การประยุกต์ใช้ขั้นตอนวิธีหิ่งห้อยสำหรับค้นหาแผนการทดลองแบบลาตินไฮเปอร์คิวในการจำลองการทดลองด้วยคอมพิวเตอร์

Application of Firefly Algorithm for Searching Latin Hypercube Design in Computer Simulated Experiments

วิศรุต ขวัญคุ้ม[1] จิรารัตน์ เอี่ยมสอาด[2] ธนภัทร เอี่ยมตาล[3]

[1]หลักสูตรวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏวไลยอลงกรณ์ จ.ปทุมธานี
[2]หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏพิบูลสงคราม จ.พิษณุโลก
[3]ฝ่ายบริหารคลัสเตอร์และโปรแกรมวิจัย สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช.) จ.ปทุมธานี
[1]wisrut@vru.ac.th

บทคัดย่อ

งานวิจัยนี้ประยุกต์ใช้ขั้นตอนวิธีหิ่งห้อย (Firefly algorithm: FA) สำหรับค้นหาแผนการทดลองแบบลาตินไฮเปอร์คิวในการจำลองการทดลองด้วยคอมพิวเตอร์ ซึ่งขั้นตอนวิธีดังกล่าวสามารถนำมาประยุกต์ใช้ในการสร้างแผนการทดลองที่เหมาะสม (Optimal design) ได้ดี โดยนำคำตอบที่ได้จากขั้นตอนวิธีหิ่งห้อยมาเปรียบเทียบกับขั้นตอนวิธีสับเปลี่ยนการกระโดดของกบ (Shuffled frog leaping algorithm: SFL) ผลที่ได้การทดลองพบว่าค่าเฉลี่ยของคำตอบในการค้นหาแผนการทดลองที่เหมาะสมมีค่าใกล้เคียงกัน และเมื่อพิจารณาเวลาในการประมวลผลพบว่าขั้นตอนวิธีหิ่งห้อยหรือ FA ใช้เวลาสั้นกว่าวิธีสับเปลี่ยนการกระโดดของกบหรือ SFL ในทุกขนาดปัญหา

คำสำคัญ: การออกแบบลาตินไฮเปอร์คิว ขั้นตอนวิธีการค้นหา ขั้นตอนวิธีหิ่งห้อย การจำลองการทดลองด้วยคอมพิวเตอร์

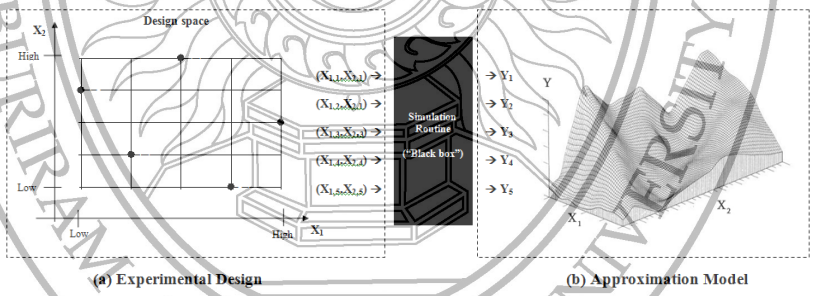
ABSTRACT

This paper applies Firefly algorithm (FA) for searching the optimal design of Latin hypercube experimental plan for computer simulated experiment (CSE). The FA and the Latin hypercube design (LHD) could generate the optimal experimental plan. Then compare the result to Shuffled frog leaping algorithm (SFL). The experimental result shows that the average of the result of optimal design from the FA is approximate to the average of the result of experimental plan from the SFL. However, the FA spend computational time less than the SFL for all problems.

Keywords: Latin hypercube design, Search Algorithm, Firefly algorithm, Computer simulation experiment

