Tạp chí Tin học và Điều khiển học, T.22, S.2 (2006), 134–140

GIẢI THUẬT DI TRUYỀN MÃ HÓA SỐ THỰC VỚI TOÁN TỬ LAI GHÉP SBX

VŨ MẠNH XUÂN[1], NGUYỄN THANH THUỶ[2]

[1]*Khoa Khoa học tự nhiên, Đại học Thái Nguyên*
[2]*Khoa Công nghệ thông tin, Trường Đại học Bách khoa Hà Nội*

Abstract. Simulated Binary Crossover operator (SBX) in Real-Coded Genetic Algorithm (RCGA) was introduced in 1995 by K. Deb and R. B. Agrawal. In recent years, many studies about RCGA have been used as the basis genetic operator. The purpose of this paper is to investigate some other probability distributions in order to control the β of SBX.

Tóm tắt. Toán tử lai ghép SBX (Simulated Binary Crossover) trong giải thuật di truyền mã hóa số thực (RCGA - Real-Code Genetic Algorithm) được K. Deb và R. B. Agrawal giới thiệu năm 1995. Trong vài năm gần đây, nhiều nghiên cứu về RCGA đã sử dụng SBX làm toán tử di truyền cơ bản. Mục đích của bài báo này khảo sát một số phân phối xác suất khác để điều chỉnh tham số β của toán tử SBX.

MỞ ĐẦU

Toán tử lai ghép là một trong các toán tử cơ bản của giải thuật di truyền (GA - Genetic Algorithm) và khi sử dụng mã hóa số thực (RCGA) thì có nhiều dạng khác nhau được nghiên cứu và giới thiệu. Năm 1995, K. Deb và các cộng sự đã đề xuất toán tử lai ghép SBX (Simulated Binary Crossover) và toán tử này đã được sử dụng trong nhiều nghiên cứu khác. Khác với các dạng kinh điển của toán tử lai ghép, trong SBX các cá thể con sinh ra không sử dụng trực tiếp các giá trị thành phần của cá thể cha mẹ mà nhận một giá trị nào đó được điều khiển bởi một tham số β xác định tỷ lệ khoảng cách cha mẹ và con cái. Dựa trên việc điều chỉnh tham số β, tính đa dạng của quần thể được tăng cường và khả năng hội tụ của giải thuật cao hơn. Nhiều nghiên cứu gần đây cũng quan tâm đến các thuật toán lai các kỹ thuật tìm kiếm địa phương với các toán tử di truyền theo hướng tương tự (Deb, Anand và Joshi, 2002; Dietzfelbinger, Naudts, Van Hoyweghen và Wegerner, 2003).

Trong [1], các tác giả đã đề xuất toán tử lai ghép SBX sử dụng một hàm phân phối xác suất để tính giá trị tham biến điều khiển β. Các kết quả thực nghiệm đã chứng tỏ toán tử này cho hiệu quả tốt hơn nhiều so với các toán tử lai ghép truyền thống khác. Tuy nhiên như các tác giả [1] đã nhận xét, có thể chọn các hàm phân phối khác để điều chỉnh tham số của toán tử này. Tiếp tục ý tưởng đó, chúng tôi tiến hành khảo sát một số hàm phân phối xác suất thông dụng khác và thử nghiệm qua một số bài toán tối ưu chuẩn đã được nhiều tác giả đề cập đến ([1–5]). Các kết quả thử nghiệm được trình bày tường minh để độc giả có thể theo dõi và chọn lựa giải pháp hợp lý.

Bài báo được cấu trúc như sau: Phần thứ nhất trình bày cụ thể toán tử SBX đã được giới thiệu trong [1] và phân tích tham số điều khiển của toán tử này. Phần thứ hai đưa ra khảo sát một số hàm phân phối xác suất khác. Phần thứ ba trình bày kết quả thử nghiệm trên một số bài toán tối ưu với các hàm phân phối đã nêu. Phần thứ tư là nhận xét và kết luận.

1. GIẢI THUẬT DI TRUYỀN VỚI TOÁN TỬ LAI GHÉP SBX

Toán tử SBX là toán tử lai ghép áp dụng cho giải thuật di truyền mã hóa số thực (RCGA - Real-Code Genetic Algorithm), tạo hai cá thể con từ một cặp cá thể cha mẹ đã chọn. Dưới đây xin trình bày lại thuật toán đã được giới thiệu trong [1].

