

อภิธานศัพท์



การปรับปรุงเพื่อเพิ่มประสิทธิภาพของอัลกอริทึมการสืบค้นเพื่อค้นหา  
แผนการทดลองที่เหมาะสมที่สุด

Modified search algorithm for the construction of the optimal designs



สำนักหอสมุด มหาวิทยาลัยธรรมศาสตร์  
วังทองเพ็ญ..... - 5 JUL 2011  
เลขทะเบียน..... 15662011  
เลขเรียกหนังสือ..... อ. อ. ๑๙

จรัสศรี รุ่งรัตนอุบล  
อนามัย นาคอุดม

7/6/11  
A63  
A161ก  
2551

ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ  
ภาควิชาคณิตศาสตร์  
คณะวิทยาศาสตร์  
มหาวิทยาลัยธรรมศาสตร์

รายงานวิจัยคณะวิทยาศาสตร์ มหาวิทยาลัยธรรมศาสตร์  
งบประมาณวิจัยคณะวิทยาศาสตร์ มหาวิทยาลัยธรรมศาสตร์ประจำปี 2551

## ประกาศคุณูปการ

งานวิจัยนี้สำเร็จลุล่วงได้ด้วยดีด้วยความกรุณาอย่างยิ่งจากคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวรที่ได้ให้ทุนสนับสนุนในการทำวิจัย

ขอขอบคุณ ดร. อนามัย นาอุดมผู้ร่วมโครงการที่ให้ความช่วยเหลือด้านสถิติวิเคราะห์การโปรแกรม และนายธรรมรัตน์ ธรรมมา ผู้ให้ความร่วมมือในการเขียนโปรแกรมและเขียนบทความวิจัย เพื่อนำเสนอในงานประชุมวิชาการด้านการวิจัยดำเนินงานแห่งชาติ ประจำปี 2551 ที่จัดขึ้น ณ โรงแรมเจ้าพระยาปาร์ค กรุงเทพมหานคร ในวันที่ 24-25 กรกฎาคม 2551

นอกจากนี้ข้าพเจ้าใคร่ขอขอบคุณ นิสิตช่วยวิจัยทุกท่านที่ให้ความร่วมมืออย่างดียิ่งจนทำให้งานวิจัยนี้ประสบความสำเร็จอย่างน่าพอใจยิ่ง

คุณค่าและประโยชน์อันพึงมีจากงานวิจัยนี้ ผู้วิจัยขอมอบอุทิศแด่ผู้มีพระคุณทุก ๆ ท่าน

จรัสศรี รุ่งรัตนอุบล

กันยายน 2551



**ชื่อเรื่อง** การปรับปรุงเพื่อเพิ่มประสิทธิภาพของอัลกอริทึมการสืบค้นเพื่อค้นหาแผนการทดลองที่เหมาะสมที่สุด

**ชื่อเรื่องภาษาอังกฤษ** Modified search algorithm for the construction of the optimal designs

**หัวหน้าโครงการวิจัย** ดร. จรัสศรี รุ่งรัตนอุบล

**ผู้ร่วมวิจัย** ดร. อนามัย นาอุดม

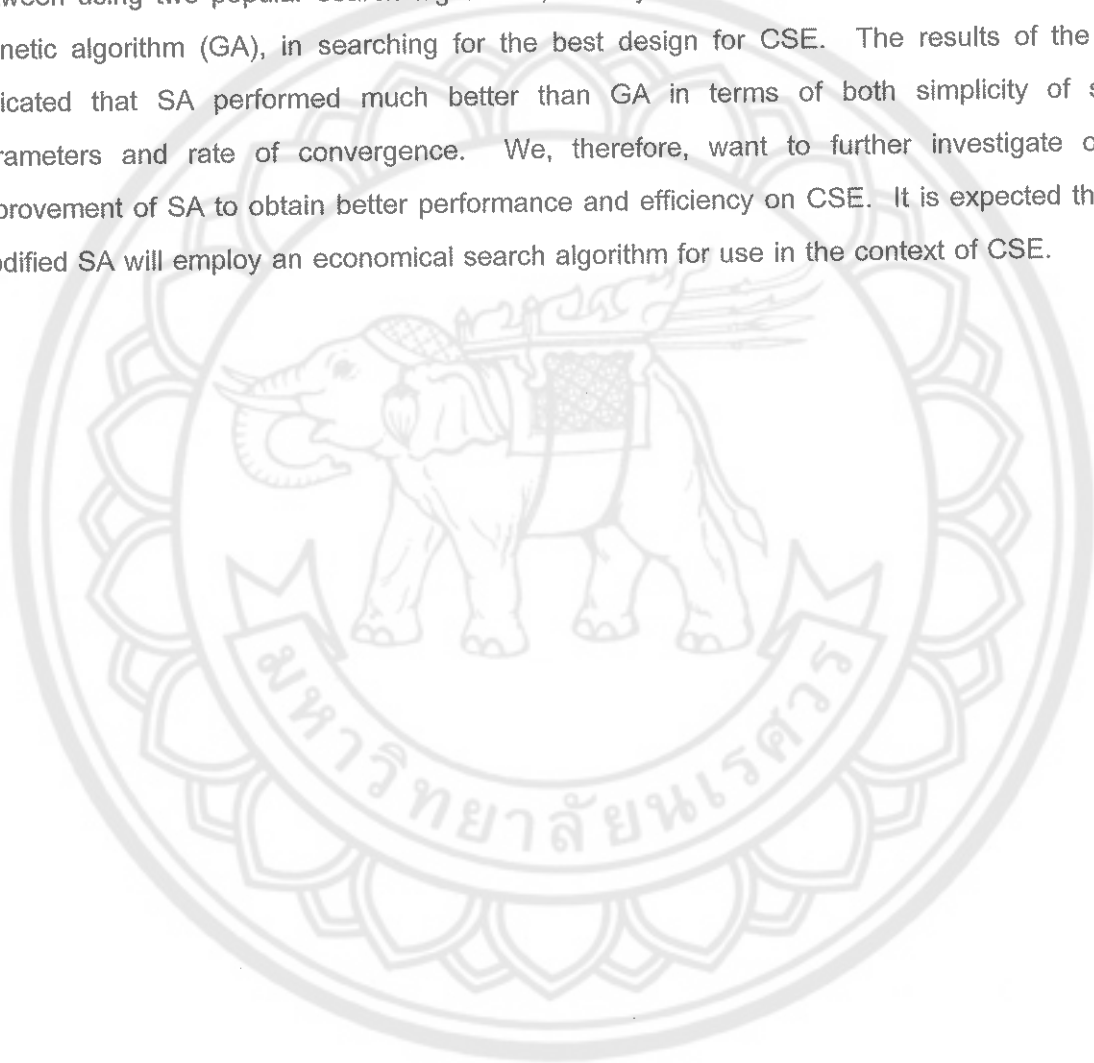
**งบประมาณวิจัยคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวรประจำปี 2551**

## บทคัดย่อ

การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer simulated experiment: CSE) ได้ถูกนำมาใช้อย่างแพร่หลายในงานวิจัยด้านวิทยาศาสตร์ และด้าน วิศวกรรมศาสตร์ ประสิทธิภาพและผลที่ได้จากการจำลองการทดลองด้วยคอมพิวเตอร์นั้นจะขึ้นอยู่กับการวางแผนทดลองเพื่อเลือกกลุ่มของตัวแปรอิสระเข้ามาทำการทดลองด้วยคอมพิวเตอร์ โดยทั่วไปการค้นหาแผนการทดลองที่เหมาะสมที่สุดสามารถทำได้โดยการใช้ อัลกอริทึมการสืบค้น (Search algorithm) ควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสม (Optimality criteria) จากการวิจัยก่อนหน้านี้ผู้วิจัยทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมการสืบค้น 2 ชนิด คือ Simulated annealing algorithm (SA) และ Genetic algorithm (GA) ซึ่งผลจากการวิจัยพบว่า SA มีประสิทธิภาพดีกว่า GA ทั้งในด้านความง่ายในการกำหนดพารามิเตอร์ตั้งต้น (Simplicity) รวมไปถึงความสามารถในการลู่เข้าสู่เกณฑ์ในการเลือกค่าที่เหมาะสม (Rate of convergence) ที่ดีกว่า ดังนั้นงานวิจัยนี้จึงมุ่งศึกษาเพื่อเพิ่มประสิทธิภาพของ SA ให้ดียิ่งขึ้น ซึ่งผลจากการวิจัยครั้งนี้ จะนำไปสู่การเลือกใช้ SA เพื่อสืบค้นแผนการทดลองที่เหมาะสมที่สุดของแต่ละมิติปัญหาใด ๆ สามารถทำได้อย่างรวดเร็ว และได้แผนการทดลองที่มีลักษณะที่เหมาะสมเหมือนเดิมทุกประการ แต่ใช้เวลาในการสืบค้นน้อยลง

## Abstract

Currently computer simulated experiments (CSE) have been extensively used in sciences and engineering applications. Selecting a proper design to run CSE is very critical in terms of both efficiency and reliability. Normally the best design is generated by using a search algorithm along with pre-specified optimality criteria. In the previous study, we compared the efficiency between using two popular search algorithms, namely Simulated annealing algorithm (SA) and Genetic algorithm (GA), in searching for the best design for CSE. The results of the study indicated that SA performed much better than GA in terms of both simplicity of setting parameters and rate of convergence. We, therefore, want to further investigate on the improvement of SA to obtain better performance and efficiency on CSE. It is expected that this modified SA will employ an economical search algorithm for use in the context of CSE.



# สารบัญ

ประกาศคุณูปการ.....	II
บทคัดย่อ.....	III
Abstract.....	IV
บทที่ 1.....	5
บทนำ.....	5
หลักการและเหตุผลที่เสนอโครงการ.....	5
วัตถุประสงค์.....	7
แผนงานวิจัยและขั้นตอนแผนงาน.....	8
บทที่ 2.....	9
เอกสารและงานวิจัยที่เกี่ยวข้อง.....	9
การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment: CSE).....	9
อัลกอริทึมการสืบค้นและเกณฑ์การเลือกค่าที่เหมาะสม.....	15
เกณฑ์การเลือกค่าที่เหมาะสม.....	18
บทที่ 3.....	21
วิธีดำเนินการวิจัย.....	21
อุปกรณ์ที่ใช้ในการดำเนินงานวิจัย.....	21
เครื่องมือที่ใช้ในการออกแบบ.....	21
DACE (A MATLAB Kriging Toolbox Version 2.0).....	26
บทที่ 4.....	30
ผลการวิจัย.....	30
การทำงานของ SA และการกำหนดค่าพารามิเตอร์ตั้งต้นต่างๆ.....	31
การคำนวณหาค่าเกณฑ์เหมาะสมที่สุด (Optimality Criteria).....	32
ทดสอบแผนการทดลองที่เหมาะสมที่สุดจาก SA โดยใช้ DACE (A MATLAB Kriging Toolbox) ...	34
ปรับปรุงอัลกอริทึม SA (Modified Simulated Annealing: MSA).....	42
การคำนวณหาค่า $\phi_p$ โดยไม่ต้องมีการคำนวณซ้ำ.....	45
ผลทดสอบและวัดผลระหว่าง MSA1 กับ MSA2.....	49

บทที่ 5 .....	51
สรุปผลการวิจัย.....	51
สรุปผลการวิจัย.....	51
ข้อเสนอแนะ .....	51
บรรณานุกรม.....	53
ภาคผนวก.....	55



# บทที่ 1

## บทนำ

**ชื่อโครงการ (ไทย)** การปรับปรุงเพื่อเพิ่มประสิทธิภาพของอัลกอริทึมการสืบค้นเพื่อค้นหาแผนการทดลองที่เหมาะสมที่สุด  
(อังกฤษ) Modified search algorithm for the construction of the optimal designs

**ชื่อหัวหน้าโครงการ (ไทย)** ดร. จรัสศรี รุ่งรัตนอุบล (70%)  
(อังกฤษ) Dr. Jaratsri Rungrattanaubol  
คุณวุฒิ/สาขาวิชา PhD (Computer science)  
ตำแหน่ง อาจารย์  
ที่ทำงาน ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ  
คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร  
โทรศัพท์ (055) 261-000 ต่อ 3205  
โทรสาร (055) 261000 ต่อ 3201  
e-mail jaratsrir@nu.ac.th.

**ผู้ร่วมโครงการวิจัย (ไทย)** ดร. อนามัย นาอุดม (30%)  
(อังกฤษ) Dr. Anamai Na-udom  
คุณวุฒิ/สาขาวิชา Ph.D. (Applied Statistics)  
ตำแหน่ง อาจารย์  
ที่ทำงาน ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร  
โทรศัพท์ (055) 261-000 ต่อ 3259  
โทรสาร (055) 261000 ต่อ 3201  
e-mail jaratsrir@nu.ac.th

### หลักการและเหตุผลที่เสนอโครงการ

ในปัจจุบันการจำลองการทดลองด้วยคอมพิวเตอร์ได้เข้ามามีบทบาทในการศึกษารูปแบบความสัมพันธ์ของตัวแปรอิสระและตัวแปรตามมากขึ้น โดยทั่วไปแล้ว เทคนิคการจำลองด้วยคอมพิวเตอร์มักจะใช้เวลาและสิ้นเปลืองทรัพยากรมาก ดังนั้นนักวิจัยทั่วไปจึงได้มีการคิดค้นที่จะค้นหากลุ่มของระดับตัว

แปรอิสระที่เหมาะสม หรือ แผนการทดลองที่เหมาะสมเพื่อนำไปใช้ในการจำลองการทดลองด้วยคอมพิวเตอร์ ซึ่งการสร้างแผนการทดลองที่เหมาะสมที่สุดนั้นจะต้องใช้ อัลกอริทึมการสืบค้น (Search algorithm) ควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสม (Optimality criteria)

ในการศึกษาวิจัยก่อนหน้าผู้วิจัยได้ทำการเปรียบเทียบอัลกอริทึมการสืบค้น 2 ชนิดคือ Simulated Annealing algorithm (SA) และ Genetic Algorithm (GA) และผลจากการศึกษาพบว่า SA มีประสิทธิภาพเหนือกว่า GA ในด้านความง่ายของการกำหนดพารามิเตอร์ตั้งต้น (Simplicity) รวมไปถึงเวลาที่ต้องใช้ในการสืบค้น และในด้านการลู่เข้าสู่ค่าที่ดีกว่าของเกณฑ์ที่เหมาะสมที่สุด (Rate of convergence) ดังนั้นงานวิจัยนี้จึงเป็นงานวิจัยต่อเนื่อง โดยผู้วิจัยจะทำการปรับปรุงประสิทธิภาพของ SA ให้ดียิ่งขึ้น โดยพิจารณาเกณฑ์ในการเลือกค่าที่เหมาะสมแบบ  $\phi_p$

การทำงานของ SA สามารถแบ่งได้เป็น 5 ขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1: กำหนดค่าพารามิเตอร์ตั้งต้น

- $I_{max}$  (จำนวนครั้งในการค้นหาแผนการทดลองที่ดีขึ้น)
- $t_0$  (ค่าอุณหภูมิตั้งต้น)
- $C_t$  (อัตราการลดอุณหภูมิเมื่อแผนการทดลองที่ดีขึ้นไม่ถูกค้นพบ)
- $n$  (จำนวนวันของแผนการทดลอง)
- $d$  (จำนวนตัวแปรอิสระในแผนการทดลอง)
- $\phi_p$  (เกณฑ์ที่ใช้ในการเลือกค่าที่เหมาะสมของแผนการทดลอง)

ขั้นตอนที่ 2: สร้างแผนการทดลองตั้งต้น ( $X$ ) ซึ่งมีมิติเท่ากับ  $n \times d$  ตามขั้นตอนที่ 1 โดยใช้วิธีการสุ่ม โดยที่

$$X_{best} = X \text{ และ } t = t_0$$

ขั้นตอนที่ 3: กำหนดลักษณะการวนลูปทำงานของอัลกอริทึมเพื่อสืบค้นแผนการทดลองที่ดีขึ้น

$$I = 0 \text{ และ } Counter = 0$$

ขั้นตอนที่ 4: กำหนดวิธีการสุ่มเพื่อหาแผนการทดลองที่ดีขึ้นโดยการใช้โครงสร้างของเพื่อนบ้าน (Neighborhood structure) โดยกำหนดให้  $X_{try} = X$  เมื่อ  $X_{try}$  เกิดจากการสุ่มโดยการแลกเปลี่ยนสมาชิกในสตมภ์ใด ๆ ที่ถูกเลือกมาโดยสุ่ม และภายในสตมภ์ที่ถูกเลือกนั้นให้ทำการแลกเปลี่ยนสมาชิกระหว่างสองแถวใด ๆ ที่ถูกเลือกมาโดยสุ่ม

ขั้นตอนที่ 5: ตรวจสอบว่า  $X_{try}$  เป็นแผนการทดลองที่ดีขึ้นหรือไม่ ถ้าดีขึ้นให้ทำการแทนที่  $X = X_{try}$  และกำหนดค่า  $Counter = 1$  แต่ถ้า  $X_{try}$  ไม่ใช่แผนการทดลองที่ดีขึ้น ให้คำนวณความน่าจะเป็นในการแทนที่  $X = X_{try}$  โดยใช้สูตร  $\exp(-[\phi_p(D_{try}) - \phi_p(D)]/t)$  ถ้าเกิดการแทนที่ให้กำหนด  $Counter = 1$

ขั้นตอนที่ 6: ถ้า  $\phi_p(X_{try}) < \phi_p(X_{best})$  ให้กำหนด  $I = 1$  และ  $X_{best} = X_{try}$  ในทางตรงกันข้าม ให้กำหนด  $I = I + 1$



**ขั้นตอนที่ 7:** ถ้า  $I < I_{max}$  ให้กลับไปทำขั้นตอนที่ 4

**ขั้นตอนที่ 8:** ตรวจสอบว่ามีการแทนที่ด้วยแผนการทดลองที่ดีกว่าหรือไม่ ( $Counter = 1$ ) ให้ทำการลดอนุกรมมิโดยใช้สูตร  $t = t \times C_t$  และกลับไปทำขั้นตอนที่ 3

**ขั้นตอนที่ 9:** หยุดการสืบค้นและรายงานแผนการทดลองที่ดีที่สุด  $X_{best}$

จากขั้นตอนการทำงานทั้งหมดของ SA ที่ได้กล่าวข้างต้น ผู้วิจัยสังเกตเห็นว่า เราสามารถเพิ่มประสิทธิภาพการทำงานของ SA ได้โดยการปรับปรุง การคำนวณหาค่า  $\phi_p$  ในแต่ละครั้งของการแลกเปลี่ยนสมาชิกในขั้นตอนที่ 4 เพื่อให้ SA ทำการสืบค้นได้รวดเร็วขึ้น ซึ่งแนวคิดนี้สามารถทำได้โดยการคำนวณค่า  $\phi_p$  เฉพาะสมาชิกในแผนการทดลองที่ได้รับผลกระทบจากการแลกเปลี่ยนสมาชิกเท่านั้น ประเด็นต่อมาที่จะทำให้สามารถเพิ่มประสิทธิภาพของ SA ได้คือ การเปรียบเทียบเพื่อจะทำการแทนที่ด้วยแผนการทดลองที่ดีกว่าในขั้นตอนที่ 6 นั้นสามารถทำได้โดยการกำหนดระยะทางเริ่มต้น (Threshold distance) ที่เหมาะสมในการเปรียบเทียบว่าควรจะทำการแทนที่หรือไม่ ซึ่งวิธีการนี้จะส่งผลให้เวลาที่ใช้ในการสืบค้นของ SA ลดลงอย่างเห็นได้ชัด แต่ยังคงสามารถรายงานแผนการทดลองที่มีคุณสมบัติดีเท่าเดิมทุกประการ

จากที่กล่าวมาข้างต้น จะเห็นได้ว่า การปรับปรุงเพื่อเพิ่มประสิทธิภาพในการสืบค้นของ SA จะทำให้ลดเวลาในการสืบค้น รวมไปถึง อัตราความเร็วของการเข้าสู่ค่าที่เหมาะสม (Rate of convergence) นั้นจะเป็นไปได้อย่างรวดเร็วยิ่งขึ้น ซึ่งผลที่ได้จากการวิจัยนี้จะส่งผลให้ผู้วางแผนการทดลองเพื่อจำลองการทดลองด้วยคอมพิวเตอร์ สามารถสร้างแผนการทดลองที่เหมาะสมที่สุดได้อย่างรวดเร็วยิ่งขึ้น

### วัตถุประสงค์

- 1) เพื่อปรับปรุงประสิทธิภาพของอัลกอริทึมการสืบค้น (SA) ให้มีประสิทธิภาพมากยิ่งขึ้น
- 2) เพื่อเสนออัลกอริทึมการสืบค้นที่เหมาะสมสำหรับสร้างแผนการทดลองเพื่อการจำลองการทดลองด้วยคอมพิวเตอร์

### เป้าหมาย

งานวิจัยนี้เป็นงานวิจัยต่อเนื่องจากงานวิจัยก่อนหน้านี้ โดยมีวัตถุประสงค์เพื่อปรับปรุงประสิทธิภาพของอัลกอริทึมการสืบค้นให้มีประสิทธิภาพมากยิ่งขึ้น ซึ่งผลจากงานวิจัยนี้จะส่งผลให้ผู้ใช้งานสามารถใช้อัลกอริทึมการสืบค้นเพื่อสร้างแผนการทดลองที่เหมาะสมที่สุดได้อย่างรวดเร็วและมีประสิทธิภาพยิ่งขึ้น

งานวิจัยนี้จะมีผลกระทบในเชิงวิชาการและการศึกษา เนื่องจากผู้ใช้งานสามารถนำผลจากการวิจัยไปใช้ในการค้นหาแผนการทดลองที่เหมาะสมที่สุดเมื่อทราบมิติของปัญหาใด ๆ ได้อย่างรวดเร็วและมีประสิทธิภาพมากขึ้น โดยการใช้ อัลกอริทึมการสืบค้นที่ได้จากการปรับปรุงแล้ว

## ความพร้อมด้านอุปกรณ์/เครื่องมือ และสถานที่ดำเนินงานวิจัย

เครื่องมือที่ใช้ในงานวิจัยประกอบด้วย เครื่องคอมพิวเตอร์ประสิทธิภาพสูง ของภาควิชา  
วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา

## แผนงานวิจัยและขั้นตอนแผนงาน

- 1) ศึกษาลักษณะการทำงานของ SA อย่างละเอียด และพิจารณาแต่ละขั้นตอน เพื่อหาทางปรับปรุง และเพิ่มประสิทธิภาพของ SA ให้ดียิ่งขึ้น
- 2) ศึกษาแนวทางในการปรับปรุง SA ให้ดียิ่งขึ้น
- 3) กำหนดมิติของปัญหาที่สนใจการศึกษา
- 4) พัฒนาโปรแกรม R เพื่อจำลองกรณีต่าง ๆ
- 5) รันโปรแกรมเพื่อเก็บรวบรวมข้อมูล
- 6) วิเคราะห์ข้อมูล
- 7) สรุปผลการปรับปรุงประสิทธิภาพของ SA และเปรียบเทียบกับ SA แบบเดิม
- 8) จัดทำรูปเล่มรายงาน

## บทที่ 2

### เอกสารและงานวิจัยที่เกี่ยวข้อง

การศึกษาการปรับปรุงอัลกอริทึมการค้นหาเพื่อการจำลองการทดลองด้วยคอมพิวเตอร์ (Modification on Search Algorithm for Computer Simulated Experiment) ได้ทำการทบทวนวรรณกรรมที่เกี่ยวข้อง โดยมีเนื้อหาประกอบด้วย

การจำลองการทดลองด้วยคอมพิวเตอร์

อัลกอริทึมการค้นหาและเกณฑ์การเลือกค่าที่เหมาะสม

สถิติที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้อง

#### การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment: CSE)

การทดลองโดยทั่วไปสามารถแบ่งเป็น 2 ประเภทใหญ่ คือ การทดลองทางกายภาพหรือการทดลองในห้องปฏิบัติการ (Physical Experiment) และ การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment) โดยส่วนมากการทดลองในชีวิตประจำวันมักจะเป็นการทดลองแบบที่ทำการทดลองกันในห้องทดลอง โรงงาน หรือแปลงเกษตร เป็นต้น ซึ่งล้วนแต่เป็นการทดลองทางกายภาพ (Physical Experiment) ซึ่งมักจะเกิดความผิดพลาดที่เกิดจากการทดลอง เนื่องจากปัจจัยด้านสภาพแวดล้อม ดังนั้นการทดลองในแต่ละครั้งอาจจะได้ผลลัพธ์ที่แตกต่างกันหรือเหมือนกัน ทั้งที่ให้ค่าตัวแปรเข้าชุดเดียวกัน (Non-Deterministic) การมีข้อผิดพลาดจากการทดลอง นั้นทำให้เกิดความซับซ้อนในการวิเคราะห์ข้อมูลและการสร้างรูปแบบจำลอง ดังนั้นผู้วิจัยต้องเลือกปัจจัยที่เกี่ยวข้องในการทดลองจำนวนน้อยๆ เพื่อง่ายในการสำรวจตรวจสอบความสัมพันธ์ระหว่างตัวแปรเข้าและผลลัพธ์ หรืออีกทางเลือกหนึ่งผู้ทำการทดลองต้องใช้ทางสถิติเข้ามาช่วยในการวางแผนการทดลอง การใช้สถิติในการวางแผนการทดลองจะอยู่บนพื้นฐานของตัวแบบทางสถิติ (Statistical Model) แผนการทดลองที่ดี จะส่งผลต่อประสิทธิภาพของตัวแบบทางสถิติที่สร้างขึ้น การวางแผนการทดลองสำหรับการทดลองทางกายภาพมีหลายแบบ เช่น Fractional Factorial Design บน ANOVA Model, Optimum Design บน Regression Model เป็นต้น ซึ่งเป็นตัวแบบที่นิยมใช้กันมาก การทดลองทางกายภาพ จะมีค่าใช้จ่ายสูงและเปลืองเวลา เพราะการทดลองทางกายภาพทำได้ยากหรือสิ้นเปลืองทรัพยากรในการทดลอง เพราะต้องคำนึงถึงปัจจัยที่เกี่ยวข้องหลากหลายปัจจัย (Fang, Li and Sudjianto, 2005)

