วิธีค้นหาค่าเหมาะที่สุดด้วยการใช้กรรมวิธีค้นหาที่เหมาะสมในไฮเปอร์คิวบ์ที่ปรับได้

Optimization Methods Through Selecting Suitable Search Techniques in Adaptive Hypercubes

รังสิมันต์ สิทธิกร
Rungsimant Sitdhikorn
สถาบันนวัตกรรมมหานคร มหาวิทยาลัยเทคโนโลยีมหานคร
Mahanakorn Institute of Innovation, Mahanakorn University of Technology, Thailand

***Corresponding Author E-mail**: Rungsimant@mut.ac.th

Received: 28/08/24, **Revised**: 24/09/24, **Accepted**: 26/09/24

บทคัดย่อ

บทความนี้เสนอ 2 วิธีค้นหาค่าเหมาะที่สุดวัตถุประสงค์เดียวที่ไม่มีเงื่อนไขบังคับ โดยการสร้างบริเวณค้นหาที่เรียกว่าไฮเปอร์คิวบ์ ซึ่งตำแหน่งและขอบเขตแปรตามค่าฟังก์ชันวัตถุประสงค์การค้นหาในไฮเปอร์คิวบ์จะเลือกใช้ขั้นตอนวิธีค้นหาเชิงสำรวจหรือขั้นตอนวิธีค้นหาแบบกลุ่มอนุภาคที่ค่าสัมประสิทธิ์ปรับได้ (Adaptive Coefficient Particle Swarm Optimizer: ACPSO) โดยพิจารณาจากผลการค้นหาที่ผ่านมาดีขึ้นหรือไม่ สมรรถนะของวิธีค้นหาจะถูกทดสอบด้วยฟังก์ชันเทียบเคียงจำนวน 7 ฟังก์ชันที่จำนวน 10, 15 และ 20 มิติ วิธีค้นหาที่เสนอทั้งสองสามารถหาคำตอบได้ถึง 5 ฟังก์ชัน โดยสามารถค้นหาค่าที่ใกล้คำตอบของฟังก์ชัน Rosenbrock ซึ่งวิธีค้นหาส่วนใหญ่ติดกับดักการค้นหา วิธีค้นหาที่เสนอยังคงสมรรถนะดีเช่นเดิมแม้จำนวนมิติของฟังก์ชันมากขึ้น

คำสำคัญ: การหาค่าเหมาะที่สุด ไฮเปอร์คิวบ์ วิธีค้นหาค่าเหมาะที่สุดเชิงกลุ่มอนุภาค

Abstract

This paper proposes two methods for unconstrained single objective optimization by creating a search space called a hypercube, where the position and boundaries vary according to the objective function values. The search within the hypercube utilizes either an exploratory search algorithm or an Adaptive Coefficient Particle Swarm Optimizer (ACPSO) based on the improvement of previous search results. The performance of these search methods is tested using seven benchmark functions with dimensions of 10, 15, and 20. The proposed methods successfully find solutions for up to five functions. They can obtain search values close to the solution of the Rosenbrock function, whereas most search methods tend to get trapped in the search process. The proposed search methods maintain their good performance even as the number of dimensions of the functions increases.

Keywords: Optimization, Hypercube, PSO

1. บทนำ

การหาคำตอบของปัญหาทางวิศวกรรมด้วยวิธีหาค่าเหมาะที่สุด (optimization method) เป็นแนวทางที่สะดวกและรวดเร็ว สำหรับปัญหาที่ซับซ้อนหรือไม่สามารถหาคำตอบด้วยวิธีทางคณิตศาสตร์ วิธีนี้จะอาศัยการวนซ้ำของการสุ่ม การทดสอบ และการปรับปรุงของค่าหรือตำแหน่งในบริเวณค้นหาของปัญหา จำนวนตำแหน่งที่สุ่มขึ้นกับวิธีค้นหา ลักษณะและจำนวนตัวแปรหรือมิติของปัญหา ค่าฟังก์ชันวัตถุประสงค์ของปัญหาจะเป็นตัวบ่งชี้การสิ้นสุดของการวนซ้ำค้นหา การประเมินสมรรถนะของวิธีค้นหาจะทดสอบกับปัญหาที่เรียกว่า ฟังก์ชันเทียบเคียง (benchmark function) เนื่องจากเป็นปัญหาที่รู้คำตอบ เมื่อสิ้นสุดการค้นหาแล้วได้ค่าฟังก์ชันวัตถุประสงค์ดีที่สุดเท่ากับค่าฟังก์ชันวัตถุประสงค์ของคำตอบ ไม่ว่าจากฟังก์ชันพหุฐานนิยม (multimodal function) หรือฟังก์ชันฐานนิยมเดียว (unimodal function) จะเรียกคำตอบนี้ว่าค่าเหมาะที่สุดในวงกว้าง (global optimum) แต่ถ้าได้ค่าฟังก์ชันวัตถุประสงค์ดีที่สุดต่างออกไป หมายความว่าวิธีค้นหานั้นติดกับดักค้นหา ตำแหน่งที่ค้นหาได้จะเรียกว่าค่าเหมาะที่สุดเฉพาะบริเวณ (local optimum) สำหรับค่าฟังก์ชันวัตถุประสงค์ของปัญหาจริงหรือฟังก์ชันเทียบเคียงมักกำหนดให้เป็นค่าน้อยที่สุด การเปรียบเทียบสมรรถนะโดยรวมของวิธีค้นหาจะประเมินจากทั้งค่าฟังก์ชันวัตถุประสงค์ดีที่สุดและจำนวนการประเมินฟังก์ชัน (Function Evaluation: FE)

วิธีค้นหาค่าเหมาะที่สุดเชิงกลุ่มอนุภาค (Particle Swarm Optimizer: PSO) เสนอโดย [1] มีโครงสร้างง่าย สามารถหาคำตอบของฟังก์ชันฐานนิยมเดียวที่จำนวนมิติไม่มากได้ แต่จะติดกับดักค้นหาของฟังก์ชันพหุฐานนิยม อย่างไรก็ตามเป็นวิธีค้นหาที่นิยมนำไปพัฒนา จึงมีรูปแปรมากที่สุด เช่น [2] เสนอวิธี CLPSO ที่ใช้ความน่าจะเป็นเลือกอนุภาคสำหรับปรับปรุงตำแหน่งของอนุภาคแต่ละมิติที่แยกกันสามารถหาคำตอบได้ในหลายฟังก์ชันแต่จำกัดอยู่ที่ 10 มิติ ซึ่ง CLPSO-LHS โดย Qing Wu และคณะ [3] ได้เพิ่มความสามารถค้นหาเฉพาะถิ่นด้วยการสุ่มแบบ Latin (Latin Hypercube Sampling) สามารถหาคำตอบของฟังก์ชันได้ถึง 30 มิติ แต่ทั้งสองบทความล้วนติดกับดักค้นหาในฟังก์ชัน Rosenbrock เช่นเดียวกับขั้นตอนวิธีค้นหาอื่น วิธีค้นหาโดย [4] เสนอการจำกัดบริเวณ

