



การปรับปรุงวิธีการค้นแบบสุ่มหลายจุดสำหรับการหาค่าเหมาะสมต่ำสุด ของฟังก์ชันมัลติโมดอล

An adaptive Multi-Point Random Search (AMPRS) for Global Minimum Optimization Value of Multimodal Functions

ชวนพิศ ถันชนนาง

Chuanpit Tunchonnang

ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏรำไพพรรณี

Department of Mathematics, Faculty of Science and Technology, Rambhai Barni Rajabhat University

e-mail: chuanpit.t@rbu.ac.th

(Received: May 22, 2018 ; Revised : July 31, 2018 ; Accepted : August 6, 2018)

บทคัดย่อ

วิจัยนี้ศึกษาการปรับปรุงวิธีการค้นแบบสุ่มหลายจุดสำหรับค่าเหมาะสมต่ำสุดของฟังก์ชันมัลติโมดอล ซึ่งมีองค์ประกอบหลักคือ จำนวนประชากรที่ค้น ทิศทางของจุดที่ค้น ระยะกระโดด การดำเนินการเชิงประชากร และอัตราการแลกเปลี่ยนส่วนประกอบของเวกเตอร์ ซึ่งได้ทดลองกับฟังก์ชันทดสอบ 5 ฟังก์ชัน ที่มีจำนวนตัวแปร $N = 2, 5, 10, 20$ และ 30 อัตราการแลกเปลี่ยนส่วนประกอบของเวกเตอร์ $CR = 0.1, 0.3, 0.5, 0.7$ และ 0.9 และจำนวนประชากร $NP = 10, 30, 50, 70$ และ 90 จากการศึกษาพบว่า อัตราการแลกเปลี่ยนส่วนประกอบของเวกเตอร์ $CR = 0.9$ และจำนวนประชากร $NP = 50$ เป็นค่าที่เหมาะสมต่ำสุดของฟังก์ชันมัลติโมดอล โดยการปรับปรุงวิธีการค้นแบบสุ่มหลายจุด

คำสำคัญ : การหาค่าเหมาะสมต่ำสุด, ฟังก์ชันมัลติโมดอล, วิธีการค้นแบบสุ่มหลายจุด

Abstract

This research studied an adaptive Multi-Point Random Search (AMPRS) for global minimum value of multimodal functions consisted of main elements including population number, a search direction, step size, population operation and crossover rate of vectors. In which this method was performed by using 5 selected test functions with the following parameters: the variable number $N = 2, 5, 10, 20$ and 30 ; the crossover rate $CR = 0.1, 0.3, 0.5, 0.7$ and 0.9 , and the population number $NP = 10, 30, 50, 70$ and 90 . The study found that the crossover rate $CR = 0.9$ and the population number $NP = 50$ were the suitable global minimum optimization value of multimodal functions using an adaptive Multi-Point Random Search.

Keywords : Global Minimum Optimization, Multimodal Functions, Adaptive Multi-Point Random Search (AMPRS)

บทนำ

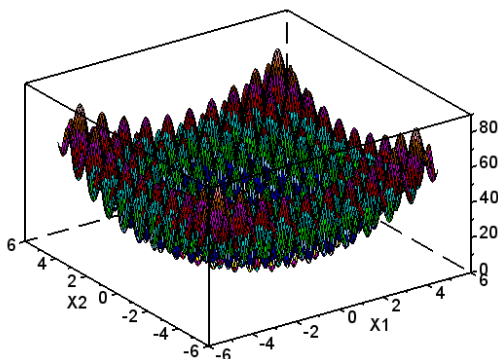
ปัญหาค่าเหมาะสมต่ำสุด คือ การหาค่าต่ำสุดของฟังก์ชัน ดังนั้นการแก้ปัญหาต่ำสุดเหมาะสมจึงขึ้นอยู่กับฟังก์ชันวัตถุประสงค์ที่กำหนดขอบเขตของการค้น เราสนใจศึกษาการปรับปรุงวิธีการค้นแบบสุ่มหลายจุดสำหรับการหาค่าเหมาะสมต่ำสุดของฟังก์ชันมัลติโมดอลซึ่งเป็นวิธีที่ไม่ใช้อนุพันธ์ในการหาคำตอบและเป็นวิธีการทำซ้ำโดยใช้จุดค้นหลายจุด (Multi-point random search : MPRS) ซึ่งมีองค์ประกอบหลักคือ จำนวนประชากรที่ค้น ทิศทางของจุดที่ค้น ระยะเวลาโคต การดำเนินการเชิงประชากร และอัตราการแลกเปลี่ยนส่วนประกอบของเวกเตอร์ สำหรับฟังก์ชันวัตถุประสงค์เป็นฟังก์ชันหลายตัวแปร วิธีการค้นแบบสุ่มที่ใช้จุดค้นเพียงจุดเดียวจะค้นได้ไม่ครอบคลุมเพราะมีจุดต่ำสุดหรือจุดสูงสุดเฉพาะที่หลายจุด ดังนั้นเราจึงใช้จุดค้นหลายจุดหรือการค้นเชิงประชากร (population based search) เพื่อให้สามารถค้นคำตอบได้ทั่วถึง และสามารถกำหนดการเพื่อแลกเปลี่ยนส่วนประกอบของจุดค้นต่างๆ

กำหนดให้ $f: D \subseteq R^N \rightarrow R$ เป็นฟังก์ชันวัตถุประสงค์ โดยที่ $x = (x_1, x_2, x_3, \dots, x_N)$ จะเรียก f ว่าเป็นฟังก์ชันหลายตัวแปร (Variable Functions) เมื่อ f มีจุดต่ำสุดเฉพาะที่หรือจุดสูงสุดเฉพาะที่หลายจุด เช่น

$$\text{Rastrigin Function: } f(x_1, x_2) = 2(10) + [(x_1^2 - 10 \cos(2\pi x_1)) + x_2^2 - 10 \cos(2\pi x_2)]$$

