



การเปรียบเทียบค่าความเหมาะสมสำหรับขั้นตอนวิธีแบบเพรียกระโดด

A Comparison of Optimization Problems for Planthopper Algorithm

แก้วใจ อารณพิศาล (Kaewjai Apornpisarn)* พยุง มีสัจ (Phayung Meesad)**

และ วาทีนี นัยเพียร (Vatinee Nuipian)**

Received : July 19, 2018

Revised : September 11, 2018

Accepted : October 10, 2018

บทคัดย่อ

บทความวิจัยนี้เป็นการนำเสนอขั้นตอนวิธีเพรียกระโดดสำหรับการหาค่าความเหมาะสมที่สุดเชิงตัวเลข ที่เกิดจากแนวคิดของพฤติกรรมการหาอาหารแบบเป็นกลุ่ม ด้วยการเคลื่อนที่เพื่อเปลี่ยนตำแหน่ง ซึ่งเป็นการเข้าใกล้แหล่งอาหารคือ การลู่เข้าสู่คำตอบ และจากการเปรียบเทียบขั้นตอนวิธีการค้นหาแบบนกดูเหว่า แบบค้างคาว และแบบแมลงหวี่มีข้อจำกัดในเรื่องการกำหนดพารามิเตอร์เป็นค่าคงที่ และความเร็วของการลู่เข้าสู่คำตอบ ผู้วิจัยจึงทำการเปรียบเทียบกับ 3 ขั้นตอนวิธีข้างต้น จากผลการทดสอบของการหาค่าต่ำสุดของฟังก์ชันทางคณิตศาสตร์ 4 ฟังก์ชันคือ ฟังก์ชัน Sphere ฟังก์ชันไกรเวงก์ ฟังก์ชันโรเซนบร็อก และฟังก์ชันแอกคัสลีย์ พบว่าขั้นตอนวิธีแบบเพรียกระโดดให้ค่าความเหมาะสมเฉลี่ยที่ดีที่สุดในฟังก์ชันไกรเวงก์ ($\bar{x} = 0.00012$, $S.D. = 0.00191$) และฟังก์ชันแอกคัสลีย์ ($\bar{x} = 0.03102$, $S.D. = 0.16076$) ดังนั้นวิธีที่นำเสนอสามารถแก้ไขปัญหาการหาค่าความเหมาะสมที่สุดได้เทียบเคียงกับวิธีอื่น และดีกว่าวิธีอื่นในบางฟังก์ชัน

Abstract

In this research, we discuss the planthopper algorithm (PA) as an algorithm for finding the optimal values. The proposed algorithm is inspired by the food searching habit of the planthopper. To evaluate the proposed algorithm, we tested on optimization of benchmark mathematic functions, comparing to the Cuckoo Search Algorithm, Bat Algorithm and Fruit Fly Algorithm. Those algorithms have limitations on the speed of convergence into solutions and the determination of parameters as constant. In the problem of finding the minimum of 4 mathematical functions, the function is the Sphere function, Griewank function, Rosenbrock function and Ackley function. The Planthopper algorithm gives the best average fitness in the Griewank function ($\bar{x} = 0.00012$, $S.D. = 0.00191$) and the Ackley function ($\bar{x} = 0.03102$, $S.D. = 0.16076$). The results showed that the proposed algorithm could solve the optimization problems comparable to other existing algorithms, even better in some problems.

คำสำคัญ: การหาค่าความเหมาะสม การคำนวณเชิงวิวัฒนาการ อัลกอริธึม ขั้นตอนวิธีแบบเพรียกระโดด

Keywords: Optimization, Evolutionary computation, Heuristic, Planthopper Algorithm.

* คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

* Faculty of Science and Technology, Nakhon Pathom Rajabhat University.

** ภาควิชาการจัดการเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

** Department of Information Technology Management, Faculty of Information Technology, King Mongkut's University of Technology North Bangkok.

*** ภาควิชาคอมพิวเตอร์ศึกษา คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

*** Department of Computer Education, Faculty of Technical Education, King Mongkut's University of Technology North Bangkok.

1. บทนำ

การหาค่าความเหมาะสมที่สุด (Optimization) เป็นวิธีการที่ใช้ในการหาคำตอบที่ดีที่สุดของปัญหาภายใต้เงื่อนไขหรือข้อจำกัดที่กำหนดขึ้น [1] ถือว่าเป็นสิ่งที่ช่วยในการแก้ปัญหาด้านวิทยาการคอมพิวเตอร์ งานด้านปัญญาประดิษฐ์ การวิจัยดำเนินงาน และสาขาอื่นๆ ที่เกี่ยวข้องได้เป็นอย่างดี [2] ขั้นตอนวิธีแบบเมตาฮิวริสติกที่เหมาะสมที่สุด (Metaheuristic optimization algorithm) เกิดจากการเฝ้าสังเกตพฤติกรรมในธรรมชาติที่สนใจ และนำมาสกัดเป็นขั้นตอนวิธี (Algorithm) [3] เพื่อแก้ไขปัญหาที่มีความซับซ้อนได้อย่างรวดเร็ว

