การกำหนดค่าพารามิเตอร์ให้เหมาะสมของอัลกอริทึมอาณานิคมผึ้งประดิษฐ์ด้วยวิธีพื้นผิวตอบสนอง

อภิรักษ์ ขัดวิลาศ* ไพโรจน์ จันทร์แก้ว และ จักรพันธ์ ถาวรงามยิ่งสกุล

สาขาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ตาก 41/1 หมู่ 7 ถนนพหลโยธิน ตำบลไม้งาม อำเภอเมืองตาก จังหวัดตาก 63000

รับบทความ 6 สิงหาคม 2565 แก้ไขบทความ 2 เมษายน 2566 ตอบรับบทความ 10 เมษายน 2566

บทคัดย่อ

อัลกอริทึมอาณานิคมผึ้งประดิษฐ์เป็นหนึ่งในวิธีแบบเมตาฮิวริสติกส์แบบกลุ่มที่มีประสิทธิภาพที่ยอดเยี่ยมและการใช้งานที่เรียบง่าย แต่เป็นที่ทราบกันดีว่าวิธีแบบเมตาฮิวริสติกส์นั้นถ้าต้องการให้เกิดประสิทธิภาพสูงสุดต้องมีการกำหนดค่าพารามิเตอร์ของอัลกอริทึมให้เหมาะสมด้วย ดังนั้นงานวิจัยนี้จึงมีวัตถุประสงค์เพื่อใช้วิธีพื้นผิวตอบสนองในการวิเคราะห์ระดับของพารามิเตอร์ที่เหมาะสมของอัลกอริทึมอาณานิคมผึ้งประดิษฐ์กับปัญหาทดสอบแบบตัวแปรต่อเนื่อง ในการดำเนินการวิจัยนี้จะเริ่มจากการออกแบบการทดลองแบบ 3^k แฟคทอเรียล โดยพารามิเตอร์ที่ถูกนำมาใช้ในการออกแบบ ประกอบไปด้วย รอบในการวนซ้ำ (Iteration) จำนวนประชากรผึ้ง (Number of population) และขีดจำกัด (Limit) หลังจากนั้นนำแบบการทดลองที่ได้ไปทำการทดลองเพื่อเก็บข้อมูล และนำข้อมูลที่ได้จากการทดลองมาทำการวิเคราะห์ด้วยวิธีพื้นผิวตอบสนอง จากผลการวิเคราะห์ข้อมูลพบว่า ผลกระทบหลักซึ่งเกิดจากการเปลี่ยนระดับของทั้งสามพารามิเตอร์ ผลกระทบร่วมระหว่างจำนวนประชากรผึ้งและขีดจำกัด และโมเดลกำลังสองมีนัยสำคัญทางสถิติที่ 0.05 (P-value<0.05) ขั้นตอนสุดท้ายคือการนำตัวแบบการถดถอยไปหาค่าพารามิเตอร์ที่ส่งผลให้ได้คำตอบที่ดีที่สุด ผลจากการวิเคราะห์ตัวแบบการถดถอยพบว่า ค่าพารามิเตอร์ที่เหมาะสมของ อัลกอริทึมอาณานิคมผึ้งประดิษฐ์ในการแก้ปัญหาฟังก์ชันคณิตศาสตร์แบบตัวแปรต่อเนื่อง ควรกำหนดระดับของ รอบในการวนซ้ำ จำนวนประชากรผึ้ง และขีดจำกัด เป็น 900 40 และ 90 ตามลำดับ

คำสำคัญ : อาณานิคมผึ้งประดิษฐ์; วิธีพื้นผิวตอบสนอง; เมตาฮิวริสติกส์

ผู้นิพนธ์ประสานงานโทร: +668 6926 9589, ไปรษณีย์อิเล็กทรอนิกส์: aphirak@rmutl.ac.th

Parameter optimization of Artificial Bee Colony Algorithm with Response Surface Methodology

Aphirak Khadwilard* Phairoach Chunkaew and
Chakkraphan Thawonngamyingsakul

Department of Mechanical Engineering, Faculty of Engineering, Rajamangala University of Technology Lanna Tak
41/1 Moo. 7, Paholayothin Road, Mai Ngam, Muang, Tak, 63000

Received 6 August 2022; Revised 2 April 2023; Accepted 10 April 2023

Abstract

The Artificial Bee Colony Algorithm is one of the metaheuristic with excellent performance and simple implementation. But it is well known that a metaheuristic approach is required to optimize the algorithm's parameters. Therefore, this research aims to use the response surface methodology for analyze the optimal level parameters of the artificial bee colony algorithm for continuous variable benchmark problems. To carry out this research, a 3^K factorial experimental design was used, with the design parameters consisting of iteration, number of population, and limit. After that, the experimental table was used to collect data, and the data obtained from the experiments was analyzed by the response surface methodology. From the results of the data analysis, it was found that the main effects resulting from the level change of the three parameters, the interaction effect of the bee population and the limit, and the quadratic model were statistically significant at 0.05 (P-value <0.05). The final step was to run the regression model for finding the parameters that resulted in the best answer. The results of the regression model analysis revealed that the optimum parameters of the Artificial Bee Colony Algorithm to solve continuous variable mathematics functions should be set at iteration, bee population, and limits of 900, 40 and 90, respectively.

Keywords : Artificial Bee Colony; Response Surface Methodology; Metaheuristics

Corresponding Author. Tel.: +668 6926 9589, E-mail Address: aphirak@rmutl.ac.th

1. บทนำ

อัลกอริทึมการหาค่าที่เหมาะสมเป็นเทคนิคทางคณิตศาสตร์ซึ่งใช้ในการแก้ปัญหาทางวิศวกรรมในทางปฏิบัติที่หลากหลายมีวิธีการหาค่าที่เหมาะสมแบบดั้งเดิม เช่น Steepest descent method [1], Simplex algorithm [2], Newton's method [3], Conjugate direction method [4] และวิธีอื่น ๆ วิธีเหล่านี้โดยทั่วไปต้องการฟังก์ชันวัตถุประสงค์ที่มีความต่อเนื่องและหาค่าอนุพันธ์ได้ ด้วยการพัฒนาเทคโนโลยีทางวิทยาศาสตร์และวิศวกรรม ปัญหาการหาค่าที่เหมาะสม (Optimization problems) มีเพิ่มมากขึ้นเรื่อย ๆ และได้พัฒนาเป็นแบบ Non-convex, Multi-modal, Non-linear และ Non-differentiable และไม่ต่อเนื่อง ซึ่งไม่เหมาะที่จะใช้การหาค่าที่เหมาะสมแบบดั้งเดิม

