การพัฒนาวิธีการหาค่าที่ดีที่สุดแบบพาร์ทิเคิลสวอมออฟทิไมเซชั่นด้วยการเลียนแบบโครงสร้างการเรียนรู้ทางสังคมแบบหลากหลาย

PARTICLE SWARM OPTIMIZATION ALGORITHMS WITH MULTIPLE SOCIAL LEARNING STRUCTURES

พิศุทธิ์ พงศ์ชัยฤกษ์
Pisut Pongchairerks

ภาควิชาวิศวกรรมการผลิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีไทย-ญี่ปุ่น
Production Engineering Program, Faculty of Engineering,
Thai-Nichi Institute of Technology.

บทคัดย่อ

วิธีการหาค่าที่ดีที่สุดแบบพาร์ทิเคิลสวอมออฟทิไมเซชั่น (PSO) จัดเป็นวิธีการค้นหาแบบสุ่มโดยอาศัยจุดเริ่มต้นหลายจุดซึ่งอาศัยหลักการที่ประยุกต์มาจากหลักการทางจิตวิทยาสังคม PSO แสดงให้เราเห็นถึงพฤติกรรมทางสังคมได้ดีพอๆ กับการใช้งานในการแก้ไขปัญหาของวิศวกรรม โดยเฉพาะอย่างยิ่ง PSO ช่วยในการหาค่าที่ดีที่สุดในปัญหายากๆ ได้อย่างมีประสิทธิภาพและรวดเร็ว ในช่วงหลายปีที่ผ่านมา นักวิจัยมีความพยายามที่จะเพิ่มขีดความสามารถของ PSO โดยการปรับปรุงและพัฒนา PSO โดยการเลียนแบบโครงสร้างการเรียนรู้ทางสังคมแบบต่างๆ บทความวิชาการฉบับนี้ได้ทำการสรุป PSO ในรูปแบบต่างๆ ที่มีการเลียนแบบโครงสร้างการเรียนรู้ทางสังคมที่แตกต่างกัน

คำสำคัญ: พาร์ทิเคิลสวอมออฟทิไมเซชั่น, PSO, อัลกอลิทึ่ม

Abstract

Particle swarm optimization (PSO) is a population-based stochastic search algorithm based on social-psychological principles. It provides insights into social behaviors as well as contributions to engineering applications, widely used as a powerful optimization technique. In recent years, the researchers have attempted to enhance the performance of PSO by improving its social learning structures. This paper summarizes several PSO versions based on different social learning structures.

Keywords: Particle swarm optimization, PSO, Algorithm

บทนำ

วิธีการหาค่าที่ดีที่สุดแบบพาร์ทิเคิลสวอมออฟทิไมเซชั่น (PSO) ได้ถูกนำเสนอเป็นครั้งแรกโดย Kennedy และ Eberhart ในปี 1995 [1] วิธีการหาค่าที่ดีที่สุดแบบ PSO นี้จัดอยู่ในหมวดหมู่ของวิธีการค้นหาแบบสุ่มโดยอาศัยความสัมพันธ์กันของคำตอบเริ่มต้นหลายคำตอบ (Population–based Stochastic Search Algorithm) ซึ่งอาศัยหลักการการเลียนแบบพฤติกรรมทางสังคมของสัตว์สังคม เช่น ฝูงนก หรือ ฝูงปลา เป็นต้น PSO แสดงให้เห็นถึงพฤติกรรมทางสังคมได้ดีพอๆ กับการนำมันมาประยุกต์ใช้แก้ปัญหาทางวิศวกรรม โดยเฉพาะปัญหาในการหาค่าที่ดีที่สุด หลักการทำงานของ PSO ซึ่งจำลองมาจากการใช้ชีวิตของฝูงสัตว์ในธรรมชาตินั้นเป็นดังต่อไปนี้ ฝูงสัตว์ (Swarm) ใน PSO แท้จริงแล้วคือ กลุ่มของพาร์ทิเคิลในพื้นที่หลายมิติ ซึ่งแต่ละพาร์ทิเคิลจะต้องเก็บข้อมูลเวคเตอร์สองเวคเตอร์คือ เวคเตอร์ระบุตำแหน่งและเวคเตอร์ระบุความเร็วในการบิน เราสมมุติให้ฝูงพาร์ทิเคิลบินอยู่ในพื้นที่ค้นหาหลายมิติ ตัวพาร์ทิเคิลแต่ละตัวในฝูงจะต้องจำตำแหน่งที่ดีที่สุดของตัวมันเอง และตำแหน่งที่ดีที่สุดของเพื่อนร่วมฝูง ความหมายของคำว่า *ตำแหน่งที่ดีที่สุด* ในที่นี้ หมายถึง ตำแหน่งที่มีค่าคำตอบที่ดีที่สุดนั่นเอง สมาชิกในฝูงจะทำการสื่อสารข้อมูลของตำแหน่งกับสมาชิกตัวอื่นและทำการเปลี่ยนความเร็วและเคลื่อนที่ไปยังตำแหน่งใหม่เพื่อให้เข้าใกล้ตำแหน่งที่ดีที่สุดมากยิ่งขึ้น โดยทั่วไปแล้ว การสื่อสารเกิดขึ้นได้โดยผ่านทางสองวิธีคือ

– ผ่านทางตำแหน่งที่ดีที่สุดสากล (Global Best Position) ซึ่งคือ ตำแหน่งที่ดีที่สุดที่ฝูงทั้งฝูงค้นพบ

– ผ่านทางตำแหน่งที่ดีที่สุดเฉพาะกลุ่ม (Neighbors' best Position) ซึ่งหมายความว่าพาร์ทิเคิลแต่ละตัวจะสื่อสารกับเฉพาะพาร์ทิเคิลตัวอื่นบางตัวเท่านั้น

