# KHẢO SÁT MỘT SỐ GIẢI THUẬT TIẾN HOÁ GIẢI BÀI TOÁN TỐI ƯU SỐ

Vũ Mạnh Xuân (*Khoa KH Tự nhiên & X ã hội - Đại học Thái Nguyên*)

## 1. Mở đầu

Các bài toán tối ưu số có nhiều ứng dụng thực tế và đã được nghiên cứu từ lâu. Trong các kỹ thuật tính toán mềm, thuật toán di truyền (GA – Genetic Algorithm) với các biến thể của nó và thuật toán mô phỏng tôi luyện (SA – Simulated Annealing) là những thuật toán chủ yếu sử dụng để giải các bài toán tối ưu. Một cách tổng quát, một bài toán tối ưu số có thể xem là một cặp (S, f), trong đó $S \subseteq R^n$ và $f : S \rightarrow R$ là một hàm n biến. Bài toán đặt ra là tìm véc tơ $x = (x_1, x_2, ..., x_n) \in S$ sao cho f(x) đạt giá trị cực tiểu trên S, nghĩa là với mọi $y \in S$ phải có $f(x) \leq f(y)$. Hàm f ở đây có thể không liên tục nhưng cần bị chặn trên S.

Bài báo này tổng hợp những nét cơ bản của các thuật toán: Giải thuật di truyền (GA), chiến lược tiến hoá (ES - Evolution Straitegy) và giải thuật mô phỏng tôi luyện (SA). Các kết quả thử nghiệm cả 3 thuật toán trên cho một số hàm nhiều biến nhằm rút ra những điểm mạnh của từng thuật toán đối với mỗi loại bài toán cụ thể.

Bài báo được cấu trúc như sau: Phần kế tiếp trình bày khái quát các thuật toán cơ bản đã nêu. Sau đó là kết quả thử nghiệm các thuật toán này cho một số hàm cụ thể. Phần cuối cùng là một số nhận xét và kết luận.

## 2. Các thuật toán cơ bản

### Thuật toán di truyền (GA - Genetic Algorithm)

Thuật toán di truyền (GA) và các biến thể của nó đã được phát triển rất mạnh trong những năm gần đây. GA mô phỏng quá trình tiến hoá tự nhiên và sử dụng các thuật ngữ thông dụng của tự nhiên như lai ghép, đột biến, ... . GA cổ điển sử dụng mã hoá nhị phân, mỗi nhiễm sắc thể được mã hoá bởi một chuỗi bít nhị phân. Các toán tử sử dụng trong quá trình tiến hoá gồm chọn lọc, lai ghép và đột biến. Trong các bài toán tối ưu hàm nhiều biến, GA thường được sử dụng chủ yếu với mã hoá số thực (RCGA _ Real Coded Gêntic Algorithm). Mỗi nhiễm sắc thể được mã hoá bởi một véc tơ gồm n thành phần thực. Như vậy có thể xem một quần thể có m cá thể như một ma trận thực cấp mxn. Với cách mã hoá này các toán tử lai ghép và đột biến có nhiều dạng khác nhau. Có thể mô tả vắn tắt thuật toán như sau ([1], [3]):

B1) Khởi tạo ngẫu nhiên quần thể có m cá thể.

B2) Lặp khi điều kiện dừng chưa thoả

Chọn các cá thể cha mẹ để tiến hoá.

Thực hiện lai ghép các cá thể cha mẹ để tạo sinh các cá thể con

Thực hiện đột biến một cá thể con với xác suất $p_m$

Chọn các cá thể sống sót cho lần tạo sinh tiếp theo

B3) Cá thể có độ thích nghi cao nhất trong lần tạo sinh cuối cùng là cá thể cần tìm.

### Chiến lược tiến hoá (ES – Evolution Straitegy)

Chiến lược tiến hoá (ES) là một biến thể của GA, ES cũng sử dụng mã hoá số thực giống như trong GA, song quá trình tiến hoá thường chỉ sử dụng một loại toán tử di truyền là phép đột

biến. Mỗi cá thể trong ES là một cặp véc tơ thực $(x_i, \eta_i)$ với i=1,..., μ. Véc tơ x = $(x_i)$ chỉ một lời giải của bài toán; còn mỗi $\eta_i$ tính phân bố của từng thành phần $x_i$ tương ứng.

