Differential Evolution as the Global Optimization Technique and its Application to Structural Optimization

Satoshi Kitayama[1], Masao Arakawa[2,] Koetsu Yamazaki[3,]

[1] *Kanazawa University, Kakuma-machi, Kanazawa, 920-1192, Japan*

+81-76-234-4758

+81-76-234-4668

kitagon@t.kanazawa-u.ac.jp

[2] *Kagawa University, Hayashi-cho, Takamatsu, Kagawa, 761-0396, Japan*

+81-87-864-2223

+81-87-864-2223

[3] *Kanazawa University, Kakuma-machi, Kanazawa, 920-1192, Japan*

+81-76-234-4666

+81-76-234-4668

Abstract

In this paper, the basic characteristics of the Differential Evolution (DE) are examined. Thus, one is the meta-heuristics, and the other is the global optimization technique. It is said that DE is the global optimization technique, and also belongs to the meta-huristics. Indeed, DE can find the global minimum through numerical experiments. However, there are no proofs and useful investigations with regard to such comments. In this paper, the DE is compared with the Generalized Random Tunneling Algorithm (GRTA) and the Particle Swarm Optimization (PSO) that are the global optimization techniques for continuous design variables. Through the examinations, some common characteristics as the global optimization technique are clarified in this paper. Through benchmark test problems including structural optimization problems, the search ability of DE as the global optimization technique is examined.

Key words: Differential Evolution, Global Optimization, Structural Optimization

1. INTRODUCTION

Global optimization techniques are roughly classified into two classes: Thus, deterministic and stochastic techniques [1]. The Tunneling Algorithm (TA) that is a gradient-based optimization technique is one of the deterministic global optimizations [2]. On the other hand, the GA which is population-based optimization technique belongs to the stochastic global optimization technique [3]. The gradient-based global optimization techniques such as the TA often include the mathematical programming for local search, so that the search point always improves the objective function satisfying the constraints. The stochastic global optimization techniques, such as Simulated Annealing (SA), often accept the point at which the objective function is not improved. As the result, a global minimum can be found through the search iteration while escaping from local minimum.

Nowadays, owing to the rapid progress of computers, population-based Evolutionary Algorithms (EA), such as the Genetic Algorithm (GA), the Particle Swarm Optimization (PSO) [4], the Differential Evolution (DE) [5], the Ant Colony Optimization (ACO) [6] and etc., have been widely developed in comparison with the gradient-based optimization techniques, and have been applied to the optimum design problems[7,8,9,10,11,12,13]. EAs are often called the meta-heuristics that are the framework of the optimization techniques based on the analogies in the life and experiences. For example, the idea of GA is based on the Darwin's theory of evolution, and the PSO developed by Kennedy et al. obtain the basic idea from the social behaviors such as birds, fishes, and human society. The ACO is also an optimization technique that can obtain the hints from the behavior of ants. Most of EAs include the random elements in the algorithm, and then they belong to the stochastic optimization techniques. EAs also include the some parameters in the algorithm that the user determines in advance. For example, the GA should determine the parameters in advance, which are the mutation ratio and the crossover ratio. In the ACO, there are three parameters (α, β, and ρ) to update the pheromone, and we have to determine these parameters in advance.

In the framework called meta-heuristics, it is difficult to explain the reason why these methods can find global minimum. In the gradient-based global optimization techniques, we can easily explain why these methods can find global minimum, since the local search methodology is often included into the algorithm.

On the other hand, for example, the GA cannot explicitly explain why the GA can find global minimum, in comparison with the gradient-based global optimization techniques. The idea of the survival of the fittest may be one of the reasons to find a global minimum, however, only such idea may not be enough to explain why the global minimum can be found. The PSO that is one of the meta-heuristics has the search direction vector implicitly, and the generation methodology of the search point at the next iteration is clarified. The neighborhood of the search point is also clarified. The PSO also has the similar structure in comparison with the gradient-based global optimization methods [14]. Thus, it is easy to explain why the PSO can find a global minimum in comparison with the GA. The PSO is also analyzed from the anther aspect. Therefore, the PSO can be interpreted as the dynamics. It is noteworthy that the convergence of the PSO is determined from the stability of dynamics [15,16,17]. From above discussions, it is important to consider the reason why global minimum can be found. We consider that there are two important factors for global optimization techniques as follows:

(G1) It is preferable to include the search direction vector into the algorithm. The search direction is clarified by the search direction vector. It is expected that the search direction vector will turn to the direction that the objective is improved for local search, while the search direction vector will also turn to the direction that the objective is not improve for global search. The latter implies escaping from a local minimum.

(G2) It is preferable to include the randomness into the algorithm. In general, a global minimum is unknown. In order to find a global minimum, the randomness is one of the powerful tools for global search. The randomness is included into the TA for determining the next start point, and is also included into the PSO, the GA, the SA, and the ACO. In the case of population-based optimization techniques, the randomness can be expected to provide the diversity among the particles. In addition, the randomness can often give the perturbation of a search point. The search point can sometimes escape from a local minimum by this perturbation.

In addition, the stability analysis plays an important role for convergence. For example, the short convergence of ACO is discussed in Ref.[18].

The DE is one of the population-based global optimization techniques for continuous design variables [19]. In the DE, operations which are called the

mutation and the crossover are introduced like the GA, and then the DE may belong to the framework of meta-heuristics. The original DE mainly focuses on the single objective optimization problems, and several DE versions have also been proposed by many researchers [20]. The DE has been applied to the multi-objective optimization problems [21] and the discrete design variable problems [22]. The DE is now taken notice as well as the PSO, and then the search ability of both the DE and the PSO is often compared through benchmark problems. However, the DE as the global optimization technique is not well examined. The stability and convergence of the DE have been well discussed in Ref.[23]. In Ref.[23], the stability and convergence are discussed under assumptions that particles have crowed into a small neighborhood around an optimum. Then the stability and convergence using the Lyapunov stability theorem are discussed. In this research, the DE is interpreted as the dynamics, as well as the PSO. In other words, the stability and the convergence are discussed with the eigenvalue problems. Ref.[23] plays an important role in the stability and convergence of DE. However, we would like to discuss on the convergence of DE from the mathematical optimization point of view. In the gradient-based optimization techniques, the search direction vector and the step-size play important roles for searching an optimum. If the DE has some similar structures to the gradient-based optimization techniques, it is also possible to discuss the convergence. In addition, the PSO and the DE are suitable for finding the continuous optimum, so that they will belong to the same category among global optimization techniques. If similar structures between them can be found, the characteristics of the DE will be more clarified. Thus, it is important to compare the DE with the gradient-based global optimization techniques and the PSO. The authors consider that the DE may have some common characteristics in comparison with the gradient-based global optimization techniques and the PSO.

In this paper, the DE is compared with the PSO and the Generalized Random Tunneling Algorithm (GRTA) developed by the authors that is one of the gradient-based global optimization techniques [24]. As we mentioned above, two important factors for global optimization techniques are considered. We focus on the GRTA, the PSO, and the DE, and clarify some common characteristics among three methods. Secondly, the DE is examined through eleven periodical function

problems and two structural optimization problems. Through these benchmark problems, the search ability and the characteristics of the DE are well examined.

2. DIFFERENTIAL EVOLUTION

2.1 Problem Definition

In this paper, the following single-objective optimization problem is considered.

$$f(\boldsymbol{x}) \rightarrow \min \tag{1}$$

$$x_{i,L} \le x_i \le x_{i,U} \qquad i = 1, 2, \cdots, n \tag{2}$$

$$g_j(\boldsymbol{x}) \le 0 \qquad j = 1, 2, \cdots, m \tag{3}$$

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$ represents the continuous design variables, and *n* is the number of design variables. $f(\boldsymbol{x})$ is the objective function to be minimized, and $g_j(\boldsymbol{x})$ denotes the behavior constraints, and *m* is the number of behavior constraints. $x_{i,L}$ and $x_{i,U}$ denote the lower and upper bound on *i*-th design variable, respectively.

2.2 Basic Algorithm of DE

The DE developed by Storn and Price in 1995 is one of the population-based global optimization techniques for continuous design variables, such as the PSO. The DE is also the robust global optimization technique and is widely used. Many versions of the DE have been proposed, and DE/rand/1/bin is selected as the basic model in this paper, where "DE" represents the Differential Evolution, "rand" indicates that particles selected to compute the mutation are chosen at random. "1" is the number of pairs of particles, and "bin" means a binomial recombination. In DE/rand/1/bin, three particles in the design variable space are selected at random, and the operations that are the mutation and the crossover are introduced for generating the new particle at the next iteration. The roles of these operations are described in section 2.3 and 2.4, respectively. First, the basic algorithm of DE/rand/1/bin is described as follow:

(STEP0) The number of particles *popsize*, the mutation ratio *F*, the crossover ratio *Cr*, and the maximum search iteration number, $k_{\max}$, are set. The iteration counter *k* is initialized as *k*=1.

(STEP1) All particles are generated at random in the design variable space.