โดยที่ $x_1, x_2 \in [-5.12, 5.12]$ ซึ่งมีจุดต่ำสุดที่ $f(x_1, x_2) = (0, 0)$

Rastrigin Function



ภาพที่ 1 แสดง $f(x_1, x_2)$: Rastrigin Function

ในงานวิจัยนี้ศึกษาการหาค่าเหมาะสมต่ำสุดของฟังก์ชันมัลติโมดอล โดยการปรับปรุงวิธีการค้นแบบสุ่มหลายจุดที่มีการแลกเปลี่ยนส่วนประกอบของจุดที่ค้นด้วยอัตราการแลกเปลี่ยนที่ต่างกัน

Sheela ได้นำเสนอวิธีค้นแบบสุ่มที่มีระยะกระโดดที่เหมาะสม (Optimized step-size random search) ซึ่งใช้จุดค้นเพียงจุดเดียว ใช้ทิศทางแบบสุ่มและใช้ระยะกระโดดของการค้นจากการประมาณค่าด้วยพหุนามกำลังสอง และได้ทดสอบกับฟังก์ชันอย่างง่ายที่มีตัวแปร 2 ถึง 4 ตัว

Holland ได้นำเสนอขั้นตอนวิธีเชิงพันธุกรรม (Genetic algorithm: GA) ซึ่งใช้ในการถ่ายทอดประชากรรุ่นก่อนหน้าไปยังรุ่นถัดไป การดำเนินการประกอบด้วย การเลือกสายพันธุ์ (selection) การข้ามสายพันธุ์ (crossover) และการกลายพันธุ์ (mutation)

Storn และ Price ได้นำเสนอวิธีวิวัฒนาการเชิงผลต่าง (differential evolution: DE) สำหรับการหาค่าเหมาะสมที่สุดบน R^N ซึ่งมีการดำเนินการเชิงผลต่างและการแลกเปลี่ยนส่วนประกอบของเวกเตอร์

Mohamed, Sabry และ Motaz Khorshid ได้นำเสนอวิธีวิวัฒนาการเชิงผลต่าง (differential evolution: DE) สำหรับการหาค่าเหมาะสมที่สุดโดยใช้ทิศทาง และกฎการกลายพันธุ์ (mutation rule) เพื่อเพิ่มอัตราการเข้าสู่คำตอบที่ดีที่สุด

Zelda B. Zabinsky ได้นำเสนอวิธีการค้นแบบสุ่ม (Random search Algorithms) สำหรับหาค่าเหมาะสมซึ่งมีการค้นเพียงจุดเดียวและค้นหลายจุดเพื่อหาค่าเหมาะสมโดยใช้ทิศทาง และหลักความน่าจะเป็นของ Boltzmann distribution Francisco J. Solis and Roger J-B. Wets ได้นำเสนอเทคนิคการหาค่าต่ำสุดโดยใช้การค้นแบบสุ่ม (Minimization by Random Search Techniques) ซึ่งใช้ทิศทางและสร้างจุดโดยการสุ่ม จากนั้นเปรียบเทียบจุดถ้าจุดใดให้ค่าต่ำสุดจุดนั้นถือเป็นคำตอบที่ดีที่สุด ณ ขณะนั้น

วัตถุประสงค์ของการวิจัย

โดยวัตถุประสงค์ของการวิจัยเพื่อศึกษาและปรับปรุงวิธีการค้นแบบสุ่มหลายจุดสำหรับการหาค่าเหมาะสมต่ำสุดของฟังก์ชันมัลติโมดอล หาอัตราการแลกเปลี่ยนส่วนประกอบของจุดค้นที่เหมาะสม หาจำนวนประชากรที่เหมาะสม และเพื่อทดสอบประสิทธิภาพของการปรับปรุงวิธีการค้นแบบสุ่มหลายจุดสำหรับ



การหาค่าเหมาะสมต่ำสุดของฟังก์ชันมัลติโมดอล

วิธีดำเนินการวิจัย

งานวิจัยนี้จะนำแนวคิดของวิธีการค้นแบบสุ่มหลายจุด ขั้นตอนวิธีเชิงพันธุกรรม และขั้นตอนวิธีเชิงวิวัฒนาการเชิงผลต่างมาใช้เพื่อปรับปรุงวิธีการค้นแบบสุ่มหลายจุดสำหรับการหาค่าเหมาะสมต่ำสุดของฟังก์ชันมัลติโมดอล ดังนั้นจะกล่าวถึงขั้นตอนดังกล่าวโดยละเอียดดังนี้

วิธีการค้นแบบสุ่มหลายจุด (Multi-point Random Search)

วิธีการค้นแบบสุ่มเป็นวิธีการสุ่มที่หาค่าเหมาะสมที่สุดที่ไม่ใช้อนุพันธ์ในการหาคำตอบ ซึ่งมีขั้นตอนดังนี้

ขั้นตอนที่ 1 : กำหนดฟังก์ชันวัตถุประสงค์ $f(X) = f(x_1, x_2, x_3, \dots, x_N)$ เมื่อ $x = (x_1, x_2, x_3, \dots, x_N)$ กำหนดจำนวนการเรียกใช้ฟังก์ชันสูงสุด (NFMAX)

$$\begin{bmatrix} \text{ขั้นตอนที่ 2: สร้างประชากรเริ่มต้นแบบสุ่ม} \\ x_{11} & x_{12} & x_{13} & \dots & x_{1N} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2N} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{NP1} & x_{NP2} & x_{NP3} & \dots & x_{NPN} \end{bmatrix}$$