Có thể mô tả ES như sau ([2], [5]):

B1) Khởi tạo quần thể ban đầu gồm μ cá thể $(x_i, \eta_i)$; i=1,..., μ.

Thiết lập biến đếm k=1.

B2) Tính độ thích nghi mỗi cá thể $(x_i, \eta_i)$ dựa trên hàm mục tiêu f(x).

B3) Với mỗi cá thể cha mẹ $(x_i, \eta_i)$, tạo các cá thể con $(x'_i, \eta'_i)$ như sau

$$\eta'_i(j) = \eta_i(j)\exp(\tau' N(0,1) + \tau N_j(0,1)) \quad (1)$$

$$x'_i(j) = x_i(j) + \eta'_i(j) N_j(0,1) \quad (2)$$

ở đây $x_i(j)$, $x'_i(j)$, $\eta_i(j)$ và $\eta'_i(j)$ là thành phần thứ j của các véc tơ $x_i$, $x'_i$ , $\eta_i$ và $\eta'_i$ tương ứng. N(0,1) là số ngẫu nhiên trong [0, 1]. $N_j(0,1)$ là số ngẫu nhiên trong [0,1] ứng với mỗi j. Các tham số τ' và τ thường được chọn là $(\sqrt{2\sqrt{n}})^{-1}$ và $(\sqrt{2n})^{-1}$ . Số các cá thể con được tạo ra là λ.

B4) Tính độ thích nghi mỗi cá thể vừa tạo.

B5) Chọn trong μ cá thể trong quần thể gồm cả cha mẹ và các con vừa tạo để làm cha mẹ cho lần tạo sinh kế tiếp.

B6) Dừng nếu thoả điều kiện dừng. Nếu không k=k+1, quay lại bước 3.

Tuỳ thuộc vào cách chọn tạo sinh mà người ta gọi tên ES, chẳng hạn nếu chọn μ cá thể cho lần sinh kế tiếp chỉ từ λ con mới tạo ra (λ>μ) thì ta gọi là (μ, λ) ES; nếu μ cá thể được chọn từ (μ+λ) cá thể (cả cha mẹ lẫn các con) thì gọi là (μ+λ) ES.

**Thuật toán mô phỏng tôi luyện (SA – Simulated Annealing)**

SA là thuật toán mô phỏng quá trình tôi luyện thép, xuất phát từ nhiệt độ ban đầu $T_0$ và một cá thể khởi tạo, tiến hành tạo sinh cá thể mới; quá trình này kết thúc nếu nhiệt độ giảm đến một ngưỡng (độ đông) nào đó. SA được mô tả như sau ([4], [7]):

B1) Khởi tạo ngẫu nhiên một cá thể X;

B2) Khởi tạo tham số nhiệt độ ban đầu $T_0$;

B3) k=1 ; $T_k = T_0$;

B4) Tiến hành các thao tác sau :

Y = generate(X, $T_k$);

Nếu accept(X, Y, $T_k$) thì X = Y ;

$T_{k+1}$ = update($T_k$);

k=k+1;

B5) Nếu điều kiện dừng chưa thoả, quay lại bước 4.

Thuật toán trên sử dụng các hàm sau :

Hàm generate(X, $T_k$) là hàm tạo sinh một cá thể từ X với nhiệt độ hiện tại là $T_k$ . Hàm này phụ thuộc vào xác suất $G_{XY}$ ($T_k$) có hàm mật độ xác suất là $g_{XY}(\Delta X)$ (trong đó ΔX = Y - X). Hàm accept(X,Y,$T_k$) được quyết định bởi xác suất $A_{XY}(T_k)$ cho phép chấp nhận Y thay thế X tại thời điểm có nhiệt độ $T_k$ . Hàm update($T_k$) là hàm giảm nhiệt độ còn gọi là sơ đồ tôi luyện. Có nhiều dạng hàm $g_{XY}(\Delta X)$ cũng như lịch trình tôi luyện đã được giới thiệu trong [4], [7].

## 3. Kết quả thử nghiệm

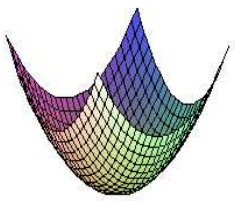
### 3.1 Các hàm thử nghiệm

Chúng tôi chọn 6 hàm sau để tiến hành thử nghiệm. Các hàm này đã được giới thiệu trong [2], [4], [5] .