เพื่อแก้ปัญหาการหาค่าที่เหมาะสมที่ซับซ้อน จึงได้มีการเสนอวิธีการหาค่าที่เหมาะสมแบบต่าง ๆ วิธีการที่ได้รับความนิยมเป็นอย่างมากคือ อัลกอริทึมเชิงวิวัฒนาการ (Evolutionary Algorithms) ซึ่งต่อมาได้ถูกเรียกว่า เมตาฮิวริสติกส์ (Metaheuristics) มีความโดดเด่นเนื่องจากมีประสิทธิภาพสูง มีความยืดหยุ่นในการใช้งาน และมีศักยภาพในการประยุกต์ใช้อย่างมาก [5]-[8] อัลกอริทึมเชิงวิวัฒนาการ หรือเมตาฮิวริสติกส์สามารถแบ่งคร่าว ๆ ได้เป็นสามลักษณะ [9] คือ 1. อัลกอริทึมเมตาฮิวริสติกส์จากการวิวัฒนาการทางชีววิทยา (Metaheuristic algorithms based on the evolution of biology) 2. อัลกอริทึมเมตาฮิวริสติกส์จากพฤติกรรมของกลุ่มประชากร (Metaheuristic algorithms based on the behavior of the population) 3. อัลกอริทึมเมตาฮิวริสติกส์จากปรากฏการณ์ที่ไม่ได้มาจากชีววิทยา (Metaheuristic algorithms based on the nonbiological phenomenon) อัลกอริทึมเมตาฮิวริสติกส์จากการวิวัฒนาการทางชีววิทยาส่วนใหญ่จะใช้เพื่อจำลองกระบวนการวิวัฒนาการทางชีววิทยา และขั้นตอนที่สำคัญคือการสลับสายพันธ์ (Crossover) และการกลายพันธุ์ (Mutation) ของยีน ตัวอย่างเช่น อัลกอริทึมทางพันธุกรรม (Genetic Algorithm: GA) [10], [11] และวิธีการวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution: DE) [12] เป็นต้น อัลกอริทึมเมตาฮิวริสติกส์จากพฤติกรรมของกลุ่มประชากรโดยพื้นฐานแล้วได้จากจำลองพฤติกรรมต่าง ๆ ของสิ่งมีชีวิตตามธรรมชาติ เช่น พฤติกรรมการหาอาหาร พฤติกรรมการค้นหา และพฤติกรรมการย้ายถิ่น เป็นต้น ตัวอย่างเช่น การหาค่าที่เหมาะสมแบบอาณานิคมมด (Ant Colony Optimization; ACO) [13] การหาค่าที่เหมาะสมแบบกลุ่มอนุภาค (Particle swarm optimization; PSO) [14] และอัลกอริทึมอาณานิคมผึ้งประดิษฐ์ (Artificial Bee Colony; ABC) [15] ในส่วนของอัลกอริทึมเมตาฮิวริสติกส์จากปรากฏการณ์ที่ไม่ได้มาจากชีววิทยาจะใช้การจำลองตามกฎของธรรมชาติ เช่น อัลกอริทึมวัฏจักรของน้ำ (Water Cycle Algorithm; WCA) [16] อัลกอริทึมการค้นหาแบบแรงโน้มถ่วง (Gravitational Search Algorithm; GSA) [17] และการหาค่าที่เหมาะสมแบบปฏิกิริยาเคมี (Chemical Reaction Optimization; CRO) [18] เป็นต้น

อัลกอริทึมอาณานิคมผึ้งประดิษฐ์ [15] เป็นหนึ่งในอัลกอริทึมเมตาฮิวริสติกส์แบบกลุ่ม เนื่องจากประสิทธิภาพที่ยอดเยี่ยมและการใช้งานที่เรียบง่าย อัลกอริทึมอาณานิคมผึ้งประดิษฐ์จึงได้รับความสนใจอย่างกว้างขวาง ในการที่จะทำให้อัลกอริทึมแบบเมตาฮิวริสติกส์สามารถที่จะทำงานได้เต็มประสิทธิภาพนั้น สิ่งที่สำคัญอย่างหนึ่งคือการกำหนดค่าพารามิเตอร์ที่เหมาะสมให้กับอัลกอริทึม เพื่อให้เหมาะสมกับปัญหาแต่ละชนิด ต่อมา Akay และ Karaboga [19], Yap และคณะ [20] ได้นำเสนอเกี่ยวกับการปรับพารามิเตอร์สำหรับอัลกอริทึมอาณานิคมผึ้งประดิษฐ์ โดยในงานวิจัยเหล่านี้ได้ศึกษาพารามิเตอร์แต่ละตัวแยกกันด้วยวิธีการทดลองทีละปัจจัย (One factor at a time) ซึ่งไม่ได้ทำการศึกษาผลกระทบร่วมระหว่างปัจจัย ดังนั้นในงานวิจัยนี้จึงสนใจที่จะทำการประยุกต์ใช้การออกแบบการทดลองและวิเคราะห์ผลด้วยวิธีพื้นผิวตอบสนองกับปัญหาตัวแปรแบบต่อเนื่องกับอัลกอริทึมอาณานิคมผึ้ง

ประดิษฐ์ โดยพารามิเตอร์หลักที่นำมาพิจารณา ได้แก่ รอบในการวนซ้ำ (Iteration) จำนวนประชากรผึ้ง (Number of population) และขีดจำกัด (Limit)เพื่อเป็นแนวทางในการหาค่าพารามิเตอร์ที่เหมาะสมให้กับวิธีเมตาฮิวริสติกส์ต่อไป

2. อัลกอริทึมอาณานิคมผึ้งประดิษฐ์ (Artificial Bee Colony Algorithm)