คุณลักษณะหลัก 2 ลักษณะของการแก้ปัญหา คือ ระยะ Intensification เป็นการค้นหาผลลัพธ์ที่ดีและคัดเลือกลุ่มแข่งหรือผลลัพธ์ และระยะ Diversification เป็นการรับรองขั้นตอนวิธีที่ค้นหาพื้นที่ให้มีประสิทธิภาพมากขึ้น [4] โดยจุดประสงค์ในการพัฒนา คือ ต้องการหาคำตอบที่รวดเร็วเพื่อแก้ไขปัญหาขนาดใหญ่ เช่น ปัญหาด้านวิศวกรรม ปัญหาการจัดเส้นทางขนส่ง และปัญหาการวางแผนทรัพยากรให้มีประสิทธิภาพมากขึ้น โดยมีหลายขั้นตอนวิธี เช่น จีเนติก (Genetic Algorithm: GA) ซึ่งเป็นการค้นหาแบบฝูงอนุภาค (Particle Swarm Optimization: PSO) หรือการล่าเหยื่อของฝูงสัตว์ [5], [6] ตัวอย่างเช่น ขั้นตอนวิธีหาค่าเหมาะสมแบบฝูงมด (Ant Colony Optimization: ACO) เป็นการหาอาหารในเส้นทางที่สั้นที่สุดด้วยการปล่อยฟีโรโมน [7] ขั้นตอนวิธีแบบหิ่งห้อย (Firefly Algorithm: FA) เป็นการดึงดูดกันของหิ่งห้อยโดยอาศัยการกระพริบแสง [8] ขั้นตอนวิธีการค้นหาแบบนกคูหา (Cuckoo Search : CS) เป็นการนำไข่ไปให้นกชนิดอื่นฟักและรูปแบบการบินของผีเสื้อและแมลงวัน [9] ขั้นตอนวิธีแบบค้างคาว (Bat Algorithm: BA) เป็นการล่าเหยื่อโดยใช้เสียงสะท้อน [10] ขั้นตอนวิธีแบบแมลงหวี่ (Fruit Fly Optimization Algorithm: FOA) เป็นการหาอาหารของแมลงหวี่โดยใช้กลิ่นของอาหาร [11] ซึ่งขั้นตอนวิธีการค้นหาแบบนกคูหาที่มีข้อจำกัดในเรื่องของการกำหนดค่าความเป็นไปได้ในการทิ้งรังและ Step size มีค่าคงที่ หากมีการกำหนดค่าทั้งสองใหม่จะทำให้ประสิทธิภาพลดลง และไม่สามารถหาคำตอบที่ดีที่สุดได้ ขั้นตอนวิธีแบบค้างคาว มีข้อจำกัดในเรื่องของความเร็วของการลู่เข้าสู่คำตอบลดลง หลังจากรันขั้นตอนวิธีในระยะเริ่มต้นไปแล้ว และขั้นตอนวิธีแบบแมลงหวี่ มีข้อจำกัดในเรื่องของการลู่เข้าสู่

คำตอบช้า

งานวิจัยนี้จึงนำเสนอการหาค่าความเหมาะสมสำหรับขั้นตอนวิธีแบบเพี้ยนกระโดด ซึ่งเป็นการหาค่าความเหมาะสมที่ได้จากการศึกษาพฤติกรรมกรรมการหาอาหารแบบเป็นกลุ่ม โดยการเปรียบเทียบกับขั้นตอนวิธีการค้นหาแบบนกคูหา แบบค้างคาว และแบบแมลงหวี่ ด้วยการใช้ฟังก์ชันทางคณิตศาสตร์ (Numerical Optimization) เพื่อทดสอบประสิทธิภาพในการหาค่าความเหมาะสม

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

จากแนวคิดการแก้ปัญหาทางงานวิจัย มีทฤษฎีที่เกี่ยวข้องดังนี้

2.1 การหาค่าความเหมาะสม (Optimization)

ปัญหาการหาค่าความเหมาะสม สามารถแสดงได้ดังนี้ [12]

กำหนดให้ ฟังก์ชัน $f : A \rightarrow R$ แมปจากค่าบางส่วนในเซต A ไปเป็นค่าจริง ต้องการค้นหาค่า x_0 ในเซต A ซึ่งหากค่า $f(x_0) \leq f(x)$ สำหรับค่า x ทั้งหมดในเซต A เรียกว่าการหาค่าต่ำสุด (Minimization) และหากค่า $f(x_0) \geq f(x)$ สำหรับค่า x ทั้งหมดในเซต A เรียกว่าการหาค่าสูงสุด (Maximization)

การหาค่าความเหมาะสมมีหลายวิธี ดังนี้ ขั้นตอนวิธีค้นหาค่าความเหมาะสมโดยมีพื้นฐานจากวิธีการทางคณิตศาสตร์ ได้แก่ การโปรแกรมเชิงเส้น (Linear programming) ใช้กับปัญหาซึ่งความสัมพันธ์กันระหว่างตัวแปรทุกตัวเป็นแบบเชิงเส้น การโปรแกรมเลขจำนวนเต็ม (Integer programming) เป็นโปรแกรมเชิงคณิตศาสตร์ที่มีข้อจำกัดของตัวแปรว่า จะต้องเป็นเลขจำนวนเต็มที่ไม่ใช่ค่าเป็นลบ การโปรแกรมเชิงกำลังสอง (Quadratic programming) เป็นฟังก์ชันกำลังสองที่มีข้อจำกัดของปัญหาสมการหรืออสมการเชิงเส้น และการโปรแกรมไม่เป็นเส้นตรง (Nonlinear programming) เป็นโปรแกรมเชิงคณิตศาสตร์ที่ใช้หาคำตอบ กรณีที่ตัวแปรตัดสินใจมีความสัมพันธ์กันลักษณะที่ไม่เป็นเส้นตรง จะมีฟังก์ชันอย่างน้อยที่สุด 1 ฟังก์ชันไม่เป็นเส้นตรง เป็นต้น และการหาค่าความเหมาะสมที่มีพื้นฐานจากวิธีการเชิงสามัญสำนึก (Heuristics) ได้แก่ ขั้นตอนวิธีมีมติก (Memetic algorithm) จีเนติก (Genetic Algorithm: GA) ขั้นตอนวิธีหาค่าเหมาะสมแบบฝูงมด (Ant Colony Optimization) ขั้นตอนวิธีแบบหิ่งห้อย (Firefly Algorithm) [13] ขั้นตอนวิธีการค้นหาแบบ

