

CUTTLEFISH ALGORITHM

MEMBERS

1. 6410110326 ปาณิสรา ลั้งษ์ทอง
2. 6410110706 มาเรียมณี สุระอเอง
3. 6410110721 ศศิธร คักดีแก้ว



Reference: Cuttlefish Algorithm - A Novel Bio-Inspired Optimization Algorithm
By Adel Sabry Eesa, Adnan Mohsin Abdulazeez Brifcani, Zeynep Orman

KEY IDEA CONCEPT

CUTTLEFISH ALGORITHM (CFA) เป็นอัลกอริทึมที่ได้รับแรงบันดาลใจจากพฤติกรรมการเปลี่ยนสีของหมึกกระดอง โดยเลียนแบบการสะท้อนแสงและการแสดงความชัดเจนของลวดลายที่ใช้ในการอำพรางตัว อัลกอริทึมนี้ใช้

- กระบวนการสะท้อน (REFLECTION)
- การมองเห็นลวดลาย (VISIBILITY)

ในการค้นหา GLOBAL OPTIMUM SOLUTION โดยการปรับเปลี่ยนค่าของโซลูชันปัจจุบันให้ใกล้เคียงกับค่าที่ดีที่สุด

THEORY

การเปลี่ยนแปลงของหมึกกระดองเป็นการปรับตัวเพื่อเพิ่มความสามารถในการอยู่รอด ซึ่งเทียบเท่ากับการค้นหาโซลูชันที่ดีที่สุดในการหาค่าเหมาะสมสุด (OPTIMIZATION PROBLEM) ในแง่ของ COST FUNCTION จะมองเห็นลวดลายที่ดีที่สุดเป็น GLOBAL OPTIMUM SOLUTION และลวดลายที่ยังไม่ใกล้เคียงเป็นค่า COST FUNCTION ที่สูง การสะท้อนแสงและการปรับลวดลายเปรียบเสมือนการปรับค่าโซลูชันเพื่อหาโซลูชันที่เหมาะสมที่สุด

NATURAL BEHAVIOR

การเปลี่ยนแปลงของหมึกกระดองเป็นการปรับตัวเพื่อเพิ่มความสามารถในการอยู่รอด ซึ่งเทียบเท่ากับการค้นหาโซลูชันที่ดีที่สุด
ในปัญหาการหาค่าเหมาะสมที่สุด (OPTIMIZATION PROBLEM)
ในแง่ของ COST FUNCTION

- การมองเห็นลวดลายเหมาะสมที่สุดเป็น GLOBAL OPTIMUM SOLUTION
- การมองเห็นลวดลายที่ยังไม่ถูกต้องทำให้ค่า COST FUNCTION สูง

การสะท้อนแสงและการปรับลวดลายเปรียบเสมือนการปรับค่าโซลูชันเพื่อหาโซลูชันที่เหมาะสมที่สุด

CUTTLEFISH ALGORITHM (CFA)

CFA ทำงานโดยเริ่มต้นด้วยการสุ่มค่าโซลูชัน แล้วค่อยๆ
ปรับค่าโดยใช้กระบวนการสะท้อนแสงและการมองเห็น
ลดทลาย กระบวนการทำงานแบ่งเป็น 4 กลุ่มหลัก

กลุ่ม 1: การสะท้อนแสงจาก CHROMATOPHORES และ IRIDOPHORES เพื่อสร้างโซลูชันใหม่ (GLOBAL SEARCH)

กลุ่ม 2: การสะท้อนแสงจาก IRIDOPHORES เพื่อปรับโซลูชันใกล้เคียง (LOCAL SEARCH)

กลุ่ม 3: การสะท้อนแสงจาก LEUCOPHORES ที่ใช้ค่าเฉลี่ยของจุดที่ดีที่สุด (LOCAL SEARCH)

กลุ่ม 4: การสะท้อนแสงจาก LEUCOPHORES ในสภาพแวดล้อมเพื่อหาค่าเริ่มต้นใหม่ (GLOBAL SEARCH)

CUTTLEFISH ALGORITHM (CFA)

ตัวอย่างการใช้ ALGORITHM แก้ปัญหา:

อัลกอริทึม CFA ถูกทดสอบกับปัญหาการ OPTIMIZATION เช่น ROSENBROCK'S VALLEY FUNCTION ซึ่งเป็นฟังก์ชันที่หาจุดต่ำสุด โดยมีการกำหนดค่าของฟังก์ชัน ดังนี้

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2], \quad -2.048 \leq x_i \leq 2.048, \quad i=1, \dots, n$$

$$F_{\min}(X) = 0, \quad X(1, 1, \dots, 1)$$

การประเมินผล

การประเมินผลของ CFA ได้ทำการเปรียบเทียบกับอัลกอริธึมการเพิ่มประสิทธิภาพที่เป็นที่นิยมอื่นๆ เช่น Genetic Algorithms (GA), Particle Swarm Optimization (PSO) และ Bees Algorithm (BA) โดยมีการใช้ 12 ฟังก์ชันทดสอบที่รู้จักกันดี ซึ่งได้แก่ De Jong, Griewangk, Ackley, Rastrigin, และอื่น ๆ

COMPARISON OF CFA WITH GA, PSA AND BEES ALGORITHM IN TERM MEAN NUMBER OF FUNCTIONS EVALUATION AND SUCCESS RATE, (100 RUN, 200 ITERATION, 10,000 FUNCTION EVALUATIONS)