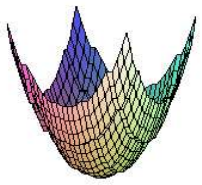
Các hàm thử nghiệm đều được xét với số chiều n=30; miền S được giới hạn cho mỗi biến $x_i$ và cho trong bảng. Giá trị $f_{min}$ là giá trị đúng tính trong miền xác định tương ứng với mỗi hàm.

| Test Function | n | S | $f_{min}$ |
|---|---|---|---|
| $f_1 = \sum_{i=1}^{n} x_i^2$ | 30 | [-10,10] | 0 |
| $f_2 = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$ | 30 | [-10,10] | 0 |
| $f_3 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$ | 30 | [-10,10] | 0 |
| $f_4 = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$ | 30 | [-10, 10] | 0 |
| $f_5 = \sum_{i=1}^{n} (-x_i \sin \sqrt{\lvert x_i \rvert})$ | 30 | [-500,500] | -12569.5 |
| $f_6 = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$ | 30 | [-10, 10] | 0 |

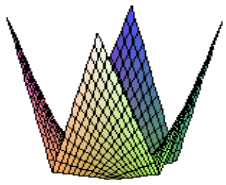
Các hàm nêu trên có đồ thị trong trường hợp số chiều n=2 được mô tả dưới đây:



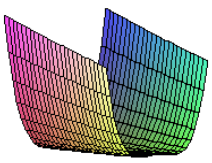
*Hình 1. Đồ thị hàm f1*



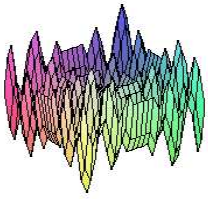
*Hình 2. Đồ thị hàm f2*



*Hình 3. Đồ thị hàm f3*



*Hình 4. Đồ thị hàm f4*



*Hình 5. Đồ thị hàm f5*

*Hình 6. Đồ thị hàm f6*

### 3.2 Thiết lập tham số thử nghiệm

#### 3.2.1 Thuật toán di truyền

Trong bài báo này, GA sử dụng mã hoá số thực, kích cỡ quần thể được cố định m=30. Để tiện so sánh, chỉ một toán tử lai ghép được sử dụng. Qua thử nghiệm, chúng tôi thấy toán tử lai ghép số học ([3], [6], [8]) có độ ổn định cao và khả năng hội tụ tương đối tốt nên được chọn sử dụng trong thử nghiệm này. Tại mỗi lần tạo sinh chỉ một cá thể con sinh ra, cá thể con này sẽ thay thế cá thể cha hoặc mẹ nếu nó tốt hơn. Chương trình tiến hành 20 lần chạy độc lập, mỗi lần tạo sinh 2000 thế hệ, giá trị tốt nhất của hàm mục tiêu được ghi nhận lại và trình bày trong các bảng dưới đây.

#### 3.2.2 Chương trình tiến hoá

Chương trình tiến hoá dùng trong thử nghiệm này là (1+1)ES. Quần thể khởi tạo ban đầu gồm 1 cá thể. Tại mỗi lần lặp, cá thể này sinh ra 1 con, sau đó từ cá thể ban đầu và cá thể con tạo được ta chọn cá thể tốt nhất (giá trị hàm mục tiêu nhỏ nhất) cho lần tạo sinh kế tiếp. Trong thử nghiệm chúng tôi sử dụng công thức (1') sau thay cho công thức (1) (xem [5])

$$\eta'_i(j) = \eta_i(j)\exp(\tau N(0,1)) \qquad (1')$$

Việc thử nghiệm cũng được tiến hành 20 lần chạy độc lập, mỗi lần chạy thực hiện lặp tạo sinh 2000 lần. Sau mỗi lần chạy, giá trị hàm mục tiêu của cá thể tốt nhất trong lần lặp cuối cùng được ghi nhận. Trung bình cộng của các lần chạy được tính và trình bày trong bảng.

#### 3.2.3 Thuật toán mô phỏng tôi luyện

Thuật toán mô phỏng tôi luyện dùng trong thử nghiệm này là một dạng cải biên của thuật toán tôi luyện rất nhanh VFSA (Very Fast Simulated Annealing) theo [4], [7].

Ký hiệu $X(k) = (x_1(k), x_2(k), \ldots, x_n(k))$ là véc tơ thu được tại bước thứ k, trong đó với mỗi $i = 1, \ldots, n$ có $x_i(k) \in [L_i, R_i]$.