กำหนดให้ x^{New} เมื่อ $x^{New} = (x_1^{New}, x_2^{New}, x_3^{New}, \dots, x_N^{New})$ โดยที่ $x^{New} \in [L, U]$ กำหนดให้ $xb = x^{New}$ เป็นจุดที่ให้ค่าต่ำสุดและให้ $fb = f(x^{New})$ เป็นค่าฟังก์ชันต่ำสุด ณ ขณะนั้น

ขั้นตอนที่ 3 : สุ่มเวกเตอร์ทิศทางขนาดหนึ่งหน่วย V และสร้างเวกเตอร์ W โดย $W = x^{New} + SV$ เมื่อ S คือระยะกระโดด (Step size)

ขั้นตอนที่ 4 : เปรียบเทียบค่า $f(W)$ กับ $f(x^{New})$ และ $f(W)$ กับ $f(xb)$

ถ้า $f(W) < f(x^{New})$ ให้ $x^{New} = W$, $f(x^{New}) = f(W)$

ถ้า $f(W) < f(xb)$ ให้ $xb = W$, $f(xb) = f(W)$

ถ้าไม่พบจุดที่ดีขึ้นในแต่ละรุ่นของประชากรให้ลดระยะกระโดด

ขั้นตอนที่ 5 : ถ้าจำนวนการเรียกใช้ฟังก์ชันน้อยกว่าหรือเท่ากับ NFMAX ให้ทำซ้ำขั้นตอนที่ 3-4 แต่ถ้ามากกว่า NFMAX ให้หยุดดำเนินการจะได้จุดที่ให้ค่าต่ำสุด xb ซึ่งให้ค่าต่ำสุด fb

วิธีวิวัฒนาการเชิงผลต่าง (Differential Evolution)

เป็นวิธีที่ใช้ในการหาค่าที่เหมาะสมที่สุดซึ่งพัฒนาจาก

GA เมื่อโดเมนของการค้นเป็น R^N โดยมีขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 : สร้างประชากรของคำตอบเริ่มต้น (initialization population) ที่มีมิติเท่ากับ จำนวนตัวแปร N และกำหนดจำนวนประชากรเท่ากับ NP จำนวนรุ่นประชากรเท่ากับ NG

ขั้นตอนที่ 2 : สร้างประชากรในรุ่นถัดไป

ขั้นตอนที่ 2.1 ขั้นตอนการกลายพันธุ์ (mutation)

สำหรับแต่ละเวกเตอร์เป้าหมาย (target vector) $x_{i,g}$ ในแต่ละรุ่นของประชากรที่ g เมื่อ $i = 1, 2, 3, \dots, NP$ สร้างเวกเตอร์กลายพันธุ์ (mutant vector) $V_{i,g}$ ดังสมการ

$$V_{i,g} = x_{r1,g} + F(x_{r2,g} - x_{r3,g})$$

เมื่อ x_{r1}, x_{r2} และ x_{r3} คือ เวกเตอร์ที่ต่างกันที่ได้จากการสุ่มจากประชากรที่ 1 ถึง NP ซึ่งต่างจากเวกเตอร์เป้าหมาย และ F คือ scaling factor ที่มีค่าอยู่ในช่วง $[0, 1]$

ขั้นตอนที่ 2.2 ขั้นตอนการสลับสายพันธุ์ (crossover)

สำหรับแต่ละ $i = 1, 2, 3, \dots, NP$ สร้างเวกเตอร์ $U_{i,g}$ ดังสมการ

$$U_{i,g,j} = \begin{cases} V_{i,g,j} : & \text{rand}() \leq CR \text{ or } j = j_{rand} \\ x_{i,g,j} : & \text{Otherwise} \end{cases}$$

เมื่อ CR คือ อัตราการสลับสายพันธุ์ที่มีค่าอยู่ในช่วง $[0,1]$, $j = 1, 2, 3, \dots, NP$ และ j_{rand} คือ เลขจำนวนเต็มสุ่มของเวกเตอร์ที่มีค่าตั้งแต่ 1 ถึง N

ขั้นตอนที่ 2.3 ขั้นตอนการเลือกสายพันธุ์ (selection)

เป็นขั้นตอนในการเลือกประชากรในรุ่นถัดไปโดยพิจารณาจากค่าของฟังก์ชันที่จุด $x_{i,g}$ กับ $U_{i,g}$ ดังสมการ

$$x_{i,g+1} = \begin{cases} U_{i,g} : & f(U_{i,g}) < f(x_{i,g}) \\ x_{i,g} : & f(U_{i,g}) \geq f(x_{i,g}) \end{cases}$$

เมื่อ คือ อัตราการสลับสายพันธุ์ที่มีค่าอยู่ในช่วง $[0,1]$, $j = 1, 2, 3, \dots, NP$ และ j_{rand} คือ เลขจำนวนเต็มสุ่มของเวกเตอร์ที่มีค่าตั้งแต่ 1 ถึง N

