

การเพิ่มประสิทธิภาพอัลกอริทึมผลต่างวิวัฒนาการด้วยวิธี Restart mechanism

ปฎิภาณ พลวิรัตน์¹, ศิวตล เสถียรพัฒนากุล² และ ดวงเพ็ญ เจตน์พิพัฒน์พงษ์^{*3}

คณะวิศวกรรมศาสตร์กำแพงแสนมหาวิทยาลัยเกษตรศาสตร์ อ.กำแพงแสน จ.นครปฐม 73140

Received: 21 May 2025; Revised: 15 October 2025; Accepted: 11 November 2025

บทคัดย่อ

งานวิจัยชิ้นนี้ได้นำเสนอการเพิ่มประสิทธิภาพอัลกอริทึมผลต่างวิวัฒนาการด้วยวิธี Restart mechanism (Enhanced Differential Evolution with Restart Mechanism for Numerical Optimization: RMDE) ที่ปรับปรุงประสิทธิภาพในการค้นหาคำตอบของอัลกอริทึม An enhanced Differential Evolution algorithm with novel control parameter adaption schemes for numerical optimization (PaDE) โดยมีการนำกลไก Restart mechanism จาก Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism (AMECoDEs) ที่ประกอบไปด้วยกลไกช่วยเพิ่มความหลากหลายของประชากรและกลไกช่วยให้ประชากรลู่เข้า นอกจากนี้ RMDE ได้มีการนำเสนอการปรับปรุงการปรับค่าพารามิเตอร์ได้แก่ ค่าเฉลี่ยอัตราการครอสโอเวอร์ และ ค่าเฉลี่ยแฟกเตอร์ขยายผลต่าง ในการทดสอบ RMDE ด้วยชุดปัญหาทดสอบ CEC2017 กับอัลกอริทึมผลต่างวิวัฒนาการที่เป็น state-of-the-art ได้แก่ jSO MPaDE JADE SHADE และ PaDE ที่ขนาดของปัญหา 10 และ 30 ตัวแปรตัดสินใจ พบว่า ในปัญหาขนาด 10 ตัวแปรตัดสินใจ jSO SHADE และ MPaDE มีประสิทธิภาพในการค้นหาคำตอบดีกว่า RMDE ในขณะที่ JADE และ PaDE มีประสิทธิภาพในการค้นหาคำตอบใกล้เคียงกันกับ RMDE แต่เมื่อขนาดของปัญหาเพิ่มขึ้นเป็น 30 ตัวแปรตัดสินใจพบว่า RMDE มีประสิทธิภาพในการค้นหาคำตอบได้ดีกว่าอัลกอริทึม JADE MPaDE และ PaDE ส่วน jSO และ SHADE พบว่ามีประสิทธิภาพในการค้นหาคำตอบใกล้เคียงกับ RMDE นอกจากนี้เมื่อพิจารณาผลการค้นหาคำตอบในฟังก์ชันประเภท Hybrid พบว่าอัลกอริทึม RMDE มีประสิทธิภาพในการค้นหาคำตอบได้ดีกว่าอัลกอริทึมตัวอื่นทุกตัวที่นำมาเปรียบเทียบ สำหรับการประยุกต์ใช้งาน RMDE และ PaDE ในปัญหาการจัดการคลังสินค้าพบว่า RMDE มีประสิทธิภาพในการค้นหาคำตอบได้ดีกว่า PaDE ในปัญหาขนาดกลางและปัญหาขนาดใหญ่

คำสำคัญ: อัลกอริทึมผลต่างวิวัฒนาการด้วยกลไกการเริ่มใหม่, เมต้าฮิวริสติกส์, การจัดการคลังสินค้า

* Corresponding author. E-mail: fengdpj@ku.ac.th

^{1,2,3} ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

An enhanced Differential Evolution with Restart Mechanism for Numerical Optimization: RMDE

Patipan Polvirat¹, Siwadol Sateanpattanakul² Duangpen Jetpipattanapong^{*3}

Faculty of Engineering at Kamphaeng Saen, Kasetsart University, Kamphaeng Saen,
Nakhon Pathom 73140

Received: 21 May 2025; Revised: 15 October 2025; Accepted: 11 November 2025

Abstract

This paper proposes an enhanced Differential Evolution algorithm with a Restart Mechanism (RMDE) for numerical optimization. The RMDE algorithm improves the search performance of an enhanced Differential Evolution algorithm with novel control parameter adaptation schemes (PaDE) by incorporating a restart mechanism. This mechanism, adapted from the Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism (AMECoDEs), includes strategies to enhance population diversity and accelerate convergence. Additionally, RMDE refines the parameter adaptation techniques for the average crossover value and the average scaling factor. Experimental results on the CEC2017 test suite compared RMDE with several state-of-the-art differential evolution algorithms, including jSO, MPADe, JADE, SHADE, and PaDE, on problems with 10 and 30 dimensions. For the 10-variable problems, jSO, SHADE, and MPADe demonstrated superior search performance compared to RMDE, while JADE and PaDE showed comparable performance. However, as the problem size increased to 30 variables, RMDE significantly outperformed JADE, MPADe, and PaDE, with jSO and SHADE performing similarly. Furthermore, when considering the hybrid function type, the RMDE algorithm demonstrated superior search performance compared to all other algorithms. In a practical application to warehouse management problems, RMDE proved to be more effective than PaDE at finding solutions for medium and large-scale problems.

Keywords: different evolution algorithm with a restart mechanism, metaheuristics, warehouse management

* Corresponding author. E-mail: fengdpj@ku.ac.th

^{1,2,3} Department of Computer Engineering, Faculty of Engineering at Kamphaeng Saen, Kasetsart University

1. บทนำ

อัลกอริทึมผลต่างวิวัฒนาการ (Differential Evolution Algorithm: DE) [1] เป็นหนึ่งในอัลกอริทึมเชิงวิวัฒนาการ (Evolutionary Algorithm: EA) ซึ่งอาศัยประชากรในการค้นหาคำตอบ (Population based search) ที่ได้รับความนิยมนำมาใช้ในการแก้ไขปัญหาการหาค่าที่ดีที่สุด (Optimization) ไม่ว่าจะเป็น ปัญหาการออกแบบทางวิศวกรรม ปัญหาการออกแบบโครงข่ายประสาทเทียม ปัญหาการจัดสรรพลังงาน เป็นต้น สำหรับขั้นตอนการค้นหาคำตอบของ DE ประกอบไปด้วย 4 ขั้นตอนได้แก่ 1. การสร้างประชากรเริ่มต้น (Initialization) 2. การกลายพันธุ์ (Mutation) 3. การครอสโอเวอร์ (Crossover) และ 4. การคัดเลือก (Selection) จากขั้นตอนของ DE ที่ไม่ซับซ้อนและทำให้ประชากรสามารถเข้าสู่ได้อย่างรวดเร็วส่งผลให้ DE มีความสามารถในการค้นหาคำตอบได้รวดเร็วและมีประสิทธิภาพ แต่ในบางครั้งจากการเข้าสู่อย่างรวดเร็วของ DE ส่งผลให้ประชากรไปติดอยู่ในบริเวณคำตอบที่ดีที่สุดแบบท้องถิ่น (Local Optima) และไม่สามารถหลุดออกจากบริเวณดังกล่าว

เพื่อให้ประชากรสามารถหลุดออกจากบริเวณที่เป็น Local optima และพัฒนาคำตอบได้อย่างมีประสิทธิภาพ จึงได้มีงานวิจัยที่นำเสนอการใช้วิธี Restart mechanism เช่น Wetweeraopong and Puphasuk [2] ที่นำเสนอการปรับปรุงอัลกอริทึมผลต่างวิวัฒนาการเพื่อนำมาใช้ระบบสมการไม่เชิงเส้น ในชื่อ An improved differential evolution algorithm with a restart technique to solve systems of nonlinear equations (DE-R) ด้วยการใช้วิธี Restart technique เพื่อป้องกันไม่ให้ประชากรเข้าสู่บริเวณคำตอบที่เป็น Local optima อย่างรวดเร็วและป้องกันไม่ให้เกิดการหยุดพัฒนาคำตอบ โดยในทุก 200 รอบ DE-R จะทำการสุ่มประชากรออกมาบางส่วนเพื่อนำมาสร้างชิ้นใหม่ด้วยวิธีการเช่นเดียวกันกับการสร้างประชากรเริ่มต้น นอกจากนี้ยังมีการใช้กลไกการกลายพันธุ์สองแบบได้แก่กลไกการกลายพันธุ์แบบเดิม (basic-mutation) และกลไกการกลายพันธุ์แบบที่ใช้ประชากรที่ดี (xbest-mutation) และในขั้นตอนการกลายพันธุ์ DE-R จะทำการสุ่มตัวเลขด้วยการแจกแจงเอกรูป ถ้า

ตัวเลขที่สุ่มได้มีค่าน้อยกว่า 0.5 อัลกอริทึม DE-R จะใช้กลไกการกลายพันธุ์แบบ basic mutation และ ถ้าตัวเลขที่สุ่มได้มีค่ามากกว่า 0.5 อัลกอริทึม DE-R จะเลือกกลไกการกลายพันธุ์แบบ xbest-mutation [2] นอกจากนี้ DE-R ยังปรับเปลี่ยนค่าแฟกเตอร์ขยายผลต่าง (F) จากเดิมที่ใช้ค่าคงที่มาเป็นค่าสุ่มเลือกในช่วง $[0.5, 0.7]$ จากผลการทดลองพบว่า DE-R ที่นำเสนอมีประสิทธิภาพในการค้นหาคำตอบได้ดีกว่าอัลกอริทึม DE PSO และ PSO-NMM ที่นำมาเปรียบเทียบ Lin and Meng [3] ได้นำเสนอการปรับปรุงอัลกอริทึมผลต่างวิวัฒนาการชื่อ An adaptive differential evolution with enhanced diversity and restart mechanism (ADE-DMRM) ที่ปรับปรุงและพัฒนากลไก Restart mechanism โดยวัดค่าความหลากหลายของประชากรที่คำนวณได้จากรากที่สองของผลรวมระยะทางจากประชากรเฉลี่ยกับประชากรทุกตัวในรอบการคำนวณปัจจุบัน หากค่าความหลากหลายที่คำนวณได้มีค่าน้อยกว่าที่กำหนด ADE-DMRM จะใช้กลไก Restart mechanism ที่อาศัยประชากรที่ดีที่สุด ประชากรปัจจุบัน และประชากรเฉลี่ยในการสร้างประชากรขึ้นมาใหม่ จากผลการทดลองในชุดปัญหาทดสอบ CEC2013 CEC2014 และ CEC2017 พบว่า ADE-DMRM มีประสิทธิภาพในการค้นหาคำตอบได้ดีกว่าอัลกอริทึมตัวอื่นที่นำมาเปรียบเทียบ Cui et al. ได้นำเสนออัลกอริทึม Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism (AMECoDEs) [4] ที่พัฒนากลไก Restart mechanism ซึ่งประกอบไปด้วย กลไกเพิ่มความหลากหลายของประชากรที่ใช้เพิ่มความหลากหลายของประชากรเมื่อค่าความหลากหลายของประชากรมีค่าน้อยกว่าที่กำหนด และกลไกช่วยให้ประชากรเข้าสู่ที่ใช้เพื่อช่วยให้ประชากรเข้าสู่บริเวณ global optima เมื่อประชากรไม่สามารถพัฒนาคำตอบได้ดีขึ้นเป็นเวลานาน จากผลการทดลองพบว่า AMECoDEs มีประสิทธิภาพในการค้นหาคำตอบได้ดีกว่าอัลกอริทึมผลต่างวิวัฒนาการตัวอื่นที่เป็น state-of-the-art

นอกจากการใช้วิธี Restart mechanism แล้ว อัลกอริทึมผลต่างวิวัฒนาการยังใช้วิธีปรับปรุงกลไกการกลายพันธุ์และวิธีการปรับค่าพารามิเตอร์ เช่น อัตรา

การครอสโอเวอร์ (Crossover rate : CR) และ แฟกเตอร์ขยายผลต่าง (Scaling Factor: F) เพื่อช่วยเพิ่มประสิทธิภาพในการค้นหาคำตอบของอัลกอริทึม เช่น Liu et al. [5] ได้นำเสนอ A novel clustering-based differential evolution with 2 multi-parent crossover for global optimization (2-MPCs-CDE) ที่มีการแบ่งประชากรออกเป็นกลุ่มด้วยวิธี K-mean clustering และปรับปรุงกลไกการกลายพันธุ์โดยใช้จุด centroid ของแต่ละกลุ่มมาช่วยในการกลายพันธุ์ นอกจากนี้ยังมีการนำเอาวิธีการแบบ Multi-parent crossover ที่ช่วยให้ประชากรมีความหลากหลายและสามารถค้นหาคำตอบได้อย่างมีประสิทธิภาพ Brest et al. [6] ได้นำเสนอ Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems (jDE) ที่ได้ทำการปรับค่าพารามิเตอร์ได้แก่ อัตราการครอสโอเวอร์ (CR) และแฟกเตอร์ขยายผลต่าง (F) ของประชากรแต่ละตัวด้วยวิธีการสุ่มค่า CR ด้วยการแจกแจงแบบเอกรูป ในช่วง $[0,1]$ และ สุ่มค่า F ด้วยการแจกแจงแบบเอกรูป ในช่วง $[0.1,0.9]$

