

Application of Gravity Vectors and Moving Vectors for the Acceleration of both Differential Evolution and Interactive Differential Evolution

Funaki, Ryohei
Graduate School of Design, Kyushu University

Takagi, Hideyuki
Faculty of Design, Kyushu University

<https://hdl.handle.net/2324/1808909>

出版情報 : International Conference on Genetic and Evolutionary Computing. 2011, pp.287-290, 2011-08-29. IEEE

バージョン :

権利関係 :



Application of Gravity Vectors and Moving Vectors for the Acceleration of both Differential Evolution and Interactive Differential Evolution

Ryohei FUNAKI

Graduate School of Design, Kyushu University
Fukuoka, Japan

Email: rfunaki620802@kyudai.jp

Hideyuki TAKAGI

Faculty of Design, Kyushu University
Fukuoka, Japan

<http://www.design.kyushu-u.ac.jp/~takagi/>

Abstract—We propose and evaluate two methods for accelerating differential evolution and interactive differential evolution (IDE). The first acceleration method, which we call DE/gravity, aims to realize performance similar to that of paired-comparison-based IDE/best while removing the requirement that the IDE user must choose the best individual among all displayed individuals. The second acceleration method generates not only a conventional trial vector but also a second and third trial vector. It calculates a moving average vector, X_{moving} , for the population between generations, and compares a given target vector with the three trial vectors of a conventional trial vector, a target vector + X_{moving} , and a trial vector + X_{moving} , and uses the best one among the four vectors as offspring in the next generation. We evaluate these acceleration methods and a conventional method by applying them to Gaussian mixture models and demonstrate the effectiveness of our proposed methods.

Keywords—interactive evolutionary computation; differential evolution; acceleration; gravity vector; moving vector;

I. INTRODUCTION

Evolutionary computation (EC) is a multiple searching point-based optimization method wherein operations modeling biological evolution are iterated to find the global optimum of a fitness function. It encompasses genetic algorithms (GA), genetic programming, evolutionary strategies, evolutionary programming, particle swarm optimization, differential evolution (DE), and others.

Interactive EC (IEC) is an EC method that uses the subjective evaluations of an IEC user as fitness values [1]. Because the IEC user must repeatedly evaluate individuals generated by EC, user fatigue remains a problem with IEC. This is why IEC population sizes are usually small, such as 10 - 20 individuals, and the maximum number of search generations is also small, with 10 - 20 generations, whereas normal EC uses large population sizes and many searching generations. To reduce the human fatigue problem, several approaches have been proposed, including improving IEC user interfaces, developing fast EC algorithms, and acceleration methods for existing algorithms.

One of the approaches is tournament interactive GA (IGA) [2], [3]. In this case, fitness values are not given by its user directly but rather determined based on the number of wins during tournament comparisons. The advantage of this

method is that user fatigue is reduced thanks to the use of paired comparisons, but unfortunately its convergence may also be slower than normal IGA as the tournament does not compare all individuals and produces less comparison information.

Several IECs adopting multiple EC algorithms have been proposed, and Interactive Differential Evolution (IDE) [4], [5], [6] is one of them. A paired comparison-based IDE [7] that passively uses a paired comparison between a target vector and a trial vector can improve IEC interface without modifying the original DE algorithm, unlike tournament IGA. Its faster convergence than conventional IGA was demonstrated through simulation evaluation [7].

The objective of this paper is to propose and evaluate acceleration methods for IDE that reduce human fatigue. We compare the convergence of the IDE using two acceleration methods with that of a conventional IDE using the RAND method (IDE/rand) that uses a randomly selected individual as the base vector and with the BEST method (IDE/best), where the best individual is used as the base vector.

II. INTERACTIVE DIFFERENTIAL EVOLUTION

User fatigue with paired comparison-based IDE is less than that of other IECs because of its nature of choosing either one of two individuals [7]. This is why we think that DE is one of the best EC algorithms for use with IEC.