ปัจจุบันคงปฏิเสธไม่ได้ว่า “คอมพิวเตอร์” ได้เข้ามามีบทบาทสำคัญยิ่งต่อการทำงานไม่ว่าจะเป็นสาขาอาชีพใด จะเห็นได้ว่าคอมพิวเตอร์หรือสมองกลอัจฉริยะนี้ได้เข้ามาพัฒนาและช่วยเพิ่มประสิทธิภาพการทำงานของมนุษย์ได้มากมายมหาศาล คุณสมบัติพิเศษอันโดดเด่นไม่ว่าจะเป็นประสิทธิภาพในการจัดเก็บข้อมูล ความรวดเร็วในการวิเคราะห์หรือประมวลผลและความสามารถในการทำงานซ้ำๆ กันอย่างไม่เหน็ดเหนื่อย จึงมีการนำคอมพิวเตอร์มาประยุกต์ใช้ให้เกิดประโยชน์และเหมาะสมในแต่ละงานอย่างไม่หยุดยั้ง และสิ่งเหล่านี้คือที่มาของการคิดค้นการจำลองการทำงานด้วยโปรแกรมคอมพิวเตอร์เพื่อมาใช้กับงานทางด้านวิทยาศาสตร์ และเทคโนโลยี โดยเฉพาะอย่างยิ่งใช้ในการจำลองการทดลองที่ทำได้ยากหรือทำไม่ได้จริงทางกายภาพ เช่น โปรแกรมการพยากรณ์อากาศ โปรแกรมการศึกษาพฤติกรรมของโครงสร้าง (ตึกและชิ้นส่วนอุตสาหกรรม) ภายใต้ภาวะกดดันและเงื่อนไขต่างๆ เครื่องจำลองปอเก็บสำหรับวิศวกรรมปิโตรเคมีรวมไปถึงการทำงานของนิวเคลียร์ เป็นต้น เพราะข้อจำกัดต่างๆ ไม่เอื้ออำนวยให้ทำการทดลองทางกายภาพได้จริง เช่น มีงบประมาณจำกัด มีผลกระทบต่อสิ่งแวดล้อม ข้อจำกัดทางด้านกฎหมาย เทคโนโลยี ณ ปัจจุบันที่ยังไม่สนับสนุนการทดลองจริง เป็นต้น ทำให้เกิดการศึกษาดูแลที่เรียกว่า การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment: CSE) ขึ้นมา ซึ่งการจำลองการทดลองด้วยคอมพิวเตอร์พัฒนาก้าวหน้าเรื่อยๆ ตามการพัฒนาทางเทคโนโลยีคอมพิวเตอร์ฮาร์ดแวร์และซอฟต์แวร์ที่เจริญก้าวหน้าขึ้น

ประโยชน์ของ CSE มีด้วยกันมากมายหลายประการ ดังตัวอย่างต่อไปนี้

1. ลดการสูญเสียทรัพยากร เช่น การสร้างชิ้นส่วนปีกเครื่องบิน การซ่อมรถ การทดลองยา ถ้าใช้การจำลองเหตุการณ์จะไม่ต้องเสียค่าวัสดุอุปกรณ์ ไม่ต้องเสียทหาร ไม่ต้องเสียชีวิตคนใช้
2. ประหยัดเวลา ประหยัดทรัพย์ เพราะการทดลองในห้องทดลอง บนโต๊ะทำงาน หรือทดลองด้วยคอมพิวเตอร์จึงทำให้ค่าใช้จ่ายน้อย การแก้สมการต่างๆ ในบางครั้งการทดลองแทนค่าบนคอมพิวเตอร์จะเร็วกว่าการแก้สมการตามปกติ
3. หากสามารถสร้างแบบจำลองที่ดีได้ด้วยคอมพิวเตอร์ เราก็จะเห็นภาพหรือตัวเลขการเปลี่ยนแปลงของตัวแปรต่างๆ ในระบบเป็นขั้นเป็นตอนได้ดีกว่าการหาคำตอบสุดท้ายแบบธรรมดา

ในงานวิจัยนี้ การจำลองการทดลองด้วยคอมพิวเตอร์ หมายถึง การใช้โปรแกรมคอมพิวเตอร์เพื่อจำลองรูปแบบของระบบใดๆ โดยอาศัยข้อมูลต่างๆ ที่เกี่ยวข้อง (Input Variable) ร่วมกับสมการทางคณิตศาสตร์ที่อธิบายการทำงานของระบบโดยได้ค่าของตัวแปรตาม (Output Variable) เพื่อใช้เป็นตัวแทนของการทำงานของระบบที่สนใจศึกษา

จากนิยามและคณะ ได้อธิบายกระบวนการในการจำลองการทดลองด้วยคอมพิวเตอร์ ว่ามีส่วนประกอบด้วยกันทั้งหมด 3 ส่วน ดังแสดงไว้ในภาพคือ

1. Experimental Design คือ การออกแบบและการวางแผนการทดลอง ซึ่งประกอบด้วยค่าของปัจจัยที่เกี่ยวข้อง (ตัวแปรเข้า) ระดับต่างๆ กล่าวคือ การออกแบบกลุ่มของค่าของตัวแปรเข้าที่ต้องการศึกษาและตามข้อจำกัดของการทดลอง (จำนวนวัน)

2. Simulation Routine หรือ กล่องดำ (Black Box) คือ กระบวนการทำงานที่เราไม่สามารถมองเห็นและคาดว่าเกิดอะไรขึ้นภายใน ซึ่งกระบวนการในกล่องดำ นี้จะใช้รูปแบบสมการทางคณิตศาสตร์ที่มีความซับซ้อน เพื่อให้ได้ผลลัพธ์ออกมา ซึ่งผลลัพธ์ที่ได้นี้จะแปรผันตามระดับค่าต่างๆ

ของปัจจัยที่เกี่ยวข้องที่ป้อนเข้าไป โดยส่วนใหญ่การส่งจุดทดลองในแต่ละรัน เข้าไปในกล่องดำนั้นจะใช้ งบประมาณและเวลานานมาก ทำให้ต้องมีการวางแผนการทดลองก่อนส่งไปประมวลผลที่กล่องดำ เพื่อให้ได้กลุ่มของตัวแปรตามที่มีประสิทธิภาพ

3. Approximation Model คือ การสร้างโมเดลเพื่อการประมาณเกิดจากการนำแผนการทดลอง กับผลลัพธ์ที่ได้หลังจากกระบวนการในกล่องดำ มาสร้างรูปแบบความสัมพันธ์ที่เหมาะสม ให้สามารถ นำไปใช้ประโยชน์ต่อไป จุดมุ่งหมายของการสร้างโมเดลเพื่อการประมาณ คือ การให้ได้มาซึ่งโมเดลที่ สามารถพยากรณ์ผลลัพธ์หรือตัวแปรตามได้โดยที่ไม่จำเป็นต้องส่งจุดทดลองไปประมวลผลบนกล่อง ดำ ทำให้ประหยัดงบประมาณและเวลาในการทำงาน (Simpson, *et al.* , 2001)

### การวางแผนการทดลอง (Experimental Design)

การวางแผนการทดลองมีความสำคัญอย่างมาก ทั้งนี้เพราะแผนการทดลองที่ดีย่อมนำมาซึ่ง ผลลัพธ์หรือโมเดลที่มีความน่าเชื่อถือสูง เพื่อสามารถใช้เป็นตัวแทนของระบบจริงได้ ซึ่งการวาง แผนการทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ จะแตกต่างจากการวางแผนการทดลอง ทางกายภาพ เพราะการจำลองการทดลองด้วยคอมพิวเตอร์จะเป็นรูปแบบที่แน่นอน (Deterministic) คือ ที่แต่ละครั้งใส่ตัวแปรเข้าใดๆ มีค่าเหมือนกันไปก็จะได้ผลลัพธ์หรือตัวแปรตามค่าเดิมทุกครั้ง (Simpson, *et al.*, 2001, Rungrattanaubol and Na-udom, 2007) การจำลองการทดลองด้วย คอมพิวเตอร์จะไม่มีปัจจัยภายนอกที่จะทำให้ค่าเปลี่ยนแปลงเหมือนการทดลองทางกายภาพ ทำให้การ วางแผนการทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ก็จะแตกต่างกับการวางแผนการ ทดลองทางกายภาพ แชคและคณะได้เสนอ แนวคิดของการวางแผนการทดลอง สำหรับ CSE ว่าจะต้อง มีความแตกต่างจากแผนการทดลองแบบคลาสสิก เนื่องจากผลลัพธ์จาก CSE เป็นแบบแน่นอน จึงไม่ จำเป็นต้องมีการทำซ้ำ (Replicates) ดังนั้นการออกแบบการทดลองด้วยคอมพิวเตอร์จะเป็นการ ออกแบบโดยใช้แนวคิดแบบใหม่ ซึ่งเป็นแบบกระจายจุดทดลองให้ทั่ว Design Space ให้ได้มากที่สุด ซึ่ง เรียกว่า Space Filling Design ซึ่งเหมาะกับรูปแบบการทำงานแบบแน่นอน (Deterministic) (Sacks, *et al.*, 1989) การวางแผนการทดลองในลักษณะนี้เป็นแบบ Search Based ซึ่งใช้อัลกอริทึมการค้นหา มาเลือกแผนการทดลองที่ดีที่สุดที่กระจายจุดทดลอง ครอบคลุม Design Space ทั้งหมด ภายใต้กับ เกณฑ์หาค่าเหมาะสม (Optimality Criteria)

ดังได้กล่าวไปแล้วในการวางแผนการทดลองสำหรับ CSE นั้น สิ่งหนึ่งที่ผู้ออกแบบต้องคำนึง เสมอ คือ ลักษณะการแบ่งช่วงของค่าข้อมูลของแต่ละตัวแปรเข้า ซึ่งในการวางแผนนั้น ได้มีผู้วิจัยหลาย ท่านได้นำเสนอรูปแบบหรือคลาสของการวางแผนไว้หลายค่า

### คลาสของการวางแผน (Class of Design)

รูปแบบของการวางแผนการทดลองสำหรับ Space Filling ของการจำลองการทดลองด้วย คอมพิวเตอร์นั้นมีหลายแบบ โดยในงานวิจัยนี้จะยกตัวอย่างมาทั้งหมด 3 แบบคือ Latin Hypercubes

Design (LHD) Hammersley Sequence Sampling (HSS) และ Uniform Design ซึ่งเป็นแบบที่นิยมใช้ ในการวางแผนการทดลองในปัจจุบัน

### 1. Latin Hypercubes Design (LHD)

ใช้ครั้งแรกโดย McKay, et al. ใน 1979 เป็นเมทริกซ์ ( $X$ ) มีมิติ  $n \times d$  โดยที่  $n$  คือ จำนวน รัน และ  $d$  คือ จำนวนตัวแปรเข้า ซึ่งแต่ละหลัก  $d$  ประกอบด้วยรันตั้งแต่  $\{1, 2, \dots, n\}$  เรียงสลับเปลี่ยนกัน และสามารถเทียบกับค่าจริงของแต่ละตัวแปรได้ เมื่อใช้การแบ่งช่วงของแต่ละตัวแปรในพื้นที่การ ออกแบบ (Design Space) แล้ว จะได้เป็นค่าที่ไม่ซ้ำกันในแต่ละคอลัมน์ ในการแบ่งช่วงข้อมูลตาม รูปแบบของ Latin Hypercubes นั้นค่าในแต่ละช่วงหาได้จาก จะนิยมแปลงให้อยู่ในช่วง  $[0, 1]$  เพื่อความ สะดวก รวดเร็ว ในการคำนวณและสามารถเทียบกับค่าจริงได้ง่าย LHD สามารถสร้างได้โดยแนวคิด ของการสุ่มแบบแบ่งชั้นภูมิ (Stratified Sampling) (McKay, Beckman and Conover, 1979) ทำให้ มั่นใจโดยทุกๆส่วนย่อยของ Design Space แบ่งตัวแปรเข้าในระดับที่เท่าๆ กัน มีความเป็นไปได้เท่าๆ กัน การเลือกตัวอย่างบน Latin Hypercubes เป็น

$$X_{ij} = \frac{\pi_{ij} - U_{ij}}{n}$$

โดยที่  $\pi_{ij}$  เป็นส่วนของเมทริกซ์  $n \times d$  ที่ประกอบด้วยสตมภ์  $\pi_i, (i=1, 2, \dots, d)$  โดยแต่ละสตมภ์  $\pi_i, (i=1, 2, \dots, d)$  จะสุ่มเรียงสลับเปลี่ยนโดยอิสระด้วย  $1, 2, \dots, n$  และ  $U_{ij}$  คือค่า  $n \times d$  สำหรับ i.i.d. uniform  $U[0, 1]$  สุ่มค่าโดยอิสระของ  $\pi_{ij}$

$X_1$	$X_2$	$X_3$	$X_4$
1	2	4	3
2	4	1	4
3	5	3	1
4	3	2	2
5	1	5	5

### 2. Hammersley Sequence Sampling (HSS)

Hammersley Sequence Sampling เตรียมการออกแบบการทดลองที่มีคลาดเคลื่อนต่ำ สำหรับการกำหนดจุด  $n$  ในมิติ  $k$  ของ hypercube (Kalagnanam and Diwekar, 1997) เตรียม คุณสมบัติความเป็นอันหนึ่งอันเดียวได้ดีกว่าบน  $k$  มิติ มากกว่า Latin Hypercubes คลาดเคลื่อนต่ำมี ความหมายว่ามีความเป็นอันหนึ่งอันเดียวกันของการกระจายของจุดใน Design Space

### 3. Uniform Design (UD)

Uniform Design เตรียมจุดที่ออกแบบการกระจายอย่างเดียวกันในโดเมนของการทดลอง Uniform Design เป็นรูปแบบหนึ่งของ Fractional Factorial Design ซึ่งเพิ่มคุณสมบัติความเป็นแบบ

เดียวกัน นิยมใช้กันอย่างแพร่หลายตั้งแต่ปี 1980 (Fang, 1980) ถ้าโดเมนของการทดลองมีจำกัด Uniform Design จะเหมือน Latin Hypercubes เมื่อโดเมนของการทดลองต่อเนื่อง หลักที่แตกต่างของการออกแบบคือใน Latin Hypercubes จะใช้การสุ่มแต่ Uniform Design จะเลือกจากค่าตรงกลาง นอกจากนี้ Latin Hypercubes ต้องการความสมดุลหนึ่งมิติสำหรับทุกระดับของตัวแปร แต่ Uniform Design ต้องการความสมดุลหนึ่งมิติสำหรับทุกระดับของตัวแปรแล้วยังต้องการความเป็นหนึ่งเดียวกันใน  $n$  มิติ ดังนั้นการออกแบบจะเหมือนกันในหนึ่งมิติแต่จะยิ่งแตกต่างกันมากขึ้นเมื่อมิติมากขึ้น การสร้าง Uniform Design ทำได้ยากกว่าและมีข้อจำกัดมากกว่า LHD

โดยในงานวิจัยนี้ ผู้วิจัยจะเลือกใช้ คลาสของการวางแผนแบบ Latin Hypercubes Design (LHD) เพราะเป็นที่นิยมในงานด้าน CSE และมีความยืดหยุ่น เมื่อกำหนดมิติปัญหา  $n \times d$  ใดๆ

### กล่องดำ หรือ Black Box

Simulation Routine หรือ กล่องดำ (Black Box) คือ กระบวนการทำงานที่เราไม่สามารถมองเห็นและคาดว่าเกิดอะไรขึ้นภายใน ซึ่งกระบวนการในกล่องดำ เป็นกระบวนการที่มีการทำงานซับซ้อน โดยมักจะจำลองการทำงานของระบบการทำงานหรือปรากฏการณ์บางอย่างใดอย่างหนึ่ง ในรูปแบบของสมการคณิตศาสตร์ที่มีความซับซ้อนและยากต่อการอธิบายลักษณะการทำงาน ในมุมมองของผู้ใช้ กล่องดำนี้จะเปรียบเสมือนกล่องที่เราไม่ทราบได้ เพื่อศึกษากลไกดังกล่าว ในกลุ่มช่วงตัวแปรที่กำหนด  $X$  เพื่อได้ค่า  $Y$  ออกมา ซึ่งผลลัพธ์ที่ได้นี้จะแปรผันตามระดับค่าต่างๆ ของปัจจัยที่เกี่ยวข้อง ( $X$ ) ที่ป้อนเข้าไป กล่องดำจะเกิดจากผู้เชี่ยวชาญทางด้านใดด้านหนึ่งที่ศึกษาและพัฒนาขึ้น เช่น กล่องดำจำลองการทำงานของปฏิกิริยาปฏิกิริยาปฏิกิริยา กล่องดำจำลองการทำงานของเครื่องกลึงโลหะ เป็นต้น ซึ่งเป็นระบบที่ซับซ้อนจำลองปรากฏการณ์ที่จะเกิดเมื่อใส่ค่าตัวแปรเข้าไป การประมวลผลกล่องดำในแต่ละครั้งจะต้องใช้งบประมาณและเวลามาก จึงเป็นเหตุให้ต้องมีการวางแผนการทดลองในขั้นตอนก่อนหน้าเพื่อประหยัดงบประมาณและเวลา และเพื่อนำผลลัพธ์ดังกล่าว กับตัวแปรเข้า นำมาสร้างเป็นโมเดลประมาณค่า เพื่อใช้พยากรณ์ค่าคำตอบ โดยไม่ต้องไปประมวลผลกับกล่องดำอีก



### โมเดลประมาณค่า (Approximation Models)

โมเดลประมาณค่า (Approximation Models) คือการนำกลุ่มตัวแปรเข้า (Input) และค่าผลลัพธ์ (Output) ที่ได้จากกล่องดำ มาสร้างเป็นโมเดลเพื่อประมาณค่า เพื่อใช้พยากรณ์ค่าคำตอบ เทคนิคที่ใช้ในการสร้างโมเดลมีด้วยกันหลายตัว ในที่นี้จะยกตัวอย่างเทคนิคที่ใช้กันทั่วไป 2 แบบ คือ Response Surfaces Methodology และ Kriging Model

## Response Surfaces Methodology (RSM)

Response Surfaces เป็นเทคนิคที่เป็นที่นิยมใช้ โดยในขั้นต้นพัฒนาขึ้นเพื่อการวิเคราะห์สำหรับการวางแผนการทดลองทางกายภาพ (Box and Wilson, 1951) ในการวางแผนการทดลองนั้นจะประกอบด้วยตัวแปรหลายตัวโดยเฉพาะการออกแบบโมเดลแบบ Polynomial Response Surface Models ที่ถูกใช้อย่างมีประสิทธิภาพสำหรับสร้างโมเดลประมาณค่า ที่มีปัจจัยในหลากหลาย หลายงานประยุกต์การออกแบบโมเดลแบบ Second-Order Polynomial Response Surface Models มีรูปแบบดังนี้

$$\hat{y} = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i < j}^k \sum_{i < j} \beta_{ij} x_i x_j$$

ซึ่งพารามิเตอร์  $\beta_0, \beta_i, \beta_{ii}, \beta_{ij}$  คำนวณได้จากการถดถอยวิธีกำลังสองน้อยที่สุด (Least Squares Regression) ส่วน  $\hat{y}$  เป็นค่าพยากรณ์จากสมการที่สร้างจากสมการถดถอยด้วยวิธีกำลังสองน้อยที่สุด จากค่าจริง  $y(x)$  คือ ค่าจริงของตัวแปรออกโดยใช้สมการ

$$\hat{\beta} = [X'X]^{-1} X'y$$

โดย  $X$  คือ เมทริกซ์การวางแผนสำหรับจุดตัวอย่าง  $X'$  คือ ทรานสโพสของ  $X$  และ  $y$  คือเวกเตอร์หลักที่ประกอบด้วยคำตอบสนองของแต่ละจุดตัวอย่าง Polynomial Response Surface Models สร้างได้ง่ายและสะดวกเนื่องจากมีโปรแกรมสำเร็จรูปที่รองรับการทำงานของ RSM

## Kriging Model

Kriging Model เป็นเทคนิคที่ในขั้นต้นพัฒนาเพื่อประยุกต์ใช้ในทางด้านธรณีสถิติ (Geostatistics) การสร้าง Kriging Model ใช้หลักแนวคิดคือ ตัวแปรตาม ( $y$ ) สามารถเขียนอยู่ในรูปแบบฟังก์ชันโพลีโนเมียล (Polynomial Function) ของชุดตัวแปรเข้า รวมกับความคลาดเคลื่อนที่เกิดจากกระบวนการเฟ้นสุ่ม (Stochastic Process) ซึ่งแทนด้วย  $Z(x)$  ความคลาดเคลื่อนนี้มีค่าเฉลี่ยเท่ากับ 0 และความแปรปรวนร่วมระหว่าง  $Z(u)$  และ  $Z(v)$  ใดๆ ในกระบวนการเฟ้นสุ่มมีค่าเท่ากับ  $V(u,v) = \sigma^2 R(u,v)$  เมื่อ  $\sigma^2$  แทนความแปรปรวนของระบบและ  $R(u,v)$  คือฟังก์ชันสหสัมพันธ์ (Correlation Function) และสามารถแทนความสัมพันธ์ด้วยรูปแบบต่าง ๆ ซึ่งรูปแบบของฟังก์ชันสหสัมพันธ์เหล่านี้สามารถดูได้จาก Koehler and Owen (1996) โดยทั่วไปแล้วตัวแปรออก  $y$  สามารถเขียนในรูปแบบทางคณิตศาสตร์ได้ดังนี้

$$y = \sum_{i=1}^k \beta_i f_i(x) + Z(x)$$

โดย  $Z(x)$  ถูกสมมติให้เป็นจริงของกระบวนการสุ่มด้วยค่ากลางศูนย์และฟังก์ชันความสัมพันธ์เชิงระยะทาง (Spatial Correlation Function) ตามสมการ

$$\text{Cov}[Z(x_i), Z(x_j)] = \sigma^2 R(x_i, x_j)$$



โดย  $\sigma^2$  คือ ความแปรปรวนของกระบวนการ ฟังก์ชันความสัมพันธ์สามารถเลือกใช้ได้หลากหลายวิธี  
อย่างไรก็ตามวิธีที่เป็นที่นิยม คือ Gaussian Correlation (Sacks, et al, 1989) รายละเอียดการสร้าง  
Kriging Model ได้อธิบายไว้ในบทที่ 3

## อัลกอริทึมการสืบค้นและเกณฑ์การเลือกค่าที่เหมาะสม

การออกแบบการทดลองที่ดีจะส่งผลให้ได้โมเดลจำลองที่มีความแม่นยำและมีประสิทธิภาพใน  
การพยากรณ์ การออกแบบการทดลองนั้นจะจำกัดด้วยจำนวนตัวแปรและจำนวนครั้งในการรัน เพราะ  
การทำการทดลองในแต่ละครั้งจะใช้เวลาและเงินเป็นจำนวนมาก การออกแบบการทดลองที่ดีที่สุดจะมา  
จาก “การเลือกทางเลือกที่ดีที่สุด” ซึ่งจะเรียกว่า การทำงานแบบการหาค่าเหมาะสม (Optimization)

ในงานวิจัยนี้ได้นำอัลกอริทึมการสืบค้นมาใช้ในการเลือกแผนการทดลองที่ดีที่สุด เพราะจำนวน  
แผนการทดลองที่เป็นไปได้มีจำนวนมหาศาล เป็นไปไม่ได้ที่จะนำทุกๆ แผนการทดลองมาทำการทดลอง  
จริง แผนการทดลองที่ได้จากอัลกอริทึมการสืบค้น จะเป็นแผนการทดลองที่เหมาะสมที่สุด เพื่อใช้เป็น  
ตัวแทนของแผนการทดลองทั้งหมดให้ได้ดี อัลกอริทึมการสืบค้นที่นิยมกันในปัจจุบันนั้นมีหลายแบบ  
เช่น Simulated Annealing: SA, Genetic Algorithm: GA, Tabu Search: TA เป็นต้น โดยอัลกอริทึม  
การสืบค้นจะต้องทำงานควบคู่ไปกับเกณฑ์การเลือกแผนการทดลองที่เหมาะสม (Optimality Criteria)  
ในที่นี้จะกล่าวถึงสองอัลกอริทึมการสืบค้น คือ อัลกอริทึมเจเนติก (Genetic Algorithm: GA) และ  
อัลกอริทึมซิมูเลตเตดแอนนีลลิ่ง (Simulated Annealing :SA)

### อัลกอริทึมเจเนติก (Genetic Algorithm: GA)

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) เป็นเทคนิคสำหรับค้นหาผลเฉลย (Solutions)  
หรือคำตอบโดยประมาณของปัญหา โดยอาศัยหลักการจากทฤษฎีวิวัฒนาการจากชีววิทยา และ การ  
คัดเลือกตามธรรมชาติ (Natural Selection) นั่นคือ สิ่งมีชีวิตที่เหมาะสมที่สุดจึงจะอยู่รอด กระบวนการ  
คัดเลือกได้เปลี่ยนแปลงสิ่งมีชีวิตให้เหมาะสมยิ่งขึ้น ด้วยตัวปฏิบัติการทางพันธุกรรม (Genetic  
Operator) เช่น การสืบพันธุ์ (Inheritance หรือ Reproduction), การกลายพันธุ์ (Mutation), การ  
แลกเปลี่ยนยีน (Recombination) ขั้นตอนวิธีเชิงพันธุกรรมเป็นการจำลองการทำงานทางคอมพิวเตอร์  
เพื่อแก้ปัญหาการหาค่าเหมาะสมที่สุด (Optimal Solution) โดยการแทนคำตอบที่มีอยู่ในลักษณะ  
โครโมโซม (Chromosomes) แล้วปรับปรุงคำตอบแต่ละชุด (เรียกว่า Individual) ด้วยวิธีการต่าง ๆ ซึ่ง  
เกี่ยวข้องกับการวิวัฒนาการ (Evolutionary Operation) การเปลี่ยนแปลงยีนแบบสุ่ม ด้วยตัวปฏิบัติการ  
ทางพันธุกรรม (Evolutionary Operator) เพื่อให้ได้คำตอบที่ดีขึ้น โดยทั่วไปจะแทนคำตอบด้วย  
เลขฐานสอง (สายอักขระของเลข 0 และ 1) การวิวัฒนาการ (Evolution) เพื่อหาคำตอบที่เหมาะสมที่สุด (The  
Fitness Solution) จะเริ่มจากประชากรที่ได้จากการสุ่มทั้งหมดและจะทำการเป็นรุ่น ๆ ในแต่ละรุ่นคำตอบ  
หลายชุดจะถูกสุ่มเลือกขึ้นมาเปลี่ยนแปลง ซึ่งอาจจะทำให้เกิดการกลายพันธุ์ หรือสับเปลี่ยนยีนระหว่าง  
กัน จนได้ประชากรรุ่นใหม่ ที่มีค่าความเหมาะสม (Fitness) มากขึ้น การวิวัฒนาการนี้จะทำไปเรื่อย ๆ  
จนกระทั่งพบคำตอบที่มีค่าความเหมาะสมตามต้องการ (Wikipedia, 2007) GA ที่จะแสดงเป็นการ

ประยุกต์ GA นำเสนอโดย Liefvendel และ Stocki (2006) เพื่อสืบค้นหาแผนการทดลองที่เหมาะสมที่สุดบนคลาสการออกแบบ LHD จะปรับปรุงวิธีการสร้างทายาท (Generating Offsprings) และวิธีการกลายพันธุ์ (Generating Mutation) GA จะเริ่มต้นด้วยการสร้างประชากรเริ่มต้นประกอบด้วย  $m$  LHD สำหรับเมทริกซ์  $n \times d$  รุ่นใหม่จะถูกสร้างโดยการ Crossover และประยุกต์การกลายพันธุ์ การค้นสืบค้นจะทำงานไปเรื่อยๆ จะหยุดเมื่อตรงกับเกณฑ์ ขั้นตอนโดยละเอียดมีดังต่อไปนี้ จะให้  $m$  คือขนาดของประชากรทั้งหมด,  $m_i$  คือ จำนวนของรูปแบบที่สร้างในแต่ละรุ่น ( $m_i = 10 \times m$ ),  $\omega$  คือ อัตราการกลายพันธุ์