ค้นหาที่ปรับได้ที่เรียกว่าไฮเปอร์คิวบ์ สามารถหาคำตอบของฟังก์ชัน Rosenbrock ได้ถึง 100 มิติ ด้วยจำนวนการประเมินฟังก์ชัน (Function Evaluation: FE) เพียงแค่ 2,000 แต่ไม่ได้อธิบายถึงวิธีกำหนดตำแหน่งเริ่มต้นของไฮเปอร์คิวบ์ นอกจากนี้บทความถัดมาในโดยผู้วิจัยคนเดิมแสดงข้อมูลทดสอบที่ขัดแย้งกับผลการทดสอบในบทความก่อนหน้า โดยผลทดสอบใน [5] ของฟังก์ชัน Griewank จำนวน 30 มิติ ได้ค่าฟังก์ชันวัตถุประสงค์ดีที่สุดเท่ากับ 0.05 โดยใช้ FE เท่ากับ 2,500 ในขณะที่ผลทดสอบใน [4] ได้ค่าฟังก์ชันวัตถุประสงค์เท่ากับ 0 โดยใช้ FE เท่ากับ 750 อย่างไรก็ตามแนวคิดวิธีจำกัดบริเวณค้นหานี้จะถูกนำไปปรับใช้ในบทความนี้

บทความนี้เสนอวิธีค้นหาค่าเหมาะที่สุด 2 วิธี โดยจะสร้างบริเวณค้นหาที่เรียกว่าไฮเปอร์คิวบ์ (Hypercube: Hc) ให้ตำแหน่งและความกว้างของบริเวณค้นหาที่เหมาะสม วิธีค้นหาใน Hc จะเลือกใช้ขั้นตอนวิธีค้นหาเชิงสำรวจหรือขั้นตอนวิธีค้นหา PSO ที่ปรับปรุง โดยพิจารณาจากค่าฟังก์ชันวัตถุประสงค์จากการค้นหาที่ผ่านมาดีขึ้นหรือไม่ โดยจะอธิบายหลักการค้นหาของวิธี PSO ในหัวข้อที่ 2 แล้วจึงอธิบายวิธีค้นหาที่เสนอ วิธีทำให้ไฮเปอร์คิวบ์สามารถปรับได้ และวิธีการปรับปรุงขั้นตอนวิธี PSO ในหัวข้อที่ 3 ผลทดสอบสมรรถนะวิธีค้นหาที่เสนอและการเปรียบเทียบกับการค้นหาวิธีอื่นแสดงในหัวข้อที่ 4 สุดท้ายจะสรุปวิธีค้นหาที่เสนอและแนวทางการพัฒนาในหัวข้อที่ 5

2. ขั้นตอนวิธี PSO

การหาค่าเหมาะที่สุดของปัญหาหลายตัวแปรหรือหลายมิติด้วยวิธี PSO จะวนซ้ำการสุ่มค่าที่อาจจะเป็นคำตอบ ซึ่งเรียกว่าตำแหน่งของอนุภาค ภายในขอบเขตค้นหาของปัญหา ด้วยจำนวนการสุ่มแต่ละครั้งเท่ากับ N จากนั้นอนุภาคทั้งหมดจะถูกนำไปทดสอบในปัญหา การวนซ้ำจะสิ้นสุดลง ถ้ามีอนุภาคที่ค่าฟังก์ชันวัตถุประสงค์ดีที่สุดเท่ากับค่าฟังก์ชันวัตถุประสงค์ของคำตอบในปัญหานั้น แต่ถ้าเงื่อนไขไม่เป็นจริงจะปรับปรุงตำแหน่งของแต่ละอนุภาค แล้วนำกลับไปทดสอบจนกว่าจะได้คำตอบหรือเกินจำนวนการวนซ้ำที่กำหนด

กำหนดให้ $X_i = \left(x_i^1, x_i^2, \ldots, x_i^d\right)$ คือตำแหน่งของอนุภาคที่ถูกสุ่มตัวที่ i ของปัญหาจำนวน d มิติ เมื่อนำ X_i ไปทดสอบในปัญหา $f(X)$ แล้วได้ค่าดังสมการที่ (1) แสดงว่า X_i คือคำตอบ (X^*) ของปัญหานั้น แต่ถ้าไม่เป็นจริงก็จะวนซ้ำการปรับปรุงตำแหน่ง X_i ด้วยสมการ (2) และ (3) เสนอโดย [6] แล้วทดสอบจนกว่าจะได้คำตอบหรือเกินจำนวนการวนซ้ำที่กำหนด สำหรับ $PBest_i = \left(pBest_i^1, pBest_i^2, \ldots, pBest_i^d\right)$ คือตำแหน่งของอนุภาคตัวที่ i ที่มีค่าฟังก์ชันวัตถุประสงค์ดีที่สุดจากการค้นหาจนถึงปัจจุบัน โดยอนุภาค $PBest_i$ ตัวใดที่มีค่าฟังก์ชันวัตถุประสงค์ดีที่สุดจะเรียกว่า $GBest = \left(gBest^1, gBest^2, \ldots, gBest^d\right)$ สำหรับวัตถุประสงค์การค้นหา คือการหาตำแหน่งหรือค่าเหมาะที่สุดที่ทำให้ค่าฟังก์ชันวัตถุประสงค์น้อยที่สุด

$$f\left(X^*\right) = \min\left(f\left(X\right)\right) \tag{1}$$

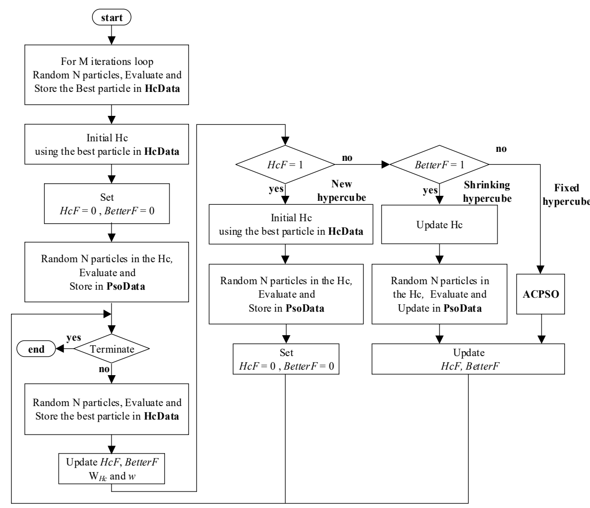
$$\begin{aligned} v_i^d \leftarrow\ & w * v_i^d + c1 * rand_i^d * \left(pBest_i^d - x_i^d\right) \\ & + c2 * rand_i^d * \left(gBest^d - x_i^d\right) \end{aligned} \tag{2}$$