Paired comparison-based IDE is performed as follows. (1) A mutant vector, X_{mutant} , is created from randomly selected individuals X_{r1} (base vector), X_{r2} , and X_{r3} (DE/rand). In DE/best, the best individual, X_{best} , is used as the base vector:

$$\begin{aligned} \text{DE/rand: } X_{mutant} &= X_{r1} + F \times (X_{r2} - X_{r3}) \\ \text{DE/best: } X_{mutant} &= X_{best} + F \times (X_{r2} - X_{r3}), \end{aligned}$$

(2) a trial vector, X_{trial} , is generated by crossing X_{mutant} and a target vector, X_{target} , (3) X_{trial} and X_{target} are compared, with the better one chosen, and (4) X_{target} is replaced with the better individual chosen in (3). These steps are repeated for all individuals in each generation. The key to the paired comparison-based IDE is in step (3).

Unlike normal optimization with a fitness function, it is not simple to use DE/best for IDE. In a paired comparison-

based IDE, the user does not give fitness values to individuals, but rather relatively compares paired individuals and chooses the better ones. Furthermore, the dominant relationship among all individuals is unknown because the IDE user compares only a target vector and a trial vector. This means that it is necessary for the paired comparison-based IDE/best method to display all individuals to an IDE user and let the user choose the best individual before performing the DE operations. However, it is difficult to require an IDE user to choose the best individual, especially for time sequential tasks such as creating music, sounds, or movies, as well as conventional IGA.

III. PROPOSAL OF TWO ACCELERATION METHODS FOR BOTH DE AND IDE

A. Proposal I: Using a population gravity vector

It is said that DE/best is faster than DE/rand for many cases [8], and our preliminary simulation results for IDE supports this tendency. We would like to use DE/best in IDE to reduce IDE user fatigue. However, as mentioned in the previous section, the paired comparison-based IDE/best requires an IDE user to select the best individual in population, which increases user fatigue especially in the case of sounds, movies, and other time-sequential tasks.

We propose a method that has the same fast convergence performance as IDE/best without requiring the user to choose the best individual. We hypothesized that the reason why DE/best converges faster was that a trial vector crossed by a target vector and a mutant vector is generated near the best individual, i.e. the base vector. Based on this assumption, we propose a method we call DE/gravity where the gravity of individuals is used as a base vector instead of the best individual. This method is based on the assumption that the landscape shapes of the most IEC tasks are roughly a big valley even if they are multimodal, and converge towards the direction of the center of the individuals (see Figure 1).

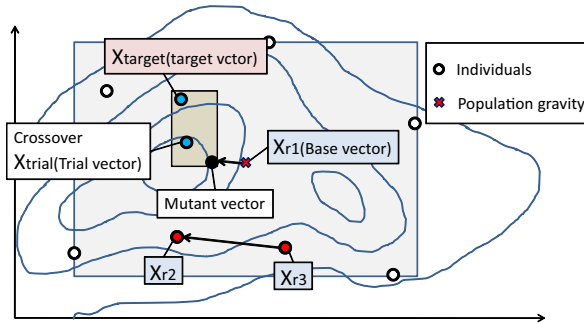


Figure 1. Searching to the direction of the gravity of individuals on a landscape of a big valley with multi-modal characteristics.

B. Proposal II: Using a population movement vector

In this method, a vector representing the average movement from the previous generation is calculated and added

to individuals in the current generation. Since each DE individual has an explicit corresponding trial vector or target vector in the previous generation, we can calculate a moving vector from the previous generation to the current generation for each individual.

In this paper, we compare four vectors of a target vector (X_{target}), a trial vector (X_{trial}), $X_{target} + \text{an average moving vector}$ (X_{moving}), and $X_{trial} + X_{moving}$ and choose the best one among the four vectors as offspring. The procedure is detailed below.

- 1) Calculating an average moving vector from the previous generation.

Accumulate a differential vector, $X_{trial} - X_{target}$, when X_{trial} is better than X_{target} ; the accumulated vector is a moving vector, X_{moving} , from the current generation to the next generation (Figure 2(a)).

- 2) Adding new candidates.

Display not only X_{target} and X_{trial} , as in conventional IDE, but also $X_{target} + X_{moving}$ and $X_{trial} + X_{moving}$ to the IDE user (Figure 2(b)). The user chooses one of them to be the offspring for the next generation.