ขั้นตอนที่ 3 : กลับไปทำขั้นตอนที่ 2.2 จนครบจำนวน

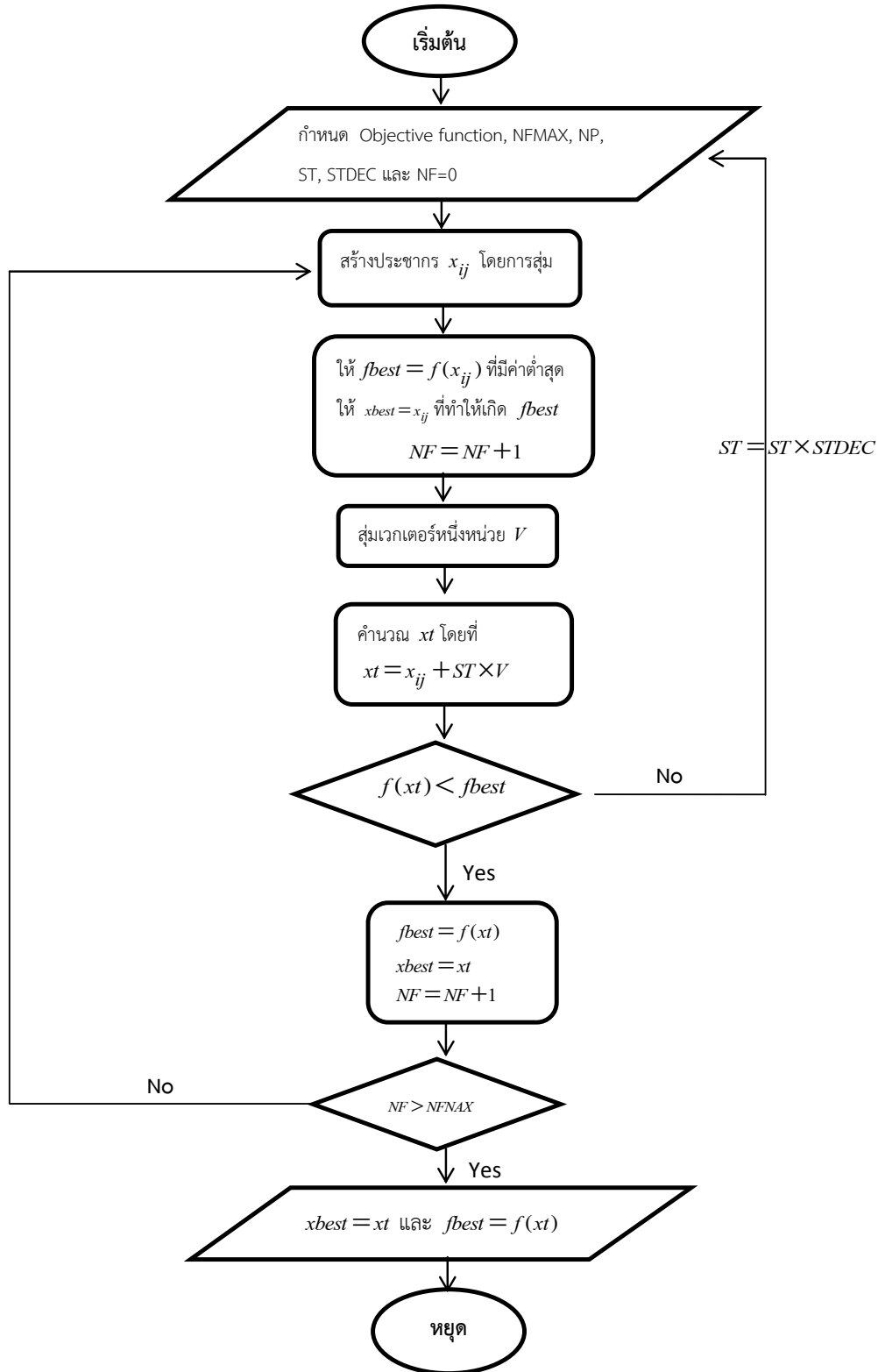


รุ่นประชากร NG

ซึ่งในงานวิจัยนี้จะเสนอการปรับปรุงวิธีการค้นแบบสุ่ม
หลายจุดสำหรับการหาค่าเหมาะสมต่ำสุดของฟังก์ชันเหมาะสม

ต่ำสุด ซึ่งเป็นอีกวิธีหนึ่งที่ใช้แก้ปัญหของฟังก์ชันที่มีจุดต่ำสุด
เฉพาะที่หรือจุดสูงสุดเฉพาะที่หลายจุดที่สามารถค้นค่าตอบได้
ครอบคลุมซึ่งมีขั้นตอนอย่างละเอียด และมีแผนผังอัลกอริทึม

แผนผังอัลกอริทึม การปรับปรุงวิธีการค้นแบบสุ่มหลายจุด



ภาพที่ 2 แสดงแผนผังอัลกอริทึม การปรับปรุงวิธีการค้นแบบสุ่มหลายจุด



(Flowchart Algorithm) ประกอบความเข้าใจมากยิ่งขึ้นดังนี้

3.1 ศึกษาและปรับปรุงวิธีการค้นแบบสุ่มหลายจุดสำหรับการหาค่าเหมาะสมที่สุดของฟังก์ชันมัลติโมดอล มีขั้นตอนดังนี้

ขั้นตอนที่ 1 : กำหนดฟังก์ชันวัตถุประสงค์ (Objective function) $f(x) = f(x_1, x_2, x_3, \dots, x_N)$ เมื่อ $x = (x_1, x_2, x_3, \dots, x_N)$ กำหนดจำนวนการเรียกใช้ฟังก์ชันสูงสุด (NFMAX) จำนวนประชากร (NP) ระยะกระโดด (ST) และค่าตัวคูณระยะกระโดด (STDEC)

$$\begin{bmatrix} \text{ขั้นตอนที่ 2 : สร้างประชากรเริ่มต้นแบบสุ่ม} \\ x_{11} & x_{12} & x_{13} & \dots & x_{1N} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2N} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{NP1} & x_{NP2} & x_{NP3} & \dots & x_{NPN} \end{bmatrix}$$

เมื่อ $x_{ij} \in [L, U]$ ให้ x_{best} เป็นจุดที่ให้ค่าต่ำสุดและ f_{best} เป็นค่าฟังก์ชันต่ำสุด