นกคู้เหว้า (Cuckoo Search) ขั้นตอนวิธีแบบค้างคาว (Bat Algorithm) และขั้นตอนวิธีแบบแมลงหวี่ (Fruit Fly Algorithm) เป็นต้น

2.2 ขั้นตอนวิธีการค้นหาแบบนกคู้เหว้า (Cuckoo Search: CS)

เป็นขั้นตอนวิธีที่อาศัยหลักการธรรมชาติการวางไข่ของนกคู้เหว้า โดยจะทำการวางไข่ลงไปในรัง ซึ่งรังที่จะทำการวางไข่จะถูกสุ่มเพื่อเลือกรังที่มีคุณภาพสูง เพราะจะทำให้ตัวอ่อนมีอัตราการรอดชีวิตสูงตามไปด้วย จึงมีผลทำให้ได้นกคุ้นถัดไป และขั้นตอนการบินหารังใหม่นั้นจะใช้หลักการของ Levy Flight ที่มีการเคลื่อนที่ไปได้ไกลกว่าปกติ [9] นำมาประยุกต์ใช้แก้ปัญหาแบบ Optimization อย่างแพร่หลาย เช่น ปัญหาการหาค่าที่เหมาะสมที่สุดเชิงตัวเลข (Numerical optimization problems) เป็นต้น

2.3 ขั้นตอนวิธีแบบค้างคาว (Bat Algorithm: BA)

เป็นขั้นตอนวิธีที่มีแนวคิดจากพฤติกรรมการหาอาหารของค้างคาว ได้ถูกนำเสนอครั้งแรกโดย Xin-She Yang โดยการปล่อยคลื่นเสียงที่มีความถี่สูงของค้างคาว (Sonar) เพื่อใช้ในการนำทางและหาอาหาร (Echolocation) [10]

2.4 ขั้นตอนวิธีแบบแมลงหวี่ (Fruit Fly Optimization Algorithm: FOA)

เป็นขั้นตอนวิธีที่มีแนวคิดจากพฤติกรรมในการค้นหาแหล่งอาหาร ซึ่งเป็นสัตว์ที่มีประสาทสัมผัสในการมองเห็นและรับกลิ่นที่ดีมาก ในการค้นหาแหล่งอาหารจะมีกระบวนการที่สำคัญอยู่ 2 กระบวนการ คือ กระบวนการแรกจะใช้ประสาทสัมผัสในการรับกลิ่นของแหล่งอาหาร และกระบวนการที่สองจะเป็นการใช้ประสาทสัมผัสในการมองเห็น โดยจะบินไปยังแหล่งอาหารเพื่อทำการค้นหาอาหาร [11]

2.5 พฤติกรรมของเพรียกระโดด (Planthopper)

เพรียกระโดดอยู่ในอันดับ Hemiptera มีวงจรชีวิต 3 ระยะ คือ ไข่ ตัวอ่อน และตัวเต็มวัย เป็นแมลงที่สามารถกระโดดได้อย่างรวดเร็ว และพลังมากที่สุดในบรรดาแมลงทั้งหมด กลไกของการกระโดดคล้ายหนังสือที่เคลื่อนไหวอย่างรวดเร็วของขาหลังซ้ายและขวาของตัวมันเอง ประสิทธิภาพในการกระโดด วัดจากระยะทาง (s) ดังสมการที่ 1

$$s = v \cos \theta (2v \sin \theta) / g \quad (1)$$

กำหนดให้ v คือ ความเร็วที่ take-off และ θ คือ มุมที่ใช้ take-off องศา g คือ ความเร่งเนื่องจากความโน้มถ่วง (9.81 m/s^2) [14]

จากสมการที่ 1 นำ $v \cos \theta$ คูณเข้าไปในวงเล็บ จะได้

$$s = (2v^2 \cos \theta \sin \theta) / g \quad (2)$$

เนื่องจาก $2 \cos \theta \sin \theta = \sin 2\theta$

$$s = (v^2 \sin 2\theta) / g \quad (3)$$

3. วิธีดำเนินการวิจัย

3.1 ขั้นตอนวิธีแบบเพรียกระโดด มีขั้นตอนดังต่อไปนี้ [15]

ขั้นตอนแรก กำหนดพารามิเตอร์เริ่มต้น ประกอบด้วย รอบการทำงาน (Maxgen) จำนวนประชากรเพรียกระโดด (Sizepop) มุมที่ใช้ take-off (Angle) และค่า Body mass