อัลกอริทึมอาณานิคมผึ้งประดิษฐ์ถูกนำเสนอโดย Karaboga [15] โดยได้รับแรงบันดาลใจจากพฤติกรรมอันชาญฉลาดของการหาอาหารและการเต้นรำของอาณานิคมผึ้ง อัลกอริทึมนี้เป็นการหาค่าที่เหมาะสมแบบกลุ่มประชากร (Population-based) และพยายามค้นหาค่าคำตอบที่ดีที่สุดซึ่งอาจจะเป็นค่าต่ำสุดหรือค่าสูงสุด (Global minimum or maximum) ด้วยวิธีการวนซ้ำ โดยเงื่อนไขของการจบการค้นหาสำหรับอัลกอริทึมอาณานิคมผึ้งประดิษฐ์อาจจะเป็นจำนวนรอบสูงสุดหรือค่าความผิดพลาดที่ยอมรับได้ ประชากรในกลุ่มอาณานิคมผึ้งประดิษฐ์ประกอบด้วยผึ้งสามชนิดคือ ผึ้งงาน (Employed Bee) ผึ้งสังเกตการณ์ (Onlooker Bee) และผึ้งสำรวจ (Scout Bee) โดยผึ้งงานและผึ้งสังเกตการณ์ จะสำรวจแหล่งน้ำหวานใหม่โดยอ้างอิงแหล่งน้ำหวานที่พบรอบ ๆ รัง ในขณะที่ผึ้งสำรวจจะทำการค้นหาเป็นวงกว้างในบริเวณพื้นที่ของคำตอบที่เป็นไปได้ ในอัลกอริทึมอาณานิคมผึ้งประดิษฐ์จะกำหนดให้จำนวนแหล่งน้ำหวานรอบ ๆ รังเท่ากับจำนวนผึ้งงานนอกจากนี้จำนวนผึ้งงานจะเท่ากับจำนวนผึ้งสังเกตการณ์ ในการวนซ้ำจนพบเงื่อนไขการหยุด จำนวนผึ้งสำรวจในอัลกอริทึมอาณานิคมผึ้งประดิษฐ์อาจจะเกิดขึ้นเพียงแค่หนึ่งตัวก็ได้ขึ้นอยู่กับการตั้งเงื่อนไขตัวนับการละทิ้ง (Abandonment counter: AC) รูปที่ 1 แสดงแนวคิดกระบวนการหาอาหารของผึ้งในอาณานิคมผึ้งประดิษฐ์ โดยสามารถสรุปขั้นตอนของอัลกอริทึมอาณานิคมผึ้งประดิษฐ์ซึ่งมีสี่ขั้นตอนดังต่อไปนี้ ขั้นตอนเริ่มต้น (Initialization Phase) ขั้นตอนผึ้งงาน (Employed Bee Phase) ขั้นตอนผึ้งสังเกตการณ์ (Onlooker Bees Phase) และขั้นตอนผึ้งสำรวจ (Scout Bee Phase) ตามลำดับ

2.1 ขั้นตอนเริ่มต้น (Initialization Phase)

อัลกอริทึมอาณานิคมผึ้งประดิษฐ์เริ่มทำงานโดยสุ่มสร้างผลเฉลยเริ่มต้น (Initial solutions) ซึ่งผลเฉลยนี้ถูกสร้างขึ้นสำหรับผึ้งงาน (Employed bees) โดยใช้สมการที่ (1)

$$x_{i,j} = x_j^{\min} + \lambda(x_j^{\max} - x_j^{\min}), \quad i = 1 \ldots N, j = 1 \ldots D \tag{1}$$

โดยที่

$x_{i,j}$ = มิติที่ jth ของผึ้งงานลำดับที่ ith

$x_j^{\min}$ = ขอบเขตล่างของพารามิเตอร์ มิติที่ jth

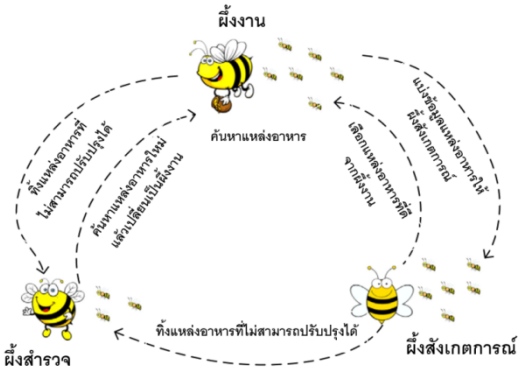
$x_j^{\max}$ =ขอบเขตบนของพารามิเตอร์ มิติที่ jth

λ = เป็นตัวเลขสุ่มในช่วง [0, 1]

N = จำนวนผึ้งงาน

D = มิติของปัญหาการหาค่าเหมาะสม

นอกจากนี้ ตัวนับการละทิ้ง (AC) ของผึ้งงานแต่ละตัวจะถูกรีเซ็ตในขั้นตอนนี้



รูปที่ 1 กระบวนการหาอาหารของผึ้งในอาณานิคมผึ้งประดิษฐ์ [9]

2.2 ขั้นตอนผึ้งงาน (Employed Bee Phase)

ในขั้นตอนนี้ผลเฉลยชุดใหม่จะถูกสร้างขึ้นสำหรับผึ้งงานแต่ละตัว โดยเริ่มจากผลเฉลยของผึ้งงานจะถูกคัดลอกไปยังผลเฉลยใหม่ ($v_i = x_i$) จากนั้นอัพเดตค่าพารามิเตอร์ของผลเฉลยเพียงมิติเดียวโดยใช้สมการที่ (2)

$$\begin{aligned} & v_{i,j} = x_{i,j} + \phi(x_{i,j} - x_{k,i}) \\ & i,k \in \{1,2,\ldots,N\}, \\ & j \in \{1,2,\ldots,D\} \text{ and } i \neq k \end{aligned} \quad (2)$$

โดยที่

$v_{i,j}$ = มิติที่ jth ของผลเฉลยใหม่ลำดับที่ ith

$x_{i,j}$ = มิติที่ jth ของผึ้งงานลำดับที่ ith

$x_{k,j}$ = มิติที่ jth ของผึ้งงานลำดับที่ kth

ϕ = ตัวเลขสุ่มในช่วง [-1, +1]