ขั้นตอนที่ 3 : ให้ x_i เป็นประชากรตัวที่ i เมื่อ $i = 1, 2, 3, \dots, NP$ สุ่มเวกเตอร์ทิศทางขนาดหนึ่งหน่วย V และสร้างเวกเตอร์ xt จาก x_i โดยที่ $ST \times V$

ขั้นตอนที่ 4 : สุ่มเวกเตอร์ x_j เมื่อ $j \neq i$ ใช้การแลกเปลี่ยนประกอบของเวกเตอร์ด้วยอัตราการแลกเปลี่ยน CR ระหว่างเวกเตอร์ st กับ x_j โดยที่

$$xt_k = \begin{cases} x_{j,k} : & rand() \leq C \text{ or } k = I_i \\ xt_k : & \text{Otherwise} \end{cases}$$

เมื่อ I_i คือ จำนวนเต็มสุ่มระหว่าง 1 ถึง N , $rand() \in [0, 1]$, $k = 1, 2, 3, \dots, N$ และ CR คือ อัตราการแลกเปลี่ยนส่วนประกอบของเวกเตอร์ โดยที่ $CR \in [0, 1]$

ขั้นตอนที่ 5 : เปรียบเทียบค่า $f(xt)$ กับ $f(x_i)$

$$\text{ถ้า } f(xt) < f(x_i) \text{ ให้ } x_i = xt, f(x_i) = f(xt)$$

$$\text{ถ้า } f(xt) < f(x_{best}) \text{ ให้ } x_{best} = xt, f(x_{best}) = f(xt)$$

ขั้นตอนที่ 6 : ถ้าไม่พบจุดที่ดีขึ้นในแต่ละรุ่นประชากรให้ $ST = ST \times STDEC$

ขั้นตอนที่ 7 : ถ้าจำนวนการเรียกใช้ฟังก์ชันน้อยกว่าหรือเท่ากับ NFMAX ให้ทำซ้ำขั้นตอนที่ 3 - 6 แต่ถ้ามากกว่า NFMAX ให้หยุดดำเนินการจะได้จุดที่ให้ค่าต่ำสุด x_{best} ซึ่งให้ค่าฟังก์ชัน

ต่ำสุด f_{best}

3.2 หาอัตราการแลกเปลี่ยนส่วนประกอบของจุดที่ค้นเหมาะสม

กำหนดให้ $CR = 0.1, 0.3, 0.5, 0.7, 0.9$ ใช้ในการแลกเปลี่ยนประกอบของจุดเวกเตอร์ด้วยอัตราการแลกเปลี่ยน CR ระหว่างเวกเตอร์ xt กับ x_j เมื่อ $j \neq i$ โดยที่

$$xt_k = \begin{cases} x_{j,k} : & rand() \leq CR \text{ or } k = I_i \\ xt_k : & \text{Otherwise} \end{cases}$$

เมื่อ I_i คือ จำนวนเต็มสุ่มระหว่าง 1 ถึง N , $rand() \in [0, 1]$, $k = 1, 2, 3, \dots, N$ และ CR คือ อัตราการแลกเปลี่ยนส่วนประกอบของเวกเตอร์ โดยที่ $CR \in [0, 1]$

1. ใช้ฟังก์ชันทดสอบ 5 ฟังก์ชัน โดยทำการทดลอง 10 ซ้ำ และ N คือ มิติของโดเมน เมื่อ $N = 2, 5, 10, 20$ และ 30 กำหนดให้ $NFMAX = 1000 \times N$

2. กำหนดขอบเขตล่างและขอบเขตบนของแต่ละตัวแปร โดยที่ $L \leq x_i \leq U$ และ $ST = 0.9 \times (U - L)$, $STDEC = 0.9$

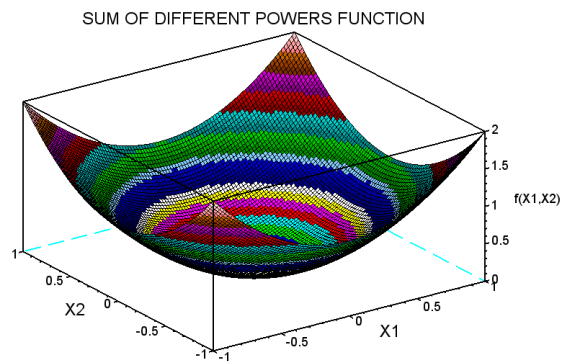
3. กำหนดอัตราการแลกเปลี่ยนส่วนประกอบของจุดเวกเตอร์ $CR = 0.1, 0.3, 0.5, 0.7, 0.9$ และจำนวนประชากร $NP = 10, 30, 50, 70, 90$

3.3 ฟังก์ชันทดสอบ

$$f_1 = \sum_{i=1}^{|i+1|} |x_i| \quad x_i \in [-1, 1], \quad \text{โดยที่}$$

$i = 1, 2, 3, \dots, N$

จุดที่ให้ค่าต่ำสุด คือ $x^* = (0, 0, 0, \dots, 0)$ ซึ่งให้ค่าฟังก์ชันต่ำสุด



ภาพที่ 3

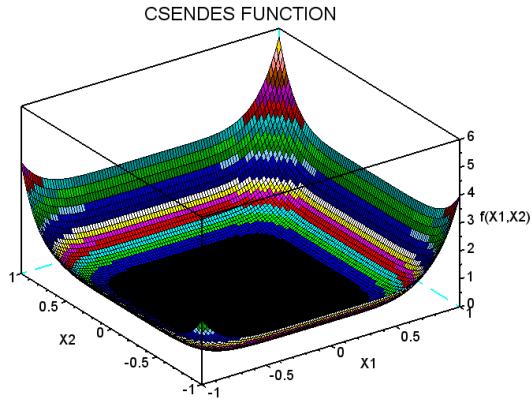
แสดง $f_1(x_1, x_2)$: Sum of Different Powers Function



$$f_2 = \sum_{i=1}^N x_i^6 \left[2 + \sin\left(\frac{1}{x_i}\right) \right]$$

$$x_i \in [-1, 1], i = 1, 2, 3, \dots, N$$

จุดที่ให้ค่าต่ำสุด คือ $x^* = (0, 0, 0, \dots, 0)$ ซึ่งให้ค่าฟังก์ชันต่ำสุด $f_1(x^*) = 0$

