การสร้างแผนการทดลองแบบละตินไฮเปอร์คิวบ์ที่เหมาะสมโดยใช้อัลกอริทึมการค้นหาเฉพาะที่แบบวนซ้ำ

ทนงศักดิ์ บุตรวงศ์[1], จรัสศรี รุ่งรัตนาอุบล[2] และอนามัย นาอุดม*[3]
มหาวิทยาลัยนเรศวร 99 หมู่ 9 ตำบลท่าโพธิ์ อ.เมือง จ.พิษณุโลก 65000

บทคัดย่อ

การจำลองการทดลองด้วยคอมพิวเตอร์ถูกนำมาใช้อย่างแพร่หลายในงานวิจัยด้านต่าง ๆ เพื่อศึกษารูปแบบความสัมพันธ์ระหว่างตัวแปรเข้าและตัวแปรตอบสนอง ไม่ว่าจะเป็นงานวิจัยพื้นฐานรวมไปถึงงานวิจัยประยุกต์ เช่น งานวิจัยด้านวิศวกรรมศาสตร์ ด้านวิทยาศาสตร์ และด้านปิโตรเคมี เป็นต้น การออกแบบการทดลองจัดเป็นปัจจัยที่สำคัญอย่างยิ่งต่อความสำเร็จของงานวิจัยต่าง ๆ เหล่านี้ โดยแผนการทดลองหนึ่ง ๆ จะประกอบด้วยตัวแปรเข้าจำนวน d ตัวแปรและจำนวนจุดทดลองหรือจำนวนรันเท่ากับ n รัน ซึ่งจะเรียกว่ามิติของแผนการทดลองขนาด $(n \times d)$ โดยเมื่อมิติมีขนาดใหญ่ขึ้นจะทำให้แผนการทดลองที่เป็นไปได้ทั้งหมดมีจำนวนมากมายมหาศาล $(n!)^d$ ทำให้การสร้างแผนการทดลองที่เหมาะสมจำเป็นต้องใช้อัลกอริทึมการค้นหาควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสม งานวิจัยนี้ผู้วิจัยประยุกต์ใช้อัลกอริทึมการค้นหาเฉพาะที่แบบวนซ้ำ (Iterated local search algorithm: ILS) ร่วมกับเกณฑ์เลือกค่าความเหมาะสมแบบ ϕ_p เพื่อค้นหาแผนการทดลองแบบละตินไฮเปอร์คิวบ์ (Latin Hypercube Design) ที่เหมาะสมภายใต้มิติของการทดลองที่แตกต่างกัน ผลที่ได้จากการศึกษาพบว่าอัลกอริทึม ILS สามารถค้นหาแผนการทดลองที่เหมาะสมได้อย่างมีประสิทธิภาพสำหรับทุกมิติปัญหาที่ศึกษา

คำสำคัญ: การจำลองการทดลองด้วยคอมพิวเตอร์, อัลกอริทึมการค้นหาเฉพาะที่แบบวนซ้ำ, แผนการทดลองละตินไฮเปอร์คิวบ์, เกณฑ์เลือกค่าความเหมาะสม ϕ_p

*Corresponding author. E-mail: anamain@nu.ac.th

[1]นิสิตปริญญาโท ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

[2]ผู้ช่วยศาสตราจารย์ ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

[3]ผู้ช่วยศาสตราจารย์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

Construction of the optimal Latin hypercube designs using the iterated local search algorithm

Tanongsak Bootwong[1], Jaratsri Rungrattanaubol[2] and Anamai Na-udom[3]

Naresuan University, 99 Moo 9 T. Tapoh A. Mueang Phitsanulok 65000, Thailand

Abstract

Computer simulated experiments have been extensively used to investigate the relationship between input variables and output response in various fields, including natural science and applied sciences such as engineering and petrochemicals. In the context of computer simulated experiments, an experimental design plays a very important role in the success of the simulation. The design consists of d input variables and n experimental runs. Hence the design is referred to as a dimension of experiment ($n \times d$) in which a dimension of the design is large, and the number of all possible experimental designs increases exponentially $(n!)^d$. Therefore the optimal design is usually obtained by using a search algorithm along with the pre-specified optimality criteria. This paper applies the iterated local search algorithm (ILS) along with ϕ_p optimality criteria to construct the optimal Latin hypercube design (LHD). The results indicate that ILS is able to effectively construct the optimal Latin hypercube design for any dimension of the problem under study.

Keywords: Computer simulated experiments, Iterated local search algorithm, Latin hypercube design, ϕ_p optimality criterion

*Corresponding author. E-mail: anamain@nu.ac.th

[1]Master student in Department of Computer Science and IT, Faulty of Science, Naresuan University

[2]Assistant Professor in Department of Computer Science and IT, Faulty of Science, Naresuan University

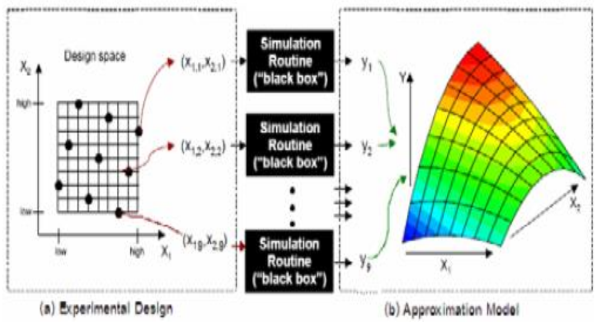
[3]Assistant Professor in Department of Mathematics, Faulty of Science, Naresuan University