Shang and Sanderson [7] นำเสนอ Adaptive Differential Evolution With Optional External Archive (JADE) ได้ปรับปรุงกลไกการกลายพันธุ์ DE/current-to-best/1 ขึ้นใหม่ในชื่อ DE/current-to-pbest/1 และ JADE ยังมีการนำเสนอมูลวิธีการปรับค่าพารามิเตอร์ CR และ F จากค่าเฉลี่ยของ CR และ F ในประชากรแต่ละตัวที่ทำให้ประชากรเกิดการยอมรับคำตอบใหม่ในแต่ละรอบการคำนวณ ในเวลาต่อมา Tanabe and Fukunaga [8] ได้นำเสนอ Success-history based parameter adaptation for Differential Evolution (SHADE) ที่ปรับปรุง JADE ด้วยการสร้างเซตของค่าเฉลี่ย CR และ F และสุ่มค่าเฉลี่ยดังกล่าวเพื่อนำมาใช้ในการสร้าง CR และ F ของประชากรแต่ละตัว นอกจากนี้ Tanabe and Fukunaga [9] ได้นำแนวคิดของ SHADE มาปรับปรุงโดยเพิ่มเติมกลไกการปรับลดประชากรเชิงเส้นในชื่อ Improving the search performance of SHADE using linear population size reduction (LSHADE)

Meng et al. [10] ได้นำแนวคิดของ SHADE และ LSHADE มาทำการปรับปรุงในชื่อ Parameters with Adaptive Learning Mechanism (PALMDE) for the enhancement of Differential Evolution (PALMDE) ที่มีการแบ่งประชากรออกเป็น 8 กลุ่มในแต่ละกลุ่มจะปรับค่าพารามิเตอร์ CR ด้วยค่าความน่าจะเป็นในการเลือกกลุ่ม และค่า F ด้วยวิธีเดียวกันกับ SHADE และใช้วิธีการปรับลดประชากรของ LSHADE มาปรับปรุง PALMDE เป็น LPALMDE ในเวลาต่อมา Meng et al. [11] ยังได้นำเสนอ An enhanced Differential Evolution algorithm with novel control parameter adaption schemes for numerical optimization (PaDE) ที่ปรับปรุงมาจาก LSHADE โดยปรับเปลี่ยนกลไกการลดประชากรของ LSHADE เนื่องจากกลไกการลดประชากรแบบเชิงเส้น (Linear population size reduction) ใน LSHADE ทำให้ประชากรลดลงอย่างรวดเร็วและทำให้การค้นหาคำตอบไม่มีประสิทธิภาพ ดังนั้น PaDE ได้นำเสนอมูลไกการลดประชากรแบบพาราโบลา (Parabolic population size reduction) ที่จะค่อยๆ ลดจำนวนของประชากรลงในแต่ละรอบการคำนวณทำให้ PaDE มีประสิทธิภาพในการค้นหาคำตอบได้ดีกว่า LSHADE ที่ใช้วิธีการปรับลดประชากรแบบเชิงเส้น นอกจากนี้ PaDE ยังมีการแบ่งประชากรออกเป็น 4 กลุ่มด้วยวิธีการเดียวกันกับ PALMDE โดยจะปรับค่า CR เฉพาะกลุ่มที่มีค่าความน่าจะเป็นในการถูกเลือกน้อยที่สุด เมื่อทำการทดสอบประสิทธิภาพในการค้นหาคำตอบของ PaDE เทียบกับอัลกอริทึม state-of-the-art ของ DE ได้แก่ JADE, jDE, SHADE, LSHADE และ PALMDE พบว่า PaDE มีประสิทธิภาพในการค้นหาคำตอบที่ดีกว่าอัลกอริทึม state-of-the-art ของ DE

อย่างไรก็ตามถึงแม้ว่าประสิทธิภาพของ PaDE จะสูงกว่าอัลกอริทึม state-of-the-art ของ DE แต่ด้วยกลไกการกลายพันธุ์ของ PaDE ที่อาศัยการสุ่มประชากรที่ดีที่สุดจากกลุ่มประชากรที่ดีที่สุด โดยการสุ่มในลักษณะดังกล่าว บางครั้งอาจจะสุ่มได้ประชากรที่ติดอยู่ในบริเวณ Local optima ส่งผลให้ประชากรในรุ่นถัดไปติดอยู่ในบริเวณ Local optima ตามไปด้วย เนื่องจาก PaDE ยังไม่มีวิธีการที่มีประสิทธิภาพที่สามารถทำให้ประชากรที่ติดอยู่ในบริเวณ

Local optima หลุดพ้นออกจากบริเวณดังกล่าวได้ เพื่อช่วยให้ประชากรหลุดออกจาก local optima ผู้วิจัยนำเสนอการเพิ่มประสิทธิภาพอัลกอริทึมผลต่างวิวัฒนาการด้วยวิธี Restart Mechanism for Numerical Optimization (RMDE) โดยมีรายละเอียดดังนี้

1. เพิ่มกลไก Restart mechanism ที่นำมาจากอัลกอริทึม AMECODEs [4] โดยกลไกดังกล่าวประกอบไปด้วย กลไกการเพิ่มความหลากหลายของประชากร และกลไกช่วยให้ประชากรรู้เข้า ซึ่งช่วยให้ประชากรมีความหลากหลายเพิ่มขึ้นและพัฒนาคำตอบได้อย่างมีประสิทธิภาพ [4]

2. ปรับปรุงค่าพารามิเตอร์ค่าเฉลี่ยแฟกเตอร์ขยายผลต่างโดยให้ประชากรในแต่ละกลุ่มมีค่าเฉลี่ยแฟกเตอร์ขยายผลต่างแยกออกจากกัน

ในส่วนถัดไปของงานวิจัยชิ้นนี้ประกอบไปด้วยหัวข้อที่ 2 การทบทวนวรรณกรรม หัวข้อที่ 3 อัลกอริทึมที่นำเสนอ หัวข้อที่ 4 ผลการทดลอง หัวข้อที่ 5 สรุปผลการทดลอง

2. การทบทวนวรรณกรรม

2.1 อัลกอริทึมผลต่างวิวัฒนาการ (Differential Evolution algorithm: DE)

อัลกอริทึมผลต่างวิวัฒนาการ (DE) ที่ใช้วิธีการพัฒนาคำตอบของประชากร $X_i^G = [x_{i,1}^G, x_{i,2}^G, \dots, x_{i,D}^G]$ ในแต่ละรอบการคำนวณ เมื่อ i คือลำดับของประชากร G คือรอบการคำนวณและ D คือจำนวนตัวแปรตัดสินใจ DE ประกอบไปด้วยขั้นตอนทั้งหมด 4 ขั้นตอนได้แก่ 1. สร้างประชากรเริ่มต้น (Initialization) 2. การกลายพันธุ์ (Mutation) 3. การครอสโอเวอร์ (Crossover) และ 4. การคัดเลือก (Selection) ดังนี้

2.1.1 การสร้างประชากรเริ่มต้น (Initialization)

เป็นขั้นตอนการสร้างเซตของประชากรเริ่มต้น P^0 ที่ประกอบไปด้วยประชากรจำนวน NP ตัว โดยที่ $P^0 = \{X_1^0, X_2^0, X_3^0, \dots, X_{NP}^0\}$ ด้วยวิธีการสุ่มที่อยู่ในปริภูมิคำตอบ (search space) ดังสมการที่ (1)

$$x_{i,j}^0 = lb_j + rand(0,1) \cdot (ub_j - lb_j) \quad (1)$$

โดยที่ $x_{i,j}^0$ คือ ตัวแปรตัดสินใจที่ j ของประชากรตัวที่ i

lb_j คือ ขอบเขตล่างของตัวแปรตัดสินใจที่ j

ub_j คือ ขอบเขตบนของตัวแปรตัดสินใจที่ j

$rand(0,1)$ คือ ฟังก์ชันสุ่มของการแจกแจงแบบเอกรูป ในช่วง $[0,1]$

2.1.2 การกลายพันธุ์ (Mutation)

เป็นขั้นตอนการสร้างเวกเตอร์กลายพันธุ์หรือมิวแทนเวกเตอร์ (Mutant vector: V) ของประชากรตัวที่ i ในรอบการคำนวณที่ G จากกลไกการกลายพันธุ์ (Mutation strategy) DE/rand/1 ดังสมการที่ (2)

$$V_i^G = X_{r_1}^G + F \cdot (X_{r_2}^G - X_{r_3}^G) \quad (2)$$

โดยที่ V_i^G คือมิวแทนเวกเตอร์ของประชากรตัวที่ i ในรอบการคำนวณที่ G ที่ประกอบไปด้วยตัวแปรตัดสินใจ $v_{i,j}^G$ จำนวน D ตัว

$X_{r_1}^G, X_{r_2}^G, X_{r_3}^G$ คือ เวกเตอร์สุ่มที่ได้จากประชากรโดยเวกเตอร์สุ่มทั้ง 3 ต้องแตกต่างกันและไม่เหมือนกับเวกเตอร์ X_i^G

2.1.3 การครอสโอเวอร์ (Crossover)

เป็นขั้นตอนการสร้างไตรอัลเวกเตอร์ (Trial vector: U_i^G) ของประชากรตัวที่ i ในรอบการคำนวณที่ G ดังสมการที่ (3)

$$u_{i,j}^G = \begin{cases} v_{i,j}^G & \text{if } rand(0,1) \leq CR \\ x_{i,j}^G & \text{otherwise} \end{cases} \quad (3)$$

โดยที่ $u_{i,j}^G$ คือตัวแปรตัดสินใจที่ j ใน U_i^G และ

$$U_i^G = \{u_{i,1}^G, u_{i,2}^G, \dots, u_{i,j}^G\}$$

เมื่อเสร็จสิ้นกระบวนการ Crossover อัลกอริทึม DE จะทำการตรวจสอบค่าของตัวแปรตัดสินใจแต่ละตัวของ U_i^G ว่าอยู่ในช่วงขอบเขตบนและขอบเขตล่างที่กำหนดหรือไม่ หากตัวแปรตัดสินใจมีค่าน้อยกว่าค่าขอบเขตล่าง ($u_{i,j}^G < lb_j$) จะกำหนดให้ $u_{i,j}^G = lb_j$ และในทำนองเดียวกันหากตัวแปร

ตัดสินใจดังกล่าวมีค่ามากกว่าค่าขอบเขตบน ($u_{i,j}^G > ub_j$) จะกำหนดให้ $u_{i,j}^G = ub_j$

2.1.4 การคัดเลือก (Selection)

เป็นขั้นตอนการคัดเลือกประชากรรุ่นถัดไปโดยการนำค่า fitness value ของ U_i^G และ X_i^G มาทำการเปรียบเทียบและคัดเลือกเวกเตอร์ที่ให้ค่า fitness value ดีที่สุดไปเป็นประชากรรุ่นถัดไป สำหรับการคัดเลือกแสดงดังสมการที่ (4)

$$X_i^{G+1} = \begin{cases} U_i^G & \text{if } f(U_i^G) < f(X_i^G) \\ X_i^G & \text{otherwise} \end{cases} \quad (4)$$

โดยที่ X_i^{G+1} คือประชากรตัวที่ i ในรอบการคำนวณถัดไป $f(U_i^G), f(X_i^G)$ คือค่า fitness value ของ U_i^G และ X_i^G ตามลำดับ

Algorithm 1: Pseudo code of stochastic universal selection.