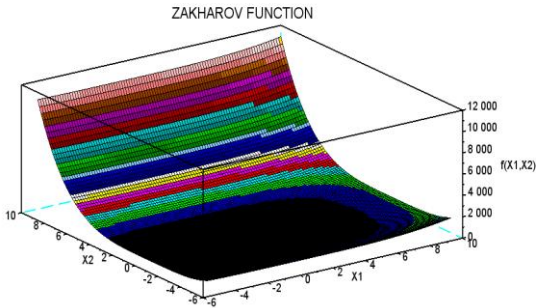


ภาพที่ 4 แสดง $f_2(x_1, x_2)$: Csendes Function

$$f_3 = \sum_{i=1}^N x_i^2 + \left(\sum_{i=1}^N 0.5ix_i \right)^2 + \left(\sum_{i=1}^N 0.5ix_i \right)^4$$

$$x_i \in [-5, 10], i = 1, 2, 3, \dots, N$$

โดยที่ จุดที่ให้ค่าต่ำสุด คือ $x^* = (0, 0, 0, \dots, 0)$ ซึ่งให้ค่าฟังก์ชันต่ำสุด $f_1(x^*) = 0$

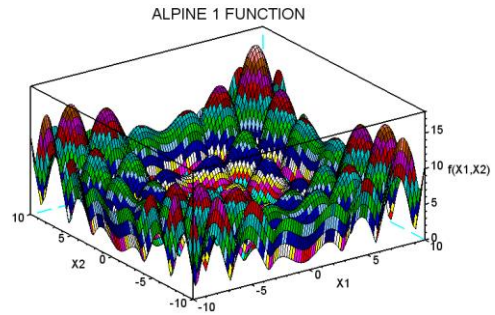


ภาพที่ 5 แสดง $f_3(x_1, x_2)$: Zakharov Function

$$f_4 = \sum_{i=1}^N |x_i \sin(x_i) + 0.1x_i|$$

$$x_i \in [-10, 10], i = 1, 2, 3, \dots, N$$

จุดที่ให้ค่าต่ำสุด คือ $x^* = (0, 0, 0, \dots, 0)$ ซึ่งให้ค่าฟังก์ชันต่ำสุด $f_1(x^*) = 0$

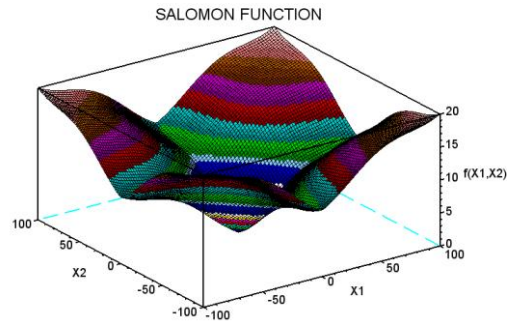


ภาพที่ 6 แสดง $f_4(x_1, x_2)$: Alpine 1 Function

$$f_5 = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^N x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^N x_i^2}$$

$$x_i \in [-100, 100], i = 1, 2, 3, \dots, N$$

โดยที่ จุดที่ให้ค่าต่ำสุด คือ $x^* = (0, 0, 0, \dots, 0)$ ซึ่งให้ค่าฟังก์ชันต่ำสุด $f_1(x^*) = 0$



ภาพที่ 7 แสดง $f_5(x_1, x_2)$: Salomon Function

3.4 หาค่า CR ที่เหมาะสม

กำหนดให้ $N = 30$, $NP = 10, 30, 50, 70, 90$ และ $CR = 0.1, 0.3, 0.5, 0.7, 0.9$ จะได้ว่า NP ที่เหมาะสม คือ $NP = 50$ ซึ่งแสดงผลค่า Mean และ SD ของ f_{best} สำหรับ ฟังก์ชันทดสอบ เมื่อ $NP = 50$ ดังต่อไปนี้

ตารางที่ 1 แสดงค่า Mean และ SD ของ f_{best} สำหรับฟังก์ชันทดสอบ $f_1 - f_5$ เมื่อ $CR = 0.1, 0.3, 0.5$

$f(x)$	CR					
	0.1		0.3		0.5	
	Mean	SD	Mean	SD	Mean	SD
f_1	1.994D-14	2.258D-14	1.350D-14	1.942D-14	0	0
f_2	0	0	0	0	0	0
f_3	0.000018	0.0000387	0.0000097	0.0000089	0.0000051	0.0000074
f_4	0.00004	0.0001041	0.0000206	0.0000351	0.0000159	0.0000021
f_5	1.6594333	0.2713378	1.0688684	0.2665839	1.9183168	0.5047651

ตารางที่ 2 แสดงค่า Mean และ SD ของ f_{best} สำหรับฟังก์ชันทดสอบ $f_1 - f_5$ เมื่อ $CR = 0.7, 0.9$

$f(x)$	CR			
	0.7		0.9	
	Mean	SD	Mean	SD
f_1	0	0	0	0
f_2	0	0	0	0
f_3	0.0000059	0.0000038	0.0000009	0.0000007
f_4	0.0000116	0.0000019	0.0000102	0.0000016
f_5	3.6276857	0.9484118	0.369313	0.2494319