### อัลกอริทึมซิมูเลตเตดแอนนีลลิ่ง (Simulated Annealing :SA)

Morris และ Mitchell ได้ดัดแปลงอัลกอริทึมสืบค้น Simulated Annealing เพื่อหาแผนการทดลองที่ดีที่สุด โดยใช้เกณฑ์ค่าที่เหมาะสมที่สุดควบคุมไปด้วยคือ เกณฑ์  $\phi$  แนวคิดของอัลกอริทึม SA เป็นหลักการสลับค่าสองจุดในแผนการทดลองโดยการสุ่ม (Perturbation) ของจุดทดลองในแผนการทดลอง (Morris and Mitchell, 1995) อย่างไรก็ตามในอัลกอริทึม SA ที่มีการสลับค่าจะบอกไม่ได้ว่าดีขึ้นหรือแย่ลง ต้องใช้เกณฑ์  $\phi$  ในขั้นตอนสืบค้นเป็นตัวบอก ขั้นตอนของอัลกอริทึม SA มีดังนี้

#### ขั้นตอนที่ 1

กำหนดค่าเริ่ม  $l_{max}$  (จำนวนครั้งมากที่สุดของการสลับแล้วปรับปรุงดีขึ้น),  $t_0$  (อุณหภูมิความเย็นเริ่มต้น),  $C_i$  (ตัวประกอบใช้ลด  $t_0$  เมื่อสลับแล้วไม่ดีขึ้นในหลักเกณฑ์  $\phi$ ) ค่า  $t_0$  จะเป็นตัวควบคุมความน่าจะเป็นในการสลับแล้วไม่ได้ปรับปรุงดีขึ้น กระบวนการสืบค้นดำเนินการไปเรื่อย ถ้าไม่มีการปรับปรุงให้ดีขึ้นก็จะลดอุณหภูมิ โดยคูณ  $t_0$  ด้วย  $C_i$  โดยถ้าค่า  $t_0$  มีค่าสูง จะมีโอกาสได้แผนการทดลองที่ไม่ปรับปรุงเป็นไปได้อย่างสูง ทำให้เราต้องลดค่าอุณหภูมิโดยการคูณด้วย  $C_i$  ซึ่งค่า  $C_i$  จะเป็นค่าน้อยขึ้นอยู่กับขนาดมิติแผนการทดลอง

#### ขั้นตอนที่ 2

สุ่มสร้าง LHD ขนาด  $n \times d$  แล้วให้  $D_{best} = D, t = t_0$

#### ขั้นตอนที่ 3

กำหนด  $l = 1, Label = 0$

#### ขั้นตอนที่ 4

กำหนด  $D_{lv} = D$  สุ่มเลือกหลัก ( $j$ ) ของเมทริกซ์  $D_{lv}$  และสุ่มสองค่าในหลัก ( $j$ ) แล้วสลับค่ากัน  $D_{aj} \leftrightarrow D_{bj}$  ตัวอย่างดังภาพ

1	2	4	0	→	1	2	4	0
3	4	0	3		3	0	0	3
2	1	3	4		2	1	3	4
4	0	1	2		4	4	1	2
0	3	2	1		0	3	2	1

ภาพ แสดงการสลับค่า 2 ค่าในหลักที่ 2 ของ  $5 \times 4$  LHD

จากภาพเป็นการสุ่มสลับค่าในหลักที่ 2 และ 4 ของแถวที่ 2

ขั้นตอนที่ 5

ถ้า  $\phi_p(D_{try}) < \phi_p(D)$  หรือความน่าจะเป็น  $e^{-(\phi_p(D_{try}) - \phi_p(D))/t}$  ให้  $D = D_{try}$  และ  $Label = 1$

ขั้นตอนที่ 6

ถ้า  $\phi_p(D_{try}) < \phi_p(D_{best})$  แล้วกำหนด  $l = 1$  และ  $D_{best} = D_{try}$  ถ้าไม่ ให้  $l = l + 1$

ขั้นตอนที่ 7

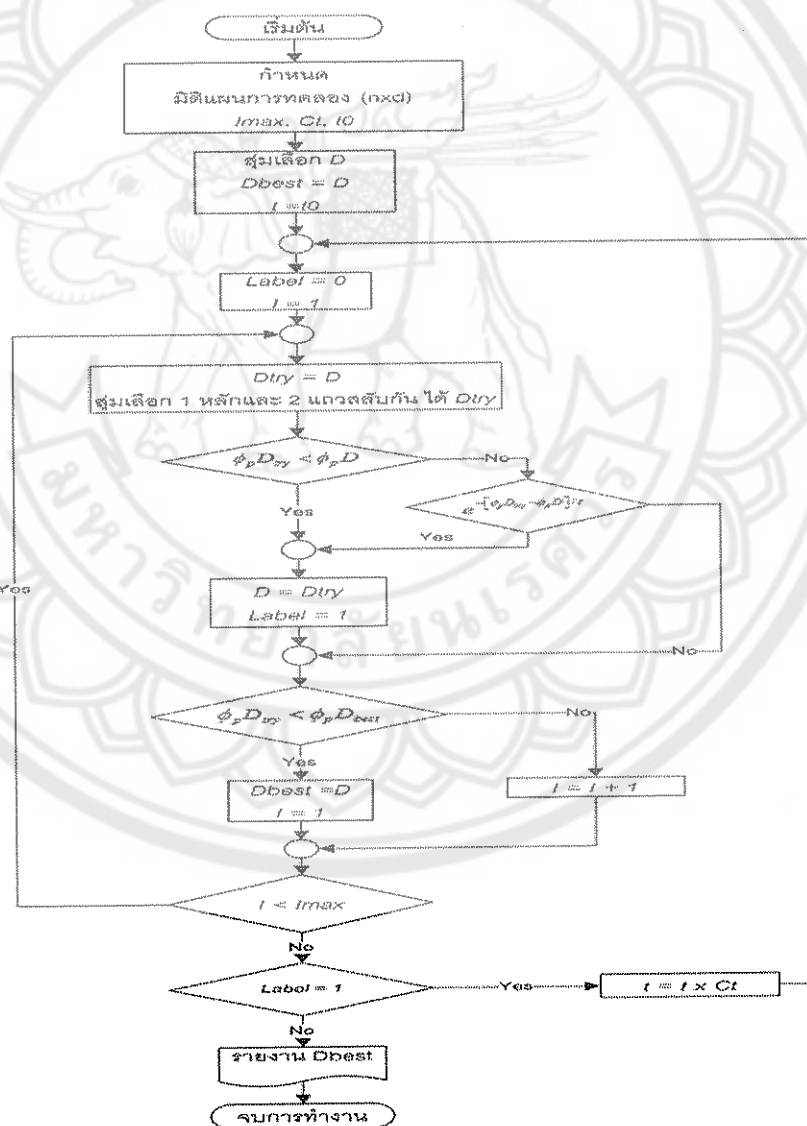
ถ้า  $l < l_{max}$  ให้กลับไปขั้นตอนที่ 4

ขั้นตอนที่ 8

ถ้า  $Label = 1$  ให้กำหนด  $t = t \times C_t$  แล้วกลับไปขั้นตอนที่ 3

ขั้นตอนที่ 9

จบกระบวนการและรายงานค่า  $D_{best}$  ฟังก์ชันของกระบวนการนี้ตั้งภาพ



ภาพแสดงฟังก์ชันของ SA

## เกณฑ์การเลือกค่าที่เหมาะสม

จากอัลกอริทึมที่อธิบายไว้ในข้างต้นนั้นหลักการทำงานของอัลกอริทึมการสืบค้นต้องทำงานร่วมกับเกณฑ์การเลือกค่าที่เหมาะสม (Optimality Criteria) เสมอ ในการค้นหาแผนการทดลองที่ดีนั้นผู้วิจัยหลายท่านได้เสนอเกณฑ์การเลือกไว้หลายเกณฑ์ เช่น Maximin Distance Criterion, Entropy Criterion และ Centered L<sub>2</sub> Discrepancy Criterion โดยในการวิจัยนี้จะเลือกใช้ Maximin Distance Criterion และ  $\phi_p$  Criterion มาเป็นเกณฑ์ ทั้งนี้เพราะลักษณะเกณฑ์ทั้งสองตัวนี้มีคุณสมบัติเด่น คือ ไม่ขึ้นต่อโมเดล (Model Independent) ทำให้สะดวกในการวางแผนการทดลอง เพื่อนำไปใช้กับการทดลองต่างๆ ไปได้

### แม็กซิมิน (Maximin)

โดยหลักการของเกณฑ์การเลือกค่าที่เหมาะสม แบบแม็กซิมิน (Maximin) นั้นจะต้องหาระยะห่างจุดทดลองบนแผนการทดลองก่อน การหาระยะห่างของจุดนั้นหาได้สองแบบคือ Rectangular Distance และ Euclidean Distance ดังสมการ

$$d(x_s, x_t) = \sum_{i=1}^k |x_{si} - x_{ti}| \quad \text{และ} \quad d(x_s, x_t) = \left[ \sum_{i=1}^k (x_{si} - x_{ti})^2 \right]^{1/2} \quad \text{ตามลำดับ}$$

งานวิจัยนี้ได้เลือกใช้ Euclidean Distance โดยจากแผนการทดลองขนาด  $n \times d$  เมื่อนำมาหาระยะห่างจุดจะได้เป็นเมทริกซ์ระยะห่างจุดขนาด  $n \times n$  เป็นเมทริกซ์แบบสมมาตรเส้นทแยงมุมจะมีค่าเป็น 0 ค่าระยะห่างที่น้อยที่สุดของเมทริกซ์ระยะห่างนี้ จะนำมาเป็นตัวเปรียบเทียบกับ เมทริกซ์ของแผนการทดลองอื่นๆ คือ ค่าที่น้อยที่สุดของเมทริกซ์ระยะห่างของแผนการทดลองใดมีค่ามากกว่ากัน (Maximin Design) จะเลือกแผนการทดลองนั้น ตัวอย่างเช่น แผนการทดลองขนาด  $9 \times 2$

0.375	0.875
0.125	0.375
0.625	0.125
0.250	0
1	0.250
0.875	0.625
0.500	0.500
0	0.75
0.75	1

หาค่าระยะห่างโดยเลือกใช้ Euclidean Distance ได้ดังนี้

$$d_{11} = \left( (0.375 - 0.375)^2 + (0.875 - 0.875)^2 \right) = 0$$

$$d_{12} = \left( (0.375 - 0.125)^2 + (0.875 - 0.375)^2 \right) = 0.5590$$

$$d_{21} = \left( (0.125 - 0.375)^2 + (0.375 - 0.875)^2 \right) = 0.5590 \quad \vdots$$

$$d_{98} = ((0.75 - 0)^2 + (1 - 0.75)^2) = 0.7906$$

$$d_{99} = ((0.75 - 0.75)^2 + (1 - 1)^2) = 0$$

ซึ่งได้เป็นเมทริกซ์ระยะห่าง ดังนี้

0	0.5590	0.7906	0.8839	0.8839	0.5590	0.3953	0.3953	0.3953
0.5590	0	0.5590	0.3953	0.8839	0.7906	0.3953	0.3953	0.8839
0.7906	0.5590	0	0.3953	0.3953	0.5590	0.3953	0.8839	0.8839
0.8839	0.3953	0.3953	0	0.7906	0.8839	0.5590	0.7906	1.1180
0.8839	0.8839	0.3953	0.7906	0	0.3953	0.5590	1.1180	0.7906
0.5590	0.7906	0.5590	0.8839	0.3953	0	0.3953	0.8839	0.3953
0.3953	0.3953	0.3953	0.5590	0.5590	0.3953	0	0.5590	0.5590
0.3953	0.3953	0.8839	0.7906	1.1180	0.8839	0.5590	0	0.7906
0.3953	0.8839	0.8839	1.1180	0.7906	0.3953	0.5590	0.7906	0

จากเมทริกซ์ระยะห่าง ต้องหาค่าที่น้อยที่สุด จากตัวอย่างคือค่า 0.3953 ค่านี้จะเป็นตัวเปรียบเทียบกับค่าระยะห่างที่น้อยที่สุดของแผนการทดลองอื่นๆ โดยจะพิจารณาว่าค่ามากกว่าจะเป็นแผนการทดลองที่เหมาะสมกว่า

ฟายฟี ( $\phi_p$ )

โดยหลักการของเกณฑ์การเลือกค่าที่เหมาะสม แบบ  $\phi_p$  นั้นเป็นวิธีที่พัฒนาจาก Maximin Distance Criterion นำเสนอโดย Morris และ Mitchell (1993) จะหาค่าได้จากสูตร

$$\phi_p = \left[ \sum_{j=1}^m J_j d_j^{-p} \right]^{1/p}$$

โดย  $d_j$  คือ ลิสต์ของระยะห่าง และ  $J_j$  คือ ลิสต์ของดัชนี ซึ่งเป็นจำนวนคู่ของจุดทดลองที่ถูกแยกด้วยระยะ  $d_j$  และ  $p$  เป็นเลขจำนวนเต็มบวก ซึ่งต้องมีค่ามากพอ จากงานวิจัยก่อนหน้าค่า  $p$  ที่มากพอคือ 5 จากตัวอย่างแผนการทดลองข้างต้น เมื่อหาระยะห่างจุดได้เป็นเมทริกซ์ระยะห่างขนาด  $n \times n$  คือ  $(9 \times 9)$  แล้ว ต้องนำมาเรียงลำดับจากน้อยไปมาก แต่ก่อนจะเรียงลำดับ จะเห็นว่าเป็นเมทริกซ์ที่ได้นั้นเป็นเมทริกซ์สมมาตร (Symmetric Matrix) ซึ่งจะคำนวณเฉพาะส่วนด้านบนของเส้นทแยง เพื่อลดเวลาในการคำนวณ โดยที่ความถูกต้องของคำตอบยังเหมือนกัน (Morris and Mitchell, 1995) ดังนั้นเมทริกซ์ที่จะพิจารณาเป็นดังนี้

0	0.5590	0.7906	0.8839	0.8839	0.5590	0.3953	0.3953	0.3953
0	0	0.5590	0.3953	0.8839	0.7906	0.3953	0.3953	0.8839
0	0	0	0.3953	0.3953	0.5590	0.3953	0.8839	0.8839
0	0	0	0	0.7906	0.8839	0.5590	0.7906	1.1180
0	0	0	0	0	0.3953	0.5590	1.1180	0.7906
0	0	0	0	0	0	0.3953	0.8839	0.3953
0	0	0	0	0	0	0	0.5590	0.5590
0	0	0	0	0	0	0	0	0.7906
0	0	0	0	0	0	0	0	0

นำมาเรียงระยะห่างน้อยไปมากจะได้ลิสต์ของระยะห่าง  $(d_1, d_2, \dots, d_m)$  ซึ่งเป็นเฉพาะส่วนบนของเส้นทแยงมุมของเมทริกซ์ระยะห่าง จะมีจำนวน  $m$  เป็นค่าตั้งแต่ 1 ถึง  $\binom{n}{2}$  และได้ลิสต์ของดัชนี  $(J_1, J_2, \dots, J_m)$  โดย  $J_i$  ลิสต์ของดัชนี ซึ่งเป็นจำนวนคู่ของจุดทดลองที่ถูกแยกด้วยระยะ  $d_i$  จากตัวอย่างจะได้ ลิสต์ของระยะห่างเป็น

(0.3953, 0.5590, 0.7906, 0.8839, 1.1180)

ลิสต์ของดัชนี  $(J_1, J_2, \dots, J_m)$  เป็น

(12, 8, 6, 8, 2)

แล้วนำค่า  $d_i$  และ  $J_i$  มาคำนวณตามสูตรดังกล่าวไปแล้ว โดยค่า  $\phi$  มีค่าน้อยแสดงว่าแผนการทดลองนั้นเป็นแผนการทดลองที่ดี

### สถิติที่เกี่ยวข้อง

เกณฑ์ที่ใช้วัดความแม่นยำในการพยากรณ์ คือ Root Mean Square Error: RMSE จะใช้วัดค่าความแม่นยำให้แผนการทดลองที่เกิดจากอัลกอริทึม SA กับ แผนการทดลองที่ได้จาก DACE MATLAB Toolbox ซึ่งเป็น Kriging Model คำนวณได้ตามสูตร

$$RMSE = \sqrt{\frac{\sum_{i=1}^k (y_i' - \hat{y}_i')^2}{k}}$$

โดยที่  $y_i'$  คือ ผลลัพธ์ที่เกิดจากกล่องดำ,  $\hat{y}_i'$  คือ ค่าพยากรณ์ที่ได้จาก Kriging Model,  $k$  คือ จำนวนจุดทดสอบ

## บทที่ 3

### วิธีดำเนินการวิจัย

---

ในบทนี้จะกล่าวถึงอุปกรณ์และเครื่องมือในการดำเนินการวิจัย โดยมีเนื้อหาประกอบด้วย

อุปกรณ์ที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการออกแบบ

Kriging Approximation Model

DACE (A MATLAB Kriging Toolbox Version 2.0)

#### อุปกรณ์ที่ใช้ในการดำเนินงานวิจัย

ประกอบด้วยเครื่องคอมพิวเตอร์จำนวน 1 เครื่อง ส่วนประกอบที่สำคัญคือ

CPU Intel Core 2 Duo T5500 1.66GHz

RAM 1GB DDR2

HDD 160GB SATA2

OS Microsoft Windows XP SP2

Software MATLAB 7.4.0.287 (R2007a)

DACE (A MATLAB Kriging Toolbox version 2.0)

#### เครื่องมือที่ใช้ในการออกแบบ

ในการปรับปรุงอัลกอริทึมของงานวิจัยในจะใช้การเขียนผังงานโครงสร้างในการอธิบายรายละเอียดการทำงานของอัลกอริทึมการสืบค้น

ผังงานโครงสร้าง (Structured Flowchart)

จะเป็นเครื่องมือ(Tools) ที่ใช้อธิบายรายละเอียดการทำงานตามขั้นตอนการทำงาน (Algorithm) ของระบบงานใดๆ โดยใช้ สัญลักษณ์ (Symbol) แทนคำสั่ง, ใช้ข้อความ (Statement) ในสัญลักษณ์แทนตัวแปร และตัวดำเนินการทางการคำนวณและการเปรียบเทียบ มีการกำหนดไว้เป็นมาตรฐานโดยสถาบันมาตรฐานแห่งชาติอเมริกัน (The American National Institute, ANSI) และ องค์การระหว่างประเทศว่าด้วยการมาตรฐาน (International Standard Organization, ISO) (Chapin, 1997) ผังงาน

โครงสร้างแบ่งออกเป็น 2 ประเภท คือ ผังงานระบบ (System Flowchart) และผังงานโปรแกรม (Programming Flowchart)

### 1. ผังงานระบบ (System Flowchart)

เป็นผังงานระดับกว้าง แสดงขั้นตอนการทำงานทั้งหมดของระบบ (ส่วนต่างๆ ที่เกี่ยวข้องกับงานทั้งหมด) ทั้งวัสดุอุปกรณ์ เครื่องมือ เครื่องคอมพิวเตอร์ บุคลากร เป็นต้น จะแสดงขั้นตอนตั้งแต่เริ่มต้นว่าเริ่มจากส่วนใดของระบบงาน ผ่านไปยังหน่วยงานใด มีกิจกรรมอะไรบ้าง ซึ่งบางส่วนของงานเกี่ยวข้องกับคน และในบางส่วนเป็นวิธีการ ที่ต้องใช้คอมพิวเตอร์ในการประมวลผล

### 2. ผังงานโปรแกรม (Programming Flowchart)

แสดงขั้นตอนการทำงานอย่างละเอียด เป็นเอกสารกำกับโปรแกรม เพื่อใช้ประโยชน์ในขั้นตอนการแก้ไขปรับปรุงโปรแกรมในภายหลัง

ประโยชน์ของผังงานโครงสร้าง ใช้แสดงความคิดให้เห็นเป็นรูปภาพ ช่วยให้เข้าใจลำดับการทำงาน และความสัมพันธ์ระหว่างขั้นตอน ทำให้การเขียนโปรแกรมทำได้สะดวก เพื่อใช้เป็นสื่อกลางในการติดต่อประสานความคิดระหว่างผู้ที่เกี่ยวข้อง เช่น ระหว่างผู้เขียนโปรแกรมกับผู้ใช้โปรแกรม ช่วยในการทดสอบหรือทบทวนขั้นตอนการทำงานเพื่อหาข้อผิดพลาด ทั้งในขั้นการทดสอบโปรแกรม และขั้นบำรุงรักษาโปรแกรมเมื่อนำไปใช้งาน ให้ตรงกับความต้องการของผู้ใช้ ผังงานโครงสร้างยังแสดงความสัมพันธ์ของการทำงานต่างๆ อย่างเป็นลำดับขั้นตอนโดยสามารถแบ่งลักษณะของความสัมพันธ์เป็นรูปแบบต่างๆ ได้แก่ การทำงานแบบมีลำดับ (Sequence) การทำงานแบบให้เลือกทำ (Selection) และการทำงานแบบทำซ้ำ (Iteration)

### Kriging Approximation Model

จากที่ได้กล่าวไว้แล้วในบทที่ 2 ว่ากระบวนการประมวลผลในการจำลองการทดลองด้วยคอมพิวเตอร์นั้น ในแต่ละวันมักจะใช้เวลายาวนานมากเนื่องจากระบบที่ต้องการศึกษามีความซับซ้อนมาก นอกจากนี้การใช้รหัสไฟไนต์เอลิเมนต์ (Finite Element Code) ป้อนเข้าไปใน ตัวจำลองคอมพิวเตอร์ (Computer Simulation) แต่ครั้งนั้น จะต้องเสียค่าใช้จ่ายสูงมาก ซึ่งส่งผลให้การทำการทดลองด้วยคอมพิวเตอร์ในทางปฏิบัติสามารถทำได้ภายใต้จำนวนวันที่จำกัดเท่านั้น ดังนั้น นักสถิติจึงพยายามพัฒนาตัวแบบการพยากรณ์ทางสถิติขึ้นมาใช้ เพื่อหารูปแบบความสัมพันธ์ของตัวแปรเข้า (Input Variable) และตัวแปรออก (Output Response) ภายใต้จำนวนวันที่จำกัด เพื่อนำรูปแบบที่สร้างขึ้นไปใช้ในการพยากรณ์ต่อไป ตัวแบบการพยากรณ์ที่ถูกพัฒนาขึ้นมาเพื่อใช้งานวิจัยด้านการจำลองการทดลองด้วยคอมพิวเตอร์ มีชื่อเรียกว่า Kriging Model ซึ่งผู้ที่คิดค้นตัวแบบนี้ คือ Sacks, et al. (1989) การสร้าง Kriging Model ใช้หลักแนวคิดคือ ตัวแปรตาม ( $y$ ) สามารถเขียนอยู่ในรูปแบบฟังก์ชันพหุนาม (Polynomial Function) ของชุดตัวแปรเข้า รวมกับความคลาดเคลื่อนที่เกิดจากกระบวนการ फैนสุ่ม (Stochastic Process) ซึ่งแทนด้วย  $Z(x)$  ความคลาดเคลื่อนนี้มีค่าเฉลี่ยเท่ากับ 0 และความแปรปรวนร่วมระหว่าง  $Z(u)$  และ  $Z(v)$  ใดๆ ในกระบวนการ फैนสุ่มมีค่าเท่ากับ  $V(u,v) = \sigma^2 R(u,v)$  เมื่อ  $\sigma^2$  แทนความแปรปรวนของระบบและ  $R(u,v)$  คือฟังก์ชันสหสัมพันธ์ (Correlation Function)



และสามารถแทนความสัมพันธ์ด้วยรูปแบบต่าง ๆ ซึ่งรูปแบบของฟังก์ชันสหสัมพันธ์เหล่านี้สามารถดูได้จาก Koehler and Owen (1996) โดยทั่วไปแล้วตัวแปรออก  $y$  สามารถเขียนในรูปแบบทางคณิตศาสตร์ได้ดังนี้

$$y = \sum_{j=1}^k \beta_j f_j(x) + Z(x) \quad (1)$$

จากสมการที่ (1) เพื่อเป็นการปรับรูปแบบความสัมพันธ์ของตัวแปรให้ง่ายขึ้น และทำให้สามารถประมาณพารามิเตอร์ได้ง่ายขึ้น ในงานประยุกต์ด้านต่าง ๆ จะใช้ ค่าคงที่แทนฟังก์ชันโพลีโนเมียล ในส่วนแรกของสมการที่ (1) (Sacks, et al., 1989, Welch, et al., 1992, Morris and Mitchell, 1995, Bates, et al., 1996) ดังนั้นสมการที่ (1) สามารถเขียนใหม่ได้

$$y = \beta + Z(x) \quad (2)$$

นอกจากนี้นักวิจัยส่วนใหญ่ได้แสดงไว้ว่าการใช้สมการที่ (2) จะไม่มีผลกระทบใด ๆ ในแง่ของความแม่นยำในการพยากรณ์ (Sacks, et al., 1989, Welch, et al., 1992) โดยทั่วไปแล้วรูปแบบฟังก์ชันสหสัมพันธ์ของ  $Z(x)$  ในส่วนที่ 2 ทางขวามือของสมการที่ (1) ถูกพิจารณาให้อยู่ในรูปแบบของฟังก์ชันสหสัมพันธ์เกาส์เซียน (Gaussian Correlation Function) (Sacks, et al., 1989, Welch et al., 1992, Allen, et al., 2003) ซึ่งสามารถเขียนแทนได้ดังนี้

$$R(X_i, X_j) = \prod_{j=1}^d \exp(-\theta_j |X_i - X_j|^{p_j}) \quad (3)$$

เมื่อ  $0 \leq p_j \leq 2$  และ  $\theta_j > 0$

โดยปกติแล้วความแม่นยำในการพยากรณ์ของโมเดล Kriging จะขึ้นอยู่กับค่าพารามิเตอร์ต่าง ๆ ในส่วนแรกของสมการที่ (1) โดยใช้แนวคิดของวิธีการ กำลังสองน้อยที่สุดทั่วไป (Generalized Least Squares) ซึ่งใช้หลักการหาค่าประมาณของพารามิเตอร์ที่ทำให้ความคลาดเคลื่อนจากการประมาณยกกำลังสองให้มีค่าน้อยที่สุด ส่วนค่าประมาณพารามิเตอร์ต่าง ๆ ในฟังก์ชันสหสัมพันธ์ในส่วนที่สองด้านขวามือของสมการที่ (1) จะใช้วิธี Maximum Likelihood Estimation (MLE) (Sacks, et al., 1989, Welch, et al., 1992) ซึ่ง Welch, et al. (1992) ได้เสนอวิธีการหาค่าประมาณของพารามิเตอร์ของฟังก์ชันสหสัมพันธ์ไว้ โดยเริ่มต้นจากการหาค่าล็อกของฟังก์ชัน likelihood ดังนี้

$$l(\beta, \sigma^2, \theta, p) = -\frac{1}{2} [n \ln \sigma^2 + \ln |R| + (y - 1\beta)^T R^{-1} (y - 1\beta) / \sigma^2] \quad (4)$$

ค่าประมาณแบบกำลังสองน้อยที่สุดทั่วไปของพารามิเตอร์  $\beta$  ในสมการที่ (1) คือ

$$\hat{\beta} = (1^T R^{-1} 1)^{-1} 1^T R^{-1} y \quad (5)$$

และค่าประมาณ MLE ของ  $\sigma^2$  คือ

$$\hat{\sigma}^2 = 1/n (y - 1\hat{\beta})^T R^{-1} (y - 1\hat{\beta}) \quad (6)$$