1. บทนำ

ในการศึกษารูปแบบความสัมพันธ์ระหว่างตัวแปรเข้า (Input variables) และตัวแปรตอบสนอง (Output response) ที่เกิดขึ้นในระบบที่มีความซับซ้อนซึ่งไม่สามารถทำการทดลองทางกายภาพ (Physical experiments) ได้ เนื่องจากข้อจำกัดด้านวัตถุทดลอง รวมไปถึงความเสี่ยงที่จะเกิดขึ้นต่อมนุษย์และสิ่งแวดล้อม เช่น การทดลองเกี่ยวกับกัมมันตภาพรังสี หรือ การขุดเจาะน้ำมันเป็นต้น [1] ดังนั้นการจำลองการทดลองด้วยคอมพิวเตอร์ (Computer simulated experiment : CSE) ซึ่งเป็นเทคนิคที่ใช้วิธีเชิงคำนวณ (Numerical method) ผ่านตัวแบบทางคณิตศาสตร์จึงถูกนำมาใช้อย่างแพร่หลายในการศึกษารูปแบบความสัมพันธ์ระหว่างตัวแปรเข้าและตัวแปรตอบสนองที่เกิดขึ้นภายในระบบเหล่านี้ [2] ดังนั้นคุณภาพและความน่าเชื่อถือของผลลัพธ์ที่ได้จากการจำลองการทดลองด้วยคอมพิวเตอร์จะขึ้นอยู่กับการเลือกกลุ่มของตัวแปรเข้าหรือการออกแบบการทดลอง (Design of experiments) เพื่อทำการประมวลผลผ่านกระบวนการจำลอง (Simulation routine) เพื่อให้ได้ค่าของตัวแปรตอบสนองนั่นเอง

การจำลองการทดลองด้วยคอมพิวเตอร์ หรือ CSE มีความแตกต่างจากการทดลองทางกายภาพซึ่งต้องมีการทำซ้ำ (Replication) จุดทดลอง เพื่อประมาณค่าความคลาดเคลื่อนที่เกิดจากการทดลอง (Experimental error) แต่การทำงานของ CSE จะเป็นการทำงานแบบที่แน่นอน (Deterministic) กล่าวคือเมื่อทำการทดลอง ณ จุดทดลอง 2 จุดใด ๆ ที่มีค่าของตัวแปรเข้าที่เหมือนกันผ่านการจำลองด้วยคอมพิวเตอร์ จะให้ผลลัพธ์ที่มีค่าเท่ากันเสมอ [3] ดังนั้นจึงส่งผลให้การออกแบบการทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ไม่จำเป็นต้องทำซ้ำจุดทดลอง แต่จะพยายามเน้นการกระจายจุดทดลองให้ครอบคลุมพื้นที่ให้มากที่สุดเท่าที่จะเป็นไปได้ คลาสของการออกแบบการทดลอง (Class of design) ที่ได้รับความนิยมมากในงานด้าน CSE คือ แผนการทดลองแบบละตินไฮเปอร์คิวบ์ (Latin hypercube design: LHD) ซึ่งถูกนำเสนอโดย [4] ต่อมา [5] ได้อธิบายการจำลองการทดลองด้วยคอมพิวเตอร์โดยแบ่งขั้นตอนการทำงานออกเป็น 3 ส่วนดังรูปที่ 1

1) Experimental design คือการออกแบบการทดลองหรือการคัดเลือกจุดทดลองในที่นี้คือการกำหนดค่าของแต่ละตัวแปรต้น (Input variable) ในการทดลอง ซึ่งจากรูปที่ 1 คือ ตัวแปร x_1 และ x_2 โดยในภาพได้คัดเลือกจุดทดลองทั้งหมด 9 จุด โดยพยายามกระจายจุดทดลองให้ครอบคลุมปริภูมิการทดลองให้มากที่สุด

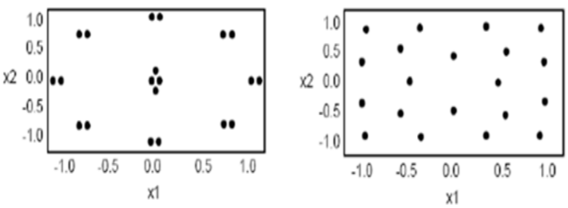


รูปที่ 1 ส่วนประกอบของการจำลองการทดลองด้วยคอมพิวเตอร์ดัดแปลงจาก [5]

2) Simulation routine หรือ Black box คือกระบวนการทำงานที่เราไม่สามารถรับรู้ได้ว่าข้างในนั้นทำงานอย่างไร โดยขั้นตอนในการประมวลผลในกล่องดำส่วนใหญ่นั้นจะอยู่ในรูปแบบของสมการคณิตศาสตร์ที่มีความซับซ้อนและเป็นโปรแกรมคอมพิวเตอร์ระดับสูงที่ถูกสร้างขึ้นจากผู้เชี่ยวชาญทางด้านนั้น ๆ ที่ได้ศึกษาและพัฒนาขึ้น โดยผลลัพธ์ที่ได้มาจากการทำงานของกล่องดำนั้นเกิดจากการใส่ค่าตัวแปรต้นเข้าไปเพื่อให้ได้ผลลัพธ์หรือตัวแปรตาม เพื่อนำตัวแปรตามไปพัฒนาเป็นโมเดลเพื่อการประมาณที่มีประสิทธิภาพต่อไป