Input: Selection probability $P(j)$, $j \in \{1, 2, \dots, k\}$ of each group and population size ps of all individuals;
Output: The index of group which the corresponding individual is distributed into;

```

1 Initialization  $space = \frac{1}{ps}$ ,  $sumP = 0$ , generate a random value  $rnd$ ,  $rnd \in (0, 1)$ , allocate three arrays  $label$   $nlabel$  and  $index$  with the size all equaling to  $ps$ .
2 for  $i = 1; i < ps; i++$  do
3    $rndn(i) = rnd + \frac{i}{space}$ ;
4    $label(i) = 1$ ;
5    $index(i) = 0$ ;
6 end
7 for  $j = 1; j \leq k; j++$  do
8    $sumP = sumP + P(j)$ ;
9   for  $i = 1; i < ps; i++$  do
10     $nlabel(i)$  records binary inverted value of  $label(i)$ ;
11     $label(i) = nlabel(i) \& (rndn(i) < sumP)$ ;
12     $index(i) = index(i) + label(i) \cdot j$ ;
13     $label(i) = label(i) | index(i)$ ;
14  end
15 end
16 shuffle the elements in array  $index$ ;
17 return  $index$ ;
```

รูปที่ 1 ขั้นตอนการทำงานของ Stochastic universal selection

2.2 An enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization (PaDE)

PaDE เป็นอัลกอริทึมที่พัฒนาต่อยอดจาก DE โดยเพิ่มประสิทธิภาพในการค้นหาคำตอบจากสามวิธี ได้แก่ 1. เพิ่มกลไกการแบ่งกลุ่มประชากร 2. ปรับปรุงกลไกการกลายพันธุ์ และ 3. ปรับขนาดประชากรแบบพาราโบลา ดังนี้

2.2.1 เพิ่มกลไกการแบ่งกลุ่มประชากร

การแบ่งกลุ่มประชากรใน PaDE จะแบ่งประชากรออกเป็น 4 กลุ่ม ($k = 4$ เมื่อ k คือจำนวนกลุ่ม) โดยที่ในแต่ละกลุ่มจะมีค่าพารามิเตอร์ได้แก่ ค่าเฉลี่ย crossover rate (μ_{CR_j}) และค่าความน่าจะเป็นในการเลือกกลุ่มที่ j (p_j) ในช่วงเริ่มต้นจะกำหนดค่า μ_{CR_j} ของทุกกลุ่มให้มีค่าเป็น 0.6 และกำหนดค่า p_j ของทุกกลุ่มให้มีค่าเป็น 0.25 เพื่อให้กลุ่มทั้ง 4 ได้รับโอกาสในการคัดเลือกที่เท่ากันในช่วงแรกของการค้นหาคำตอบ สำหรับการจัดประชากรแต่ละตัวลงไปในกลุ่มทั้งหมด 4 กลุ่ม โดยใช้วิธีการจัดกลุ่มแบบ Stochastic universal selection เนื่องจากวิธีการดังกล่าวจะช่วยลดความลำเอียงในการจัดประชากรลงในกลุ่ม [11] นอกจากนี้ในผลการทดสอบประสิทธิภาพของ PaDE และอัลกอริทึมตัวอื่น พบว่าการที่ PaDE แบ่งประชากรออกเป็น 4 กลุ่มด้วยวิธี Stochastic universal selection ช่วยทำให้ PaDE มีประสิทธิภาพในการค้นหาคำตอบได้ดีกว่าอัลกอริทึมตัวอื่นที่นำมาเปรียบเทียบ [11]

หลังจากการแบ่งกลุ่มประชากร PaDE จะเข้าสู่กระบวนการกลายพันธุ์ การครอสโอเวอร์ และการคัดเลือกประชากรในรุ่นถัดไป หลังจากนั้น PaDE จะคำนวณอัตราส่วนของประชากรที่ยอมรับคำตอบใหม่ในกลุ่มที่ j ดังสมการที่ (5)

$$ratio_j = \begin{cases} \frac{ns_j^2}{ns \cdot (ns_j + nf_j)} & \text{if } ns_j > 0 \\ 0.01 & \text{otherwise} \end{cases},$$

$$ns = \sum_{k=1}^4 ns_j, \quad (5)$$

โดยที่ $ratio_j$ คือ ค่าอัตราส่วนของประชากรที่ยอมรับคำตอบใหม่ในกลุ่มที่ j

ns คือ จำนวนประชากรทั้งหมดที่ยอมรับคำตอบใหม่

ns_j คือ จำนวนประชากรที่ยอมรับคำตอบใหม่ในกลุ่มที่ j

nf_j คือ จำนวนประชากรที่ไม่ยอมรับคำตอบใหม่ในกลุ่มที่ j

พร้อมทั้งทำการปรับค่าความน่าจะเป็นในการคัดเลือกกลุ่มที่ j (p_j) ดังสมการที่ (6)

$$p_j = \frac{ratio_j}{\sum_{j=1}^4 ratio_j} \quad (6)$$

2.2.2 การปรับปรุงกลไกการกลายพันธุ์

ในการปรับปรุงกลไกการกลายพันธุ์ PaDE สร้างมิวแทนเวกเตอร์ V_i^G ของประชากรตัวที่ i ในรอบการคำนวณที่ G ด้วยวิธี DE/current-to-pbest/1 แสดงดังสมการที่ (7)

$$V_i^G = X_i^G + F_i \cdot (X_{pbest}^G - X_i^G) + F_i \cdot (X_{r_1}^G - X_{r_2}^G) \quad (7)$$

โดยที่ F_i คือ ค่าแฟกเตอร์ขยายผลต่างของประชากรตัวที่ i

X_{pbest}^G คือ ประชากรที่สุ่มจากประชากรที่ดีที่สุดจำนวน $p \cdot NP$ ตัว ในรอบการคำนวณที่ G

X_i^G คือ ประชากรตัวที่ i ในรอบการคำนวณที่ G

$X_{r_1}^G$ คือ ประชากรที่สุ่มจากเซตของประชากรรุ่นปัจจุบัน (P^G)

$X_{r_2}^G$ คือ ประชากรที่สุ่มจากเซตของประชากรรุ่นปัจจุบันและประชากรรุ่นก่อนหน้าที่ถูกคัดออก ($P^G \cup A$) A เป็นเซตของประชากรที่ถูกคัดออกก่อนหน้านี้ โดยที่ PaDE มีการปรับปรุงกลไกการกลายพันธุ์ใหม่ในชื่อ Time stamp-based mutation strategy โดยใช้วิธี Time stamp เพื่อกำจัดประชากรในเซต A เมื่อประชากรของเซต A มีค่า time stamp น้อยกว่า 0 โดยค่า Time stamp ของประชากรแต่ละตัวในเซต A คำนวณได้จากสมการที่ (8)

$$t = T^0 - r^d \cdot (n - m) \quad (8)$$

โดยที่ t คือ ค่า Time stamp ในรอบการคำนวณปัจจุบัน

T^0 คือ ค่า Time stamp เริ่มต้นใน PaDE กำหนดค่า $T^0 = 70$

I^d คือ ค่าอัตราการสลายตัวของประชากรในเซต A

โดย PaDE กำหนด $r^d = 0.04$ (เพราะการใช้ r^d เป็น 0.04 ช่วยให้ประชากรที่อยู่ในเซต A ถูกกำจัดออกอย่างช้า ๆ เพื่อให้อัลกอริทึม PaDE มีประชากรในการใช้หาคำตอบได้ดีขึ้น)

n คือ รอบการคำนวณปัจจุบัน

m คือ รอบการคำนวณที่เพิ่มประชากรตัวที่ i เข้ามาในเซต A ตามลำดับ

A คือ เซตของประชากรที่ถูกคัดออกในรอบการคำนวณ โดยในรอบการคำนวณแรกกำหนดให้ $A = \phi$

นอกจากนี้ในสมการที่ (7) PaDE จะคำนวณค่าแฟกเตอร์ขยายผลต่างของประชากรที่ตัวที่ i (F_i) จากการสุ่มด้วยการแจกแจงแบบโคชีที่ค่าเฉลี่ย μ_F และค่าสเกล 0.1 ดังสมการที่ (9)

$$F_i = \text{cauchy}(\mu_F, 0.1) \quad (9)$$

โดยที่ μ_F คือ ค่าเฉลี่ยแฟกเตอร์ขยายผลต่าง

จากสมการที่ (9) หากค่าแฟกเตอร์ขยายผลต่างของประชากรที่ตัวที่ i (F_i) ที่สุ่มได้มีค่าน้อยกว่าหรือเท่ากับ 0 ($F_i \leq 0$) อัลกอริทึม PaDE จะทำการสุ่มค่า F_i ใหม่จนกว่าค่าที่ได้จะมีค่ามากกว่า 0 ($F_i \geq 0$) และหากค่า F_i ที่สุ่มได้มีค่ามากกว่า 1 อัลกอริทึม PaDE จะกำหนดให้ ($F_i = 1$) [11]

สำหรับขั้นตอนการครอสโอเวอร์ PaDE จะใช้วิธีการเดียวกันกับ DE แต่ PaDE จะคำนวณค่าอัตราการครอสโอเวอร์ของประชากรตัวที่ i (CR_i) จากการสุ่มด้วยการแจกแจงแบบปกติที่ค่าเฉลี่ย μ_{CR_j} และค่าความแปรปรวน 0.1 ดังสมการที่ (10)

$$CR_i = \begin{cases} \text{randn}(\mu_{CR_j}, 0.1) & \text{if } \mu_{CR_j} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

โดยที่ CR_i คือ ค่า อัตราการครอสโอเวอร์ของประชากรตัวที่ i

μ_{CR_j} คือ ค่าเฉลี่ยอัตราการครอสโอเวอร์ของกลุ่มที่ j

$randn(\mu_{CR_j}, 0.1)$ คือ ฟังก์ชันสุ่มด้วยการแจกแจงปกติ

จากสมการที่ (10) หากค่าอัตราการครอสโอเวอร์ของประชากรตัวที่ i (CR_i) ที่สุ่มได้มีค่าน้อยกว่าหรือเท่ากับ 0 ($CR_i \leq 0$) อัลกอริทึม PaDE จะกำหนดให้ค่า $CR_i = 0$ และถ้าหากค่า CR_i ที่สุ่มได้มีค่ามากกว่า 1 ($CR_i > 1$) อัลกอริทึม PaDE จะกำหนดให้ค่า $CR_i = 1$ [11]

จากสมการที่ (9) พบว่าการที่ PaDE เลือกใช้ค่าสเกลของการแจกแจงแบบโคชี เท่ากับ 0.1 เพื่อต้องการให้ค่า F_i ที่สุ่มได้จากสมการที่ (9) มีความใกล้เคียงกับค่าเฉลี่ยแฟกเตอร์ขยายผลต่าง μ_F มากที่สุดนอกจากนี้ด้วยการแจกแจงแบบโคชียังช่วยให้ค่า F_i ที่สุ่มได้มีโอกาสสูงที่จะสุ่มได้ค่าที่เหวี่ยงจาก μ_F เพื่อให้ประชากรสามารถค้นพบคำตอบในบริเวณใหม่ได้

นอกจากนี้ในสมการที่ (10) พบว่าการที่ PaDE เลือกใช้ค่าความแปรปรวน 0.1 เนื่องจากค่าความแปรปรวนดังกล่าวทำให้ค่า CR_i ที่สุ่มได้จากค่าเฉลี่ยอัตราการครอสโอเวอร์ของกลุ่มที่ j (μ_{CR_j}) มีความเบี่ยงเบนไปจากค่าเฉลี่ยทั้งสองไม่มากนักทำให้ PaDE มีค่าพารามิเตอร์ที่เหมาะสมต่อการพัฒนาคำตอบของประชากร นอกจากนี้ค่าความแปรปรวน 0.1 ได้รับความนิยมในการนำไปใช้ในอัลกอริทึม DE พัฒนาต่อมาจาก JADE [7] เช่น ADE-DMRM [3] AMECODEs [4] SHADE [8] LSHADE [9] และ LPALMDE [10] เป็นต้น

นอกจากนี้ในการปรับค่าเฉลี่ยแฟกเตอร์ขยายผลต่าง (μ_F) ที่ใช้ในสมการที่ (9) อัลกอริทึม PaDE จะคำนวณจากค่าเฉลี่ยเมอร์แบบถ่วงน้ำหนักโดยเลือกใช้เฉพาะค่าแฟกเตอร์ขยายผลต่างของประชากรตัวที่ i (F_i) ที่ทำ

ให้ประชากรตัวที่ i ยอมรับคำตอบใหม่แสดงดังสมการที่ (11)

$$\mu_F = \begin{cases} \text{mean}_{WL}(S_F) & \text{if } S_F \neq \emptyset \\ \mu_F & \text{otherwise} \end{cases} \quad (11)$$

$$\text{โดยที่ } \text{mean}_{WL}(S_F) = \frac{\sum_{l=1}^{|S_F|} w_l \cdot S_{F_l}^2}{\sum_{l=1}^{|S_F|} w_l \cdot S_{F_l}}$$

$$w_l = \frac{f(X_l - U_l)}{\sum_{l=1}^{|S_F|} f(X_l - U_l)}$$

mean_{WL} คือ ฟังก์ชันหาค่าเฉลี่ยลิเมอร์แบบถ่วงน้ำหนัก

S_F คือ เซตของ F_i ที่ทำให้ประชากรตัวที่ i ยอมรับคำตอบใหม่

w_l คือ ค่าน้ำหนักที่ใช้ในการคำนวณค่าเฉลี่ยลิเมอร์

U_l คือ ประชากรที่มีการยอมรับคำตอบใหม่

X_l คือ ประชากรก่อนการเปลี่ยนแปลงคำตอบ

ในการปรับค่าเฉลี่ยอัตราการครอสโอเวอร์ของกลุ่มที่ j (μ_{CR_j}) ที่ใช้ในสมการที่ (10) อัลกอริทึม PaDE จะเลือกกลุ่มที่มีค่าความน่าจะเป็นในการคัดเลือกกลุ่มที่ j (p_j) น้อยที่สุดมาทำการปรับค่า μ_{CR_j} ด้วยการคำนวณค่าเฉลี่ยเมอร์แบบถ่วงน้ำหนักโดยเลือกใช้เฉพาะอัตราการครอสโอเวอร์ของประชากรตัวที่ i (CR_i) ที่ทำให้ประชากรตัวที่ i ยอมรับคำตอบใหม่แสดงดังสมการที่ (12)

$$\mu_{CR_{idx}} = \begin{cases} \text{mean}_{WL}(S_{CR}), & \text{if } S_{CR} \neq \emptyset \& \max(S_{CR}) > 0 \\ 0, & \text{if } S_{CR} \neq \emptyset \& \mu_{CR_{idx}} = 0 \\ \mu_{CR_{idx}}, & \text{otherwise} \end{cases} \quad (12)$$

$$\text{โดยที่ } w_l = \frac{f(X_l - U_l)}{\sum_{l=1}^{|S_{CR}|} f(X_l - U_l)}$$

$$\text{mean}_{WL}(S_{CR}) = \frac{\sum_{l=1}^{|S_{CR}|} w_l \cdot S_{CR_l}^2}{\sum_{l=1}^{|S_{CR}|} w_l \cdot S_{CR_l}}$$