เมื่อแทนค่า  $\hat{\beta}$  และ  $\hat{\sigma}^2$  เข้าไปในฟังก์ชัน Likelihood ในสมการที่ (4) เพื่อหาค่าประมาณของ  $\theta$  และ  $p$  จะมีความสัมพันธ์กับการหาค่าสูงสุดเชิงตัวเลขของสมการต่อไปนี้

$$-\frac{1}{2}(n \ln \hat{\sigma}^2 + \ln |R|) \quad (7)$$

จากสมการที่ (7) จะเห็นได้ว่าการหาค่าประมาณของพารามิเตอร์  $\theta$  และ  $p$  จะขึ้นอยู่กับข้อมูลที่ได้จากการทดลองในขั้นตอนการรวบรวมข้อมูลนั่นเอง หลังจากที่ได้ค่าประมาณของพารามิเตอร์ทั้งหมดแล้ว ขั้นตอนต่อไปคือ สร้างตัวพยากรณ์  $\hat{y}(x)$  ของ  $y(x)$  เพื่อเป็นตัวแทนรูปแบบความสัมพันธ์ของตัวแปรเข้าและตัวแปรออก ซึ่ง Sacks, et al. (1989) และ Welch, et al. (1992) ได้เสนอไว้ว่าตัวประมาณเชิงเส้นไม่เอนเอียงที่ดีที่สุด (Best Linear Unbiased Predictor: BLUP) เพื่อพยากรณ์ค่า  $y$  ณ ค่าของตัวแปรเข้าใด ๆ ที่ไม่ได้ทำการทดลอง (Untried Input) คือ

$$\hat{y}(x) = \hat{\beta} + r^T(x)R^{-1}(y - 1\hat{\beta}) \quad (8)$$

ซึ่ง  $r(x)$  คือเวกเตอร์ของฟังก์ชันสหสัมพันธ์ ระหว่างค่าความคลาดเคลื่อน ( $Z(x)$ ) ที่ทำการทดลองทั้งหมด  $n$  ครั้งและตัวแปรเข้าที่ไม่เคยทดลอง  $x$  ใด ๆ นั่นเอง

ตัวแบบพยากรณ์แบบ Kriging ได้รับความสนใจและถูกนำมาใช้อย่างกว้างขวางในงานด้านการจำลองการทดลองด้วยคอมพิวเตอร์ (Sacks, et al., 1989, Welch, et al., 1992, Morris and Mitchell, 1995, Ye, Li and Sudjianto, 2000, Simpson, et al., 2001) ทั้งนี้ Simpson, et al. (2001) ได้รายงานไว้ว่าตัวแบบ Kriging มีความยืดหยุ่นมากในแง่ของการเลือกใช้รูปแบบฟังก์ชันสหสัมพันธ์ที่หลากหลาย ซึ่งข้อดีและข้อเสียของการใช้ฟังก์ชันที่แตกต่างกัน ได้ถูกนำเสนอไว้ในงานวิจัยของ Koehler and Owen (1996)

ในงานวิจัยนี้ผู้วิจัยได้นำตัวแบบ Kriging เข้ามาใช้เพื่อศึกษาความสามารถในการพยากรณ์เมื่อใช้แผนการทดลองที่เหมาะสมที่สุด ที่ได้จากอัลกอริทึมการสืบค้นเข้ามาทำการจำลองการทดลองด้วยคอมพิวเตอร์ ซึ่งข้อมูลที่ใช้ในการวิเคราะห์ในงานวิจัยนี้ได้จากการจำลอง (Simulation) ภายใต้มีติปัญหาที่แตกต่างกันออกไป ปัญหาการทดสอบซึ่งเป็นที่นิยมใช้สำหรับประเมินความสามารถในการพยากรณ์ของตัวแบบประเภทต่าง ๆ ซึ่งปัญหาเหล่านี้ประกอบไปด้วย จำนวนตัวแปรตั้งแต่ 2 ถึง 8 ตัว ดังแสดงในตาราง

ตารางแสดงปัญหาทดสอบ



สำนักหอสมุด

5 JUL 2011

Problem	Function
RM2	$f(x_1, x_2) = 0.5(x_1 - x_2^{-1}) - x_1 - 5x_2^{-1}$ $1 \leq x_1, x_2 \leq 100$
Problem 5	$f(x_1, x_2) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1$ $-1.5 \leq x_1 \leq 4, -3 \leq x_2 \leq 3$
Problem 15	$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ $-1 \leq x_1, x_2 \leq 1$
Branin	$f(x_1, x_2) = \left[ x_2 - 5.1 \left( \frac{x_1}{2\pi} \right)^2 + \frac{5x_1}{\pi} - 6 \right]^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos(x_1) + 10$ $-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15$
Mystery	$f(x_1, x_2) = 2 + 0.01(x_2 - x_1^2)^2 + (1 - x_1)^2 + 2(2 - x_2)^2 + 7 \sin(0.5x_1) \sin(0.7x_1x_2)$ $0 \leq x_1, x_2 \leq 1$
RM3	$f(x_1, x_2, x_3) = 0.01x_2x_3^{-1} + 0.01x_1 + 0.005x_1x_3$ $1 \leq x_1, x_2, x_3 \leq 100$
Problem 26	$f(x_1, x_2, x_3) = (x_1 - x_2)^2 + (x_2 - x_3)^4$ $-10 \leq x_1, x_2, x_3 \leq 10$
MS5	$f(x_1, x_2, x_3, x_4, x_5) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^3 + (x_3 - x_4)^4 + (x_4 - x_5)^4$ $-5 \leq x_i \leq 5, i = 1, 2, \dots, 5$
Cyclone*	$y = 174.42 \left( \frac{x_1}{x_2} \right) \left( \frac{x_3}{x_2 - x_1} \right)^{0.85} \times \sqrt{\frac{1 - 2.62 \left\{ 1 - 0.36 \left( x_4/x_2 \right)^{-0.56} \right\}^{3/2} \left( x_4/x_2 \right)^{1.16}}{x_6 x_7}}$
Borehole**	$y_0 = \left\{ 2\pi T_v (H_v - H_i) \right\} / \left\{ \ln(r/r_w) + \left[ 1 + \frac{2LT_v}{\ln(r/r_w) \phi_w^2 K_w} + \frac{r_w}{r} \right] \right\}$

\* รายละเอียดช่วงค่าตัวแปรของปัญหา Cyclone อธิบายในตาราง

\*\* รายละเอียดช่วงค่าตัวแปรของปัญหา Borehole อธิบายในตาราง

ตารางแสดงรายละเอียดช่วงค่าตัวแปรของปัญหา Cyclone (Palmer and Tsui, 2001)

Input Variable	Variable Description	Lower Limit	Upper Limit
$x_1$	Diameter of the cyclone part 1 (m)	0.09	0.11
$x_2$	Diameter of the cyclone part 2 (m)	0.27	0.33
$x_3$	Diameter of the cyclone part 3 (m)	0.09	0.11
$x_4$	Diameter of the cyclone part 4 (m)	0.09	0.11
$x_5$	Physical coefficient	1.35	1.65
$x_6$	Velocity of the gas (m/sec)	14.1	17.6
$x_7$	Height of the cyclone (m)	0.675	0.825

ตารางแสดงรายละเอียดช่วงค่าตัวแปรของปัญหา Borehole (Ye, et al, 2000)

Input Variable	Variable Description	Lower Limit	Upper Limit
$r_w$	Radius of borehole (m)	0.05	0.15
$r$	Radius of influence	100	50000
$T_u$	Transmissivity of upper aquifer ( $m^2 / yr$ )	63070	115600
$H_u$	Potentiometric head of upper aquifer ( $m^2 / yr$ )	900	1110
$T_l$	Transmissivity of lower aquifer ( $m^2 / yr$ )	63.1	116
$H_l$	Potentiometric head of lower aquifer ( $m^2 / yr$ )	700	820
$L$	Length of borehole (m)	1120	1680
$K_w$	Hydraulic conductivity of borehole (m/yr)	9855	12045

DACE (A MATLAB Kriging Toolbox Version 2.0)

จากที่กล่าวไว้ในข้างต้น ว่าผู้วิจัยสนใจศึกษาความสามารถในการพยากรณ์ของตัวแบบ Kriging ซึ่งการประเมินตัวแบบ Kriging ผู้วิจัยใช้ชุดของโปรแกรมคอมพิวเตอร์ DACE (Design and Analysis of Computer Experiments) ที่พัฒนาโดย (Lophaven, Nielsen, Søndergaard, 2002) ซึ่งเป็นเครื่องมือในโปรแกรม MATLAB เพื่อสร้างตัวแบบสำหรับการประมาณ (Approximation Model) ในหัวข้อนี้จะกล่าวถึงรายละเอียดและการใช้ชุดโปรแกรมคอมพิวเตอร์ DACE ซึ่งประกอบด้วยขั้นตอนหลัก ๆ 2

ขั้นตอนคือ การสร้างตัวแบบโดยการประมาณค่าพารามิเตอร์ต่าง ๆ ที่เกี่ยวข้องซึ่งมีชื่อเรียกว่า ฟังก์ชัน Dace ฟิต (Dacefit Funtion) และส่วนที่สองคือการนำตัวแบบที่สร้างไปใช้ในการพยากรณ์รวมถึงการคำนวณค่าความแม่นยำของแผนการทดลองคือฟังก์ชันพยากรณ์ (Predictor Function)

### Dacefit Function

เป็นฟังก์ชันสร้างตัวแบบสำหรับการประมาณ โดยจะต้องป้อนค่าต่าง ๆ เข้าไปเพื่อให้โปรแกรมทำงาน ซึ่งประกอบไปด้วยแผนการทดลองที่เหมาะสมที่สุด ค่าของตัวแปรออก และรูปแบบของสมการโพลีโนเมียล จากสมการ 1 และการกำหนดรูปแบบฟังก์ชันสหสัมพันธ์ (Correlation Model) การเรียกใช้ฟังก์ชัน สามารถทำได้ดังนี้

```
[dmodel, perf] = dacefit(S, Y, regr, corr, theta0, [lob, upb])
```

ค่าต่าง ๆ ที่กำหนดเป็นพารามิเตอร์ ในคำสั่งข้างต้น มีความหมายดังต่อไปนี้

S	คือ ชุดตัวแปรอิสระหรือแผนการทดลองที่เหมาะสมที่สุด
Y	คือ ค่าตอบที่ได้จากกล่องค่าของแผนการทดลอง S หรือตัวแปรออก
regr	คือ รูปแบบของสมการโพลีโนเมียล
corr	คือ รูปแบบฟังก์ชันสหสัมพันธ์
theta0	คือ ค่าเบื้องต้นของ $\theta$ ที่กำหนดขึ้นเพื่อให้โปรแกรมเริ่มต้นค้นหาค่าที่เหมาะสมที่สุดที่ทำให้สมการ (7) มีค่าสูงสุด
lob, upb	คือ ช่วงของค่า $\theta$ ที่เป็นไปได้ ซึ่งกำหนดในขอบเขต ค่าต่ำสุด – ค่าสูงสุด

เมื่อโปรแกรมทำการประมวลผลเสร็จสิ้นแล้วก็จะปรากฏผลลัพธ์ค่าต่าง ๆ ซึ่งผลลัพธ์เหล่านี้มีความหมายดังต่อไปนี้

dmodel คือ รูปแบบที่ดีที่สุดของ Kriging Model ซึ่งประกอบด้วยค่าต่าง ๆ นี้

regr	รูปแบบของสมการโพลีโนเมียล
cor	รูปแบบฟังก์ชันสหสัมพันธ์
theta	ค่าประมาณของพารามิเตอร์ในฟังก์ชันสหสัมพันธ์
beta	ค่าประมาณแบบกำลังสองน้อยที่สุดทั่วไปของพารามิเตอร์ $\beta$ ในสมการ (5)

sigma2 ค่าประมาณการแปรปรวนของการประมวลผล  $\sigma^2$  ในสมการที่ (6)

perf คือ ค่าข้อมูลเกี่ยวกับกระบวนการประมาณค่าพารามิเตอร์ที่เหมาะสมที่สุด ซึ่งมี

nv จำนวนครั้งในการประมวลผลเพื่อหาค่าพารามิเตอร์ที่ดีที่สุด ของ  $\hat{\theta}$

## Predictor Function

เป็นฟังก์ชันเพื่อนำตัวแบบ Kriging ไปใช้ในการพยากรณ์ ณ จุดใด ๆ ในบริเวณที่สนใจศึกษา การเรียกใช้ฟังก์ชัน ประกอบไปด้วยค่าต่าง ๆ ดังนี้

`[y, mse] = predictor(X, dmodel)`

ค่าต่าง ๆ ที่กำหนดเป็นพารามิเตอร์ ในคำสั่งข้างต้น มีความหมายดังต่อไปนี้

X จุดตาข่าย (Grid Point) ในขอบเขตของตัวแปรเข้าที่สนใจจะพยากรณ์ค่าของตัวแปรออก จุดตาข่ายนี้สามารถสร้างได้จากฟังก์ชัน `gridsamp` โดยใช้หลักการสร้างจุดตาข่าย (Grid Point) การเรียกใช้ฟังก์ชัน `gridsamp` ดังนี้

`x = gridsamp(range, q)`

ค่าต่าง ๆ ที่กำหนดในคำสั่งข้างต้น มีความหมายดังต่อไปนี้

range ขอบเขตที่เป็นไปได้ของตัวแปรเข้าแต่ละตัว (ค่าต่ำสุด-สูงสุด)

q จำนวนจุดตาข่ายทั้งหมดที่ต้องการสร้างจากขอบเขตทั้งหมดของแต่ละตัวแปร

ผลลัพธ์ที่ได้จะประกอบด้วย

X จุดตาข่าย ขนาดอาเรย์  $n \times d$

dmodel รูปแบบที่ดีที่สุดของ Kriging Model

ผลลัพธ์ที่ได้ ดังนี้

y ผลลัพธ์จากการพยากรณ์ ซึ่ง  $y(i) = \hat{y}(x(i,:))$

และการวัดความแม่นยำของการพยากรณ์โดยใช้ ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อน (Root Mean Square Error: RMSE) ซึ่งสามารถคำนวณได้จากสูตรต่อไปนี้

$$RMSE = \sqrt{\frac{\sum_{i=1}^k (y_i' - \hat{y}_i')^2}{k}} \quad (9)$$

เมื่อ  $k$  แทนจำนวนจุดทดสอบ (Test Points) ที่สร้างจากจุดตาข่ายทั้งหมด  $y_i'$  แทนค่าตัวแปรออกที่แท้จริงที่ได้จากการแทนค่าตัวแปรเข้า ณ จุดตาข่ายต่าง ๆ และ  $\hat{y}_i'$  แทนค่าพยากรณ์ที่ได้ตัวแบบ Kriging ณ จุดตาข่ายต่าง ๆ

ส่วนฟังก์ชันอื่นๆ ที่เกี่ยวข้องกับการสร้างตัวแบบ Kriging นั้น ชุดโปรแกรม DACE ได้มีการเตรียมไว้หลายทางเลือกดังต่อไปนี้

รูปแบบสมการโพลิโนเมียลแบบต่าง ๆ (จากสมการที่ 1) ประกอบด้วย

regpoly0 สมการโพลิโนเมียลกำลังศูนย์ (Zero Order Polynomial)

regpoly1 สมการโพลิโนเมียลกำลังหนึ่ง (First Order Polynomial)

regpoly2 สมการโพลิโนเมียลกำลังสอง (Second Order Polynomial)

รูปแบบฟังก์ชันสหสัมพันธ์แบบต่าง ๆ ประกอบด้วย

correxp	ฟังก์ชันสหสัมพันธ์แบบเอ็กซ์โปเนนเชียล (Exponential)
correxp	ฟังก์ชันสหสัมพันธ์แบบเอ็กซ์โปเนนเชียลทั่วไป (Generalized exponential)
corrgauss	ฟังก์ชันสหสัมพันธ์แบบเกาส์เซียน (Gaussian)
corrlin	ฟังก์ชันสหสัมพันธ์แบบเชิงเส้น (Linear)
corr spherical	ฟังก์ชันสหสัมพันธ์แบบ Spherical
corr spline	ฟังก์ชันสหสัมพันธ์แบบ Cubic Spline

จากอุปกรณ์และเครื่องมือในการดำเนินการวิจัยที่กล่าวมาแล้วทั้งหมด ผู้วิจัยดำเนินการศึกษาวิจัย และทำการวิเคราะห์ข้อมูลจนครบปัญหาทดสอบตามตารางโดยใช้โปรแกรม MATLAB Version 7.4.0.287 ผลที่ได้จากการวิจัยนี้จะได้นำเสนอในบทถัดไป







## บทที่ 4

### ผลการวิจัย

การวางแผนการทดลอง (Experimental Design) มีความสำคัญอย่างยิ่งต่อการสร้างโมเดลที่มีประสิทธิภาพ เพื่อใช้เป็นตัวต้นแบบหรือเป็นตัวพยากรณ์ (Predictor) ของระบบที่ทำการศึกษาได้อย่างถูกต้องแม่นยำ (Montgomery, 1992) ทำให้ประหยัดงบประมาณและเวลาที่ใช้ในการทดลองกับระบบจริง การเลือกชุดตัวแปรอิสระหรือตัวแปรเข้า (Input Variable) สำหรับการจำลองการทดลองด้วยคอมพิวเตอร์เป็นวิธีการวางแผนการทดลองแบบหนึ่ง ซึ่งการเลือกตัวแปรอิสระบนขอบเขตของค่าระดับต่างๆ ของปัจจัยที่เกี่ยวข้อง (Design Space) มีค่าที่เป็นไปได้มากมายมหาศาลขึ้นกับขนาดของมิติปัญหาที่เราสนใจศึกษา ทำให้การค้นหาแผนการทดลองที่เหมาะสมที่สุดสำหรับมิติปัญหานั้น ๆ เป็นไปได้ยาก ดังนั้น จึงต้องมีการใช้อัลกอริทึมการค้นหา (Search Algorithm) ควบคู่กับเกณฑ์ที่เหมาะสมที่สุดดำเนินการค้นหาแผนการทดลองที่เหมาะสมที่สุด จากนั้นนำแผนการทดลองที่ได้นี้ป้อนค่าเข้าไปในกล่องดำ ซึ่งจะให้ได้ค่าผลลัพธ์ของตัวแปรตาม (Output Response) จากนั้นนำแผนการทดลองที่ใช้กับผลลัพธ์ที่ได้มาหารูปแบบความสัมพันธ์ โดยการสร้างโมเดลประมาณค่าขึ้นมา ถ้าแผนการทดลองที่ใช้ในการทดลองนั้นเป็นแผนการทดลองที่เหมาะสมที่สุด ก็จะส่งผลให้โมเดลที่สร้างขึ้นมานั้นเป็นตัวแทนที่ดีของระบบและสามารถนำไปพยากรณ์ได้อย่างถูกต้องแม่นยำ ในบทนี้จะกล่าวถึงอัลกอริทึมการค้นหาที่ถูกลำเอียงมาใช้อย่างแพร่หลายสำหรับการออกแบบการทดลองด้วยคอมพิวเตอร์ ซึ่งอัลกอริทึมการค้นหาที่นิยมใช้กันคือ SA (Simulated Annealing) และเนื่องจากการศึกษาวิจัยก่อนหน้านี้พบว่า SA มีประสิทธิภาพดีกว่า GA ทั้งในด้านความเรียบง่ายในการกำหนดพารามิเตอร์ตั้งต้น (Simplicity) รวมไปถึงความสามารถในการลู่เข้าสู่เกณฑ์การเลือกค่าที่เหมาะสมที่รวดเร็วกว่า (Rungrattanaubol and Naudom, 2007) ทั้งนี้ในงานวิจัยดังกล่าวยังพบ SA ใช้เวลาในการสืบค้นนานมากขึ้นเมื่อมิติปัญหาที่มีขนาดเพิ่มขึ้น ดังนั้นผู้วิจัยจึงมีความสนใจที่จะพัฒนาปรับปรุงอัลกอริทึมการค้นหา SA ให้มีประสิทธิภาพมากยิ่งขึ้น ซึ่งเนื้อหาทั้งหมดที่จะกล่าวถึงในบทนี้ประกอบไปด้วยหัวข้อดังต่อไปนี้

การทำงานของ SA และการกำหนดค่าพารามิเตอร์ตั้งต้นต่างๆ

การคำนวณหาค่าเกณฑ์เหมาะสมที่สุด (Optimality Criteria)

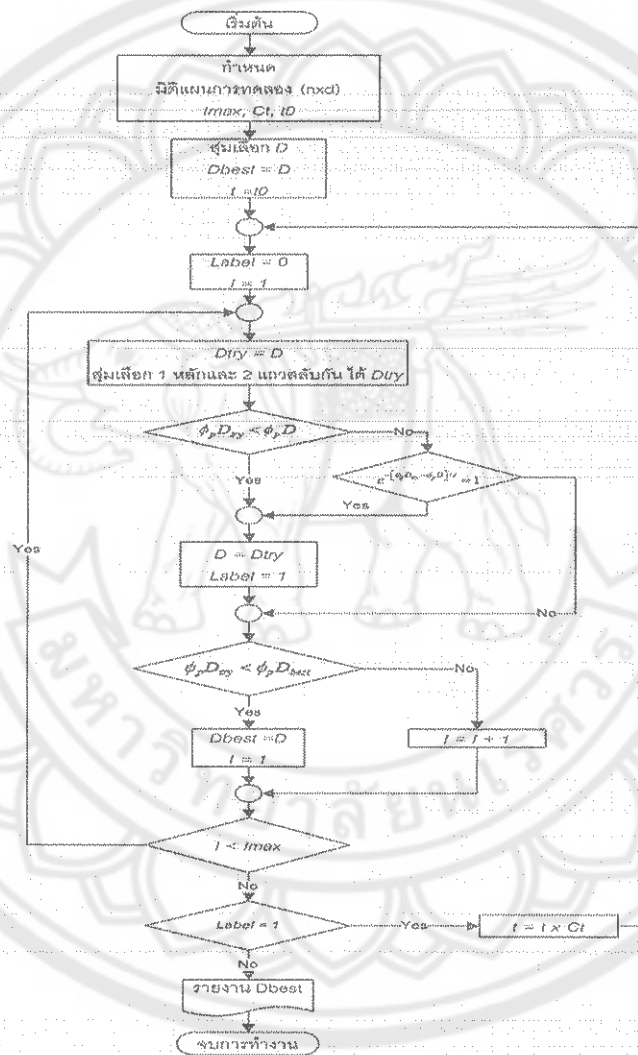
ทดสอบประสิทธิภาพของแผนการทดลองที่เหมาะสมที่สุดโดยใช้ DACE (A MATLAB Kriging Toolbox)

รายละเอียดการปรับปรุงอัลกอริทึม SA และการคำนวณหาค่า Optimality Criteria (Modified Simulated Annealing: MSA)

เปรียบเทียบผลลัพธ์ที่ได้จาก MSA1 และ MSA2

การทำงานของ SA และการกำหนดค่าพารามิเตอร์ตั้งต้นต่าง ๆ

จากบทที่ 2 ได้กล่าวถึงหลักการทำงานของ SA มาแล้ว แสดงได้ดังภาพ



ภาพแสดงผังงานการทำงานของ SA แบบพื้นฐาน

จากการทบทวนวรรณกรรมของงานวิจัยก่อนหน้า นำไปสู่การกำหนดพารามิเตอร์ตั้งต้นสำหรับอัลกอริทึมการสืบค้น SA ดังต่อไปนี้

$l_{max}$  คือ จำนวนการสับเปลี่ยนที่มากที่สุดเพื่อค้นหาแผนการทดลองที่ดีขึ้นก่อนที่จะมีการลดอุณหภูมิ โดยค่านี้จะต้องมากพอที่จะทำให้ไกลของ SA หลุดออกจากค่าที่ดีที่สุดในระดับกลุ่ม (local minimum) จากงานวิจัยของ (Morris and Mitchell, 1995) และ (Na-udom, 2007) ซึ่งได้รายงานการ

กำหนดค่าของ  $I_{\max}$  ไว้ ดังนั้นในงานวิจัยนี้จึงกำหนดให้  $I_{\max} = 1000$  สำหรับ  $d \leq 6$  และ  $I_{\max} = 500$  สำหรับ  $d \geq 7$

$t_0$  คือ อุณหภูมิความเย็นเริ่มต้น

$C_i$  คือ อัตราในการลดอุณหภูมิ โดยกำหนดให้เท่ากับ 0.95 เพื่อทำการลดอุณหภูมิอย่างช้า ๆ เมื่อกระบวนการสืบค้นไม่สามารถค้นหาแผนการทดลองที่ดีขึ้น

จะเห็นได้ว่า ค่า  $t_0$  นั้นจะเป็นตัวควบคุมความน่าจะเป็นที่แผนการทดลองที่ดีกว่า จะถูกยอมรับและเกิดการแทนที่แผนการทดลองที่ได้ในขั้นตอนก่อนหน้า โดยปกติแล้วการกำหนดค่า  $t_0$  ที่มีค่ามาก ๆ จะส่งผลให้อุณหภูมิของแผนการทดลองที่ไม่มีประสิทธิภาพมากขึ้น จะถูกยอมรับและเกิดการแทนที่สูงขึ้น ดังนั้นเมื่อกระบวนการสืบค้นดำเนินการไปเรื่อย ๆ ถ้าไม่พบแผนการทดลองที่ดีขึ้น SA ก็ จะทำการลดอุณหภูมิ โดยทำการคูณ  $t_0$  ด้วยค่า  $C_i$  ซึ่งจะทำให้ค่า  $t_0$  มีค่าลดลง และจะส่งผลโอกาสที่จะยอมรับแผนการทดลองที่ไม่มีการเปลี่ยนแปลงในทางที่ดีขึ้น มีค่าลดลงตามลำดับ

### การคำนวณหาค่าเกณฑ์เหมาะสมที่สุด (Optimality Criteria)

ในการออกแบบการทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์นั้น จะต้องมีการพิจารณาเกณฑ์ของค่าที่เหมาะสมที่สุด เพื่อเปรียบเทียบว่าแผนการทดลองใดดีกว่ากัน ซึ่งแผนการทดลองที่นิยมใช้ในการจำลองการทดลองด้วยคอมพิวเตอร์ จะต้องมีการกระจายจุดทดลองให้ครอบคลุมขอบเขตที่เป็นไปได้ทั้งหมดของการทดลอง (Design Space) ให้มากที่สุด ซึ่งแผนการทดลองลักษณะนี้มีชื่อเรียกว่า แผนการทดลองแบบเติมเต็มขอบเขต (Space Filling Design) แผนการทดลองประเภทนี้ถูกนำมาใช้อย่างแพร่หลาย เนื่องจากมีความเหมาะสมกับรูปแบบที่คงตัว (Deterministic) ของตัวแปรตามที่ได้จากจำลองการทดลองด้วยคอมพิวเตอร์ ซึ่งเกณฑ์ที่ใช้เลือกแผนการทดลองที่เหมาะสมนั้น ถูกนำเสนอไว้หลายวิธี ดังที่กล่าวไว้แล้วในบทที่ 2 ในงานวิจัยนี้ผู้วิจัยเลือกใช้ เกณฑ์  $\phi_p$  (Phi-p) ซึ่งคิดค้นโดย Morris และ Mitchell (Morris and Mitchell, 1995) เพื่อใช้ในการเปรียบเทียบแผนการทดลองว่าแผนการทดลองใดเป็นแผนการทดลองที่เหมาะสมที่สุดสำหรับแต่ละมิติปัญหา โดยการพิจารณาว่าแผนการทดลองที่ดีที่สุดจะต้องให้ค่า  $\phi_p$  ที่ต่ำที่สุด (Minimization of  $\phi_p$ ) เท่าที่จะเป็นไปได้ หลักการคำนวณหาค่า  $\phi_p$  นั้น มีแนวคิดดังนี้