3) Approximation model คือ การสร้างโมเดลเพื่อการประมาณจากการนำแผนการทดลองกับผลลัพธ์ที่ได้หลังจากกระบวนการใน Simulation routine หรือ Black box มาสร้างรูปแบบความสัมพันธ์ที่เหมาะสมเพื่อให้สามารถนำไปใช้ประโยชน์ในการพยากรณ์ต่อไป โดยจุดมุ่งหมายหลักของการสร้างโมเดลเพื่อการประมาณ คือ การให้ได้มาซึ่งโมเดลที่สามารถพยากรณ์ผลลัพธ์หรือตัวแปรตอบสนองได้โดยที่ไม่จำเป็นต้องส่งจุดทดลองไปประมวลผลที่ Simulation routine หรือ Black box อีกครั้งจึงทำให้ประหยัดงบประมาณและสามารถหลีกเลี่ยงความเสี่ยงของการเกิดอันตรายจากการทดลองได้ [6]

โดยสรุปแล้วการออกแบบการทดลองเพื่อจำลองการทดลองด้วยคอมพิวเตอร์จะเป็นลักษณะของการกระจายจุดการทดลองเพื่อให้ครอบคลุมปริภูมิการทดลองให้มากที่สุด หรือเรียกว่าแผนการทดลองแบบเติมเต็มปริภูมิการทดลอง (Space filling design) ดังรูปที่ 2(b) ซึ่งจะมีความแตกต่างจากการทดลองแบบกายภาพที่ต้องมีการทำการทดลองซ้ำ (รูปที่ 2(a))



(a) แบบกายภาพ (b) แบบเติมเต็มปริภูมิ

รูปที่ 2 แผนการทดลองแบบกายภาพและแบบเติมเต็มปริภูมิการทดลอง [5]

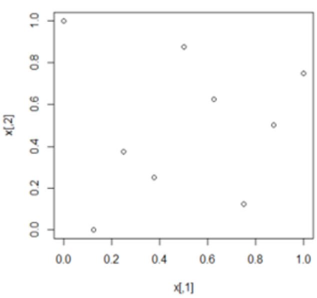
กลุ่มการออกแบบแผนการทดลองหรือคลาสการออกแบบแผนการทดลอง (Class of design) สำหรับ CSE สามารถทำได้โดยการประยุกต์ใช้อัลกอริทึมการค้นหาแบบต่าง ๆ ควบคู่กับเกณฑ์เลือกค่าเหมาะสม (Optimality criteria) เพื่อค้นหาแผนการทดลองที่เหมาะสมที่สุด (Optimal designs) ในคลาสภายใต้มิติของปัญหาที่กำหนดขึ้น ยกตัวอย่างเช่นงานวิจัยของ [7] ที่ได้พัฒนาอัลกอริทึมการค้นหาชื่อว่า Simulated annealing (SA) ภายใต้เกณฑ์การเลือกค่าเหมาะสม ϕ_p เพื่อค้นหาแผนการทดลอง LHD ที่เหมาะสม จากนั้น [8] ได้เสนอวิธีการค้นหาแผนการทดลองที่เหมาะสมที่สุดโดยใช้อัลกอริทึม CP ภายใต้เกณฑ์ Integrated mean square error (IMSE) และเกณฑ์เอ็นโทรพี (Entropy criteria) ต่อมา [9] ได้ดัดแปลงอัลกอริทึมวิวัฒนาการแบบสุ่ม (Enhanced stochastic evolutionary algorithm) เพื่อทำการค้นหาแผนการทดลองที่เหมาะสมโดยพิจารณาเกณฑ์ที่หลากหลาย เช่น เกณฑ์ Maximin distance เกณฑ์ ϕ_p และเกณฑ์ Entropy ในปีถัดมา [10] ทำการประยุกต์อัลกอริทึมแบบเจเนติก (Genetic algorithm: GA) เพื่อทำการค้นหาแผนการทดลองที่ดีที่สุด ภายใต้เกณฑ์ ϕ_p และ เกณฑ์ Maximin จากงานวิจัยของ [11] ที่ได้ทำการเปรียบเทียบประสิทธิภาพของ SA และ GA ในแง่ของความเร็วในการลู่เข้าสู่ค่าที่เหมาะสม (Rate of convergence) และพบว่า SA สามารถทำงานได้ดีกว่า GA เกือบทุกกรณีศึกษา นอกจากนี้ [12] เสนอการสร้างแผนการทดลองแบบ LHD โดยพิจารณาเกณฑ์ Maximin distance และเกณฑ์ Minimum pair-wise correlation ควบคู่กันโดยใช้อัลกอริทึมที่พัฒนาขึ้นมาใหม่และเปรียบเทียบผลที่ได้กับงานวิจัยที่ได้รายงานไว้ใน [7] พบว่าประสิทธิภาพของแผนการทดลองมีคุณสมบัติที่ดีเหมือนกัน นอกจากนี้ [13] ได้เสนออัลกอริทึมสำหรับสร้างแผนการทดลอง LHD ที่มีมิติใหญ่ขึ้นเมื่อเทียบกับงานวิจัยก่อนหน้า นอกจากนี้ยังพบว่าวิธีที่นำเสนอขึ้นมาก่อให้เกิดแผนการทดลองที่มีคุณสมบัติที่ดีมากในแง่ของความเติมเต็มปริภูมิและคุณสมบัติเชิงตั้งฉาก จากนั้น [14] เสนอการวางแผนการทดลองแบบ LHD เชิงตั้งฉากโดยพยายามทำให้ค่าสหสัมพันธ์ระหว่างคอลัมน์ของแผนการทดลองมีค่าน้อยที่สุด ซึ่งเป็นการขยายงานที่นำเสนอไว้ใน [15] เพื่อทำให้สามารถสร้างแผนการทดลองที่มีมิติใหญ่ขึ้นได้ ส่วน [16] เสนอการสร้างแผนการทดลองแบบ LHD ที่มีค่า pair-wise correlation ให้ลู่เข้าสู่ 0 ส่วนงานวิจัยของ [17] ได้นำเสนออัลกอริทึมในการสร้างแผนการทดลองที่เหมาะสมโดยพิจารณาเกณฑ์ ϕ_p นอกจากนี้ [18] ได้เสนอวิธีสร้างแผนการ