(STEP2) The following procedure is applied to all particles.

(STEP2-1) Particle d, denoted by $\boldsymbol{x}_d^k$, selects three particles ($\boldsymbol{x}_{r1}^k$,$\boldsymbol{x}_{r2}^k$,$\boldsymbol{x}_{r3}^k$) at random., where $d \neq r1 \neq r2 \neq r3$.

(STEP2-2: Mutation) New particle denoted by $\boldsymbol{v}_d^k$ is generated by the mutation. The mutation in the DE is given by Eq.(4).

$$\boldsymbol{v}_d^k = \boldsymbol{x}_1^k + F(\boldsymbol{x}_2^k - \boldsymbol{x}_3^k) \tag{4}$$

(STEP2-3: Crossover) New particle denoted by $\boldsymbol{u}_d^k$ is generated by the crossover between $\boldsymbol{x}_d^k$ and $\boldsymbol{v}_d^k$.

(STEP2-4) Objective function is evaluated at $\boldsymbol{x}_d^k$ and $\boldsymbol{u}_d^k$, and then the particle d is updated according to the following criteria.

$$\left.\begin{array}{l} f(\boldsymbol{u}_d^k) \leq f(\boldsymbol{x}_d^k) \rightarrow \boldsymbol{x}_d^k = \boldsymbol{u}_d^k \\ f(\boldsymbol{u}_d^k) > f(\boldsymbol{x}_d^k) \rightarrow \boldsymbol{x}_d^k = \boldsymbol{x}_d^k \end{array}\right\} \tag{5}$$

(STEP3) The iteration counter is increased $k = k + 1$.

(STEP4) If k is less than $k_{\max}$, return to STEP2. Otherwise, the algorithm will terminate.

2.3 Mutation and Global Search

In general, the update scheme of gradient-based optimization techniques for unconstrained optimization problems is given by Eq.(6) [25].

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k \tag{6}$$

where $\boldsymbol{x}^k$ denotes the current design point. $\boldsymbol{d}^k$ represents the search direction vector at k-th iteration, which is given by $-\nabla f(\boldsymbol{x}^k) \big/ \left\| \nabla f(\boldsymbol{x}^k) \right\|$. α^k is the deterministic step-size. The objective function is always improved at the new point $\boldsymbol{x}^{k+1}$. Thus, $f(\boldsymbol{x}^{k+1}) < f(\boldsymbol{x}^k)$. The terminal criterion of the gradient-based optimization techniques can be determined the following equations:

$$\left.\begin{array}{c} f(\boldsymbol{x}^{k+1}) - f(\boldsymbol{x}^k) \leq \varepsilon \\ \left\| \boldsymbol{d}^k \right\| \leq \varepsilon \end{array}\right\} \tag{7}$$

Convergence can be explained with Eq.(7) in the gradient-based optimization techniques. In the DE, convergence can be explained with Eq.(5), in which $\boldsymbol{u}_d^k$ represents the new search point generated by the crossover between $\boldsymbol{x}_d^k$ and $\boldsymbol{v}_d^k$. Here let us consider a case as shown in Fig.1.

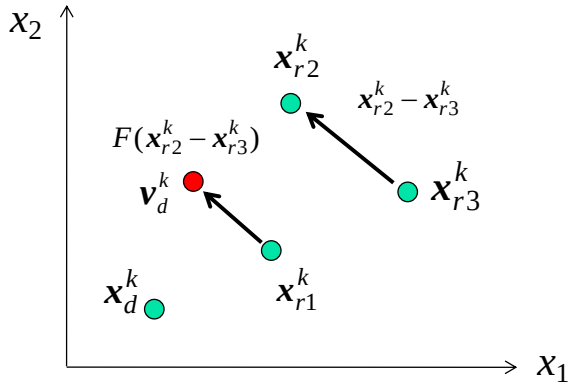


Fig.1 New point $\boldsymbol{v}_d^k$ in two dimensions

Compared with Eqs.(4) and (6), it is possible to consider that $\boldsymbol{x}_{r1}^k$ represents the basis vector, $\boldsymbol{x}_{r2}^k$-$\boldsymbol{x}_{r3}^k$ denotes the search direction vector, and F in Eq.(4) also denotes the fixed step-size. F in Eq.(4) is applied to all design variables, but this parameter can be also applied to each design variable as follows:

$$v_{d,i}^k = x_{1,i}^k + F_i(x_{2,i}^k - x_{3,i}^k) \tag{8}$$

$$F_i = F + d(rand(0,1) - 0.5) \tag{9}$$

$$d < 2F \tag{10}$$

where $x_{d,i}^k$ ($i = 1, 2, \cdots, n$) is the i-th design variable of particle d. $rand(0,1)$ represents the random number between 0 and 1. Through above discussions, it is possible to consider that the update scheme of DE is similar to the one of gradient-based optimization techniques. On the gradient-based optimization techniques, new search point always improves the objective function. On the other hand, the DE chooses three points ($\boldsymbol{x}_{r1}^k$, $\boldsymbol{x}_{r2}^k$, $\boldsymbol{x}_{r3}^k$), so that new point sometimes does not improve the objective function due to the randomness. Therefore, the search direction vector, $\boldsymbol{x}_{r2}^k$-$\boldsymbol{x}_{r3}^k$,turns to the direction that the objective function is not improved. This implies that it is possible to escape from local minimum, and then the global search ability will be enhanced. Let us explain the search direction vector in the DE with an illustrative example, as shown in Fig.2. In Fig.2, the arrow shows the search direction vector. Fig.2(a) shows a case where three particles locate in a same convex space. In addition, we assume the following relationship among three particles:

$$f(\boldsymbol{x}_{r1}^k) \le f(\boldsymbol{x}_{r2}^k) \le f(\boldsymbol{x}_{r3}^k) \tag{11}$$

In Fig.2 (a), $\boldsymbol{v}_d^k$ will be generated between $\boldsymbol{x}_{r2}^k$ and $\boldsymbol{x}_{r3}^k$ when the mutation ratio F is smaller than the $\left\|\boldsymbol{x}_{r2}^k - \boldsymbol{x}_{r3}^k\right\|$. The crossover is a stochastic operator in the DE, so let us assume that $\boldsymbol{v}_d^k$ is replaced as $\boldsymbol{u}_d^k$ by the crossover. In this case, the objective function at $\boldsymbol{u}_d^k$ will be improved, and the following equation can be established:

$$f(\boldsymbol{u}_d^k) \le f(\boldsymbol{x}_{r1}^k) \le f(\boldsymbol{x}_{r2}^k) \le f(\boldsymbol{x}_{r3}^k) \tag{12}$$

Here, we consider Eq.(5). If $f(\boldsymbol{u}_d^k)$ is less than $f(\boldsymbol{x}_d^k)$, $\boldsymbol{u}_d^k$ will become $\boldsymbol{x}_d^k$. Therefore, new search point $\boldsymbol{x}_d^k$ is generated into the same convex space. When Eq.(12) is established through the search iteration, the convergence will be expected.

Next, let us consider a case where three particles locate in a non-convex space as shown in Fig.2(b). If $\left\|\boldsymbol{x}_{r2}^k - \boldsymbol{x}_{r3}^k\right\|$ is nearly equal to zero, $\boldsymbol{v}_d^k$ will be generated around $\boldsymbol{x}_{r1}^k$. Otherwise, $\boldsymbol{v}_d^k$ will be generated between $\boldsymbol{x}_{r1}^k$ and $\boldsymbol{x}_{r2}^k$, or between $\boldsymbol{x}_{r1}^k$ and $\boldsymbol{x}_{r3}^k$. The crossover is a stochastic operator in the DE, so let us assume that $\boldsymbol{v}_d^k$ is replaced as $\boldsymbol{u}_d^k$ by the crossover. In Fig.2(b), the search direction turns to the direction that the objective function is not improved. Therefore, it may be possible to escape from a local minimum. In addition, the crossover also plays an important role for escaping from a local minimum. If the relation by Eq.(11) is not satisfied, concentration among the particles cannot be expected. In this case, it is expected that the diversity will be maintained.