ในวิธี PSO แบบมาตรฐาน [1–2] ตำแหน่งที่ดีที่สุดสากลถูกใช้เป็นแหล่งข้อมูลเดียวที่ใช้ติดต่อกันในฝูง การใช้ตำแหน่งที่ดีที่สุดสากลนำมาสู่การทำให้พาร์ทิเคิลในฝูงบินมากระจุกตัวรวมกันอย่างรวดเร็ว ดังนั้นจึงทำให้ฝูงพาร์ทิเคิลติดกับอยู่ในจุดที่ดีที่สุดเฉพาะบริเวณ (Local Optimum) ด้วยจุดอ่อนข้อนี้ของ PSO แบบมาตรฐานนี้ ทำให้นักวิจัยทำการปรับปรุงวิธี PSO แบบใหม่ๆ โดยการใช้ตำแหน่งที่ดีที่สุดเฉพาะกลุ่มแบบต่างๆ [3–4] แทนการใช้ตำแหน่งที่ดีที่สุดสากล

สัญลักษณ์และคำจำกัดความที่ใช้ในวิธี PSO

เพื่ออธิบายวิธีการหาค่าที่ดีที่สุดแบบ PSO เราจำเป็นจะต้องรู้สัญลักษณ์และคำจำกัดความของ PSO ดังต่อไปนี้

– พาร์ทิเคิล (Particle) คือ สมาชิกตัวหนึ่งในประชากร (Population) โดยพาร์ทิเคิลหนึ่งตัวประกอบด้วย ตำแหน่ง (Position) และความเร็ว (Velocity) ตัวพาร์ทิเคิลรู้ตำแหน่งปัจจุบันของมันและรู้ค่าคำตอบของตำแหน่งนั้นๆ ตัวพาร์ทิเคิลรู้ตำแหน่งที่ดีที่สุดที่ตัวมันเองเคยหาเจอ ที่เรียกว่า Personal Best Position รู้ตำแหน่งที่ดีที่สุดเฉพาะกลุ่ม และรู้ค่าคำตอบของตำแหน่งนั้นๆ

– ประชากร หรือ ฝูง (Swarm) คือ เซตของกลุ่มพาร์ทิเคิล K ตัว ตั้งแต่ตัวที่ 1 ถึงตัวที่ K

– ตำแหน่งของพาร์ทิเคิลตัวที่ i ที่การวนซ้ำครั้งที่ t ถูกเขียนแทนด้วย $X_i(t)$ โดยตำแหน่งดังกล่าวประกอบด้วยมิติ D มิติ คือ $X_i(t) = (x_{i1}(t),\ldots, x_{id}(t),\ldots,x_{iD}(t))$ โดยที่ $x_{id}(t)$ คือ ค่าของตำแหน่งของมิติที่ d ของตัวพาร์ทิเคิลตัวที่ i แต่ละตำแหน่ง $X_i(t)$ สามารถแปลงเป็นคำตอบ (Solution) ของปัญหาทางคณิตศาสตร์ โดยค่านอกตำแหน่งมีถูกจำกัดในขอบเขต $[X_{\min}, X_{\max}]$

– ค่าความเหมะสม (Fitness Value): $f(X_i(t))$ คือ ค่าของคำตอบที่แปลงมาจากตำแหน่ง

$X_i(t)$ โดยจะถูกเรียกว่าค่าความเหมาะสมของตำแหน่ง $X_i(t)$

- ความเร็ว (Velocity): $V_i(t)$ แทนค่าความเร็วของตัวพาร์ทิเคิลตัวที่ i ที่การวนซ้ำครั้งที่ t คือ ถูกเขียนแทนด้วยค่าเวคเตอร์ที่มีมิติ D มิติ คือ $V_i(t) = (v_{i1}(t),\ldots, v_{id}(t),\ldots, v_{iD}(t))$ โดย vid(t) คือค่าของความเร็วที่มิติที่ d ของตัวพาร์ทิเคิลตัวที่ i ที่การวนซ้ำครั้งที่ t และ $V_i(t + 1)$ คืออัตราเร็วที่ตัวพาร์ทิเคิลตัวที่ i จะเคลื่อนที่จากตำแหน่ง $X_i(t)$ ไปตำแหน่ง $X_i(t + 1)$

- ความเร็วสูงสุด (Maximum Velocity): $V_{\max}$ คือ ขีดจำกัดของความเร็ว โดยแต่ละ $v_{id}(t)$ ไม่สามารถมีค่าออกนอกช่วง $[-V_{\max}, V_{\max}]$

- น้ำหนักแรงเฉื่อย (Inertia Weight): $w(t)$ คือพารามิเตอร์ที่ใช้ในการควบคุมผลกระทบของความเร็วที่การวนซ้ำก่อนหน้า ที่จะมีผลต่อความเร็วในการวนซ้ำปัจจุบันของตัวพาร์ทิเคิลทั้ง K ตัว

- ตำแหน่งที่ดีที่สุดส่วนตัว (Personal Best Position): P_i คือ ตำแหน่งที่ถูกพบโดยตัวพาร์ทิเคิลตัวที่ i ที่มีค่าความเหมาะสมที่ดีที่สุด โดยเขียนแทนด้วย $P_i = (p_{i1},\ldots, p_{id},\ldots, p_{iD})$