ขั้นตอนที่ 2 กำหนดตำแหน่งของเพรียกระโดดเริ่มต้น

ขั้นตอนที่ 3 สร้างคำตอบใหม่โดยการปรับค่า Body Mass ค่าความเร็ว ระยะทาง และตำแหน่งของเพรียกระโดดแต่ละตัว โดยใช้สมการ

$$v = v_{min} + r(v_{min} - v_{max}) \quad (4)$$

โดย v คือ ความเร็วที่ take-off

v_{min} คือ ความเร็วต่ำสุดที่ take-off มีค่าเท่ากับ 0

v_{max} คือ ความเร็วสูงสุดที่ take-off มีค่าเท่ากับ 5

$r \in [0, 1]$ คือเวกเตอร์จากการสุ่ม (Random Vector) ได้มาจากการแจกแจงแบบสม่ำเสมอ (Uniform Distributions) และใช้สมการ (3) โดยกำหนดให้มุมที่ใช้ take-off มีค่าเท่ากับ 45 องศา

การปรับตำแหน่งของเพรียกระโดด

$$x_{new} = (x_{old} - x_{best}) + s \quad (5)$$

กำหนดให้ x_{new} คือ ตำแหน่งใหม่ของเพรียกระโดดที่ได้จากการปรับระยะทาง x_{old} คือตำแหน่งเดิมของเพรียกระโดด x_{best} คือตำแหน่งที่ดีที่สุดแบบ global และ s คือระยะทางที่เกิดขึ้นจากการกระโดด

ขั้นตอนที่ 4 ถ้า r ที่สุ่มขึ้นมีค่ามากกว่า Body mass ให้เลือกคำตอบที่ดีที่สุดตามฟังก์ชันวัตถุประสงค์และสร้างคำตอบเฉพาะที่ (Local Solution) รอบ ๆ คำตอบที่ดีที่สุด

ขั้นตอนที่ 5 การแทนค่าตำแหน่งของเพรียกระโดดลงไปในฟังก์ชันจุดประสงค์

$$F = \text{fitness}(x_{new}) \quad (6)$$

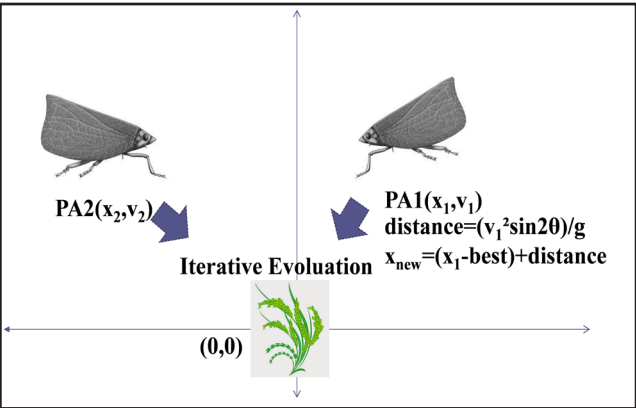
ขั้นตอนที่ 6 คือ การตรวจสอบตำแหน่งของเพรียกระโดดร่วมกับฟังก์ชันจุดประสงค์

$$[fmin\ best] = \min (Fitness) \tag{7}$$

ขั้นตอนที่ 7 คือ การเก็บค่า โดยทำการบันทึกตำแหน่งและคำตอบที่ดีที่สุดไว้

$$\begin{aligned} best &= xnew \\ fmin &= bestFitness \end{aligned} \tag{8}$$

ขั้นตอนที่ 8 คือ การทำซ้ำ โดยเข้าสู่กระบวนการทำซ้ำในขั้นตอนที่ 3 ถึง 7 จนกว่าจะพบค่าที่เข้าใกล้ 0 มากที่สุด ซึ่งเป็นการลู่เข้าสู่คำตอบที่ดีที่สุด



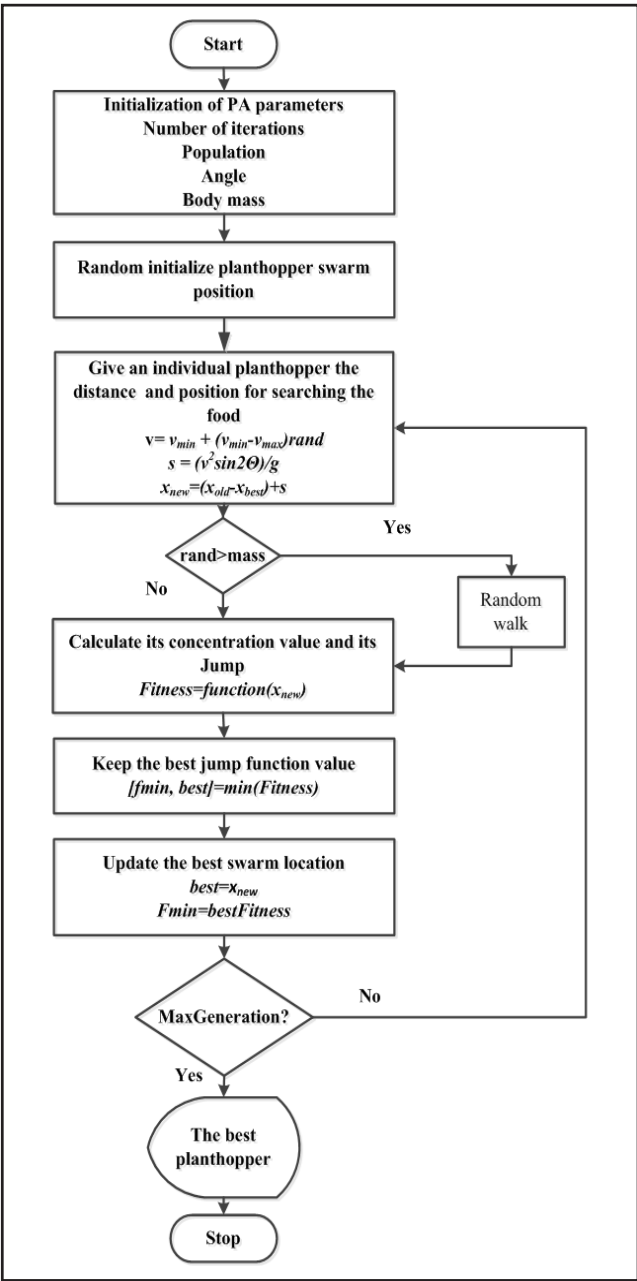
ภาพที่ 1 กระบวนการการค้นหาอาหารของเพรียกระโดด
ขั้นตอนวิธีแบบเพรียกระโดด แสดงดังภาพที่ 2