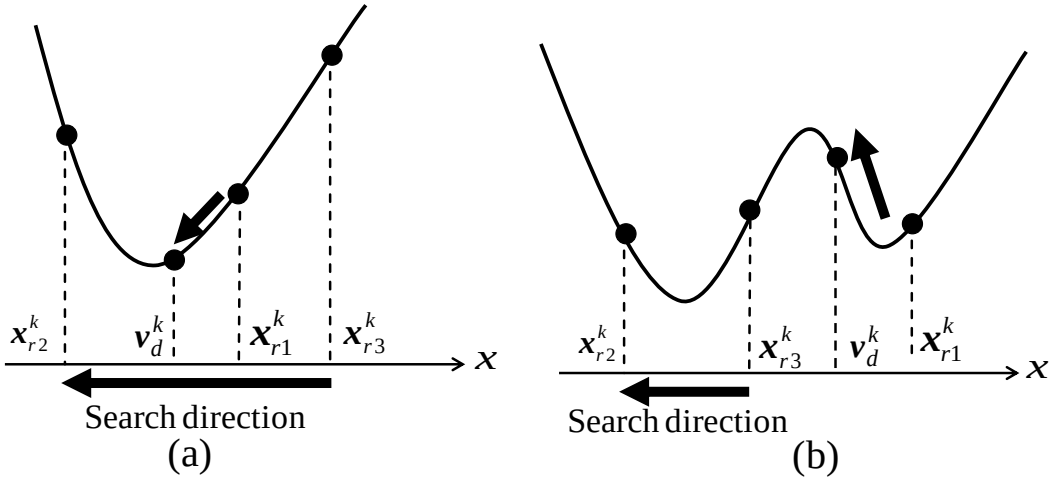


Fig.2 An illustrative example

2.4 Crossover and the Neighborhood

The crossover plays an important role in the DE. $x_{d,i}^{k}$ represents the *i*-th design variable of $\boldsymbol{x}_d^k$, and $v_{d,i}^{k}$ also denotes the *i*-th design variable of $\boldsymbol{v}_d^k$. In the crossover, the crossover point is determined at random. The element of crossover point is inherited from $\boldsymbol{v}_d^k$, and the element of $\boldsymbol{u}_d^k$ that is a new point is determined. Secondly, random number *r* between 0 and 1 is generated to each design variable, and *r* is compared with the crossover ratio, *Cr*. If *r* is less than *Cr*, the element of $\boldsymbol{u}_d^k$ inherits from $\boldsymbol{v}_d^k$. Otherwise, the element of $\boldsymbol{u}_d^k$ inherits from $\boldsymbol{x}_d^k$. Fig.3 is an illustrative example in eight design variables.

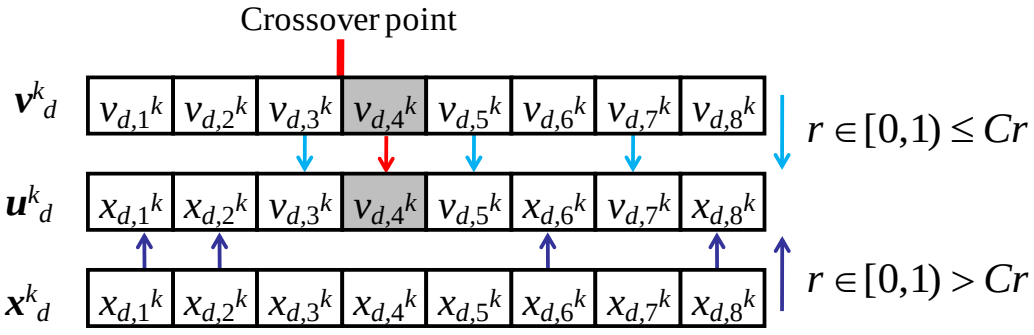


Fig.3 Crossover in DE

First, the crossover point is determined at random for determining the element of $\boldsymbol{u}_d^k$. In this example, the crossover point is 4-th design variable of $\boldsymbol{v}_d^k$. As the result, the element of $\boldsymbol{u}_d^k$ inherits from $\boldsymbol{v}_d^k$, as shown in Fig.3. Secondly, random number *r* is generated. If *r* is less than *Cr* that is predetermined, the element of $\boldsymbol{u}_d^k$ inherits from $\boldsymbol{v}_d^k$. Otherwise, the element of $\boldsymbol{u}_d^k$ inherits from $\boldsymbol{x}_d^k$. As the result, all elements of $\boldsymbol{u}_d^k$ are determined.

For simplicity, let us consider the role of crossover in two dimensions, as shown in Fig.4.

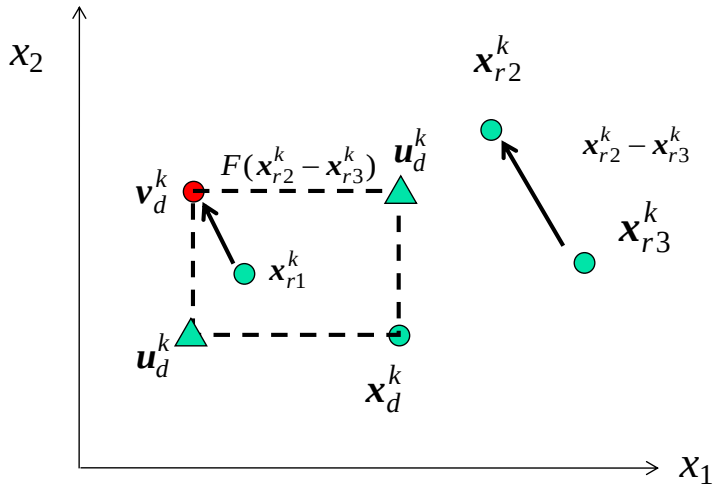


Fig.4 Generation of the neighborhood by crossover in the DE

New point $\boldsymbol{u}_d^k$ generated by the crossover may be denoted by triangle in Fig.3, or may be $\boldsymbol{v}_d^k$ itself, stochastically. Thus, the crossover in the DE generates the point of neighborhood around $\boldsymbol{x}_d^k$ directly. It is also possible to consider that the crossover gives the perturbation of $\boldsymbol{v}_d^k$, and this perturbation plays an important role in the global optimization.

2.5 Descent Property of Swarm

Descent property is an important character of the population-based optimization techniques. In the GA, many particles flock around the highest fitness through the generation, and then the global minimum can be found. Thus, one of the reasons to find the global minimum is the descent property of particles. In the DE, the descent property is given by Eq.(5).

2.6 Diversity and Concentration of Particles

In the population-based optimization techniques, it is very important to possess the diversity and concentration among particles. The diversity leads to the global search, while the concentration among the particles result in the local search for finding a optimum with high accuracy. In the DE, basis vector $\boldsymbol{x}_{r1}^k$ is selected at random, and this selection affects on the local/global search. If $\boldsymbol{x}_{r1}^k$ is close to $\boldsymbol{x}_d^k$, $\boldsymbol{u}_d^k$ produced by the mutation and the crossover may be close to $\boldsymbol{x}_d^k$, as shown in left hand side of Fig.5. $\boldsymbol{u}_d^k$ is the trial point, as shown in Eq.(5). Considering the descent property of particles, the local search will be enhanced in this case. Otherwise, $\boldsymbol{u}_d^k$ is generated faraway from $\boldsymbol{x}_d^k$, as shown in right hand side of Fig.5. In this case, global search will be performed, and this implies that the search will proceed maintaining the diversity.

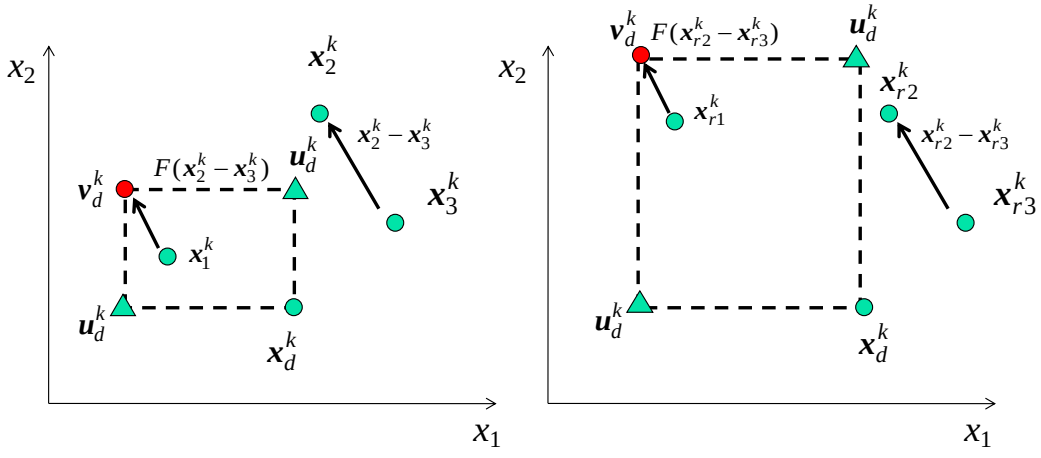


Fig.5 Local/Global search by the selection of basis vector $\boldsymbol{x}_1^k$

3. DIFFERENTIAL EVOLUTION AS THE GLOBAL OPTIMIZATION TECHNIQUE

In this section, the DE as the global optimization technique is considered. The DE is compared with two global optimization techniques. One is the Particle Swarm Optimization (PSO) which is one of the population-based optimization techniques, and another is the Generalized Random Tunneling Algorithm (GRTA) which is the gradient-based global optimization technique. In comparison with these two global optimization techniques, the DE is understood as the global optimization technique. Therefore, the DE possesses the desirable property of global optimization method. The aim of this section is to investigate the similarity between these global optimization methods, so that the algorithms of the PSO and the GRTA are not described in detail here. Please refer to Refs.[14] and [24].

