Tính toán tối ưu sử dụng thuật toán lai tạo thành bởi thuật toán tiến hóa vi sai và tối ưu suy giảm độ dốc

Optimization using a hybrid algorithm based on differential evolution and gradient descent method

***Nguyễn Ngọc Sơn, **Hồ Phạm Huy Ánh, **Trương Đình Châu**
***ĐH Công Nghiệp TP Hồ Chí Minh, **DCSELAB, ĐH Bách Khoa TP Hồ Chí Minh**
e-Mail: nguyenngocson@iuh.edu.vn

Tóm tắt
Trong bài báo này, tác giả đề xuất thuật toán lai HDE được tạo thành bằng cách lai ghép thuật toán tiến hóa vi sai cơ bản DE và thuật toán suy giảm độ dốc GD. Đầu tiên, chúng tôi sử dụng khả năng tìm kiếm toàn cục của thuật toán DE để tìm giải pháp tối ưu trong không gian tìm kiếm và sau đó sử dụng khả năng tìm kiếm cục bộ chính xác của thuật toán GD để tăng tốc độ hội tụ. Chất lượng của thuật toán HDE được kiểm chứng trên một số hàm benchmark và được so sánh với các thuật toán khác như thuật toán tối ưu hóa bầy đàn PSO, thuật toán tiến hóa vi sai cơ bản DE. Các kết quả này chứng tỏ chất lượng và hiệu quả của thuật toán HDE.
Từ khóa: Thuật toán di truyền, tối ưu tiến hóa vi sai, thuật toán suy giảm độ dốc, bài toán tối ưu hóa.

Abstract: In this paper, we propose a hybrid algorithm HDE is created by combining a traditional differential evolution algorithm and a gradient descent method. Initially, we use global searching ability of the DE algorithm to find the global optimal solution in the search space and then precise local seaching of the GD method in that region to converge to optimal solution. The performance of HDE algorithm is tested on some benchmark functions and is compared with the other algorithms such as particle swarm optimization and traditional differential evolution. These results demonstrate the performance and efficiency of HDE algorithm.
Keywords: Genetic algorithm, differential evolution, gradient descent method, optimization problem.

Ký hiệu

Ký hiệu	Ý nghĩa
CR	xác suất lai ghép
F	hệ số đột biến
NP	Kích thước quần thể

Chữ viết tắt

EAs	evolution algorithms
DE	differential evolution
GD	gradient descent
GA	genetic algorithm
HDE	hybrid differential evolution
PSO	particle swarm optimization

1. Phần mở đầu
Tối ưu hóa được xem như một lĩnh vực kinh điển của toán học và có rất nhiều ứng dụng trong các lĩnh vực khác nhau như thiết kế chế tạo máy, điều khiển tự động, quy hoạch tài nguyên…Có rất nhiều thuật toán đã và đang được sử dụng để tìm kiếm lời giải tối ưu cho bài toán tối ưu hóa. Trong số đó, các thuật toán tiến hóa được xem như lựa chọn đầy hứa hẹn.