- ตำแหน่งที่ดีที่สุดสากล (Global Best Position): P_g เป็นสัญลักษณ์ที่ใช้แทนตำแหน่งที่ดีที่สุดสากล และแทนด้วยเวคเตอร์ของมิติ D มิติ คือ $P_g = (p_{g1},\ldots, p_{gd},\ldots, p_{gD})$ ตำแหน่งที่ดีที่สุดสากล คือ ตำแหน่งที่ดีที่สุดที่ถูกพบโดยฝูง

- ตำแหน่งที่ดีที่สุดเฉพาะบริเวณ (Local Best Position): P_{li} ใช้เขียนแทนตำแหน่งที่ดีที่สุดเฉพาะบริเวณของตัวพาร์ทิเคิลตัวที่ i และเขียนแทนด้วยเวคเตอร์ขนาด D มิติ คือ $P_{li} = (p_{li1},\ldots, p_{lid},\ldots, p_{liD})$ โดยที่ p_{lid} คือ ตำแหน่งที่ดีที่สุดเฉพาะบริเวณของตัวพาร์ทิเคิลตัวที่ i ในมิติที่ d โดย P_{li} คือตำแหน่งที่ดีที่สุดที่ถูกพบโดยกลุ่มพาร์ทิเคิลจำนวน k ตัวที่มีลำดับหมายเลขติดกัน ยกตัวอย่างเช่น สมมุติเรากำหนดให้ จำนวนพาร์ทิเคิลในฝูงทั้งหมดเป็น K=5 และจำนวนพาร์ทิเคิลในกลุ่มที่เรียงตามลำดับหมายเลขเป็น k=3 ดังนั้น กลุ่มของตัวพาร์ทิเคิลตัวที่ 1 ประกอบด้วย พาร์ทิเคิลตัวที่ 5 ตัวที่ 1 และตัวที่ 2, กลุ่มของตัวพาร์ทิเคิลตัวที่ 2 ประกอบด้วย พาร์ทิเคิลตัวที่ 1–3, กลุ่มของตัวพาร์ทิเคิลตัวที่ 3 ประกอบด้วย พาร์ทิเคิลตัวที่ 2–4, กลุ่มของตัวพาร์ทิเคิลตัวที่ 4 ประกอบด้วย พาร์ทิเคิลตัวที่ 3–5 และ กลุ่มของตัวพาร์ทิเคิลตัวที่ 5 ประกอบด้วย พาร์ทิเคิลตัวที่ 4 ตัวที่ 5 และตัวที่ 1

- ตำแหน่งที่ดีที่สุดของกลุ่มที่ตำแหน่งใกล้กัน (Near Neighbor Best Position): P_{ni} แทนด้วยตำแหน่งที่ดีที่สุดของกลุ่มที่ตำแหน่งใกล้กัน [4] ของพาร์ทิเคิลตัวที่ i และถูกแทนด้วยเวคเตอร์ขนาด D มิติ ดังนี้คือ สำหรับปัญหาหาค่าต่ำสุด $P_{ni} = (p_{ni1},\ldots, p_{nid},\ldots, p_{niD})$ ซึ่ง $p_{nid} = p_{jd}$ ที่มีค่าที่สูงที่สุดของ FDR (j, i, d) ที่แสดงไว้ในสมการที่ 1 สำหรับปัญหาหาค่าสูงสุด ต่ำสุด $P_{ni} = (p_{ni1},\ldots, p_{nid},\ldots, p_{niD})$ ซึ่ง $p_{nid} = p_{jd}$ ที่มีค่าที่สูงที่สุดของ FDR (j, i, d)

$$FDR(j,i,d) = \frac{f(X_i) - f(P_j)}{|p_{jd} - x_{id}|} \text{ เมื่อ } i \neq j \quad (1)$$

เมื่อ FDR ถูกเรียกว่า Fitness-Distance-Ratio และ p_{jd} คือ ตำแหน่งที่ดีที่สุดส่วนตัวของตัวพาร์ทิเคิลตัวที่ j ที่มิติที่ d

- ค่าคงที่อัตราเร่ง (Acceleration Constant) แต่ละประเภทของตำแหน่งที่ดีที่สุด อันได้แก่ ตำแหน่งที่ดีที่สุดส่วนตัว, ตำแหน่งที่ดีที่สุดสากล, ตำแหน่งที่ดีที่สุดเฉพาะบริเวณ และตำแหน่งที่ดีที่สุดของกลุ่มที่ตำแหน่งใกล้กัน จะมีค่าคงที่อัตราเร่งเป็นของมันเอง โดยค่าคงที่อัตราเร่งคือ พารามิเตอร์ที่มีผลกระทบต่อค่าความเร็ว โดย c_p คือ ค่าคงที่อัตราเร่งของตำแหน่งที่ดีที่สุดส่วนตัว, c_g คือ ค่าคงที่อัตราเร่งของตำแหน่งที่ดีที่สุดสากล, c_l คือ ค่าคงที่อัตราเร่งของตำแหน่งที่ดีที่สุดเฉพาะบริเวณ และ c_n คือ ค่าคงที่อัตราเร่งของตำแหน่งที่ดีที่สุดของกลุ่มที่ตำแหน่งใกล้กัน

– ค่าการวนซ้ำสูงสุด (Maximum Iteration): T ใช้เขียนแทนค่าการวนซ้ำสูงสุด โดย วิธี PSO จะหยุดการวนซ้ำเมื่อจำนวนการวนซ้ำเท่ากับค่าการวนซ้ำสูงสุด

ต่อจากนี้ บทความนี้จะทำการอธิบายวิธีการหาค่าที่ดีที่สุดแบบพาร์ทิเคิลสวอมออฟทิไมเซชั่นในแบบต่างๆ กัน ซึ่งแต่ละแบบใช้การจำลองโครงสร้างการเรียนรู้ทางสังคมที่แตกต่างกันไป