3.1 DE and PSO

The PSO, which mimics the social behavior, is one of the global optimizer for continuous design variables. In the PSO, each particle has the velocity and the position. When the position and velocity of particle d at k-th iteration are represented by $\boldsymbol{x}_d^k$ and $\boldsymbol{v}_d^k$, respectively, the position and velocity of particle d at k+1 iteration are calculated by the following equations.

$$\boldsymbol{x}_d^{k+1} = \boldsymbol{x}_d^k + \boldsymbol{v}_d^{k+1}\Delta t \tag{13}$$

$$\boldsymbol{v}_d^{k+1} = w\boldsymbol{v}_d^k + \frac{c_1 r_1(\boldsymbol{p}_d^k - \boldsymbol{x}_d^k)}{\Delta t} + \frac{c_2 r_2(\boldsymbol{p}_g^k - \boldsymbol{x}_d^k)}{\Delta t} \tag{14}$$

where Δt represents the time-step and is set as unit. The coefficient w in (14) is called as the inertia term. The parameters r_1 and r_2 denote random numbers between [0,1]. The weighting coefficients c_1 and c_2 are recommended to keep the following relationship.

$$c_1 + c_2 \le 4 \tag{15}$$

In general, c_1=c_2=2 are often used. $\boldsymbol{p}_d^k$ in Eq.(14) is called the p-best, which represents the best position of particle d till k-th iteration. $\boldsymbol{p}_g^k$ is called as the g-

best, which denotes the best position in the swarm till *k*-th iteration. The g-best in the swarm is selected among the p-best. Thus, if the p-best is updated, it is examined whether the objective function at the g-best is improved or not. If the objective function is improved, then, the g-best is replaced. The descent property of the PSO has been well described in Ref.[14], and we consider the similarity between the PSO and the DE.

Combining Eqs.(13) and (14), the following equation is obtained.

$$\boldsymbol{x}_d^{k+1} = \boldsymbol{x}_d^k + w\boldsymbol{v}_d^k + \alpha(\boldsymbol{q} - \boldsymbol{x}_d^k) \tag{16}$$

where α and $\boldsymbol{q}$ are given as follows:

$$\alpha = c_1 r_1 + c_2 r_2 \tag{17}$$

$$\boldsymbol{q} = \frac{c_1 r_1 \boldsymbol{p}_d^k + c_2 r_2 \boldsymbol{p}_g^k}{c_1 r_1 + c_2 r_2} \tag{18}$$

From above equations, it is clear that the PSO has the search direction vector, $\boldsymbol{q}$-$\boldsymbol{x}_d^k$, and the stochastic step-size, α . In addition, it is possible to consider that $\boldsymbol{x}_d^k + w\boldsymbol{v}_d^k$ in Eq.(16) represents the basis vector. The similarities between the DE and the PSO can be summarized as follows:

(1) Both methods have the search direction vector, implicitly. In the DE, the implicit search direction vector is given by $\boldsymbol{x}_{r2}^k$-$\boldsymbol{x}_{r3}^k$. On the other hand, $\boldsymbol{q}$-$\boldsymbol{x}_d^k$, represents the implicit search direction vector in the PSO.

(2) Both methods have the basis vector. In the DE, $\boldsymbol{x}_{r1}^k$ is the basis vector, which is selected at random. On the other hand, in the PSO, $\boldsymbol{x}_d^k + w\boldsymbol{v}_d^k$ in Eq.(16) denotes the basis vector, where it is clear from Eq. (14) that $\boldsymbol{v}_d^k$ includes the randomness.

(3) Bothe methods include the descent property in the algorithm. This is one of the common and important characteristics of population-based optimization techniques.

3.2 DE and GRTA

The GRTA developed by the authors is one of the simple and effective global optimizers, which belongs to the gradient-based optimization techniques. The GRTA consists of three parts, which are called the minimization phase, the tunneling phase, and the constraint phase: The objective of minimization phase is to find the local minimum by the mathematical programming. Tunneling phase tries to find the new search point, at which the objective function is improved.

Finally, the feasibility of new search point is examined in the constraint phase. If the new search point is feasible, this new search point is replaced the new start point. Through these parts, the global minimum can be found for the constrained optimization problem. The terminal criterion is determined by the minimum step-size $T_{\min}$. The users can determine only three parameters: (1) Initial step-size,T. (2) Minimum step-size $T_{\min}$ to terminate the algorithm, (3) Maximum search iteration $it_{\max}$. The algorithm of GRTA is shown in Fig. 6.

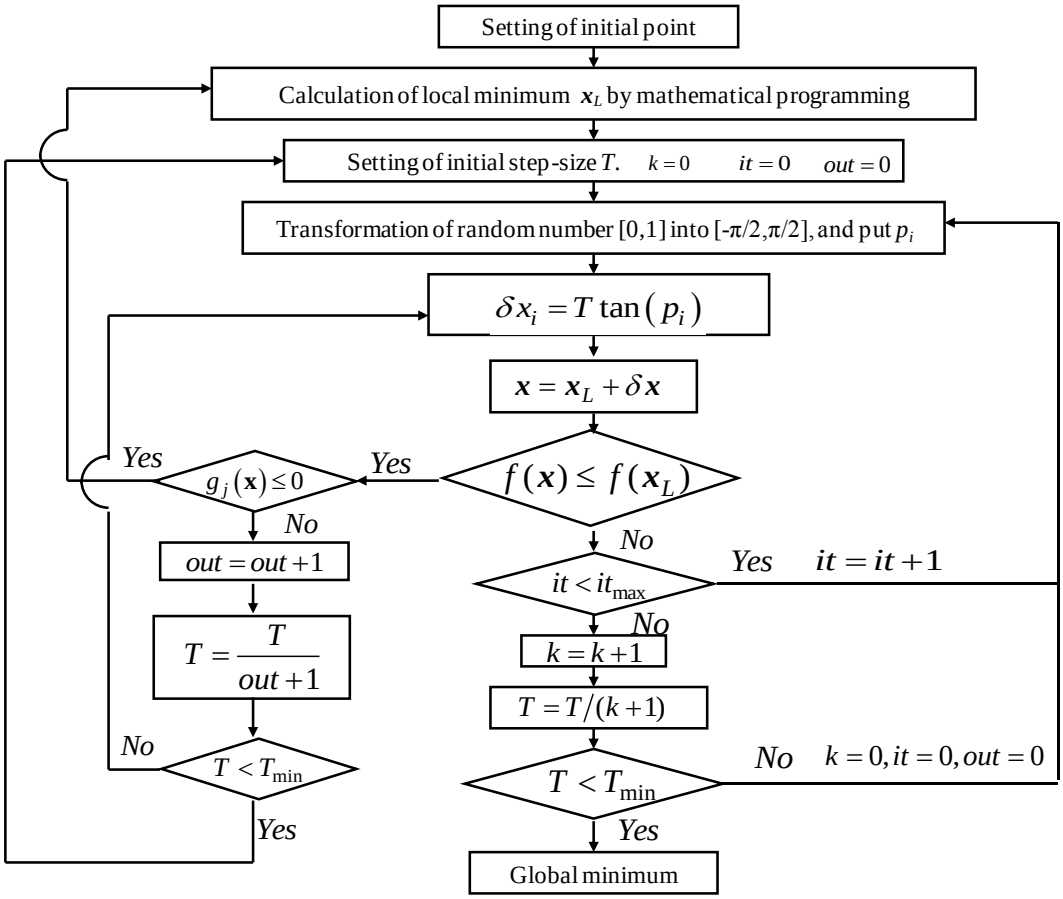


Fig.6 The algorithm of GRTA

In the GRTA, local minimum $\boldsymbol{x}_L = (x_{L,1}, x_{L,2}, \cdots, x_{L,n})^T$ can be found by the mathematical programming. Then, perturbation $\boldsymbol{\delta x}$ at $\boldsymbol{x}_L$ is provided, and new search point is generated as follow:

$$x_i = x_{L,i} + T\tan(p_i) \quad i = 1, 2, \cdots, n \tag{19}$$

where x_i represents the i-th design variable of new search point, T is the step-size, and p_i denotes the random number between $-\pi/2$ and $\pi/2$ that are generated to the i -th design variable. The objective function at new search point by Eq.(19) is compared with the one at $\boldsymbol{x}_L$. If the objective function at new search point is

improved, the feasibility is examined. If the new search point is feasible, this point is replaced the new start point.

Let us consider the similarity between the DE and the GRTA. In the GRTA, the improvement of the objective function is always examined when the new search point is generated. Therefore, it is clear that both methods have the descent property. The basis vector in the GRTA is the local minimum $\boldsymbol{x}_L$, the search direction vector is given by $\tan(p_i)$, and T is the step-size from basis vector. It is clear from Eq.(19) that the randomness provides the perturbation of $\boldsymbol{x}_L$. Compared Eq.(8) with Eq.(19), the methodology to generate the new search point is very similar. In addition, it is noteworthy that the search direction vector in both methods includes the randomness.