ทดลองแบบ LHD ที่เหมาะสมโดยใช้ อัลกอริทึมการค้นหาเฉพาะที่แบบวนซ้ำ (Iterated local search algorithm: ILS) ซึ่งผลที่ได้พบว่าวิธีการดังกล่าวสามารถทำงานได้ดีเมื่อเปรียบเทียบกับงานที่เสนอไว้ใน [7] และ [9] แต่อย่างไรก็ตามการศึกษาดังกล่าวได้เสนอเฉพาะกรณีที่มิติปัญหามีขนาดเล็กเท่านั้น ดังนั้นงานวิจัยนี้จึงเน้นการขยายขอบเขตของการศึกษาและปรับปรุงประสิทธิภาพของ ILS ให้ดียิ่งขึ้น โดยเน้นคลาสการออกแบบ LHD โดยพิจารณาเกณฑ์เลือกค่าเหมาะสมแบบ ϕ_p เพื่อค้นหาแผนการทดลองที่เหมาะสม

2. ทฤษฎีที่ใช้ในงานวิจัย

2.1 คลาสการออกแบบละตินไฮเปอร์คิวบ์

แผนการทดลองแบบละตินไฮเปอร์คิวบ์ หรือ LHD คือการออกแบบการทดลองที่เน้นการกระจายจุดทดลองให้ครอบคลุมปริภูมิของการทดลองให้มากที่สุด โดยในแผนการทดลองหนึ่ง ๆ จะประกอบด้วย n จุดทดลองหรือรัน และแต่ละจุดทดลองจะประกอบด้วยตัวแปรเข้า d ตัว ซึ่งสามารถมองเป็นเมทริกซ์ที่มีมิติเท่ากับ $n \times d$ ได้ แผนการทดลองแบบ LHD จะมีลักษณะสำคัญคือ ค่าที่กำหนดให้ตัวแปรแต่ละคอลัมน์จะต้องมีค่าไม่ซ้ำกัน ดังแสดงในรูปที่ 3



$$\longrightarrow \begin{bmatrix} 0.500 & 0.875 \\ 0.750 & 0.125 \\ 0.875 & 0.500 \\ 1.000 & 0.750 \\ 0.125 & 0.000 \\ 0.000 & 1.000 \\ 0.375 & 0.250 \\ 0.625 & 0.625 \\ 0.250 & 0.375 \end{bmatrix}$$

รูปที่ 3 ตัวอย่างแผนการทดลอง LHD ขนาด *9 x 2*

รูปที่ 3 แสดงตัวอย่างการกระจายจุดทดลองกรณีจำนวนตัวแปรเข้าที่สนใจศึกษาเท่ากับ 2 ตัวแปร และต้องการทำการทดลองทั้งหมด 9 รัน โดยรูปทางด้านขวามือแสดงเมทริกซ์ของการทดลอง *9 x 2* ที่สอดคล้องกับแผนการทดลองทางด้านซ้าย จะเห็นได้ว่าค่าของสมาชิกในแต่ละคอลัมน์ในเมทริกซ์จะไม่ซ้ำกัน

ในงานวิจัยนี้ได้ใช้รูปแบบการจัดค่าให้อยู่ในช่วง [0,1] ที่มีระยะห่างเท่า ๆ กันซึ่งเมื่อพิจารณาแล้วการสร้างแผนการทดลอง LHD ใหม่เกิดจากการสลับสมาชิกในแต่ละคอลัมน์ ดังนั้นจำนวนแผนการทดลองที่เป็นไปได้ทั้งหมดคือ $(n!)^d$ ซึ่งในตัวอย่างนี้เท่ากับ $(9!)^2$ หรือประมาณ 131,681 ล้านแผนการทดลอง ซึ่งการหาแผนการทดลองที่ดีที่สุดในที่นี้ทำได้ยาก และยิ่งเมื่อมิติมีขนาดใหญ่ขึ้นยิ่งทำให้การค้นหาแผนการทดลองที่ดีที่สุดในเวลาจำกัดแทบเป็นไปไม่ได้ ดังนั้นนักวิจัยส่วนใหญ่จึงใช้วิธีค้นหาแบบฮิวริสติก (Heuristic search algorithm) โดยจะเป็นการค้นหาแผนการทดลองที่เหมาะสมควบคู่กับเกณฑ์การหาค่าความเหมาะสม (Optimality criteria) ที่กำหนดไว้