S_{CR} คือ เซตของ CR_i ที่ทำให้ประชากรตัวที่ i ยอมรับคำตอบใหม่

w_i คือ ค่าน้ำหนักที่ใช้ในการคำนวณค่าเฉลี่ย
ลิเมอรั

แบบเชิงเส้น (เส้นสีดำ) [11]

$$NP = \text{round}\left(\frac{NP_{min} - NP_{init}}{nfe_{max} - NP_{init}} \cdot (nfe - NP_{init})^2 + NP_{init}\right) \quad (13)$$

U_i คือ ประชากรที่มีการยอมรับคำตอบใหม่

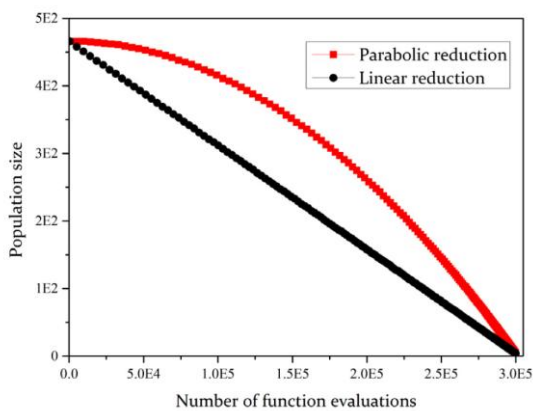
X_i คือ ประชากรก่อนการเปลี่ยนแปลงคำตอบ

idx คือ หมายเลขของกลุ่มที่ถูกเลือกให้ทำการ
ปรับค่า μ_{CR_j} ใหม่

$\mu_{CR_{idx}}$ ค่าเฉลี่ยอัตราการครอสโอเวอร์ของกลุ่มที่
ถูกเลือกปรับค่าพารามิเตอร์

2.2.3 การปรับลดประชากรแบบพาราโบลา

การปรับลดประชากรแบบพาราโบลาใน PaDE ถูกนำเสนอเพื่อแก้ไขปัญหาประชากรลดลงอย่างรวดเร็วใน LSHADE ที่ใช้วิธีการปรับลดขนาดของประชากรแบบเชิงเส้นแสดงดังรูปที่ 2 ซึ่งจะสังเกตได้ว่าวิธีการปรับลดประชากรแบบเชิงเส้นทำให้ประชากรลดลงอย่างรวดเร็วจนอัลกอริทึมมีจำนวนประชากรที่ไม่เพียงพอต่อการค้นหาคำตอบ ในขณะที่วิธีการปรับลดประชากรแบบพาราโบลาที่นำเสนอใน PaDE จะค่อยๆ ลดจำนวนประชากรลงทำให้อัลกอริทึมมีจำนวนประชากรที่เพียงพอต่อการพัฒนาคำตอบ [11] โดย PaDE กำหนดจำนวนประชากรเริ่มต้น (NP_{init}) จำนวนประชากรน้อยที่สุด (NP_{min}) และจำนวนครั้งในการประเมินฟังก์ชันสูงสุด (nfe_{max}) สำหรับการปรับลดประชากรแบบพาราโบลาแสดงดังสมการที่ (1)



รูปที่ 2 จำนวนประชากรของวิธีการปรับลดประชากร
แบบพาราโบลา (เส้นสีแดง) และวิธีปรับลดประชากร

โดยที่ NP คือ จำนวนประชากรในรอบถัดไป

nfe คือ จำนวนครั้งในการประเมินฟังก์ชัน

NP_{init} คือ จำนวนประชากรเริ่มต้น

NP_{min} คือ จำนวนประชากรที่น้อยที่สุด

nfe_{max} คือ จำนวนครั้งในการประเมินฟังก์ชันสูงสุด

3. วิธีการที่นำเสนอ

ในการพัฒนาอัลกอริทึม การเพิ่มประสิทธิภาพอัลกอริทึมผลต่างวิวัฒนาการด้วยวิธี Restart mechanism (RMDE) ที่ปรับปรุงอัลกอริทึม PaDE สองส่วนได้แก่ 1 วิธีการปรับปรุงค่าเฉลี่ยแฟกเตอร์ขยายผลต่างในแต่ละกลุ่ม และ 2 กลไก Restart mechanism โดย RMDE ยังคงใช้วิธี Stochastic universal selection ในการแบ่งประชากรออกเป็น 4 กลุ่มเช่นเดียวกับอัลกอริทึม PaDE เนื่องจากการแบ่งกลุ่มออกเป็น 4 กลุ่มและการใช้วิธี Stochastic universal selection มาจัดกลุ่มประชากรช่วยให้อัลกอริทึมมีประสิทธิภาพในการค้นหาคำตอบได้ดี [11]

3.1 วิธีการปรับค่าเฉลี่ยแฟกเตอร์ขยายผลต่างในแต่ละกลุ่ม

ใน PaDE แบ่งประชากรออกเป็น 4 กลุ่มโดยแต่ละกลุ่มใช้ค่าเฉลี่ยแฟกเตอร์ขยายผลต่าง (μ_F) เป็นค่าเดียวกันทุกกลุ่มแต่ในอัลกอริทึม RMDE ใช้ค่าเฉลี่ยแฟกเตอร์ขยายผลต่างแยกกันในแต่ละกลุ่ม ดังสมการที่ (14)

$$\mu_{F_{idx}} = \begin{cases} mean_{WL}(S_F), & \text{if } S_F \neq \phi \text{ \& } \max(S_F) > 0 \\ 0, & \text{if } S_F \neq \phi \text{ \& } \mu_{F_{idx}} = 0 \\ \mu_{F_{idx}}, & \text{otherwise} \end{cases} \quad (14)$$

$$\text{โดยที่ } mean_{WL}(S_F) = \frac{\sum_{l=1}^{|S_F|} w_l \cdot S_{F_l}^2}{\sum_{l=1}^{|S_F|} w_l \cdot S_{F_l}}$$

$$w_l = \frac{std(X_l - U_l)}{\sum_{l=1}^{|S_F|} std(X_l - U_l)}$$

w_l คือ ค่าน้ำหนักที่ใช้ในการคำนวณค่าเฉลี่ยลิเมอร์

U_l คือ ประชากรที่มีการยอมรับคำตอบใหม่

X_l คือ ประชากรก่อนการเปลี่ยนแปลงคำตอบ

S_F คือ เซตของค่า F_l ที่ทำให้ประชากรตัวที่ i ยอมรับคำตอบใหม่

idx คือ หมายเลขของกลุ่มที่ถูกเลือกให้ปรับค่าพารามิเตอร์ μ_{F_j}

$\mu_{F_{idx}}$ คือ ค่าเฉลี่ยแฟกเตอร์ขยายผลต่างของกลุ่มที่ถูกเลือกปรับค่าพารามิเตอร์

μ_{F_j} คือ ค่าเฉลี่ยแฟกเตอร์ขยายผลต่างของกลุ่มที่ j

สำหรับการปรับค่าเฉลี่ยอัตราการครอสโอเวอร์ของกลุ่มที่ j (μ_{CR_j}) ของ RMDE นั้นจะใช้วิธีเดียวกันกับ PaDE ดังสมการที่ (12) โดย RMDE จะทำการปรับค่าพารามิเตอร์ μ_{CR_j} และ μ_{F_j} ของแต่ละกลุ่มวนไปเรื่อย ๆ แทนการใช้วิธีการเดิมที่ปรับค่าพารามิเตอร์เฉพาะกลุ่มที่มีค่าความน่าจะเป็นในการคัดเลือกกลุ่มที่ j (p_j) น้อยที่สุด โดยวิธีการปรับค่าพารามิเตอร์ที่นำเสนอใน RMDE ทำให้แต่ละกลุ่มมีค่าพารามิเตอร์ที่เหมาะสมต่อการพัฒนาคำตอบ นอกจากนี้ในช่วงเริ่มต้นของการค้นหาคำตอบ RMDE จะกำหนดค่าเฉลี่ยอัตราการครอสโอเวอร์ของกลุ่มที่ j (μ_{CR_j}) และกำหนดค่าเฉลี่ยแฟกเตอร์ขยายผลต่างของกลุ่มที่ j (μ_{F_j}) ของทุกกลุ่มเป็น 0.5 และจำกัดประชากรส่วนเกินในเซต A ของ RMDE จะกำจัดประชากรส่วนเกินในเซต A โดยลบประชากรที่มีค่า time stamp (τ) ที่น้อยที่สุดจำนวน $|A| - 1.6 \cdot NP$ ตัวและการคำนวณค่า τ ใช้วิธีการเดียวกันกับ PaDE ที่แสดงในสมการที่ (8)

3.2 กลไก Restart mechanism

สำหรับกลไก Restart mechanism ใน RMDE ผู้วิจัยได้นำมาจากอัลกอริทึม AMECODEs [4] โดยกลไกดังกล่าวแบ่งออกเป็นสองส่วนได้แก่ 1. กลไกการเพิ่มความหลากหลายของประชากร และ 2. กลไกช่วยให้ประชากรรู้เข้า สำหรับกลไกการเพิ่มความหลากหลายของประชากรจะถูกใช้เมื่อค่าความหลากหลายของประชากร

(DP^c) น้อยกว่าหรือเท่ากับ 0.01 โดยที่

$$DP^G = \frac{1}{NP} \cdot \sqrt{\sum_{i=1}^{NP} \left\| X_i - \frac{1}{NP} \sum_{j=1}^{NP} X_j \right\|^2}$$

และสร้างประชากรรุ่นถัดไปดังสมการที่ (15) และกลไกช่วยให้ประชากรลู่เข้าจะถูกเลือกใช้เมื่อประชากรไม่สามารถพัฒนาคำตอบได้เป็นจำนวน D รอบโดยสร้าง

$$x_{i,j}^G = \begin{cases} lb_j + rand(0,1) \times (ub_j - lb_j) & \text{if } j \in J \\ x_{i,j}^G & \text{otherwise} \end{cases} \quad (15)$$

$$u_{i,j}^G = \begin{cases} x_{pbest,j}^G + rand(0,1) \times (x_{r_1,j}^G - x_{r_2,j}^G) + 0.1 \times rand(0,1) & \text{if } j \in J \\ x_{pbest,j}^G & \text{otherwise} \end{cases} \quad (16)$$

โดยที่ $x_{i,j}^G$ คือ ตัวแปรตัดสินใจที่ j ของประชากรที่ i

J คือ เซตของตัวแปรตัดสินใจในประชากรที่ i ที่ถูกสุ่ม โดยที่ $J \subset [1,2,...,D]$

$rand(0,1)$ คือ ฟังก์ชันสุ่มด้วยการแจกแจงเอกรูป ในช่วง $[0,1]$

ประชากรรุ่นถัดไปแสดงดังสมการที่ (16) สำหรับการทำงานของกลไก Restart mechanism แสดงดังรูปที่ 3 อัลกอริทึมการทำงานของ RMDE ที่นำเสนอในงานวิจัยแสดงดังรูปที่ 4 และผังงานของอัลกอริทึม RMDE แสดงดังรูปที่ 5

Algorithm 2: Restart mechanism

```

1  Input: Population  $P^G$ , Fitness value  $f$ ,  $counter_i$ ,  $nfe$ 
2  Output: New Population  $P^G$ 
3  Calculate  $DP^G$  from Eq 16
4  if  $DP^G \leq 0.01$ 
5      for  $i$  from 1 to  $NP$  do
6          if  $counter_i > D$  and  $i$  is not best population
7              Generate  $X_i^G$  from Eq 15
8              Evaluate fitness value of new  $X_i^G$ 
9          end
10     end
11 else
12     for  $i$  from 1 to  $NP$  do
13         if  $counter_i > D$  and  $i$  is not best population
14             Generate  $U_i^G$  from Eq 16
15             if  $f(U_i^G) \leq f(X_i^G)$ 
16                  $counter_i = 0$ 
17                  $X_i^G = U_i^G$ 
18             else
19                  $counter_i = counter_i + 1$ 
20             end
21         end
22     end
23 end
24 return New Population  $P^G$ 
    
```