วิธี PSO แบบมาตรฐาน

วิธี PSO แบบมาตรฐานถูกคิดค้นขึ้นโดย Kennedy และ Eberhart ในปี 1995 [1] โดยมีการปรับปรุงเพิ่มเติมภายหลังโดย Shi และ Eberhart [2] ด้วยการเพิ่มเติมตัวพารามิเตอร์น้ำหนักแรงเฉื่อย w ลงในสมการ หลังจากนั้นวิธี PSO แบบมาตรฐานนี้ก็กลายเป็นวิธี PSO ที่นิยมใช้กันแพร่หลายมากที่สุดจนถึงปัจจุบัน วิธี PSO แบบมาตรฐานใช้การเปรียบเทียบตำแหน่งของตัวพาร์ทิเคิลแต่ละตัวกับตำแหน่งที่ดีที่สุดสากลเท่านั้น ในการเชื่อมต่อความสัมพันธ์ระหว่างตัวพาร์ทิเคิลในฝูง วิธี PSO แบบมาตรฐานนี้ใช้สมการที่ 2 และสมการที่ 3 ในการเปลี่ยนค่าความเร็วและตำแหน่ง ดังที่จะแสดงดังต่อไปนี้

$$\left.\begin{aligned} v_{id}(t+1) &= w(t)v_{id}(t)+c_p u\left(p_{id}-x_{id}(t)\right)+c_g u\left(p_{gd}-x(t)\right) \\ v_{id}(t+1) &= \begin{cases} -V_{\max} \text{ if } v_{id}(t+1) \le -V_{\max} \\ V_{\max} \text{ if } v_{id}(t+1) \ge V_{\max} \end{cases} \end{aligned}\right\} \quad (2)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (3)$$

ขั้นตอนการทำงานของ PSO แบบมาตรฐานมีดังต่อไปนี้คือ

ขั้นตอนที่ 1: ตั้งค่าพารามิเตอร์ ตั้งค่าการวนซ้ำปัจจุบัน t = 1 กำหนดตำแหน่งและความเร็วของตัวพาร์ทิเคิล K ตัวในฝูง

ขั้นตอนที่ 2: สำหรับตัวพาร์ทิเคิลแต่ละตัว แปลงตำแหน่งไปเป็นคำตอบของปัญหาที่กำหนด และทำการประเมินค่าคำตอบนั้นๆ ในฐานะที่เป็นค่าความเหมาะสมของตำแหน่งนั้นๆ โดยขั้นตอนการแปลงตำแหน่งไปเป็นคำตอบขึ้นอยู่กับแต่ละปัญหาโดยเฉพาะไม่เหมือนกัน

ขั้นตอนที่ 3: ปรับตำแหน่งที่ดีที่สุดส่วนตัว

ขั้นตอนที่ 4: ปรับตำแหน่งที่ดีที่สุดสากล

ขั้นตอนที่ 5: ปรับความเร็วและตำแหน่งของตัวพาร์ทิเคิลทุกตัวด้วยสมการที่ 2 และสมการที่ 3

ขั้นตอนที่ 6: ถ้าเงื่อนไขการหยุดทำงานสมบูรณ์ ให้หยุดการวนซ้ำ แต่ถ้าเงื่อนไขการหยุดการทำงานไม่สมบูรณ์ ให้ตั้งค่า $t = t + 1$ แล้วกลับไปทำงานที่ขั้นตอนที่ 2

วิธี PSO แบบมาตรฐานสามารถเขียนได้ในรูปคำสั่งทางคอมพิวเตอร์ได้ดังที่แสดงไว้ใน [5–6] ภาพที่ 1 แสดง Pseudo Code ที่เทียบเท่ากับขั้นตอนทั้งหกของวิธี PSO แบบมาตรฐาน ที่กล่าวมาข้างต้นนี้

```
Initialize PSO system; // Step 1

DO
    FOR i = 1 to K DO
    Decode X(i) and compute Fitness(X(i));        // Step 2
    IF Fitness(X(i)) < Fitness(P(i)) THEN P(i) = X(i) and Fitness(P(i)) = Fitness(X(i)); // Step 3
    IF Fitness(X(i)) < Fitness(Pg) THEN Pg = X(i) and Fitness(Pg) = Fitness(X(i)); // Step 4
    END DO

    FOR i = 1 to K DO // Step 5
        FOR d = 1 to D DO
        v(i,d) = w*v(i,d) + cp*u*(p(i,d) – x(i,d)) + cg*u*(pg(d) – x(i,d));
        x(i,d) = x(i,d) + v(i,d);
        END DO
    END DO
UNTIL (Stopping criterion is met) // Step 6
```

ภาพที่ 1 Pseudo Code ของวิธี PSO แบบมาตรฐาน

วิธี PSO แบบใช้ตำแหน่งที่ดีที่สุดเฉพาะบริเวณ

หลายปีต่อมาหลังจากการคิดค้นวิธี PSO แบบมาตรฐาน นักวิจัยยังคงพยายามคิดค้นวิธี PSO ในรูปแบบอื่นๆ โดยการปรับปรุงโครงสร้างการเรียนรู้ทางสังคมของวิธี PSO แบบมาตรฐาน ยกตัวอย่าง เช่น วิธี PSO แบบใช้ตำแหน่งที่ดีที่สุดเฉพาะบริเวณ ตัวพาร์ทิเคิลแต่ละตัวในวิธี PSO แบบนี้จะอยู่ในผลกระทบจากสมาชิกของกลุ่มที่เรียงตามลำดับตัวเลข หรือกล่าวอย่างง่ายๆ ได้ว่า วิธี PSO แบบนี้จะใช้สมการที่ 4 แทนการใช้สมการที่ 2 ของวิธี PSO แบบมาตรฐาน

$$\left.\begin{aligned} v_{id}(t+1) &= w(t)v_{id}(t)+c_p u\left(p_{id}-x_{id}(t)\right)+c_l u\left(p_{lid}-x(t)\right) \\ v_{id}(t+1) &= \begin{cases} -V_{\max} \text{ if } v_{id}(t+1) \le -V_{\max} \\ V_{\max} \text{ if } v_{id}(t+1) \ge V_{\max} \end{cases} \end{aligned}\right\} \quad (4)$$