Planthopper algorithm
Objective function $f(x)$, $x = (x_1, \dots, x_d)^T$
Initialize the planthopper population $x_i = (i = 1, 2, \dots, n)$
Initialize take-off angle, body mass r
While ($t < \text{Max number of iterations}$)
Generate new solutions by adjusting distance, and updating velocities and locations/solutions
If ($\text{rand} > r$)
Select a solution among the best solutions
Generate a local solutions around the selected best solution
end if
Rank the Planthopper and find the current best
End while
Post-process results and visualization

ภาพที่ 3 Pseudo code of the Planthopper Algorithm (PA)

3.2 ทดสอบประสิทธิภาพ

การวัดประสิทธิภาพของขั้นตอนวิธีแบบเพรียกระโดดสามารถวัดจากการใช้ฟังก์ชันทางคณิตศาสตร์จำนวน 4 ฟังก์ชัน คือ ฟังก์ชัน Sphere ฟังก์ชันไกรแวงก์ ฟังก์ชันโรเซนบร็อก และฟังก์ชันแอกเคิลส์ ดังรายละเอียดต่อไปนี้



ภาพที่ 2 ผังงานของ Planthopper Algorithm

3.2.1 ฟังก์ชัน Sphere (Sphere function)

$$f(x) = \sum_{i=1}^n (x_i^2) \tag{8}$$

ฟังก์ชัน Sphere เมื่อ $-100 \leq x_i \leq 100$ สำหรับคำตอบของฟังก์ชันนี้คือ $x = 0$ ที่ทำให้ $f(x) = 0$

3.2.2 ฟังก์ชันไกรแวงก์ (Griewank function)

$$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{9}$$

ฟังก์ชันไกรแวงก์ เมื่อ $-600 \leq x_i \leq 600$ สำหรับคำตอบของฟังก์ชันนี้คือ $x_i = 0$ ที่ทำให้ $f(x_i) = 0$



3.2.3 ฟังก์ชันโรเซนบร็อก (Rosenbrock function)

$$f(x) = \sum_{i=1}^{n-1} [100((x_{i+1} - x_i^2))^2 + (1 - x_i)^2] \quad (10)$$

ฟังก์ชันโรเซนบร็อก เมื่อ $-2.048 \geq x_i \leq 2.048$ สำหรับคำตอบของฟังก์ชันนี้คือ $x_i = 0$ ที่ทำให้ $f(x_i) = 0$

3.2.4 ฟังก์ชันแอกคิลีย์ (Ackley function)

$$f(x) = -20 \cdot \exp \left(-0.2 \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e \quad (11)$$

ฟังก์ชันแอกคิลีย์ เมื่อ $-32.768 \geq x_i \leq 32.768$ สำหรับคำตอบของฟังก์ชันนี้คือ $x_i = 0$ ที่ทำให้ $f(x_i) = 0$

4. ผลการดำเนินงาน

ขั้นตอนวิธีแบบเพี้ยนกระโดดได้รับการพัฒนาด้วยโปรแกรม MATLAB เพื่อทดสอบฟังก์ชันทางคณิตศาสตร์ทั้ง 4 ฟังก์ชัน โดยทำการเปรียบเทียบการลู่เข้าสู่คำตอบกับการค้นหาแบบนกดูเหว่า แบบค้างคาว และแบบแมลงหวี่ตามลำดับ

4.1 กำหนดพารามิเตอร์

กำหนดพารามิเตอร์ที่ใช้ในการทดสอบกับขั้นตอนวิธีทั้ง 4 แบบคือ จำนวนของกลุ่มประชากรเท่ากับ 50 และจำนวนรอบการทำซ้ำเท่ากับ 50 รอบ พารามิเตอร์ตัวอื่นที่มีลักษณะเฉพาะตามขั้นตอนวิธี เช่น ขั้นตอนวิธีการค้นหาแบบนกดูเหว่า ต้องแทนค่าความเป็นไปได้ในการทิ้งรัง = 0.25 และขั้นตอนวิธีค้างคาวต้องกำหนดความถี่ $fmin = 0$ และ $fmax = 5$ ความดังของเสียง = 0.5 และ อัตราการปล่อยพัลส์ (Pulse rate) = 0.5

4.2 ผลการทดสอบ

ส่วนที่ 1 การทดสอบการลู่เข้าสู่คำตอบ เมื่อทำการปรับเปลี่ยนความแรงเนื่องจากความโน้มถ่วง (g)