N = จำนวนผึ้งงาน

D = มิติของปัญหาการหาค่าที่เหมาะสม

นอกจากนี้ ผลเฉลยลำดับที่ k และมิติของปัญหาที่ j จะถูกสุ่มเลือกจากประชากรผึ้งงานและระหว่างขนาดมิติของปัญหา ตามลำดับ

หลังจากที่ผลเฉลยชุดใหม่ถูกสร้างและคำนวณค่าฟังก์ชันวัตถุประสงค์เฉพาะสำหรับปัญหาแล้ว ค่าความเหมาะสมของคำตอบ (Fitness value) ของผลเฉลยชุดใหม่ และผลเฉลยของผึ้งงานจะถูกคำนวณดังนี้

$$fit_i = \begin{cases} \dfrac{1}{1+f_i} & \text{if } (f_i \geq 0) \\ 1 + abs(f_i) & \text{otherwise} \end{cases} \quad (3)$$

โดยที่

fit_i = ค่าความเหมาะสมของคำตอบของผึ้งงานลำดับที่ ith

f_i = ค่าฟังก์ชันวัตถุประสงค์ของผึ้งงานลำดับที่ ith

หากค่าความเหมาะสมของคำตอบใหม่ดีกว่าค่าความเหมาะสมของคำตอบของผึ้งงาน ผึ้งงานจะถูกแทนที่ด้วยผลเฉลยของตัวใหม่ ($v_i = x_i$) และตัวนับการละทิ้ง (AC) ของผึ้งงานจะถูกรีเซ็ต ถ้าไม่เป็นไปตามเงื่อนไขนี้ ตัวนับการละทิ้งจะเพิ่มขึ้น 1 (AC =AC +1)

2.3 ขั้นตอนผึ้งสังเกตการณ์ (Onlooker Bees Phase)

ในอัลกอริทึมอาณานิคมผึ้งประดิษฐ์ ผึ้งสังเกตการณ์แต่ละตัวจะเลือกผึ้งงานเพื่อปรับปรุงผลเฉลย ในการเลือกนี้จะเลือกตามค่าความเหมาะสมของคำตอบ (Fitness value) ของผึ้งงาน โดยใช้วงล้อรูเล็ต (Roulette wheel) ซึ่งสร้างจากสมการที่ (4)

$$p_i = \frac{fit_i}{\sum_{j=1}^{N} fit_j} \quad (4)$$

โดยที่

p_i = ความน่าจะเป็นของผึ้งงานลำดับที่ ith ที่จะถูกเลือกโดยผึ้งสังเกตการณ์

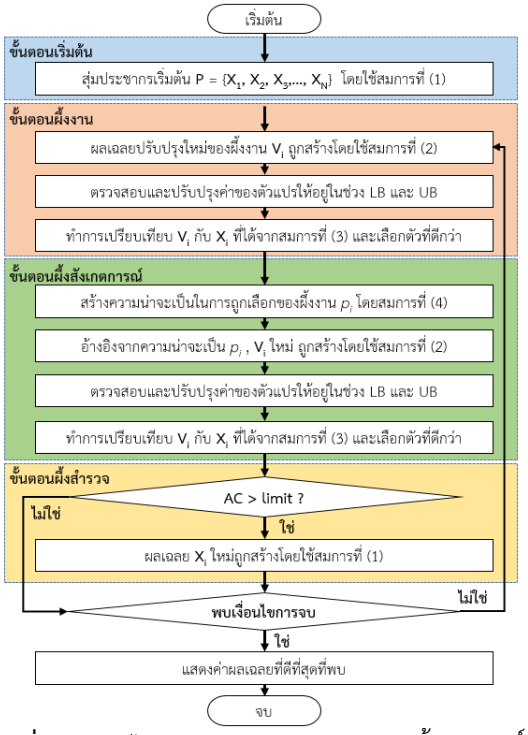
โดยที่ผึ้งสังเกตการณ์พยายามปรับปรุงผลเฉลยของผึ้งงานที่คัดเลือกมาโดยใช้สมการที่ (2) หากค่าความเหมาะสมของคำตอบใหม่ที่พบดีกว่าค่าความเหมาะสมของคำตอบของผึ้งงาน ผึ้งสังเกตการณ์จะเปลี่ยนไปเป็นผึ้งงาน และรีเซ็ตตัวนับการละทิ้ง (AC) ของผึ้งงานถ้าไม่เข้าเงื่อนไขนี้ตัวนับการละทิ้งจะเพิ่มขึ้น 1 (AC = AC+1) โดยสรุปแล้ว ผึ้งสังเกตการณ์และผึ้งงานจะใช้สมการเดียวกันในการค้นหาโดยรอบ (Exploitation) บริเวณแหล่งอาหาร

2.4 ขั้นตอนผึ้งสำรวจ (Scout Bee Phase)

ตัวนับการละทิ้ง (AC) ของผึ้งงานทั้งหมดจะถูกตรวจสอบด้วยตัวเลขซึ่งกำหนดโดยผู้ใช้งานอัลกอริทึม คือตัวแปรขีดจำกัด (Limit) ผึ้งงานซึ่งไม่สามารถปรับปรุง

ผลเฉลยด้วยตนเองได้จนกระทั่งตัวนับการละทิ้ง (AC) ถึงขีดจำกัด (Limit) จะกลายเป็นผึ้งสำรวจ โดยที่การสร้างผลเฉลยสำหรับผึ้งสำรวจจะใช้สมการที่ (1) และตัวนับการละทิ้งจะถูกรีเซ็ต หลังจากนั้นผึ้งสำรวจซึ่งสร้างผลเฉลยใหม่สำหรับตัวมันเองจะเปลี่ยนมาเป็นผึ้งงานอีกครั้ง ดังนั้น ผึ้งสำรวจในอัลกอริทึมอาณานิคมผึ้งประดิษฐ์จึงป้องกันการหยุดนิ่ง (Stagnation) ของประชากรผึ้งงาน

จากคำอธิบายข้างต้น ผังการไหลของอัลกอริทึมอาณานิคมผึ้งประดิษฐ์ได้ถูกแสดงดังรูปที่ 2



รูปที่ 2 ผังการไหลของอัลกอริทึมอาณานิคมผึ้งประดิษฐ์

3. วิธีพื้นผิวตอบสนองสำหรับการหาค่าที่เหมาะสมของกระบวนการ (Response surface methodology for process optimization)