Giả sử $x = (x_1, x_2, ..., x_n)$ và $y = (y_1, y_2, ..., y_n)$ là hai cá thể cha mẹ được chọn để tạo sinh. Khi đó hai cá thể con $c^1 = (c_1^1, ..., c_n^1)$ và $c^2 = (c_1^2, ..., c_n^2)$ được sinh ra theo công thức:

$$c_i^1 = 0{,}5 \times ((1 + \beta_q) \times x_i + (1 - \beta_q) \times y_i) \tag{1}$$

$$c_i^2 = 0{,}5 \times ((1 - \beta_q) \times x_i + (1 + \beta_q) \times y_i) \tag{2}$$

trong đó β_q được tính bởi công thức:

$$\beta_q = \begin{cases} (2u)^{\frac{1}{\eta+1}}, & \text{nếu } u \leqslant 0{,}5 \\ (\dfrac{1}{2(1-u)})^{\frac{1}{\eta+1}}, & \text{ngược lại} \end{cases} \tag{3}$$

với u là số ngẫu nhiên trong [0, 1], η là tham số điều khiển.

Toán tử SBX có thể được mô tả vắn tắt như sau:

Bước 1. Chọn ngẫu nhiên số thực $u \in [0, 1]$.

Bước 2. Tính β_q theo công thức (3).

Bước 3. Tính các cá thể con c^1 và c^2 theo công thức (1) và (2).

Lưu ý là có thể tính các thành phần của cá thể con với cùng một giá trị β_q hoặc với mỗi i, tính β_q ứng với thành phần thứ i một cách độc lập.

Trong các công thức (1) và (2), bằng cách gán cho β_q các giá trị thích hợp ta sẽ nhận được các dạng khác quen thuộc của toán tử lai ghép như lai ghép một điểm, lai ghép nhiều điểm, lai ghép mặt nạ (lai ghép đều) hay lai ghép số học ([6,9]). Thật vậy:

\+ *Lai ghép một điểm với điểm lai ghép là* t $(1 \leqslant t \leqslant n-1)$ ứng với cách chọn $\beta_k = -1$ với $0 < k < t$ và $\beta_k = 1$ trong đoạn còn lại.

\+ *Lai ghép nhiều điểm* ứng với cách đặt $\beta_k = -1$ trong từng đoạn con của $[1, n]$ và $\beta_k = 1$ tại các đoạn còn lại.

\+ *Lai ghép mặt nạ* ứng với cách đặt $\beta_k = -1$ hoặc $\beta_k = 1$ với một xác suất đều tại các giá trị $k \in [1, n]$.

\+ *Lai ghép trung bình* ứng với cách đặt $\beta_k = 0$ với mọi k.

\+ *Lai ghép số học* ứng với cách đặt $\beta_k = 2r - 1$ trong đó r là số ngẫu nhiên với phân phối đều trên (0, 1).

\+ *Lai ghép heuristic:* Giả sử x là cá thể tốt hơn y ($f(x) < f(y)$ với bài toán tìm min và $f(x) > f(y)$ với bài toán tìm max). Bằng cách đặt $\beta_k = 1 + 2d$ trong đó d là số ngẫu nhiên trong [0, 1] ta thu được con duy nhất tính bởi $c_k = x_k + d(x_k - y_k)$.

+ *Toán tử lai ghép BLX-α* ứng với cách đặt $\beta_k = 2r - 1$, trong đó r là số ngẫu nhiên được lấy trong $[-\alpha, 1+\alpha]$. Dạng biểu diễn thông thường của BLX-α là: Mỗi thành phần z_i của cá thể con được chọn theo phân phối ngẫu nhiên đều trong khoảng $[\min_k - I\alpha, \max_k + I\alpha]$ trong đó $\min_k = \min(x_k, y_k), \max_k = \max(x_k, y_k), I = \max_k - \min_k$. Tham số α thường được chọn là 0,5.

Mặt khác, từ các công thức (1) và (2) ta cũng có

$$c_i^1 - c_i^2 = \beta_q(x_i - y_i).$$

Như vậy, tham số β_q có vai trò điều khiển khoảng cách giữa các cá thể con sinh ra tùy theo khoảng cách giữa các cá thể cha mẹ chúng. Sự biến đổi của β_q làm tăng hay giảm khả năng các cá thể con sinh ra ở gần hay xa cha mẹ. Do đó, bằng cách thay đổi các giá trị η trong quá trình tính toán ta sẽ điều chỉnh được tính đa dạng của quần thể.