จากตารางที่ 1 เป็นผลการทดสอบการลู่เข้าสู่คำตอบเมื่อทำการปรับเปลี่ยนความแรงเนื่องจากความโน้มถ่วง (g) พบว่าค่าเฉลี่ยของการลู่เข้าสู่คำตอบมีค่าที่ดี เมื่อปรับเปลี่ยนให้เหมาะสมกับแต่ละฟังก์ชัน โดยฟังก์ชัน Sphere ที่ $g = 9.0$ ได้ค่าความเหมาะสมเฉลี่ยเท่ากับ 0.02377 ฟังก์ชันไกรแวงก์ที่ $g = 4.0$ ได้ค่าความเหมาะสมเฉลี่ยเท่ากับ 0.00001 ฟังก์ชันโรเซนบร็อกที่ $g = 9.0$ ได้ค่าความเหมาะสมเฉลี่ย

ตารางที่ 1 ค่าเฉลี่ยการลู่เข้าสู่คำตอบเมื่อความแรงเนื่องจากความโน้มถ่วง (g) เปลี่ยนไป

	Sphere		Griwank		Rosenbrock		Ackley	
	ค่าเฉลี่ย	STDEV	ค่าเฉลี่ย	STDEV	ค่าเฉลี่ย	STDEV	ค่าเฉลี่ย	STDEV
0.5	0.10854	0.24641	0.00070	0.00405	1.38210	2.25928	0.09660	0.43143
1	0.06413	0.16889	0.00102	0.00439	0.95759	2.20378	0.06171	0.28001
1.5	0.09503	0.19065	0.00019	0.00168	0.82861	2.09647	0.06360	0.30717
2	0.07240	0.17902	0.00113	0.00305	1.02258	4.06328	0.05284	0.27770
2.5	0.09591	0.20692	0.00019	0.00119	0.62219	1.75106	0.04120	0.27962
3	0.05827	0.15608	0.00023	0.00132	0.54842	1.52268	0.01956	0.12986
3.5	0.05729	0.13773	0.00066	0.00249	0.49724	1.40992	0.03267	0.20876
4	0.03640	0.11135	0.00001	0.00013	0.53162	1.49683	0.05132	0.25931
4.5	0.04948	0.12779	0.00016	0.00107	0.47754	1.13981	0.00941	0.09204
5	0.06464	0.18216	0.00003	0.00027	0.49988	1.58484	0.01471	0.08104
5.5	0.03439	0.09932	0.00036	0.00182	0.56244	1.58951	0.04071	0.21119
6	0.03642	0.19165	0.00022	0.00238	0.48886	2.49127	0.01540	0.08975
6.5	0.03035	0.08876	0.00022	0.00145	0.55342	1.49254	0.01205	0.09540
7	0.03927	0.11941	0.00022	0.00121	0.47321	1.09599	0.01876	0.11716
7.5	0.05223	0.12972	0.00018	0.00133	0.43549	1.26436	0.01067	0.07677
8	0.05142	0.13185	0.00027	0.00144	0.47160	1.94053	0.03435	0.22497
8.5	0.03205	0.07640	0.00016	0.00267	0.48841	1.89542	0.02282	0.17902
9	0.02377	0.08258	0.00105	0.00312	0.33930	1.13955	0.02990	0.19710
9.5	0.05513	0.16906	0.00029	0.00286	0.49344	2.14744	0.03398	0.18852
10	0.03673	0.13880	0.00003	0.00022	0.47649	1.56939	0.02241	0.15245

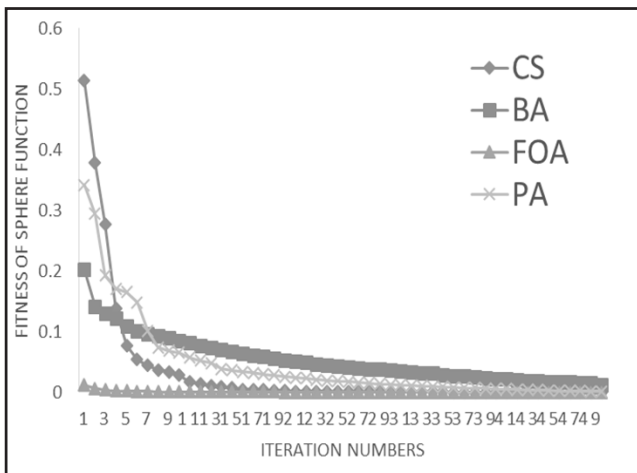
เท่ากับ 0.33930 และฟังก์ชันแอกคิลีย์ที่ $g = 4.5$ ได้ค่าความเหมาะสมเฉลี่ยเท่ากับ 0.00941 ค่าเฉลี่ยของการลู่เข้าสู่คำตอบมีค่าที่ดี เมื่อปรับเปลี่ยนความแรงเนื่องจากความโน้มถ่วงให้เหมาะสมกับแต่ละฟังก์ชัน

ส่วนที่ 2 ผลการทดสอบด้วยฟังก์ชันทางคณิตศาสตร์ทั้ง 4 ฟังก์ชัน โดยกำหนดมุมที่ใช้ในการกระโดดเท่ากับ 45 องศา และปรับเปลี่ยนความแรงเนื่องจากความโน้มถ่วงให้เหมาะสมกับแต่ละฟังก์ชันดังตารางที่ 2