$$x_i^d \leftarrow x_i^d + v_i^d \tag{3}$$

เมื่อ w คือ น้ำหนักเฉื่อย (inertia weight) $c1$ คือสัมประสิทธิ์การเร่งความเร็วไปยังตำแหน่งที่ดีที่สุดของอนุภาค (cognitive acceleration coefficient) $c2$ คือสัมประสิทธิ์การเร่งความเร็วไปยังตำแหน่งที่ดีที่สุดของกลุ่ม (social acceleration coefficient) $rand_i^d$ คือ การสุ่มค่าแบบการแจกแจงเอกรูป (uniform distribution sampling) v_i^d คือความเร็วของอนุภาค

3. วิธีค้นหาค่าเหมาะที่สุดที่เสนอ

3.1 การค้นหาค่าในไฮเปอร์คิวบ์ที่ปรับได้



รูปที่ 1 ขั้นตอนวิธีค้นหาค่าเหมาะที่สุดที่เสนอ

วิธีจำกัดบริเวณค้นหาให้เล็กลงด้วยไฮเปอร์คิวบ์ (Hypercube: Hc) รูปที่ 1 อาจสามารถค้นหาคำตอบของปัญหาหลายตัวแปรได้ แต่จะต้องหาตำแหน่งที่เหมาะสมสำหรับสร้าง Hc การใช้วิธีค้นหาเชิงสำรวจด้วยการสุ่มตำแหน่งของอนุภาคแบบการแจกแจงเอกรูป (uniform distribution sampling) จำนวน N ตัวในขอบเขตค้นหาของปัญหา แล้วนำอนุภาคไปทดสอบในปัญหา จะมีอนุภาคที่ค่าฟังก์ชันวัตถุประสงค์ดีที่สุด ถ้าวนซ้ำการสุ่มและทดสอบนี้เป็นจำนวน M ก็จะได้อนุภาคที่ค่าฟังก์ชันวัตถุประสงค์ดีที่สุดจำนวน M ตัว เมื่อเก็บข้อมูลอนุภาคเหล่านี้ใน HcData ก็จะมีตัวเลือกที่เหมาะสมสำหรับสร้าง Hc โดยจะเลือกอนุภาคใน HcData ที่ค่าฟังก์ชันวัตถุประสงค์ดีที่สุด กำหนดเป็นตำแหน่งกึ่งกลางของ Hc ให้มีความกว้าง W_{Hc} เริ่มต้นเท่ากันของทุกมิติเท่ากับ $2\,\%$ ของขอบเขตค้นหาของปัญหา อนุภาคที่ถูกเลือกจะถูกลบออกจาก HcData เพื่อไม่ให้กรณีที่สร้าง Hc ใหม่ มีบริเวณค้นหาเริ่มต้นที่ซ้ำกัน

จากนั้นกำหนดตัวบ่งชี้การสร้าง Hc ใหม่ (HcF) และตัวบ่งชี้การปรับปรุงขนาด Hc ($BetterF$) ให้ทั้งสองมีค่าเท่ากับ 0 เมื่อสุ่มตำแหน่ง

ของอนุภาคใน Hc เป็นจำนวน N ประเมินค่าฟังก์ชันวัตถุประสงค์ของอนุภาค แล้วเก็บข้อมูลอนุภาคใน PsoData ซึ่งประกอบด้วย $PBest_i$ และ $CBest$ (เมื่อ *BetterF=1*) หรือ $GBest$ (เมื่อ *BetterF=0*) สำหรับ $CBest$ คืออนุภาคตัวที่มีค่าฟังก์ชันวัตถุประสงค์ดีที่สุดจากการค้นหาล่าสุด

เมื่อค่าตัวบ่งชี้ *HcF*=0 หมายถึงการค้นหาจะยังคงใช้ Hc เดิม แต่ถ้า *HcF*=1 จะมาจาก 3 เงื่อนไขคือ วิธีค้นหาสามารถหาคำตอบได้ การเกินจำนวนการวนซ้ำที่กำหนดของการค้นหาใน Hc เดิม และผลรวมค่าฟังก์ชันวัตถุประสงค์ดีที่สุดจาก $10\,\%$ ของอนุภาค จากการค้นหาที่ผ่านมาใน Hc เดิม มีค่าที่ไม่ดีขึ้นติดต่อกันเกินจำนวนที่กำหนด ทั้ง 3 เงื่อนไขนี้ก็จะไปสู่การสร้าง Hc ใหม่

เมื่อค่าตัวบ่งชี้ *BetterF*=0 ในกรณีที่การค้นหาก่อนหน้ามาจากการปรับปรุงขนาด (Shrinking hypercube) จะหมายถึงอนุภาค $CBest$ ล่าสุดมีค่าฟังก์ชันวัตถุประสงค์น้อยกว่า $30\,\%$ ของอนุภาค $CBest$ ในวงรอบค้นหาก่อนหน้า(จะเรียกว่า $PvBest$) แต่ในกรณีที่การค้นหาก่อนหน้ามาจาก ACPSO จะหมายถึงอนุภาค $GBest$ ล่าสุดมีค่าฟังก์ชันวัตถุประสงค์น้อยกว่า $30\,\%$ ของอนุภาค $GBest$ ในวงรอบค้นหาก่อนหน้า (จะเรียกว่า $PvGBest$) ด้วยค่าบ่งชี้นี้จะให้การค้นหาไปสู่ขั้นตอนวิธี ACPSO ซึ่งยังคงใช้ Hc เดิม แต่ถ้า *BetterF*= 1 หมายถึงทั้ง $CBest$ และ $GBest$ ในปัจจุบันมีค่าฟังก์ชันวัตถุประสงค์ลดลงมากกว่า $30\,\%$ จากค่าก่อนหน้า ก็จะไปปรับขนาด Hc โดยจะให้ $CBest$ หรือ $GBest$ เป็นตำแหน่งกึ่งกลางใหม่ของ Hc ซึ่งขึ้นกับว่าการค้นหาก่อนหน้านี้มาจากการปรับปรุงขนาดไฮเปอร์คิวบ์ หรือมาจาก ACPSO สำหรับความกว้างของ Hc จะคำนวณด้วยสมการ (4) หรือ (5)

$$W_{\mathrm{Hc1}} \leftarrow \left(1 - 0.7*\exp\left(-0.3*\frac{f\left(CBest\left(GBest\right)\right)}{f\left(PvBest\left(PvGBest\right)\right)}\right)\right)*W_{\mathrm{Hc1}} \qquad (4)$$

$$W_{\mathrm{Hc2}} \leftarrow \left(0.5 + 0.2*\left(\frac{f\left(CBest\left(GBest\right)\right)}{f\left(PvBest\left(PvGBest\right)\right)}-0.1\right)^3\right)*W_{\mathrm{Hc2}} \qquad (5)$$