จากตารางที่ 1-2 สำหรับ $N = 30$, และ $NP = 50$ พบว่าอัตราการแลกเปลี่ยนส่วนประกอบของเวกเตอร์ที่เหมาะสมที่สุดสำหรับฟังก์ชันทดสอบทั้ง 5 ฟังก์ชัน f_1, f_2, f_3, f_4 และ f_5 คือ $CR = 0.9$

3.5 ทดสอบประสิทธิภาพการใช้งานกับมิติต่างๆ

กำหนด $N = 2, 5, 10, 20, 30$ ซึ่งจากการทดลองข้างต้นพบว่า $NP = 50$ และ $CR = 0.9$ เป็นค่าที่เหมาะสมที่สุด จึงเลือกมาทดสอบประสิทธิภาพการใช้งานมิติต่างๆ กับฟังก์ชันทดสอบ 5 ฟังก์ชันโดยการปรับปรุงวิธีการค้นแบบสุ่มหลายจุด ได้ผลการทดลองดังตารางที่ 3-4

ตารางที่ 3 แสดงค่า Mean และ SD ของ f_{best} สำหรับฟังก์ชันทดสอบ $f_1 - f_3$ เมื่อ $N = 2, 5, 10, 20, 30$

N	$f(x)$					
	f_1		f_2		f_3	
	Mean	SD	Mean	SD	Mean	SD
2	2.921D-14	4.514D-14	0	0	0	0
5	1.241D-10	1.865D-10	0	0	0	0
10	3.219D-12	5.761D-12	0	0	0	0
20	3.925D-13	1.200D-12	0	0	0	0
30	0	0	0	0	0	0



ตารางที่ 4 แสดงค่า Mean และ SD ของ f_{best} สำหรับฟังก์ชันทดสอบ $f_4 - f_5$ เมื่อ $N = 2, 5, 10, 20, 30$

	$f(x)$			
	f_4		f_5	
N	Mean	SD	Mean	SD
2	0.0005027	0.0003321	0.0404116	0.0560341
5	0.0000435	0.0001884	0.0005632	0.0003625
10	0.0000429	0.0001041	0.0003733	0.0001441
20	0.0000226	0.0000247	0.0018389	0.0007202
30	0.0000102	0.0000016	0.369313	0.2494319

จากตารางที่ 3 - 4 ได้นำค่า C ที่เหมาะสมที่สุดสำหรับฟังก์ชันทดสอบกับมิติต่างๆ พบว่าสามารถลู่เข้าสู่ค่าตอบได้ทั้งหมด 5 ฟังก์ชัน

ผลการวิจัย

ผลการวิจัยนี้เราได้นำวิธีการปรับปรุงวิธีการค้นแบบสุ่มหลายจุด (An adaptive Multi-Point Random Search : AMPRS) ซึ่งมีองค์ประกอบหลักคือ จำนวนประชากรที่ค้น ทิศทางของจุด

ที่ค้น ระยะกระโดด การดำเนินการเชิงประชากร และอัตราการแลกเปลี่ยนส่วนประกอบของเวกเตอร์ซึ่งนำผลการทดลองการลู่เข้าสู่คำตอบของแต่ละฟังก์ชัน และจำนวนตัวแปร $N = 2, 5, 10, 20, 30$ มาเปรียบเทียบกับวิธีการค้นแบบสุ่ม (Random Search method : RS) ซึ่งมีองค์ประกอบหลักคือ จำนวนประชากรที่ค้น ทิศทางของจุดที่ค้น ระยะกระโดด สำหรับฟังก์ชันทดสอบทั้ง 5 ฟังก์ชัน คือ f_1 : Sum of Different Powers Function, f_2 : Csendes Function, f_3 : Zakharov Function, f_4 : Alpine 1 Function และ f_5 : Salomon Function ซึ่งได้ผลดังตารางที่ 5

ตารางที่ 5 แสดงค่า Mean และ SD ของวิธี RS และ AMPRS สำหรับฟังก์ชันทดสอบ $f_1 - f_5$ เมื่อ $N = 2, 5, 10, 20, 30$

การเปรียบเทียบวิธี Random Search Method (RS) และ An adaptive Multi-Point Random Search (AMPRS) สำหรับฟังก์ชันทดสอบ $f_1 - f_5$					
		Random Search (RS)		An adaptive Multi-Point Random Search (AMPRS)	
$f(x)$	N	Mean	SD	Mean	SD
f_1	2	0.0000007	0.0000017	2.921D-14	4.514D-14
	5	1.111D-08	9.419D-09	1.241D-10	1.865D-10
	10	0.0000002	6.875D-08	3.219D-12	5.761D-12
	20	0.0000005	9.186D-08	3.925D-13	1.200D-12
	30	0.0000006	8.886D-08	0	0
f_2	2	0	0	0	0
	5	0	0	0	0
	10	0	0	0	0
	20	0	0	0	0
	30	4.710D-09	1.131D-08	0	0
f_3	2	0.0000004	0.0000321	0	0
	5	0.0000005	0.0000007	0	0
	10	0.0000004	0.0000006	0	0
	20	0.0000097	0.0000231	0	0
	30	0.0000009	0.0000007	0	0
f_4	2	0.0006636	0.0010803	0.0005027	0.0003321
	5	0.2888063	0.6380882	0.0000435	0.0001884
	10	0.6649655	0.6684844	0.0000429	0.0001041
	20	4.1845295	2.405158	0.0000226	0.0000247
	30	9.1358615	3.7098488	0.0000102	0.0000016
f_5	2	0.0999114	0.0942361	0.0404116	0.0560341
	5	3.7694301	2.4508145	0.0005632	0.0003625
	10	22.648772	6.1932201	0.0003733	0.0001441
	20	77.717505	14.360564	0.0018389	0.0007202
	30	128.39623	15.893479	0.369313	0.2494319