2. TÍNH TOÁN THAM SỐ β

Phát triển ý tưởng trên, chúng tôi sử dụng một số hàm phân phối xác suất khác và tiến hành thực nghiệm trên một số các bài toán tối ưu nhằm tìm ra một giải pháp tốt cho vấn đề điều khiển tham số để tính β_q thay cho (3). Thuật toán như đã nêu phần trên, ở đây chỉ thay đổi công thức (3) tính tham số β_q, việc tính các cá thể con vẫn theo (1) và (2).

- Phân phối chuẩn

Tính β tùy thuộc số ngẫu nhiên x theo công thức

$$\beta = \frac{1}{\sigma\sqrt{2\pi}} \exp[-\frac{1}{2}(\frac{x^2}{\sigma^2})]. \tag{4}$$

Phân phối này có đường cong mật độ dạng hình chuông nhận trục hoành làm tiệm cận. Trong tự nhiên, nhiều quy luật biến đổi theo phân phối chuẩn, chẳng hạn trọng lượng, chiều cao của người lớn, chỉ số thông minh của trẻ em, các sai số đo đạc, độ bền dẻo của máy móc,... Phân phối này có kỳ vọng EX = 0 và phương sai VX = σ^2. Việc tính β phụ thuộc vào chọn tham số σ, bằng thực nghiệm chúng tôi thấy nếu chọn σ nhỏ thì cho kết quả tốt.

- Phân phối Loga chuẩn

Trong trường hợp này β được tính theo số ngẫu nhiên x bởi công thức

$$\beta = \frac{1}{\sigma\sqrt{2\pi}} \frac{1}{x} \exp[-\frac{(\ln x)^2}{2\sigma^2}]. \tag{5}$$

Tương tự như phân phối chuẩn, một dạng khác được quan tâm đến là phân phối loga chuẩn. Phân phối này có kỳ vọng EX = $\exp(\sigma^2/2)$ và phương sai là VX = $\exp(\sigma^2/2) * (exp(\sigma^2) - 1)$.

- Phân phối mũ

Tính β tùy thuộc số ngẫu nhiên x theo công thức

$$\beta = \lambda \times \exp(-\lambda \times x). \tag{6}$$

Trong thực tế, nhiều đại lượng ngẫu nhiên phù hợp với phân phối mũ, chẳng hạn thời gian phục vụ của một hệ đám đông, thời gian phục vụ của một dụng cụ điện tử, sự phân rã của nguyên tử phóng xạ,... Phân phối mũ là trường hợp riêng của họ phân phối Veibull có nhiều ứng dụng trong lâm nghiệp. Phân phối này có kỳ vọng EX = $1/\lambda$ và phương sai là $1/\lambda^2$. Bằng thử nghiệm chúng tôi thấy α tốt nhất khoảng từ 20 đến 30.

3. KẾT QUẢ THỬ NGHIỆM

Nhằm kiểm nghiệm tác động các tham số của toán tử SBX như nêu trên, chúng tôi tiến hành làm thử nghiệm trên một số hàm đã được khảo sát trong [1–4].

Chương trình thử nghiệm được viết bằng phần mềm MatLab. Các bài toán thử nghiệm đều là tìm cực tiểu. Trong chương trình thử nghiệm chúng tôi chỉ sử dụng toán tử lai ghép (xác suất đột biến bằng 0).

Toán tử chọn lọc tạo sinh có tính tinh hoa, cụ thể như sau: Hai cá thể cha mẹ qua lai ghép tạo được hai cá thể con mới. Chọn hai cá thể có giá trị hàm mục tiêu nhỏ nhất trong bốn cá thể (hai con và hai cha mẹ) thay cho hai cá thể cha mẹ ban đầu.

Để đối sánh, chúng tôi sử dụng một toán tử lai ghép kinh điển là lai ghép đều (lai ghép mặt nạ) và toán tử lai ghép BLX-0.5. Với mỗi hàm thử nghiệm, cùng một quần thể khởi tạo, thực hiện tiến hóa sáu lần độc lập, mỗi lần chọn một dạng toán tử lai ghép SBX sử dụng một hàm mật độ khác nhau với mỗi toán tử, chương trình tiến hành lặp 10000 lần, cứ sau 1000 lần lặp thì ghi lại kết quả để khảo sát.