เมื่อ

$W_{\mathrm{Hc1(2)}}$ คือ ความกว้างของ Hc ของวิธีค้นหาที่เสนอ Hc1 และ Hc2

$f(CBest)$ คือ ค่าฟังก์ชันวัตถุประสงค์ของอนุภาค $CBest$, $f(GBest)$ คือค่าฟังก์ชันวัตถุประสงค์ของอนุภาค $GBest$

$f(PvBest)$ คือ ค่าฟังก์ชันวัตถุ ประสงค์ของอนุภาค $PvBest$

$f(PvGBest)$ คือ ค่าฟังก์ชันวัตถุประสงค์ของอนุภาค $PvGBest$

เมื่อวงรอบการค้นหาเริ่มต้น จะตรวจสอบเงื่อนไขหยุดการค้นหา โดยสามารถเลือกได้ว่าจะหยุดการค้นหาเมื่อเจอคำตอบหรือหยุดเมื่อเกินจำนวนการวนซ้ำที่กำหนด ถ้าเงื่อนไขไม่เป็นจริง จะยังคงสุ่มอนุภาคจำนวน N ตัว ในขอบเขตค้นหาของปัญหาแล้วเก็บข้อมูลของอนุภาคที่ค่าฟังก์ชันวัตถุประสงค์ดีที่สุดลงใน HcData เช่นเดียวกับก่อนเข้าสู่วงรอบการค้นหา จากนั้นจะปรับปรุงตัวบ่งชี้ *HcF* และ *BetterF* ซึ่งถ้า *BetterF=1* จะปรับปรุง $W_{\mathrm{Hc1(2)}}$ ด้วยสมการ (4) หรือ (5) และ $w_{\mathrm{Hc1(2)}}$ ด้วยสมการ (6) หรือ (7)

$$w_{\mathrm{Hc1}} = 0.3*\left(1 - 0.7*\exp\left(-0.3*\frac{f\left(CBest\left(GBest\right)\right)}{f\left(PvBest\left(PvGBest\right)\right)}\right)\right) \qquad (6)$$

$$w_{\mathrm{Hc2}} = 0.05 + 0.2*\left(\frac{f\left(CBest\left(GBest\right)\right)}{f\left(PvBest\left(PvGBest\right)\right)}-0.1\right)^3 \qquad (7)$$

เมื่อ $w_{\mathrm{Hc1(2)}}$ คือ น้ำหนักเฉื่อยใน ACPSO ของวิธีค้นหา Hc1 และ Hc2 จากนั้นก็จะไปสู่การจัดการ Hc โดยกรณีที่ *HcF*=1 จะสร้าง Hc ใหม่ แต่ถ้า *HcF*=0 และ *BetterF*=1 จะไปยังการปรับขนาด Hc นอกนั้นจะไปยังวิธีค้นหา ACPSO ใน Hc เดิม

3.2 ขั้นตอนวิธี ACPSO

วิธี PSO ซึ่งให้ w, $c1$, $c2$ มีค่าคงที่และใช้ปรับปรุงตำแหน่งของอนุภาคทุกตัว แต่ [7-8] แสดงวิธีทำให้ค่า w เปลี่ยนแปลงได้ ทำให้สมรรถนะค้นหาดีขึ้น ในวิธีค้นหาที่เสนอจะมีกรณีที่ Hc ถูกสร้างขึ้นใหม่ซึ่งเปรียบเสมือนกับการเริ่มต้นการค้นหา ดังนั้นขั้นตอนวิธี ACPSO จะให้ $w_{\mathrm{Hc1(2)}}$ มีค่าขึ้นอยู่กับค่าฟังก์ชันวัตถุประสงค์ ดังสมการ (6) และ (7) ส่วน $c1$ และ $c2$ ในสมการ (2) จะให้แต่ละอนุภาคมีค่าที่ต่างกัน ดังนั้น $c1_{\mathrm{i}}$ และ $c2_{\mathrm{i}}$ มีค่าดังสมการ (8) และ (9)

$$c1_{\mathrm{i}} = \left(1 - 0.3*\mathrm{e}^{-0.7*\frac{f(X_i)}{f(PBest_i)}}\right)*c1_{\mathrm{init}} \qquad (8)$$

$$c2_{\mathrm{i}} = \left(1 - 0.3*\mathrm{e}^{-0.7*\frac{f(X_i)}{f(GBest)}}\right)*c2_{\mathrm{init}} \qquad (9)$$

เมื่อ $c1_{\mathrm{init}} = 1$ และ $c2_{\mathrm{init}} = 2$ ค่าความเร็ว v_i^d ในวิธี PSO จะถูกจำกัดไม่ให้เกินค่าที่กำหนด ถ้าความเร็วของอนุภาคในมิติใดที่เกินจะถูกจำกัดให้มีค่าเท่ากับความเร็วสูงสุด จากนั้นถ้าปรับปรุงตำแหน่งอนุภาคด้วยสมการ (3) แล้วตำแหน่งของอนุภาคในมิติใดเกินขอบเขตค้นหาของปัญหา ก็จะให้ตำแหน่งที่เกินนั้นมีค่าเท่ากับขอบเขตค้นหาในมิตินั้น วิธีนี้จะมีโอกาสบ่อยครั้งที่ในระหว่างการค้นหา ตำแหน่งที่ปรับปรุงของอนุภาคจะซ้ำกัน เพื่อให้การค้นหามีความหลากหลาย โดยถ้าหลังจากการปรับปรุงตำแหน่ง แล้วตำแหน่งของอนุภาคในมิติใด อยู่ต่ำกว่าขอบเขตค้นหาด้านล่างของ Hc ปัจจุบัน (lower bound: LB_i) จะใช้สมการ (10) แต่ถ้าตำแหน่งของอนุภาคอยู่เกินขอบเขตด้านบนของ Hc ปัจจุบัน (upper bound: UB_i) จะใช้สมการ (11) เมื่อ $Trunc$ คือ การตัดเศษตัวเลข (Truncation)

$$x_i^d \leftarrow LB_i + \left(UB_i - LB_i\right)*Trunc\left(\frac{LB_i - x_i^d}{UB_i - LB_i}\right) \qquad (10)$$

$$x_i^d \leftarrow UB_i - \left(UB_i - LB_i\right)*Trunc\left(\frac{x_i^d - UB_i}{UB_i - LB_i}\right) \qquad (11)$$