สรุปและอภิปรายผลการวิจัย

งานวิจัยนี้ได้นำเสนอการหาค่าเหมาะสมต่ำสุดของฟังก์ชันมัลติโมดอลโดยการปรับปรุงวิธีการค้นแบบสุ่มหลายจุด (An adaptive Multi-Point Random Search (AMPRS)) ซึ่งอาศัยแนวคิดการกำหนดทิศทางของการค้นจากวิธีการค้นแบบสุ่มหลายจุดและแนวคิดจากขั้นตอนการสลับสายพันธุ์ของขั้นตอนวิวัฒนาการเชิงผลต่าง ซึ่งได้ทดสอบกับฟังก์ชันทดสอบ 5 ฟังก์ชัน f_1, f_2, f_3, f_4 และ f_5 ซึ่งมีจำนวนตัวแปร $N = 2, 5, 10, 20, 30$ จำนวนประชากร $NP = 10, 30, 50, 70, 90$ และอัตราการแลกเปลี่ยนส่วนประกอบของเวกเตอร์ $CR = 0.1, 0.3, 0.5, 0.7, 0.9$ จากตารางที่ 5 แสดงค่า Mean และ SD ของวิธี RS และ AMPRS สำหรับฟังก์ชันทดสอบ $f_1 - f_5$ เมื่อ $N = 2, 5, 10, 20, 30$ ตามลำดับ โดยการปรับปรุงวิธีการค้นแบบสุ่มหลายจุดสำหรับ $N = 2, 5, 10, 20$ และ 30 พบว่ามีการเข้าสู่ค่าตอบทั้ง 5 ฟังก์ชันและเมื่อพิจารณาความแม่นยำของคำตอบจะเรียงตามความยากของฟังก์ชันจาก ฟังก์ชัน f_5, f_4, f_3, f_1 และ f_2

จากผลการวิจัยพบว่าการปรับปรุงวิธีการค้นแบบสุ่มหลายจุด มีประสิทธิภาพในการค้นสำหรับฟังก์ชันมัลติโมดอลที่ใช้ทดสอบทั้ง 5 ฟังก์ชันซึ่งได้อัตราการแลกเปลี่ยนส่วนประกอบของจุดเวกเตอร์ (CR) คือ $CR = 0.9$ และจำนวนจุดที่ค้น (NP) คือ $NP = 50$ พบว่าสามารถเข้าสู่ค่าตอบทั้ง 5 ฟังก์ชัน ซึ่งสะท้อนคุณสมบัติของฟังก์ชันมัลติโมดอลที่มีจุดต่ำสุดเฉพาะที่หรือจุดสูงสุดเฉพาะที่หลายจุด ซึ่งจุดเหล่านั้นอาจจะอยู่ใกล้หรือไกลกันดังนั้นจึงมีอัตราการและเปลี่ยนแปลงส่วนประกอบของเวกเตอร์ในอัตราที่สูง ในขณะที่วิธีการค้นแบบสุ่ม (Random Search method : RS) การเข้าสู่ค่าตอบจะไม่สามารถเข้าสู่ค่าตอบได้เมื่อมีจำนวนตัวแปร (N) ที่สูงขึ้นและความยากของฟังก์ชันที่มีจุดต่ำสุดหรือจุดสูงสุดเฉพาะที่หลายจุด

ข้อเสนอแนะ

ในการศึกษาครั้งต่อไปเราสามารถปรับปรุงวิธีการค้นแบบสุ่มหลายจุดสำหรับหาค่าเหมาะสมต่ำสุดของฟังก์ชันมัลติโมดอลโดยทำการปรับปรุงวิธีการสร้างเวกเตอร์ทิศทางของการค้น การปรับระยะกระโดด และอัตราการแลกเปลี่ยนส่วนประกอบของจุดเวกเตอร์ให้เหมาะสมมากขึ้น ซึ่งน่าจะทำให้การค้นมีประสิทธิภาพมากยิ่งขึ้น

เอกสารอ้างอิง

- Sheela, B.V. 1979. An Optimized step-size random Search (OSSRS). **Computer Method in Applied Mechanics and Engineering**. 19: 99-106.
- Holland, J.H. 1975. Adaptation in Natural and Artificial Systems, **University of Michigan Press, Ann Arbor, Michigan; re-issued by MIT Press**.
- Storn, R.N. and Price, K.N. 1997. Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. **Journal of Global Optimization**. 11: 341-359.
- Mohamed, Ali W., Hegazy Z. Sabry and Motaz Khorshid. 2012. An alternative differential evolution algorithm for global optimization. 3: 149-165.
- Zabinsky, Zelda B. 2009. Random Search Algorithms, **Department of Industrial and Engineering, University of Washington**. Seattle. April 5.
- Price K., Storn R., and Lampinen J. 2005. Differential Evolution - A Practical Approach to Global Optimization, **Springer. Berlin Heidelberg New York**.
- Sotiropoulos, D.G., Plagianakos, V.P. and Vrahatis, M.N. An Evolutionary Algorithm for Minimizing Multimodal Function, **Department of Mathematics, Uiversity of Patras, GR-261.10 Patras, Greece. University of Patras Artificial Inteligence Research Center-UPAIRC**.
- Zelda B. Zabinsky. 2009. Random Search Algorithms, **Department of Industrial and Systems Engineering, University of Washington, Seattle, Wa, 98195-2650, USA**.
- Francisco, J.Solis and Roger, J-B. Wets. 1981. Minimization By Random Search Techniques, **Mathematics of Operations Research**. 6 : 1.