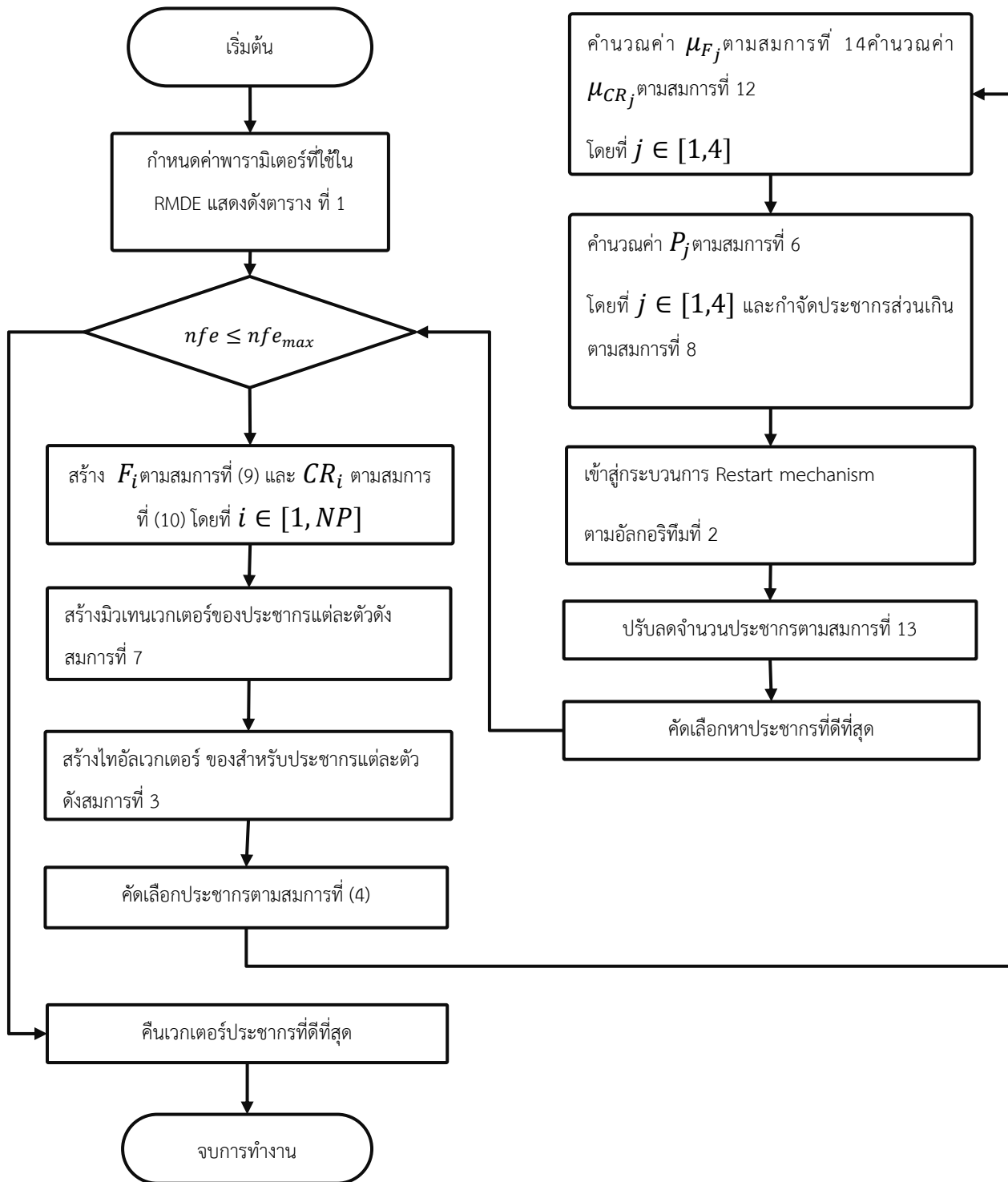
รูปที่ 3 ขั้นตอนการทำงานของกลไก Restart mechanism

Algorithm 3: RMDE Algorithm

```

1  Input: Initialization population ( $P^G$ ),  $G = 0$ 
2  Setting all parameter value from table 1.
3  Output: best population  $P^{best}$  and fitness value  $f(P^{best})$ 
4  for  $nfe \leq nfe_{max}$  do
5      Divided population ( $P^G$ ) into  $k$  group in Algorithm1
6      Calculate fitness value of population ( $f(P^G)$ )
7      Generate set of best population ( $pbest$ ) with size  $p \cdot NP$ 
8      for  $i$  from 1 to  $NP$  do
9          Generate  $F_i$  from Eq 9
10         Generate  $CR_i$  from Eq 10
11         Generate mutant vector ( $V_i^G$ ) from Eq 7
12         Generate trial vector ( $U_i^G$ ) from Eq 3
13         Checking boundary in each decision variable of  $U_i^G$ 
14         Calculate fitness value of  $U_i^G$  as  $f(U_i^G)$ 
15         if  $f(U_i^G) \leq f(X_i^G)$ 
16             Add  $F_i$  in set  $S_F$  and Add  $CR_i$  in set  $S_{CR}$ 
17             Add  $X_i^G$  in set  $A$ 
18              $counter_i = 0$ 
19         else
20              $counter_i = counter_i + 1$ 
21         end
22     end
23      $nfe = nfe + NP$ ,  $G = G + 1$ 
24     Update  $\mu_{F_k}$  from Eq 14
25     Update  $\mu_{CR_k}$  from Eq 12
26     Clear set  $S_F = \phi$  and clear  $S_{CR} = \phi$ 
27     Adjust archive set  $A$  from Eq 8
28     Restart mechanism in Algorithm2
29     Calculate  $P_i$  for each group from Eq 6
30     Population size reduction from Eq 13
31 end
32 return best individual from  $P^{best}$  and fitness value  $f(P^{best})$ 
    
```

รูปที่ 4 ขั้นตอนการทำงานของ RMDE



รูปที่ 5 ผลงานของอัลกอริทึม RMDE

ตารางที่ 1 ค่าพารามิเตอร์ที่ใช้ในการทดลองของแต่ละอัลกอริทึม

Algorithm name	Parameter setting
jSO	$NP = 18D \square 4, \mu_{CR} = 0.8, \mu_F = 0.3, k = 5, p = 0.11, r^{arc} = 2.6$
MPADE	$NP = 100, \mu_{CR} = 0.5, \mu_F = 0.5, p = 0.05, c = 0.1, MF = 100$
JADE	$NP = 100, \mu_{CR} = 0.5, \mu_F = 0.5, p = 0.05, c = 0.1$
SHADE	$NP = 100, H = 100, \mu_{CR_h} = 0.5, \mu_{F_h} = 0.5, p = 0.11, c = 0.1$
PaDE	$NP = 25 \log(D) \sqrt{D} \square 4, \mu_{CR_j} = 0.6, \mu_F = 0.8, k = 4, p = 0.11$
RMDE	$NP = 25 \log(D) \sqrt{D} \square 4, \mu_{CR_j} = 0.5, \mu_{F_j} = 0.5, k = 4, p = 0.11, \varepsilon = 0.001$

4. ผลการทดลอง

ในการทดสอบประสิทธิภาพของ RMDE ที่นำเสนอประกอบไปด้วย 2 ส่วนได้แก่ 1.การทดสอบประสิทธิภาพของ RMDE และ 2 การประยุกต์ใช้ RMDE ในปัญหาการจัดการสินค้าคงคลัง

4.1 การทดสอบประสิทธิภาพของ RMDE

ผู้วิจัยทำการเปรียบเทียบ RMDE กับ อัลกอริทึม DE ที่เป็น state-of-the-art ได้แก่ JADE SHADE jSO [12] MPADE และ PaDE โดยผู้วิจัยได้พัฒนาอัลกอริทึม RMDE และ state-of-the art ตัวอื่นที่นำมาเปรียบเทียบกับภาษา Python เวอร์ชัน 3.12 และทดสอบประสิทธิภาพของอัลกอริทึมทั้งหมดในคอมพิวเตอร์ AMD Ryzen Pro 4650g แรม 64 GB SSD 1TB สำหรับค่าพารามิเตอร์ของแต่ละอัลกอริทึมแสดงดังตารางที่ 1 สำหรับการทดสอบในส่วนที่ 1 ผู้วิจัยได้เลือกใช้ชุดปัญหาทดสอบ CEC2017 [13] ในการทดสอบประสิทธิภาพของอัลกอริทึมที่นำเสนอ โดยที่ชุดปัญหาทดสอบดังกล่าวมีจำนวนฟังก์ชันทั้งหมด 29 ฟังก์ชันแบ่งเป็น Unimodal ($f_1 - f_2$) 2 ฟังก์ชัน Multimodal ($f_3 - f_9$) 7 ฟังก์ชัน Hybrid ($f_{10} - f_{19}$) 10 ฟังก์ชัน และ Composition ($f_{20} - f_{29}$) 10 ฟังก์ชัน โดยทำการทดลองในปัญหาขนาด 10 และ 30 ตัวแปรตัดสินใจ และกำหนดค่า maximum number of function evaluation (nfe_{max}) ให้มีค่าเป็น $nfe_{max} = 10000 \cdot D$ เมื่อ D คือจำนวนตัวแปรตัดสินใจและในแต่ละปัญหามีการทำซ้ำทั้งหมด 30 ซ้ำ แสดงผลการทดลองในรูปแบบค่าเฉลี่ยของความผิดพลาดดังตารางที่ 2 และ ตารางที่ 3 นอกจากนี้

กำหนดให้สัญลักษณ์ “W/L/E” ที่อยู่ในส่วนท้ายของตารางมีความหมายว่า “W (Winning) แสดงถึงอัลกอริทึมที่มีประสิทธิภาพเหนือกว่า” “L (Losing) แสดงถึงอัลกอริทึมที่มีประสิทธิภาพด้อยกว่า” และ “E (Equivalent) แสดงถึงอัลกอริทึมที่มีประสิทธิภาพเทียบเท่า” เมื่อเปรียบเทียบกับอัลกอริทึม RMDE

จากตารางที่ 2 พบว่าในปัญหาขนาด 10 ตัวแปรตัดสินใจ RMDE มีประสิทธิภาพในการค้นหาคำตอบแยกกว่าอัลกอริทึม jSO SHADE และ MPADE เกินกว่าครึ่งหนึ่งของจำนวนฟังก์ชันทั้งหมดในชุดปัญหาทดสอบทั้งหมดส่วนอัลกอริทึม JADE และ PaDE มีประสิทธิภาพในการค้นหาคำตอบใกล้เคียงกับ RMDE เมื่อขนาดของปัญหาเพิ่มขึ้นเป็น 30 ตัวแปรตัดสินใจแสดงผลการทดลองดังตารางที่ 3 พบว่าอัลกอริทึม RMDE มีประสิทธิภาพในการค้นหาคำตอบโดยรวมเพิ่มขึ้น โดย RMDE ค้นหาคำตอบดีกว่า JADE และ MPADE ได้ดีขึ้น จากเดิม 12 และ 5 ฟังก์ชันในปัญหาขนาด 10 ตัวแปรตัดสินใจ เพิ่มขึ้นเป็น 16 และ 18 ฟังก์ชันในปัญหาขนาด 30 ตัวแปรตัดสินใจ ส่วนอัลกอริทึม RMDE เมื่อเทียบกับ jSO และ SHADE พบว่า RMDE มีประสิทธิภาพในการค้นหาคำตอบใกล้เคียงกัน ถึงแม้ว่าอัลกอริทึม SHADE จะมีประสิทธิภาพในการค้นหาคำตอบดีกว่า RMDE ในภาพรวมแต่ในฟังก์ชันประเภท Hybrid อัลกอริทึม RMDE ค้นหาคำตอบได้ดีกว่า SHADE เป็นจำนวน 7 จาก 10 ฟังก์ชัน นอกจากนี้ในปัญหาขนาด 30 ตัวแปรตัดสินใจเมื่อพิจารณาฟังก์ชันประเภท Hybrid พบว่าอัลกอริทึม RMDE มีประสิทธิภาพในการค้นหาคำตอบดีกว่าอัลกอริทึม SHADE JADE และ MPADE เป็นจำนวนเกินกว่าครึ่งหนึ่งของจำนวนฟังก์ชัน Hybrid ทั้งหมด ส่วนปัญหา

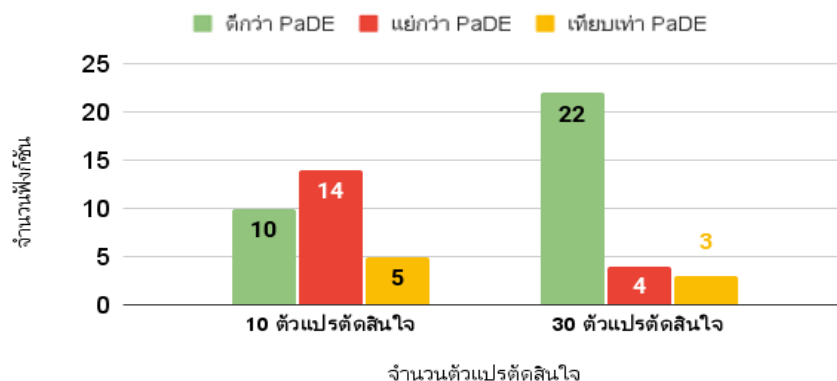
ฟังก์ชันประเภท Unimodal และ Multimodal พบว่า อัลกอริทึม RMDE มีประสิทธิภาพในการค้นหาคำตอบดีกว่า jSO MPAD และ PaDE เป็นจำนวนเกินกว่าครึ่งหนึ่งของ จำนวนฟังก์ชัน Unimodal และ Multimodal รวมกัน