1. บทนำ

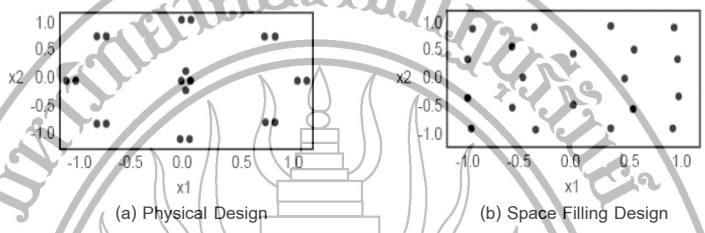
การทำการทดลองส่วนใหญ่ได้มีการประยุกต์ใช้เทคนิคที่เรียกว่า การวางแผนการทดลองเข้ามาช่วยออกแบบในการดำเนินการเก็บรวบรวมข้อมูล เพื่อให้ได้ผลการทดลองที่มีความถูกต้องแม่นยำและเป็นไปตามหลักการวางแผนการทดลอง โดยทั่วไปการวางแผนการทดลองจะประกอบด้วยการกำหนดค่าของตัวแปรอิสระและทำการทดลองจนได้มาซึ่งข้อมูลของค่าตัวแปรตาม ซึ่งสามารถทำการทดลองได้ทั้งการทดลองกายภาพ (Physical experiment) หรือการจำลองการทดลองด้วยคอมพิวเตอร์ (Computer simulated experiment: CSE) ก็ได้ ในกรณีที่ต้องการศึกษาเกี่ยวกับระบบที่มีความซับซ้อนที่ไม่สามารถทำการทดลองทางกายภาพได้ ซึ่งอาจเนื่องมาจากข้อจำกัดต่าง ๆ ดังนั้นการนำคอมพิวเตอร์เข้ามาช่วยคำนวณหรือการจำลองการทดลองขึ้นนั้นสามารถแก้ปัญหาเหล่านี้ได้ ซึ่งกระบวนการจำลองการทดลองด้วยคอมพิวเตอร์ประกอบด้วย 3 ส่วนหลัก คือ 1) Experimental design คือ การวางแผนการทดลอง ซึ่งประกอบด้วยการเลือกค่าของปัจจัยที่เกี่ยวข้องระดับต่าง ๆ ของตัวแปรอิสระ 2) Simulation routine หรือ กล่องดำ (Black box) คือ กระบวนการทำงานที่เราไม่สามารถมองเห็นและคาดการณ์ได้ว่าเกิดอะไรขึ้นภายใน ซึ่งกระบวนการในกล่องดำนี้อาจอยู่ในรูปแบบสมการทางคณิตศาสตร์ที่มีความซับซ้อน และ 3) Approximation model คือ การสร้างโมเดลเพื่อการประมาณ เกิดจากการนำแผนการทดลองกับผลลัพธ์ที่ได้ มาสร้างรูปแบบความสัมพันธ์ที่เหมาะสม เพื่อให้สามารถนำไปใช้ประโยชน์ต่อไปได้โดยไม่ต้องทำการทดลองใหม่



ภาพที่ 1 ส่วนประกอบหลักของการจำลองการทดลองด้วยคอมพิวเตอร์

จากองค์ประกอบทั้ง 3 ส่วนที่กล่าวมานั้นจะเห็นว่า การออกแบบการทดลองที่เหมาะสมมีส่วนสำคัญอย่างยิ่งต่อการได้มาซึ่งแบบจำลองที่ถูกต้องและสามารถใช้เป็นต้นแบบของระบบ (Prototype) ได้ ดังนั้นทฤษฎีและเทคนิคต่าง ๆ เพื่อการออกแบบ CSE ได้ถูกนำเสนอโดยนักวิจัยไว้หลายท่าน โดยหลักการวางแผนการทดลองด้วยคอมพิวเตอร์นั้นจะแตกต่างจากการวางแผนการทดลองทางกายภาพที่เน้นการจัดกลุ่ม (Blocking) และการทำซ้ำ (Replication) รวมไปถึงการสุ่ม (Randomization) ดังแสดงในภาพที่ 2(a) เพื่อให้ได้ค่าผลลัพธ์ที่แม่นยำ แต่การออกแบบการทดลองสำหรับ CSE นั้นจะแตกต่างกันออกไป เนื่องจากผลลัพธ์ที่ได้จากการทดลองด้วยคอมพิวเตอร์จะมีลักษณะเป็นรูปแบบที่แน่นอน (Deterministic) นั่นคือ

เมื่อกำหนดค่าตัวแปรอิสระเหมือนกันทุก ๆ ครั้งก็จะได้ผลลัพธ์ที่เหมือนเดิมเสมอ จึงไม่จำเป็นต้องมีการทำซ้ำเหมือนการทดลองทางกายภาพ ดังนั้นจึงส่งผลให้การวางแผนการทดลองสำหรับ CSE เน้นการออกแบบการทดลองแบบเติมเต็มปริภูมิ (Space filling design) ที่เน้นการกระจายจุดให้ครอบคลุมปริภูมิการทดลองให้มากที่สุดเท่าที่จะเป็นไปได้ เพื่อนำผลลัพธ์ที่ได้ไปสร้างเป็นโมเดลเพื่อการประมาณที่มีประสิทธิภาพ ดังแสดงรูปแบบการออกแบบการทดลองในภาพที่ 2 (b)