Kennedy (1999) [3] กล่าวไว้ว่าฝูงพาร์ทิเคิลในวิธี PSO แบบมาตรฐาน มีแนวโน้มที่จะเกิดการกระจุกตัวเร็วกว่าการใช้วิธี PSO แบบใช้ตำแหน่งที่ดีที่สุดเฉพาะบริเวณ ซึ่งเป็นเหตุให้ฝูงพาร์ทิเคิลในวิธี PSO แบบมาตรฐานติดอยู่ในจุดที่ดีที่สุดเฉพาะบริเวณ และเสียความสามารถในการหาจุดที่ดีที่สุดแท้จริงไป ยิ่งไปกว่านั้นงานวิจัย [3] ได้ทำการเปรียบเทียบประสิทธิภาพของวิธี PSO แบบใช้ตำแหน่งที่ดีที่สุดเฉพาะบริเวณ กับวิธี PSO แบบมาตรฐาน โดยใช้ปัญหาทางคณิตศาสตร์ 4 ปัญหาพบว่า วิธี PSO แบบใช้ตำแหน่งที่ดีที่สุดเฉพาะบริเวณหาค่าคำตอบได้ดีกว่าวิธี PSO แบบมาตรฐาน ถึง 2 ปัญหาจากทั้งหมด 4 ปัญหา

วิธี PSO แบบใช้การผสมพันธุ์และประชากรย่อย

กลุ่มนักทดลองที่นำโดย Løvberg [5] ได้นำเสนอวิธี PSO สองแบบซึ่งผสมแนวคิดของ PSO เข้ากับแนวคิดวิธีเชิงพันธุกรรม (Genetic Algorithm) หรือ เรียกย่อๆ ว่า GA [6] โดยวิธี PSO ทั้งสองแบบได้แก่ วิธี PSO แบบใช้การผสมพันธุ์ และวิธี PSO แบบใช้ประชากรย่อย

วิธี PSO แบบใช้การผสมพันธุ์เป็นวิธีที่ปรับปรุงมาจากวิธี PSO แบบมาตรฐานโดยนำเอาการปรับปรุงค่าคำตอบจาก GA นั่นคือ การผสมพันธุ์ (Breeding) มาใส่ลงในวิธี PSO แบบมาตรฐาน โดยในแต่ละการวนซ้ำ วิธี PSO แบบนี้จะเหมือนกับวิธี PSO แบบมาตรฐานทุกประการ แต่จะมีขั้นตอนที่เพิ่มขึ้นมาคือ PSO จะทำการเลือกตัวพาร์ทิเคิลแบบสุ่มออกมาสองตัวในฐานะที่เป็นพ่อและแม่ จากนั้นใช้วิธีการ Crossover เพื่อให้ได้ลูกพาร์ทิเคิลออกมาสองตัว แล้วนำตัวลูกพาร์ทิเคิลสองตัวที่ได้นั้นไปแทนที่ตัวพ่อแม่ ที่ทำแบบนี้เพื่อให้ปริมาณตัวพาร์ทิเคิลในฝูงคงที่ ตำแหน่งของลูกทั้งสองตัวในแต่ละมิติได้มาจากวิธีการ Crossover บนตำแหน่งของพ่อแม่ในมิตินั้นๆ ดังแสดงในสมการที่ 5 และ 6

$$child_1(x_i) = u \times parent_1(x_i) + (1-u) \times parent_2(x_i) \quad (5)$$

$$child_2(x_i) = u \times parent_2(x_i) + (1-u) \times parent_1(x_i) \quad (6)$$

ค่าเวคเตอร์ความเร็วของลูกๆ หาได้โดยผลรวมเวคเตอร์ความเร็วของพ่อแม่ซึ่งผ่านการ Normalize จากขนาดเวคเตอร์ความเร็วของพ่อแม่แต่ละตัว ดังแสดงในสมการที่ 7 และ 8

$$child_1(V) = \frac{parent_1(V)+parent_2(V)}{\left|parent_1(V)+parent_2(V)\right|} \times \left|parent_1(V)\right| \quad (7)$$

$$child_1(V) = \frac{parent_1(V)+parent_2(V)}{\left|parent_1(V)+parent_2(V)\right|} \times \left|parent_1(V)\right| \quad (8)$$

วิธี PSO แบบใช้ประชากรย่อย เป็นวิธี PSO ที่ถูกพัฒนาขึ้นมาจากวิธี PSO แบบใช้การผสมพันธุ์ที่กล่าวมาข้างต้น นั่นคือวิธี PSO ทั้งสองมีรูปแบบการปรับตำแหน่งและความเร็วของตัวพาร์ทิเคิลเหมือนกัน แต่วิธี PSO แบบใช้ประชากรย่อยจะแบ่ง

ฝูงพาร์ทิเคิลออกเป็นฝูงย่อยๆ หลายฝูง โดยมีวัตถุประสงค์ที่ว่าฝูงย่อยจะสามารถแยกกันหาตำแหน่งที่ดีที่สุด ซึ่งอาจจะดีกว่าการหาตำแหน่งที่ดีที่สุดเป็นฝูงใหญ่ ความแลกเปลี่ยนข้อมูลระหว่างฝูงจะเกิดขึ้นได้เมื่อพ่อแม่จากต่างฝูงมีการผสมพันธุ์กัน