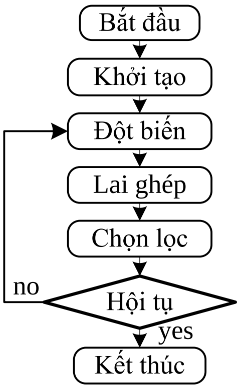
Thuật toán tiến hóa vi sai DE được xem như là một trong những thuật toán tối ưu hóa ngẫu nhiên mạnh mẽ nhất được sử dụng hiện nay. Thuật toán tiến hóa vi sai được giới thiệu lần đầu dưới dạng một báo cáo kỹ thuật bởi hai nhà khoa học R. Storn và K.V. Price vào năm 1995, tài liệu [1]. Thuật toán tiến hóa vi sai có khả năng xử lý rất hiệu quả các bài toán tìm cực trị hàm không khả vi, hàm phi tuyến và các hàm đa mục tiêu. Thuật toán tiến hóa vi sai với các ưu điểm như có khả năng tìm kiếm chính xác lời giải tối ưu toàn cục mà không phân biệt giá trị tham số ban đầu, tốc độ hội tự nhanh, ít thông số liên quan đã làm cho thuật toán DE trở nên một trong những công cụ mạng mẽ nhất trong lĩnh vực tối ưu hóa. Vesterstrom và các cộng sự [2] đã đánh giá chất lượng của các thuật toán DE, PSO và các thuật toán tối ưu khác kiểm chứng trên 34 hàm benchmark thông dụng. Karaboga và các cộng sự [3] đã ứng dụng thuật toán DE để thiết kế bộ lọc FIR và thực hiện so sánh chất lượng của thuật toán DE với thuật toán di truyền GA. Kết quả từ các nghiên cứu này đã chỉ ra rằng chất lượng và hiệu quả của thuật toán DE khi so sánh với các thuật toán GA, PSO. Tuy nhiên, nhược điểm chung của các thuật toán tiến hóa là hiện tượng mà ở đó không có sự cải thiện khả năng hội tụ trong quần thể qua một loạt các thế hệ. Hiện tượng này làm cho không thể tìm trong không gian tìm kiếm mới lời giải tối ưu toàn cục. Thời gian xảy ra hiện tượng này thường không thể xác định. Để thoát khỏi hiện tượng này, các giải thuật tìm kiếm cục bộ được áp dụng để tìm giải pháp khả thi hơn trong không gian cục bộ. Các bài báo [4-7] đã lai ghép thành công các thuật toán suy giảm độ dốc GD với các thuật toán tiến hóa GA, PSO để cải thiện chất lượng hội tụ của thuật toán.

Trong bài báo này, tác giả đề xuất thực hiện lai ghép thuật toán suy giảm độ dốc GD với thuật toán tiến hóa vi sai DE để cải thiện chất lượng hội tụ của thuật toán DE cơ bản. Thuật toán sau lai ghép HDE mang cả ưu điểm của thuật toán DE với khả năng tìm kiếm toàn cục và thuật toán GD với khả năng tìm kiếm chính xác cục bộ. Kết quả thực nghiệm kiểm chứng trên một số hàm benchmark được sử dụng để chứng tỏ chất lượng và hiệu quả của thuật toán đề xuất HDE khi so sánh với các thuật toán tối ưu khác.

2. Nội dung chính

2.1 Thuật toán tiến hóa vi sai cơ bản

Thuật toán tiến hóa vi sai DE được R.Storn và K.V. Price giới thiệu lần đầu năm 1995, tài liệu [1]. Cho đến nay, thuật toán tiến hóa vi sai đã trở nên phổ biến. Lưu đồ thực hiện thuật toán tiến hóa vi sai cơ bản được mô tả như H.1.



H.1 *Lưu đồ thực hiện thuật toán DE*

<u>Khởi tạo</u>: Giả sử chúng ta muốn tìm kiếm lời giải tối ưu cho một hàm thực với *D* tham số. Đầu tiên, chúng ta phải chọn kích thước quần thể *NP*. Thuật toán DE bắt đầu bằng cách tạo ra một cách ngẫu nhiên *NP* vector *D* chiều. Mỗi vector này được gọi là một cá thể và được biểu diễn như sau:

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}] \qquad (1)$$

Trong đó: G là số thế hệ, $G = 0,1,\ldots,G_{\max}$ và $i = 1,2,\ldots,NP$.

Mỗi vector tham số (cá thể) trong quần thể được giới hạn trong không gian tìm kiếm nhất định và nó không được vượt ra khỏi hai đầu giới hạn. Giới hạn dưới và giới hạn trên được ký hiệu tương ứng như sau:

$\vec{X}_{\min} = x_{1,\min},\ldots,x_{D,\min}$ và $\vec{X}_{\max} = x_{1,\max},\ldots,x_{D,\max}$.