ภาพที่ 2 ตัวอย่างแผนการทดลองทางกายภาพ และแบบเติมเต็มปริภูมิ

สำหรับการสร้างแผนการทดลองแบบเติมเต็มปริภูมินั้น โดยทั่วไปนิยมใช้ขั้นตอนวิธีสืบค้น (Search algorithm) เข้ามาช่วยค้นหาการออกแบบการทดลองที่เหมาะสมโดยพิจารณาจากเกณฑ์ในการเลือกค่าเหมาะสม

จากการศึกษางานวิจัยที่เกี่ยวข้องพบว่ามีขั้นตอนวิธีสืบค้นหลายรูปแบบที่ได้นำมาใช้กับการวางแผนการทดลองสำหรับปัญหา CSE บ้างแล้ว ได้แก่ วิธีสโทแคสติกอีโวลูชั่นนารี่ (Stochastic evolutionary : SE) (Jin, R. and others , 2005) ซิมูเลทเทดแอนนีลลิง (Simulated annealing : SA) (Thamma, T., 2008) ขั้นตอนวิธีเจเนติก (Genetic algorithm: GA) (วิศรุต ขวัญคุ้ม, 2560) Enhanced stochastic evolutionary : ESE and Modified enhancement stochastic evolutionary : MESE) (Na-udom, A. and others, 2016) โดยแต่ละวิธีจะมีกลยุทธ์ในการค้นหาคำตอบที่แตกต่างกันออกไป เพื่อให้ได้แผนการทดลองที่เหมาะสมที่สุด สำหรับในงานวิจัยนี้มีวัตถุประสงค์เพื่อทดสอบและประยุกต์ใช้ขั้นตอนวิธีสืบค้นในการค้นหาแผนการทดลองที่เหมาะสมสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ โดยประยุกต์ใช้ขั้นตอนวิธีหิ่งห้อย (Firefly algorithm : FA) ซึ่งจากการค้นคว้ายังไม่พบงานวิจัยก่อนหน้าที่นำมาประยุกต์ใช้กับการค้นหาแผนการทดลองที่เหมาะสมมาก่อน จากนั้นจึงนำผลเฉลยที่ได้มาเปรียบเทียบกับขั้นตอนวิธีสับเปลี่ยนการกระโดดของกบ (Shuffled frog leaping algorithm : SFL)

2. วิธีดำเนินการวิจัย

2.1 การออกแบบลาตินไฮเปอร์คิว (Latin hypercube design: LHD)

แผนการทดลองแบบลาตินลาตินไฮเปอร์คิว หรือ LHD เป็นคลาสการออกแบบที่ได้รับความนิยมกันอย่างแพร่หลายในการออกแบบแผนการทดลองที่เหมาะสมสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ (McKay M.D., 1997) ซึ่ง LHD เป็นเมทริกซ์ (□) ที่มีมิติขนาด □ × □ โดยที่ □ คือจำนวน

รัน และ □ คือ จำนวนตัวแปรอิสระ ซึ่งแต่ละสดมภ์ของแผนการทดลองจะประกอบด้วยการเรียงสับเปลี่ยนแบบสุ่มของสมาชิกตั้งแต่ $\{1,2,\dots,□\}$ และเนื่องจากแผนการทดลองแบบ LHD สามารถสร้างได้โดยใช้แนวคิดของการสุ่มแบบแบ่งชั้นภูมิ (Stratified sampling) ทำให้มั่นใจว่าในทุก ๆ ส่วนย่อยของปริภูมิการทดลองมีการแบ่งตัวแปรอิสระในระดับที่เท่า ๆ กัน และมีโอกาสถูกเลือกทุกระดับและค่าของแต่ละระดับจะไม่มีการคาบเกี่ยวกัน สามารถเขียนได้ดังสมการต่อไปนี้