ประสิทธิภาพของวิธี PSO แบบใช้การผสมพันธุ์ และวิธี PSO แบบใช้ประชากรย่อยนี้ถูกนำมาเปรียบเทียบกับวิธี PSO แบบมาตรฐาน และวิธี GA แบบมาตรฐาน โดยผลลัพธ์แสดงให้เห็นว่าวิธี PSO วิธีใหม่นี้ทำงานได้ดีในฟังก์ชั่นที่มีจุดที่ดีที่สุดหลายจุด และมีประสิทธิภาพไม่ดีนักในฟังก์ชั่นที่มีจุดที่ดีที่สุดจุดเดียวเมื่อเปรียบเทียบกับวิธีแบบมาตรฐาน

วิธี Dissipative PSO

กลุ่มนักวิจัยนำโดย Xie [7] ทำการปรับปรุงวิธี PSO แบบมาตรฐานโดยการนำเสนอ Negative Entropy ให้กับ PSO โดยการวิจัยของ Xie [7] เสนอว่า Negative Entropy ความน่าจะเป็นที่ฝูงพาร์ทิเคิลจะมาติดอยู่ในจุดที่ดีที่สุดเฉพาะบริเวณได้ โดยเราสามารถใส่ Negative Entropy ลงใน PSO ได้โดยใช้สมการที่ 9 และ 10 ภายหลังจากใช้สมการที่ 2 และ 3 ในวิธี PSO แบบมาตรฐานเพื่อการปรับค่าความเร็วและตำแหน่งของตัวพาร์ทิเคิลแต่ละตัว

$$v_{id}(t+1) = U[-V_{\max}, V_{\max}] \text{ ถ้า } c_v > U[0, 1] \quad (9)$$

$$x_{id}(t+1) = U[X_{\min}, X_{\max}] \text{ ถ้า } c_x > U[0, 1] \quad (10)$$

โดย c_v และ c_x คือ Chaotic factor ในช่วง [0, 1]

ในงานวิจัยของ Pongchairerks และ Kachitvichyanukul [8] แสดงให้เห็นว่า Dissipative PSO ให้ผลลัพธ์ที่ดีกว่า วิธี PSO แบบมาตรฐานในฟังก์ชั่นที่มีจุดที่ดีที่สุดหลายจุด

วิธี FDR-PSO

กลุ่มนักวิจัยนำโดย Veeramachaneni [4] นำเสนอวิธี PSO ซึ่งตัวพาร์ทิเคิลจะติดต่อสื่อสารกันภายในกลุ่มย่อยของฝูง โดยกลุ่มย่อยจะถูกแยกโดย Fitness-Distance-Ratio (FDR) แรงจูงใจในการพัฒนาวิธี FDR-PSO คือเพื่อเพิ่มความสามารถในการหนีออกจากกระจุกตัวกันในจุดที่ดีที่สุดเฉพาะบริเวณ แนวความคิดที่ใช้ใน PSO แบบนี้คือ การเพิ่มตำแหน่งที่ดีที่สุดมากขึ้น ทำให้ตัวพาร์ทิเคิลมีเวลาในการค้นหาพื้นที่บริเวณต่างๆ ก่อนที่จะเข้ามารวมตัวกัน เพื่อให้ได้ผลสำเร็จตามที่คาด [4] ได้ทำการแนะนำตำแหน่งที่ดีที่สุดของกลุ่มที่ตำแหน่งใกล้กัน (Near neighbor Best Position) ดังที่กล่าวไว้แล้วในตอนต้นของรายงานฉบับนี้ โดย FDR-PSO จะปรับความเร็วโดยใช้สมการที่ 11 แทนการใช้สมการที่ 2 ใน PSO แบบมาตรฐาน

$$\left.\begin{aligned} v_{id}(t+1) &= w(t)v_{id}(t) + c_p u\left(p_{id} - x_{id}(t)\right) + c_g u\left(p_{gd} - x(t)\right) \\ &\quad + c_n u\left(p_{nid} - x_{id}(t)\right) \\ v_{id}(t+1) &= \begin{cases} -V_{\max} \text{ if } v_{id}(t+1) \le -V_{\max} \\ V_{\max} \text{ if } v_{id}(t+1) \ge V_{\max} \end{cases} \end{aligned}\right\} \quad (11)$$

ประสิทธิภาพของ FDR-PSO ถูกนำมาเปรียบเทียบกับ วิธี PSO แบบมาตรฐาน และวิธี GA แบบมาตรฐาน โดยผลลัพธ์ที่ได้จาก FDR-PSO หาคำตอบได้ดีกว่าวิธีอื่นๆ ในทุกปัญหาที่ทำการทดสอบ

วิธี GLN-PSO

ในปี ค.ศ. 2009 Pongchairereks และ Kachitvichyanukul [8] ได้นำเสนอวิธี GLN-PSO ซึ่งใช้ตำแหน่งที่ดีที่สุดส่วนตัว, ตำแหน่งที่ดีที่สุดสากล, ตำแหน่งที่ดีที่สุดเฉพาะบริเวณ และตำแหน่งที่ดีที่สุดของกลุ่มที่ตำแหน่งใกล้กันไปพร้อมๆ กัน โดยมีเป้าหมายหลักที่จะให้ตัวพาร์ทิเคิลกระจายกันหาจุด