เมื่อพิจารณาจำนวนฟังก์ชันที่ RMDE สามารถ ค้นหาคำตอบได้ ดีกว่า แย่กว่า และเทียบเท่ากับอัลกอริทึม PaDE ที่เป็นอัลกอริทึมต้นแบบของ RMDE แสดงรูปที่ 6 พบว่าในปัญหาขนาด 10 ตัวแปรตัดสินใจ RMDE มีจำนวน ฟังก์ชันค้นหาคำตอบได้ดีกว่า PaDE เป็นจำนวน 10 ฟังก์ชัน ในขณะที่ PaDE มีจำนวนฟังก์ชันที่ค้นหาคำตอบได้ดีกว่า RMDE เป็นจำนวน 14 ฟังก์ชันซึ่งเป็นจำนวนที่มีความ แตกต่างกันไม่มากนัก แต่เมื่อขนาดของปัญหาเพิ่มขึ้นเป็น 30 ตัวแปรตัดสินใจพบว่า RMDE มีจำนวนฟังก์ชันที่ค้นหา คำตอบได้ดีกว่า PaDE สูงถึง 22 ฟังก์ชันในขณะที่ PaDE มี จำนวนฟังก์ชันที่ค้นหาคำตอบได้ดีกว่า RMDE เพียง 4 ฟังก์ชันเท่านั้น สำหรับฟังก์ชันประเภท Hybrid เมื่อ พิจารณาจำนวนฟังก์ชันที่ RMDE ค้นหาคำตอบได้ ดีกว่า แย่กว่า และเทียบเท่า กับอัลกอริทึม PaDE แสดงดังรูปที่ 7 พบว่าในปัญหาขนาด 10 ตัวแปรตัดสินใจ RMDE มี ประสิทธิภาพในการค้นหาคำตอบใกล้เคียงกับ PaDE โดยมี จำนวนฟังก์ชันที่ RMDE ค้นหาคำตอบได้ดีกว่า PaDE เป็น จำนวน 5 ฟังก์ชันและมีจำนวนฟังก์ชันที่ PaDE ค้นหา คำตอบได้ดีกว่า RMDE เป็นจำนวน 4 ฟังก์ชัน เมื่อขนาด

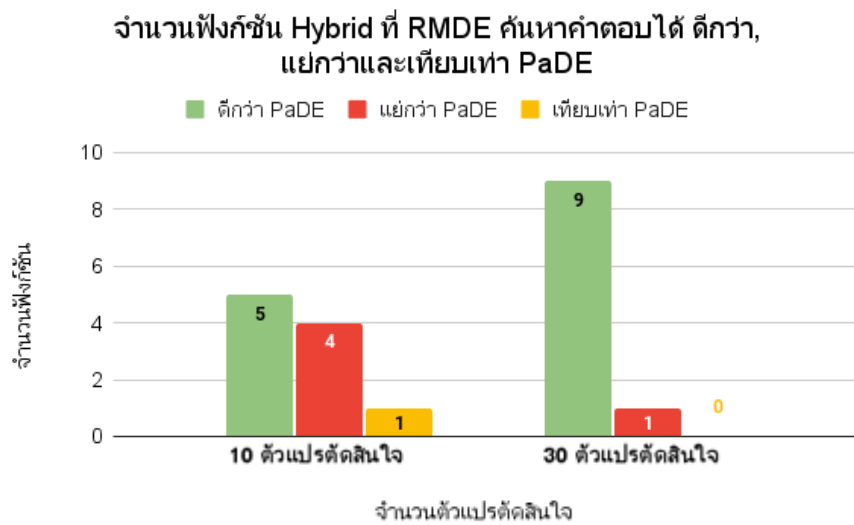
ของปัญหาเพิ่มขึ้นเป็น 30 ตัวแปรตัดสินใจ อัลกอริทึม RMDE มีประสิทธิภาพในการค้นหาคำตอบดีกว่า PaDE เพิ่มขึ้น โดย RMDE ค้นหาคำตอบได้ดีกว่า PaDE เป็น จำนวน 9 ฟังก์ชัน นอกจากนี้ในฟังก์ชัน Composition เมื่อ พิจารณาจำนวนฟังก์ชันที่ RMDE ค้นหาคำตอบได้ ดีกว่า แย่กว่า และเทียบเท่า กับอัลกอริทึม PaDE แสดงดังรูปที่ 8 พบว่าปัญหาขนาด 10 ตัวแปรตัดสินใจประสิทธิภาพในการ ค้นหาคำตอบของ PaDE ในฟังก์ชัน Composition ดีกว่า RMDE เป็นจำนวน 7 ฟังก์ชัน เมื่อขนาดของปัญหาเพิ่มขึ้น เป็น 30 ตัวแปรตัดสินใจ RMDE มีประสิทธิภาพในการ ค้นหาคำตอบดีกว่า PaDE เป็นจำนวน 8 ฟังก์ชัน

จากผลการทดลองที่แสดงดังรูปที่ 6-8 ชี้ให้เห็นว่า วิธีการที่นำเสนอใน RMDE ได้แก่การกำหนดให้แต่ละกลุ่มมี ค่าพารามิเตอร์ μ_{F_k} แยกกัน รวมถึงปรับปรุงวิธีการปรับ ค่าพารามิเตอร์ μ_{F_k} และ μ_{CR_k} พร้อมกับนำกลไก Restart mechanism เข้ามาใช้ในอัลกอริทึม ถึงแม้ว่าใน ปัญหาขนาดเล็ก RMDE และ PaDE จะมีประสิทธิภาพใน การค้นหาคำตอบที่ไม่แตกต่างกันมากนักแต่เมื่อขนาดของ ปัญหาเพิ่มขึ้นอัลกอริทึม RMDE มีประสิทธิภาพในการ ค้นหาคำตอบได้ดีกว่า PaDE ที่เป็นอัลกอริทึมต้นแบบ

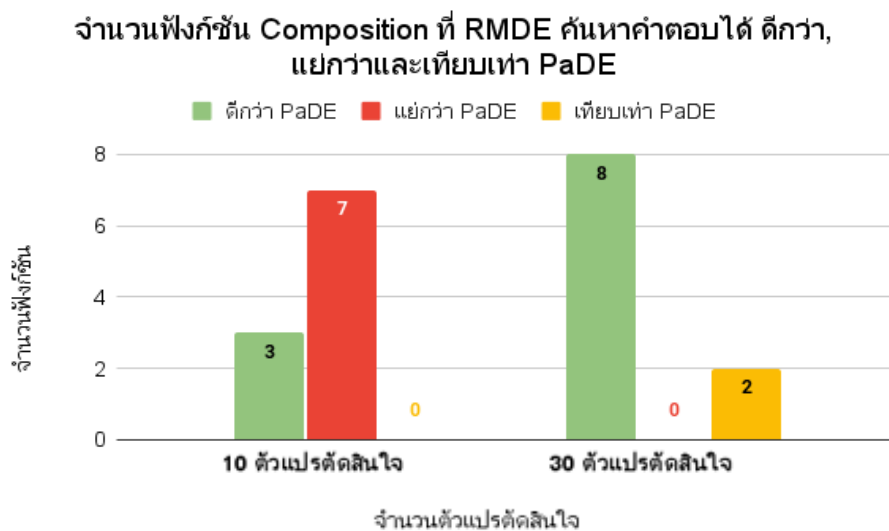
จำนวนฟังก์ชันในชุดปัญหาทดสอบ CEC2017 ที่ RMDE ค้นหาคำตอบได้ ดีกว่า, แย่กว่า และเทียบเท่า PaDE



รูปที่ 6 จำนวนฟังก์ชันในชุดปัญหาทดสอบ CEC2017 ที่ RMDE ค้นหาคำตอบได้ ดีกว่า แย่กว่า และ เทียบเท่า เมื่อ เปรียบเทียบกับ PaDE ในปัญหาขนาด 10 และ 30 ตัวแปรตัดสินใจ



รูปที่ 7 จำนวนฟังก์ชันประเภท Hybrid ในชุดปัญหาทดสอบ CEC2017 ที่ RMDE ค้นหาคำตอบได้ ดีกว่า แย่กว่า และเทียบเท่า เมื่อเปรียบเทียบกับ PaDE ในปัญหาขนาด 10 และ 30 ตัวแปรตัดสินใจ



รูปที่ 8 จำนวนฟังก์ชันประเภท Composition ในชุดปัญหาทดสอบ CEC2017 ที่ RMDE ค้นหาคำตอบได้ ดีกว่า แย่กว่า และเทียบเท่า เมื่อเปรียบเทียบกับ PaDE ในปัญหาขนาด 10 และ 30 ตัวแปรตัดสินใจ

ตารางที่ 2 ค่าเฉลี่ยของความผิดพลาดในปัญหาทดสอบ CEC2017 ขนาด 10 ตัวแปรตัดสินใจ

No	jSO	JADE	SHADE	MPADE	PaDE	RMDE
F1	<u>0.00e+00(0.00e+00)</u>	1.89e-15(8.12e-15)	<u>0.00e+00(0.00e+00)</u>	3.32e-15(1.16e-14)	1.42e-15(4.34e-15)	<u>0.00e+00(0.00e+00)</u>
F2	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>
F3	<u>9.90e+00(0.00e+00)</u>	<u>9.90e+00(0.00e+00)</u>	<u>9.90e+00(0.00e+00)</u>	<u>9.90e+00(0.00e+00)</u>	<u>9.90e+00(0.00e+00)</u>	<u>9.90e+00(0.00e+00)</u>
F4	3.18e+00(1.21e+00)	9.82e+00(1.97e+00)	3.14e+00(6.77e-01)	<u>3.02e+00(1.58e+00)</u>	4.20e+00(2.86e+00)	4.25e+00(2.57e+00)
F5	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>
F6	6.25e+00(2.11e+00)	1.58e+01(3.07e+00)	8.51e+00(1.37e+00)	<u>5.85e+00(2.25e+00)</u>	7.92e+00(4.57e+00)	6.47e+00(4.20e+00)
F7	1.73e+03(3.05e+00)	<u>1.31e+03(3.77e+00)</u>	1.73e+03(2.69e+00)	1.73e+03(1.82e+00)	<u>1.31e+03(4.06e+00)</u>	1.73e+03(4.95e+00)
F8	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>
F9	5.00e+01(3.56e+01)	5.39e+02(1.40e+02)	8.71e+01(6.50e+01)	1.08e+02(1.15e+02)	<u>3.06e+01(4.97e+01)</u>	5.87e+01(1.13e+02)
F10	4.10e+00(3.43e-01)	<u>3.96e+00(8.80e-03)</u>	<u>3.96e+00(3.05e-04)</u>	<u>3.96e+00(5.82e-04)</u>	4.49e+00(9.30e-01)	4.23e+00(8.61e-01)
F11	<u>3.82e-05(0.00e+00)</u>	4.51e+01(1.08e+02)	2.74e+01(9.52e+01)	<u>3.82e-05(0.00e+00)</u>	9.68e+01(1.83e+02)	1.21e+02(2.34e+02)
F12	2.39e+01(3.07e+00)	<u>2.35e+01(3.48e+00)</u>	2.43e+01(2.83e+00)	2.56e+01(2.71e+00)	2.60e+01(4.44e+00)	2.50e+01(2.87e+00)
F13	1.65e+01(7.86e+00)	2.50e+01(2.95e+00)	<u>6.32e+00(7.67e+00)</u>	1.49e+01(6.04e+00)	1.80e+01(8.34e+00)	1.56e+01(9.26e+00)
F14	3.01e+00(9.84e-02)	3.01e+00(4.74e-01)	3.02e+00(1.14e-01)	3.05e+00(4.12e-02)	3.25e+00(7.08e-01)	<u>2.85e+00(9.73e-01)</u>
F15	2.47e+01(6.62e+01)	1.03e+01(3.97e+01)	2.47e+01(6.63e+01)	<u>3.01e+00(1.55e-02)</u>	1.03e+01(3.97e+01)	1.76e+01(5.50e+01)
F16	1.74e+01(8.47e+00)	2.05e+01(4.19e+00)	<u>9.79e+00(6.45e+00)</u>	1.11e+01(5.46e+00)	1.23e+01(9.03e+00)	1.62e+01(7.88e+00)
F17	<u>1.74e-01(1.94e-01)</u>	1.26e+01(4.64e+00)	1.78e-01(1.82e-01)	3.12e-01(1.60e-01)	1.44e+01(9.14e+00)	1.12e+01(9.96e+00)
F18	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>	<u>0.00e+00(0.00e+00)</u>
F19	1.64e+00(6.11e-01)	1.26e+01(5.12e+00)	4.55e+00(2.67e+00)	3.05e+00(1.57e+00)	<u>1.55e+00(1.63e+00)</u>	2.89e+00(5.38e+00)
F20	<u>1.00e-04(1.15e-13)</u>	<u>1.00e-04(1.13e-13)</u>	4.21e-01(1.09e+00)	<u>1.00e-04(1.13e-13)</u>	4.21e-01(1.09e+00)	7.40e-01(1.36e+00)
F21	2.54e+02(2.48e+02)	6.06e+02(3.52e+02)	2.70e+02(1.75e+02)	<u>9.12e+00(4.99e+01)</u>	4.14e+02(1.81e+02)	3.32e+02(2.36e+02)
F22	3.20e+02(2.28e+00)	3.27e+02(2.57e+00)	3.20e+02(1.98e+00)	<u>3.00e+02(7.88e+01)</u>	3.23e+02(3.43e+00)	3.24e+02(3.40e+00)
F23	3.77e+02(2.19e+00)	3.76e+02(1.41e+00)	<u>3.75e+02(6.56e-01)</u>	3.76e+02(1.44e+00)	3.76e+02(2.45e+00)	3.78e+02(3.28e+00)
F24	<u>4.10e+02(0.00e+00)</u>	<u>4.10e+02(0.00e+00)</u>	<u>4.10e+02(0.00e+00)</u>	<u>4.10e+02(0.00e+00)</u>	<u>4.10e+02(1.89e+00)</u>	4.11e+02(3.57e+00)
F25	2.52e+03(1.67e+03)	3.63e+03(1.29e+03)	3.35e+03(1.65e+03)	<u>4.10e+02(5.55e+01)</u>	3.57e+03(1.16e+03)	3.72e+03(1.25e+03)
F26	8.04e-01(1.17e-01)	7.31e-01(8.73e-02)	7.26e-01(7.95e-02)	<u>4.07e-02(1.13e-02)</u>	1.19e+00(9.68e-02)	2.03e+00(3.69e-01)
F27	8.82e-01(4.83e+00)	<u>2.00e-04(2.12e-13)</u>	<u>2.00e-04(1.72e-13)</u>	1.63e-01(2.43e-01)	2.04e-04(2.07e-05)	9.39e+00(1.18e+01)
F28	9.91e+03(2.45e+03)	1.02e+04(2.36e+03)	1.10e+04(3.14e+03)	<u>7.51e+03(1.10e+03)</u>	1.14e+04(3.64e+03)	1.13e+04(3.46e+03)
F29	1.56e+04(8.14e-03)	1.85e+04(1.63e+03)	1.56e+04(1.27e-02)	<u>1.50e+04(7.99e+02)</u>	1.57e+04(2.01e+02)	1.56e+04(3.02e+00)
W/L/E	17/4/8	12/12/5	16/5/8	18/5/6	14/10/5	-/-/-