วิธีพื้นผิวตอบสนองเป็นวิธีการทางสถิติและทางคณิตศาสตร์ที่ตรวจสอบความสัมพันธ์ระหว่างพารามิเตอร์กระบวนการอินพุต และตัวแปรการตอบสนอง (เอาต์พุต) วิธีนี้ได้รับการแนะนำโดย Montgomery [21] วิธีพื้นผิวตอบสนอง (RSM) ใช้การออกแบบการทดลองตามลำดับเพื่อให้ได้การตอบสนองที่ดีที่สุด วิธีนี้เป็นวิธีการที่ช่วยในการพัฒนา ปรับปรุง และปรับพารามิเตอร์กระบวนการให้เหมาะสม ขั้นตอนการออกแบบของวิธีพื้นผิวตอบสนอง (RSM) ประกอบด้วยขั้นตอนต่อไปนี้ 1. การออกแบบชุดการทดลอง 2. การกำหนดแบบจำลองทางคณิตศาสตร์ที่เหมาะสมที่สุด 3.การหาชุดปัจจัยการทดลองที่เหมาะสมที่สุดเพื่อสร้างการตอบสนองสูงสุดหรือต่ำสุด 4. แสดงผลกระทบหลัก (Main effect) และผลกระทบร่วม (Interaction effect) ของตัวแปรกระบวนการและพื้นผิวตอบสนอง

4. วิธีดำเนินการวิจัย

4.1 การออกแบบการทดลองและผลการทดลอง

การออกแบบการทดลองแบบแฟคทอเรียล 3^k (3^k factorial design) [22] ถูกออกแบบให้กับพารามิเตอร์ของอัลกอริทึมอาณานิคมผึ้งประดิษฐ์ซึ่งประกอบไปด้วย พารามิเตอร์ 3 ตัว [19], [20] พารามิเตอร์ตัวแรกคือ รอบในการวนซ้ำ (Iteration; MaxIter) ซึ่งจะเป็นตัวที่กำหนดรอบในการหยุดค้นหาผลเฉลยของอัลกอริทึม ในขณะที่พารามิเตอร์ตัวที่สองจะเป็น จำนวนประชากรผึ้ง (Number of Population; NP) จะเป็นตัวกำหนดประชากรของผึ้งในการค้นผลเฉลย และตัวสุดท้ายคือ ขีดจำกัด (Limit) เป็นตัวแปรที่กำหนดการละทิ้งแหล่งอาหารปัจจุบันของผึ้งงานและผึ้งสังเกตการณ์ เพื่อเป็นผึ้งสำรวจในการค้นหาแหล่งอาหารใหม่ ในงานวิจัยนี้ ปัจจัยการทดลองและค่าระดับปัจจัยที่พิจารณาได้ถูกแสดงไว้ในตารางที่ 1 การออกแบบการทดลองแฟคทอเรียล 3^k ซึ่งมีการทดลอง 3 ซ้ำ และผลที่ได้จากการทดลองแสดงดังตารางที่ 2

ตารางที่ 1 ปัจจัยการทดลองและระดับปัจจัย

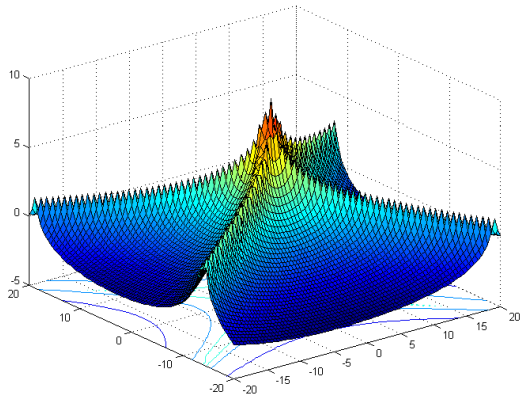
ปัจจัย	ระดับปัจจัย (Coded)		
	-1	0	+1
รอบในการวนซ้ำ	100	500	900
จำนวนประชากรผึ้ง	10	25	40
ขีดจำกัด	10	50	90

ตารางที่ 2 แบบการทดลองแฟคทอเรียล 3^k และผลการทดลอง

RUN	MaxIter	NP	Limit	Avg. Solution
1	100	10	10	8.35364
2	100	10	50	8.26353
3	100	10	90	8.24138
4	100	25	10	9.32991
5	100	25	50	8.83608
6	100	25	90	8.98511
7	100	40	10	8.94659
8	100	40	50	9.48359
9	100	40	90	9.51797
10	500	10	10	9.22240
11	500	10	50	8.70006
12	500	10	90	8.74521
13	500	25	10	9.45969
14	500	25	50	9.48091
15	500	25	90	9.50506
16	500	40	10	9.32106
17	500	40	50	9.48359
18	500	40	90	9.52111
19	900	10	10	9.28512
20	900	10	50	9.26896
21	900	10	90	8.74521
22	900	25	10	9.50483
23	900	25	50	9.52193
24	900	25	90	9.50735
25	900	40	10	9.44367
26	900	40	50	9.51646
27	900	40	90	9.52252

4.2 ปัญหาที่ใช้ในการทดลอง

ในงานวิจัยนี้ มีการใช้ฟังก์ชันทางคณิตศาสตร์ชนิดตัวแปรต่อเนื่องที่ไม่เป็นเชิงเส้น คือฟังก์ชันทดสอบโกลด์สเตนไพรซ์ (Goldstein-price function) จากสมการที่ (5) ซึ่งลักษณะของฟังชันจะเป็นแบบ Multi-modal โดยมีจุด Local optimum จำนวนมาก ดังแสดงในรูปที่ 3 เพื่อเป็นแนวทางในการประยุกต์ใช้วิธีพื้นผิวตอบสนองในการหาค่าพารามิเตอร์ที่เหมาะสมให้กับอัลกอริทึมอาณานิคมผึ้งประดิษฐ์



รูปที่ 3 ฟังก์ชันทดสอบโกลด์สเตนไพรซ์ (Goldstein-price function)

$$f(x,y) = 10 - \log_{10}[\{1+(1+x+y)^2 (19-14x+3x^2-14y+6xy+3y^2)\} * \{30+(2x-3y)^2(18-32x+12x^2+48y-36xy+27y^2)\}] \quad (5)$$