Kết quả trong các bảng dưới đây có cấu trúc như sau: Mỗi bảng gồm 10 cột số liệu, mỗi cột ghi giá trị hàm mục tiêu của cá thể tốt nhất tìm được tại bước lặp tương ứng (số lần lặp ghi trên dòng đầu tiên của bảng). Mỗi dòng của bảng ghi kết quả tiến hóa ứng với một dạng toán tử lai ghép, cụ thể là:

- Dòng thứ nhất là kết quả khi sử dụng toán tử lai ghép đều (lai ghép mặt nạ).
- Dòng thứ hai là kết quả khi sử dụng toán tử lai ghép BLX-0.5.
- Dòng thứ ba là sử dụng toán tử SBX trong đó β được tính theo (3), tham số η chọn bằng 0,1.
- Các dòng thứ tư, năm và sáu là kết quả tương ứng với toán tử SBX mà tham số β được tính lần lượt theo các công thức (4), (5) và (6) với các tham số là $\sigma = 0{,}07$ và $\lambda = 25$.

Các tham số η, σ, λ chọn như trên căn cứ nhiều thử nghiệm và thấy kết quả tương đối tốt hơn các giá trị khác.

3.1. Hàm Sphere

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \quad \text{với } n = 30, -5{,}12 \leqslant x_i \leqslant 5{,}12$$

Hàm này là hàm đơn phương thức, liên tục và lồi chặt, có một cực tiểu toàn cục tại $\alpha = (0, 0, ...)$ và giá trị cực tiểu $f_1(\alpha) = 0$. Kết quả thực nghiệm cho trong Bảng 1.

Bảng 1. Kết quả thử nghiệm hàm f_1 (Sphere)

	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
1	16.281	16.281	16.281	16.281	16.281	16.281	16.281	16.281	16.281	16.281
2	4.7537	4.7485	4.7485	4.7485	4.7485	4.7485	4.7485	4.7485	4.7485	4.7485
3	20.738	6.3076	3.6336	2.1296	1.4892	0.85226	0.46707	0.25184	0.15685	0.07644
4	0.41251	0.07133	0.04738	0.03891	0.03714	0.03706	0.01159	0.00249	0.0013	0.00128
5	0.14216	0.01482	0.00992	0.00685	0.00677	0.00676	0.00569	0.00569	0.00412	0.00219
6	0.57621	0.00895	0.00042	2.87E-05	1.18E-06	7.87E-08	3.49E-09	2.73E-10	1.46E-11	7.08E-13

3.2. Hàm Rosenbrok

$$f_2(x) = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad \text{với} \quad n = 30, -5{,}12 \leqslant x_i \leqslant 5{,}12.$$

Đây là một hàm liên tục, đơn phương thức với nhiều cực trị địa phương. Đã có nhiều tác giả quan tâm khảo sát hàm này, một trong những khó khăn chính là sự tác động qua lại phi tuyến giữa các biến. Hàm này có cực tiểu toàn cục tại $\alpha = (1, 1, ...)$ và giá trị hàm $f_2(\alpha) = 0$. Kết quả thử nghiệm đối với hàm Rosenbrok được cho trong Bảng 2.

Bảng 2. Kết quả thử nghiệm hàm f_2 (Rosenbrok)

	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
1	6438.4	6438.4	6438.4	6438.4	6438.4	6438.4	6438.4	6438.4	6438.4	6438.4
2	1867.7	1866.4	1866.4	1866.4	1866.4	1866.4	1866.4	1866.4	1866.4	1866.4
3	7979.9	2423.9	740.79	233.96	131.34	89.684	68.168	51.852	41.819	36.05
4	142.03	62.748	53.668	52.381	51.599	51.518	51.44	41.928	41.899	41.858
5	126.09	83.315	65.069	57.931	33.142	30.727	30.646	30.501	30.124	30.111
6	86.637	32.596	28.634	28.26	28.035	27.854	27.735	27.599	27.526	27.409

3.3. Hàm Schwefel

$$f_3(x) = \sum_{i=1}^{n}(\sum_{j=1}^{n} x_j)^2 \quad \text{với} \quad n = 30, -5{,}12 \leqslant x_i \leqslant 5{,}12.$$

Đây là một hàm liên tục, đơn phương thức với nhiều cực trị địa phương. Điều khó khăn nhất đối với hàm này là gradient không hướng theo trục tọa độ. Hàm này có cực tiểu toàn cục tại $\alpha = (0, 0, ...)$ và giá trị hàm $f_3(\alpha) = 0$. Kết quả thử nghiệm được cho trong Bảng 3.