กำหนดให้  $X$  เป็นเมทริกซ์แผนการทดลองใด ๆ มีขนาดเป็น  $n \times d$  โดย  $n$  คือ จำนวนรันในแผนการทดลอง และ  $d$  คือ จำนวนตัวแปรเข้าของแผนการทดลอง ดังนั้น  $X$  สามารถเขียนแทนด้วย

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}_{n \times d}$$

จากเมทริกซ์  $X_{(n \times d)}$  ที่กำหนดข้างต้น ทำการคำนวณหาระยะห่างระหว่างจุด  $x_i$  และ  $x_j$  ของเมทริกซ์การทดลองโดยใช้สูตรการคำนวณระยะห่างแบบ Euclidean Distance จากสูตร

$$d(x_i, x_j) = \left[ \sum_{k=1}^d (x_{ik} - x_{jk})^2 \right]^{1/2} \quad (10)$$

หลังจากคำนวณระยะห่างครบทุกจุดทดลองแล้ว จะได้เมทริกซ์ระยะห่างซึ่งเป็นเมทริกซ์จัตุรัสขนาด  $n \times n$  ดังนี้

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

โดย  $d_{ij}$  เป็นค่าระยะห่างระหว่างจุด  $i$  กับ  $j$  จากสูตรการหาระยะห่างระหว่างจุดแบบ Euclidean Distance จะทำให้ได้ เมทริกซ์แบบสมมาตร (Symmetric Matrix) โดยสมาชิกในแนวเส้นทแยงมุม (Diagonal) มีค่าเป็นศูนย์ทั้งหมด และสมาชิกที่อยู่ตำแหน่งตรงกันที่อยู่ด้านบนและด้านล่างของเส้นทแยงมุมจะมีค่าสมมาตรกัน ในงานวิจัยนี้ผู้วิจัยเลือกใช้สมาชิกเฉพาะส่วนด้านบนของเส้นทแยงมุมนำมาทำการคำนวณหาค่า  $\phi_p$  เนื่องจากทำให้การการคำนวณค่า  $\phi_p$  สามารถทำได้ง่ายขึ้น และส่งผลให้ได้ผลลัพธ์ที่ยังคงคุณสมบัติเทียบเท่ากับการใช้สมาชิกทุกค่าของเมทริกซ์ระยะห่างในการคำนวณ  $\phi_p$  (Jin, Chen and Sudjianto, 2005)

จากข้อมูลค่าระยะห่างทั้งหมดที่อยู่ส่วนบนของแนวทแยงมุม นำมาจัดลำดับ จากน้อยไปมากจะได้ลิสต์ของระยะห่าง  $(d_1, d_2, \dots, d_m)$  และได้ลิสต์ของดัชนี  $(J_1, J_2, \dots, J_m)$  โดย  $J_j$  เป็นจำนวนคู่ของจุดทดลองที่ถูกแยกด้วยระยะ  $d_j$  จากนั้นนำค่าลิสต์ของระยะห่างและลิสต์ของดัชนีมาคำนวณหาค่า  $\phi_p$  จากสูตร

$$\phi_p = \left[ \sum_{j=1}^m J_j d_j^{-p} \right]^{1/p} \quad (11)$$

จากสูตรดั้งเดิมของการคำนวณ  $\phi_p$  ที่จะต้องใช้สมาชิกทั้งจากค่าลิสต์ของระยะห่างและลิสต์ของดัชนี จะเห็นได้ว่าค่อนข้างจะยุ่งยาก และซับซ้อน และจากการวิจัยของ (Jin, Chen and Sudjianto, 2003) และ (Na-udom, 2007) แสดงให้เห็นว่าเราสามารถคำนวณ  $\phi_p$  ได้ง่ายขึ้นแต่ยังคงคุณสมบัติของเกณฑ์ค่าที่เหมาะสมที่เท่าเดิม โดยใช้สมาชิกในลิสต์ของระยะห่าง  $(d_1, d_2, \dots, d_m)$  เท่านั้นในการคำนวณ ดังนั้นสูตรอย่างง่ายในการคำนวณค่า  $\phi_p$  คือ

$$\phi_p = \left[ \sum_{1 \leq i < j \leq n} (1/d_{ij})^p \right]^{1/p} \quad (12)$$

ซึ่งสูตรข้างบนนี้สัมพันธ์กับสูตรที่มีรูปแบบอย่างง่ายที่มีการนำมาใช้ในงานวิจัยของ Jin, Chen และ Sudjianto (2005) ดังนี้

$$\phi_p = \left[ \sum_{1 \leq i < j \leq n} (d_{ij})^{-p} \right]^{1/p} \quad (13)$$

ดังนั้นในงานวิจัยนี้ ผู้วิจัยเลือกใช้สูตรอย่างง่ายในการคำนวณค่า  $\phi_p$  ในกระบวนการสืบค้นใน SA

## ทดสอบแผนการทดลองที่เหมาะสมที่สุดจาก SA โดยใช้ DACE (A MATLAB Kriging Toolbox)

เพื่อเป็นการทดสอบคุณสมบัติของแผนการทดลองที่เหมาะสมที่สุดที่ได้จากการค้นหาโดยใช้ SA ภายใต้เกณฑ์  $\phi_p$  ที่จะส่งผลให้สามารถสร้างตัวแบบที่มีความแม่นยำในการพยากรณ์สูงขึ้นหรือไม่ ผู้วิจัยจึงนำแผนการทดลองที่เหมาะสมที่สุดที่ได้จากอัลกอริทึมการสืบค้นไปทดสอบกับชุดโปรแกรมที่มีชื่อว่า DACE (Design and Analysis of Computer Experiments) ซึ่งเป็นเครื่องมือที่ถูกเขียนบนโปรแกรมภาษา MATLAB โดย (Lophaven, Nielsen and Søndergaard, 2002) เพื่อสร้างตัวแบบสำหรับการประมาณ (Approximation Model) ซึ่งมีชื่อเรียกว่า Kriging Model ซึ่งมีหลักการทำงานดังต่อไปนี้

**ขั้นตอนที่ 1** กำหนดปัญหาที่ใช้ในการทดสอบ (Test Problem) และสร้างแผนการทดลองให้สอดคล้องกับมิติของปัญหาทดสอบ

**ขั้นตอนที่ 2** สร้างค่าตัวแปรตามจากปัญหาทดสอบที่กำหนดภายใต้ค่าต่าง ๆ ของแผนการทดลองที่สร้างขึ้น

**ขั้นตอนที่ 3** จากแผนการทดลองในขั้นตอนที่ 1 และค่าของตัวแปรตามในขั้นตอนที่ 2 นำมาสร้างตัวแบบของการพยากรณ์โดยใช้ชุดเครื่องมือ DACE

**ขั้นตอนที่ 4** สร้างจุดทดสอบ (Test Point) จากจุดตาข่ายของบริเวณตัวแปรเข้าที่สนใจโดยใช้รูปแบบความสัมพันธ์ที่กำหนดตามปัญหาทดสอบ จะได้ค่าของตัวแปรตาม ณ ระดับต่าง ๆ ของจุดตาข่าย

**ขั้นตอนที่ 5** พยากรณ์ค่าตัวแปรตาม ณ จุดทดสอบต่าง ๆ โดยการใช้ตัวแบบการพยากรณ์จากขั้นตอนที่ 3 และทำจนครบทุกจุดบนตาข่าย

**ขั้นตอนที่ 6** คำนวณค่า RMSE จากค่าตัวแปรตามจากขั้นตอนที่ 4 และค่าพยากรณ์ของตัวแปรตามที่ได้จากขั้นตอนที่ 5

จากขั้นตอนที่กล่าวมาข้างต้น ผู้วิจัยเลือกแผนการทดลองมา 2 ประเภทคือ แผนการทดลองแบบลาตินไฮเปอร์คิวบ์แบบสุ่ม และแผนการทดลองแบบลาตินไฮเปอร์คิวบ์ที่เหมาะสมที่สุด เพื่อทำการเปรียบเทียบประสิทธิภาพของการพยากรณ์เมื่อใช้แผนการทดลองทั้งสองประเภทนี้ เพื่อให้ง่ายต่อการทำความเข้าใจในขั้นตอนต่าง ๆ ที่กล่าวมาทั้งหมด ผู้วิจัยจะแสดงตัวอย่างในการใช้ชุดเครื่องมือ DACE กับปัญหาทดสอบที่มีชื่อว่า RM2 (จากตาราง) ดังขั้นตอนต่อไปนี้

**ขั้นตอนที่ 1** จากปัญหาทดสอบ RM2 ซึ่งมีสมการเป็น

$$f(x_1, x_2) = 0.5(x_1 - x_2^{-1}) - x_1 - 5x_2^{-1}$$

ประกอบไปด้วยจำนวนตัวแปรเข้า 2 ตัว คือ  $x_1$  กับ  $x_2$  ผู้วิจัยเลือกใช้แผนการทดลองที่ประกอบไปด้วย 9 วัน ดังเมตริกซ์ต่อไปนี้

0.375	0.875
0.125	0.375
0.625	0.125
0.250	0
1.000	0.250
0.875	0.625
0.500	0.500
0	0.750
0.750	1.000

(a)

0.625	0.750
0.250	0.125
0.750	0.875
0.875	0.500
0.500	0.625
0	1.000
0.125	0.375
0.375	0
1.000	0.250

(b)

ภาพแสดงแผนการทดลองแบบลาตินไฮเปอร์คิวบ์ (a) แบบสุ่ม และ (b)แบบที่เหมาะสมที่สุด

จะเห็นได้ว่าแผนการทดลองที่เลือกมานั้น มีค่าของตัวแปรอิสระแต่ละตัวอยู่ในช่วงตั้งแต่ 0 ถึง 1 โดยแบ่งเป็นช่วงเท่าๆ กัน เรียกว่า ช่วงที่เป็นยูนิท (Unit Interval) บนคลาสการออกแบบที่เรียกว่า ลาตินไฮเปอร์คิวบ์ (LHD) ซึ่งหลักการในการใช้ช่วงยูนิทนี้เพื่อความสะดวกในการคำนวณและง่ายต่อการแปลงค่าให้เป็นค่าที่แท้จริงของแต่ละปัญหาทดสอบอีกด้วย ซึ่งหลักการสร้างแผนการทดลองในคลาสการออกแบบที่เรียกว่า LHD ได้อธิบายไว้แล้วในบทที่ 2

ขั้นตอนต่อไปเป็นการนำแผนการทดลองมาปรับค่าตัวแปรอิสระแต่ละตัว ให้อยู่ในช่วงค่าจริงที่กำหนดของแต่ละปัญหาทดสอบ ซึ่งปัญหา RM2 มีค่า  $x_1$  กับ  $x_2$  อยู่ในช่วงตั้งแต่ 1 ถึง 100 ( $1 \leq x_1, x_2 \leq 100$ ) ดังนั้นแผนการทดลองแบบ LHD ข้างต้นจะถูกแปลงค่าตัวแปรอิสระแต่ละตัวแปรให้อยู่ในช่วงค่าจริง คือ  $1 \leq x_1, x_2 \leq 100$  โดยคำนวณได้จากสูตร

$$x_{\text{actual}} = \text{lower bound} + (x_{\text{unit}} \times (\text{upper bound} - \text{lower bound}))$$

โดย  $x_{\text{actual}}$  คือ ค่าจริง,  $x_{\text{unit}}$  คือ ค่าจากยูนิท เมตริกซ์, **Lower Bound** คือ ค่าจริงต่ำสุด, **Upper Bound** คือ ค่าจริงสูงสุด จะได้แผนการทดลองในช่วงค่าจริงเป็นดังนี้

38.125	87.625
13.375	38.125
62.875	13.375
25.750	1.000
100.000	25.750
87.625	62.875
50.5000	50.500
1.000	75.250
75.250	100.000

(a)

62.875	75.250
25.750	13.375
75.250	87.625
87.625	50.500
50.500	62.875
1.000	100.000
13.375	38.125
38.125	1.000
100.000	25.750

(b)

ภาพแสดงแผนการทดลองในช่วงค่าจริง (a) แบบสุ่ม และ (b) แบบที่เหมาะสมที่สุด  
 ขั้นตอนที่ 2 นำแผนการทดลองที่ปรับค่าให้อยู่ในช่วงจริงจากขั้นตอนที่ 2 มาคำนวณหาค่าของตัวแปรตามโดยการแทนค่าของตัวแปรเข้าในปัญหาทดสอบ จะได้ค่าตัวแปรตาม (Y) ดังนี้

-19.1253
-6.8318
-31.8487
-18.3750
-50.2136
-43.9000
-25.3589
-0.5731
-37.6800

(a)

-31.5106
-13.2862
-37.6878
-43.9214
-25.3375
-0.5550
-6.8318
-24.5625
-50.2136

(b)

ภาพ แสดงค่าของตัวแปรตามของแผนการทดลอง (a) แบบสุ่ม และ (b) แบบที่เหมาะสมที่สุด

ขั้นตอนที่ 3 สร้างตัวแบบพยากรณ์โดยใช้ชุดเครื่องมือ DACE ดังรายละเอียดที่กล่าวไว้แล้วในบทที่ 3 โดยเริ่มจากการเลือกรูปแบบของฟังก์ชันสหสัมพันธ์ (Correlation Function) เพื่อหาค่าสัมประสิทธิ์สหสัมพันธ์ที่เหมาะสมที่สุดและเลือกรูปแบบของสมการโพลีโนเมียลที่ต้องการ จากนั้นใส่ค่าพารามิเตอร์ต่างๆ ในฟังก์ชัน dacefit ที่ DACE เตรียมไว้ให้ เพื่อสร้างตัวแบบสำหรับการประมาณเมื่อได้ตัวแบบสำหรับการประมาณแล้ว ในที่นี้คือ dmodel ก็จะใช้เป็นตัวแทนของระบบเพื่อใช้ในการพยากรณ์ต่อไป

ขั้นตอนที่ 4 สร้างจุดตาข่ายเป็น 2 มิติเพราะแผนการทดลองมีตัวแปร 2 ตัว โดยต้องกำหนดจำนวนจุดตาข่ายให้มากพอเพื่อศึกษาความสามารถในการพยากรณ์ได้ทั่วถึงบริเวณทั้งหมดของปัญหาทดสอบ ซึ่งการสร้างจุดตาข่ายสามารถทำได้โดยการเรียกใช้ฟังก์ชัน  $X = \text{gridamp}([1 \ 1; 100 \ 100], 10)$

และได้  $X$  เป็นจุดตาข่าย จำนวน  $10 \times 10$  จุด ในช่วง  $[1,100]$  เพื่อใช้ในการทดลองกับ RM2 จากนั้นนำ  $X$  ที่ได้ไปใส่ในปัญหาทดสอบ จะได้  $y_i'$  ซึ่งเป็นค่าของตัวแปรตัวแปรตามที่ได้จากการแทนค่าของตัวแปรเข้าที่ระดับต่าง ๆ ของจุดตาข่ายลงในปัญหาทดสอบนั่นเอง

**ขั้นตอนที่ 5** นำตัวแบบสำหรับการพยากรณ์จากขั้นตอนที่ 4 ในที่นี้คือ dmodel มาพยากรณ์โดยใช้ฟังก์ชัน predictor ซึ่งเรียกใช้ด้วยคำสั่ง predictor( $X$ , dmodel) จะได้  $\hat{y}_i'$  ซึ่งเป็นค่าประมาณของตัวแปรตามที่ได้จาก Kriging model

**ขั้นตอนที่ 6** นำค่า  $y_i'$  และ  $\hat{y}_i'$  มาคำนวณหาความแม่นยำของการพยากรณ์โดยใช้ ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อน (Root Mean Square Error: RMSE) จากสูตร

$$RMSE = \sqrt{\frac{\sum_{i=1}^k (y_i' - \hat{y}_i')^2}{k}}$$

โดยที่  $k$  คือ จำนวนจุดตาข่าย

จากสูตรในการคำนวณจะเห็นได้ว่า ถ้าตัวแบบในการพยากรณ์มีความเหมาะสม จะส่งผลให้ค่า  $y_i'$  และ  $\hat{y}_i'$  มีค่าใกล้เคียงกัน ซึ่งจะทำให้ค่า RMSE ก็จะมีค่าน้อยลง ดังนั้นในการเลือกใช้ตัวแบบของการพยากรณ์ที่ดีที่สุดที่สุดหมายถึงการเลือกตัวแบบที่ให้ค่า RMSE ที่น้อยที่สุดนั่นเอง ในงานวิจัยนี้ ผู้วิจัยเลือกปัญหาการทดสอบซึ่งเป็นที่นิยมใช้สำหรับประเมินความสามารถในการพยากรณ์ของตัวแบบประเภทต่าง ๆ ซึ่งปัญหาเหล่านี้ประกอบไปด้วย จำนวนตัวแปรตั้งแต่ 2 ถึง 8 ตัว ดังแสดงในตาราง

จากปัญหาทดสอบทั้งหมด ดังแสดงในตาราง ผู้วิจัยเลือกแผนการทดลองที่เหมาะสมที่สุดจาก SA (OLHD) และ แผนการทดลองแบบสุ่ม (RLHD) มากับใช้เครื่องมือ DACE ประมวลผลการพยากรณ์และแสดงความแม่นยำของการพยากรณ์โดยใช้ค่า RMSE เป็นเกณฑ์ในการเปรียบเทียบประสิทธิภาพของแผนการทดลองทั้งสองประเภท สำหรับแต่ละปัญหาทดสอบได้เลือกแผนการทดลองทั้งสองประเภทมาอย่างละ 10 แผนการทดลอง เพื่อวัดความแม่นยำในการพยากรณ์โดยเฉลี่ย นอกจากนี้ผู้วิจัยได้ทำการเปรียบเทียบ ณ ระดับค่าของพารามิเตอร์ในฟังก์ชันสหสัมพันธ์ที่แตกต่างกัน 3 ระดับดังนี้  $\theta = 0.01, 0.05$  และ  $0.10$  ดังปรากฏผลตามตาราง



ตารางแสดงค่า RMSE ที่ได้จากปัญหาทดสอบต่าง ๆ

Problem	มิติแผนการทดลอง		ชนิดแผนการทดลอง	theta ( $\theta$ )	RMSE			
	d	n			Mean	Min	Max	SD
RM2	2	9	RLHD	0.01	2.1333	0.9198	4.0080	0.8604
			OLHD	0.01	1.5309	1.4174	2.0122	0.1774
			RLHD	0.05	1.8347	1.0422	2.6563	0.5661
			OLHD	0.05	1.6886	1.3938	2.1373	0.2699
			RLHD	0.10	1.9376	1.1663	3.0210	0.6429
			OLHD	0.10	2.0199	1.6368	2.2752	0.3297
Problem 5	2	9	RLHD	0.01	8.9786	7.9620	10.2268	0.7330
			OLHD	0.01	7.8817	7.8771	7.8847	0.0039
			RLHD	0.05	7.7068	6.4255	9.1753	0.9430
			OLHD	0.05	6.8442	6.8138	6.8645	0.0262
			RLHD	0.10	6.5031	4.9614	7.9625	1.0627
			OLHD	0.10	5.8940	5.8397	5.9302	0.0467
Problem 15	2	9	RLHD	0.01	68.1076	63.0423	73.8229	3.7315
			OLHD	0.01	61.0075	60.9737	61.0300	0.0290
			RLHD	0.05	66.1398	60.7052	74.2666	4.4514
			OLHD	0.05	60.3872	60.3462	60.4146	0.0353
			RLHD	0.10	64.5274	58.9065	74.8535	5.3669
			OLHD	0.10	59.7786	59.7295	59.8113	0.0422
Branin	2	9	RLHD	0.01	5.60E+05	4.10E+05	6.37E+05	6.84E+04
			OLHD	0.01	5.03E+05	5.02E+05	5.04E+05	1.25E+03
			RLHD	0.05	5.40E+05	3.87E+05	6.14E+05	6.99E+04
			OLHD	0.05	4.99E+05	4.97E+05	5.00E+05	1.29E+03
			RLHD	0.10	5.27E+05	3.87E+05	6.18E+05	6.84E+04
			OLHD	0.10	4.94E+05	4.93E+05	4.95E+05	1.32E+03

ตาราง (ต่อ)

Problem	มิติ		ชนิด แผนกา ร	theta ( $\theta$ )	RMSE			
	แผนการ ทดลอง	ทดลอง			Mean	Min	Max	SD
	d	n						
Mystery	2	9	RLHD	0.01	19.8080	5.4960	45.5576	14.4356
			OLHD		5.2202	5.1591	5.2608	0.0525
			RLHD	0.05	15.1565	5.4874	27.7040	8.5213
			OLHD		5.0065	4.9885	5.0185	0.0155
			RLHD	0.10	12.7114	5.5269	23.6692	6.3963
			OLHD		4.8147	4.8008	4.8355	0.0179
RM3	3	19	RLHD	0.01	0.3610	0.2319	0.6604	0.1570
			OLHD		0.2239	0.1860	0.2631	0.0223
			RLHD	0.05	0.9804	0.6032	1.4091	0.2657
			OLHD		0.7899	0.6688	0.9083	0.0714
			RLHD	0.10	1.7149	0.9753	2.4396	0.4205
			OLHD		1.4969	1.2191	1.6827	0.1429
Problem 26	3	19	RLHD	0.01	3.10E+04	1.96E+04	5.47E+04	1.02E+04
			OLHD		1.85E+04	1.72E+04	2.05E+04	1.09E+03
			RLHD	0.05	2.55E+04	1.79E+04	3.09E+04	4.41E+03
			OLHD		1.85E+04	1.74E+04	1.95E+04	781.7257
			RLHD	0.10	2.39E+04	1.88E+04	2.82E+04	3.16E+03
			OLHD		1.94E+04	1.81E+04	2.07E+04	868.8268
MS5	5	51	RLHD	0.01	6.94E+03	4.09E+03	9.28E+03	1.75E+03
			OLHD		5.30E+03	4.37E+03	6.18E+03	567.8364
			RLHD	0.05	6.94E+03	4.09E+03	9.28E+03	1.75E+03
			OLHD		5.30E+03	4.37E+03	6.18E+03	567.8364
			RLHD	0.10	6.94E+03	4.09E+03	9.28E+03	1.75E+03
			OLHD		5.30E+03	4.37E+03	6.18E+03	567.8364
Cyclone	7	99	RLHD	0.01	0.002050	0.001514	0.002979	0.000431
			OLHD		0.001532	0.001106	0.001914	0.000246
			RLHD	0.05	0.004343	0.003121	0.005427	0.000718
			OLHD		0.003326	0.002940	0.003723	0.000237
			RLHD	0.10	0.005786	0.005097	0.006700	0.000418
			OLHD		0.004522	0.003876	0.004980	0.000361

ตาราง (ต่อ)

Problem	มิติ		ชนิด	theta ( $\theta$ )	RMSE			
	แผนการ				Mean	Min	Max	SD
	ทดลอง	ทดลอง	d	n				
Borehole	8	129	RLHD	0.01	62606.7605	36730.694	105539.904	18667.933
			OLHD		49991.0112	26826.864	62538.9317	11728.929
			RLHD	0.05	2773.0409	1994.9508	4607.2016	736.407
			OLHD		2245.04815	1261.3561	2999.4729	521.096
			RLHD	0.10	894.811527	675.57283	1338.0858	183.195
			OLHD		695.206848	425.39648	1013.6736	165.410

จากตาราง เมื่อใช้ปัญหาทดสอบ RM2 จะเห็นได้ว่า ค่าเฉลี่ยของ RMSE ของตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ OLHD มีค่าต่ำกว่า ตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ RLHD เมื่อ  $\theta$  มีค่าเท่ากับ 0.01 และ 0.05 อย่างไรก็ตามเมื่อ  $\theta$  มีค่าเพิ่มขึ้นเป็น 0.10 ค่าเฉลี่ยของ RMSE ตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ OLHD มีค่าสูงกว่าค่าเฉลี่ยของ RMSE ตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ RLHD ซึ่งสามารถอธิบายได้ว่า ค่า  $\theta$  ที่สูงขึ้นอาจไม่เหมาะสมกับตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ OLHD จากการพิจารณาการเปลี่ยนแปลงของค่าเฉลี่ย RMSE เมื่อมีการเปลี่ยนแปลงระดับของ  $\theta$  จะเห็นได้ว่าค่าเฉลี่ย RMSE มีการเปลี่ยนแปลงขึ้นลงตลอดเวลา ซึ่งแสดงให้เห็นว่าความสามารถในการพยากรณ์ของตัวแบบ Kriging มีความไว (Sensitivity) ต่อค่าพารามิเตอร์ในฟังก์ชันสหสัมพันธ์ เมื่อพิจารณาค่าวัดการกระจายโดยใช้ค่าส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation: SD) ของค่า RMSE เมื่อใช้แผนการทดลองทั้งหมด 10 แผนการทดลอง จะเห็นได้ว่า ตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ OLHD จะให้ค่าการกระจายของ RMSE ที่น้อยกว่าตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ RLHD ซึ่งแสดงให้เห็นว่าการใช้แผนการทดลองแบบ OLHD จะก่อให้เกิดความสามารถในการพยากรณ์ที่ค่อนข้างคงที่เนื่องจาก OLHD สามารถค้นหาแผนการทดลองที่เหมาะสมเท่า ๆ กันนั่นเอง

เมื่อพิจารณาผลที่ได้จากปัญหาทดสอบ Problem5, Problem15, Branin, Mystery และ Borehole จะเห็นได้ว่า ค่าเฉลี่ยของ RMSE ของตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ OLHD มีค่าต่ำกว่าตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ RLHD สำหรับทุกระดับของค่า  $\theta$  จากการพิจารณาการเปลี่ยนแปลงของค่าเฉลี่ย RMSE เมื่อ  $\theta$  มีค่าเพิ่มขึ้น จะเห็นได้ว่าค่าเฉลี่ย RMSE

มีการเปลี่ยนแปลงที่ลดลง ซึ่งแสดงให้เห็นว่าความสามารถในการพยากรณ์ของตัวแบบ Kriging สำหรับปัญหาเหล่านี้ ณ ระดับค่า  $\theta$  ที่มีค่าสูง จะทำให้ได้ตัวแบบที่มีความถูกต้อง แม่นยำมากกว่าค่า  $\theta$  ที่ระดับต่ำกว่า เมื่อพิจารณาค่าวัดการกระจายโดยใช้ค่าส่วนเบี่ยงเบนมาตรฐาน (SD) ของค่า RMSE ที่ได้จากแผนการทดลองทั้งหมด 10 แผนการทดลอง จะเห็นได้ว่า ตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ OLHD จะให้ค่าการกระจายของ RMSE ที่น้อยกว่าตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ RLHD ซึ่งแสดงให้เห็นว่าการใช้ตัวแบบพยากรณ์ที่สร้างจาก OLHD จะก่อให้เกิดความสามารถในการพยากรณ์ที่ค่อนข้างคงที่ ดังนั้นในการเลือกใช้แผนการทดลองที่เหมาะสมใด ๆ จาก OLHD จะส่งผลให้สามารถสร้างตัวแบบเพื่อการพยากรณ์ที่มีความแม่นยำไม่แตกต่างกันมากนัก