Function	GA	PSO	BA	CFA
1. d=120	6962 (59%)	****	****	1311 (100%)
2. d=120	6889.5 (53%)	****	****	3052 (100%)
3. d=120	7426.5 (50%)	****	****	2336.5 (100%)
4. d=120	6919.5 (58%)	****	****	2220 (100%)
5. d=120	7116.5 (53%)	****	****	1703.5 (100%)
6. d=2	9901 (1%)	9707.5 (3%)	1448 (100%)	236 (100%)
7. d=2	9900.5 (1%)	1407.5 (100%)	7197 (46%)	968.5 (100%)
8. d=2	****	2094 (100%)	5868 (72%)	335.5 (100%)
9. d=2	****	3046 (100%)	****	876 (100%)
10. d=2	****	3622 (86%)	5385 (85%)	560 (100%)
11. d=2	5731 (72%)	1465 (100%)	9628.5 (7%)	446 (100%)
12. d=2	9999 (1%)	1447 (100%)	2753 (93%)	893.5 (100%)

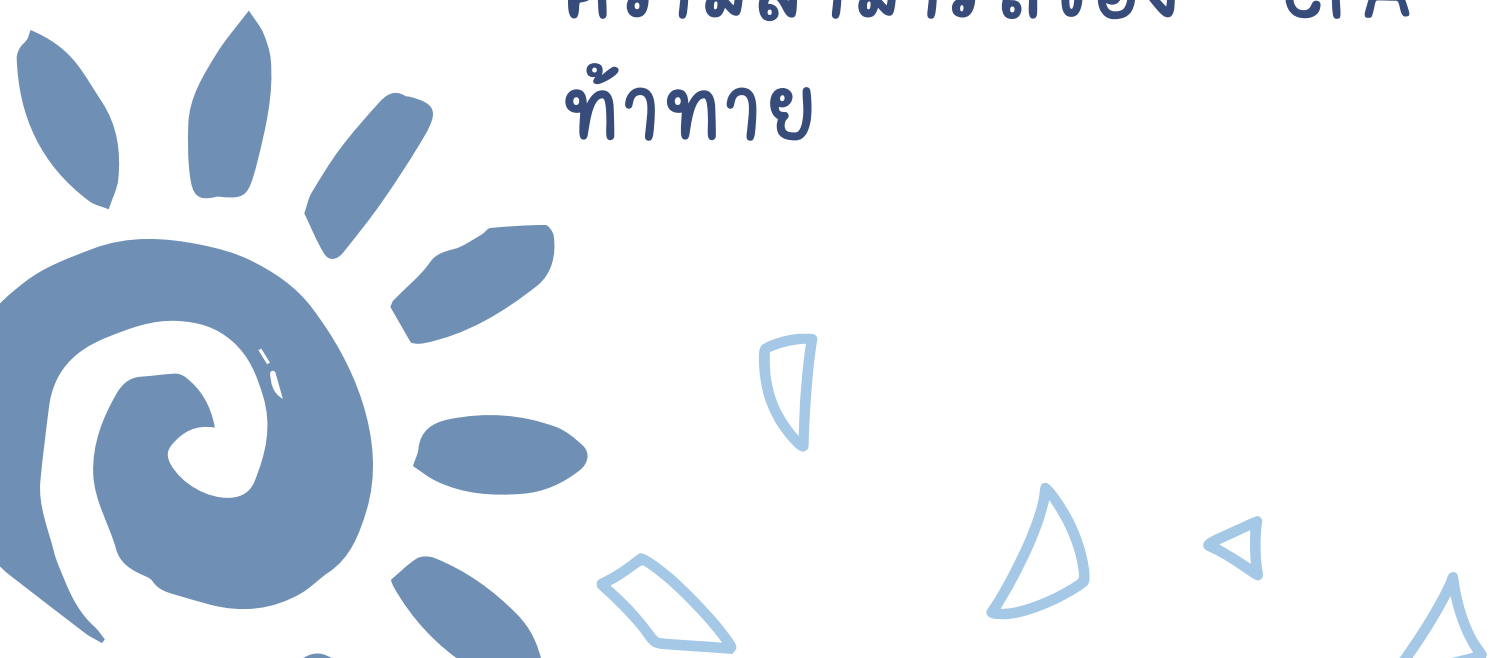
EXAMPLE

ผลลัพธ์

- ผลลัพธ์ที่ได้จากการทดลองแสดงให้เห็นว่า CFA มีประสิทธิภาพที่เหนือกว่าอัลกอริธึมอื่นๆ ในแง่ของความถูกต้องและความเร็วในการหาคำตอบที่ดีที่สุด โดย CFA สามารถหาคำตอบที่ใกล้เคียงกับค่าต่ำสุดได้อย่างรวดเร็วและมีอัตราความสำเร็จในการหาค่าต่ำสุดสูงถึง 100% ในหลายฟังก์ชัน

การวิเคราะห์

- การวิเคราะห์ผลการทดลองแสดงให้เห็นว่า CFA มีความสามารถในการค้นหาคำตอบที่ดีที่สุดได้ดีกว่า GA, PSO, และ BA ในทุกฟังก์ชันที่ทดสอบ โดยเฉพาะในฟังก์ชันที่มีความซับซ้อนสูง ซึ่งแสดงให้เห็นถึงความสามารถของ CFA ในการจัดการกับปัญหาการเพิ่มประสิทธิภาพที่ท้าทาย





Thank You