Bảng 3. Kết quả thử nghiệm hàm f_3 (Schwefel)

	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
1	101.21	58.051	38.558	38.558	38.558	38.558	38.558	38.558	38.558	38.558
2	36.903	36.867	36.867	36.867	36.867	36.867	36.867	36.867	36.867	36.867
3	76.323	58.272	45.801	37.214	34.304	29.913	24.716	20.079	18.803	16.55
4	5.1747	3.8968	3.4806	3.2492	3.0733	2.8831	2.8341	2.7433	2.668	2.6118
5	7.877	6.4855	5.8245	5.2824	4.7924	4.5289	4.304	4.1516	4.0691	4.0034
6	7.5504	5.2028	3.6491	3.1052	2.5327	2.2207	1.9759	1.7219	1.5817	1.416

3.4. Hàm Rastringin

$$f_4(x) = 10.n + \sum_{i=1}^{n}(x_i^2 - 10.\cos(2\pi x_i)) \quad \text{với} \quad n = 30, -5{,}12 \leqslant x_i \leqslant 5{,}12.$$

Hàm Rastringin là dạng hàm liên tục, đa phương thức, có rất nhiều cực trị địa phương và có cực tiểu toàn cục tại $\alpha = (0, 0, ...)$ và $f_4(\alpha) = 0$. Kết quả thử nghiệm cho trong Bảng 4.

Bảng 4. Kết quả thử nghiệm hàm f_4 (Rastringin)

	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
1	112.24	112.24	112.24	112.24	112.24	112.24	112.24	112.24	112.24	112.24
2	109.19	108.72	108.72	108.72	108.72	108.72	108.72	108.72	108.72	108.72
3	226.12	202.22	169.6	148.52	130.79	100.44	86.214	86.214	76.963	76.302
4	190.39	81.075	45.307	19.718	13.561	12.926	12.84	12.821	12.817	12.536
5	193.05	167.86	151.05	128.24	113.91	82.179	79.06	78.995	78.825	78.821
6	136.64	111.82	103.85	73.814	51.86	39.5	32.842	31.591	30.701	26.404

3.5. Hàm Griewangk

$$f_5(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1 \quad \text{với} \quad n = 30, -5{,}12 \leqslant x_i \leqslant 5{,}12.$$

Hàm Griewangk là hàm liên tục, đa phương thức. Hàm này rất khó tối ưu vì nó không phân ly.f_5 có cực tiểu toàn cục tại $\alpha = (0, 0, ...)$ và $f_5(\alpha) = 0$. Kết quả thử nghiệm cho trong Bảng 5.

Bảng 5. Kết quả thử nghiệm hàm f_5 (Griewangk)

	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
1	18.534	18.534	18.534	18.534	18.534	18.534	18.534	18.534	18.534	18.534
2	5.8434	5.8314	5.8314	5.8314	5.8314	5.8314	5.8314	5.8314	5.8314	5.8314
3	15.307	7.9454	3.5889	1.9034	0.75165	0.25695	0.11711	0.06927	0.0351	0.02078
4	0.72336	0.20178	0.15249	0.10969	0.10667	0.10507	0.10505	0.10503	0.10477	0.10476
5	0.5911	0.11398	0.02912	0.01688	0.01289	0.01078	0.01038	0.01034	0.01013	0.01011
6	0.75766	0.0471	0.00187	9.57E-05	6.02E-06	3.68E-07	1.45E-08	1.02E-09	7.10E-11	3.52E-12

4. NHẬN XÉT VÀ KẾT LUẬN

Từ các bảng kết quả trên, chúng ta có một số nhận xét như sau:

- Các toán tử lai ghép kinh điển như lai ghép mặt nạ, BLX-α thường chỉ có tác dụng đối với số lần tạo sinh nhỏ, với số lần tạo sinh lớn thì không tìm được cá thể tốt hơn (xem cột thứ nhất trong các bảng ứng với lần lặp thứ 1000 và cột cuối ứng với lần lặp thứ 10000). Mặc dù giải thuật di truyền còn một số tham số tác động khác như xác suất đột biến, kích cỡ quần thể, song nếu chỉ sử dụng toán tử lai ghép thì cả các dạng lai kinh điển khác như lai số học, lai một điểm, lai đa điểm cũng tương tự như lai ghép đều. Trong khi toán tử SBX với cách chọn tham số điều khiển β như nêu trên thì vẫn tạo được các cá thể tốt hơn nếu tiếp tục tăng số lần tạo sinh.