2.2 อัลกอริทึมการค้นหาเฉพาะที่แบบวนซ้ำ

อัลกอริทึมการค้นหาเฉพาะที่แบบวนซ้ำ หรือ ILS ถูกนำเสนอโดย [18] โดยกลไกจะประกอบด้วยขั้นตอนหลักดังรูปที่ 4

```
procedure Iterated Local Search
    s0 = GenerateInitialSolution
    s* = LocalSearch(s0)
    repeat
        s' = Perturbation(s*, history)
        s*' = LocalSearch(s')
        s* = AcceptanceCriterion(s*, s*', history)
    until termination condition met
end
```

รูปที่ 4 รหัสเทียมของ ILS [18]

โดยกลไกการค้นหาประกอบด้วย 2 กระบวนการหลัก คือ LocalSearch() และ Perturbation() โดยกลไก LocalSearch() จะเน้นการค้นหาที่พยายามหลีกเลี่ยงจุด

ใกล้เคียง ส่วน Perturbation() เป็นกลไกเพื่อหลีกเลี่ยงการตกหลุมการค้นหาคำตอบเฉพาะที่ (Local optimal) โดยงานวิจัยนี้ได้ประยุกต์ใช้อัลกอริทึม ILS ตามงานวิจัยของ [18] โดยเลือกเกณฑ์เลือกค่าความเหมาะสม ϕ_p โดยมีขั้นตอนการทำงานดังรูปที่ 6 ซึ่งสามารถอธิบายโดยละเอียดได้ดังต่อไปนี้

<u>ขั้นตอนที่ 1</u> สุ่มสร้างเมทริกซ์แผนการทดลองขนาด *n x d* (X_0)

<u>ขั้นตอนที่ 2</u> ทำการคำนวณหาค่าระยะห่างที่น้อยที่สุด (Minimal distance) จากสมการ (1)

$$\min_{j \neq i} d(x_i, x_j) = D_1(X) \tag{1}$$

เพื่อเตรียมเซ็ทตำแหน่งของจุดวิกฤต (Critical points) ก่อนจะเข้าสู่กระบวนการ LocalSearch

โดยค่า min คือค่าระยะห่างระหว่างจุดที่น้อยที่สุด โดยในที่นี้จะบันทึก ตำแหน่งของจุด ในลักษณะของเซ็ท

$\zeta(X) \subseteq \{1,\ldots,n\}$ โดยที่

$i \in \zeta(X)$

$j \in \{1,\ldots,n\}$

$l \in \{1,\ldots,d\}$

ซึ่งในที่นี้ขอยกตัวอย่างการหาจุดวิกฤตดังนี้

ถ้ากำหนดให้แผนการทดลองมีขนาด 5 x 4 โดยที่มีค่าต่าง ๆ ดังแผนการทดลอง *x*

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

เมื่อคำนวณตารางระยะห่างยูคลิด (Euclidean distance) ของแผนการทดลองจะได้ว่า

$$d(x) = \begin{bmatrix} 0 & 0.7905 & 1.1180 & 1.3693 & 0.7905 \\ 0 & 0 & 0.7905 & 1.0606 & 0.8602 \\ 0 & 0 & 0 & 1.6201 & 1.0606 \\ 0 & 0 & 0 & 0 & 1.3693 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

จากนั้นพิจารณาที่ตำแหน่งระยะห่างน้อยที่สุด (Minimal distance) ของ d(x) จากสมการ (1) ได้ค่าเท่ากับ 0.7905 ดังนั้นเซ็ทของจุดวิกฤตที่ได้คือ (1,2), (1,5) และ (2,3)

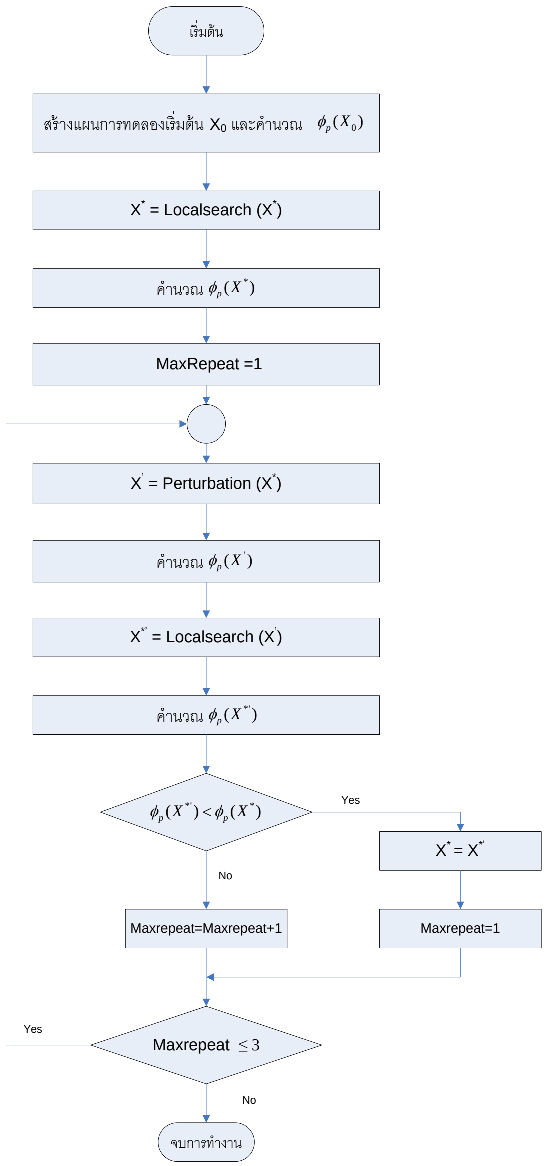
การสลับค่า (Swap) ในคอลัมน์ใดคอลัมน์หนึ่งระหว่าง x_i และ x_j จะทำตามเงื่อนไขต่อไปนี้

$$y_{rh} = \begin{cases} x_{rh} & if\ r \neq i, j \quad or\ h \neq l \\ x_{ih} & if\ r = j \quad and\ h = l \\ x_{jh} & if\ r = i \quad and\ h = l \end{cases}$$