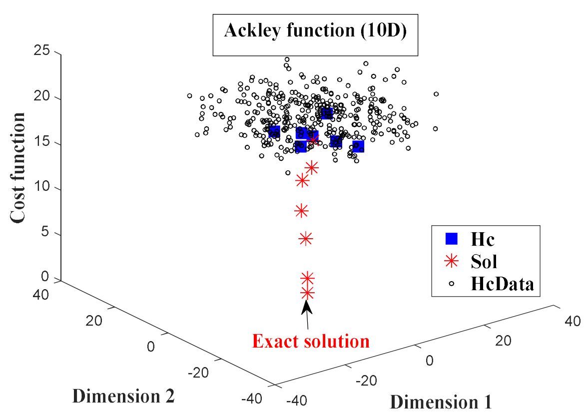
4. ผลการทดลอง

4.1 วิธีกำหนดการทดสอบ

การเปรียบเทียบสมรรถนะระหว่างวิธีค้นหา มักใช้ฟังก์ชันเกณฑ์เปรียบเทียบด้วยค่า FE เท่ากัน เมื่อ FE คือผลคูณระหว่างจำนวนประชากรหรืออนุภาคกับจำนวนการวนซ้ำ การเปรียบเทียบวิธีนี้อาจไม่ยุติธรรมในบางกรณี เช่น วิธี CLPSO ใน [5] ถูกออกแบบให้ใช้จำนวนอนุภาคเท่ากับ 10 แล้วทดสอบด้วย FE เท่ากับ 30,000 จะมีจำนวนการวนซ้ำเท่ากับ 3,000 ในขณะที่ PSOrank ใน [9] ได้นำวิธี CLPSO มาทดสอบเองใหม่ เพื่อเปรียบเทียบสมรรถนะกับวิธี PSOrank ปรากฏว่าด้วยจำนวน FE 30,000 เท่าเดิม แต่ใช้จำนวนอนุภาคเท่ากับ 30 จำนวนการวนซ้ำจึงลดลงมาเหลือเท่ากับ 1,000 ปรากฏว่าทำให้ฟังก์ชันวัตถุประสงค์ของวิธี CLPSO มีค่าที่แย่ลง นอกจากนี้การเปรียบเทียบวิธีค้นหาที่เน้นความเร็วการลู่เข้าสู่ค่าฟังก์ชันวัตถุประสงค์ดีที่สุดที่ค้นหาได้ โดยไม่ได้คำนึงว่าค่าฟังก์ชันวัตถุประสงค์ดีที่สุดที่ค้นหาได้นั้น จะเป็นค่าที่น้อยที่สุดหรือไม่ จึงขัดแย้งกับวัตถุประสงค์หลักของการค้นหา แม้ว่าความเร็วจะเป็นดัชนีวัดตัวหนึ่งของการประเมินสมรรถนะการค้นหาก็ตาม

ดังนั้นวิธีที่เสนอคือ Hc1 และ Hc2 จะเปรียบเทียบสมรรถนะกับวิธีค้นหาอื่น ด้วยจำนวนการวนซ้ำเท่ากัน ปัญหา 10 มิติจะใช้จำนวนการวนซ้ำเท่ากับ 3,000 ปัญหา 15 มิติ ใช้จำนวน 8,000 และปัญหา 20 มิติ ใช้จำนวน 10,000 แต่ละปัญหาจะทดสอบซ้ำจำนวน 30 ครั้ง ปัญหาสำหรับทดสอบจะใช้ฟังก์ชันเทียบเคียง โดยเลือกฟังก์ชันจากผลทดสอบใน [10] ซึ่งวิธีค้นหา PSO ติดกับดักค้นหา ดังนั้นจะเลือกฟังก์ชัน Sphere (F1), Rosenblock (F2), Ackley (F3), Griewank (F4), Weierstrass (F5), Rastrigin (F6) และ Schwefel 2.26 (F7) โดยมีรูปสมการดังตารางที่ 7

4.2 ผลการทดสอบกับฟังก์ชันเทียบเคียง



รูปที่ 2 ลักษณะการลู่เข้าสู่คำตอบของฟังก์ชัน Ackley

ลักษณะการลู่เข้าสู่คำตอบของวิธีค้นหา Hc1 ในรูปที่ 2 ของฟังก์ชัน Ackey (F3) จำนวน 10 มิติ ตำแหน่งของอนุภาคใน HcData จะกระจายโดยรอบขอบเขตค้นหาของปัญหา เนื่องมาจากการค้นหาเชิงสำรวจ เมื่อสิ้นสุดการค้นหาจะมีจำนวน 7 Hc ที่ถูกสร้างขึ้น และได้ค่าค้นหา (Sol) เท่ากับคำตอบ จากการค้นหาในวงรอบครั้งที่ 283 แม้ว่าจำนวนการวนซ้ำจะกำหนดเป็น 3,000 แต่วิธีค้นหานี้กลับสามารถหาคำตอบได้ก่อนครบจำนวนการวนซ้ำ

การทดสอบจำนวนอนุภาคที่เหมาะสมสำหรับวิธีค้นหาที่เสนอจะทดสอบทั้ง 7 ฟังก์ชันจำนวน 10, 15 และ 20 มิติ โดยใช้จำนวนอนุภาคตั้งแต่ 10-200 เพื่อปรับแต่งค่าพารามิเตอร์ในสมการ (5) - (8) โดยจะแสดงเฉพาะผลทดสอบของฟังก์ชันจำนวน 10 มิติ แสดงดังตารางที่ 1 และตารางที่ 2