- Trong các phân phối xác suất chọn để tính β thì phân phối mũ cho kết quả nói chung là tốt hơn các phân phối khác (dòng cuối cùng trong các bảng trên), đặc biệt đối với các hàm Sphere và Griewangk.

Khi tiến hành thử nghiệm cùng các hàm trên với miền xác định của mỗi biến không đối xứng như trường hợp trên và lời giải đúng không nằm giữa mà gần biên của miền xác định (chẳng hạn $-1 \leqslant x_i \leqslant 12$ với mọi i) thì kết quả cũng không biến động mấy so với các bảng đã trình bày ở trên. Nói chung tính β theo phân phối mũ vẫn cho kết quả tốt nhất.

Cũng cần nói thêm rằng do nhiều yếu tố ngẫu nhiên mà các kết quả thu được sau mỗi lần thực hiện thuật toán là không trùng nhau, song nhìn chung các kết quả này dao động không nhiều và tương đối ổn định cho mỗi dạng toán tử sử dụng.

Như vậy có thể thấy tham số β có tác động đáng kể đến sự hội tụ của giải thuật di truyền qua các toán tử lai ghép tạo hai con từ một cặp cha mẹ. Gần đây, đã có nhiều nghiên cứu cải tiến các tham số điều khiển các toán tử nhằm tăng tính đa dạng của quần thể (xem [5]). Một số hướng nghiên cứu khác là tìm cách tích hợp nhiều kỹ thuật tìm kiếm như giải thuật di truyền kết hợp với thuật toán leo đồi hoặc các kỹ thuật tìm kiếm địa phương (local search) khác. Việc tiếp tục nghiên cứu, thử nghiệm và chọn cách điều chỉnh các tham số sao cho có hiệu quả tốt và tích hợp với các toán tử khác, các kỹ thuật khác nhằm nâng cao hiệu quả của giải thuật là việc làm cần thiết và có ý nghĩa. Ngoài ra các nghiên cứu về toán tử lai ghép nhiều cha mẹ cũng đang phát triển và tỏ ra khả quan trong RCGA.

TÀI LIỆU THAM KHẢO

[1] Kalyanmoy Deb and Hans-Georg Beyer, Self-Adaptive Genetic Algorithms with Simulated Binary Crossover, *Technical Report* No. CI-61/99, March, 1999.

[2] Kalyanmoy Deb, Dhiraj Joshi, and Ashish Anand, Real-coded evolutionary algorithms with parent-centric recombination, *KanGAL Report* No. 2001003, 2001.

[3] Ko-Hsin Liang, Xin Yao, Charles S. Newton, *Adapting Self-adaptive Parameters in Evolutionary Algorithms*, 1998.

[4] F. Herrera, M. Lozano, Gradual Disrtibuted Real-Code Genetic Algorithms, *Technical Report* #DECSAI-97-01-03, 1997

[5] Manuel Lozano, Francisco Herrera, Natalio Krasnogor, Daniel Molina, Real-coded memetic algorithms with crossover Hill-Climbing, *Evolutionary Computatin* **12** (3) (2004) 273–302.

[6] Osamu Takahashi, Hajime Kita, Shigenobu Kobayashi, *A Real-Coded Genetic Algorithm using Distance Dependent Alternation Model for Complex Function Optimization*, GECCO, 2000.

[7] Unlrich Bodenhofer, Genetic Algorithms : Theory and Applications, *Lecture Notes, 2003/2004.*

[8] Tống Đình Quỳ, *Giáo trình Xác suất Thống kê*, NXB Giáo dục, 1999.

[9] Nguyễn Đình Thúc, *Lập trình tiến hóa*, NXB Giáo dục, 2001.

Nhận bài ngày 9 - 3 - 2006