3.3 Similarities of three global optimization methods

Table 1 shows the summary of three global optimization methods. Table 1 is summarized the following point of view: (1) Generation of the new search point, (2) Search direction vector, (3) Step-size, and (4) Basis vector.

Table 1 Comparison of three global optimization techniques

	GRTA	PSO	DE
Generation of the new search point	$x_i = x_{L,i} + T\tan(p_i)$	$\boldsymbol{x}_d^{k+1} = \boldsymbol{x}_d^k + w\boldsymbol{v}_d^k + \alpha(\boldsymbol{q} - \boldsymbol{x}_d^k)$ $\alpha = c_1 r_1 + c_2 r_2 \quad \boldsymbol{q} = \frac{c_1 r_1 \boldsymbol{p}_d^k + c_2 r_2 \boldsymbol{p}_g^k}{c_1 r_1 + c_2 r_2}$	$v_{d,i}^k = x_{r1,i}^k + F_i(x_{r2,i}^k - x_{r3,i}^k)$ + Mutation
Search direction vector	$\tan(p_i)$	$\boldsymbol{q} - \boldsymbol{x}_d^k$	$\boldsymbol{x}_{r2}^k - \boldsymbol{x}_{r3}^k$
Step-size	T	$c_1 r_1 + c_2 r_2$	$F + d(rand(0,1) - 0.5)$
Basis vector	$\boldsymbol{x}_L$	$\boldsymbol{x}_d^k + w\boldsymbol{v}_d^k$	$\boldsymbol{x}_{r1}^k$

As described in introduction, we consider that there are two important factors for global optimization techniques. One is to include the search direction vector, and the other is to include the randomness. In the DE, the search direction vector is represented by the difference vector between $\boldsymbol{x}_{r2}{}^k$ and $\boldsymbol{x}_{r3}{}^k$. In the selection of three particles, the randomness is employed. New point $\boldsymbol{u}_d{}^k$ is generated by the crossover. In order to generate $\boldsymbol{u}_d{}^k$, the randomness is also employed. The randomness in the crossover is closely related to the neighborhood, as described in section 2.4. Therefore, the randomness provides the perturbation of $\boldsymbol{u}_d{}^k$.

Finally, it is clear from Eq. (5) that the objective at $\boldsymbol{u}_d^k$ is examined. In addition, Eq.(5) guarantees the descent property. The PSO has the search direction vector, implicitly. The randomness is employed in the search direction vector. In addition, the stochastic step-size is also included into the update scheme. The neighborhood of PSO is clarified in Ref.[14]. The descent property is also guaranteed by the g-best. Finally, let us consider the GRTA. In the GRTA, the randomness is employed for generating new search point. The GRTA utilized the mathematical programming in the minimization phase. Therefore, the search direction vector is used for local search.

As described above, it is possible to consider that these three methods have some common characteristics. These common characteristics may be one of the reasons why the DE can find a global minimum.

4. NUMERICAL EXAMPLES

Through numerical examples, the search ability of the DE is examined in this section. In the DE, the mutation ratio F and the crossover ratio Cr are determined in advance. Considered the similarity of both the DE and the GRTA, Eqs.(8) and (9) are used. The following values are used in this paper.

$$F = 0.8 \tag{20}$$

$$d = 0.0001 \tag{21}$$

In addition, crossover ratio Cr is set as 0.5. If different CPUs are employed, the computational time will be different. We consider that the efficiency of the algorithm should be measured by the function calls instead of the computational time. Therefore, the comparison with the function calls is a fair method for evaluating the efficiency.

4.1 Numerical Example for Unconstrained Optimization Problems

Typical numerical example shown in Table 2 is considered here.

Table 2 Benchmark problems for unconstrained optimization problems

No.	Name	Number of design variables	Objective	Side constraints	Objective at global minimum
1	2^n minima	10	$f(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{n}\left(x_i^4 - 16x_i^2 + 5x_i\right) \rightarrow \min$	$-5 \le \boldsymbol{x} \le 5$	$f(\boldsymbol{x}_G) = -391.661$
2	Griewank	10	$f(\boldsymbol{x}) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) \rightarrow \min$	$-10 \le \boldsymbol{x} \le 10$	$f(\boldsymbol{x}_G) = 0$
3	Ackley	10	$f(\boldsymbol{x}) = 22.71828 - 20\exp\left[-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right] - \exp\left[\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right] \rightarrow \min$	$-30 \le \boldsymbol{x} \le 30$	$f(\boldsymbol{x}_G) = 0$
4	Rastrigin	6	$f(\boldsymbol{x}) = \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right] \rightarrow \min$	$-5.12 \le \boldsymbol{x} \le 5.12$	$f(\boldsymbol{x}_G) = 0$
5	Michalewics	5	$f(\boldsymbol{x}) = -\sum_{i=1}^{n}\sin(x_i) \times \left\{\sin(\frac{ix_i^2}{\pi})\right\}^{20} \rightarrow \min$	$0 \le \boldsymbol{x} \le \pi$	$f(\boldsymbol{x}_G) = -4.687658$
6	Branin	2	$f(\boldsymbol{x}) = \left[x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right]^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10 \rightarrow \min$	$-10 \le \boldsymbol{x} \le 10$	$f(\boldsymbol{x}_G) = 0.397887$
7	Six-Hump Camel-Back	2	$f(\boldsymbol{x}) = x_1^2\left(4 - 2.1x_1^2 + \frac{1}{3}x_1^4\right) + x_1x_2 + 4x_2^4 - 4x_2^2 \rightarrow \min$	$-3 \le \boldsymbol{x} \le 3$	$f(\boldsymbol{x}_G) = -1.031628$
8	Shubert	2	$f(\boldsymbol{x}) = \sum_{i=1}^{5} i\cos\left[(i+1)x_1 + i\right]\sum_{i=1}^{5} i\cos\left[(i+1)x_2 + i\right] \rightarrow \min$	$-10 \le \boldsymbol{x} \le 10$	$f(\boldsymbol{x}_G) = -186.730909$
9	Bohachevsky	2	$f(\boldsymbol{x}) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \rightarrow \min$	$-50 \le \boldsymbol{x} \le 50$	$f(\boldsymbol{x}_G) = 0$
10	Easom	2	$f(\boldsymbol{x}) = -\cos(x_1)\cos(x_2)\exp[-(x_1-\pi)^2 - (x_2-\pi)^2] \rightarrow \min$	$-100 \le \boldsymbol{x} \le 100$	$f(\boldsymbol{x}_G) = -1.0$
11	Goldstein & Price	2	$f(\boldsymbol{x}) = [1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \rightarrow \min$	$-2 \le \boldsymbol{x} \le 2$	$f(\boldsymbol{x}_G) = 3.000$

The swarm population size is set to 30, and the maximum number of search iterations is set to 500 (From No.1 to No.5). The swarm population size is set to 20, and the maximum number of search iterations is set to 200 (From No.6 to No.12). All particles are distributed randomly between x_i^L and x_i^U at k=1. In the PSO, the inertia term w decreases linearly as follow [26]:

$$w = w_{\max} - (w_{\max} - w_{\min})/k_{\max} \times k \tag{22}$$

where $w_{\max}$=0.9 and $w_{\min}$=0.4 are used. Twenty trials have been performed in order to examine the stability and the accuracy of solution. In addition, the function calls are also one of the important aspects in order to examine the efficiency. In the population-based optimization techniques, the function calls can be simply calculated by multiplying the total number of iterations by the population size when no convergence criterion is assigned. In this paper, the following convergence criterion to evaluate the function calls is used: (a) the objective function is not improved through 20 iterations. The numerical result is shown in Table 3. These numerical test problems presented in this sub-section are solved by Mathematica (version 6.0.3). Four global optimization techniques

(Nelder-Mead Method, Simulated Annealing, Random Search, and Differential Evolution) are included into the Mathematica, and the readers can use these global optimization techniques by NMinimize command. The PSO and the DE is programmed by the authors (Fortran 77 code is used to construct the programming code of the PSO and the DE), so that the DE included into Mathematica is not used. In addition, the function calls in Mathematica are not clear. As the result, it is impossible to compare the efficiency exactly.

Twenty trials were performed with different random seed, and the results are listed in Table 3.