ตารางที่ 1 การประเมินจำนวนอนุภาคสำหรับวิธีค้นหา Hc1 ใน F1-F7

Pop	F1	F2	F3	F4	F5	F6	F7
10	2.24e-18	1.24e-01	2.95e-03	2.22e-02	8.98e-01	2.18e-11	1.91e-10
20	2.73e-26	3.32e-06	1.25e+00	0	4.53e-03	6.82e-13	0
30	2.90e-20	1.84e-06	1.45e-07	2.10e-15	4.55e-02	0	0
40	3.34e-50	5.49e-07	3.12e-10	0	2.33e-01	0	0
50	2.40e-53	4.20e-08	5.10e-06	1.44e-15	2.83e-03	0	0
60	8.10e-69	3.70e-07	2.84e-14	0	1.85e-03	0	0
70	4.12e-39	7.65e-17	3.55e-15	0	1.58e-03	0	0
80	2.06e-49	4.17e-08	3.55e-15	0	1.00e-03	0	0
90	7.12e-36	9.05e-20	3.55e-15	0	2.57e-04	0	0
100	3.66e-59	1.04e-08	3.55e-15	0	8.98e-03	0	0
150	1.62e-71	5.49e-24	3.55e-15	0	1.73e-02	0	0
200	4.40e-82	3.75e-08	0	0	1.26e-03	0	0

ในตารางที่ 1 แสดงวิธีค้นหา Hc1 เมื่อใช้จำนวนอนุภาคเท่ากับ 20 สามารถหาคำตอบของ F4 และ F7 ได้ ถ้าเพิ่มจำนวนอนุภาคจะสามารถหาคำตอบได้ถึง 4 ฟังก์ชัน ส่วนอีก 3 ฟังก์ชัน จะเห็นว่าค่าฟังก์ชันวัตถุประสงค์มีค่าน้อย หมายความว่าวิธีค้นหา Hc1 ไม่ติดกับดักค้นหา ดังนั้นถ้าใช้จำนวนอนุภาคตั้งแต่ 70 ขึ้นไป จะมีสมรรถนะโดยรวมที่ดี แต่ถ้าต้องการค้นหาคำตอบจากฟังก์ชันให้ได้มากที่สุดทั้ง 3 มิติ จะต้องใช้จำนวนอนุภาคเท่ากับ 100

ตารางที่ 2 การประเมินจำนวนอนุภาคสำหรับวิธีค้นหา Hc2 ใน F1-F7

Pop	F1	F2	F3	F4	F5	F6	F7
10	2.06e-23	1.44e-07	2.75e+00	5.32e-15	1.84e-01	0	0
20	2.46e-60	4.06e-04	2.96e-04	0	1.72e-01	0	0
30	6.73e-118	5.66e-07	1.42e-14	0	7.51e-03	0	0
40	1.06e-106	3.11e-17	3.55e-15	0	1.53e-04	0	0
50	3.42e-110	2.66e-07	3.55e-15	0	3.97e-05	0	0
60	2.39e-142	3.95e-05	3.55e-15	0	5.76e-03	0	0
70	7.12e-139	7.71e-06	3.55e-15	0	8.77e-04	0	0
80	7.56e-107	1.54e-10	3.55e-15	0	1.34e-03	0	0
90	3.60e-138	1.14e-12	3.55e-15	0	0	0	0
100	5.24e-146	2.40e-15	3.55e-15	0	9.24e-04	0	0
150	2.68e-108	3.52e-12	3.55e-15	0	0	0	0
200	2.87e-106	4.65e-13	3.55e-15	0	0	0	0

ในตารางที่ 2 แสดงวิธีค้นหา Hc2 เมื่อใช้จำนวนอนุภาคเพียงแค่ 10 สามารถหาคำตอบของ F6 และ F7 ได้ โดยสามารถหาคำตอบได้ 4 ฟังก์ชันเมื่อเพิ่มจำนวนอนุภาค ส่วนอีก 3 ฟังก์ชันมีค่าวัตถุประสงค์ที่น้อยมาก เมื่อใช้จำนวนอนุภาค 90 จะมีสมรรถนะโดยรวมที่ดีซึ่งการ

ปรับแต่งค่าพารามิเตอร์ในสมการ (5) - (8) จะพิจารณาจากสมรรถนะวิธีค้นหาทั้ง 10, 15 และ 20 มิติ ซึ่งจะได้ว่าปัญหา 10, 15 และ 20 มิติ จะใช้อนุภาคจำนวน 100, 150 และ 200 ตามลำดับ

การเปรียบเทียบสมรรถนะกับวิธีค้นหาอื่นจะใช้ฟังก์ชันเทียบเคียง 7 ฟังก์ชันจำนวน 10 มิติ จำนวนการวนซ้ำ 3,000 โดยจะทดสอบซ้ำแต่ละฟังก์ชันจำนวน 30 ครั้ง แสดงค่าเฉลี่ย (mean) และค่าเบี่ยงเบนมาตรฐาน (std) ของค่าฟังก์ชันวัตถุประสงค์ดีที่สุดดังตารางที่ 3 และ 4

ตารางที่ 3 เปรียบเทียบสมรรถนะวิธีค้นหาใน F1-F4 จำนวน 10 มิติ

Algorithm		F1	F2	F3	F4
PSO-cf	mean	9.84e-105	6.98e-01	9.18e-01	1.19e-01
	std	4.21e-104	1.46+00	1.01e+00	7.11e-02
UPSO	mean	9.84e-118	1.40e+00	1.33e+00	1.04e-01
	std	3.56e-117	1.88e+00	1.48e+00	7.10e-02
FIPS	mean	3.15e-30	2.78e+00	3.75e-15	1.31e-01
	std	4.56e-30	2.26e-01	2.13e-14	9.32e-02
CPSO-H	mean	4.98e-45	1.53e+00	1.49e-14	4.07e-02
	std	1.00e-44	1.70e+00	6.98e-15	2.80e-02
CLPSO	mean	5.15e-29	2.46e+00	4.32e-14	4.56e-03
	std	2.16e-28	1.70e+00	2.55e-14	4.81e-03
CLPSO-LHS	mean	1.37e-20	9.09e-01	1.12e-10	0
	std	9.46e-21	2.17e+00	3.01e-11	0
Hc1	mean	4.10e-27	1.08e-07	2.01e-15	4.07e-17
	std	7.93e-27	1.95e-07	1.88e-15	5.97e-17
Hc2	mean	2.61e-70	2.53e-10	2.96e-15	0
	std	5.05e-70	4.69e-10	9.86e-16	0

ตารางที่ 4 เปรียบเทียบสมรรถนะวิธีค้นหาใน F5-F7 จำนวน 10 มิติ

Algorithm		F5	F6	F7
PSO-cf	mean	6.69e-01	1.25e+01	9.87e+02
	std	7.17e-01	5.17e+00	2.76e+02
UPSO	mean	1.14e+00	1.17e+01	1.08e+03
	std	1.17e+00	6.11e+00	2.68e+02
FIPS	mean	2.02e-03	2.12e+00	7.10e+01
	std	6.40e-03	1.33e+00	1.50e+02
CPSO-H	mean	1.07e-15	0	2.13e+02
	std	1.67e-15	0	1.41e+02
CLPSO	mean	0	0	0
	std	0	0	0
CLPSO-LHS	mean	5.56e-11	0	0
	std	5.55e-11	0	0
Hc1	mean	6.19e-04	1.89e-15	0
	std	9.05e-04	3.66e-15	0
Hc2	mean	3.50e-04	0	0
	std	4.43e-04	0	0