ที่ดีที่สุดในบริเวณต่างๆ ออกเป็นกลุ่มย่อยๆ พร้อมๆ กันก่อน แล้วจึงค่อยๆ รวมตัวกันค้นหาคำตอบที่ดีที่สุดในภายหลัง GLN-PSO จะปรับความเร็วโดยใช้สมการที่ 12 แทนการใช้สมการที่ 2 ใน PSO แบบมาตรฐาน

$$\left.\begin{aligned} v_{id}(t+1) &= w(t)v_{id}(t)+c_p u\left(p_{id}-x_{id}(t)\right)+c_g u\left(p_{gd}-x(t)\right) \\ &\quad +c_l u\left(p_{lid}-x_{id}(t)\right)+c_n u\left(p_{nid}-x_{id}(t)\right) \\ v_{id}(t+1) &= \begin{cases} -V_{\max} \text{ if } v_{id}(t+1) \le -V_{\max} \\ V_{\max} \text{ if } v_{id}(t+1) \ge V_{\max} \end{cases} \end{aligned}\right\} \quad (12)$$

ขั้นตอนการทำงานของ GLN-PSO มีดังต่อไปนี้คือ

ขั้นตอนที่ 1: ตั้งค่าพารามิเตอร์ ตั้งค่าการวนซ้ำปัจจุบัน t = 1 กำหนดตำแหน่งและความเร็วของตัวพาร์ทิเคิล K ตัวในฝูง

ขั้นตอนที่ 2: สำหรับตัวพาร์ทิเคิลแต่ละตัว แปลงตำแหน่งไปเป็นคำตอบของปัญหาที่กำหนด และทำการประเมินค่าคำตอบนั้นๆ ในฐานะที่เป็นค่าความเหมาะสมของตำแหน่งนั้นๆ โดยขั้นตอนการแปลงตำแหน่งไปเป็นคำตอบขึ้นอยู่กับแต่ละปัญหาโดยเฉพาะไม่เหมือนกัน

ขั้นตอนที่ 3: ปรับตำแหน่งที่ดีที่สุดส่วนตัว

ขั้นตอนที่ 4: ปรับตำแหน่งที่ดีที่สุดสากล

ขั้นตอนที่ 5: ปรับตำแหน่งที่ดีที่สุดเฉพาะบริเวณ

ขั้นตอนที่ 6: ปรับตำแหน่งที่ดีที่สุดของกลุ่มที่ตำแหน่งใกล้กัน

ขั้นตอนที่ 7: ปรับความเร็วและตำแหน่งของตัวพาร์ทิเคิลทุกตัวด้วยสมการที่ 12 และสมการที่ 3

ขั้นตอนที่ 8: ถ้าเงื่อนไขการหยุดทำงานสมบูรณ์ ให้หยุดการวนซ้ำ แต่ถ้าเงื่อนไขการหยุดการทำงานไม่สมบูรณ์ ให้ตั้งค่า $t = t + 1$ แล้วกลับไปทำงานที่ขั้นตอนที่ 2

งานวิจัย [8] นำ GLN-PSO มาเปรียบเทียบกับวิธี PSO แบบมาตรฐาน และวิธี FDR-PSO ในปัญหาสมการทางคณิตศาสตร์เหล่านี้คือ Sphere, Rosenbrock, Ratrigin, Griewank, Ackley และ Move axis parallel hyper-ellipsoid (MAPH) [3, 9] โดยงานวิจัยนี้กำหนดให้ PSO ทั้งสามเวอร์ชั่นใช้ค่าพารามิเตอร์ที่เหมือนกันดังนี้คือ ขนาดประชากร (K)=20 และหยุดการประมวลผลเมื่อค่าการวนซ้ำสูงสุด (T)=1000 รอบ แล้วนำค่าที่ดีที่สุดที่ได้จากการประมวลผล 20 รอบมาเปรียบเทียบกัน ผลการทดลองดังแสดงในตารางที่ 1 คือ GLN-PSO หาผลลัพธ์ได้ดีที่สุดในทุกๆ ปัญหา ตามมาด้วย FDR-PSO และ Standard PSO

ตารางที่ 1 แสดงผลลัพธ์ที่ได้จากการเปรียบเทียบใน [8]

ปัญหา	เวอร์ชั่น		
	Standard PSO	**FDR-PSO**	**GLN-PSO**
Sphere	631.51 (1 sec.)	3.11 (2 sec.)	0.002 (2 sec.)
Rosenbrock	950.10 (1 sec.)	201.72 (2 sec.)	132.600 (3 sec.)
Rastrigin	251.22 (1 sec.)	179.31 (2 sec.)	102.700 (2 sec.)
Griewank	2.90 (1 sec.)	0.56 (2 sec.)	0.050 (2 sec.)
Ackley	6.54 (1 sec.)	3.95 (2 sec.)	2.060 (2 sec.)
MAPH	896.34 (1 sec.)	0.61 (2 sec.)	0.001 (2 sec.)