ตารางที่ 3 ค่าเฉลี่ยของความผิดพลาดในปัญหาทดสอบ CEC2017 ขนาด 30 ตัวแปรตัดสินใจ

No	jSO	JADE	SHADE	MPADE	PaDE	PIMDE
F1	4.17e-14(2.30e-14)	<u>0.00e+00(0.00e+00)</u>	3.32e-15(6.11e-15)	5.28e-11(2.18e-10)	4.26e-08(1.76e-07)	1.85e-14(6.62e-15)
F2	1.99e-14(1.32e-14)	<u>1.89e-15(7.21e-15)</u>	<u>1.89e-15(7.21e-15)</u>	1.80e-14(1.39e-14)	2.84e-14(1.06e-14)	1.71e-14(1.42e-14)
F3	<u>2.97e+01(2.56e-14)</u>	<u>2.97e+01(1.81e-14)</u>	<u>2.97e+01(1.81e-14)</u>	<u>2.97e+01(1.92e-14)</u>	<u>2.97e+01(3.46e-14)</u>	<u>2.97e+01(2.56e-14)</u>
F4	4.56e+01(6.67e+00)	9.87e+01(8.45e+00)	<u>2.03e+01(2.94e+00)</u>	3.29e+01(8.92e+00)	3.19e+01(1.10e+01)	2.18e+01(5.65e+00)
F5	3.79e-15(1.44e-14)	1.89e-15(1.04e-14)	<u>0.00e+00(0.00e+00)</u>	1.33e-14(2.45e-14)	2.46e-14(3.23e-14)	9.47e-15(2.15e-14)
F6	7.94e+01(6.45e+00)	1.46e+02(9.03e+00)	5.59e+01(5.30e+00)	<u>3.18e+01(9.78e+00)</u>	6.32e+01(2.13e+01)	5.42e+01(8.44e+00)
F7	3.14e+03(2.09e+01)	1.79e+03(2.36e+01)	3.13e+03(2.39e+01)	3.11e+03(2.02e+01)	<u>1.77e+03(1.22e+01)</u>	3.12e+03(1.43e+01)
F8	3.79e-14(5.45e-14)	2.98e-03(1.63e-02)	<u>3.41e-14(5.30e-14)</u>	1.19e-02(5.12e-02)	2.43e-11(1.30e-10)	1.49e-02(3.39e-02)
F9	2.30e+03(2.08e+02)	5.29e+03(3.01e+02)	1.85e+03(2.98e+02)	2.06e+03(4.47e+02)	<u>1.45e+03(2.20e+02)</u>	1.48e+03(2.21e+02)
F10	<u>1.56e+01(1.46e+00)</u>	7.45e+01(5.47e+01)	5.87e+01(1.01e+02)	8.97e+01(8.84e+01)	5.00e+01(5.39e+01)	2.43e+01(8.75e+00)
F11	<u>1.46e+01(5.51e+01)</u>	4.07e+03(2.71e+03)	1.45e+02(1.44e+02)	4.03e+02(1.11e+03)	7.15e+02(1.56e+03)	1.53e+02(1.96e+02)
F12	1.41e+02(6.77e+00)	3.35e+02(3.19e+02)	1.02e+02(9.61e+00)	1.28e+02(3.64e+01)	4.27e+02(9.74e+02)	<u>9.99e+01(7.55e+00)</u>
F13	4.83e+01(1.97e+01)	8.33e+01(1.19e+01)	3.11e+01(3.09e+00)	5.73e+01(2.19e+01)	4.04e+01(1.29e+01)	<u>2.91e+01(8.72e+00)</u>
F14	<u>1.26e+01(2.88e+00)</u>	5.45e+02(1.00e+03)	1.44e+02(1.55e+02)	3.31e+02(3.50e+02)	1.01e+03(1.96e+03)	3.33e+01(1.25e+01)
F15	<u>1.07e+01(5.07e-01)</u>	1.11e+01(1.59e+00)	1.93e+01(4.31e+01)	1.25e+01(6.64e-01)	1.13e+01(1.62e+00)	1.21e+01(9.26e-01)
F16	5.52e+01(1.48e+01)	1.29e+02(8.86e+01)	1.29e+02(2.06e+02)	<u>4.81e+01(4.13e+01)</u>	1.39e+02(1.50e+02)	1.33e+02(1.57e+02)
F17	<u>2.24e+01(9.59e-01)</u>	9.50e+01(1.42e+02)	2.68e+01(3.78e+00)	1.21e+02(2.29e+02)	5.25e+02(1.07e+03)	9.47e+01(3.88e+02)
F18	6.35e+00(2.74e+00)	2.25e+01(2.61e+00)	9.23e-01(1.63e+00)	2.91e+00(2.74e+00)	3.45e+00(7.03e+00)	<u>6.33e-01(8.65e-01)</u>
F19	5.23e+01(1.10e+01)	4.57e+01(4.31e+00)	2.15e+01(5.42e+00)	<u>7.68e+00(4.97e+00)</u>	1.96e+01(7.99e+00)	1.70e+01(6.78e+00)
F20	<u>1.00e-04(8.30e-14)</u>	<u>1.00e-04(0.00e+00)</u>	<u>1.00e-04(0.00e+00)</u>	6.80e-01(2.07e+00)	9.47e-01(2.46e+00)	4.68e-01(1.78e+00)
F21	3.11e+03(1.24e+03)	5.92e+03(3.03e+02)	2.50e+03(2.81e+02)	2.77e+03(5.61e+02)	2.05e+03(4.55e+02)	<u>2.04e+03(2.26e+02)</u>
F22	3.99e+02(5.97e+00)	4.66e+02(1.02e+01)	<u>3.83e+02(5.52e+00)</u>	3.96e+02(7.89e+00)	3.98e+02(7.30e+00)	3.95e+02(8.98e+00)
F23	<u>1.26e+03(6.66e+00)</u>	<u>1.26e+03(8.81e+00)</u>	<u>1.26e+03(7.78e+00)</u>	1.27e+03(1.32e+01)	1.45e+03(8.61e+01)	<u>1.26e+03(7.33e+00)</u>
F24	<u>4.30e+02(1.73e-13)</u>	<u>4.30e+02(0.00e+00)</u>	<u>4.30e+02(0.00e+00)</u>	<u>4.30e+02(1.57e-13)</u>	<u>4.30e+02(3.52e-13)</u>	<u>4.30e+02(2.02e-13)</u>
F25	<u>4.74e+03(9.70e-13)</u>	<u>4.74e+03(6.09e-13)</u>	<u>4.74e+03(1.15e-12)</u>	<u>4.74e+03(9.70e-13)</u>	5.04e+03(8.68e+01)	<u>4.74e+03(9.09e-13)</u>
F26	2.12e+00(5.80e-02)	2.19e+00(1.68e-01)	2.12e+00(9.56e-02)	<u>2.09e+00(3.26e-02)</u>	5.19e+00(8.60e-01)	2.14e+00(1.03e-01)
F27	<u>1.48e+01(1.72e+01)</u>	2.62e+01(1.47e+01)	2.62e+01(1.47e+01)	3.41e+01(2.12e-13)	3.41e+01(2.31e-13)	3.41e+01(2.29e-13)
F28	<u>4.10e+06(1.39e+00)</u>	4.28e+06(3.08e+05)	<u>4.10e+06(9.78e-01)</u>	4.11e+06(3.98e+04)	7.65e+06(1.71e+07)	4.12e+06(6.12e+04)
F29	<u>1.91e+04(2.61e+03)</u>	2.32e+05(5.98e+04)	2.72e+04(6.97e+03)	4.78e+04(1.41e+04)	6.74e+04(4.31e+04)	3.75e+04(9.30e+03)
W/L/E	13/12/4	9/16/4	14/11/4	7/18/4	4/22/3	-/-/-

4.2 การประยุกต์ใช้ RMDE ในปัญหาการจัดการคลังสินค้า

ในหัวข้อนี้ผู้วิจัยนำ RMDE มาประยุกต์ใช้กับปัญหา A constrained multi-item EOQ inventory model for reusable items [14] ที่สินค้าแต่ละชนิดจะถูกสั่งซื้อด้วยวิธีการแบบประหยัด (Economic Order Quantity: EOQ) ที่มีวัตถุประสงค์ในการหาค่าใช้จ่ายในการจัดการคลังสินค้า (Total cost: TC) ที่น้อยที่สุด (Minimized Total cost: $MinTC$) โดยแบ่งขนาดของปัญหาออกเป็น 3 ขนาดได้แก่ ปัญหาขนาดเล็กที่มีจำนวนสินค้า 5 ชนิด, ปัญหาขนาดกลางที่มีจำนวนสินค้า 25 ชนิด, และปัญหาขนาดใหญ่ที่มีจำนวนสินค้า 50 ชนิด โดยที่สินค้าแต่ละชนิดประกอบไปด้วยจำนวนตัวแปรตัดสินใจทั้งหมด 2 ตัวทำให้ปัญหามีขนาดเล็ก ปัญหาขนาดกลาง และปัญหาขนาดใหญ่ ประกอบไปด้วยจำนวนตัวแปรตัดสินใจทั้งหมด 10 50 และ 100 ตัว นอกจากนี้ในปัญหาแต่ละขนาดแบ่งออกเป็นปัญหาย่อยทั้งหมด 10 ปัญหา สำหรับตัวแปรตัดสินใจและค่าพารามิเตอร์ที่ใช้ในปัญหาดังกล่าวแสดงดังตารางที่ 4 และ ตารางที่ 5 สำหรับตัวแบบทางคณิตศาสตร์ที่ประกอบไปด้วยฟังก์ชันวัตถุประสงค์ในการหา TC ดัง

สมการที่ (17) และกำหนดสมการเงื่อนไขดังสมการที่ (18)-(22) ในการแก้ไขปัญหาดังกล่าวที่อยู่ในรูปแบบ Constraint optimization ผู้วิจัยได้ใช้วิธี Penalty function [14] เพื่อเปลี่ยนปัญหาที่อยู่ในรูปแบบ Constraint optimization ให้อยู่ในปัญหารูปแบบ Unconstraint optimization ทำการทดสอบซ้ำทั้งหมด 10 รอบและกำหนดค่า maximum number of function evaluation ที่ $2000 \cdot P$ โดยที่ P คือจำนวนสินค้าในปัญหาแต่ละขนาด แสดงผลการค้นหาคำตอบของ ปัญหาแต่ละขนาดในรูปแบบของค่า Error value ดังตารางที่ 6 จากการทดลองพบว่าในปัญหาขนาดเล็กประสิทธิภาพในการค้นหาคำตอบของ RMDE และ PaDE มีประสิทธิภาพในการค้นหาคำตอบที่ใกล้เคียงกัน อย่างไรก็ตามเมื่อปัญหามีขนาดใหญ่ขึ้นพบว่า RMDE มีประสิทธิภาพในการค้นหาคำตอบมากขึ้น โดยในปัญหาขนาดกลาง RMDE สามารถค้นหาคำตอบได้ดีกว่า PaDE ถึง 8 จาก 10 ปัญหาย่อยและในปัญหาขนาดใหญ่ RMDE สามารถหาคำตอบได้ดีกว่า PaDE ในทุกปัญหาเนื่องจากกลไก Restart mechanism ใน RMDE ช่วยให้ RMDE มีความหลากหลายของประชากรมากขึ้นและทำให้ RMDE ค้นพบคำตอบที่ใกล้เคียงกับ optimal solution ได้ดีกว่า PaDE

ตารางที่ 4 ตัวแปรตัดสินใจในปัญหาการจัดการสินค้าคงคลัง

ชื่อตัวแปรตัดสินใจ	ความหมาย
q_i	ปริมาณการสั่งซื้อและปริมาณการใช้สินค้าชนิดที่ i โดยที่ $q_i \geq 0$
p_i	อัตราส่วนของปริมาณสินค้าชนิดที่ i ที่อยู่ในคลังสินค้าต่อปริมาณการสั่งซื้อสินค้าชนิดที่ i โดยที่ $p_i \geq 0$