จากตารางที่ 3 วิธีค้นหา UPSO จะดีที่สุดเมื่อทดสอบใน F1 ทั้งวิธี CLPSO-LHS และ Hc2 สามารถหาคำตอบของ F4 ได้ วิธีที่เสนอ Hc1 และ Hc2 มีค่าฟังก์ชันวัตถุประสงค์ใน 4 ฟังก์ชันที่น้อยมาก แสดงว่าไม่ติดกับดักการค้นหา ส่วนในตารางที่ 4 วิธี CLPSO สามารถหาคำตอบของ F5-F7 ส่วนวิธี CLPSO-LHS และ Hc2 สามารถหาคำตอบในฟังก์ชัน F6-F7 วิธี Hc1 หาคำตอบได้ในฟังก์ชัน F7 เมื่อพิจารณาสมรรถนะโดยรวมทุกฟังก์ชัน วิธีค้นหา Hc2 จะดีที่สุด ตามด้วยวิธี Hc1

จากผลการทดสอบวิธีที่เสนอทุกฟังก์ชันทั้ง 3 มิติ แสดงดังตารางที่ 5 และตารางที่ 6 วิธีค้นหาทั้ง Hc1 และ Hc2 สามารถหาคำตอบได้ถึง 5 ฟังก์ชันที่จำนวน 10, 15 และ 20 มิติ โดยที่วิธี Hc2 จะมีสมรรถนะโดยรวมที่ดีกว่า Hc1 ซึ่งด้วยข้อมูลในตารางทั้งสองนี้สามารถบอกถึงความสามารถการค้นหาคำตอบของทั้ง Hc1 และ Hc2 ซึ่งการเปรียบเทียบสมรรถนะโดยมากมักแสดงข้อมูลค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของค่าฟังก์ชันวัตถุประ สงค์เท่านั้น จึงไม่ทราบว่าวิธีค้นหานั้นสามารถหาคำตอบได้หรือไม่

5. สรุปและวิเคราะห์

การจำกัดบริเวณค้นหาให้เล็กลงด้วยไฮเปอร์คิวบ์ โดยที่ตำแหน่งและขนาดของไฮเปอร์คิวบ์เปลี่ยนแปลง ที่ขึ้นกับผลการค้นหาที่ผ่านมาดีขึ้นหรือไม่ ถ้าการค้นหาดีขึ้นจะให้ขนาดของไฮเปอร์คิวบ์ลดลง โดยที่การค้นหาในไฮเปอร์คิวบ์จะใช้ขั้นตอนวิธีค้นหาเชิงสำรวจ แต่ถ้าการค้นหาไม่ดีขึ้นจะคงขนาดของไฮเปอร์คิวบ์และการค้นหาจะใช้ขั้นตอนวิธี ACPSO การปรับปรุงตำแหน่งของอนุภาคของ ACPSO อาจทำให้ตำแหน่งของอนุภาคบางมิติอยู่นอกขอบเขตไฮเปอร์คิวบ์ จึงป้องกันโดยใช้สมการคำนวณตำแหน่งใหม่ เพื่อไม่ให้ตำแหน่งของอนุภาคซ้ำกันระหว่างการค้นหา ทำให้การค้นหามีความหลากหลาย วิธีค้นหาทั้งสองที่เสนอสามารถหาคำตอบในหลายฟังก์ชันได้ โดยสามารถหาค่าใกล้คำตอบของฟังก์ชัน Rosenbrock ซึ่งวิธีค้นหาโดยส่วนใหญ่จะติดกับดักค้นหา จากข้อมูลที่ทดสอบ วิธีค้นหา Hc2 มีสมรรถนะดีกว่า Hc1 โดยในหลายฟังก์ชันสามารถหาคำตอบได้ก่อนครบจำนวนการวนซ้ำ

สำหรับการทดสอบวิธีค้นหาในปัญหาจริงมีความสำคัญอย่างยิ่ง เนื่องจากแต่ละมิติของปัญหาจริงอาจมีขอบเขตค้นหาที่ขนาดต่างกันมาก นอกจากนี้ตำแหน่งของคำตอบในแต่ละมิติอาจมีค่าต่างกัน ซึ่งในฟังก์ชันเทียบเคียงที่ตำแหน่งคำตอบในแต่ละมิติมีค่าเดียวกัน ดังนั้นวิธีค้นหาที่สามารถหาคำตอบได้ในฟังก์ชันเทียบเคียง อาจติดกับดักค้นหาในคำตอบจริง โดยเฉพาะวิธีค้นหาที่อาศัยระยะยูคลิด (Euclid distance) ข้อด้อยวิธีค้นหาที่เสนอคือจำนวนอนุภาคและจำนวนการประเมินฟังก์ชันที่มาก แต่ด้วยโครงสร้างของวิธีที่เสนอ สามารถปรับปรุงได้ง่าย โดยสามารถเปลี่ยนวิธีค้นหาเชิงสำรวจที่มีประสิทธิภาพมากขึ้นหรือใช้ขั้นตอนวิธีค้นหาอื่นแทน ACPSO นอกจากนี้สิ่งสำคัญที่ทำให้การค้นหาวิธีนี้ประสบผลสำเร็จคือ การออกแบบเงื่อนไขที่เหมาะสมของการสร้างไฮเปอร์คิวบ์ใหม่ ที่ทำให้ไม่ติดกับดักค้นหา

เอกสารอ้างอิง

[1] R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” *In Proc. The Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39-43.

[2] J. J. Liang, A. K. Qin, P. N. Suganthan and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *in IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, June., pp. 281-295, 2006.

[3] Q. Wu. Et al., "A modified comprehensive learning particle swarm optimizer and its application in cylindricity error evaluation problem," *Journal of Mathematical Biosciences and Engineering*, vol. 16, no. 3, Feb., pp. 1190–1209, 2019.

[4] M. Tunay, "Evolutionary search algorithm based on hypercube optimization for high-dimension functions," *International Journal of Computational and Experimental Science and Engineering*, vol. 6, no. 1, Mar., pp. 42 - 62, 2020.

[5] M. Tunay and R. Abiyev, "Improve hypercube optimisation search algorithm for optimisation of high dimensional functions," *Mathematical Problems in Engineering*, 22, Apr., 2022. [Online serial]. doi.org/10.1155/2022/6872162.