ซึ่งเมื่อขยายความจะได้ รหัสเทียมดังรูปที่ 5

```
for i = 1,...,n
   for j = 1,...,n
      Step 1: if i ≠ j and {i, j} ∩ ζ(X) ≠ ∅, then:
         for ℓ = 1,...,d
            Step 2: let Y be the resulting LHD
                  after the local move involving
                  points x_i and x_j and component ℓ ;
            Step 3: if Y is better than X, then set X = Y.
         endfor
   endfor
endfor
Step 4: Repeat the loops if there has been at least an
improvement, otherwise STOP
```

รูปที่ 5 รหัสเทียมแสดงกลไกการสลับตำแหน่ง



รูปที่ 6 ผังงานแสดงกลไกการทำงานทั้งหมด

<u>ขั้นตอนที่ 3</u> กระบวนการ Perturbation เป็นการสร้างแผนการทดลองใหม่ X^* ที่ใกล้เคียงกับแผนการทดลองปัจจุบันโดยใช้แนวความคิด Cyclic order exchange (COE) หรือการสลับที่แบบหมุนตำแหน่งโดยขั้นตอนแรก (Step 1) จะทำการสุ่มเลือกแถว 2 ตำแหน่งใด ๆ ที่แตกต่างกันระหว่าง x_i และ x_j โดยที่ $i < j$ and $(j - i) \geq 2$ ใน LHD X^* หลังจากนั้นขั้นตอนที่ 2 (Step 2) จะทำการสุ่มเลือกคอลัมน์ ℓ เพื่อทำการสลับที่แบบหมุนตำแหน่ง (COE) ระหว่าง x_i และ x_j ในขั้นตอนที่ 3 (Step 3) จะทำการแทนที่คอลัมน์ $x_{(t+1)l}$ ด้วย $x_{(t)l}$ สำหรับทุกค่า $t = i, i+1, \ldots, j-1$ และแทนที่คอลัมน์ x_{il} ด้วย x_{jl} ดังแสดงตามรหัสเทียมในรูปที่ 7

<u>ขั้นตอนที่ 4</u> เกณฑ์การยอมรับค่าที่เหมาะสม (Acceptance criterion) คือจะยอมรับค่าที่เหมาะสมโดยวัดค่าความเปลี่ยนแปลงของ ϕ_p ถ้า ϕ_p ของแผนการทดลองใหม่ที่สร้างขึ้นมีค่าน้อยกว่าแผนการทดลองเดิม ก็จะมีการอัพเดทค่าแผนการทดลองที่เหมาะสมเพื่อเก็บไว้ในรอบต่อไป

Step 1:randomly select two different point x_i and x_j
such that $i < j$ and $j - i \geq 2$
Step 2:Randomly choose a component ℓ
Step 3:Replace the component $x_{(t+1)\ell}$ by $x_{t\ell}$,
$\forall t = i, i+1, \ldots, j-1$
and replace $x_{i\ell}$ by $x_{j\ell}$

รูปที่ 7 รหัสเทียมของการสลับที่แบบหมุนตำแหน่ง

ขั้นตอนที่ 5 กฎการหยุดค้นหา (Stopping condition) ในที่นี้ผู้วิจัยได้กำหนดเงื่อนไขการหยุด โดยถ้าแผนการทดลองที่สร้างขึ้นใหม่ไม่ดีขึ้น 3 รอบติดต่อกัน ให้หยุดการทำงาน

2.3 เกณฑ์การเลือกค่าที่เหมาะสม (Optimality criteria)

งานวิจัยนี้ใช้เกณฑ์แบบ ϕ_p ซึ่งถูกพัฒนามาจากเกณฑ์การเลือกค่าเหมาะสมแบบ Maximin distance criterion [7] โดยพิจารณาว่าแผนการทดลองที่ดีที่สุดนั้นต้องให้ค่า ϕ_p ที่ต่ำที่สุดเท่าที่เป็นไปได้ ยกตัวอย่างเช่น กรณีแผนการทดลอง *X* ใด ๆ จะสามารถคำนวณระยะทางระหว่างจุดทดลอง 2 จุดใด ๆ บนปริภูมิการทดลองโดยใช้ระยะทางแบบยูคลิดโดยใช้สมการ (2)

$$d(x_j, x_k) = [\sum_{i=1}^{d}(x_{ji} - x_{ki})^2]^{\frac{1}{2}} \tag{2}$$

หลังจากคำนวณระยะห่างครบทุกจุดการทดลองแล้ว จะได้เมทริกซ์ระยะห่างซึ่งเป็นเมทริกซ์จัตุรัสขนาด *n x n* ดังนี้

$$\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1j} \\ d_{21} & d_{22} & \cdots & d_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ d_{i1} & d_{i2} & \cdots & d_{ij} \end{bmatrix}$$

