

THUẬT TOÁN NSGA-II CẢI TIẾN VỚI CƠ CHẾ LAI GHÉP HỖN HỢP VÀ THÍCH NGHI

...

Ngày 18 tháng 1 năm 2026

Tóm tắt nội dung

Tóm tắt: Tài liệu này trình bày chi tiết về thuật toán *NSGA-II Improved Smart*, một biến thể nâng cao của thuật toán di truyền đa mục tiêu NSGA-II. Thuật toán đề xuất tích hợp Khởi tạo dựa trên đối lập (Opposition-Based Learning - OBL), toán tử lai ghép hỗn hợp giữa SBX và Differential Evolution (DE), cùng cơ chế điều chỉnh tham số thích nghi (Adaptive Parameter Control). Mục tiêu chính là khắc phục các hạn chế về khả năng thoát khỏi cực trị địa phương và duy trì độ đa dạng quần thể của phiên bản NSGA-II truyền thống.

Mục lục

1	Giới thiệu	2
2	Phương pháp đề xuất	2
2.1	Khởi tạo dựa trên đối lập (OBL)	2
2.2	Toán tử Lai ghép Hỗn hợp (Hybrid Operators)	2
2.2.1	Toán tử SBX và Đột biến Đa thức	2
2.2.2	Smart Differential Evolution (Smart DE)	3
2.3	Cơ chế Tham số Thích nghi (Adaptive Parameters)	3
2.4	Chiến lược Khởi động lại (Restart Strategy)	3
3	Quy trình Thuật toán	4
4	Lưu đồ thuật toán	4
5	Kết luận và Thảo luận	6

1 Giới thiệu

NSGA-II (Non-dominated Sorting Genetic Algorithm II) là một trong những thuật toán tối ưu đa mục tiêu phổ biến nhờ vào cơ chế xếp hạng không trội (Non-dominated sorting) và khoảng cách đám đông (Crowding distance). Tuy nhiên, thực tế triển khai cho thấy NSGA-II thường gặp khó khăn trong việc cân bằng giữa khả năng khai thác (exploitation) và khám phá (exploration), đặc biệt dễ dẫn đến hội tụ sớm trên các bài toán có không gian tìm kiếm phức tạp hoặc nhiều cực trị địa phương.

Thuật toán cải tiến được trình bày trong báo cáo này khắc phục các nhược điểm trên bằng cách tích hợp bốn cơ chế chính:

1. **OBL (Opposition-Based Learning):** Mở rộng vùng tìm kiếm ban đầu.
2. **Hybrid DE/SBX:** Kết hợp ưu điểm của Tiến hóa vi phân (DE) và Lai ghép SBX.
3. **Adaptive Mechanism:** Tự động điều chỉnh tham số F và CR dựa trên dữ liệu lịch sử (lấy cảm hứng từ JADE/SHADE).
4. **Restart Strategy:** Cơ chế khởi động lại nhằm phá vỡ trạng thái trì trệ của quần thể.

2 Phương pháp đề xuất

2.1 Khởi tạo dựa trên đối lập (OBL)

Thay vì chỉ khởi tạo ngẫu nhiên, thuật toán sử dụng OBL để quét không gian đối diện, giúp tăng xác suất tìm được miền nghiệm tốt ngay từ đầu. Với một cá thể $X = (x_1, x_2, \dots, x_D)$ nằm trong miền $[L, U]$, cá thể đối lập \check{X} được tính như sau:

$$\check{x}_j = L_j + U_j - x_j, \quad \forall j = 1, \dots, D \quad (1)$$

Trong thuật toán này, OBL được sử dụng ở hai giai đoạn:

- **Khởi tạo:** Kết hợp quần thể ngẫu nhiên và quần thể đối lập, chọn N cá thể tốt nhất.
- **Nhảy cóc (OBL Jump):** Mỗi 10 thế hệ, áp dụng OBL động dựa trên miền giá trị hiện tại của quần thể ($\min(X_{curr}), \max(X_{curr})$).

2.2 Toán tử Lai ghép Hỗn hợp (Hybrid Operators)

Thuật toán sử dụng một xác suất P_{DE} để lựa chọn linh hoạt giữa hai toán tử sinh sản cho mỗi cá thể.

2.2.1 Toán tử SBX và Đột biến Đa thức

Đây là toán tử tiêu chuẩn của NSGA-II.

- **Simulated Binary Crossover (SBX):** Tạo ra con dựa trên phân phối xác suất β :

$$x_{child} = 0.5 \times [(1 + \beta)x^{(1)} + (1 - \beta)x^{(2)}] \quad (2)$$

- **Polynomial Mutation:** Đột biến nhẹ để duy trì sự đa dạng gen với chỉ số phân bố $\eta_m = 20$.

2.2.2 Smart Differential Evolution (Smart DE)

Cải tiến này sử dụng thông tin láng giềng (Neighborhood) trong không gian mục tiêu.