เมื่อพิจารณาผลที่ได้จากปัญหาทดสอบ RM3 และ Cyclone จะเห็นได้ว่า ค่าเฉลี่ยของ RMSE ของตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ OLHD มีค่าต่ำกว่าตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ RLHD สำหรับทุกระดับของค่า  $\theta$  และเกิดการเปลี่ยนแปลงของค่าเฉลี่ย RMSE เมื่อมีการเปลี่ยนแปลงระดับของ  $\theta$  ให้มีค่าสูงขึ้น ซึ่งแสดงให้เห็นว่าความสามารถในการพยากรณ์ของตัวแบบ Kriging สำหรับปัญหาเหล่านี้ที่ระดับค่า  $\theta$  ที่มีค่าสูง จะส่งผลให้ตัวแบบมีความแม่นยำในการพยากรณ์น้อยลง เมื่อเทียบกับกรณีที่มีค่า  $\theta$  ต่ำกว่า ส่วนค่าวัดการกระจาย (SD) ของค่า RMSE ที่ได้จากแผนการทดลองทั้งหมด 10 แผนการทดลอง จะเห็นได้ว่า ตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ OLHD จะให้ค่าการกระจายของ RMSE ที่น้อยกว่าตัวแบบพยากรณ์ที่สร้างจากแผนการทดลองแบบ RLHD ซึ่งสามารถสรุปได้ว่าการใช้แผนการทดลองแบบ OLHD ในการสร้างตัวแบบพยากรณ์ จะทำให้ได้ตัวแบบที่มีความสามารถในการพยากรณ์ที่ค่อนข้างคงที่ เนื่องจาก OLHD สามารถค้นหาแผนการทดลองที่เหมาะสมทุกครั้ง แม้ว่าจะใช้แผนการทดลองเริ่มต้นที่แตกต่างกันก็ตาม

เมื่อพิจารณาผลที่ได้จากปัญหาทดสอบ Problem26 และ MS5 จะเห็นได้ว่า แผนการทดลองแบบ OLHD สามารถนำไปสร้างตัวแบบเพื่อการพยากรณ์ที่มีความแม่นยำมากกว่าแผนการทดลองแบบ RLHD เนื่องจาก ค่า RMSE เฉลี่ย ของกรณีแรกมีค่าต่ำกว่ากรณีหลัง ณ ทุกระดับค่า  $\theta$  จากการสังเกตการเปลี่ยนแปลงของค่าเฉลี่ย RMSE เมื่อมีการเปลี่ยนแปลงค่า  $\theta$  ให้สูงขึ้น จะไม่ส่งผลให้ค่าเฉลี่ย RMSE มีค่าเปลี่ยนแปลงไปมากนัก ซึ่งแสดงให้เห็นว่าการเปลี่ยนแปลงค่าของ  $\theta$  จะไม่ส่งผลกระทบต่อความสามารถในการพยากรณ์ของตัวแบบ Kriging สำหรับปัญหาเหล่านี้ เมื่อพิจารณาค่าวัดการกระจายโดยใช้ค่าส่วนเบี่ยงเบนมาตรฐาน (SD) ของค่า RMSE เมื่อใช้แผนการทดลองทั้งหมด 10 แผนการทดลอง จะเห็นได้ว่า แผนการทดลองแบบ OLHD จะให้ค่าการกระจายของ RMSE ที่น้อยกว่าแผนการทดลองแบบ RLHD ซึ่งแสดงให้เห็นว่าการเลือกใช้แผนการทดลองที่เหมาะสมที่สุดใด ๆ จาก OLHD ก็จะทำให้เกิดตัวแบบในการพยากรณ์ที่แม่นยำพอ ๆ กันนั่นเอง

นอกจากนี้เป็นที่น่าสังเกตว่า ค่าเฉลี่ยของ RMSE จากปัญหาทดสอบ Branin, Problem26 และ MS5 มีค่าที่ค่อนข้างสูง ซึ่งสามารถอธิบายได้ว่า ตัวแบบ Kriging ณ ระดับต่าง ๆ ของ  $\theta$  ในการศึกษาค้างนี้อาจจะยังไม่ดีพอสำหรับปัญหาเหล่านี้ และควรจะมีการศึกษาเทคนิคการประมาณค่าที่เหมาะสมที่สุดของ  $\theta$  ในการศึกษาเพิ่มเติมต่อไป

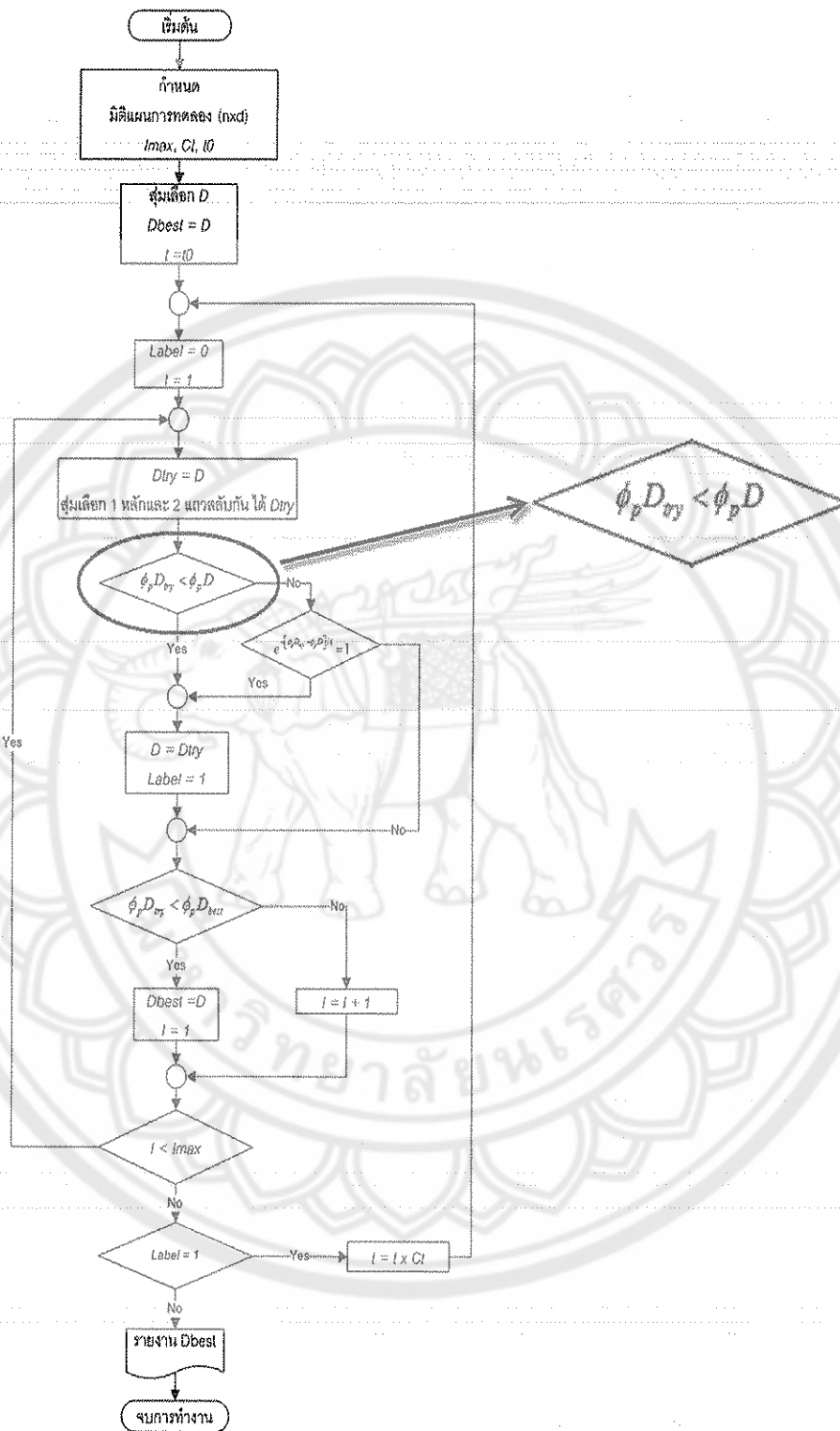
## ปรับปรุงอัลกอริทึม SA (Modified Simulated Annealing: MSA)

จากหัวข้อที่ผ่านมาสามารถสรุปได้ว่าค่าความแม่นยำในการพยากรณ์ที่ได้จากแผนการทดลองที่เหมาะสมที่สุดโดยการสืบค้นด้วย SA นั้นจะมีความแม่นยำมากกว่าแผนการทดลองแบบสุ่ม แต่หลักการทำงานของอัลกอริทึมการสืบค้น SA มีปัจจัยที่เกี่ยวข้องหลายปัจจัย เช่น การตั้งค่าพารามิเตอร์ต่างๆ ( $I_{max}, t_0, C_t$ ) การเปรียบเทียบค่า  $\phi_p$  และการคำนวณหา  $\phi_p$  เป็นต้น ซึ่งปัจจัยเหล่านี้ มีผลในการใช้เวลาในการคำนวณและกระบวนการสืบค้น ผู้วิจัยจึงได้มีการปรับปรุงอัลกอริทึม SA ในส่วนของขั้นตอนการเปรียบเทียบค่า  $\phi_p$  และการคำนวณหา  $\phi_p$  เพื่อเพิ่มประสิทธิภาพในการค้นหาแผนการทดลองที่เหมาะสมที่สุดได้รวดเร็วยิ่งขึ้น โดยยังคงได้แผนการทดลองที่มีคุณสมบัติที่ดีเท่าเดิม ซึ่งขั้นตอนในการปรับปรุง SA สามารถอธิบายได้ดังนี้

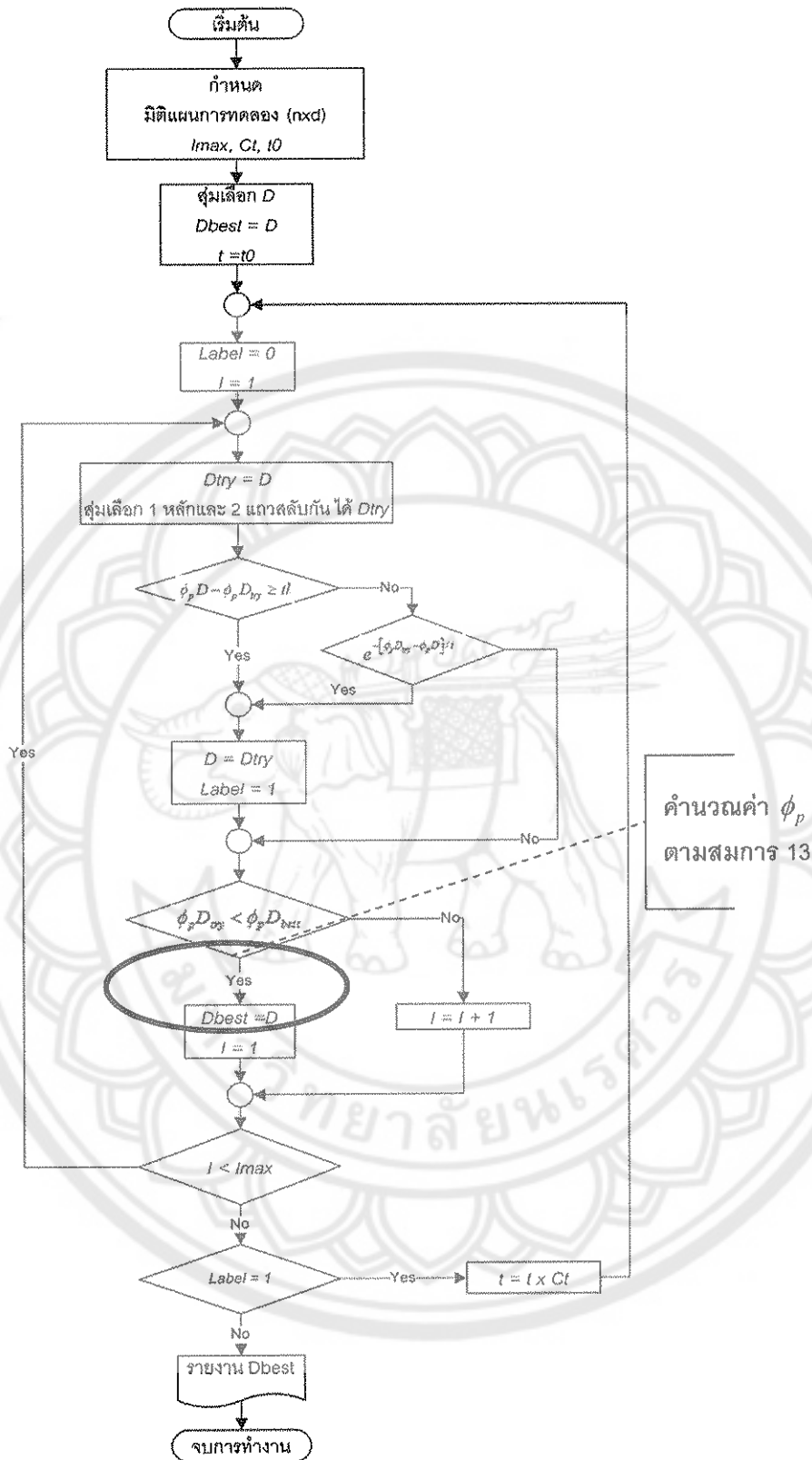
### ปรับปรุงขั้นตอนการเปรียบเทียบค่า $\phi_p$

จากผังงานของอัลกอริทึม SA ในภาพในบทที่ 3 นั้น ในขั้นตอนหลังจากการสุ่มสลับค่าในเมทริกซ์แผนการทดลองแล้ว ต้องคำนวณหา  $\phi_p$  เพื่อเปรียบเทียบว่า  $\phi_p D_{try}$  (ค่า  $\phi_p$  ใหม่หลังการสลับที่) มีค่าน้อยกว่า  $\phi_p D$  (ค่า  $\phi_p$  ก่อนหน้าหรือค่า  $\phi_p$  เริ่มต้น) ดังภาพ ซึ่งถ้าค่า  $\phi_p D_{try} < \phi_p D$  ก็ทำการแทนที่เมทริกซ์  $D$  ด้วย  $D_{try}$  แต่ถ้า ไม่เข้าเงื่อนไขก็ต้องคำนวณหาความน่าจะเป็นที่จะแทนที่ จากอัลกอริทึม SA จะเห็นว่าการเปรียบเทียบค่า  $\phi_p D_{try} < \phi_p D$  นั้นจะเกิดการแทนที่ เมทริกซ์  $D$  ด้วย  $D_{try}$  เมื่อเงื่อนไขเป็นจริงเท่านั้น และอัลกอริทึม SA ยังเปิดโอกาสให้แผนการทดลองที่ให้ค่า  $\phi_p$  ใหม่ที่มีค่ามากขึ้นจาก  $\phi_p$  เดิมด้วย โดยพิจารณาความน่าจะเป็นตามสมการ  $e^{-[\phi_p(D_{try}) - \phi_p(D)]/t}$  ที่จะเกิดการแทนที่ แต่จากงานวิจัยก่อนหน้า (Na-udom, 2007) ได้เสนอค่าคงที่ค่าหนึ่ง เรียกว่า ค่าระดับที่ยอมรับได้ (Tolerance Level,  $t$ ) ซึ่งค่านี้จะใช้เป็นค่าระดับความแตกต่างที่ยอมรับได้ ในการตัดสินใจว่าควรแทนที่เมทริกซ์  $D$  ด้วย  $D_{try}$  หรือไม่ ซึ่งตรวจสอบได้โดยพิจารณาว่า  $\phi_p D - \phi_p D_{try} \geq t$  หรือไม่ ถ้าเป็นจริงก็จะแทนที่เมทริกซ์  $D$  ด้วย  $D_{try}$  ถ้าไม่เป็นจริง ก็ไปพิจารณาความน่าจะเป็นของการแทนที่ ซึ่งค่าระดับที่ยอมรับได้ที่ถูกนำมาใช้ในงานวิจัยนี้มีค่าเท่ากับ 0.0001 เนื่องจากงานวิจัยของ Na-udom (2007) ได้ทำการศึกษาพบว่า ในการเปรียบเทียบเพื่อการแทนที่  $D$  ด้วย  $D_{try}$  จะต้องพิจารณาที่ระดับความแตกต่างที่มากพอ ซึ่งค่าที่เหมาะสมที่สุดคือ 0.0001 การเปรียบเทียบแบบนี้จะช่วยลดเวลาในการคำนวณอัลกอริทึม SA ลงได้มาก เพราะค่า  $\phi_p$  ของ  $D$  และ  $D_{try}$  ที่แตกต่างกันเพียงเล็กน้อยและมีค่าไม่เกินค่า  $t$  ก็จะไม่เกิดการแทนที่ ซึ่งจะแตกต่างจากการเปรียบเทียบแบบ  $\phi_p D_{try} < \phi_p D$  ถึงแม้ว่าค่า  $\phi_p$  ใหม่ ( $\phi_p D_{try}$ ) มีค่าแตกต่างจาก  $\phi_p$  เดิม ( $\phi_p D$ ) เพียงเล็กน้อยเท่านั้นก็จะเกิดการแทนที่ ซึ่งจะส่งผลให้ SA ใช้เวลาทำงานยาวนานขึ้น ผังงาน SA ที่เปลี่ยนแปลงขั้นตอนการเปรียบเทียบแสดงในภาพ

$$e^{-[\phi_p(D_{try}) - \phi_p(D)]/t}$$



ภาพแสดงขั้นตอนการเปรียบเทียบค่า  $\phi_p D_{iry} < \phi_p D$



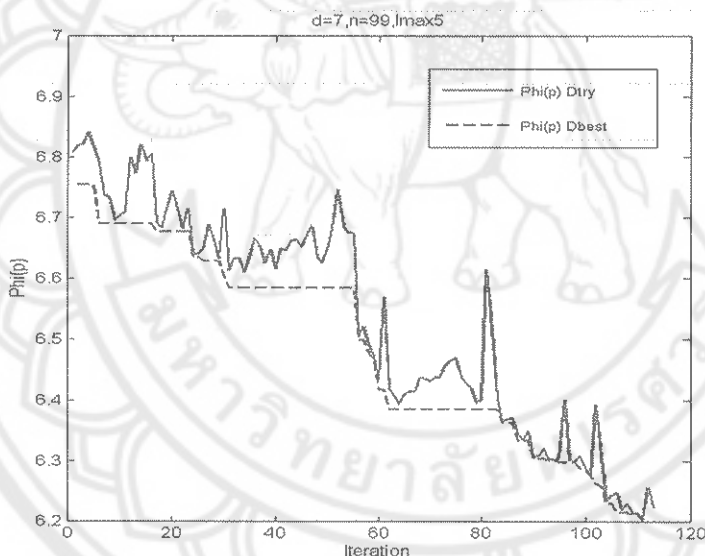
ภาพ แสดงการเปลี่ยนการเปรียบเทียบค่า  $\phi_p$  เป็น  $\phi_p D - \phi_p D_{try} \geq t$  (MSA1) จากภาพในวงรีเป็นขั้นตอนเปรียบเทียบค่า  $\phi_p$  เป็นแบบ  $\phi_p D - \phi_p D_{try} \geq t$  ที่ถูกนำมาแทนแบบเดิมที่เป็น  $\phi_p D_{try} < \phi_p D$  โดย  $t$  คำนวณระดับที่ยอมรับได้ (Tolerance Level,  $t$ ) ซึ่งมีค่าเท่ากับ 0.0001 นั้นเอง

## การคำนวณหาค่า $\phi_p$ โดยไม่ต้องมีการคำนวณซ้ำ

ในการประเมินประสิทธิภาพของอัลกอริทึม SA ในการค้นหาแผนการทดลองที่เหมาะสมที่สุดนั้น จะต้องพิจารณาค่าเกณฑ์เหมาะสมที่สุด (Optimality Criteria) ซึ่งงานวิจัยนี้ได้เลือกใช้เกณฑ์  $\phi_p$  ดังได้อธิบายไว้แล้วในหัวข้อที่ 2 ซึ่งสูตรที่มีรูปแบบอย่างง่ายที่เสนอโดย Jin, et al (2005) เป็นดังนี้

$$\phi_p = \left[ \sum_{1 \leq i < j \leq n} (d_{ij})^{-p} \right]^{1/p}$$

ค่า  $\phi_p$  ของแผนการทดลองใด ๆ จะเป็นค่าที่บ่งบอกว่าแผนการทดลองนั้น ๆ มีความเหมาะสมมากกว่าหรือด้อยกว่าแผนการทดลองอื่น ๆ หรือไม่ ถ้าค่า  $\phi_p$  มีค่าต่ำกว่าแสดงว่าแผนการทดลองนั้น ๆ มีคุณสมบัติที่ดีกว่า ซึ่งการค้นหาแผนการทดลองที่เหมาะสมที่สุดในงานด้านการจำลองการทดลองด้วยคอมพิวเตอร์ก็คือ การหาค่าต่ำสุดของ  $\phi_p$  (Minimization of  $\phi_p$ ) นั่นเอง พฤติกรรมของการเปลี่ยนแปลงค่า  $\phi_p$  ในอัลกอริทึม SA แสดงได้ดังภาพ



ภาพ แสดงพฤติกรรมของการเปลี่ยนแปลงค่า  $\phi_p$  ของ SA

จากภาพจะเห็นว่าค่า  $\phi_{p_{Dtry}}$  (เส้นทึบ) จะมีการเปลี่ยนแปลงแบบขึ้น ๆ ลง ๆ เนื่องจากในแต่ละรอบจะมีการสลับตำแหน่งแบบสุ่มระหว่างจุดทดลองใด ๆ ของแผนการทดลอง แล้วคำนวณหาค่า  $\phi_{p_{Dtry}}$  ทำให้ค่าที่ได้มีลักษณะการเปลี่ยนแปลงแบบขึ้น ๆ ลง ๆ ขึ้นอยู่กับการสลับที่ว่าจะก่อให้เกิดแผนการทดลองที่มีคุณสมบัติที่ดีขึ้นหรือไม่ ส่วนค่า  $\phi_{p_{Dbest}}$  (เส้นประ) เป็นค่า  $\phi_p$  ของแผนการทดลองที่ดีที่สุดในแต่ละรอบที่เก็บไว้ ซึ่งกราฟของค่านี้จะมีลักษณะคงที่ ณ ช่วงเวลาหนึ่ง และจะมีลักษณะที่ต่ำลงเรื่อย ๆ ถ้าเกิดการแทนที่แผนการทดลองที่ดีที่สุดที่ถูกค้นเจอก่อนหน้านี้ และค่า  $\phi_{p_{Dbest}}$  ที่ต่ำที่สุดและไม่เกิดการแทนที่เมื่อมีการสลับจุดทดลองด้วยจำนวนที่มากพอสมควร ก็จะถูกพิจารณาเป็นแผนการทดลองที่เหมาะสมที่สุด จากพฤติกรรมการคำนวณหาค่า  $\phi_p$  ในแต่ละรอบของอัลกอริทึม SA นั้น จะเห็นได้ว่าโปรแกรมต้อง



จำนวนค่า  $\phi_p$  ใหม่ทุก ๆ รอบที่เกิดการสลับที่ระหว่างจุดทดลองคู่ใด ๆ ซึ่งถ้าแผนการทดลองมีมิติที่ใหญ่ การคำนวณค่าในทุกๆ ครั้งของการสลับที่ จะเป็นการสิ้นเปลืองเวลาอย่างมากในการคำนวณ ซึ่งที่จริงแล้วจากการทำงานของอัลกอริทึม SA เมื่อมีการสลับตำแหน่งกันในแต่ละรอบการคำนวณจะมีเพียง 2 ค่า เท่านั้นที่เกิดการเปลี่ยนแปลง เพราะเกิดจากการสุ่มเลือก 1 คอลัมน์ใด ๆ แล้วสุ่มเลือก 2 แถว ในคอลัมน์ที่ถูกเลือกเพื่อทำการสลับที่ ซึ่งจะได้แผนการทดลองที่มีค่าแตกต่างกันเพียง 2 ตำแหน่งเท่านั้น

การคำนวณหาค่า  $\phi_p$  จากสมการ (10) สามารถทำได้ 3 ส่วนด้วยกัน คือการหาค่าระยะห่างจุด, การเรียงลำดับค่าระยะห่างจุด ซึ่งได้มาจากค่าระยะห่างกับค่าดัชนี และการคำนวณ  $\phi_p$  การหาค่าระยะห่างจุดจะใช้เวลาเป็น  $O(dn^2)$ , การเรียงลำดับจะใช้เวลาเป็น  $O(n^2 \log_2(n))$  (Press, et al., 1997) และการคำนวณ  $\phi_p$  จะใช้เวลาเป็น  $O(m^2 \log_2(p))$  (Jin, Chen and Sudjianto, 2003) ดังนั้นเวลาที่ใช้ทั้งหมดเท่ากับ  $O(dn^2) + O(n^2 \log_2(n)) + O(m^2 \log_2(p))$  จะเห็นว่าเวลาในการคำนวณหา  $\phi_p$  จะใช้เวลา

ก่อนที่จะแนะนำวิธีการคำนวณหา  $\phi_p$  แบบใหม่ต้องปรับสมการ (10) เพื่อหลีกเลี่ยงการเรียงลำดับระยะห่างกับค่าดัชนี ได้เป็นสมการ (11) ซึ่งเท่ากับสมการ (12) เรียกว่า  $p$ -norm

$$\phi_p = \left[ \sum_{1 \leq i < j \leq n} (1/d_{ij})^p \right]^{1/p} = \left[ \sum_{1 \leq i < j \leq n} (d_{ij})^{-p} \right]^{1/p}$$

จากสูตรการคำนวณหา  $\phi_p$  ในสมการ (12) ทำให้ปรับปรุงเวลาเป็น  $O(dn^2) + O(n^2 \log_2(p))$  การคำนวณหาค่า  $\phi_p$  โดยไม่ต้องมีการคำนวณซ้ำ จะปรับเปลี่ยนการคำนวณเฉพาะตำแหน่งที่เปลี่ยนแปลงไปเท่านั้น และไม่จำเป็นต้องคำนวณระยะห่างในทุกๆ ตำแหน่งในแผนการทดลอง ซึ่งหลักการคำนวณเป็นดังนี้

ให้  $X$  เป็นเมทริกซ์แผนการทดลองเริ่มต้น, หาค่าระยะห่างจุด โดยใช้สูตรการคำนวณระยะห่างแบบ Euclidean Distance จากสมการ (9) ใช้ได้เมทริกซ์  $d$  และหาค่า  $\phi_p$  โดยใช้สมการ (12) ขั้นตอนจะเหมือนหัวข้อ 2 ที่กล่าวไปแล้วตอนต้น แสดงดังภาพ

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}_{n \times d} \longrightarrow d(x_i, x_j) = \left[ \sum_{k=1}^d (x_{ik} - x_{jk})^2 \right]^{1/2} \longrightarrow \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1j} \\ d_{21} & d_{22} & \dots & d_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nj} \end{bmatrix}$$

$$\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1j} \\ d_{21} & d_{22} & \dots & d_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nj} \end{bmatrix} \longrightarrow \phi_p = \left[ \sum_{1 \leq i < j \leq n} (d_{ij})^{-p} \right]^{1/p} \longrightarrow \phi_p D$$