จากนั้นทำการเรียงระยะห่างระหว่างจุดทั้งหมดจากน้อยไปมาก $(d_1, d_2, \ldots, d_m)$ และสร้างดัชนี (Index) $(k_1, k_2, \ldots, k_m)$ โดยที่ *j* คือจำนวนของคู่จุดที่แบ่งตามระยะทาง d_j และคำนวณเกณฑ์ ϕ_p ได้โดยใช้สมการ (3)

$$\phi_p = \left[\sum_{j=1}^{m} k_j d_j^{-p}\right]^{\frac{1}{p}} \tag{3}$$

ต่อมา [9] ได้พัฒนาวิธีการคำนวณเกณฑ์ ϕ_p ให้ง่ายขึ้นโดยไม่ต้องนำค่าของ j_j มาเกี่ยวข้องและจะได้ผลลัพธ์ดังสมการ (4)

$$\phi_p = \left[\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{(d_{ij})^p}\right]^{\frac{1}{p}} \tag{4}$$

โดยแผนการทดลองที่เหมาะสมที่สุดคือแผนการทดลองที่ให้ค่า ϕ_p ต่ำที่สุด

3. ผลการวิจัย

งานวิจัยนี้ผู้วิจัยได้เขียนโปรแกรมด้วยโปรแกรมภาษา R ทำการค้นหาแผนการทดลองที่เหมาะสมโดยใช้อัลกอริทึม ILS โดยการรันโปรแกรมบนมิติการทดลองขนาดต่าง ๆ ดังแสดงในตารางที่ 1 โดยขนาดต่าง ๆ เป็นไปตามรูปแบบของสมการพหุนามดีกรีกำลังสอง (Second order polynomial) ตามสมการ (5)

$$n = 2d + 4\binom{d}{2} + 1 \tag{5}$$

เมื่อ *n* คือจำนวนรัน และ *d* คือจำนวนตัวแปรเข้า

ตารางที่ 1 มิติของแผนการทดลองที่ใช้ในการทดสอบ

d	2	3	4	5	6	7	8	9	10
n	9	19	33	51	73	99	129	163	201

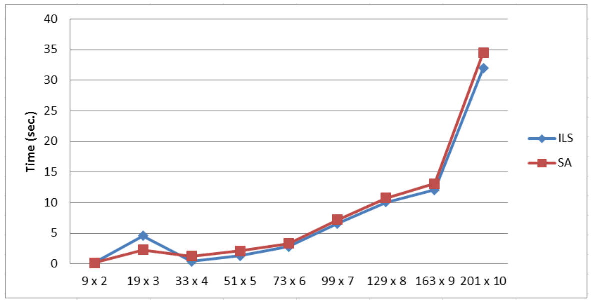
โดยผู้วิจัยได้ทำการทดสอบสำหรับแต่ละมิติปัญหาจะมีการทำซ้ำทั้งหมด 10 ครั้ง โดยได้ผลลัพธ์ดังตาราง 2 จากตาราง 2 แสดงค่าสถิติพื้นฐาน เช่น ค่าเฉลี่ย (Mean) ค่าต่ำสุด (Min) ค่าสูงสุด (Max) และค่าส่วนเบี่ยงเบนมาตรฐาน (SD) ของเกณฑ์การเลือกค่าที่เหมาะสมแบบ ϕ_p และเวลาเฉลี่ย (Average time) ที่ใช้ในการค้นหาแผนการทดลองที่

เหมาะสมสำหรับแต่ละมิติปัญหา ผลที่ได้จากตารางจะเห็นว่าอัลกอริทึมการสืบค้นแบบ ILS สามารถค้นหาแผนการทดลองที่เหมาะสมได้ภายใต้เงื่อนไขเวลาในการค้นหาอยู่ในระดับที่ยอมรับได้โดยพิจารณาจากค่าเกณฑ์ ϕ_p ที่ได้เมื่ออัลกอริทึมจบการทำงานเทียบกับงานวิจัยก่อนหน้า [11] และเมื่อพิจารณาส่วนเบี่ยงเบนมาตรฐาน (SD.) ของค่าเกณฑ์ ϕ_p ที่ได้จากการรันซ้ำจำนวน 10 ซ้ำ จะเห็นว่าค่า SD. มีค่าค่อนข้างต่ำ แสดงให้เห็นความสามารถในการทำงานของ ILS ที่ค่อนข้างจะคงที่ ซึ่งสามารถสรุปได้ว่า ILS มีความแกร่งต่อแผนการทดลองเริ่มต้นที่แตกต่างกัน

ตาราง 2 ค่าสถิติแสดงประสิทธิภาพของ ILS

n x d	ϕ_p				Average Time (sec.)
	Min	Max	Mean	SD	
9x2	4.2735	6.7144	5.6256	0.9116	0.2020
19x3	4.9454	8.4495	6.2843	1.0749	4.5870
33x4	6.3432	8.4127	6.7608	0.6165	0.4360
51x5	6.3532	7.2385	6.6553	0.2475	1.3580
73x6	6.4588	7.4146	6.7107	0.2560	2.8470
99x7	6.4958	6.7035	6.5918	0.0750	6.6067
129x8	6.5155	6.7202	6.5661	0.0625	10.0840
163x9	6.5653	6.6420	6.6139	0.0286	12.0720
201x10	6.5981	6.6278	6.6184	0.0116	31.9920