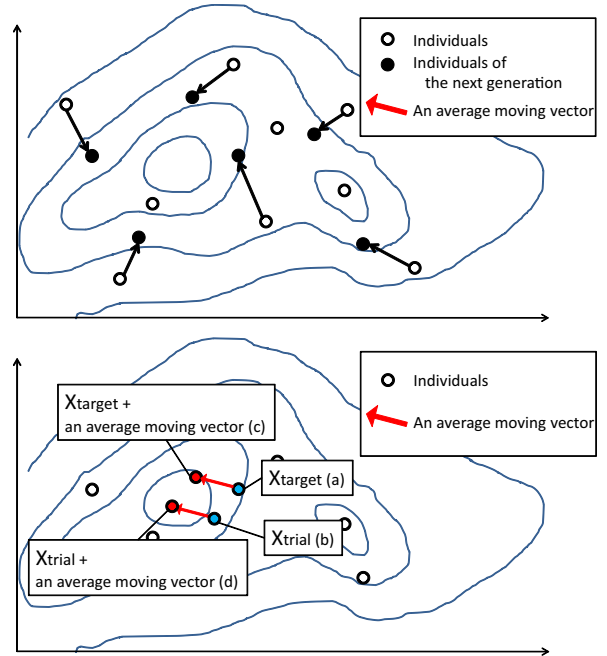


Figure 2. Proposed method II. Upper: calculating an average moving vector, X_{moving} , for the next generation. Lower: comparison of a target vector, X_{target} , a trial vector, X_{trial} , and two more offspring candidate vectors $X_{target} + X_{moving}$ and $X_{trial} + X_{moving}$.

Since the average moving vector between neighbor generations is expected to express the convergence direction of population, it is also expected that shifting whole individuals in this direction will accelerate DE convergence.

To analyze the effect of the two newly added can-

didates, we compare four IDE/(rand+moving) and four IDE/(best+moving) with (1) X_{target} and $X_{trial} + X_{moving}$, (2) X_{target} , X_{trial} and $X_{target} + X_{moving}$, (3) X_{target} , $X_{target} + X_{moving}$, and $X_{trial} + X_{moving}$, and (4) all four candidates.

IV. EXPERIMENTS

A. Experimental Conditions

The experimental conditions were as follows: the population size was 16; the optimization was performed in 10-dimensions; the experiment was run for 100 generations; uniform crossover was used with 0.8 rates; the scaling factor for the differential vector was 0.9; and 100 trial runs were performed. We compared six IDEs with t -test ($p < 0.01$): (1) IDE/gravity, (2) IDE/rand, (3) IDE/best, (4) IDE/(gravity+moving), (5) IDE/(rand+moving), and (6) IDE/(best+moving), where “/gravity” and “+moving” refer to our proposed methods I and II, respectively.

B. Pseudo IDE user

As a human being cannot repeat an IEC experiment many times with exactly the same conditions, we used an IDE simulator with a human-like fitness function. We assumed that human evaluation characteristics have a big valley shape from the macro point of view even if they are multi-modal, and that they should be rather simple, unlike ill-posed, tricky, and catastrophic benchmark functions; otherwise IEC users cannot find satisfactory solutions with a lower population size and within a fewer number of generations. Under this assumption, we use a Gaussian Mixture Model in Figure 3 described by the Eq. (1) as a human-like fitness function [7], [9].

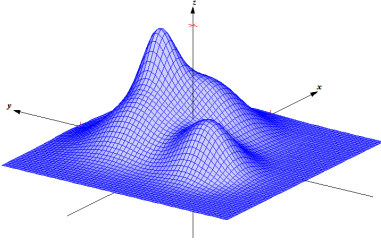


Figure 3. Example Gaussian Mixture Model consisting of four functions in a 2D space.

$$f(x_1, \dots, x_n) = \sum_{i=1}^k a_i \exp \left(- \sum_{j=1}^n \frac{(x_{ij} - \mu_{ij})^2}{2\sigma_{ij}^2} \right) \quad (1)$$

where k is the number of Gaussian functions ($k = 4$ in this paper), n is the number of dimensions ($n = 10$ in this paper), and σ_{ij} , a_{ij} , μ_{ij} are constants shown in the below:

$$\sigma = \begin{pmatrix} 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix}$$

$$a = \begin{pmatrix} 3.1 \\ 3.4 \\ 4.1 \\ 3 \end{pmatrix}$$

$$\mu = \begin{pmatrix} -1 & 1.5 & -2 & -2.5 & -1 & 1.5 & -2 & -2.5 & -1 & 1.5 \\ 0 & -2 & 3 & 1 & 0 & -2 & 3 & 1 & 0 & -2 \\ -2.5 & -2 & 1.5 & 3.5 & -2.5 & -2 & 1.5 & 3.5 & -2.5 & -2 \\ -2 & 1 & -1 & 3 & -2 & 1 & -1 & 3 & -2 & 1 \end{pmatrix}$$