$$X_{ij} = \frac{\pi_{ij} - U_{ij}}{n} \quad (1)$$

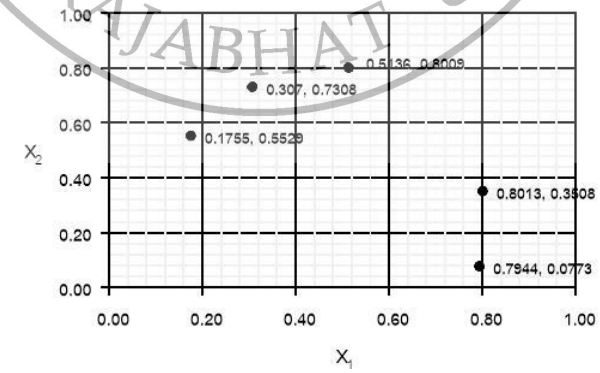
โดยที่ $□_{□□}$ เป็นส่วนของเมทริกซ์ขนาด □ × □ ที่ประกอบด้วยสดมภ์ $□_{□}$ ($□ = 1,2,\dots,□$) โดยแต่ละสดมภ์ $□_{□}$ จะสุ่มเรียงสับเปลี่ยนสมาชิก $1,2,\dots,□$ โดยอิสระต่อกัน และ $□_{□□}$ เมทริกซ์ □ × □ ที่มีสมาชิกจาก i.i.d. uniform $□[0,1]$ และเป็นอิสระกับ $□_{□□}$ จะได้แผนการทดลองในช่วงค่าจริงเป็นดังตัวอย่าง จากสมการ (1) สามารถสร้างแผนการทดลองแบบ LHD ขนาด 5×2 ได้ดังนี้ โดยกำหนดค่า $□_{□□}$ และ $□_{□□}$ ดังนี้

π_{ij} U_{ij}

$$\begin{bmatrix} 1 & 3 \\ 3 & 5 \\ 2 & 4 \\ 4 & 1 \\ 5 & 2 \end{bmatrix} \quad \begin{bmatrix} 0.1225 & 0.2355 \\ 0.4322 & 0.9953 \\ 0.4649 & 0.3461 \\ 0.0280 & 0.6134 \\ 0.9934 & 0.0246 \end{bmatrix}$$

$$\frac{1}{5}\left(\begin{bmatrix} 1 & 3 \\ 3 & 5 \\ 2 & 4 \\ 4 & 1 \\ 5 & 2 \end{bmatrix} - \begin{bmatrix} 0.1225 & 0.2355 \\ 0.4322 & 0.9953 \\ 0.4649 & 0.3461 \\ 0.0280 & 0.6134 \\ 0.9934 & 0.0246 \end{bmatrix}\right) = \begin{bmatrix} 0.1755 & 0.5529 \\ 0.5136 & 0.8009 \\ 0.3070 & 0.7308 \\ 0.7944 & 0.0773 \\ 0.8013 & 0.3508 \end{bmatrix}$$

ดังนั้นแต่ละจุดทดลองของแผนการทดลองแบบ LHD จะมีการกระจายดังภาพที่ 3



ภาพที่ 3 ตัวอย่างแผนการทดลองแบบ LHD ขนาด 5×4 แบบสุ่ม

ในทางปฏิบัติสามารถกำหนดช่วงของแผนการทดลองแบบ LHD ที่คำนวณได้ให้ค่าเป็นช่วงระยะห่างที่เท่ากันระหว่าง 0 ถึง 1 ซึ่งจาก ภาพที่ 3 จะได้เมทริกซ์ของแผนการทดลองเป็นดังนี้

$$\begin{bmatrix} 0.00 & 0.50 \\ 0.50 & 1.00 \\ 0.25 & 0.75 \\ 0.75 & 0.00 \\ 1.00 & 0.25 \end{bmatrix}$$

การค้นหาแผนการทดลองแบบ LHD ที่เหมาะสมนั้นมีความซับซ้อนและใช้เวลานาน ถ้าแผนการทดลองประกอบด้วยตัวแปรอิสระ □ ตัว และจำนวนจุดทดลอง □ จุด จะทำให้มีแผนการทดลองแบบ LHD ที่เป็นไปได้ทั้งหมดจำนวน (□!)□ แบบ ซึ่งเป็นจำนวนที่มีขนาดใหญ่ เช่น เมทริกซ์ข้างต้นที่เป็นการทดลองที่มีตัวแปรนำเข้า 2 ตัวและจำนวนจุดทดลอง 5 จุดจะมีแผนการทดลองที่เป็นไปได้ทั้งหมด 14,400 แบบ และจำนวนจะเพิ่มขึ้นเมื่อ □ และ □ มีขนาดใหญ่ขึ้น ทำให้การสร้างแผนการทดลองที่เหมาะสมสำหรับ CSE จัดเป็นปัญหาเอ็นพีแบบยาก (NP-Hard) จึงได้มีการประยุกต์ใช้ขั้นตอนวิธีสืบค้นเข้ามาใช้กับปัญหาเหล่านี้