จากรูปที่ 8 เปรียบเทียบเวลาเฉลี่ยที่ใช้ในการค้นหาแผนการทดลองที่เหมาะสมสำหรับแต่ละมิติปัญหา จะเห็นได้ว่า ILS สามารถค้นหาแผนการทดลองที่มีคุณสมบัติที่ดีทัดเทียมกันโดยใช้เวลาที่ใกล้เคียงกับอัลกอริทึม SA



รูปที่ 8 ปรียบเทียบเวลาที่ใช้ระหว่างอัลกอริทึม ILS และ SA

4. สรุปผลและข้อเสนอแนะ

งานวิจัยนี้ได้ทำการประยุกต์การใช้อัลกอริทึมการค้นหาเฉพาะที่แบบวนซ้ำสำหรับสร้างแผนการทดลองแบบ LHD ที่เหมาะสมสำหรับการจำลองการทดลองการทดลองด้วยคอมพิวเตอร์ จากผลการวิจัยแสดงให้เห็นว่า ILS สามารถนำมาประยุกต์ใช้ในการสร้างแผนการทดลองแบบ LHD ที่เหมาะสมได้อย่างมีประสิทธิภาพ อย่างไรก็ตามเมื่อมิติของปัญหาที่สนใจศึกษามีขนาดใหญ่ขึ้นจะส่งผลให้อัลกอริทึมการทำงานของ ILS ใช้เวลานานมากขึ้น ดังนั้นจึงควรมีการปรับปรุงโครงสร้างการทำงานของอัลกอริทึม ILS ให้ดียิ่งขึ้น เช่นปรับปรุงกลไกในการสลับสมาชิก และสร้างกฎการหยุดค้นหาที่เหมาะสมสำหรับ ILS นอกจากนี้ควรมีการเปรียบเทียบประสิทธิภาพของ ILS กับอัลกอริทึมอื่น ๆ ที่มีอยู่แล้วเพื่อเป็นการขยายขอบเขตของการวิจัยที่เกี่ยวข้องกับการสร้างแผนการทดลองที่เหมาะสม

5.บรรณานุกรม

[1] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, “Design and analysis of computer experiment”, *Statistical Science,* Vol. 4(4), pp. 409-435, 1989.

[2] J. Koehler and A. Owen, "Computer experiments," *Handbook of Statistics*, vol.13. New York :Elsevier Science, pp. 261-308, 1996.

[3] K. T. Fang, R. Li and A. Sudjianto. *Design and modeling for computer experiments*. New York : Chapman & Hall, 2005.

[4] M. D. Mckay, Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variable in the analysis of output from a computer code," *Technometrics.* Vol. 21, pp. 239-246, 1979.

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

[6] T. W. Simpson, J. D. Peplinski, P. M. Koch and J. K. Allen, "Metamodels for computer-based engineering design: survey and recommendations," *Engineering with Computers*, Vol. 17, pp. 129-150, 2001.

[7] M. D. Morris, and T. J. Mitchell, "Exploratory design for computational experiments," *Journal of Statistical Planning and Inference,* Vol. 43, pp. 381-402, 1995.

[8] W. Li and C. F. J Wu, "Columnwise-pairwise algorithms with applications to the construction of supersaturated designs," *Technometrics,* Vol. 39, pp. 171-179, 1997.

[9] R. Jin, W. Chen and A. Sudjianto, "An efficient algorithm for construct optimal design of computer experiment," *Journal of statistical Planning and Inference,* Vol. 134, pp. 268-287, 2005.

[10] M. Liefvendahl and R. Stocki , "A study on algorithms for optimization of Latin hypercubes," *Journal of Statistical Planning and Inference,* Vol. 136, pp. 3231-3247, 2006.

[11] J. Rungrattanaubol, A. Na-udom,"Comparison of evolutionary search algorithms in computer simulated experiments," *in Proc. 11th National Computer Science and Engineering Conference (NCSEC..2007)*, Nov. 19-21, 2007. Bangkok, Thailand, pp.26-30.

[12] V. R. Joseph and Y. Hung, "Orthogonal-Maximin Latin Hypercube Designs," *Statistica Sinica,* Vol. 18, pp. 171-186, 2008.

[13] T. M. Cioppa, and T. W. Lucas, "Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes." *Technometrics,* Vol. 49(1), pp. 45-55, 2007.

[14] D. M. Steinberg, "A construction method for orthogonal Latin hypercube designs," *Biometrika,* Vol. 93(2), pp. 279-288, 2006.

[15] K. Q. Ye, W. Li, and A. Sudjianto, "Algorithmic construction of optimal symmetric Latin hypercube designs," *Journal of Statistical planning and inference,* Vol. 90, pp. 145-159, 2000.

[16] P. Prescott, “Orthogonal-column Latin hypercube design with small samples,” *Computational Statistics and Data Analysis,* Vol. 53, pp. 1191-1200, 2009.

[17] F. A. C. Viana, G. Venter and V. Balabanov, “An algorithm for fast optimal latin hypercube design of experiments,” *International Journal for Numerical Methods in Engineering,* Vol. 82(2), pp. 135-156, 2010.

[18] A. Grosso, A. R. M. J. U. Jamali, M. Locatelli, “Finding maximin latin hypercube designs by Iterated Local Search heuristics,” European *Journal of Operational Research*, Vol. 197, pp. 541-547, 2009.