ตารางที่ 5 ค่าพารามิเตอร์ที่ใช้ในปัญหาการจัดการสินค้าคงคลัง

ชื่อพารามิเตอร์	ความหมาย
D_i	Demand ของสินค้าชนิดที่ i โดยที่ $D_i \geq 0$
m_i	จำนวนสินค้าชนิดที่ i ที่มีการนำกลับมาใช้ใหม่และต้องซื้อเข้ามาเพิ่มเติม โดยที่ $m_i \geq 0$
C_i	ต้นทุนการสั่งซื้อสินค้าต่อหน่วยของสินค้าชนิดที่ i โดยที่ $C_i \geq 0$
Au_i	ต้นทุนในการสั่งซื้อสินค้าชนิดที่ i โดยที่ $Au_i \geq 0$
Ar_i	ต้นทุนในการดำเนินงานในการสั่งซื้อสินค้าชนิดที่ i มาเพิ่มเติม โดยที่ $Ar_i \geq 0$
hu_i	ต้นทุนในการจัดเก็บสินค้าชนิดที่ i ที่ยังมีการใช้งาน โดยที่ $hu_i \geq 0$

ชื่อพารามิเตอร์	ความหมาย
hr_i	ต้นทุนในการจัดเก็บสินค้าชนิดที่ i ที่ซื้อเข้ามาเพิ่มเติม โดยที่ $hr_i \geq 0$
r_i	ต้นทุนสินค้าชนิดที่ i ที่ซื้อเข้ามาเพิ่มเติม โดยที่ $r_i \geq 0$
f_i	ปริมาณพื้นที่ที่ใช้ในการจัดเก็บสินค้าชนิดที่ i โดยที่ $f_i \geq 0$
HR	ต้นทุนสูงสุดในการจัดเก็บสินค้าที่ซื้อเข้ามาเพิ่มเติม โดยที่ $HR \geq 0$
HU	ต้นทุนสูงสุดในการจัดเก็บสินค้าที่กำลังใช้งาน โดยที่ $HU \geq 0$
WQ	พื้นที่คลังสินค้าทั้งหมดที่ใช้ในการจัดเก็บสินค้าที่กำลังใช้งาน โดยที่ $WQ > 0$
WB	พื้นที่คลังสินค้าทั้งหมดที่ใช้ในการจัดเก็บสินค้าที่ซื้อเข้ามาเพิ่มเติม โดยที่ $WB > 0$
BU	งบประมาณในการจัดการคลังสินค้า โดยที่ $BU \geq 0$
TC	ค่าใช้จ่ายทั้งหมดในการจัดการคลังสินค้า (Total cost) โดยที่ $TC \geq 0$

$$\begin{aligned}
 Min \ TC(p_i, q_i) = & \sum_{i=1}^n C_i \frac{D_i}{(m_i+1)} + \sum_{i=1}^n Au_i \frac{D_i}{(m_i+1)p_i q_i} \\
 & + \sum_{i=1}^n Ar_i \frac{D_i}{q_i} \left(\frac{m_i}{m_i+1} \right) + \sum_{i=1}^n r_i D_i \left(\frac{m_i}{m_i+1} \right) \\
 & + \sum_{i=1}^n hu_i \left(\frac{p_i q_i}{2} \right) + \sum_{i=1}^n hr_i \left(\frac{m_i}{m_i+1} \right) \frac{q_i}{2_i}
 \end{aligned} \tag{17}$$

$$\sum_{i=1}^n C_i p_i q_i \leq BU \tag{18}$$

$$\sum_{i=1}^n f_i p_i q_i \leq WQ \tag{19}$$

$$\sum_{i=1}^n f_i q_i \leq WB \tag{20}$$

$$\sum_{i=1}^n hu_i \left(\frac{p_i q_i}{2} \right) \leq HU \tag{21}$$

$$\sum_{i=1}^n hr_i \left(\frac{m_i}{m_i+1} \right) \frac{q_i}{2_i} \leq HR \tag{22}$$

ตารางที่ 6 ค่าเฉลี่ยของความผิดพลาดที่ได้จากอัลกอริทึม PaDE และ RMDE ในปัญหาขนาดเล็ก ขนาดกลาง และ ขนาดใหญ่

No	5 ชนิดสินค้า		25 ชนิดสินค้า		50 ชนิดสินค้า	
	PaDE	RMDE	PaDE	RMDE	PaDE	RMDE
F1	1.61E-01	1.61E-01	8.86E+01	3.96E+01	9.95E+02	3.55E+02
F2	2.80E-01	2.80E-01	0.00E+00	0.00E+00	1.16E+03	3.49E+02

No	5 ชนิดสินค้า		25 ชนิดสินค้า		50 ชนิดสินค้า	
	PaDE	RMDE	PaDE	RMDE	PaDE	RMDE
F3	0.00E+00	0.00E+00	1.02E+02	2.99E+01	9.87E+02	3.47E+02
F4	0.00E+00	0.00E+00	1.72E+02	4.72E+01	1.08E+03	3.12E+02
F5	3.56E-01	3.56E-01	1.08E+02	3.55E+01	9.83E+02	4.13E+02
F6	7.22E-03	7.22E-03	1.06E+02	3.75E+01	9.14E+02	2.74E+02
F7	1.76E-01	1.76E-01	1.03E+02	3.68E+01	9.21E+02	3.81E+02
F8	1.79E-01	1.79E-01	1.25E+02	4.87E+01	1.08E+03	3.57E+02
F9	2.26E-01	2.26E-01	1.15E+02	3.05E+01	9.18E+02	2.98E+02
F10	3.31E-01	3.31E-01	0.00E+00	0.00E+00	8.68E+02	3.28E+02
W/L/E	0/0/10	-/-/-	0/8/2	-/-/-	0/10/0	-/-/-

5. สรุปผลการทดลอง

งานวิจัยนี้นำเสนอการปรับปรุงประสิทธิภาพอัลกอริทึมผลต่างวิวัฒนาการด้วยวิธี Restart mechanism ที่ปรับปรุงมาจากอัลกอริทึม PaDE โดยเพิ่มกลไก Restart mechanism และ ปรับปรุงวิธีการคำนวณค่าเฉลี่ยแฟกเตอร์ขยายผลต่าง จากการทดสอบด้วยชุดปัญหาทดสอบ CEC2017 ที่ขนาดของปัญหา 10 ตัวแปรตัดสินใจพบว่า ประสิทธิภาพในการค้นหาคำตอบของอัลกอริทึม jSO SHADE และ MPADDE ดีกว่าอัลกอริทึม RMDE ส่วนอัลกอริทึม JADE และ PaDE มีประสิทธิภาพในการค้นหาคำตอบใกล้เคียงกับ RMDE แต่เมื่อเพิ่มขนาดของปัญหาเพิ่มขึ้นเป็น 30 ตัวแปรตัดสินใจพบว่า RMDE มีประสิทธิภาพในการค้นหาคำตอบโดยรวมเพิ่มขึ้นอย่างเห็นได้ชัด โดยเมื่อเปรียบเทียบกับอัลกอริทึม JADE MPADDE และ PaDE พบว่า RMDE มีจำนวนฟังก์ชันที่ค้นหาคำตอบได้ ดีกว่าเป็นจำนวนมาก และเมื่อเปรียบเทียบกับ RMDE กับอัลกอริทึม jSO และ SHADE พบว่ามีประสิทธิภาพในการค้นหาคำตอบใกล้เคียงกัน เมื่อพิจารณาผลการทดลองแยกตามประเภทของฟังก์ชัน พบว่า RMDE มีประสิทธิภาพในการค้นหาคำตอบมากที่สุด โดยให้ผลการค้นหาคำตอบ ดีกว่าอัลกอริทึม JADE MPADDE SHADE และ PaDE มากกว่าครึ่งหนึ่งของจำนวนฟังก์ชันประเภท Hybrid ทั้งหมด

สำหรับการประยุกต์ใช้ RMDE ในการแก้ไขปัญหาคลังสินค้าพบว่าในปัญหาขนาดเล็กที่มีจำนวนตัวแปรตัดสินใจ 10 ตัวแปร ประสิทธิภาพในการค้นหาคำตอบของ RMDE และ PaDE ในการแก้ไขปัญหาที่มีความแตกต่างกันไม่มากนัก เมื่อจำนวนตัวแปรตัดสินใจของปัญหาเพิ่มขึ้นเป็น 50 ตัวแปรตัดสินใจในปัญหาขนาดกลาง และ 100 ตัวแปรตัดสินใจในปัญหาขนาดใหญ่ พบว่ากลไก Restart mechanism ช่วยให้ RMDE มีประสิทธิภาพในการค้นหาคำตอบจนใกล้เคียงกับค่า optimal value ได้มากกว่า PaDE

ทั้งนี้จากผลการทดลองการเปรียบเทียบประสิทธิภาพและการนำไปประยุกต์ใช้จะเห็นได้ว่าอัลกอริทึม RMDE ประสิทธิภาพในการค้นหาคำตอบที่ดีในปัญหาที่มีจำนวนตัวแปรตัดสินใจจำนวนมาก เช่นปัญหาการเก็บเกี่ยว [15] ปัญหาการเพาะปลูก [16] ปัญหาเส้นทางการขนส่งสินค้า [17-18] เป็นต้น

6. กิตติกรรมประกาศ

งานวิจัยนี้ได้รับทุนสนับสนุนจากคณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์

7. เอกสารอ้างอิง

- [1] R. Storn and K. Price, "Differential evolution – A simple and efficient Heuristic for global

- optimization over continuous spaces,” *Journal of Global Optimization.*, vol. 11, pp. 341-359, 1997.
- [2] J. Wetweeraopong and P. Puphasuk, “An improved differential evolution algorithm with a restart technique to solve systems of nonlinear equations,” *An International Journal of Optimization and Control: Theories & Applications.*, vol. 10, no. 1, pp. 118-136, 2020.
- [3] X. Lin and Z. Meng, “An adaptative differential evolution with enhanced diversity and restart mechanism,” *Expert Systems with Applications.*, vol. 249, 2024.
- [4] L. Cui *et al.*, “Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism,” *Information Sciences.*, vol. 422, pp. 122-143, 2018.
- [5] G. Liu, Y. Li, X. Nie and H. Zheng, “A novel clustering-based differential evolution with 2 multi-parent crossovers for global optimization,” *Applied Soft Computing.*, vol. 12, no. 2, pp. 663-681, 2012.
- [6] J. Brest, S. Greiner, B. Boskovic, M. Mernik and V. Zumer, “Self-Adapting control parameters in differential evolution: a comparative study on numerical benchmark problems,” *IEEE Transactions on Evolutionary Computation.*, vol. 10, no. 6, pp. 646-657, 2006.
- [7] J. Zhang and A. C. Sanderson, “JADE: adaptive differential evolution with optional external archive,” *IEEE Transactions on Evolutionary Computation.*, vol. 13, no. 5, pp. 945-958, 2009.
- [8] R. Tanabe and A. Fukunaga, “Success-History Based Parameter Adaptation for Differential Evolution,” in *2013 IEEE Congress on Evolutionary Computation*, Cancun, Mexico, 2013, pp. 71-78.
- [9] R. Tanabe and A. Fukunaga, “Improving the Search Performance of SHADE Using Linear Population Size Reduction,” in *2014 IEEE Congress on Evolutionary Computation*, Beijing, China, 2014, pp. 1658-1665.
- [10] Z. Meng, J.-S. Pan and L. Kong, “Parameters with adaptive learning mechanism (PALM) for the enhancement of differential evolution,” *Knowledge-Based Systems.*, vol. 141, pp. 92-112, 2018.
- [11] Z. Meng, J.-S. Pan and K.-K. Tseng, “PaDE: an enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization,” *Knowledge-Based Systems.*, vol. 168, pp. 80-99, 2019.
- [12] J. Brest, M. S. Maučec and B. Bošković, “Single Objective Real-Parameter Optimization: Algorithm jSO,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*, Donostia, Spain, 2017, pp. 1311-1318.
- [13] N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang and a. B. Y. Qu, “Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization,” Singapore: Nanyang Technological University, 2016.
- [14] A. Fallahi, E. A. Bani and S. T. A. Niaki, “A constrained multi-item EOQ inventory model for reusable items: Reinforcement learning-based differential evolution and particle swarm optimization,” *Expert Systems with Applications.*, vol. 207, no. c, 2022.

- [15] J. E. Gómez-Lagos, M. C. González-Araya, W. E. Soto-Silva and M. M. Rivera-Moraga, “Optimizing tactical harvest planning for multiple fruit orchards using a metaheuristic modeling approach,” *European Journal of Operational Research.*, vol. 290, no. 1, pp. 297-312, 2021.
- [16] C. Wang et al., “Optimizing spatial food crops planting structure under water-energy-food-carbon emissions nexus constraints,” *Agricultural Water Management.*, vol. 317, 2025.
- [17] S. Wang, L. Wang and Y. Pi, “A hybrid differential evolution algorithm for a stochastic location-inventory-delivery problem with joint replenishment,” *Data Science and Management.*, vol. 5, no. 3, pp. 124-136, 2022.
- [18] L. Peng and S. Wang, “A Q-learning based arithmetic optimization algorithm for a multi-warehouse joint replenishment and delivery problem,” *Applied Soft Computing.*, vol. 178, p. 113307, 2025.