Vì vậy, các thành phần thứ *j* của véc-tơ thứ *i* được khởi tạo như sau:

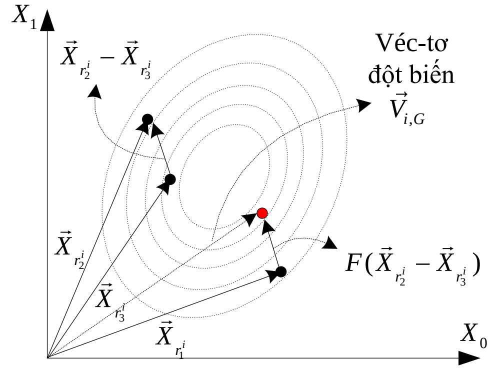
$$x_{j,i} = x_{j,\min} + rand_{j,i}[0,1]\; x_{j,\max} - x_{j,\min} \qquad (2)$$

Trong đó, $0 \le rand_{j,i}[0,1] \le 1$.

<u>Đột biến</u>: Mỗi cá thể phải trải qua quá trình đột biến, lai ghép và chọn lọc. Đột biến là sự thay đổi hoặc xáo trộn với một yếu tố ngẫu nhiên. Trong thuật toán DE, một véc-tơ mẹ ở thế hệ hiện tại được gọi là véc-tơ mục tiêu, một véc-tơ đột biến thu được từ quá trình đột biến. Để tạo ra véc-tơ đột biến cho mỗi véc-tơ mục tiêu thứ *i* từ thế hệ hiện tại, ba véc-tơ khác nhau $\vec{X}_{r_1^i}, \vec{X}_{r_2^i}, \vec{X}_{r_3^i}$ được lấy mẫu ngẫu nhiên từ quần thể ở thế hệ hiện tại. Sự lựa chọn ngẫu nhiên ba véc-tơ khác nhau $\vec{X}_{r_1^i}, \vec{X}_{r_2^i}, \vec{X}_{r_3^i}$ trong quần thể làm cho quá trình tìm kiếm lời giải tối ưu được trải rộng trong không gian tìm kiếm. Các chỉ số r_1^i, r_2^i và r_3^i là số nguyên loại trừ lẫn nhau được chọn lựa ngẫu nhiên từ khoảng [1, NP] và khác với véc-tơ cơ sở thứ *i*. Các chỉ số này được tạo ra ngẫu nhiên một lần cho mỗi véc-tơ đột biến. Hiệu của hai véc-tơ bất kỳ trong ba véc-tơ được thu nhỏ lại bởi số vô hướng *F* và sau đó được cộng với véc-tơ thứ ba để được véc-tơ đột biến $\vec{V}_{i,G}$. Quá trình này có thể biểu diễn như sau:

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F(\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) \qquad (3)$$

Trong đó, số vô hướng F $\in$ [0, 1]. Quá trình tạo véc-tơ đột biến được minh họa H.2.



H.2 *Quá trình đột biến trong không gian 2-D*

<u>Lai ghép</u>: Sau khi tạo ra véc-tơ đột biến, lai ghép được thực hiện để tăng cường sự đa dạng của quần thể. Véc-tơ đột biến $\vec{V}_{i,G}$ được lai ghép với véc-tơ mục tiêu $\vec{X}_{i,G}$ hình thành véc-tơ thử nghiệm $\vec{U}_{i,G} = [u_{1,i,G},\ldots,u_{D,i,G}]$. Thuật toán DE thường sử dụng phương pháp lai ghép nhị thức. Quá trình lai ghép nhị thức được trình bày như sau:

$$u_{j,i,G} = \begin{cases} V_{j,i,G} & if(rand_{j,i}[0,1] \le C) \\ X_{j,i,G} & otherwise \end{cases} \qquad (4)$$

Trong đó, $i = 1,2,\ldots,NP$ và $j = 1,2,\ldots,D$; C là xác suất lai ghép; $rand_{j,i}[0,1]$ là số ngẫu nhiên phân bố đều.