2.2 เกณฑ์การเลือกค่าเหมาะสม (Optimality criteria)

งานวิจัยนี้เลือกใช้หลักการของเกณฑ์การเลือกค่าที่เหมาะสมแบบ ϕ_p เนื่องจากเป็นเกณฑ์ที่ถูกนำมาใช้อย่างแพร่หลาย ในการสร้างแผนการทดลองที่เหมาะสมสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ ซึ่งเกณฑ์ ϕ_p ถูกนำเสนอโดย Morris, M. D. และ Mitchell, T. J. ในปี 1995 โดยปรับปรุงจากเกณฑ์ Maximin distance criterion สำหรับแผนการทดลอง □ ระยะทางระหว่างจุดทดลองบนปริภูมิการทดลองแบบยูคลิด (Euclidean distance) เมื่อคำนวณหาระยะทางระหว่างจุดทุกคู่ในแผนการทดลองซึ่งเขียนอยู่ในรูปของเมทริกซ์จัตุรัส □ ให้ทำการเรียงระยะทางระหว่างจุดทั้งหมดจากน้อยไปมาก $(□_1, □_2, \dots, □_□)$ และสร้างดัชนี (Index) $(□_1, □_2, \dots, □_□)$ โดย □ คือ จำนวนของคู่จุดที่แบ่งตามระยะทาง $□_□$ โดยสามารถสรุปได้ดังสมการที่ 2 และแผนการทดลองแบบ LHD ที่ให้ค่า ϕ_p ที่น้อยหมายถึงแผนการทดลองที่ดีกว่าหรือเหมาะสมกว่า

$$\phi_p = \left[\sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{1}{(d_{ij})^p} \right]^{1/p} \qquad (2)$$

2.3 ขั้นตอนวิธีหิ่งห้อย (Firefly algorithm: FA)

ขั้นตอนวิธีหิ่งหอยคือกลุ่มพฤติกรรมของระบบแบบกระจายศูนย์ซึ่งถูกนำมาประยุกต์ใช้ในด้านปัญญาประดิษฐ์ และถูกนำเสนอเป็นครั้งแรกโดย Xin-She Yang ในปี ค.ศ.2013 ซึ่งเป็นขั้นตอนวิธีเลียนแบบการดำรงชีวิตของหิ่งห้อยโดยมีการฉายแสงเพื่อใช้ในการหาคู่ครองหรือหาเหยื่อ ซึ่งได้มีการกำหนดแนวคิดไว้ดังนี้ 1) ไม่มีการแบ่งแยกเพศของหิ่งห้อยโดยหิ่งห้อยแต่ละตัวจะดึงดูดความสนใจหิ่งห้อยตัวอื่นได้โดยไม่คำนึงถึงเพศ 2) ความน่าดึงดูดของหิ่งห้อยโดยหิ่งห้อยตัวที่มีแสงสว่างน้อยกว่าจะเคลื่อนที่ไปหาตัวที่มีแสงสว่างมากกว่า 3) ความสว่างของหิ่งห้อยโดยความสว่างของหิ่งห้อยจะหาได้จากฟังก์ชันวัตถุประสงค์ (Objective function) สำหรับการทำงานของขั้นตอนวิธีหิ่งหอยถูกอธิบายไว้เป็นขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 กำหนดฟังก์ชันวัตถุประสงค์สำหรับปัญหาที่ต้องการหาค่าคำตอบซึ่งแทนด้วย □(□) โดยในงานวิจัยมีฟังก์ชันวัตถุประสงค์คือ แผนการทดลองที่เหมาะสมที่สุดซึ่งจะมีค่า ϕ_p น้อยที่สุด

ขั้นตอนที่ 2 สร้างประชากรเริ่มต้นของหิ่งห้อยด้วยการสุ่มโดยหิ่งห้อยแต่ละตัว □□ แทนคำตอบของปัญหาที่พิจารณา

ขั้นตอนที่ 3 กำหนดค่าความเข้มของแสง □□ ให้กับหิ่งห้อยแต่ละตัว □□ โดยพิจารณาค่า □(□) ของฟังก์ชันวัตถุประสงค์ในการตัดสินใจ ซึ่งแทนความหมายได้ว่าความเข้มของแสงหิ่งห้อยแต่ละตัวคือแผนการทดลองที่เหมาะสมที่สุด

ขั้นตอนที่ 4 กำหนดการดูดซับแสง □ ซึ่งการดูดซับแสงจะมีผลต่อความน่าดึงดูดของหิ่งห้อย