- **Global DE ($DE/rand/1$):** Khám phá toàn cục với xác suất 30%.

$$V_i = X_{r1} + F \cdot (X_{r2} - X_{r3}) \quad (3)$$

- **Local DE ($DE/current-to-pbest/1$):** Khai thác cục bộ dựa trên láng giềng tốt nhất (X_{pbest}).

$$V_i = X_i + F \cdot (X_{pbest} - X_i) + F \cdot (X_{r1} - X_{r2}) \quad (4)$$

Trong đó, X_{pbest} được chọn từ top K láng giềng gần nhất (dựa trên khoảng cách Euclid của hàm mục tiêu).

2.3 Cơ chế Tham số Thích nghi (Adaptive Parameters)

Các tham số điều khiển F (Scaling factor) và CR (Crossover rate) không cố định. Thuật toán ghi nhận các giá trị F, CR đã tạo ra cá thể con ưu tú (xếp hạng Rank 1). Sau mỗi thế hệ, giá trị trung bình được cập nhật theo trung bình Lehmer:

$$mean_F_{new} = \text{LehmerMean}(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F} \quad (5)$$

$$mean_CR_{new} = \text{ArithmeticMean}(S_{CR}) \quad (6)$$

Điều này giúp thuật toán tự "học" được cấu hình tham số phù hợp cho từng giai đoạn tiến hóa.

2.4 Chiến lược Khởi động lại (Restart Strategy)

Để tránh rơi vào cực trị địa phương (Local Optima stagnation), thuật toán theo dõi sự thay đổi của điểm lý tưởng (Ideal Point). Nếu sau một khoảng thời gian (ví dụ: $Gen/4$) không có sự cải thiện đáng kể ($\epsilon < 10^{-4}$):

1. Giữ lại 30% cá thể tốt nhất (Elitism).
2. Thay thế 70% cá thể còn lại bằng các cá thể ngẫu nhiên mới.
3. Reset các tham số thích nghi về mặc định.