[6] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *1998 Proc. IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 69-73.

[7] Y. S. Kushwah and R. K. Shrivastava, "Particle swarm optimization with dynamic inertia weights," *International Journal of Research and Scientific Innovation*, 25, July, 2017. [Online serial]. Available: https://rsisinternational.org/IJRSI/Issue44/129-135.pdf [Accessed July 25 2017].

[8] W. Liu et al., "A novel sigmoid-function-based adaptive weights particle swarm optimizer," in *IEEE Transactions on Cybernetics*, vol. 51, no. 2, Feb., pp. 1085-1093, 2021.

[9] R. Akbari and K. Ziarati, "A rank based particle swarm optimization algorithm with dynamic adaptation," *Journal of Computational and Applied Mathematics*, vol. 235, no.8, Aug., pp. 2694–2714, 2011.

[10] V. Plevris and G. Solorzano, "A collection of 30 multidimensional functions for global optimization benchmarking," *Data*, 11, Apr. 2022. [Online]. doi.org/10.3390/data7040046.

ตารางที่ 5 สมรรถนะวิธีค้นหา Hc1 และ Hc2 ของฟังก์ชัน F1-F4 จำนวน 10, 15 และ 20 มิติ

Function		F1			F2			F3			F4		
Dimension		10	15	20	10	15	20	10	15	20	10	15	20
Hc1	mean	4.10e-027	7.03e-036	2.30e-038	1.08e-007	1.62e-012	1.72e-008	2.01e-015	4.14e-015	8.65e-010	4.07e-017	7.40e-018	0
	std	7.93e-027	1.36e-035	4.45e-038	1.95e-007	2.34e-012	3.22e-008	1.88e-015	2.91e-015	1.67e-009	5.97e-017	1.38e-017	0
	best	3.46e-098	1.99e-078	3.36e-105	1.27e-024	3.57e-021	8.92e-023	0	0	0	0	0	0
	worst	1.23e-025	2.11e-034	6.91e-037	3.04e-006	1.10e-011	4.59e-007	7.10e-015	3.55e-015	2.59e-008	3.33e-016	1.11e-016	0
	count	0	0	0	0	0	0	14	8	7	22	28	30
Hc2	mean	2.61e-070	5.65e-088	1.83e-093	2.53e-010	3.43e-015	2.04e-013	2.96e-015	3.43e-015	2.96e-015	0	0	0
	std	5.05e-070	1.03e-087	3.54e-093	4.69e-010	4.58e-016	3.79e-013	9.86e-016	4.58e-016	9.86e-016	0	0	0
	best	1.94e-156	8.08e-143	8.08e-147	1.33e-025	4.20e-026	4.93e-030	0	0	0	0	0	0
	worst	7.84e-069	1.35e-086	5.02e-092	4.69e-024	2.07e-012	5.60e-012	3.55e-015	7.10e-015	3.55e-015	0	0	0
	count	0	0	0	0	0	0	5	5	5	30	30	30

ตารางที่ 6 สมรรถนะวิธีค้นหา Hc1 และ Hc2 ของฟังก์ชัน F5-F7 จำนวน 10, 15 และ 20 มิติ

Function		F5			F6			F7		
Dimension		10D	15D	20D	10D	15D	20D	10D	15D	20D
Hc1	mean	6.19e-004	8.31e-003	2.82e-002	1.89e-015	0	0	0	0	0
	std	9.05e-004	7.92e-003	3.03e-002	3.66e-015	0	0	0	0	0
	best	0	0	0	0	0	0	0	0	0
	worst	8.98e-003	5.40e-002	1.84e-001	5.68e-014	0	0	0	0	0
	count	6	1	2	29	30	30	30	30	30
Hc2	Mean	3.50e-004	9.48e-005	4.40e-004	0	0	0	0	0	0
	std	4.43e-004	1.41e-004	4.88e-004	0	0	0	0	0	0
	best	0	0	0	0	0	0	0	0	0
	worst	5.28e-003	9.79e-004	3.94e-003	0	0	0	0	0	0
	count	6	19	9	30	30	30	30	30	30

ตารางที่ 7 ฟังก์ชันเทียบเคียง

ชื่อฟังก์ชัน	รูปสมการ	ขอบเขตค้นหา	$f\left(X^*\right)$	X^*
Sphere (F1)	$f_1(x) = \sum_{i=1}^{d} x_i^2$	$[-100,\ 100]^d$	0	$\{0,\ 0,\ ...,\ 0\}$
Rosenbrock (F2)	$f_2(x) = \sum_{i=1}^{d-1}\left(100\cdot\left(x_{i+1}-x_i^2\right) + \left(x_i-1\right)^2\right)$	$[-2.048,\ 2.048]^d$	0	$\{1,\ 1,\ ...,\ 1\}$
Ackley (F3)	$f_3(x) = -20\cdot\exp\left(-0.2\cdot\sqrt{\frac{1}{d}\cdot\sum_{i=1}^{d}x_i^2}\right)-\exp\left(\frac{1}{d}\cdot\sum_{i=1}^{d}\cos\left(2\pi x_i\right)\right)+e+20$	$[-32.768,\ 32.768]^d$	0	$\{0,\ 0,\ ...,\ 0\}$
Griewank (F4)	$f_4(x) = \frac{1}{4000}\cdot\sum_{i=1}^{d}x_i^2 - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-100,\ 100]^d$	0	$\{0,\ 0,\ ...,\ 0\}$
Weierstrass (F5)	$f_5(x) = \sum_{i=1}^{d}\left(\sum_{k=0}^{k\max}\left(a^k\cdot\cos\left(2\pi b^k\cdot\left(x_i+0.5\right)\right)\right)\right) - d\cdot\sum_{k=0}^{k\max}\left(a^k\cdot\cos\left(\pi b^k\right)\right)$ $a = 0.5$, b=3, kmax=20	$[-0.5,\ 0.5]^d$	0	$\{0,\ 0,\ ...,\ 0\}$
Rastrigin (F6)	$f_6(x) = \sum_{i=1}^{d}\left(x_i^2 - 10\cdot\cos\left(2\pi x_i\right)\right)+10\cdot d$	$[-5.12,\ 5.12]^d$	0	$\{0,\ 0,\ ...,\ 0\}$
Schwefel (F7)	$f_7(x) = -\sum_{i=1}^{d}\left(x_i\cdot\sin\left(\sqrt{\lvert x_i\rvert}\right)\right)+418.9828872724337\cdot d$	$[-500,\ 500]^d$	0	$\{c,\ c,\ ...,\ c\}$ $c = 420.968946359982025$