จากตารางที่ 2 เป็นผลการเปรียบเทียบผลลัพธ์ของขั้นตอนวิธี 4 ประเภทได้แก่ ขั้นตอนวิธีการค้นหาแบบนกดูเหว่า แบบค้างคาว แบบแมลงหวี่ และแบบเพี้ยนกระโดดจากตารางจะเห็นว่า ขั้นตอนวิธีแบบเพี้ยนกระโดดให้ค่าความเหมาะสมที่ดีที่สุดในส่วนของฟังก์ชันไกรแวงก์ได้ค่าความเหมาะสมเฉลี่ยเท่ากับ 0.00012 และฟังก์ชันแอกคิลีย์ได้ค่าความเหมาะสมเฉลี่ยเท่ากับ 0.03102 รองลงมาเป็นฟังก์ชันโรเซนบร็อกได้ค่าความเหมาะสมเฉลี่ยเท่ากับ 0.42655 และฟังก์ชัน Sphere ได้ค่าความเหมาะสมเฉลี่ยเท่ากับ 0.04593

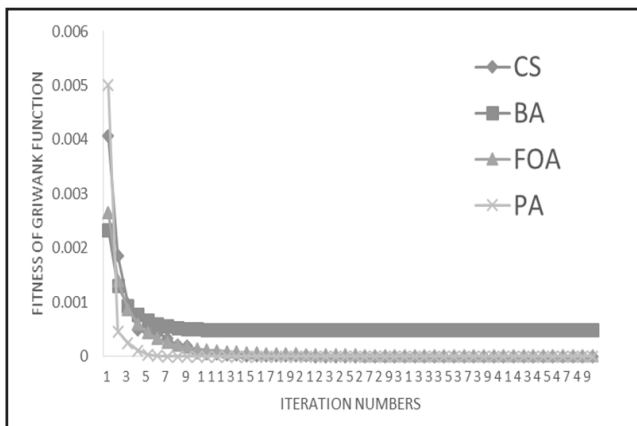
ตารางที่ 2 เปรียบเทียบผลลัพธ์ที่น้อยที่สุดของฟังก์ชันมาตรฐาน

ฟังก์ชัน	ขั้นตอนวิธี	CS	BA	FOA	PA
Sphere	ค่าเฉลี่ย	0.03339	0.05442	0.00071	0.04593
	STDEV	0.14625	0.14604	0.00194	0.15472
Griwank	ค่าเฉลี่ย	0.00019	0.00057	0.00017	0.00012
	STDEV	0.00137	0.00221	0.00044	0.00191
Rosenbrock	ค่าเฉลี่ย	0.33463	0.67159	0.56825	0.42655
	STDEV	1.34329	1.36791	0.43278	0.76182
Ackley	ค่าเฉลี่ย	0.03856	0.03935	0.06726	0.03102
	STDEV	0.15697	0.22294	0.09770	0.16076



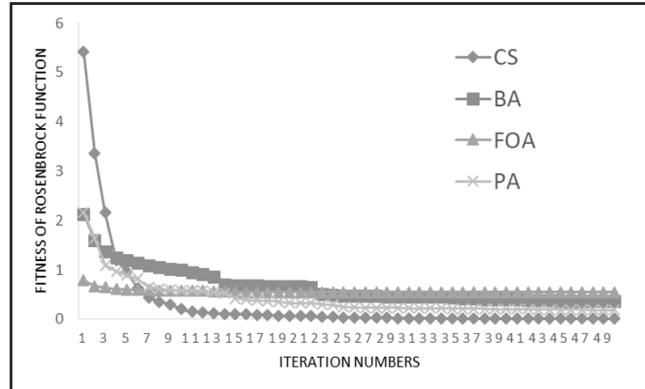
ภาพที่ 4 การเปรียบเทียบประสิทธิภาพของ CS BA FOA และ PA ด้วยฟังก์ชัน Sphere

จากภาพที่ 4 การลู่เข้าสู่คำตอบของฟังก์ชัน Sphere ที่ $g = 9.0$ พบว่า ขั้นตอนวิธีแบบแมลงหวี่สามารถหาค่าความเหมาะสมได้ดีกว่าขั้นตอนวิธีแบบเพี้ยนกระโดด



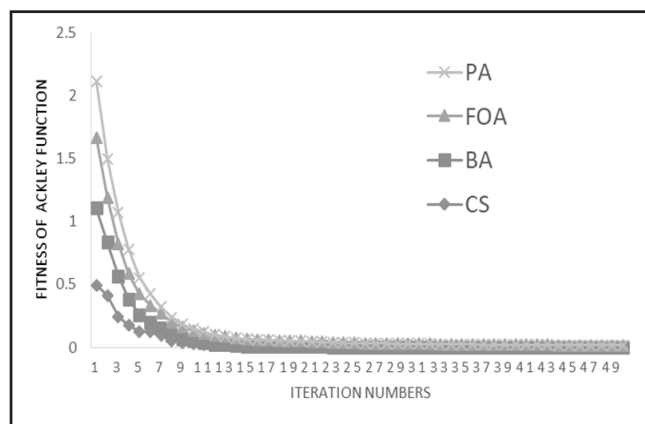
ภาพที่ 5 การเปรียบเทียบประสิทธิภาพของ CS BA FOA และ PA ด้วยฟังก์ชันไกรแวงก์

จากภาพที่ 5 การลู่เข้าสู่คำตอบของฟังก์ชันไกรแวงก์ที่ $g = 4.0$ พบว่า ขั้นตอนวิธีแบบเพี้ยนกระโดดสามารถหาค่าความเหมาะสมได้ดีกว่าขั้นตอนวิธีการค้นหาแบบนกดูเหว่าแบบค้างคาว และแบบแมลงหวี่