<u>Chọn lọc</u>: Véc-tơ mục tiêu $\vec{X}_{i,G}$ được so sánh với véc-tơ thử nghiệm $\vec{U}_{i,G}$. Quá trình chọn lọc được mô tả như sau:

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & if \;\; f(\vec{U}_{i,G}) \le f(\vec{X}_{i,G}) \\ \vec{X}_{i,G} & otherwise \end{cases} \qquad (5)$$

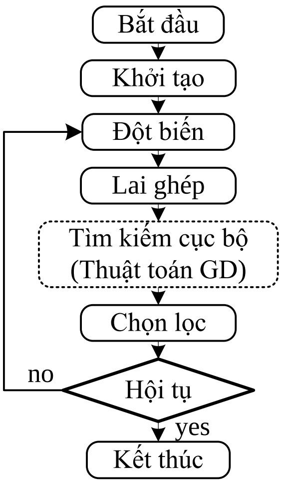
Trong đó, $f(\vec{X})$ là hàm chi phí cực tiểu.

<u>Hội tụ</u>: Quá trình trên được thực hiện trong một vòng lặp. Quá trình lặp chỉ kết thúc khi một trong các điều kiện sau đây được đáp ứng:

- Khi số thế hệ tiến tới cực đại Gmax.
- Khi giá trị hàm chi phí không thay đổi đáng kể trong quá trình lặp đi lặp lại
- Khi hàm chi phí cực tiểu đạt giá trị mong muốn.

2.2 Thuật toán lai HDE

Thuật toán lai HDE được tạo thành bằng cách lai ghép thuật toán tiến hóa vi sai cơ bản DE và thuật toán suy giảm độ dốc GD. Có nhiều cách khác nhau để áp dụng lai thuật toán GD với thuật toán DE. Ở đây, thuật toán GD được áp dụng ở giai đoạn “lai ghép” để tạo ra véc-tơ thử nghiệm mới trước khi bước vào giai đoạn chọn lọc. Thuật toán HDE sau lai ghép giúp cải thiện chất lượng hội tụ của thuật toán tiến hóa vi sai cơ bản một cách đáng kể. Chi tiết lưu đồ thực hiện thuật toán HDE được mô tả như theo H.3.



H.3 *Lưu đồ thực hiện thuật toán HDE*

Các bước khởi tạo, đột biến, lai ghép, chọn lọc và hội tụ được thực hiện tương tự như ở thuật toán tiến hóa vi sai cơ bản. Bước tìm kiếm cục bộ dựa trên thuật toán suy giảm độ dốc được mô tả như sau:

$$\vec{U}^{*}_{i,G} = \vec{U}_{i,G} - \eta \nabla(f(\vec{U}_{i,G})) \qquad (6)$$

Trong đó, η là bước nhảy của thuật toán suy giảm độ dốc, $\nabla(f(\vec{U}_{i,G}))$ là gradient của hàm chi phí $f(\vec{U}_{i,G})$.

2.3 Tối ưu hóa hàm toán học dùng thuật toán HDE

Bài toán tối ưu hóa hàm nói chung là xác định giá trị cực tiểu và cực đại của các hàm số trong điều kiện giới hạn nhất định. Bài báo này đề cập đến sự khảo sát tối ưu các hàm benchmark được mô tả ở bảng 1 dùng thuật toán HDE.

Với các công thức toán học và không gian tìm kiếm được mô tả ở bảng 1, quy trình tìm kiếm tối ưu dùng thuật toán HDE được mô tả chi tiết như sau:

