทธิภาพของ FOA ในกระบวนการหาคำตอบโดยการเพิ่มโดเมนขั้น และมิติที่สูงขึ้นโดยการปรับปรุงการหาผลเฉลยเดิมของอัลกอริทึม FOA ที่มีอยู่ให้มีค่าใกล้เคียงกับคำตอบที่ดีที่สุดนั้นจะเห็นได้ว่าเมื่อเปรียบเทียบกับอัลกอริทึม FOA เดิม ที่ยังไม่มีปรับปรุง นั้นอัลกอริทึม FOA อยู่ในลำดับที่ 9 แต่ในลำดับที่ 1 เป็นอัลกอริทึม FOA3twothree17S และ FOA2twothree14S จึงแสดงให้เห็นถึงประสิทธิภาพในการหาคำตอบที่มีคำตอบเท่ากับค่าคำตอบที่ดีที่สุดหรือมีคำตอบใกล้เคียงกับคำตอบที่ดีที่สุด โดยเป็นไปตามวัตถุประสงค์ของการศึกษา และผลลัพธ์จากงานวิจัยนี้ช่วยผู้ประกอบการลดต้นทุนและเวลาเพิ่มประสิทธิภาพการจัดส่ง เสริมการตัดสินใจเชิงกลยุทธ์ และสนับสนุนความยั่งยืนทางธุรกิจ และสิ่งแวดล้อม ทำให้สามารถแข่งขันได้ดียิ่งขึ้นในระบบเศรษฐกิจที่ขับเคลื่อนด้วยโลจิสติกส์ งานวิจัยนี้แม้จะแสดงให้เห็นถึงประสิทธิภาพของ FOA ที่ได้พัฒนาขึ้นในการแก้ปัญหา TSP ได้ดีกว่า FOA ดั้งเดิม และวิธีเมตาดาอิวิลิติกตัวอื่นๆ แต่ก็ยังมีข้อจำกัดบางประการ ได้แก่ ความไวต่อพารามิเตอร์ที่ตั้งค่า การประเมินผลเฉพาะปัญหา TSP ไม่ครอบคลุมปัญหาประยุกต์อื่น ภาระด้านเวลาในการคำนวณที่สูงขึ้นจากการเพิ่มมิติ และการใช้วิธีอิวิลิติก และการเปรียบเทียบที่ยังไม่ครอบคลุมทั้งหมด

ในการทำวิจัยครั้งต่อไปทางผู้วิจัยจะนำอัลกอริทึม FOA ที่ได้พัฒนาขึ้นไปใช้ประโยชน์โดยแนวทางการวิจัยในอนาคตจะทำการพัฒนา FOA ที่ปรับพารามิเตอร์ได้เอง (adaptive/self-adaptive FOA) การบูรณาการ Machine Learning เพื่อเสริมการตัดสินใจ การขยายอัลกอริทึมไปสู่ปัญหาการจัดเส้นทางเดินรถ (Vehicle Routing Problem, VRP) หรือ Dynamic TSP ซึ่งจะช่วยให้ FOA ที่ได้พัฒนาขึ้นมีความแข็งแกร่ง และพร้อมใช้งานจริงในสภาพแวดล้อมที่ซับซ้อนมากยิ่งขึ้น โดยการทดสอบ และประยุกต์ใช้ FOA ในการศึกษา รูปแบบปัญหา VRP โดยมีข้อจำกัด เช่น ข้อจำกัดด้านเวลา ความจุของยานพาหนะ หรือหน้าต่างเวลาการส่งมอบ เพื่อแสดงให้เห็นถึงความสามารถที่แท้จริงของอัลกอริทึมในสถานการณ์ที่ซับซ้อนมากขึ้น

เอกสารอ้างอิง

- Almufti, S., & Shaban, A. A. (2025). Comparative analysis of metaheuristic algorithms for solving the traveling salesman problems. *International Journal of Scientific World*, 7(1), 1–5.

- Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2007). *The traveling salesman problem: A computational study*. Princeton University Press.
- Bellman, R. (1962). Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM*, 9(1), 61–63.
- Crişan, G. C., Pintea, C., & Pop, P. C. (2020). Studying heuristics adaptation to a specific degree of fuzziness. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–7). IEEE.
- Crişan, P., Pintea, S., & Pop, F. (2020). Fruit fly optimization algorithm for traveling salesperson problem. *International Journal of Computer Applications*, 107(18), 1–6.
- Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *BioSystems*, 43(2), 73–81.
- Gendreau, M., & Potvin, J. Y. (2010). *Handbook of metaheuristics*. Springer.
- Han, X., Liu, Q., Wang, H., & Wang, L. (2018). Novel fruit fly optimization algorithm with trend search and co-evolution. *Knowledge-Based Systems*, 141, 1–17.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems*. MIT Press.
- Huang, C., Li, X., & Wen, Y. (2021). An OTSU image segmentation based on fruitfly optimization algorithm. *Alexandria Engineering Journal*, 60(1), 183–188.
- Huang, L., Wang, G. C., Bai, T., & Wang, Z. (2017). An improved fruit fly optimization algorithm for solving traveling salesman problem. *Frontiers of Information Technology & Electronic Engineering*, 18(10), 1525–1533.
- Huang, Y., & Su, T. (2017). Optimizing fruit fly algorithm to solve TSP problem. In *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)* (pp. 1409–1411). IEEE.
- Iscan, H., & Gunduz, M. (2017). An application of fruit fly optimization algorithm for traveling salesman problem. *Procedia Computer Science*, 111, 58–63.
- Jiang, Z. B., & Yang, Q. (2016). A discrete fruit fly optimization algorithm for the traveling salesman problem. *PLOS ONE*, 11(11).
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 – International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). IEEE.
- Laporte, G. (2010). Fifty years of vehicle routing. *Transportation Science*, 43(4), 408–416.

- Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., & Shmoys, D. B. (1985). *The traveling salesman problem: A guided tour of combinatorial optimization*. Wiley.
- Li, H. Z., Guo, S., Li, C. J., & Sun, J. Q. (2013). A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. *Knowledge-Based Systems*, 37, 378–387.
- Li, X., Zhang, J., & Wang, Y. (2018). A modified fruit fly optimization algorithm for global optimization. *Applied Intelligence*, 48(7), 1851–1865.
- Lin, S. M. (2013). Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network. *Neural Computing and Applications*, 22, 783–791.
- Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2), 498–516.
- Liu, Y., Chen, X., & Dib, O. (2023). Application of hybrid metaheuristic algorithms including FOA and PSO to traveling salesman problem. In *Intelligent Computing and Optimization Conference* (pp. 3–18). Springer.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61.
- Mousavi, S. M., Sadeghi, J., Niaki, S. T. A., Alikar, N., Bahreininejad, A., & Metselaar, H. S. C. (2014). Two parameter-tuned meta-heuristics for a discounted inventory control problem in a fuzzy environment. *Information Sciences*, 276, 42–62.
- Pan, Q. K., Sang, H. Y., Duan, J. H., & Gao, L. (2014). An improved fruit fly optimization algorithm for continuous function optimization problems. *Knowledge-Based Systems*, 62, 69–83.
- Pan, W. T. (2012). A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowledge-Based Systems*, 26, 69–74.
- Pasandideh, S. H. R., Niaki, S. T. A., & Mousavi, S. M. (2013). Two metaheuristics to solve a multi-item multiperiod inventory control problem under storage constraint and discounts. *The International Journal of Advanced Manufacturing Technology*, 69(5), 1671–1684.
- Ranjan, R. K., & Kumar, V. (2023). A systematic review on fruit fly optimization algorithm and its applications. *Artificial Intelligence Review*, 56, 13015–13069.

- Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA Journal on Computing*, 3(4), 376–384.
- Reinelt, G. (1995). *TSPLIB95: A library of sample instances for the TSP (and related problems)*. Universität Heidelberg.
- Shan, D., Cao, G., & Dong, H. (2013). LGMS-FOA: An improved fruit fly optimization algorithm for solving optimization problems. *Mathematical Problems in Engineering*, 2013(1), 108768.
- University of Heidelberg. (n.d.). *TSPLIB95: Traveling Salesman Problem library*. Institut für Informatik, Universität Heidelberg. Retrieved August 23, 2025.
- Wang, G., Ma, L., & Chen, J. (2017). A bilevel improved fruit fly optimization algorithm for the nonlinear bilevel programming problem. *Knowledge-Based Systems*, 138, 113–123.
- Wang, X., Li, H., & Yang, J. (2019). An improved fruit fly optimization algorithm for solving combinatorial optimization problems. *Expert Systems with Applications*, 127, 20–35.
- Xu, H., & Chen, Y. (2020). A chaotic fruit fly optimization algorithm for solving the traveling salesman problem. *Computers & Industrial Engineering*, 142, 106372.
- Zhang, L., Wang, S., & Liu, B. (2015). Fruit fly optimization algorithm for feature selection in classification. *Knowledge-Based Systems*, 89, 275–286.
- Zhang, Y., Wang, H., & Li, J. (2023). Adaptive fruit fly optimization algorithm for solving discrete traveling salesman problems. *Applied Soft Computing*, 137, 110436.
- Zheng, X. L., Wang, L., & Wang, S. Y. (2014). A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem. *Knowledge-Based Systems*, 57, 95–103.