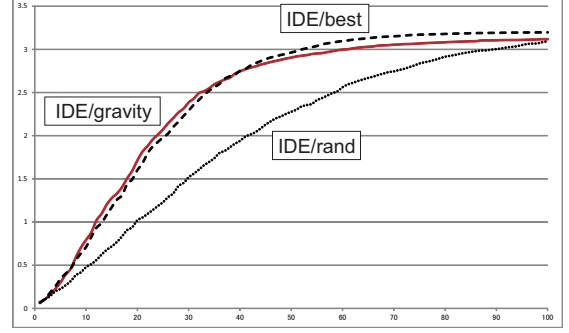


Figure 4. IDE/gravity vs. conventional IDE; average convergence curves of 6 IDE simulations with a human-like fitness function made by a Gaussian Mixture Model after 100 trial runs.

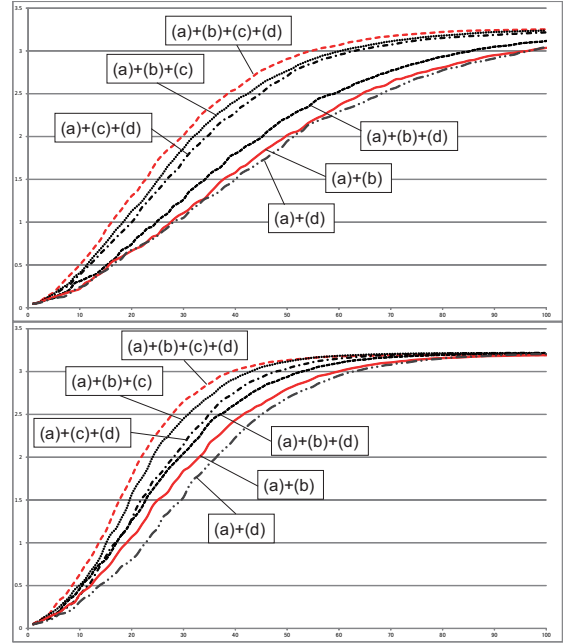


Figure 5. Analysis of performance factors in IDE/(rand+moving) (upper) and IDE/(best+moving) (lower). (a), (b), (c), and (d) are the vectors of X_{target} , X_{trial} , $X_{target} + X_{moving}$, and $X_{trial} + X_{moving}$, respectively.

V. EXPERIMENTAL RESULTS

Figures 4 - 5 show the average convergences of the six IDE simulations.

The performance of IDE/gravity was the almost same with that of IDE/best, and a statistical significant difference was not found between them (Figure 4). The proposed method II exhibited an improvement on the performances of all IDE that did not use the method. The combination of the two proposed methods, IDE/(gravity+moving), converged faster than IDE/(best+moving) and the difference in 13-16 generations was significant.

The convergence speed of IDE/(rand+moving) was in the order of: No.1: (a)+(b)+(c)+(d), No.2: (a)+(b)+(c), (a)+(c)+(d), No.3: (a)+(b)+(d), No.4: (a)+(b), (a)+(d), and that of IDE/(best+moving) was: No.1: (a)+(b)+(c)+(d), No.2: (a)+(b)+(c), No.3: (a)+(c)+(d), No.4: (a)+(b)+(d), No.5: (a)+(b) and No.6: (a)+(d) (Figure 5), where (a), (b), (c), and (d) are X_{target} , X_{trial} , $X_{target} + X_{moving}$, and $X_{trial} + X_{moving}$, respectively, and (a)+(b) is conventional DE.