+ Bước 1: Chọn các thông số như kích thước quần thể *NP*, hệ số đột biến *F*, xác suất lai ghép *C*, số thế hệ cực đại *Gmax*, số lượng các biến *n*.
+ Bước 2: Tạo ra một quần thể từ *NP* véc-tơ *n* chiều được chọn lựa ngẫu nhiên, mô tả ở (1).
+ Bước 3: Đánh giá các cá thể trong quần thể.
+ Bước 4: Với mỗi cá thể thứ i trong quần thể lựa chọn ngẫu nhiên các biến $r_1, r_2, r_3 \in [1,2,\ldots,NP]$, và $r_1 \neq r_2 \neq r_3 \neq i$.
+ Bước 5: Áp dụng toán tử đột biến cho mỗi một cá thể trong quần thể để được một véc-tơ đột biến, mô tả ở (3).
+ Bước 6: Lai ghép mỗi một véc-tơ mục tiêu trong quần thể hiện tại với một véc-tơ đột biến để tạo ra một véc-tơ thử nghiệm, mô tả ở (4).
+ Bước 7: Tìm kiếm cục bộ dựa trên giải thuật suy giảm độ dốc, mô tả ở (6).
+ Bước 8: Chọn lựa giữa véc-tơ mục tiêu và véc-tơ thử nghiệm sao cho thỏa điều kiện, mô tả ở (5).
+ Bước 9: Lặp lại chu kỳ huấn luyện từ bước 4 đến bước 8 cho đến khi tiêu chí hội tụ được đáp ứng.

Bảng 1. Các hàm benchmark được sử dụng trong bài báo

Hàm	Giới hạn	Công thức toán học
Sphere	$-100,100^{n}$	$f_1(x) = \sum_{i=1}^{n} x_i^2$
Griewank	$-600,600^{n}$	$f_2(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$
Ackley	$-30,30^{n}$	$f_3(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(-\frac{1}{n}\sqrt{\sum_{i=1}^{n}\cos(2\pi x_i)}\right)$
Rastrigin	$-5.12,5.12^{n}$	$f_4(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i))$

2.4 Kết quả thực nghiệm

Trong bài báo này, tất cả các kết quả kiểm chứng được chúng tôi thực hiện trên phần mềm Matlab 2013b được cài đặt trong máy tính có cấu hình Intel Core i3, 2.00GB RAM và tốc độ xử lý 2.53GHz.

Chất lượng của thuật toán HDE được so sánh với các thuật toán PSO và DE bằng cách so sánh quá trình hội tụ và giá trị tối ưu nhất được tìm thấy cho các hàm benchmark được mô tả ở bảng 1. Bảng 2 mô tả các tham số điều khiển của các thuật toán DE, PSO và HDE. Vì các thuật toán PSO, DE và HDE là các thuật toán tối ưu ngẫu nhiên. Do đó, cần lặp lại quy trình nhiều lần để thống kê các giá trị tìm kiếm ở các trị tối ưu nhất và trị trung bình.

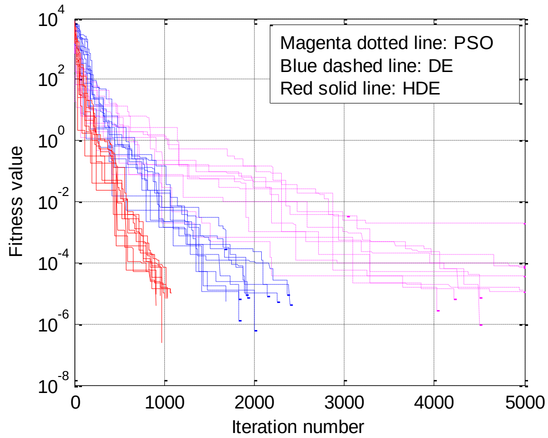
Bảng 2. Các tham số điều khiển của PSO, DE, HDE

Phương pháp	Tham số	Giá trị
DE	Kích thước quần thể, NP	18
	Hệ số đột biến, F	0.5026
	Xác suất lai ghép, CR	0.6714
	Số lượng các biến, n	2
HDE	Tốc độ học hay bước nhảy, η	0.01
PSO	Kích thước quần thể, s	149
	Trọng số quán tính, w	-0.3236
	Hệ số gia tốc, c_1	-0.1136
	Hệ số gia tốc, c_2	3.9789