ขั้นตอนที่ 5 ทำการเปรียบเทียบความค่าความเข้มของแสงหิ่งห้อย โดยหิ่งห้อยตัวที่มีความเข้มแสงน้อยกว่า □□ จะเคลื่อนที่เข้าหาตัวหิ่งห้อยที่มีความเข้มของแสงมากกว่า □□ ซึ่งในการปรับปรุงหิ่งห้อยตัวที่มีความเข้มแสงน้อยกว่าให้มีความเข้มแสงที่ดีขึ้นต้องอาศัยสมการ

$$x_i = x_i + \beta_0 e^{-\gamma r^2}\left(x_j - x_i\right) + \alpha\left(rand - 0.5\right) \qquad (3)$$

ขั้นตอนที่ 6 หาค่าความน่าดึงดูดซึ่งสัมพันธ์กับการมองเห็นแสงของหิ่งห้อยตัวอื่น ๆ โดยความดึงดูดจะเปลี่ยนแปลงไปและขึ้นกับระยะทางระหว่างหิ่งห้อยตัวที่ □ กับหิ่งห้อยตัวที่ □ ซึ่งในการคำนวณหาต้องอาศัยสมการ

$$\beta(r) = \beta_0 e^{-\gamma r^2} \qquad (4)$$

ขั้นตอนที่ 7 ทำการคำนวณหาค่าความเข้มแสงของหิ่งห้อยใหม่หรือค่าความเหมาะสมของฟังก์ชันวัตถุประสงค์ เมื่อทำจนครบทุกตัวจึงทำการเรียงลำดับหิ่งห้อยตามค่าความเข้มแสงจากมากไปน้อย และเลือกหิ่งห้อยตัวที่ดีที่สุดซึ่งมีความเข้มแสงมากที่สุดในรอบนั้น ๆ จากนั้นทำซ้ำขั้นตอนที่ 1-6 จนกระทั่งพบเงื่อนไขของการหยุดซึ่งจะได้คำตอบที่ดีสุดออกมา

2.4 ขั้นตอนวิธีชัฟเฟิลฟอร์กลิปปิ้ง (Shuffled frog leaping algorithm: SFL)

ขั้นตอนวิธีชัฟเฟิลฟอร์กลิปปิ้งถูกนำเสนอโดย Eusuff M และคณะ มีแนวคิดมาจากการเลียนแบบพฤติกรรมทางธรรมชาติของกบในการค้นหาอาหาร โดยกระบวนการในการหาคำตอบของ SFL มีหลักการทำงานโดยอาศัยขั้นตอนการคัดสรรเฉพาะที่ในแต่ละกลุ่มของประชากรกบก่อนที่จะนำประชากรกบของทุกกลุ่มมาจัดเรียงกัน ซึ่งการค้นหาในลักษณะดังกล่าวนี้จะช่วยลดการเกิดคำตอบเฉพาะที่ได้ สำหรับขั้นตอนการทำงานของ SFL มีดังนี้

ขั้นตอนที่ 1 สร้างประชากรกบด้วยการสุ่มค่า ซึ่งกบแต่ละตัวเป็นตัวแทนของผลเฉลยของปัญหาที่พิจารณา

ขั้นตอนที่ 2 นำกบทั้งหมดแทนลงในฟังก์ชันเป้าหมายเพื่อคำนวณหาค่าความเหมาะสมของกบแต่ละตัว จากนั้นทำการเรียงลำดับกบตามค่าความเหมาะสมที่ได้จากน้อยไปมากในกรณีของปัญหาการหาค่าน้อยสุด

ขั้นตอนที่ 3 แบ่งกบออกเป็น m กลุ่มย่อย โดยเรียกกลุ่มย่อยที่ได้นี้ว่ามีมีเพล็กซ์ (Memeplex) จากนั้นนำกบมาใส่ในแต่ละมีมีเพล็กซ์โดยให้กบตัวแรกใส่ในตำแหน่งแรกของมีมีเพล็กซ์ที่ 1 ให้กบตัวที่ 2 ใส่ในตำแหน่งแรกของมีมีเพล็กซ์ที่ 2 ทำในลักษณะนี้ไปจนถึงกบตัวที่ m ซึ่งจะใส่ในตำแหน่งแรกของมีมีเพล็กซ์ที่ m ซึ่งเป็นมีมีเพล็กซ์สุดท้าย จากนั้นให้กบตัวที่ m+1 ใส่ในตำแหน่งที่ 2 ของมีมีเพล็กซ์ที่ 1 ให้กบตัวที่ m+2 ใส่ในตำแหน่งที่ 2 ของมีมีเพล็กซ์ที่ 2 ทำในลักษณะเช่นนี้ไปเรื่อย ๆ จนครบจำนวนกบทุกตัว