การประยุกต์ใช้ PSO กับปัญหาทางวิศวกรรมด้านต่างๆ

เนื้อหาในส่วนนี้จะทำการสรุปตัวอย่างบทความที่ได้ประยุกต์ใช้ PSO ในปัญหาทางวิศวกรรมในด้านต่างๆ ดังแสดงในตารางที่ 2 โดยดังที่ได้กล่าวมาแล้วข้างต้นว่าตำแหน่งของพาร์ทิเคิลแต่ละตัวสามารถแปลงเป็นคำตอบ หรือ Solution ของปัญหาที่แตกต่างกันได้ โดยเทคนิคการแปลงตำแหน่งของพาร์ทิเคิลเป็นคำตอบของปัญหาหนึ่งๆ นั้นแต่ละปัญหาจะมีวิธีการเฉพาะที่แตกต่างกันไป

ตารางที่ 2 สรุปบทความที่ประยุกต์ใช้ PSO ในปัญหาทางวิศวกรรม

ปัญหา	บทความ
Sequential ordering problem	Anghinolfi et al. [10]
Job-shop scheduling problem	Pongchairerks and Kachitvichyanukul [11]
Economic load dispatch with generator constraints	Safari and Shayeghi [12]
Flowshop scheduling problem	Liu et al. [13]
Order planning problem	Zhang et al. [14]
Joint pricing and lot-sizing problem	Dye and Ouyang [15]

สรุป

รายงานฉบับนี้ได้ทำการสรุปบทความวิจัยต่างๆ ที่ทำการคิดค้นวิธี PSO แบบต่างๆ โดยมีแบบจำลองโครงสร้างการเรียนรู้ทางสังคมที่แตกต่างกัน อาทิเช่น วิธี PSO แบบมาตรฐาน, วิธี PSO แบบใช้ตำแหน่งที่ดีที่สุดเฉพาะบริเวณ, วิธี PSO แบบใช้การผสมพันธุ์และประชากรย่อย, วิธี Dissipative PSO, วิธี FDR-PSO และ วิธี GLN-PSO โดยเนื่องมาจากจุดอ่อนของวิธี PSO แบบมาตรฐานซึ่งใช้เพียงตำแหน่งที่ดีที่สุดสากลในการสื่อสารระหว่างตัวพาร์ทิเคิลภายในฝูง อันเป็นเหตุให้ฝูงพาร์ทิเคิลเกิดการกระจุกตัวที่เร็วเกินไปทำให้มีโอกาสติดกับจุดที่ดีที่สุดเฉพาะบริเวณได้ง่าย ทำให้ผลงานวิจัยรุ่นหลังๆ มุ่งเน้นการสร้าง PSO แบบจำลองโครงสร้างการเรียนรู้ทางสังคมที่พยายามแยกฝูงพาร์ทิเคิลออกมาเป็นกลุ่มย่อยๆ ซึ่งผลลัพธ์จากงานเพื่อเพิ่มขีดความสามารถในการค้นหาจุดที่ดีที่สุดที่แท้จริง ดังที่ได้กล่าวมา

เอกสารอ้างอิง

[1] Kennedy, J.; & Eberhart, R.C. (1995). Particle Swarm Optimization. In *IEEE International Conference on Neural Network*. pp. 1942-1948. New Jersey: n.p.

[2] Shi, Y.; & Eberhart, R.C. (1998). A Modified Particle Swarm Optimizer. In *IEEE International Conference on Evolutionary Computation*. pp. 69-73. New Jersey: n.p.

[3] Kennedy, J. (1999). Small Worlds and Mega-minds: Effects of Neighborhood Topology on Particle Swarm Performance. In *Congress on Evolutionary Computation*. pp. 1931-1938. New York: n.p.

[4] Veeramachaneni, K.; et al. (2003). Optimization Using Particle Swarms with Near Neighbor Interactions. In *Proceedings of the 2003 Genetic and Evolutionary Computation Conference*. pp. 110–122. Chicago: n.p.

[5] Løvberg, M.R.; Rasmussen, T.K.; & Krink, T. (2001). Hybrid Particle Swarm Optimizer with Breeding and Subpopulations. In *Proceedings of the Third Genetic and Evolutionary Computation Conference*. pp. 469–476. San Francisco: n.p.

[6] Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Michigan: University of Michigan Press.

[7] Xie, X.F.; Zhang, W.J.; & Yang, Z.L. (2002). A Dissipative Particle Swarm Optimization. In *Proceedings of IEEE Congress on Evolutionary Computation*. pp. 1456–1461. Honolulu: n.p.

[8] Pongchairerks, P.; & Kachitvichyanukul, V. (2009). A Particle Swarm Optimization Algorithm with Multiple Social Learning Structures. *International Journal of Operational Research*. 6(2): 176–194.

[9] Pohlheim, H. (2005). *Genetic and Evolutionary Algorithm Toolbox for Use with MATLAB*. from http://www/geatbx.com/docu/index.html

[10] Anghinolfi, D.; et al. (2011). A hybrid particle swarm optimization approach for the sequential ordering problem. *Computers & Operations Research*. 38(7): 1076–1085.

[11] Pongchairerks, P.; & Kachitvichyanukul, V. (2009). A Particle Swarm Optimization Algorithm on Job–Shop Scheduling Problems with Multi–Purpose Machines. *Asia–Pacific Journal of Operational Research*. 26(2): 161–184.

[12] Safari, A.; & Shayeghi, H. (2011). Iteration Particle Swarm Optimization Procedure for Economic Load Dispatch with Generator Constraints. *Expert Systems with Applications*. 38(5): 6043–6048.

[13] Liu, H.; Gao, L.; & Pan, Q. (2011). A Hybrid Particle Swarm Optimization with Estimation of Distribution Algorithm for Solving Permutation Flowshop Scheduling Problem. *Expert Systems with Application*. 38(4): 4348–4360.

[14] Zhang, T.; et al. (2011). A Hybrid Particle Swarm Optimization and Tabu Search Algorithm for Order Planning Problems of Steel Factories Based on the Make–To–Stock and Make–To_order Management Architecture. *Journal of Industrial and Management Optimization*. 7(1): 31–51.

[15] Dye, C.–Y.; & Ouyang, L.–Y. (2011). A PArticle Swarm Optimization for Solving Joint Pricing and Lot–Sizing Problem witj Fluctuating Demand and Trade Credit Financing. *Computers and Industrial Engineering*. 60(1): 127–137.