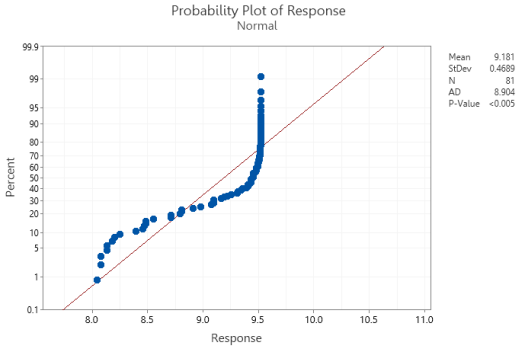
ช่วงของตัวแปร: $-20 < x < 20$; $-20 < y < 20$

ผลเฉลยที่ดีที่สุด: $f(x^*, y^*) = 9.522878$ ที่ $x^* = 0$ และ $y^* = -1$

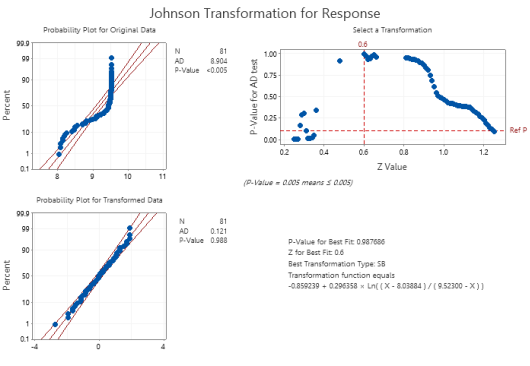
5. การวิเคราะห์ผลการทดลอง

5.1 วิเคราะห์การแจกแจงของข้อมูล

จากผลการทดลองข้อมูลผลเฉลยจะถูกนำไปวิเคราะห์ว่ามีการแจกแจงของข้อมูลเป็นแบบปกติ (Normal distribution) หรือไม่ เพื่อให้เข้าเงื่อนไขการวิเคราะห์ข้อมูลแบบพาราเมตริก ผลการวิเคราะห์ข้อมูลผลเฉลยที่ได้จากการทดลองพบว่าข้อมูลไม่มีการแจกแจงแบบปกติแสดงดังรูปที่ 4 ดังนั้นจึงต้องมีการแปลงข้อมูลโดยวิธี Johnson Transform สมการในการแปลงข้อมูลและผลจากการแปลงข้อมูลแสดงดังรูปที่ 5 ซึ่งพบว่าข้อมูลที่ถูกแปลงมีการแจกแจงแบบปกติ



รูปที่ 4 ผลการทดสอบการกระจายตัวของข้อมูลผลเฉลยจากการทดลอง



รูปที่ 5 สมการและผลการแปลงข้อมูลผลเฉลย

5.2 การวิเคราะห์ข้อมูลด้วยวิธีพื้นผิวตอบสนอง

การวิเคราะห์ความแปรปรวน (ANOVA) เป็นสิ่งที่ขาดไม่ได้ในการตรวจสอบนัยสำคัญและความสามารถของแบบจำลอง การวิเคราะห์ความแปรปรวนจะระบุว่าแบบจำลองกำลังสองที่ถูกโมเดลนั้นมีนัยสำคัญทางสถิติหรือไม่ รวมถึงตรวจสอบผลกระทบหลักที่เกิดจากการเปลี่ยนระดับของปัจจัย และผลกระทบร่วมระหว่างปัจจัยว่ามีนัยสำคัญทางสถิติหรือไม่ผลจากการวิเคราะห์ส่วนตกค้างในรูปที่ 6 แสดงให้เห็นว่า ข้อมูลที่นำไปวิเคราะห์มีการแจกแจงแบบปกติและมีการกระจายตัวของข้อมูลสม่ำเสมอ และผลการวิเคราะห์ความแปรปรวนในตารางที่ 3 แสดงให้เห็นว่าโมเดลกำลังสองมีนัยสำคัญทางสถิติ ดังนั้นจึงเหมาะสมกับการใช้การวิเคราะห์ด้วยวิธีพื้นผิวตอบสนอง นอกจากนั้นผลกระทบหลักและผลกระทบร่วมของปัจจัย มีนัยสำคัญทางสถิติที่ 0.05 (P-value<0.05) และมีค่า Lack-of-Fit 0.687 ซึ่งมากกว่า 0.05 และมีค่า R^2 = 73.52% แสดงว่าโมเดลมีความเหมาะสม

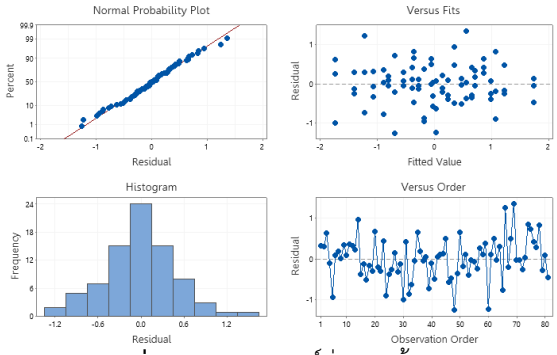
ตารางที่ 3 การวิเคราะห์ความแปรปรวน

Source	DF	F-Value	P-Value
Regression	5	41.65	0.000
MaxIter	1	52.57	0.000
NP	1	17.79	0.000
Limit	1	18.52	0.000
NP*NP	1	19.68	0.000
P*Limit	1	38.55	0.000
Error	75		
Lack-of-Fit	21	0.82	0.687
Pure Error	54		
Total	80		

R-sq 73.52% R-sq (adj) 71.76%

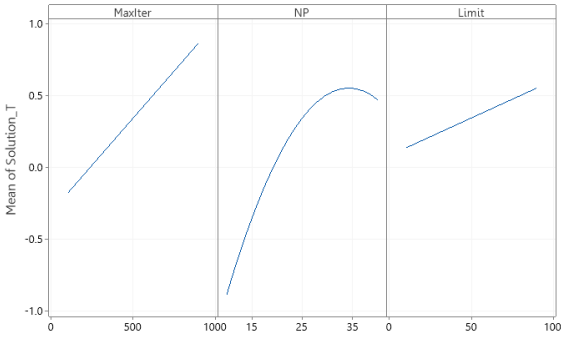
ตัวแบบการถดถอยโดยประมาณโดยใช้ตัวแปรรหัสแสดงดังสมการที่ (6)

$$\begin{aligned} Solution_T &= -2.092 \\ &+ 0.001303\ MaxIter + 0.1224\ NP \\ &- 0.01757\ Limit - 0.002454\ NP^2 \\ &+ 0.000911\ NP*Limit \end{aligned} \quad (6)$$