Table 3 Results of benchmark problems in Table 3

Name	Methods	Objective at global minimum	Best objective	Worst objective	Maen value of objective	Standard deviation of objective	Average function calls
2n minima	Nelder-Mead Method	$f(x_G) = -391.661$	-391.662000	-306.841000	-347.231886	20.034286	N/A
	Simulated Annealing		-377.525000	-335.115000	-357.733286	12.920243	N/A
	Random Search		-391.662000	-349.251000	-374.697600	10.733611	N/A
	Particle Swarm Optimization		-391.657830	-390.988150	-391.510196	0.200000	13014
	Differential Evolution		-391.662000	-391.662000	-391.662000	1.7302E-13	9729
Griewank	Nelder-Mead Method	$f(x_G) = 0$	0.000000	0.243823	0.040340	0.062167	N/A
	Simulated Annealing		0.000000	0.549409	0.084307	0.143626	N/A
	Random Search		0.000000	1.56948E-32	4.51258E-34	2.65247E-33	N/A
	Particle Swarm Optimization		8.930000E-04	7.300000E-02	2.910000E-02	2.52000E-02	11997
	Differential Evolution		0.000000	4.95733E-19	4.57183E-20	1.13823E-19	11997
Ackley	Nelder-Mead Method	$f(x_G) = 0$	-1.82846E-06	13.010400	5.456291	2.969282	N/A
	Simulated Annealing		11.699000	18.831700	16.983683	1.662878	N/A
	Random Search		11.109400	17.867300	15.813129	1.674436	N/A
	Particle Swarm Optimization		0.057838	0.098645	0.083662	0.012248	14919
	Differential Evolution		-1.82846E-06	1.82846E-06	-1.61949E-06	8.61220E-07	13527
Rastrigin	Nelder-Mead Method	$f(x_G) = 0$	2.984880	39.798200	13.247134	9.063731	N/A
	Simulated Annealing		2.984880	33.828500	12.337470	6.484971	N/A
	Random Search		2.984880	14.924400	9.125183	3.241914	N/A
	Particle Swarm Optimization		1.336274	3.167823	2.347625	0.566430	14856
	Differential Evolution		7.3326E-26	4.974800	2.558468	1.305897	6591
Michalewics	Nelder-Mead Method	$f(x_G) = -4.687658$	-4.687660	-2.622520	-3.963637	0.564961	N/A
	Simulated Annealing		-4.687660	-2.693030	-3.962234	0.593549	N/A
	Random Search		-4.687660	-3.694590	-4.347061	0.286999	N/A
	Particle Swarm Optimization		-4.687521	-4.645192	-4.669190	0.020470	14547
	Differential Evolution		-4.687658	-4.687658	-4.687658	0	7578
Branin	Nelder-Mead Method	$f(x_G) = 0.397887$	0.397887	2.791180	1.115875	1.156070	N/A
	Simulated Annealing		0.397887	0.397887	0.397887	5.85139E-17	N/A
	Random Search		0.397887	0.397887	0.397887	5.85139E-17	N/A
	Particle Swarm Optimization		0.397887	0.397937	0.397892	1.55298E-05	3462
	Differential Evolution		0.397887	0.397887	0.397887	0	1436
Six-Hump Camel-Back	Nelder-Mead Method	$f(x_G) = -1.031628$	-1.031630	-0.215464	-0.940945	0.272055333	N/A
	Simulated Annealing		-1.031628	-1.031628	-1.031628	0	N/A
	Random Search		-1.031628	-1.031628	-1.031628	0	N/A
	Particle Swarm Optimization		-1.031628	-1.031621	-1.031627	2.62217E-06	3792
	Differential Evolution		-1.031628	-1.031628	-1.031628	2.34056E-16	1394
Shubert	Nelder-Mead Method	$f(x_G) = -186.730909$	-186.731000	-123.577000	-167.784700	30.506211	N/A
	Simulated Annealing		-186.731000	-79.410900	-175.998990	33.937595	N/A
	Random Search		-186.731000	-52.553400	-154.367040	46.541084	N/A
	Particle Swarm Optimization		-186.730875	-186.268070	-186.594800	0.142103	3636
	Differential Evolution		-186.730909	-186.730697	-186.730882	6.65172E-05	2746
Bohachevsky	Nelder-Mead Method	$f(x_G) = 0$	0.000000	0.412927	0.041293	0.130579	N/A
	Simulated Annealing		0.000000	0.469882	0.129574	0.209208	N/A
	Random Search		0.000000	0.000000	0.000000	1.29927E-32	N/A
	Particle Swarm Optimization		0.000000	0.000099	0.000023	3.17084E-05	3786
	Differential Evolution		0.000000	0.000000	0.000000	0	1304
Easom	Nelder-Mead Method	$f(x_G) = -1.0$	-1.000000	0.000000	-0.100000	0.316228	N/A
	Simulated Annealing		-1.000000	0.000000	-0.100000	0.316228	N/A
	Random Search		0.000000	0.000000	0.000000	0	N/A
	Particle Swarm Optimization		-1.000000	-0.999042	-0.999832	3.00163E-04	3470
	Differential Evolution		-1.000000	-1.000000	-1.000000	0	1786
Goldstein & Price	Nelder-Mead Method	$f(x_G) = 3.000$	3.000000	84.000000	11.100000	25.614449	N/A
	Simulated Annealing		3.000000	3.000000	3.000000	0	N/A
	Random Search		3.000000	3.000000	3.000000	0	N/A
	Particle Swarm Optimization		3.000000	3.000035	3.000006	1.08335E-05	3444
	Differential Evolution		3.000000	3.000000	3.000000	0	1150

It is clear from Table 3 that the DE is superior to the PSO through all benchmark problems. In particular, the DE can find the global minimum of all

benchmark problems. The standard deviation of objective is one of the important factors for evaluating the robustness of algorithm. Therefore, smaller standard deviation of objective implies the robustness of the algorithm. It is clear from Table 3 that the DE is more robust algorithm than other four optimization techniques under the parameter settings employed in this paper. The efficiency is also compared between the PSO and the DE. Through all benchmark problems, the DE could achieve smaller function calls the PSO. As mentioned above, smaller function calls implies the efficient search. It is clear from Table 3 that the DE is superior to the PSO under the parameter settings employed in this paper. The convergence of the objective function is examined on the Griewank function and the Ackley function. In particular, the relation between the standard deviation of objective function and the search iteration is examined for investigating the search performance of the DE. Fig.7 and Fig.8 show the convergence of the the Griewank function and the Ackley function, respectively.

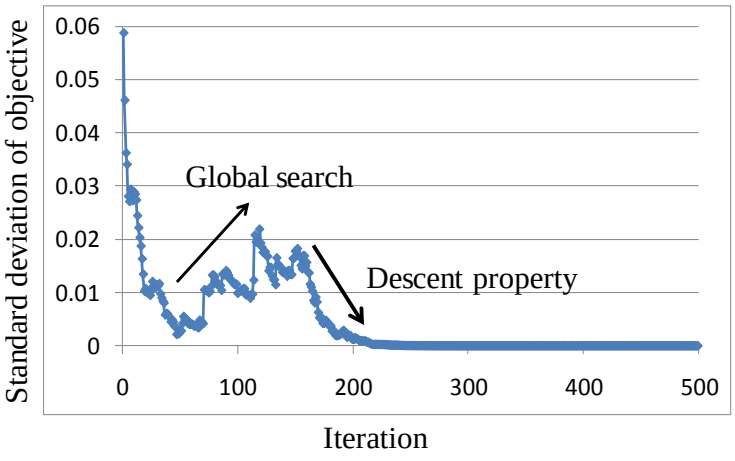


Fig.7 Convergence of the Griewank function

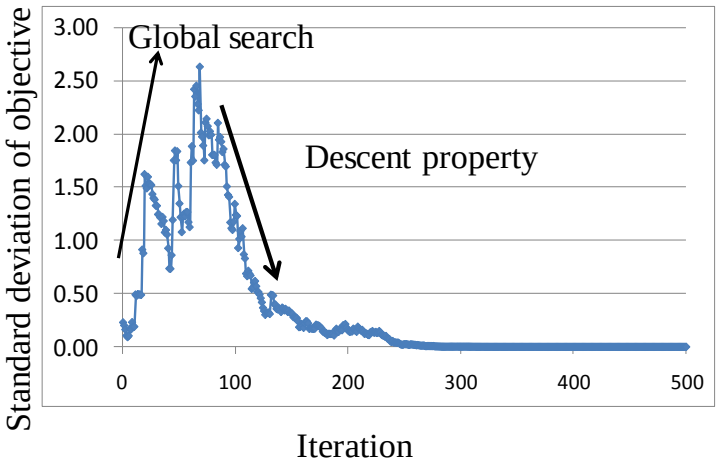


Fig.8 Convergence of the Ackley function

From the Figs.7 and 8, the global search is performed till the medium search iteration. Random selection of three particles may leads to the global search. At the end of iteration, the descent property is very clear. This implies that all particles flock around the global minimum.

4.2 Optimum Design of Tension/Compression Spring

One of the most famous test problem proposed by Arora [27] was considered. Many researchers have tested as one of the benchmark problems in the structural optimization [27,28, 29]. The design variables are (1) the wire diameter d (=x_1), (2) the mean coil diameter D (=x_2) and (3) the number of active coils N (=x_3). The problem can be stated as follows:

$$f(\boldsymbol{x}) = (2 + x_3)x_1^2 x_2 \rightarrow \min \tag{23}$$

$$g_1(\boldsymbol{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \tag{24}$$

$$g_2(\boldsymbol{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0 \tag{25}$$

$$g_3(\boldsymbol{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \tag{26}$$

$$g_4(\boldsymbol{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \tag{27}$$

$$0.05 \leq x_1 \leq 2.00 \tag{28}$$

$$0.25 \leq x_2 \leq 1.30 \tag{29}$$

$$2.00 \leq x_3 \leq 15.0 \tag{30}$$

The penalty function approach to handle the behavior constraints, which are represented from Eqs.(24) to (27), is used. In this paper, the following penalty function is adopted, and the augmented objective function to be minimized is constructed.