Khi đó $x_i(k+1)$ được tính bởi số ngẫu nhiên $z_i \in [-1,1]$ bởi:

$x_i(k+1) = x_i(k) + z_i(R_i - L_i)$

Hàm mật độ của $z_i$ là $$g_k(z_i) = \frac{1}{2(|z_i| + T(k))\ln(1 + 1/T(k))}$$

Xác suất chấp nhận $A_{XY}(T_k)$ trong trường hợp này là

$$A_{XY}(T_k) = \min\left\{1, \exp\left(-\frac{f(Y) - f(X)}{T_k}\right)\right\}$$

Sơ đồ tôi luyện sử dụng ở đây là $T(k) = T_0 \exp(-bk^{1/n})$ trong đó b là một hằng số dương. Trong thử nghiệm chúng tôi lấy b = 7. Nhiệt độ ban đầu $T_0 = 800$, nhiệt độ "đông" $T_k = 0.1$.

### 3.3 Kết quả thử nghiệm

Các bảng sau trình bày kết quả thử nghiệm. Chương trình được viết bằng MatLab. Các tham số được chọn như đã nêu trên nhằm làm cho thời gian chạy của mỗi thuật toán xấp xỉ nhau để dễ so sánh.

Cấu trúc các bảng như sau: Cột (1) trong các bảng cho giá trị trung bình hàm mục tiêu của cá thể tốt nhất sau 20 lần chạy. Cột (2) là giá trị hàm mục tiêu nhỏ nhất tìm được trong 20 lần chạy này. Cột (3) là thời gian trung bình (giây) cho mỗi lần chạy của từng thuật toán tương ứng.

*Bảng 1 : Kết quả thử nghiệm với các hàm $f_1$ , $f_2$ .*

| | Hàm $f_1$ | | | Hàm $f_2$ | | |
|---|---|---|---|---|---|---|
| | (1) | (2) | (3) | (1) | (2) | (3) |
| GA | 19.2932 | 11.466 | 0.634 | 18.7 | 12 | 0.767 |
| ES | 981.37 | 746.28 | 0.552 | 898.9 | 542 | 0.5975 |
| SA | 941.140 | 640.13 | 0.7255 | 1017.35 | 735 | 0.756 |

*Bảng 2 : Kết quả thử nghiệm với các hàm $f_3$ , $f_4$ .*

| | Hàm $f_3$ | | | Hàm $f_4$ | | |
|---|---|---|---|---|---|---|
| | (1) | (2) | (3) | (1) | (2) | (3) |
| GA | 9.47661 | 6.1968 | 0.7405 | 702.8395 | 365.58 | 0.6735 |
| ES | 4.36E+09 | 73.981 | 0.632 | 3.25E+05 | 135230 | 0.555 |
| SA | 3.65E+08 | 139.78 | 0.7915 | 2.92E+05 | 90911 | 0.755 |

*Bảng 3 : Kết quả thử nghiệm với các hàm $f_5$ , $f_6$.*

| | Hàm $f_5$ | | | Hàm $f_6$ | | |
|---|---|---|---|---|---|---|
| | (1) | (2) | (3) | (1) | (2) | (3) |
| GA | -2757.82 | -4342.8 | 0.9585 | 197.297 | 146.09 | 0.8105 |
| ES | -1112.84 | -2632 | 0.6915 | 1235.029 | 994.98 | 0.6585 |
| SA | -2890 | -5005.7 | 0.8815 | 1254.722 | 886.65 | 0.775 |

## 4. Nhận xét và kết luận

Căn cứ kết quả trong các bảng nêu trên có thể rút ra một số nhận xét sau :

Các hàm $f_1$ , $f_2$ , $f_3$ , $f_4$ là các hàm đơn phương thức (trong đó hàm $f_2$ là hàm dạng bậc thang, không liên tục), giá trị $f_{min}$ đều là 0. Kết quả trong các bảng 1 và 2 cho thấy nói chung GA cho kết quả tốt nhất, có độ ổn định cao (độ lệch giữa giá trị trung bình các lần chạy với giá trị tốt nhất tìm được). Thời gian chạy cũng như giá trị tốt nhất tìm được sau 20 lần chạy của GA là tốt hơn cả.