ภาพที่ 6 การเปรียบเทียบประสิทธิภาพของ CS BA FOA และ PA ด้วยฟังก์ชันโรเซนบร็อก

จากภาพที่ 6 การลู่เข้าสู่คำตอบของฟังก์ชัน โรเซนบร็อกที่ $g = 9.0$ พบว่า ขั้นตอนวิธีการค้นหาแบบนกดูเหว่าสามารถหาค่าความเหมาะสมได้ดีกว่าขั้นตอนวิธีแบบเพี้ยนกระโดดแบบค้างคาว และแบบแมลงหวี่



ภาพที่ 7 การเปรียบเทียบประสิทธิภาพของ CS BA FOA และ PA ด้วยฟังก์ชันแอกคัลลีย์

จากภาพที่ 7 การลู่เข้าสู่คำตอบของฟังก์ชันแอกคัลลีย์ที่ $g = 4.5$ พบว่า ขั้นตอนวิธีแบบเพี้ยนกระโดดสามารถหาค่าความเหมาะสมได้ดีกว่าขั้นตอนวิธีการค้นหาแบบนกดูเหว่าแบบค้างคาว และแบบแมลงหวี่

5. สรุป

งานวิจัยนี้นำเสนอขั้นตอนวิธีแบบเพลี่ยกระโดด (Planthopper Algorithm: PA) เป็นขั้นตอนวิธีสำหรับการหาค่าความเหมาะสมที่สุดแบบที่ได้รับความนิยมจากพฤติกรรมอาหารแบบเป็นกลุ่ม ซึ่งจากการเปรียบเทียบขั้นตอนวิธีการค้นหาแบบนกดูเหว่า แบบค้างคาว และแบบแมลงหวี่ ด้วยการหาค่าต่ำสุดของฟังก์ชันทางคณิตศาสตร์ 4 ฟังก์ชันคือ ฟังก์ชัน Sphere ฟังก์ชันไกรเวงก์ ฟังก์ชันโรเซนบร็อก และฟังก์ชันแอกคัสซี่ พบว่า ขั้นตอนวิธีแบบเพลี่ยกระโดดสามารถหาค่าความเหมาะสมได้เทียบเคียงกับขั้นตอนวิธีอื่น และดีกว่าขั้นตอนวิธีอื่นในบางฟังก์ชัน

การวิจัยต่อไปในอนาคตจำเป็นต้องเพิ่มประสิทธิภาพการหาค่าความเหมาะสมโดยเพิ่มเติมพารามิเตอร์ เช่น อัตราการตาย จำนวนเพศผู้เพศเมีย การจากฝูงไป การเข้าฝูงใหม่ เป็นต้น

6. เอกสารอ้างอิง

- [1] D. Rai and K. Tyagi. "Bio-inspired optimization techniques: a critical comparative study." *ACM SIGSOFT Software Engineering Notes*, pp. 1-7, 2013.
- [2] P. Chutima, P. Pinkoompee and P. Sanghatawatana. "Application of Memetic algorithms for Multi-Objective Sequencing Problem on Mixed-model Assembly Lines in JIT Production Systems." *In IE Network Conference (IE-Network)*, pp. 72-77, 2007.
- [3] S. Jaochaijaroenkul and D. Puangdownreong. "Novel Metaheuristics Optimization Techniques : Gang Search ." *Proceeding of the SAU National Interdisciplinary Conference 2013*. pp. 233-240, 2013.
- [4] C. Blum, and A. Roli. "Metaheuristics in combinatorial optimization: Overview and conceptual comparison." *ACM Computing Surveys*, pp. 268-308, 2003.
- [5] S. Nootyaskool. "Evolutionary computation between Genetic Algorithm and Particle Swarm Optimization." *Journal of Information Science and Technology*, Vol. 2, No. 2, pp.13-22.
- [6] R. Eberhart and J. Kennedy. "A new optimizer using particle swarm theory." *Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95)*, pp. 39-43, 1995.
- [7] M. Dorigo and G. Di Caro. "Ant colony optimization: a new meta-heuristic." *Proceedings of Evolutionary Computation*, pp. 1470-1477, 1999.
- [8] I. Fister, I. Fister Jr, X.-S. Yang, and J. Brest. "A comprehensive review of firefly algorithms." *Swarm and Evolutionary Computation*, Vol. 13, pp. 34-46.
- [9] X.S. Yang and S. Deb. "Cuckoo search via Levy flights." *World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, pp. 210-214, 2009.
- [10] X.S. Yang. "A new metaheuristic bat-inspired algorithm" *Nature inspired cooperative strategies for optimization (NISCO 2010): Springer*, pp. 65-74, 2010.
- [11] B. Xing and W.J. Gao. "Fruit Fly Optimization Algorithm." *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, Springer International Publishing, pp. 167-170, 2014.
- [12] Mathematical optimization. Available Online at https://en.wikipedia.org/wiki/Mathematical_optimization. accessed on 1 May 2017.
- [13] I. Fister, X.S. Yang, and J. Brest. "A comprehensive review of firefly algorithms." *Swarm and Evolutionary Computation*, pp. 34-46, 2013.
- [14] B. Malcolm. "Jumping mechanisms in flatid planthoppers (Hemiptera, Flatidae)." *The Journal of experimental biology* 217.14, pp. 2590-2600, 2014.
- [15] A. Kaewjai, M. Phayung and N. Vatinee. "Optimization Problems for Planthopper Algorithm." *In Proceeding of National Conference on Computing and Information Technology 2017*, pp. 1-7, 2017.