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) + r \times penalty \rightarrow \min \tag{31}$$

$$r = (1 + |f(\boldsymbol{x})|)^q \tag{32}$$

$$penalty = \begin{cases} \sum_{j=1} \exp(1 + g_j(\boldsymbol{x})) & \cdots g_j(\boldsymbol{x}) > 0 \\ 0 & \cdots g_j(\boldsymbol{x}) \le 0 \end{cases} \tag{33}$$

q in Eq.(32) is a real number greater than 1. In this paper, q is set as 2. The penalty parameter r in Eq.(32) is automatically determined by using the above penalty function approach.

The population size is set to 20, and the number of maximum search iterations is set to 500. Eleven trials were performed for comparing with previous researches. The result by the DE is listed in the last column in Table 4, and it was found that the best result could be obtained by the DE.

Table 4 Comparison of results of minimum weight design of tension/compression spring

Design Variables	Best solutions found				
	Arora	Coello	Ray	Hu	DE
x 1 (d)	0.053396	0.05148	0.050417	0.051466	0.0516868
x 2 (D)	0.39918	0.351661	0.321532	0.351384	0.3566636
x 3 (N)	9.1854	11.632201	13.979915	11.608659	11.2878946
g 1($\boldsymbol{x}$)	0.000019	-0.00208	-0.001926	-0.003336	-8.22116E-10
g 2($\boldsymbol{x}$)	-0.000018	-0.00011	-0.012944	-0.00011	-1.1952E-11
g 3($\boldsymbol{x}$)	-4.123832	-4.026318	-3.89943	-4.026318	-4.0555802
g 4($\boldsymbol{x}$)	-0.698283	-0.731239	-0.752034	-0.731324	-0.7277664
$f(\boldsymbol{x})$	0.01273	0.012705	0.01306	0.012667	0.0126612
Function Call	N/A	900000	1291	N/A	5696
Averege of $f(\boldsymbol{x})$	N/A	0.012769	0.013436	0.012719	0.0126612
Worst of $f(\boldsymbol{x})$	N/A	0.012822	0.01358	N/A	0.0126612
Standard Deviation of $f(\boldsymbol{x})$	N/A	3.9390E-05	N/A	6.4660E-05	2.4087E-09

4.3 Topology Optimization of Truss Structure in two dimensions

Let us consider the topology optimization problem of truss structure in two dimensions. This problem is taken from Ref.[24]. In general, topology optimization problem is multi-modal problem. The design variables are the cross-section area of each member. The objective function is to minimize the total volume of structure, and the nodes displacements are constrained. The structural optimization problem is generally described as follows:

$$f(\boldsymbol{A}) = \sum_{i=1}^{n} A_i L_i \rightarrow \min \tag{34}$$

$$g_j(\boldsymbol{A}) = u_k(\boldsymbol{A}) / u_a - 1 \le 0 \qquad j = 1, 2, \cdots, m \tag{35}$$

$$A_{i,\min} \le A_i \le A_{i,\max} \qquad i = 1, 2, \cdots, n \tag{36}$$

where $\boldsymbol{A} = (A_1, A_2, \cdots, A_n)^T$ represents the design variables, which is the cross-section area. $f(\boldsymbol{A})$ is the total volume of truss structure to be minimized, $g_j(\boldsymbol{A})$ denotes the nodal displacements , which are the behavior constraints, and u_a is the allowable displacement.

The truss structure in two dimensions considered in this sub-section is show in Fig.9. This structure consists of 9-node, and 28-elements. As the result, the number of design variables is 28. The Node 1 and 3 are completely fixed, and two loads are applied to the Node 4 and 7. The magnitude of two loads is P=1000[N]. The distance a between nodes is set to 100[mm], and the allowable displacement u_a is set to 1.50x10^{-2}[mm]. Two nodes displacements, which are the Node 4 and 7, are constrained. Young's modulus E in this structure is 210[GPa], and $A_{i,\min}$ and $A_{i,\max}$ are set to 1[mm^2] and 1000[mm^2], respectively.

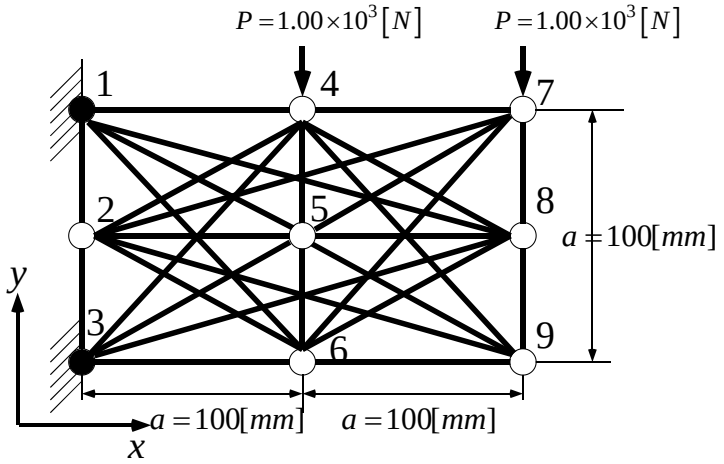
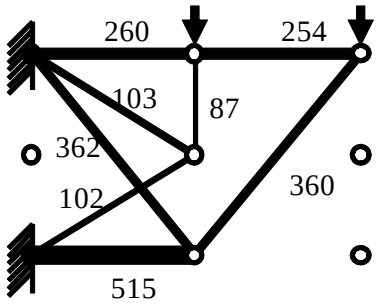
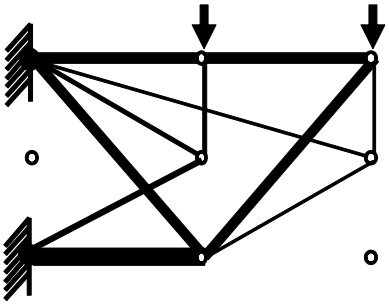
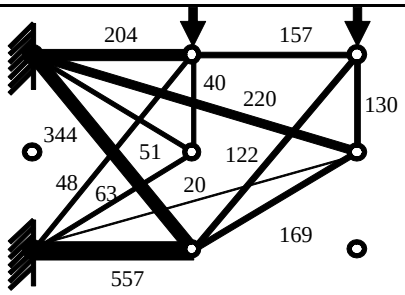
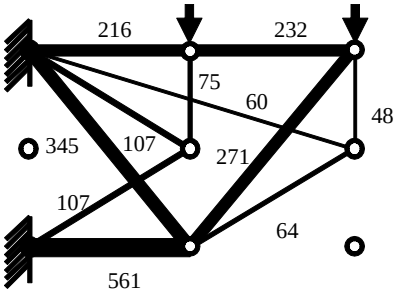
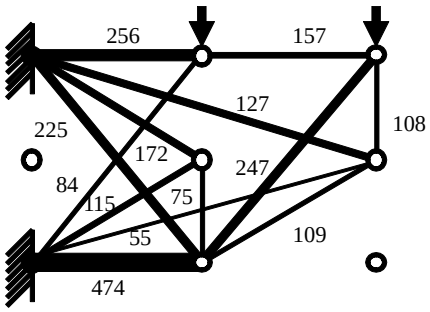


Fig.9 Truss structure in two dimensions

This problem has been solved by many methods, and Table 5 shows the results. FC in table 5 represents the function calls. Please refer Ref.[24] on detailed some parameter settings.

Table 5 Comparison of optimum topology by some global optimization method

Method	Optimum topology	Objective	Function call
GRTA		2.35E+05	1,825
PSO		2.37E+05	200,000
Simple GA		2.56E+05	1,000,000
Distributed GA		2.40E+05	1,000,000
SA		2.60E+05	142,801

The population size is set to 50, and the number of maximum search iteration is set to 4000, in order to solve this problem by the DE. The result by the DE is shown in Fig.10. The design variables attained at the lower bound are omitted in

Fig.10, and the objective function at this result is 2.35×10^5[mm^3]. From Fig.10, it is clear that the optimum topology by the DE is almost same one by the GRTA.

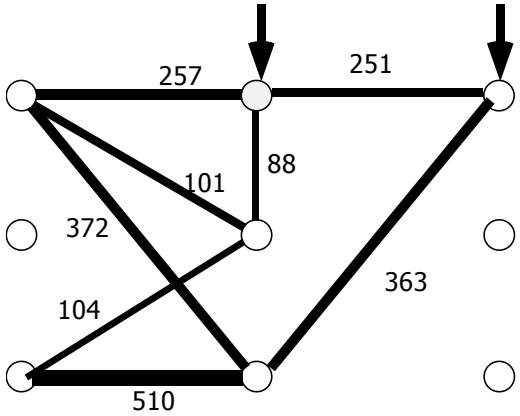


Fig.10 Optimum topology by the DE

The relationship between the standard deviation of objective function and the search iteration is shown in Fig.11.