Các hàm $f_5$ , $f_6$ là các hàm đa phương thức, có nhiều cực trị địa phương. Số cực trị địa phương tăng nhanh theo số chiều của không gian. Rất đáng ngạc nhiên là với hàm $f_5$ thì SA lại tốt hơn cả mặc dù thời gian chạy lâu hơn. Ngay cả nếu tăng số lần lặp thì ES và GA đều không tốt hơn SA trong trường hợp này. Về các tham số của thuật toán, chúng tôi cũng đã thử thay đổi số lần lặp của GA, ES cũng như nhiệt độ $T_o$ , nhiệt độ "đông" của SA, song về cơ bản các tham số đã sử dụng trên cho kết quả tốt hơn cả.

Như vậy so sánh 3 thuật toán trên, nói chung GA có độ ổn định tốt, có thể do cách làm việc là trên quần thể nhiều cá thể trong khi ES và SA chỉ tiến hành trên một cá thể tại mỗi bước tạo sinh. Tuy nhiên nếu thay đổi các tham số, chẳng hạn xét với (2+3) ES thì kết quả lại tốt hơn nhiều, ngoại trừ trường hợp hàm $f_5$. SA nói chung không cho kết quả tốt như 2 thuật toán trên, song có những trường hợp lại rất tốt như đối với hàm $f_5$.

Có thể đặt ra vấn đề nghiên cứu tích hợp các thuật toán này để tìm giải pháp tốt hơn. Trong [7] tác giả đã tích hợp SA với GA đối với toán tử đột biến và cũng đã thu được kết quả tốt. Chúng tôi cho rằng nếu tích hợp yếu tố nhiệt độ với các toán tử lai ghép hay chọn lọc của GA có thể làm đa dạng hơn quần thể và tránh được hiện tượng hội tụ sớm. Cũng có thể từ quần

thể khởi tạo ban đầu, áp dụng SA cho mỗi cá thể rồi sau đó sử dụng GA hoặc ngược lại: chạy GA trước rồi sử dụng tiếp SA hay ES nhằm tăng tốc độ hội tụ là những vấn đề có thể phát triển tiếp📖

**TÓM TẮT**

Bài báo này khảo sát và so sánh ba giải thuật trong tính toán mềm là: Giải thuật di truyền (GA), Chiến lược tiến hoá (ES) và Giải thuật mô phỏng tôi luyện (SA) trong việc giải những bài toán tối ưu số. Qua đó ta thấy được các điểm mạnh của mỗi giải thuật nhằm chọn lựa giải thuật phù hợp hoặc tìm cách kết hợp các giải thuật này để nâng cao hiệu suất tính toán trong những bài toán cụ thể.

**SUMMARY**

**Investigate several Evolution Strategies for solving the problem of numeral optimization**

This paper investigates and compares three algorithms in soft-computing: Genetic Algorithm (GA), Evolutionary Strategy (ES) and Simulated Annealing (SA) in solving the problems of numerical optimization. Thanks to the results, we understand advantages of each algorithm to choose the most suitable algorithms or combine these algorithms to enhance productivity computing in concretize problems.

**TÀI LIỆU THAM KHẢO**

[1] F.Herrera, M. Lozano (1997), *Gradual Distributed Real-Coded Genetic Algorithms*, Technical Report #DECSAI-97-01-03.

[2] Ko-Hsin Liang, Xin Yao, Charles S. Newton (2000), *Adapting Self-adaptive Parameters in Evolutionary Algorithms.*

[3] Ulrich Bodenhofer (2004), *Genetic Algorithms: Theory and Applications.*

[4] Xin Yao (1995), *A New Simulated Annealing Algorithm*, Published in International Journal of Computer Mathematics.

[5] Xin Yao and Yong Lui (1997), *Fast Evolution Strategies.*

[6] Nguyễn Đình Thúc (2001), *Lập trình tiến hoá*, Nxb GD, Hà Nội.

[7] Trần Ngọc Hà (2003), *Các hệ thống thông minh lai ứng dụng trong xử lý dữ liệu*, Luận án Tiến sĩ, Trường ĐH Bách khoa Hà Nội.

[8] Vũ Mạnh Xuân, Nguyễn Thanh Thuỷ (2006), *Biểu diễn toán tử lai ghép trong giải thuật di truyền mã hoá số thực*, Tạp chí "Khoa học và Công nghệ", Đại học Thái Nguyên, số 1 (37), tập 2, 2006 (tr 52).

Keyword: Genetic Algorithm, Simulated Annealing, Evolution Straitegy.