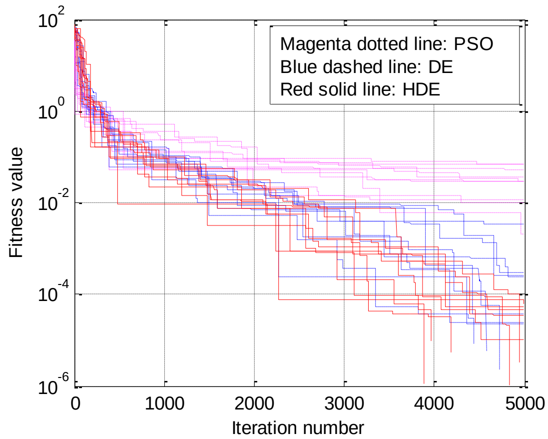
Kết quả thống kê về giá trị tối ưu nhất và trị trung bình sau 10 lần lặp quá trình tìm kiếm giá trị tối ưu của các hàm Sphere, Griewank, Ackley và Rastrigin dùng các thuật toán PSO, DE và HDE được mô tả ở bảng 3. Ngoài ra, H.4, H.5, H.6 và H.7 mô tả kết quả so sánh sự hội tụ của các thuật toán PSO, DE và HDE trong quá trình tìm kiếm giá trị tối ưu của các hàm Sphere, Griewank, Ackley và Rastrigin.

Bảng 3. Thống kê kết quả tìm kiếm dùng PSO, DE và HDE

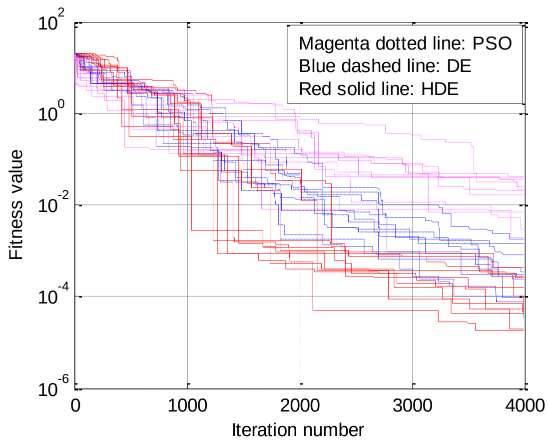
Hàm		PSO	DE	HDE
f_1	Trung bình	2.1387e-4	6.8596e-6	6.7537e-6
	Tốt nhất	9.7768e-7	6.3167e-7	2.3047e-7
f_2	Trung bình	0.0337	4.3703e-4	2.2925e-5
	Tốt nhất	0.0021	2.3483e-6	1.0576e-6
f_3	Trung bình	0.0211	5.4710e-4	1.3484e-4
	Tốt nhất	0.0019	3.3692e-5	1.7646e-5
f_4	Trung bình	0.3010	0.1051	8.9699e-5
	Tốt nhất	1.5937e-5	3.2059e-5	1.5203e-5



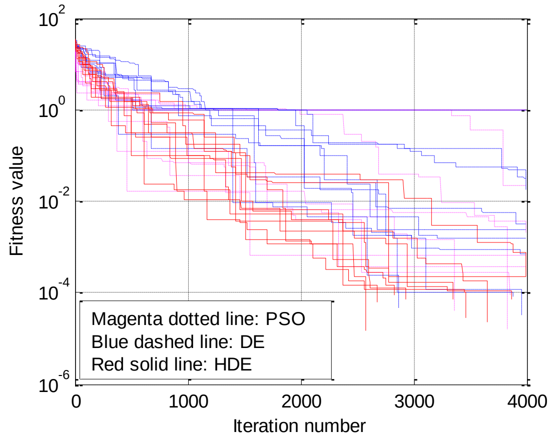
H.4 *So sánh hội tụ của các thuật toán PSO, DE và HDE với hàm Sphere*