ขั้นตอนที่ 4 ให้ $\square_{\square}$ และ $\square_{\square}$ แทนกบที่มีค่าความเหมาะสมมากสุดและน้อยสุดตามลำดับในมีมีเพล็กซ์ที่ $\square$ และให้ $\square_{\square}$ แทนกบที่มีค่าความเหมาะสมที่มากสุดในบรรดากบทุกตัว นั่นคือ $\square_{\square} = \square_{\square}$

ขั้นตอนที่ 5 เข้าขั้นตอนกลไกการสืบค้นโดยการปรับตำแหน่งของกบซึ่งคำนวณได้จากสมการ (5) จากนั้นนำค่า $\square_{\square}$ ที่ได้ปรับค่าตำแหน่ง (Position) ด้วยสมการ (6)

$$D_i = Rand(\ ) \times \left(X_b^i - X_w^i\right) \qquad (5)$$

$$\text{New position of } \tilde{X}_w = \text{Current position of } X_w + D_i \qquad (6)$$

ขั้นตอนที่ 6 นำกบในทุกมีมีเพล็กซ์มาจัดเรียงกันใหม่ตามค่าความเหมาะสมดังในขั้นตอนที่ 2 แล้วทำการวนซ้ำตามขั้นตอนที่ 3 ถึงขั้นตอนที่ 6 จนพบเงื่อนไขการหยุด

3. ผลการวิจัย

งานวิจัยนี้ได้กำหนดประชากรเท่ากับ 50 ตัว โดยมีเงื่อนไขการหยุดวนซ้ำไม่เกิน 100 ครั้งหรือแผนการทดลองที่ดีที่สุด (Global best) ที่ได้ไม่ดีขึ้นใน 10 รอบติดต่อกัน นำมาทดสอบกับแผนการทดลองที่มีขนาดแตกต่างกันโดยใช้หลักการของการสร้างโมเดลประมาณค่าตามแบบพหุนามกำลังสอง (Second order polynomial) ดังตารางที่ 1 โดยกำหนดให้ □ คือ จำนวนตัวแปรเข้าในการทดลอง และ □ คือ จำนวนรัน ซึ่งสามารถคำนวณได้จากสมการต่อไปนี้

$$n = 2d + 4\binom{d}{2} + 1 \qquad (7)$$

ตารางที่ 1 แสดงจำนวนชุดการสร้างแผนการทดลอง

□	2	4	6	8	10
□	9	33	73	129	201

ผลการทดสอบประสิทธิภาพของขั้นตอนวิธี FA ซึ่งพิจารณาค่า ϕ_p และค่าทางสถิติในการวิเคราะห์ข้อมูลคือค่าเฉลี่ย (Mean) ค่าเบี่ยงเบนมาตรฐาน (Standard deviation: SD) โดยจะทำงานซ้ำแต่ละกรณีในตาราง 2 จำนวน 30 ครั้ง ซึ่งผลการทดลองที่ได้ถูกแสดงไว้ในตารางที่ 2 พบว่าขั้นตอนวิธี FA สามารถนำมาประยุกต์ใช้กับปัญหา CSE ได้เป็นอย่างดี และเมื่อพิจารณาคำตอบที่ได้ในแต่ละปัญหาเมื่อใช้ขั้นตอนวิธี FA เทียบกับ SFL พบว่าไม่แตกต่างกันอย่างมีนัยสำคัญที่ 0.05 ในแง่ของค่าเฉลี่ยของคำตอบที่ได้โดยใช้หลักการสถิติวิเคราะห์ด้วยวิธีการทดสอบแบบ t-test และเมื่อพิจารณาเวลาในการประมวลผลพบว่าขั้นตอนวิธี FA ใช้เวลาสั้นกว่า SFL ของทุกปัญหา