Unlike IDE/best, significant differences between (a)+(b)+(c) and (a)+(c)+(d) and between (a)+(b) and (a)+(d) were not found ($p < 0.01$). The method in (c) could accelerate both IDE/rand and IDE/best significantly.

VI. DISCUSSION

The experimental results show that the proposed method could accelerate IDE to the performance level of IDE/best. Thanks to this performance boost, we can realize the performance of IDE/best without forcing an IDE user to choose the best individual as they would be forced to do with IDE/best. This characteristic is especially preferable when IDE is used for time-sequential tasks, such as those involving sounds or movies.

In IDEs using the proposed method II, (c) $X_{target} + X_{moving}$ accelerated IDE convergence, especially in the case of IDE/rand. This implies that the (c) is a major factor for accelerating DE. The frequency with which a target vector is replaced with a trial vector is not so high in normal DE. Thus, shifting a target vector toward the global optimum helps to evolve the target vector. This is our guess for why (c) was so effective.

In two cases where (d) was used instead of (b) there was reduced convergence speed for IDE/best; (a)+(b)+(c) $>$ (a)+(c)+(d), and (a)+(b) $>$ (a)+(d). Although we have not been able to make it clear, the difference in the selection of the base vector may be the reason; this significant difference was not found in IDE/rand, and the difference between DE/rand and DE/best is the selection method for the base vector. As DE/best uses the best individual as the base vector, the possibility that the location of a trial vector influenced by the base vector is closer to the global optimum is high. Adding X_{moving} to a trial vector that is already near the global optimum may have resulted in the trial overshooting the global optimum, and its beneficial effects were thus reduced.

Experimental results clearly showed the effectiveness of using the average moving vector in the proposed method II. However, this method requires the IDE user to compare three or four individuals N times (where N is population size) per generation, while paired comparison-based IDE requests an IDE user to compare only two individuals N times. As increasing the number of objects to be compared increases IDE user fatigue, the proposed method II should be adopted taking into account both the fatigue due to slow convergence without the method and fatigue due to the multiple comparisons used with the method.

VII. CONCLUSION

We proposed two methods for accelerating IDE, using a gravity vector and an average moving vector, and showed that they accelerate IDE through IDE simulations. They are applicable not only to IDE but also to DE in general.

REFERENCES

- [1] H. Takagi, "Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation", *Proceedings of the IEEE*, Vol.89, No.9, pp.1275–1296 (2001).
- [2] P. J. Angeline and J. B. Pollack, "Competitive Environments Evolve Better Solutions for Complex Tasks," in 5th International Conference on Genetic Algorithms (ICGA1993), Urbana/Champaign IL, USA, pp. 264–270 (July, 1993).
- [3] B. Johanson, "Automated Fitness Raters for the GP-Music System," University of Birmingham, Masters Degree Final Project (1997).
- [4] T. Akbal, G. N. Demir, A. E. Kanlikilicer, M. C. Kus, and F. H. Ulu, "Interactive Nature-Inspired Heuristics for Automatic Facial Composite Generation," in Genetic and Evolutionary Computation Conference (GECCO2006), Undergraduate Student Workshop, Seattle, WA, USA (July, 2006).
- [5] A. E. Kanlikilicer, "Interactive Differential Evolution for Facial Composite Generation," in Genetic and Evolutionary Computation (GECCO2006), Seattle, USA (July, 2006).
- [6] B. Kurt, A. S. Etaner-Uyar, T. Akbal, N. Demir, A. E. Kanlikilicer, M. C. Kus, and F. H. Ulu, "Active Appearance Model-Based Facial Composite Generation with Interactive Nature-Inspired Heuristics" in *Multimedia Content Representation, Classification and Security, Lecture Notes in Computer Science*. vol. 4105, pp.183–190 (July, 2006).
- [7] H. Takagi and D. Pallez, "Paired Comparison-based Interactive Differential Evolution," *The first World Congress on Nature and Biologically Inspired Computing (NaBIC2009)*, Coimbatore, India, pp.375–480 (December, 2009).
- [8] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*, Springer-Verlag (Dec., 2005).
- [9] Y. Nakano and H. Takagi, "Influence of Quantization Noise in Fitness on the Performance of Interactive PSO," in *Congress on Evolutionary Computation*, Trondheim, Norway, pp.2416–2422 (May, 2009).