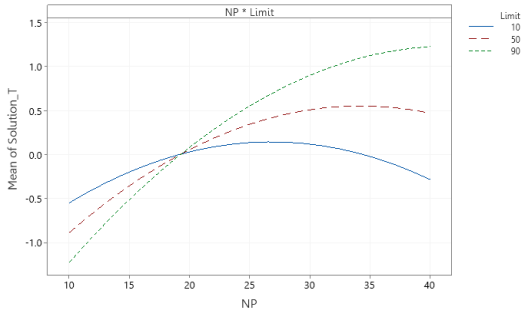


รูปที่ 6 กราฟวิเคราะห์ส่วนตกค้าง

ในการหาค่าพารามิเตอร์ที่เหมาะสมของอัลกอริทึมอาณานิคมผึ้งประดิษฐ์ ในเบื้องต้นจะทำการพิจารณากราฟผลกระทบหลักในการเปลี่ยนระดับของพารามิเตอร์ และกราฟผกระทบร่วมของพารามิเตอร์ ซึ่งเมื่อพิจารณาจากกราฟผลผลกระทบหลักในรูปที่ 7 พบว่าควรกำหนด ระดับของพารามิเตอร์ รอบในการวนซ้ำ (MaxIter) จำนวนประชากรผึ้ง (NP) และ ขีดจำกัด (Limit) ที่ประมาณ 900 34 และ 90 ตามลำดับ แต่เมื่อพิจารณาจากกราฟผลกระทบร่วมในรูปที่ 8 พบว่า ควรกำหนด จำนวนประชากรผึ้ง (NP) และ ขีดจำกัด (Limit) ที่ 40 และ 90 ตามลำดับ

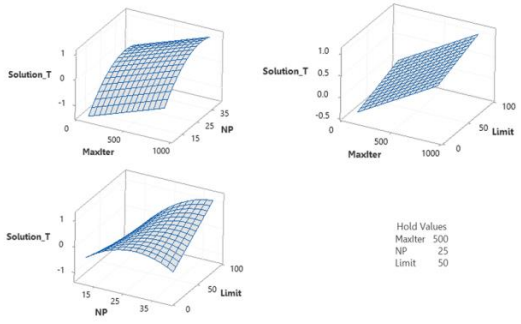


รูปที่ 7 ผลกระทบหลักของพารามิเตอร์

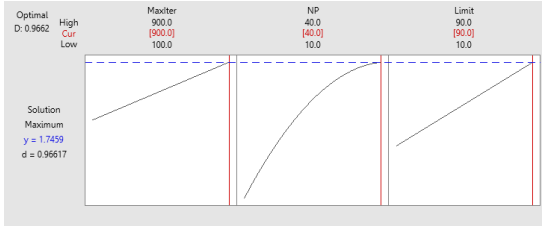


รูปที่ 8 ผลกระทบร่วมของพารามิเตอร์

เมื่อนำตัวแบบการถดถอยในสมการที่ (6) ไปพล็อตจะได้พื้นผิวตอบสนองดังรูปที่ 9 และจะสังเกตจากพื้นผิวตอบสนองได้ว่า ควรกำหนด ระดับของพารามิเตอร์ รอบในการวนซ้ำ (MaxIter) จำนวนประชากรผึ้ง (NP) และ ขีดจำกัด (Limit) ที่ประมาณ 900 40 และ 90 ตามลำดับเพื่อให้ชัดเจนมากขึ้นจึงนำตัวแบบการถดถอยในสมการที่ (6) ไปหาค่าที่ดีที่สุด โดยเครื่องมือการหาค่าที่ดีที่สุดของการตอบสนอง (Response optimizer) โดยผลจากการวิเคราะห์แสดงดังรูปที่ 10 พบว่า ควรกำหนดกำหนด ระดับของพารามิเตอร์ รอบในการวนซ้ำ (MaxIter) จำนวนประชากรผึ้ง (NP) และ ขีดจำกัด (Limit) ที่ 900 40 และ 90 ตามลำดับ