H.5 *So sánh hội tụ của các thuật toán PSO, DE và HDE với hàm Griewank*



H.6 *So sánh hội tụ của các thuật toán PSO, DE và HDE với hàm Ackley*



H.7 *So sánh hội tụ của các thuật toán PSO, DE và HDE với hàm Rastrigin*

Dựa trên các kết quả mô tả ở bảng 3, H.4, H.5, H.6 và H.7, chúng ta thấy rằng chất lượng và hiệu quả vượt trội của thuật toán lai HDE so với các thuật toán PSO và DE. Chẳng hạn, khi kiểm chứng thuật toán HDE với hàm Sphere thì sau khoảng 950 thế hệ thì đã tìm kiếm được điểm tối ưu có giá trị trung bình là 6.7537e-6. Tương tự, kiểm chứng với thuật toán DE thì sau khoảng 2000 thế hệ ta được trị trung bình 6.8596e-6 và kiểm chứng với thuật toán PSO ta được trị trung bình là 2.1387e-4.

3. Kết luận

Trong bài báo này, tác giả đã xây dựng quy trình tìm kiếm tối ưu dùng thuật toán lai HDE được tạo thành bằng cách lai ghép thuật toán tiến hóa vi sai và thuật toán suy giảm độ dốc. Chất lượng và hiệu quả của thuật toán HDE được kiểm chứng thực nghiệm trên các hàm toán học như Sphere, Griewank, Ackley, Rastrigin. Kết quả thực nghiệm chứng tỏ hiệu quả vượt trội của thuật toán HDE so với thuật toán DE và thuật toán PSO. Tương lai, tác giả sẽ sử dụng thuật toán HDE để tối ưu các trọng số mạng nơ rôn ứng dụng trong nhận dạng điều khiển các hệ phi tuyến.

Acknowledgements

This paper was supported by the NAFOSTED and the DCSELAB, VNU-HCM, Viet Nam.

Tài liệu tham khảo

[1] Storn R. and Price K.V., *Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces*, International Conference on Swarm Intelligence, USA, Technical Report, TR-95-012, 1995.

[2] Vesterstrom, Jakob, and Rene Thomsen, *A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems* ,IEEE Congress on volutionary Computation, Vol. 2, pp.1980-1987, 2004.

[3] Karaboga, Nurhan, and Bahadir Cetinkaya, *Performance comparison of genetic and*

differential evolution algorithms for digital FIR filter design, Advances in information systems. Springer, pp.482-488, 2005.

[4] Li, Zhibao, Ka Fai Cedric Yiu, and Zhiguo Feng, *A hybrid descent method with genetic algorithm for microphone array placement design,* Applied Soft Computing, vol.13, pp. 1486-1490, 2013.

[5] Kuo, R. J., and Ferani E. Zulvia, *The gradient evolution algorithm: A new metaheuristic*, Information Sciences, pp.246-265, 2015.

[6] Han, Fei, and Qing Liu, *A diversity-guided hybrid particle swarm optimization based on gradient search*, Neurocomputing, vol.136, pp. 234-240, 2014.

[7] Noel, Mathew M, *A new gradient based particle swarm optimization algorithm for accurate computation of global minimum*, Applied Soft Computing, vol.12.1, pp.353-359, 2012.

Ho Pham Huy Anh received the B.S. and the M.Sc. degrees in the Department of Electrical and Electronics Engineering from HCM City University of Technology in 1987 and in 1993, respectively. He received the Ph.D. degree from University of Ulsan, Korea in 2008. He is currently an Associate Professor, Senior Lecturer in the Faculty of Electrical and Electronics, HCM City University of Technology, HCM City, Viet Nam. Up to now, he co-authored 4 books and published over 50 papers on national and international journals and conference proceedings. His current research interests include intelligent control, robotics, novel energy applications, modeling and identification.

Nguyen Ngoc Son received the B.S. and the M.Sc. degrees in the Faculty of Electrical and Electronics Engineering from HCM City University of Technology in 2010 and in 2012, respectively. He is now PhD candidate with the Faculty of Electrical and Electronic Engineering, HCM City University of Technology, Vietnam. He is currently a Lecturer in the Faculty of Electronics Technology, Industrial University of HoChiMinh City, Viet Nam. His current research interests include robotics, intelligent control, modeling and identification.