ภาพแสดงขั้นตอนการหาค่า  $\phi_p$

สุมสลัค่าในเมทริกซ์  $X$  โดยสุมเลือกคอลัมน์ ( $k$ ) และเลือกแถว 2 แถว ( $i_1, i_2$ ) ถ้าเป็นวิธีการคำนวณหา  $\phi_p$  แบบเก่า ก็ต้องทำตามขั้นตอน ดังภาพ อีกครั้ง แต่วิธีคำนวณแบบใหม่ทำได้โดยสังเกตว่า เมื่อเกิดการสลัค่าระหว่าง ( $x_{i_1k} \leftrightarrow x_{i_2k}$ ) ก็จะมีเฉพาะค่าในแถวที่  $i_1$  และ  $i_2$  กับคอลัมน์ที่  $i_1$  และ  $i_2$  ในเมทริกซ์  $X$  เท่านั้นที่เปลี่ยนแปลงไปส่วนค่าระยะห่างระหว่างจุดอื่น ๆ จะมีค่าคงที่ ดังนั้นการคำนวณหา  $\phi_p$  แบบใหม่ โดยสำหรับทุก ๆ  $1 \leq j \leq n$  และ  $j \neq i_1, i_2$  สามารถทำได้โดยการกำหนดให้

$$s(i_1, i_2, k, j) = |x_{i_2k} - x_{jk}|^t - |x_{i_1k} - x_{jk}|^t \quad (14)$$

และนิยามระยะห่างหลังจากการสลัที่เป็น

$$d'_{i_1j} = d'_{j i_1} = [d'_{i_1j} + s(i_1, i_2, k, j)]^{1/t} \quad (15)$$

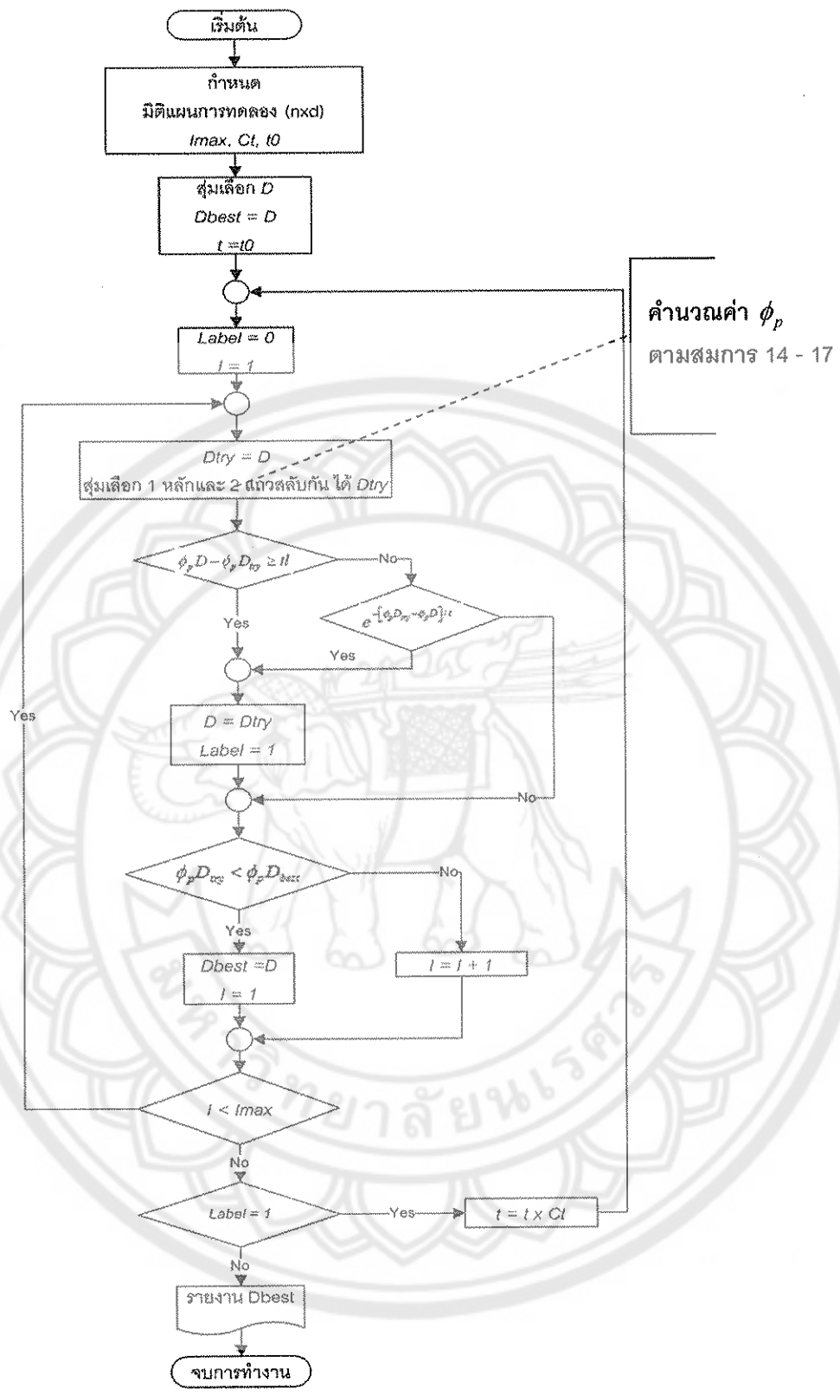
และ

$$d'_{i_2j} = d'_{j i_2} = [d'_{i_2j} + s(i_1, i_2, k, j)]^{1/t} \quad (16)$$

จากสมการ (13) - (15) การปรับปรุงค่าในแผนการทดลอง  $X$  จะใช้เวลา  $O(n)$  การคำนวณหา  $\phi_p$  แบบใหม่ หาได้จาก

$$\phi'_p = \left[ \phi_p^p + \sum_{1 \leq j \leq n, j \neq i_1, i_2} [(d'_{i_1j})^{-p} - (d_{i_1j})^{-p}] + \sum_{1 \leq j \leq n, j \neq i_1, i_2} [(d'_{i_2j})^{-p} - (d_{i_2j})^{-p}] \right]^{1/p} \quad (17)$$

โดยค่า  $t=2$  และ  $p=5$  ซึ่งการคำนวณด้วยวิธีใหม่นี้จะใช้เวลาเท่ากับ  $O(n \log_2(p))$  ทำให้การคำนวณหา  $\phi_p$  แบบใหม่ใช้เวลาทั้งหมดเป็น  $O(n) + O(n \log_2(p))$  ซึ่งจะเห็นได้ว่าวิธีใหม่นี้ใช้เวลาน้อยกว่าวิธีคำนวณแบบเดิม



ภาพแสดงการเปลี่ยนการคำนวณ  $\phi_p$  แบบไม่วนซ้ำ (MSA2)

## ผลทดสอบและวัดผลระหว่าง MSA1 กับ MSA2

จากการปรับเปลี่ยนวิธีการหา  $\phi_p$  โดยไม่ต้องมีการคำนวณซ้ำ ดังที่ได้อธิบายไว้ข้างต้น ซึ่งวิธีการนี้จะใช้วิธีการปรับค่าเฉพาะระยะห่างที่เกิดการเปลี่ยนแปลงเท่านั้น ซึ่งจะใช้เวลาเพียง  $O(n) + O(n \log_2(p))$  และเมื่อนำมาทดสอบใช้กับอัลกอริทึมการสืบค้นทั้งสองแบบคือ MSA1 และ MSA2 โดยพิจารณาเกณฑ์ค่าความเหมาะสม  $\phi_p$  โดยการสืบค้นหาแผนการทดลองที่ดีที่สุดจะใช้ค่าเริ่มต้นที่แตกต่างกันทั้งหมด 10 ครั้ง เพื่อกระบวนการสืบค้นเสร็จสิ้น ได้ทำการบันทึกแผนการทดลองที่ได้จากแต่ละอัลกอริทึม และทำการคำนวณค่าต่าง ๆ จากแผนการทดลองที่ได้ ปรากฏผลดังตาราง

ตารางแสดงเปรียบเทียบ MSA1 กับ MSA2

มิติ		แผนการทดลอง	$\phi_p$				Average Time (s)	เวลาที่ลดลง (%)
d	n		MIN	MAX	MEAN	SD		
2	9		MSA1	4.273500	4.273500	4.273500	0	19.005895
		MSA2	4.273500	4.273500	4.273500	0	14.569296	23.34
3	19	MSA1	4.907550	4.923990	4.917120	0.0147640	81.530339	
		MSA2	4.910050	4.925150	4.919335	0.0123604	41.832982	48.69
4	33	MSA1	5.205850	5.208790	5.207444	0.0043951	240.326129	
		MSA2	5.205880	5.209300	5.207675	0.0037278	77.394156	67.80
5	51	MSA1	5.421770	5.421770	5.421770	0.0030199	599.939226	
		MSA2	5.421500	5.422740	5.422104	0.0030199	117.051465	80.49
6	73	MSA1	5.590670	5.592320	5.591495	0.0035783	1371.681406	
		MSA2	5.589950	5.591980	5.590965	0.0037502	154.319319	88.75
7	99	MSA1	5.745800	5.762400	5.753610	0.0054217	1404.492691	
		MSA2	5.752970	5.753610	5.753320	0.0051843	101.488518	92.77
8	129	MSA1	5.886100	5.908000	5.898420	0.0060800	2641.087094	
		MSA2	5.897880	5.898420	5.898150	0.0044600	140.867775	94.67
9	163	MSA1	6.029400	6.045500	6.036420	0.0058380	4795.430078	
		MSA2	6.029400	6.045500	6.036420	0.0051608	170.714703	96.44
10	201	MSA1	6.159900	6.182300	6.172380	0.0066183	8436.806326	
		MSA2	6.156900	6.179800	6.170140	0.0069921	283.886448	96.64

จากตารางจะเห็นว่า MSA2 ใช้เวลาน้อยกว่า MSA1 ในการค้นหาแผนการทดลองที่เหมาะสมที่สุด โดยค่า  $\phi_p$  ของแผนการทดลองที่เหมาะสมที่สุดจากอัลกอริทึมทั้งสองมีค่าใกล้เคียงกัน เมื่อพิจารณาค่า SD จะเห็นได้ว่าการกระจายของ  $\phi_p$  ที่ได้จากแผนการทดลองจาก MSA2 จะมีค่าการกระจายที่น้อยกว่าแผนการทดลองที่ได้จาก SA ซึ่งแสดงให้เห็นว่า MSA2 สามารถค้นหาแผนการ

ทดลองที่เหมาะสมได้ค่าที่ใกล้เคียงกัน แม้ว่าจะมีการเปลี่ยนแปลงจุดเริ่มต้นในการค้นหาที่ต่างกัน สำหรับมิติปัญหาขนาดเล็ก ( $d = 2, n = 9$ ) จะเห็นได้ว่าค่า SD มีค่าเท่ากับศูนย์ ซึ่งสามารถอธิบายได้ว่า ทั้ง MSA1 และ MSA2 สามารถค้นหาแผนการทดลองที่เหมาะสมที่สุดเดิมทุกครั้ง เนื่องจากจำนวนแผนการทดลองที่เป็นไปได้ในคลาสมีจำนวนน้อย จึงทำให้การค้นหาเป็นไปได้ง่ายนั่นเองเมื่อพิจารณาเปอร์เซ็นต์ของเวลาในการสืบค้นที่ลดลงของ MSA2 เมื่อเทียบกับ MSA1 จะเห็นได้ว่า เปอร์เซ็นต์ของเวลาที่ลดลงนี้จะมีค่ามากขึ้นเรื่อย ๆ เมื่อมิติปัญหามีค่ามากขึ้น จึงสามารถสรุปได้ว่าการปรับปรุงขั้นตอนการคำนวณค่า  $\phi_p$  มีส่วนทำให้เวลาในการค้นหาแผนการทดลองที่ดีที่สุดลดลงอย่างมาก โดยเฉพาะอย่างยิ่งเมื่อต้องการค้นหาแผนการทดลองที่เหมาะสมที่สุดที่มีขนาดใหญ่ขึ้น โดยสังเกตได้จากเวลาที่ลดลงถึงประมาณร้อยละ 90 สำหรับแผนการทดลองที่มีขนาดใหญ่ โดยที่แผนการทดลองที่เหมาะสมที่สุดนั้นยังคงมีคุณสมบัติที่ดีเทียบเท่ากับแผนการทดลองที่ได้จากการค้นหาด้วย MSA1 พื้นฐาน ดังนั้น MSA2 จึงมีความเหมาะสมและควรนำไปใช้ในการค้นหาแผนการทดลองที่เหมาะสมที่สุดภายใต้เกณฑ์  $\phi_p$  สำหรับมิติปัญหาต่าง ๆ โดยเฉพาะอย่างยิ่งปัญหาที่มีประกอบด้วยจำนวนตัวแปรอิสระและจำนวนรันมาก ๆ





## บทที่ 5

### สรุปผลการวิจัย

#### สรุปผลการวิจัย

การจำลองการทดลองด้วยคอมพิวเตอร์ถูกนำมาใช้อย่างแพร่หลายในด้านวิทยาศาสตร์ประยุกต์และด้านวิศวกรรมศาสตร์ เนื่องจากการทดลองประเภทนี้มีความซับซ้อน และใช้เวลายาวนานมากในการประมวลผล ดังนั้นการวางแผนการทดลองจึงมีบทบาทสำคัญในการเลือกชุดของตัวแปรเข้าที่เหมาะสมที่สุด ก่อนจะทำการจำลองการทดลองด้วยคอมพิวเตอร์ สำหรับมีติปัญหาที่กำหนดใด ๆ จะเห็นได้ว่าจำนวนแผนการทดลองที่เป็นไปได้มีจำนวนมากมายมหาศาล ดังนั้นการเลือกแผนการทดลองที่เหมาะสมที่สุด จะต้องใช้อัลกอริทึมการสืบค้น (Search Algorithm) ควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสมที่สุด (Optimality Criteria) ซึ่งอัลกอริทึมการสืบค้นที่ถูกนำมาใช้อย่างแพร่หลายในงานด้านการจำลองการทดลองด้วยคอมพิวเตอร์คือ SA ซึ่ง SA มีประสิทธิภาพมากในการค้นหาแผนการทดลองที่ดีที่สุด เนื่องจาก SA มีกลไกที่สามารถหลุดจากค่าที่ดีที่สุดในระดับกลุ่ม (Local Minimum) ได้ อย่างไรก็ตาม SA มักใช้เวลายาวนานในการค้นหาแผนการทดลองที่เหมาะสมที่สุดสำหรับมีติปัญหาที่ประกอบด้วยจำนวนตัวแปรเข้าที่มีขนาดใหญ่ ดังนั้น การศึกษานี้จึงมุ่งเน้นในการพัฒนาและปรับปรุง SA เพื่อให้สามารถทำงานได้รวดเร็วขึ้น ภายใต้เกณฑ์การหาค่าที่เหมาะสม  $\phi_p$  ซึ่งหลักการโดยรวมคือการเน้นการปรับปรุง 2 ขั้นตอน คือ การกำหนดค่าระดับที่ยอมรับได้ในการแทนที่ (MSA1) และขั้นตอนในการคำนวณค่า  $\phi_p$  (MSA2) จากนั้นทำการเปรียบเทียบประสิทธิภาพของ MSA1 และ MSA2 ภายใต้มีติปัญหาต่าง ๆ ซึ่งผลที่ได้จากการศึกษาครั้งนี้ สามารถสรุปได้ดังต่อไปนี้

1. แผนการทดลองที่เหมาะสมที่สุดที่ได้จากการค้นหาด้วย SA สามารถนำไปสร้างตัวแบบเพื่อการพยากรณ์ที่มีความแม่นยำมากกว่าแผนการทดลองแบบสุ่ม
2. อัลกอริทึมการสืบค้น MSA2 ใช้เวลาน้อยกว่า MSA1 ในการค้นหาแผนการทดลองที่เหมาะสมที่สุดที่มีคุณสมบัติใกล้เคียงกัน โดยใช้เวลาน้อยลงสูงสุดถึงประมาณร้อยละ 97

#### ข้อเสนอแนะ

ผลสรุปที่ได้จากการศึกษานี้พบว่า อัลกอริทึมการสืบค้นแบบ MSA2 สามารถค้นหาแผนการทดลองที่เหมาะสมได้ภายในเวลาที่รวดเร็วขึ้นภายใต้เกณฑ์  $\phi_p$  และสามารถนำ MSA2 ไปประยุกต์ใช้ใน

ปัญหาต่าง ๆ ที่เกี่ยวข้องกับ การหาค่าความเหมาะสม (Optimization) ได้ อย่างไรก็ตามผู้วิจัยมีข้อเสนอแนะ ดังต่อไปนี้

ควรมีการพิจารณาเกณฑ์หาค่าที่เหมาะสมที่สุดแบบอื่น ๆ เช่น Maximin, Entropy เพื่อประยุกต์ใช้กับ MSA2

ควรมีการขยายมิติของแผนการทดลองให้มีขนาดใหญ่ขึ้น เพื่อให้ขอบเขตในการสรุปผลสามารถทำได้กว้างขึ้น

ควรมีการศึกษอัลกอริทึมการสืบค้นแบบอื่น ๆ เช่น Tabu Search, Particle Swarm Optimization มาประยุกต์ใช้ในการค้นหาแผนการทดลองที่เหมาะสมที่สุดเพื่อใช้ในการจำลองการทดลองด้วยคอมพิวเตอร์





## บรรณานุกรม

- บุญเสริม บุญเจริญผล. (1 พฤศจิกายน 2548). การจำลองเหตุการณ์ (SIMULATION). สืบค้นเมื่อ 27 มิถุนายน 2550, จาก [http://www.krirk.ac.th/education/dr\\_boonserm/simulation.pdf](http://www.krirk.ac.th/education/dr_boonserm/simulation.pdf)
- วิกิพีเดีย. (11 January 2007). ขั้นตอนวิธีเชิงพันธุกรรม. สืบค้นเมื่อ 17 กรกฎาคม 2550, จาก [http://th.wikipedia.org/wiki/Genetic\\_algorithm](http://th.wikipedia.org/wiki/Genetic_algorithm)
- Allen, T. T., Bernshteyn, M. A., Kabiri-Bamoradian, K. (2003). Constructing meta-model for computer experiments. **Journal of Quality Technology**, 35, 264-274.
- Box, G. E. P., and Wilson, K. B. (1951). On the experimental attainment of optimal conditions. **Journal of the Royal Statistic Society**. 13, 1-38.
- Chapin, N. (1997). Flowcharting with the ANSI standard: A tutorial. **ACM Computing Surveys (CSUR)**, 2(2), 119 – 146.
- Eppen, G.D., Gould, F.J., and Schmidt. (1988). **Introduction to management science** (2<sup>nd</sup> ed.) Englewood, New Jersey: Prentice – Hall.
- Fang, K. T. (1980). Experimental design by uniform distribution. **Acta Mathematicae Applicatae Sinica**, 3, 363-372.
- Fang, K T., Li, R. and Sudjianto, A. (2005). **Design and modeling for computer experiments**. New York: Chapman & Hall
- Giunta, A. A., Wojtkiewicz, S. F. Jr. and Eldred, M. S. (2003). Overview of modern design of experiments methods for computational simulations. In **Proceedings of the 41st AIAA Aerospace Sciences Meeting and Exhibit**. Reno: NV.
- Hillier, F. S. and Lieberman, G. J. (2005). **Introduction to operation research** (8<sup>th</sup> ed.). Boston, Massachusetts: McGraw-Hill.
- Jin, R., Chen, W. and Sudjianto, A. (2005). An efficient algorithm for constructing optimal design of computer experiments. **Journal of Statistical Planning and Inference**, 134, 268-287
- Johnson, M., Moore, L. and Ylvisaker, D. (1990). Minimax and maximin distance designs. **Journal of Statistical Planning and Inference**, 26, 131-148.
- Kalagnanam, J. R., and Diwekar, U. M. (1997). An efficient sampling technique for off-line quality control. **Technometrics**, 39(3), 308-319.

Koehler, J. and Owen, A. B. (1996). Computer experiment, In **Handbook of Statistics** (Vol 13, pp.261-308). New York: Elsevier Science.

McKay, M. D., Beckman, and Conover, W. J. (1979) A comparison of three methods for selecting values of input variable in the analysis of output from a computer code. **Technometrics**, 21, 239-246.

Montgomery, D.C. (2005). **Design and analysis of experiments** (6<sup>th</sup> ed.). New York: John Wiley & Sons.

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

Na-udom, A. (2007). **Experimental Design Methodology For Modeling Response From Computer Simulated Experiments**. Doctoral dissertation, Ph.D., Curtin University of Technology, Australia.

Palmer, K. and Tsui, K. L. (2001). A minimum bias latin hypercube design. **IIE Transactions**, 33, 793-808

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1997). **Numerical recipes in C: The art of scientific computing**. Cambridge: Cambridge University Press.

Rungrattanaubol, J and Na-udom, A. (2007). Comparison of evolutionary search algorithms in computer simulated experiments. **National Computer Science and Engineer Conference (NCSEC2007)**, 11, 102-106

Sacks, J., Schiller, S.B. and Welch, W.J. (1989). Design for computer experiments. **Technometrics**, 34, 15-25

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

Ye, K. Q., Li, W. and Sudjianto A. (2000). Algorithmic construction of optimal symmetric latin hypercube designs. **Journal of Statistical Planning and Inference**, 90, 145-159.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992). Screening predicting and computer experiments. **Technometrics**, 34, 15-25.

## ภาคผนวก

บทความวิจัยเพื่อนำเสนอในงานประชุมวิชาการด้านการวิจัยดำเนินงานแห่งชาติ ประจำปี 2551 ที่จัดขึ้น ณ โรงแรมเจ้าพระยาปาร์ค กรุงเทพมหานคร ในวันที่ 24-25 กรกฎาคม 2551





operations Research

Co-operative Research Network  
Operations Research

The 5<sup>th</sup> National Conference of  
Operations Research 2008

24-25 July 2008

Chaophya Park Hotel, Bangkok



# การประชุมวิชาการด้านทฤษฎีคณิตศาสตร์ ประจำปี 2551

## ตารางการดำเนินงาน

วันที่	เวลา	หัวข้อการประชุม	วิทยากร	สถานที่
วันที่ 24 กรกฎาคม 2551				
08.30 - 9.00 น.		ผู้ช่วยศาสตราจารย์ นริศ อ่องคำ		ห้องประชุมที่ 1 : Grand Reception Ballroom
9.00 น.		อธิการบดี มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ		
09.15 - 10.45 น.		Keynote: รองศาสตราจารย์ ดร.สุวิ อักษรกิตต์		
10.45 - 11.00 น.		เรื่อง "การจัดการทางทฤษฎีคณิตศาสตร์ในประเทศไทย"		
11.00 - 12.00 น.		Applied Optimization Problems : อรรถวิทย์ นฤมิตร		อาหารว่าง
11.00 - 11.20 น.		การจัดการเชิงคณิตศาสตร์ในทางทฤษฎีการสื่อสาร		การพัฒนาและจัดการเรียนการสอนทางทฤษฎีการสื่อสาร (แบบผสมผสาน)
11.20 - 11.40 น.		การวิเคราะห์เชิงคณิตศาสตร์ของปัญหาการกระจายตัว		โดเมนของทฤษฎี จำนวน (ในพระนครปฐม)
11.40 - 12.00 น.		ตัวแบบเชิงคณิตศาสตร์ของปัญหาการกระจายตัว		ระบบการกระจายตัวสำหรับปริภูมิเชิงเส้น
11.40 - 12.00 น.		การวิเคราะห์เชิงคณิตศาสตร์ของปัญหาการกระจายตัว		การวิเคราะห์ทางทฤษฎีการกระจายตัว
12.00 - 13.00 น.		การวิเคราะห์เชิงคณิตศาสตร์ของปัญหาการกระจายตัว		และเมตริกซ์ของทฤษฎีการกระจายตัว (LCD) ถึงทฤษฎีการกระจายตัว
13.00 - 14.40 น.		Applied Optimization Problems and Heuristic Methods : อรรถวิทย์ นฤมิตร		Applied Optimization Research Applied Statistics : อรรถวิทย์ นฤมิตร
13.00 - 13.20 น.		การประยุกต์ใช้ทฤษฎีการกระจายตัว		การประยุกต์ใช้ทฤษฎีการกระจายตัวในการจัดการข้อมูล
13.20 - 13.40 น.		Modification on Search Algorithm for Computer Simulated Experiment		An Application of Logistic Regression for the Efficiency Improvement of Bored Pile Casting in Wet Process System of
13.40 - 14.00 น.		การประยุกต์ใช้ทฤษฎีการกระจายตัว		Thawelchai Thampetarak and Anai Welanungsut
14.00 - 14.20 น.		การประยุกต์ใช้ทฤษฎีการกระจายตัว		การวิเคราะห์ทฤษฎีการกระจายตัว
14.20 - 14.40 น.		การประยุกต์ใช้ทฤษฎีการกระจายตัว		การประยุกต์ใช้ทฤษฎีการกระจายตัวในการจัดการข้อมูล
14.40 - 15.00 น.		Optimization Theory : อรรถวิทย์ นฤมิตร		Applied Statistics : อรรถวิทย์ นฤมิตร
15.00 - 16.20 น.		การประยุกต์ใช้ทฤษฎีการกระจายตัว		พหุคูณกำลังสองเชิงทฤษฎีการกระจายตัว
15.00 - 15.20 น.		ตัวแบบความคาดหมายเชิงคณิตศาสตร์		การศึกษาเชิงทฤษฎีการกระจายตัว
15.20 - 15.40 น.		การประยุกต์ใช้ทฤษฎีการกระจายตัว		การศึกษาเชิงทฤษฎีการกระจายตัว
15.40 - 16.00 น.		การประยุกต์ใช้ทฤษฎีการกระจายตัว		การศึกษาเชิงทฤษฎีการกระจายตัว
16.00 - 16.20 น.		Solving Non-linear Continuous Mathematical Models using Shuffled Frog Leaping and Memetic Algorithms		การศึกษาเชิงทฤษฎีการกระจายตัว





# การประชุมวิชาการด้านการศึกษาแห่งชาติ ประจำปี 2551

## เอกสารประกอบการสัมมนา

วันที่	เวลา	หัวข้อสัมมนา	วิทยากร
วันที่ 25 กรกฎาคม 2551			
08.30 - 9.00 น.		หัวข้อสัมมนาที่ 1: Grand Re-charge Balloon ผู้เข้าประชุมวิชาการลงทะเบียน และรับเอกสารการประชุม	โพนพิบูลย์ วัชรวิทย์, Rachada J
09.00 - 10.30 น.		บรรยายพิเศษ เรื่อง "การวิจัยดำเนินงานในระบบการจราจรและการขนส่ง" ดร.วชิรศักดิ์ กักรอดิคม	อาจารย์ ดร.วชิรศักดิ์ กักรอดิคม
10.30 - 10.45 น.		บรรยายพิเศษ เรื่อง "การวิจัยดำเนินงานกับงานทางการเดิน" ดร.ไพฑูริ เวทว	อาจารย์ ดร.ไพฑูริ เวทว
10.45 - 13.00 น.		Applied Operations Research: An Introduction การประยุกต์ใช้ขั้นตอนวิธีเส้นทางที่สั้นที่สุดสำหรับการขนส่งของระบบการเลือกตั้งแบบอนุกรม	น.ส.พัชราภา วัชรวิทย์ และ อ.วชิรศักดิ์ กักรอดิคม
13.00 - 14.15 น.		ประเภทและสัณฐานศิลป์เส้น รุ่งรัตน์ วิไลชัย และ ภาคภูมิ พันธุ์วิเศษ	น.ส.รุ่งรัตน์ วิไลชัย และ น.ส.ภาคภูมิ พันธุ์วิเศษ
13.00 - 13.20 น.		การนำเทคโนโลยีการติดตามรถโดยสารประจำทางมาใช้ พัฒนพงษ์ เจริญเมือง, มณฑล ศานต์นันทน์ และ เสฐียร ศานต์นันทน์	น.ส.พัฒนพงษ์ เจริญเมือง, น.ส.มณฑล ศานต์นันทน์ และ น.ส.เสฐียร ศานต์นันทน์
13.20 - 13.40 น.		การประยุกต์ใช้ทฤษฎี ANFIS ในการวิเคราะห์ข้อมูลจราจรทางบก เปี่ยมศักดิ์ วิศิษฐาภรณ์, กฤษณะ ชินสร และ ณรงค์ อิมภักดิ์	น.ส.เปี่ยมศักดิ์ วิศิษฐาภรณ์, น.ส.กฤษณะ ชินสร และ น.ส.ณรงค์ อิมภักดิ์
13.40 - 14.00 น.		การประยุกต์ใช้วิธีลดต้นทุนเงินหรือหุ้นสำหรับปัญหาการจัดสรรงาน เปี่ยมศักดิ์ วิศิษฐาภรณ์, กฤษณะ ชินสร และ ณรงค์ อิมภักดิ์	น.ส.เปี่ยมศักดิ์ วิศิษฐาภรณ์, น.ส.กฤษณะ ชินสร และ น.ส.ณรงค์ อิมภักดิ์
14.00 - 14.20 น.		การแก้ปัญหาการจัดสรรงานของพนักงานของห้างสรรพสินค้า เปี่ยมศักดิ์ วิศิษฐาภรณ์, กฤษณะ ชินสร และ ณรงค์ อิมภักดิ์	น.ส.เปี่ยมศักดิ์ วิศิษฐาภรณ์, น.ส.กฤษณะ ชินสร และ น.ส.ณรงค์ อิมภักดิ์
14.20 - 14.40 น.		การแก้ปัญหากำหนดการเพิ่มผู้โดยสาร วิชา เลิศพัฒนพงศ์, สุวิทย์ ไชยเดช, จิรประภา นุ่มน้อย และ ศิษุภย์ ชาญเจริญกุล	น.ส.วิชา เลิศพัฒนพงศ์, น.ส.สุวิทย์ ไชยเดช, น.ส.จิรประภา นุ่มน้อย และ น.ส.ศิษุภย์ ชาญเจริญกุล
14.40 - 15.00 น.		Optimization Theory for Inventory Management Sakon Wongmongkoltip	น.ส.ศศิมา นุ่มน้อย และ ศาสตราจารย์ ดร.ศศิมา นุ่มน้อย
15.00 - 15.20 น.		8 Baskets Inventory System	น.ส.ศศิมา นุ่มน้อย และ ศาสตราจารย์ ดร.ศศิมา นุ่มน้อย
15.20 - 15.40 น.		แบบจำลองสินค้าคงคลังแบบสี่เหลี่ยมมุมฉากที่มีค่าความหนาแน่นต่างกัน อานันท์ วัชรวิทย์, วิชัย นุ่มน้อย และ อรรถกร เก่งพล	น.ส.อานันท์ วัชรวิทย์, น.ส.วิชัย นุ่มน้อย และ น.ส.อรรถกร เก่งพล
15.40 - 16.00 น.		การวางแผนการผลิตตามการใช้ปริมาณความต้องการในหน่วยโดยใช้วิธีการตามความถี่สูง พิศมัย แสงชูโต, ศิษุภย์ ชาญเจริญกุล และ ศิษุภย์ ชาญเจริญกุล	น.ส.พิศมัย แสงชูโต, น.ส.ศิษุภย์ ชาญเจริญกุล และ น.ส.ศิษุภย์ ชาญเจริญกุล
16.00 - 16.20 น.		วิธีการที่มีประสิทธิภาพซึ่งที่สุดแบบไม่เชิงเส้นและมีข้อจำกัดสำหรับระบบการเชื่อมด้วยเลเซอร์ ไพฑูริ เวทว และ พงษ์พัฒน์ เหลืองไพฑูริ	น.ส.ไพฑูริ เวทว และ น.ส.พงษ์พัฒน์ เหลืองไพฑูริ
16.20 - 16.40 น.			