ตารางที่ 2 แสดงค่า ϕ_p ในเชิงสถิติ

LHD	Algorithm	ϕ_p			
		Mean	SD	Time (sec.)	P-Value
9×2	FA	4.5070	0.1360	4.87	0.1320
	SFL	4.5105	0.1377	6.07	
33×4	FA	5.0151	0.1196	266.52	0.1185
	SFL	4.9825	0.1053	307.58	
73×6	FA	6.3436	0.0712	2998.25	0.1058
	SFL	6.3505	0.0885	3225.65	
129×8	FA	6.4952	0.0523	21839.52	0.1958
	SFL	6.5185	0.4286	24742.45	
201×10	FA	6.5863	0.0258	42530.41	0.1455
	SFL	6.5732	0.0352	45052.28	

4. อภิปรายผลและสรุปผล

งานวิจัยนี้เป็นการนำขั้นตอนวิธีหิ่งห้อย มาประยุกต์ใช้สำหรับปัญหาในการค้นหาแผนการทดลองแบบลาตินลาตินไฮเปอร์คิว เพื่อสร้างแผนการทดลองที่เหมาะสมสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ โดยขั้นตอนวิธีที่นำเสนอสามารถนำมาประยุกต์ใช้ในการสร้างแผนการทดลองที่เหมาะสมได้ดี และเมื่อพิจารณาค่า ϕ_p จากผลการทดลองพบว่าค่าเฉลี่ยคำตอบที่ได้ในแต่ละปัญหาเมื่อใช้ขั้นตอนวิธีหิ่งห้อยเทียบกับขั้นตอนวิธีชัฟเฟิลฟอร์กลิปปิ้ง พบว่าไม่แตกต่างกันอย่างมีนัยสำคัญที่ 0.05 และเมื่อพิจารณาเวลาที่ใช้ในการประมวลผลพบว่าขั้นตอนวิธี FA ใช้เวลาสั้นกว่าขั้นตอนวิธี SFL ในทุกปัญหา เนื่องจากขั้นตอนวิธีของ FA มีขั้นตอนกระบวนการปรับปรุงคำตอบที่ซับซ้อนน้อยกว่าจึงส่งผลให้ใช้เวลาในการประมวลผลสั้นกว่าเมื่อเทียบกับขั้นตอนวิธี SFL สามารถสรุปได้ว่าขั้นตอนวิธีขั้นตอนวิธีหิ่งห้อยสามารถนำมาประยุกต์ใช้สำหรับปัญหาในการค้นหาแผนการทดลองแบบลาตินไฮเปอร์คิวได้อย่างมีประสิทธิภาพ ซึ่งหากนำขั้นตอนวิธีหิ่งห้อยไปใช้ในสร้างแผนการทดลองแบบ LHD ในรูปแบบอื่น ๆ เช่น Symmetric Latin hypercube design (SLHD) หรือ Orthogonal Latin hypercube design (OLHD) และหากมีการเพิ่มขนาดมิติของแผนการทดลองให้มีขนาดใหญ่ขึ้นก็จะทำให้ขอบเขตในการสรุปผลนั้นกว้างขึ้นด้วย

เอกสารอ้างอิง

วิศรุต ขวัญคุ้ม. (2560). **การเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึมการค้นหาสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์.** การประชุมวิชาการระดับชาติ วิทยาศาสตร์และเทคโนโลยีระหว่างสถาบัน ครั้งที่ 5 (ASTC2017), 1445-1450.

Eusuff M, Lansey K, Pasha F. (2006) **Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization.** Engineering Optimization, 38(2), 129-154.

Jin, R.; Chen, W. and Sudjianto, A. (2005). **An efficient algorithm for constructing optimal design of computer experiment.** Journal of Statistical Planning and Inference, 134, 268-287.

McKay M.D., Beckman R.J., and Conover W.J. (1997). **A Comparison of Three Methods of Selecting Values of Input Variables in the Analysis of Output from a Computer Code.** Technometrics, 21(2): 239-245.

Morris, M. D. and Mitchell, T. J. (1995). **Exploratory designs for computational experiment.** Journal of Statistical Planning and Inference, 43, 381-402.

Na-udom, A. & Rungrattanaubol, J. (2016). **Heuristics search algorithms for constructing optimal latin hypercube designs.** Switzerland: Springer.

Simpson, T.W.; Lin, D. K. and Chen, W. (2001). **Sampling strategies for computer experiments: Design and analysis.** International Journal of Reliability and Applications, 2(3), 209-240.

Thamma, T., Rungrattanaubol, J., and Na-udom, A. (2008). **Modification on Search Algorithm for Computer Simulated Experiment**. Proceedings of national operations research conference, 117-123.

Yang, X. S., and He, X. (2013). **Firefly algorithm: recent advances and applications.** International Journal of Swarm Intelligence, 1(1), 36-50.