รูปที่ 9 พื้นผิวตอบสนองของพารามิเตอร์



รูปที่ 10 ระดับพารามิเตอร์ที่เหมาะสมที่สุด

6. สรุป

อัลกอริทึมอาณานิคมผึ้งประดิษฐ์หนึ่งในอัลกอริทึมเมตาฮิวริสติกส์แบบกลุ่ม ในช่วงทศวรรษที่ผ่านมาเนื่องจากประสิทธิภาพที่ยอดเยี่ยมและการใช้งานที่เรียบง่าย อัลกอริทึมอาณานิคมผึ้งประดิษฐ์จึงได้รับความสนใจอย่างกว้างขวางในการที่จะทำให้อัลกอริทึมแบบเมตาฮิวริสติกส์สามารถที่จะทำงานได้เต็มประสิทธิภาพนั้น สิ่งที่สำคัญอย่างหนึ่งคือการกำหนดค่าพารามิเตอร์ที่เหมาะสมให้กับอัลกอริทึม เพื่อให้เหมาะสมกับปัญหาแต่ละชนิด โดยในงานวิจัยนี้ประยุกต์ใช้การออกแบบการทดลองและวิธีพื้นผิวตอบสนองกับปัญหาตัวแปรแบบต่อเนื่อง เพื่อเป็นแนวทางในการหาค่าพารามิเตอร์ที่เหมาะสมให้กับวิธีเมตาฮิวริสติกส์ การดำเนินการวิจัยเริ่มจากการออกแบบการทดลองแบบแฟคทอเรียล 3^k (3^k factorial design) โดยพารามิเตอร์ของอัลกอริทึมอาณานิคมผึ้งประดิษฐ์ประกอบไปด้วย พารามิเตอร์ 3 ตัว คือ รอบในการวนซ้ำ (Iteration; MaxIter) จำนวนประชากรผึ้ง (Number of Population; NP) และ ขีดจำกัด (Limit) ดังนั้นการทดลองต่อหนึ่งซ้ำ (Replication) จะมีจำนวน 27 รัน ($3^k = 3^3 = 27$) หลังจากที่ได้รูปแบบการทดลองแล้วนำไปทดลอง 3 ซ้ำ และนำผลการทดลองที่ได้ไปทำการวิเคราะห์การแจกแจงของข้อมูลว่ามีการแจกแจงแบบปกติหรือไม่ จากการวิเคราะห์พบว่าข้อมูลที่ได้จากการทดลองไม่มีการแจกแจงแบบปกติ ดังนั้นจึงต้องนำข้อมูลไปแปลงโดยวิธี Johnson Transform เมื่อข้อมูลมีการแจกแจงแบบปกติแล้วจึงนำไปวิเคราะห์ต่อด้วยวิธีพื้นผิวตอบสนอง ผลจากการวิเคราะห์พบว่า ผลกระทบหลักซึ่งเกิดจากการเปลี่ยนระดับของทั้งสามพารามิเตอร์ ผลกระทบร่วมระหว่างจำนวนประชากรผึ้ง และขีดจำกัด และโมเดลกำลังสองมีนัยสำคัญทางสถิติที่ 0.05 (P-value<0.05) ขั้นตอนสุดท้ายคือการนำตัวแบบการถดถอยไปหาค่าพารามิเตอร์ที่ส่งผลให้ได้คำตอบที่ดีที่สุด ผลจากการวิเคราะห์ตัวแบบการถดถอยพบว่าค่าพารามิเตอร์ที่เหมาะสมของอัลกอริทึมอาณานิคมผึ้งประดิษฐ์ในการแก้ปัญหาฟังก์ชันคณิตศาสตร์แบบตัวแปรต่อเนื่อง ควรกำหนดระดับของ รอบในการวนซ้ำ จำนวนประชากรผึ้ง และขีดจำกัด เป็น 900 40 และ 90 ตามลำดับ ซึ่งทั้งหมดเป็นแนวทางในการหาค่าพารามิเตอร์ที่เหมาะสมของอัลกอริทึมเมตาฮิวริสติกส์

7. เอกสารอ้างอิง

[1] P. Deift and X. Zhou, “A steepest descent method for oscillatory Riemann- Hilbert problems,” *Asymptotics for the MKdV equation. Annals of Mathematics*, vol. 137(2), pp. 295-368, 1993.

[2] D. Goldfarb and J. K. Reid, “A practicable steepest- edge simplex algorithm,” *Mathematical Programming*, vol. 12(1), pp. 361-371, 1977.

[3] L. Qi and J. Sun, “A nonsmooth version of Newton's method,” *Mathematical programming*, vol. 58(1), pp. 353-367, 1993.

[4] M. J. Best, “A feasible conjugate-direction method to solve linearly constrained minimization problems,” *Journal of Optimization Theory and Applications*, vol. 16(1), pp. 25-38, 1975.

[5] H. Rao, X. Shi, A. K. Rodrigue, J. Feng, Y. Xia, M. Elhoseny and L. Gu, “Feature selection based on artificial bee colony and gradient boosting decision tree,” *Applied Soft Computing*, vol. 74, pp. 634-642, 2019.

[6] W. Chen, M. Panahi, P. Tsangaratos, H. Shahabi, I. Ilia, S. Panahi, and B. B. Ahmad, “Applying population- based evolutionary algorithms and a neuro- fuzzy system for modeling landslide susceptibility,” *Catena*, vol. 172, pp. 212-231, 2019.

[7] Y. Xue, B. Xue and M. Zhang, "Self-adaptive particle swarm optimization for large-scale feature selection in classification," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13(5), pp. 1-27, 2019.

[8] Q. Yang, W. N. Chen, T. Gu, H. Jin, W. Mao and J. Zhang, " An adaptive stochastic dominant learning swarm optimizer for high-dimensional optimization," *IEEE Transactions on Cybernetics*, 2020.

[9] J. Yang, C. Jiangtao and Y. D. Zhang, " Artificial bee colony algorithm with adaptive covariance matrix for hearing loss detection," *Knowledge-Based Systems*, vol. 216, pp. 1-28, 2021.

[10] Y. Xinjie and G. Mitsuo, " Introduction to Evolutionary Algorithms," *Springer Science & Business Media*, 2010.

[11] A. Khadwilard, "Application of Genetic Algorithm for Optimization Problems", *RMUTP Sci J*, vol. 5, no. 2, pp. 153-163, 2011.

[12] R. Storn and K. V. Price, " Differential Evolution–A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.

[13] M. Dorigo, M. Birattari and T. Stutzle, " Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1(4), pp. 28-39, 2006.

[14] R. C. Eberhart, Y. Shi and J. Kennedy, "Swarm intelligence," *Elsevier*, 2001.

[15] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical report- tr06, Erciyes University, engineering faculty, computer engineering department*, vol. 200, pp. 1-10, 2005.

[16] H. Eskandar, A. Sadollah, A. Bahreininejad and M. Hamdi, " Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures*, vol. 110, pp. 151-166, 2012.

[17] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, " GSA: a gravitational search algorithm," *Information sciences*, vol. 179(13), pp. 2232-2248, 2009.

[18] N. Siddique and H. Adeli, " Nature-inspired chemical reaction optimization algorithms, *Cognitive computation*, vol. 9(4), pp. 411-422, 2017.

[19] B. Akay and D. Karaboga, "Parameter Tuning for the Artificial Bee Colony Algorithm," *Computational Collective Intelligence*, pp. 608-619, 2009.

[20] Y. S. Qi, T. K. Wen, S. Rui, D. Gan, Z. A. Abdul Salam, " Parameter tuning for artificial Bee Colony algorithm," Journal of Applied Technology and Innovation, vol. 5, no. 1, pp. 11-14, 2021.

[21] G. E. P. Box and K. B. Wilson, " On the experimental attainment of optimum conditions," *Journal of the Royal Statistical Society B*, vol. 13, pp. 1–45, 1951.

[22] D. C. Montgomery, *Design and analysis of experiments*, New York: John Wiley and Sons, 2001.