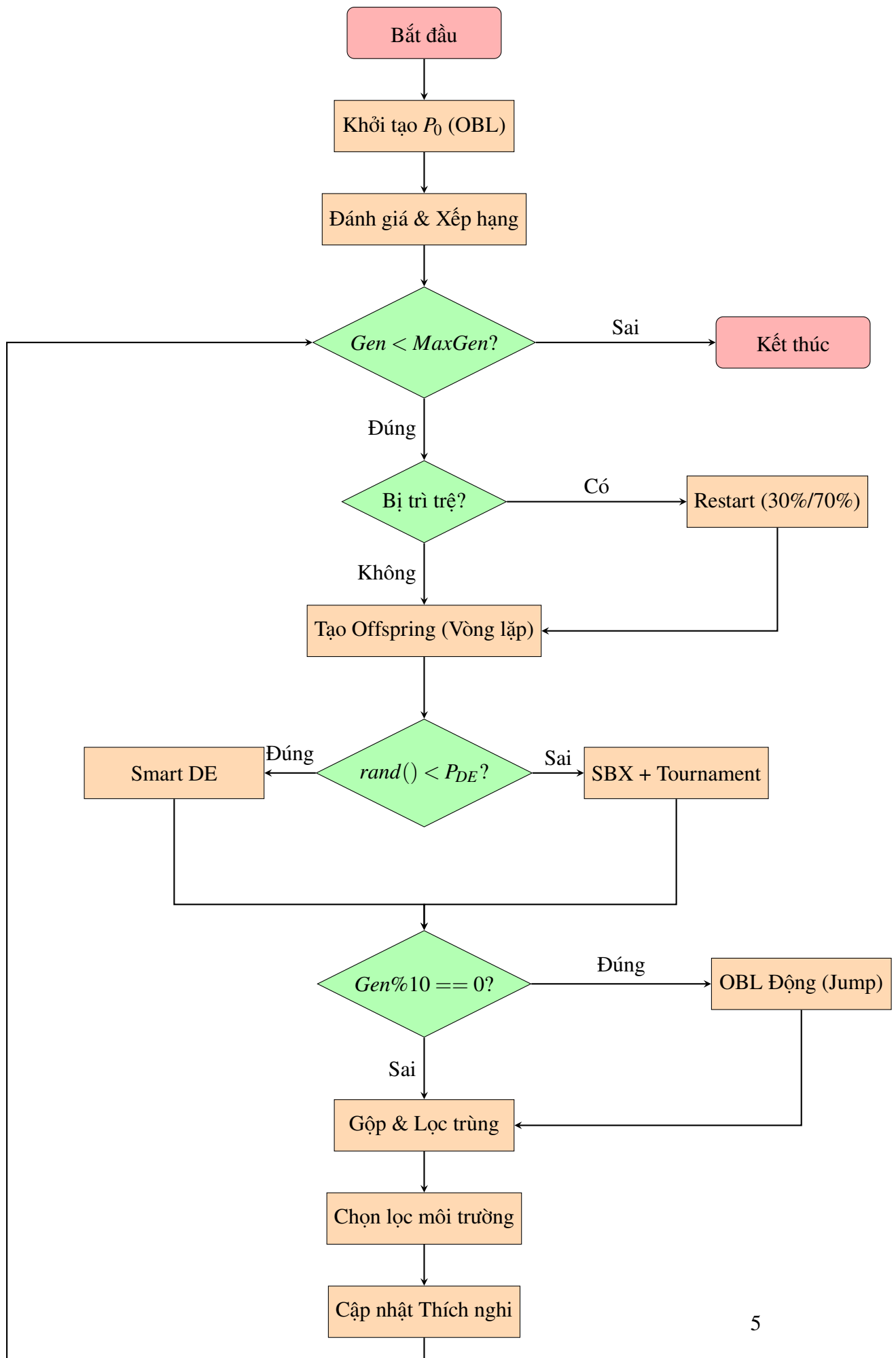
3 Quy trình Thuật toán

Quy trình tổng quát của NSGA2_Improved_Smart được mô tả như sau:

Algorithm 1 NSGA-II Cải tiến

```
1: Khởi tạo quần thể  $P_0$  sử dụng OBL.
2: Đánh giá  $P_0$ , tính Rank và Crowding Distance.
3: while  $gen < MaxGen$  do
4:   Kiểm tra điều kiện trì trệ (Stagnation check).
5:   if Stagnation detected then
6:     Thực hiện Restart (Giữ top 30%, tạo mới 70%).
7:   end if
8:    $Offspring \leftarrow \emptyset$ 
9:   for mỗi cá thể  $i$  trong  $P_{gen}$  do
10:    Tìm láng giềng  $T\_neighbors$  của  $i$ .
11:    if  $rand() < P_{DE}$  then
12:       $child \leftarrow \text{Smart\_DE\_Mutation}(i, neighbors, \text{adaptive params})$ 
13:    else
14:      Chọn cha mẹ bằng Tournament.
15:       $child \leftarrow \text{SBX\_Crossover\_Mutation}(p1, p2)$ 
16:    end if
17:    Thêm  $child$  vào  $Offspring$ .
18:  end for
19:  if  $gen \% 10 == 0$  then
20:    Sinh thêm cá thể OBL động.
21:  end if
22:   $CombinedPop \leftarrow P_{gen} \cup Offspring$ 
23:  Loại bỏ cá thể trùng lặp (Duplicate Removal).
24:   $P_{gen+1} \leftarrow \text{EnvironmentalSelection}(CombinedPop)$ 
25:  Cập nhật tham số thích nghi ( $mean\_F, mean\_CR, P_{DE}$ ).
26: end while
```

4 Lưu đồ thuật toán



5 Kết luận và Thảo luận

Quá trình cải tiến thuật toán xuất phát từ việc nhận diện các hạn chế cốt lõi của NSGA-II nguyên bản, đặc biệt là việc sử dụng các tham số cố định thường dẫn đến hiệu quả tìm kiếm không đồng đều giữa các giai đoạn khác nhau của quá trình tiến hóa. Việc áp dụng cơ chế thích nghi để tự động điều chỉnh F và CR giúp giảm thiểu đáng kể công sức thử sai khi cấu hình thuật toán, đồng thời cho phép bộ giải tự động thay đổi hành vi tìm kiếm dựa trên phản hồi từ môi trường.

Bên cạnh đó, khả năng khám phá kém ở giai đoạn khởi tạo là một điểm yếu thường thấy khiến thuật toán dễ rơi vào trạng thái trì trệ (stagnation). Việc tích hợp OBL đã giải quyết vấn đề này bằng cách mở rộng vùng tìm kiếm ngay từ đầu, giúp cải thiện tốc độ hội tụ trong giai đoạn sớm. Song song đó, chiến lược khởi động lại (Restart Strategy) đóng vai trò như một cơ chế an toàn, giúp ngăn chặn hiện tượng hội tụ sớm tại các cực trị địa phương trong các bài toán đa đỉnh phức tạp. Tóm lại, các cải tiến này cung cấp sự cân bằng cần thiết giữa tính khai thác và khám phá, đảm bảo độ ổn định cao hơn khi giải quyết các lớp bài toán đa dạng.

Tài liệu

- [1] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.
(ResearchGate Article): https://www.researchgate.net/publication/352372869_A_Comprehensive_Review_on_NSGA-II_for_Multi-Objective_Combinatorial_Optimization_Problems
- [2] Tizhoosh, H. R. (2005). Opposition-based learning: a new scheme for machine intelligence. *Proceedings of International Conference on Computational Intelligence for Modelling, Control and Automation*.
(ResearchGate Article): https://www.researchgate.net/publication/4242497_Opposition-Based_Learning_A_New_Scheme_for_Machine_Intelligence
- [3] Zhang, J., & Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5), 945-958.
(ResearchGate Preprint): https://www.researchgate.net/publication/220367809_JADE
- [4] Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712-731.
(ResearchGate Article): https://www.researchgate.net/publication/3418989_MOEAD_A_Multiobjective_Evolutionary_Algorithm_Based_on-Decomposition