## Modification on Search Algorithm for Computer Simulated Experiment

Thammarat Thamma<sup>1</sup>, Jaratsri Rungrattanaubol<sup>2\*</sup> and Anamai Na-udom<sup>3</sup>

### Abstract

Currently computer simulated experiments (CSE) have been extensively used in sciences and engineering applications. Selecting a proper design to run CSE is very critical for reliability of output response. Normally the best design is generated by using a search algorithm along with a pre-specified optimality criterion. This paper performs the modification on SA (Simulated Annealing) for CSE to improve the efficiency. From the empirical studies, the results indicate that the time required for modified SA to reach the same level of optimal design is significantly lower than that of the original SA.

**Keywords:** computer simulated experiment, search algorithm, optimality criteria

<sup>1</sup> Master student, Department of Computer Science and Information Technology, Naresuan University

<sup>2\*</sup> Lecturer, Department of Computer Science and Information Technology, Naresuan University (Corresponding Author)

<sup>3</sup> Lecturer, Department of Mathematics, Naresuan University



## 1. Introduction

Recently computer simulated experiments (CSEs) have been extensively used to investigate a sophisticated complex phenomena, especially when classic (physical) experiments are not feasible. For instance, the use of reservoir simulator to predict ultimate recovery of oil [3], the use of finite element codes to predict behavior of a metal structure under stress [8], and so on. Typically the process of CSE is considered as a black box and not known a priori. Simpson et al. [9] classified the basic steps of CSE into 3 parts: the choice of experimental design, simulation routine (black box) and developing an approximation model, respectively. These three steps are visualized in Figure 1.

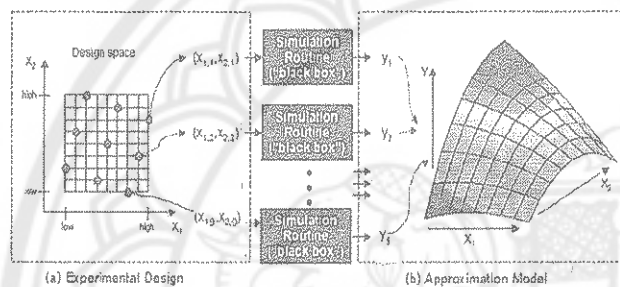


Figure 1 Computer simulated experiment [9]

Usually computer simulated experiments are computationally expensive and time consuming. Hence many efforts have gone to developing an accurate approximation model based on handful of runs. A selection of design (design of experiment) to run CSE (Figure 1(a)) is critical for the accuracy of the developed approximation model (Figure 1(b)). As computer simulated experiments are deterministic in nature, the characteristics of classical experimentations such as blocking, replication and randomization are irrelevant [8]. Therefore, the key strategy of designs for CSE is focused on spreading the design points over the region of interest. Such designs are called space filling designs. The example of a classical design and a space filling design is shown in Figure 2. In a classical design, the design points are concentrated on the boundary and center whereas in a space filling design, design points are uniformly spread.

Normally the space filling designs can be constructed by searching for a good design through the search algorithms under a pre-specified optimality criterion.

In the context of CSE, Latin hypercube designs (LHD), proposed by Mackay et al. (1979) are widely used since its construction is very simple in practice.

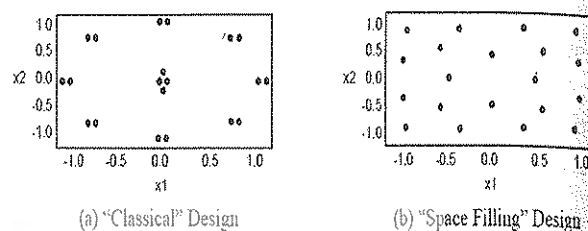


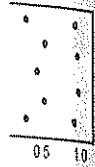
Figure 2 "Classical" and "Space Filling" Designs

The CSE are usually complex and consist of many input variables to investigate so a large number of runs are required to estimate all related parameter in the model. For instance, if the problem of interest consists of  $d$  input variables and  $n$  number of runs, the total number of all possible LHD is  $(n!)^d$ . Obviously this number explodes as values of  $n$  and  $d$  increase. Therefore the full space of LHD cannot be explored. In such cases we need search algorithm to lead us to a good design with respect to an optimality criteria. The key idea of all existing search algorithms is to use some kinds of exchange procedures to move towards better a design.

In practice, for a given optimality criterion, the search algorithm will look for a better LHD, until pre-specified tolerance level is achieved. A range of search algorithms and optimality criteria have been proposed in the context of CSE. For example, Johnson et al. [3] proposed the optimality criteria called minmax and maximin distance criteria to measure the goodness of the optimal LHD. Morris and Mitchell [6] adapted a version of Simulated annealing algorithm (SA) to search for the optimal LHD by considering  $\phi_p$  optimality criteria. Ye et al [10] proposed the construction of optimal symmetric Latin hypercube design (SLHD) and modified the columnwise-pairwise exchange algorithm (CP) to construct the optimal SLHD by considering maximin distance criterion and  $\phi_p$  optimality criteria. Jin et al. [1] developed a new algorithm called Enhanced Stochastic Evolutionary (ESE) and modified the calculation of various optimality criteria (including  $\phi_p$  criteria). Rungrattanaubol and Na-udom [7] presented a comparison on the performance of Genetic algorithm (GA) [4] and Simulated annealing algorithm (SA) with respect to  $\phi_p$  optimality criteria. The results indicate that SA performs much better than GA in terms of the simplicity of parameters setting and rate of



), proposed  
instruction is



Design

many input  
e required to  
tance, if the  
n number of  
. Obviously  
e. Therefore  
h cases we  
th respect to  
sting search  
res to move

the search  
ed tolerance  
rd optimality  
For example,  
lled minimax  
ness of the  
version of  
optimal LHD  
proposed the  
sign (SLHD)  
gorithm (CP)  
min distance  
oped a new  
(ESE) and  
ia (including  
resented a  
(GA) [4] and  
 $\phi_p$  optimality  
better than  
d rate of

convergence. Though SA has shown high performance in searching for an optimal LHD, SA still requires such a long time to search for the optimal LHD, especially when the dimension of the problems is large. Therefore this paper aims to improve the performance of SA by using the idea of setting the tolerance level to terminate SA and adjusting the calculation of the optimality criteria. In the next section, we will present the research method including the design used in this study, details of SA and the modifications of SA, followed by the results and conclusion.

## 2. Research Method

This paper aims to improve the performance of SA to search for the best LHD under  $\phi_p$  optimality criteria. We first developed both of original SA (SA) and modified SA (MSA) in MATLAB codes and simulated them, and then the optimal value of  $\phi_p$  criteria for specified dimensions of problem are recorded and compared. The details of design used, LHD, the steps of an original SA, and the modifications of SA based on avoiding of re-calculating of  $\phi_p$  criteria are presented as follows.

### 2.1 Latin hypercube design (LHD)

LHD has been extensively used in the context of CSE. It was originally proposed by McKay et al. [5]. LHD is a matrix ( $X$ ), of  $n$  rows and  $d$  columns where  $n$  is the number of runs and  $d$  is the number of input variables. LHD can be constructed based on the idea of stratified sampling to ensure that all subregions in the divided input variable space will be sampled with equal probability. A Latin hypercube sampling has

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

where  $\pi_{ij}$  are the elements of an  $n \times d$  matrix comprising of columns  $\pi_i (i = 1, 2, \dots, d)$ . Each column  $\pi_i (i = 1, 2, \dots, d)$  is independent random permutation of number  $1, 2, \dots, n$  and  $U_{ij}$  are  $n \times d$  values of i.i.d. uniform  $U[0,1]$  random variables independent of the  $\pi_{ij}$ . An example of LHD is presented in Table 1.

Table 1 The  $5 \times 4$  random LHD

$X_1$	$X_2$	$X_3$	$X_4$
1	2	5	5
2	5	3	2
3	3	1	4
4	4	2	1
5	1	4	3

As can be seen from Table 1, each column in the design matrix is referred to each input variable, while each row is referred to the design run. If an exchange takes place in any two pairs of runs within randomly selected column, a new design is occurred. LHDs are widely used in practice since its property still remains after any exchange procedures.

### 2.2 Simulated annealing algorithm (SA)

Morris and Mitchell [6] adapted simulated annealing search algorithm to find the optimal design with respected to  $\phi_p$  criteria. The steps of SA are based on the idea of exchanging two random points (Perturbation) of a dimension in the design. However, in the SA algorithm, an exchange with no improvement or deterioration in the  $\phi_p$  criteria may be also retained during search process. The steps are presented below and its flowchart is displayed in Figure 3.

**Step 1:** Set initial value of  $l_{max}$  (maximum number of perturbation to seek improvement),  $t_0$  (initial cooling temperature) and  $C_t$  (factor by which  $t_0$  is reduced when no improvement in  $\phi_p$  is attained). The factor  $t_0$  controls the probability of retaining a perturbation that results in no improvement of  $\phi_p$ . In particular, higher value of  $t_0$  leads to higher probability of retaining a non-improving perturbation. As search process continues  $t_0$  is changed by a factor  $C_t$  resulting in lower value of  $t_0$  and hence lower probability of retaining non improving perturbation of same order. Specify the dimension of the problem  $n$  and  $d$ .

**Step 2:** Generate a random LHD  $X$  of given order  $n \times d$ . Let

$$X_{best} = X, t = t_0$$

**Step 3:** Set  $l = 1, Label = 0$

**Step 4:** Let  $X_{ly} = X$

Randomly select a column say  $j$ , of matrix  $X_{ly}$  and exchange two randomly selected elements of column  $j$ , say  $X_{aj} \leftrightarrow X_{bj}$





Step 5: Set  $X = X_{try}, Label = 1$ ,

If  $\phi_p X - \phi_p X_{try} \geq tl$  or with probability  $e^{-[\phi_p X_{try} - \phi_p X] / tl}$

Step 6: If  $\phi_p X_{try} < \phi_p X_{best}$ , set  $l = 1$  and  $X_{best} = X_{try}$ ,  
else  $l = l + 1$ .

Step 7: If  $l < l_{max}$  go to Step 4.

Step 8: If  $Label = 1$ , set  $t = t \times C_l$ , go to Step 3.

Step 9: Stop and report  $X_{best}$ .

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

Let a Euclidean distance list  $(d_1, d_2, \dots, d_m)$ , is the elements list from the smallest to the largest. Then define an index list  $(J_1, J_2, \dots, J_m)$ ,  $J_j$  is the number of pairs of sites in the design separated by distance  $d_j$ . Thus  $X$  is a maximin design if among available designs, it maximizes  $d_1$  while  $J_1$  is minimized. The scalar criterion can be expressed as

$$\phi_p = \left[ \sum_{j=1}^m J_j d_j^{-p} \right]^{1/p} \quad (3)$$

where  $p$  is a positive integer,  $J_j$  and  $d_j$  are specified from  $X$ . The design that minimizes  $\phi_p$  is the optimal LHD in the class. In this study, the adaptive form of  $\phi_p$  [1] which is simpler than equation (3) is considered.

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

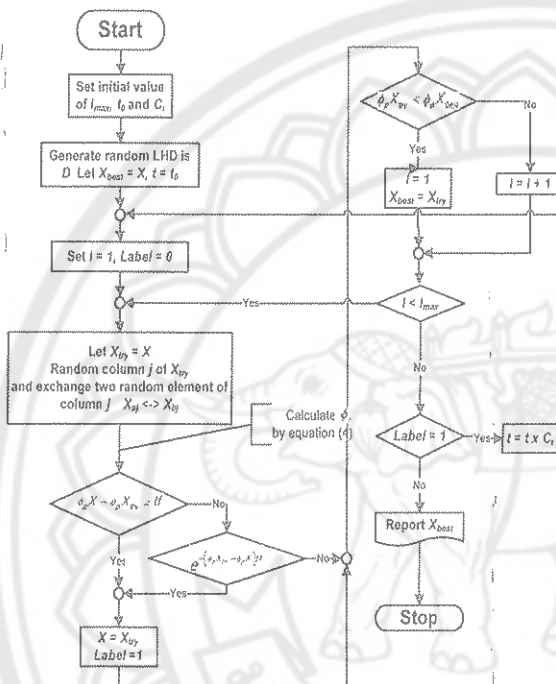


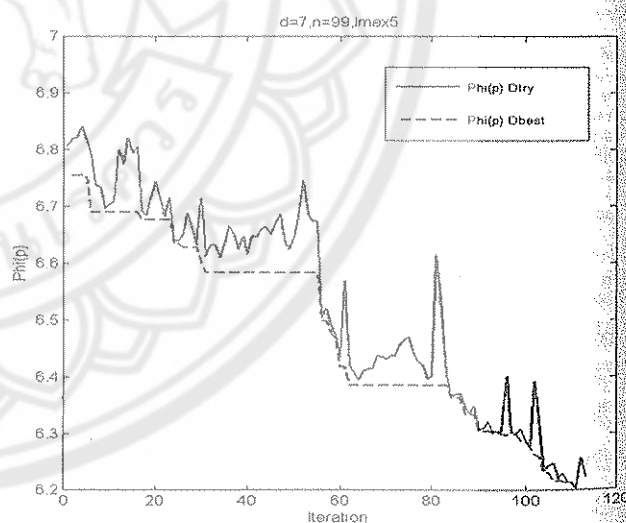
Figure 3 A flowchart of SA

### 2.3 The $\phi_p$ criterion

Morris and Mitchell [6] proposed a modification class of maximin distance criterion in order to search for the optimal design. For a given design  $X$ , the Euclidean intersite distance between the design points can be calculated from

$$d(x_i, x_j) = \left[ \sum_{k=1}^d (x_{ik} - x_{jk})^2 \right]^{1/2} \quad (2)$$

By using equation (2) to calculate all intersite distance for every pairs of design points in the design matrix will result in a symmetric matrix of intersite distance as follows.



The behavior of SA process with respect to the value of  $\phi_p$  for each iteration is displayed as Figure 4.

Figure 4 The pattern of SA process with respect to  $\phi_p$



### 2.4 Modified simulated annealing (MSA)

This paper will modify the calculation of  $\phi_p$  criteria by using the method that avoids re-calculating  $\phi_p$  value. As mentioned before, SA uses the exchange procedure between two pairs of points within the randomly selected column. Hence, after an exchange between rows  $i_1$  and  $i_2$  within column  $k$  ( $x_{i_1k} \leftrightarrow x_{i_2k}$ ), only elements in rows  $i_1$  and  $i_2$ , and columns  $i_1$  and  $i_2$  are changed in the distance matrix  $D$  [1].

For any  $1 \leq j \leq n$  and  $j \neq i_1, i_2$  let:

$$s(i_1, i_2, k, j) = |x_{i_2k} - x_{jk}| - |x_{i_1k} - x_{jk}| \quad (5)$$

then

$$d'_{i_1j} = d'_{j i_1} = [d_{i_1j} + s(i_1, i_2, k, j)]^p \quad (6)$$

and

$$d'_{i_2j} = d'_{j i_2} = [d_{i_2j} + s(i_1, i_2, k, j)]^p \quad (7)$$

Thus new  $\phi_p$  is computed by

$$\phi_p' = \left[ \phi_p^p + \sum_{1 \leq j \leq n, j \neq i_1, i_2} [(d'_{i_1j})^{-p} - (d_{i_1j})^{-p}] + \sum_{1 \leq j \leq n, j \neq i_1, i_2} [(d'_{i_2j})^{-p} - (d_{i_2j})^{-p}] \right]^{1/p} \quad (8)$$

The flowchart displays the process of MSA is presented in Figure 5. As shown in equation (5) to (8), only some rows and columns are updated to calculate  $\phi_p$  criterion in MSA. Hence the complexities or BigO of MSA is much smaller than SA as presented in Table 2.

Table 2 The complexities to calculate  $\phi_p$  criterion in SA and MSA

BigO(SA)	$O(dn^2) + O(n^2 \log_2(p))$
BigO(MSA)	$O(n) + O(n \log_2(p))$

### 2.5 Data Quality

The performance of SA and MSA is investigated via  $\phi_p$  optimality criteria. The comparison is made on the dimension of the problems as specified in Table 3. For each  $d$ , number of runs ( $n$ ) is fixed at a maximum number of unknown parameters that are required in the second order polynomial model ( $n = 2d + 4 \binom{d}{2} + 1$ ).

Table 3 Dimension of the problem

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

For each dimension of the problem in Table 3, SA and MSA are simulated 10 times for a specified tolerance and the algorithmic parameters with respect to  $\phi_p$  optimality criterion [6]. The two algorithms are implemented via codes written in MATLAB to search for an optimal LHD with respect to  $\phi_p$  criteria. SA requires input parameter settings for  $t_0$ ,  $l_{max}$ ,  $C_l$  and  $tl$ . We use the heuristics presented in [6] and our own experience of using SA to set the initial values presented below.

- $t_0$  is set to average Euclidean intersite distance of the initial LHD  $X$ .

$$t_0 = \sum_{i=1}^n \sum_{j=i+1}^n \left[ \sum_{l=1}^d (X_{il} - X_{jl})^2 \right]^{1/2} \binom{d}{2}$$

- $l_{max}$  is the maximum number of exchanges before the temperature is reduced. The number should be large enough to allow SA to move away from local minimum.  $l_{max}$  is set to 1000 for  $d \leq 6$ , and  $l_{max}$  is set to 500 for  $d \geq 7$ .

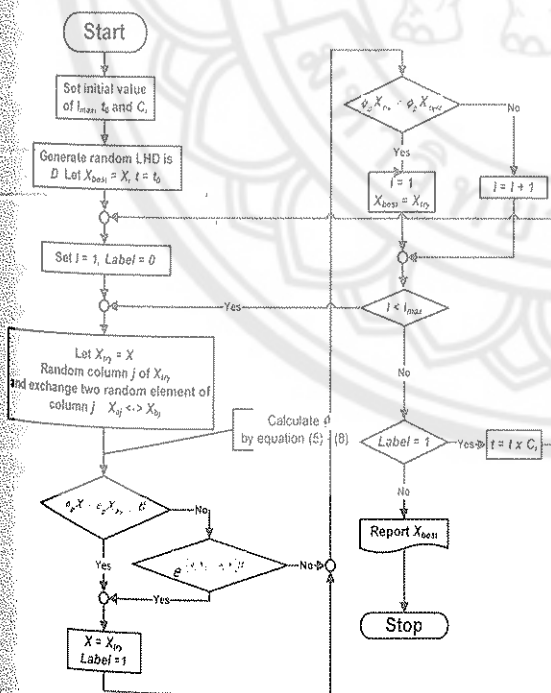


Figure 5 A flowchart of MSA



- $C_t$  is the rate at which the temperature is cooled down.  
 $C_t$  is set at 0.95, to lower the temperature slowly as the search progresses.
- The tolerance level  $\epsilon$  is set at 0.0001 as our empirical studies indicated that setting smaller tolerance levels would not result improvement in optimal value but require longer time to terminate.

### 3. Result

The values of  $\phi_p$  criteria at the termination step of SA and MSA from each dimension of problems stated in Table 3 are presented in Table 4. The descriptive statistics on the  $\phi_p$  values obtained from each search technique (SA and MSA) are displayed in columns 4-7. The results in columns 4-7 indicate that MSA performs similarly to SA in terms of minimization of  $\phi_p$  criteria. Further, the SD values appeared in column 7 displays a slightly larger amount of variation over 10 replications in SA than MSA. This indicates the consistency in the search process for MSA when different starting points are considered. The results in columns 8-9 display time elapsed for each search algorithm and the percentage of time reduction in MSA over SA. It can be concluded that from these percentages MSA converges much more quickly than SA. The maximum percentage of improvement over SA is approximately 97%. It should be noted that larger amount of improvement for MSA over SA occurs when the dimensions of problems are increased.

Table 4 Results on  $\phi_p$  and time elapsed for SA and MSA

Dimension of problem	Algorithm	$\phi_p$				Average Time (s)	% of time reduction
		MIN	MAX	MEAN	SD		
2	SA	4.274	4.274	4.274	0.000	19.006	23.34
	MSA	4.274	4.274	4.274	0.000	14.569	
3	SA	4.908	4.924	4.917	0.015	81.530	48.69
	MSA	4.910	4.925	4.919	0.012	41.833	
4	SA	5.206	5.209	5.207	0.004	240.326	67.8
	MSA	5.206	5.209	5.208	0.004	77.394	
5	SA	5.422	5.422	5.422	0.003	599.939	80.48
	MSA	5.422	5.423	5.422	0.003	117.051	
6	SA	5.591	5.592	5.591	0.004	1371.681	88.75
	MSA	5.590	5.592	5.591	0.004	154.319	
7	SA	5.746	5.762	5.754	0.005	1404.493	92.77
	MSA	5.753	5.754	5.753	0.005	101.489	
8	SA	5.886	5.906	5.898	0.006	2641.087	94.67
	MSA	5.886	5.898	5.898	0.004	140.868	
9	SA	6.029	6.046	6.036	0.006	4795.430	96.44
	MSA	6.029	6.046	6.036	0.005	170.715	
10	SA	6.180	6.182	6.172	0.007	8436.806	96.64
	MSA	6.157	6.180	6.170	0.007	283.886	

### 4. Conclusion

According to the results presented in Table 4, it can be concluded that the modified SA (MSA) performs much better than an original SA (SA) as the time required for MSA to reach the same level of optimal design is significantly smaller than that of SA. Hence MSA is recommended for the construction of the optimal LHD for CSE. Furthermore, other classes of design can be developed and collaborated with SA to search for the best design in the class. Other types of search algorithm like Particle swarm optimization (PSO) or Ant colony can be further developed in constructing an optimal LHD for CSE.

### References

- 1 R. Jin, W. Chen, and A. Sudjianto. "An efficient algorithm for constructing optimal design of computer experiments.", Journal of Statistical Planning and Inference, Vol. 134, 2006, pp. 268-287.
- 2 M. Johnson, L. Moore, and D. Ylvisaker. "Minimax and maximin distance designs.", Journal of Statistical Planning and Inference, Vol. 26, 1995, pp. 131-148.
- 3 J. Koehler, and A.B. Owen. "Computer experiment.", Handbook of Statistics, Vol. 13, 1996, pp. 261-308.
- 4 M. Liefvendahl, and R. Stocki. "Study on algorithms for optimization of Latin hypercubes.", Journal of Statistical Planning and Inference, Vol. 136, 2006, pp. 3231-3247.
- 5 M.D. McKay, R.J. Beckman, and W.J. Conover. "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code.", Technometrics, Vol. 21, 1979, pp 239-246
- 6 M.D. Morris, and T.J. Mitchell. "Exploratory design for computer experiments.", Journal of Statistical Planning and Inference, Vol. 43, 1995, pp. 381-402.
- 7 J. Rungrattanaubol and A. Na-udom. "Comparison of Evolutionary Search Algorithms in Computer Simulated Experiments." National Computer Science and Engineering Conference, Vol. 11, 2007, pp. 102-106





8 J. Sacks, S.B. Schiller, and W.J. "Designs for Computer Experiments." *Technometrics*, Vol. 31, No. 1, Feb 1989, pp. 41-47

9 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, No.3, 2001, pp. 209-240.

10 K.Q. Ye, W Li., and A Sudjianto. "Algorithmic construction of optimal symmetric Latin hypercube designs." *Journal of Statistical Planning and Inference*, Vol 90, 2000, pp. 145-159.



4, it can be  
ch better than  
to reach the  
r than that of  
ruction of the  
of design can  
for the best  
n like Particle  
er developed

algorithm for  
experiments.",  
l. 134, 2005,

Minimax and  
ical Planning

experiment",  
18.

gorithms for  
of Statistical  
1-3247.

Conover. "A  
ues of input  
puter code.",

design for  
lanning and

mparison of  
r Simulated  
d Engineer