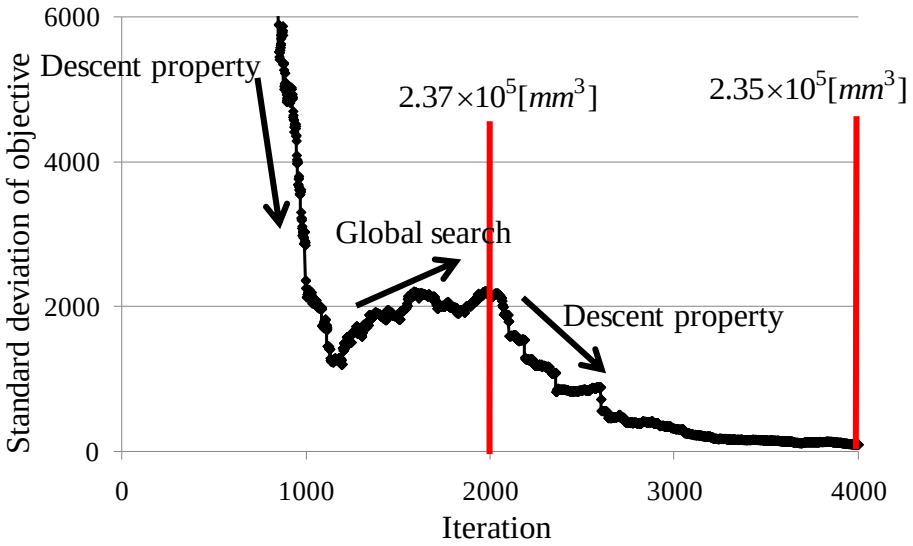


Fig.11 Convergence in the case of topology optimization problem

It is clear from Fig.11 that the result by the PSO may be local minimum, while the DE can find the global minimum as the search proceeds. In this problem, the search ability of the DE is excellent in comparison with the PSO.

5. CONCLUSIONS

As mentioned in introduction, we consider that two important factors are included into the global optimization algorithms. One is the search direction vector, and the other is the randomness. The search direction vector can clarify the search direction, and the randomness provides the perturbation. In addition, in the

case of the population-based optimization techniques, the randomness can be expected to provide the diversity among the particles. From these points of view, the characteristics of the DE as the global optimization method were examined. It is well known that the difference vector provides the search direction vector. However, the roles of the search direction vector are not well explained. We clarified the roles of the search direction vector. In addition, it was clear that the randomness commonly provided the perturbation. From investigations in this paper, the common characteristics among three methods were clarified. The DE was applied to eleven benchmark problems, and was also applied to two structural optimization problems. The function call was employed for measuring the efficiency. The standard deviation of objective was also employed for measuring robustness of the algorithm. The DE can find the global minimum of the benchmark problems with smaller function calls, compared with the PSO. It is also clear that the DE is more robust algorithm than other optimization techniques under the parameter settings employed in this paper. In addition, it was shown from numerical examples that the DE is a global optimization method that the descent property and the global search are well-balanced. In particular, the topology optimization of the truss structure is one of the most difficult problems for finding the global minimum. The DE could find the global minimum with the same function call of the PSO. However, it is also noted that the numerical results do not always imply that the DE is the best global optimizer. Therefore, the DE could obtain better results under the limited numerical examples.

Acknowledgements.

The Authors would like to thank to Prof. Yamakawa, H.(Waseda University), Prof. Nakayama, H.(Konan University), and Prof. Sugimoto, H.(HokkaiGakuen University) who provided the useful and constructive comments. Finally, we would like to thank to all reviewers who gave us the constructive comments for improving the quality of this paper.

REFERENCES

1. Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J. ed., (1989), Handbooks in Operations Research and Management Science Vol.1 OPTIMIZATION, Elsevier Science Publishers.
2. Levy, A. V., Montalvo, A., (1985), The Tunneling Algorithm for the Global Minimization Functions, SIAM Soc. Ind. Appl. Math.d J. Sci. Stat. Comput., Vo.1, No.6, pp. 15–29.
3. Goldberg, D.E., (1989), Genetic Algorithm in Search, Optimization and Machine, Addison-Wesley.
4. Kennedy, J., Eberhart, R.C., (2001), Swarm Intelligence, Morgan Kaufmann Publishers.

5. Storn, R., Price, K.V., (1997), Differential Evolution -a simple and efficient heuristic for global optimization over continuous spaces- J. of Global Optimization, Vol.11, pp.341-359.
6. Dorigo, M., Stutzle, T., (2004), Ant Colony Optimization, MIT Press, Cambridge, MA. ,
7. Venter, G., Sobieski, J.S., (2003), Particle Swarm Optimization, AIAA J., Vol.41-8, pp. 1583-1589.
8. Venter G., Sobieski J.S., (2004), Multidisciplinary Optimization of a Transport Aircraft Wing using Particle Swarm Optimization, Struct. and Multidisc. Optim., Vol. 26, No.1-2, pp.121-131.
9. Fourie, P.C., Groenwold, A.A., (2002), The Particle Swarm Optimization Algorithm in Size and Shape Optimization, Struct. and Multidisc. Optim., Vol. 23-4, pp.259-267.
10. Schutte, J.F., Groenwold, A.A., (2003), Sizing Design of Truss Structures Using Particle Swarms, Struct. and Multidisc. Optim., Vol.25, pp.261-269
11. Luh, G.C., Lin, C.Y., (2008), Optimal Design of Truss Structures Using Ant Algorthm, Struct. And Multidisc. Optim., Vol.36, pp.365-379.
12. Serra, M., Venini, R., (2006), On Some Applications of Ant Colony Optimization Metaheuristic to Plane Truss Optimization, Struct. And Multidisc. Optim., Vol.32, pp.499-506.
13. Abachizadeh, M., Tahani, M., (2009), An Ant Colony Optimization Approach to Multi-Objective Optimal Design of Symmetric Hybrid Laminates for Maximum Fundamental Frequency and Minimum Cost, Struct. And Multidisc. Optim., Vol.37, pp.367-376.
14. Kitayama, S., Arakawa, M., Yamazaki, K., (2005), Penalty Function Approach for the Mixed Discrete Non-Linear Problems by Particle Swarm Optimization, Struct. and Multidisc. Optim., Vol. 32, No.3, pp.191-202.
15. Clerc, M., Kennedy, J., (2002), The Particle Swarm Optimization – Explosion, Stability, and Convergence in a Multidimensional Complex System Space, IEEE Trans. on Evolutionary Computation, Vol.6, No.1, pp.58-73.
16. Clerc, M., (2006), Particle Swarm Optimization, ISTE USA.
17. Engelbrecht, A.D., (2005), Fundamentals of Computational Swarm Intelligence, Wiley.
18. Stutzle, T., Dorigo, M., (2002), A Short Convergence Proof for a Class of Ant Colony Optimization Algorithms, IEEE Trans. on Evolutionary Computation, Vol.6, No.4, pp.358-365.
19. Price, K.V., Storn, R., Lampinen, J.A., (2006), Differential Evolution -A Practical Approach to Global Optimization- , Springer.
20. Chakraborty U.K. ed., (2008), Advances in Differential Evolution, Springer.
21. Coello, C., et.al., (2007),.Evolutionary Algorithms for Solving Multi-Objective Problems (2nd Edition), Springer.
22. Onwubolu, G., Davehdra, D., (2009), Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization, Springer.
23. Dasgupta, S., Das, S., Biswas, A., Abraham, A., On Stability and Convergence of the population-dynamics in Differential Evolution, (2009), AI Communications, Vol. 22, pp.1-20.
24. Kitayama, S., Yamazaki, K., (2005), Generalized Random Tunneling Algorithm for Continuous Design Variables, Trans. of ASME, J. of Mechanical Design, Vol.127, pp.408-414.
25. Vanderplaats, G.N., (1999), Numerical Optimization Techniques for Engineering Design with Applications (3rd Edition)", VR&D.
26. Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S., Nakanishi, Y., (2000) A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment, IEEE Trans Power Systems, No.15, Vol. 4, pp.1232–1239
27. Arora, J.S., (1989), Introduction to Optimum Design, McGraw-Hill, New York.
28. Ray, T., Saini, P., (2001), Engineering Design Optimization Using Swarm with an Intelligent Information Sharing among Individuals, Eng. Optim., Vol.33, pp.735-748.
29. Coello Coello C.A., (2000), Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems, Computers in Industry, Vol.41, pp.113-127.
30. Hu X.H., Eberhart, R.C., Shi, Y.H., (2003), Engineering Optimization with Particle Swarm, IEEE Swarm Intelligence Symposium, pp.53-57.