International Journal of Innovative Computing, Information and Control
Volume **15**, Number **2**, April **2019**

ICIC International ©2019 ISSN 1349-4198
pp. **607–616**

PERFORMANCE ENHANCEMENT OF DIFFERENTIAL EVOLUTION BY DIRECT ALGORITHM

Jun-ichi Kushida, Akira Hara and Tetsuyuki Takahama

Graduate School of Information Sciences
Hiroshima City University
3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima-shi 731-3194, Japan
{ kushida; ahara; takahama }@hiroshima-cu.ac.jp

Received June 2018; revised October 2018

Abstract. *Differential Evolution (DE) is one of the efficient Evolutionary Algorithms (EAs) for the continuous optimization domain. Similar to other EAs, DE algorithm works on a population of candidate solutions. Population initialization in DE is an important operation because it can affect the convergence speed and also the quality of the obtained solution. If we can provide good initial population to DE, improvement of search efficiency can be expected. However, in black-box optimization, there exists no prior information about the search landscape of a given problem. Therefore, in this research we use DIRECT (DIviding RECTangles) algorithm to provide a good initial population to DE. DIRECT is a deterministic global optimization algorithm for bound-constrained problems. The algorithm, based on a space-partitioning scheme, performs both global exploration and local exploitation by one tuning parameter. In the proposed method, named DE-DIRECT, first the search by DIRECT is performed until a certain number of iterations. Next, the solutions obtained by DIRECT are used as the initial individuals of DE. The remaining search is performed by DE using DIRECT's solutions until the total budget is exhausted. In order to extract effective individuals for DE from the solution set of DIRECT, we introduce a selection method considering diversity as well as accuracy of solutions. The effectiveness of the proposed DE-DIRECT is examined and discussed by experiments on various benchmark functions.*
Keywords: Differential evolution, DIRECT, Population initialization

1. **Introduction.** Evolutionary Algorithms (EAs), such as Genetic Algorithm (GA) [1], Particle Swarm Optimization [2], Differential Evolution (DE) [3], are population-based stochastic optimization method. EAs have successfully been applied to optimization problems in various research fields [4]. However, in some real-world applications, execution of a very expensive simulation may be required to calculate the function value of the search point. In these problems, EAs must solve optimization problems under tight function evaluation budget. Differential Evolution (DE) is a stochastic direct search optimization method for solving global optimization problems in continuous domain. It has exhibited excellent performance for a wide range of benchmark problems [5, 6]. However, in order to find high quality solutions in expensive problems, further improvement is required.

In this research, we focus on the method of generating initial individuals of DE for enhancing its performance. The selection of the initial population in a population-based heuristic optimization method is very important, since it affects the search for several iterations and often has an influence on the obtained solution [7]. Therefore, various population initialization techniques have been proposed [8]. This paper presents a heuristic method based on the combination of DIRECT (DIviding RECTangles) [9] and DE. The

DOI: 10.24507/ijicic.15.02.607

DIRECT is a global optimization method first motivated by Lipschitz optimization. This method treats the design variable space as a hypercube and repeats partitioning based on the evaluation value of center point of the hyper-rectangles. In the proposed method, called DE-DIRECT, a search by DIRECT is performed first using a part of the given function evaluation budgets. Next, some individuals are selected from the search points obtained by DIRECT, and they are used as a part of initial population of DE. In order to extract effective individuals from DIRECT, we introduce a selection method considering diversity as well as accuracy of solution. By using the better solution set provided by DIRECT as the initial population, DE can start the search from the promising area in the solution space. In numerical experiments, we find out DE-DIRECT can enhance the search performance of DE for the high dimensional benchmark problems under tight function evaluation budget.

This paper is organized as follows. In the next section, we briefly describe DIRECT algorithm. In Section 3 we describe DE framework. In Section 4 we present our DE-DIRECT algorithm in detail. In Section 5 we provide computational results using benchmark problems and compare our algorithm to original DE and DIRECT. Finally, conclusions are summarized in Section 6.

2. **DIRECT.**

DIRECT is a deterministic sampling method for global minimum of a real valued objective function over a bound-constrained domain. The method does not need derivative information and the progress of the optimization is governed only by evaluations of the objective function. In this study, the following optimization problem with lower bound and upper bound constraints will be discussed.

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & L_j \leq x_j \leq U_j, \quad j = 1, \ldots, D \end{array} \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$ is a D-dimensional vector and $f(\boldsymbol{x})$ is an objective function. Values L_j and U_j are the lower bound and the upper bound of x_j, respectively. In the search space, every point satisfies the lower and upper bound constraints.

The procedure of DIRECT algorithm is outlined in Algorithm 1. It iteratively identifies and divides potentially-optimal hyper-rectangles until the evaluation budget is exhausted. FE_{direct} is the maximum number of function evaluations. In the initialization stage, the original hyper-rectangular search space is normalized into a unit D-dimensional hypercube

Algorithm 1 DIRECT

```
Normalize the search space to be the unit hypercube with center point c;
Set iteration t = 1;
Evaluate f(c), f_min = f(c), g = 0, FE = 1;
while FE < FE_direct do
   Identify the potentially optimal hyper-rectangle set M ⊂ H;
   while M ≠ ∅ do
      Take j ∈ M
      Sample new points, evaluate f at the new points and divide the hyper-rectangle;
      Update f_min, FE = FE + ΔFE
      Set M = M \ {j}
   end while
   t = t + 1
end while
```

by Equation (2).

$$\Omega = \{x \in R^D : 0 \le x_j \le 1\}, \quad j = 1, \dots, D \tag{2}$$

2.1. **The first iteration.** The DIRECT algorithm begins by evaluating f at the center of Ω, $c = (1/2, \dots, 1/2)$. In the first iteration, Ω itself is the first potentially optimal hyper-rectangle. Next step is to divide this hypercube. The algorithm samples the objective values at the points $c \pm \frac{1}{3}e_j$, where e_j is the jth unit vector. The $2D$ points sampled become centers of their own hyper-rectangles, and the algorithm continues to the next iteration.

2.2. **General iterations of DIRECT.** After the first iteration, the algorithm identifies potentially optimal hyper-rectangles. Let H be the set of hyper-rectangles created by DIRECT after t iterations and let $f_{\min}$ be the best value of the objective function found so far. A hyper-rectangle $i \in H$ with center c_i and size d_i is said to be potentially optimal if there exists $\hat{K}$ such that for an arbitrarily small $\epsilon > 0$,

$$f(c_i) - \hat{K}d_i \le f(c_j) - \hat{K}d_j, \quad \forall_j \in H \tag{3}$$

$$f(c_i) - \hat{K}d_i \le f_{\min} - \epsilon|f_{\min}| \tag{4}$$

In the above expressions, ϵ is a balance parameter which provides the user control of the balance between local and global search. Values of $\epsilon \in [10^{-7}, 10^{-3}]$ are reported to work well in [9]. In the original DIRECT, hyper-rectangle size is measured by the distance from its center to a vertex. In the paper, we measure hyper-rectangles by their longest side [10].

Figure 1 is a geometric interpretation of this definition. First, the hyper-rectangles in H is ordered in groups according to the size d_i. Equations (3) and (4) correspond to selecting rectangles on the lower convex hull of the graph. ϵ can control the position of square dot $(0, f_{\min} - \epsilon|f_{\min}|)$ which alters the lower convex hull. As a result, good hyper-rectangles in the smaller size groups may exclude. Meanwhile, the best hyper-rectangle in the largest size group is always selected.

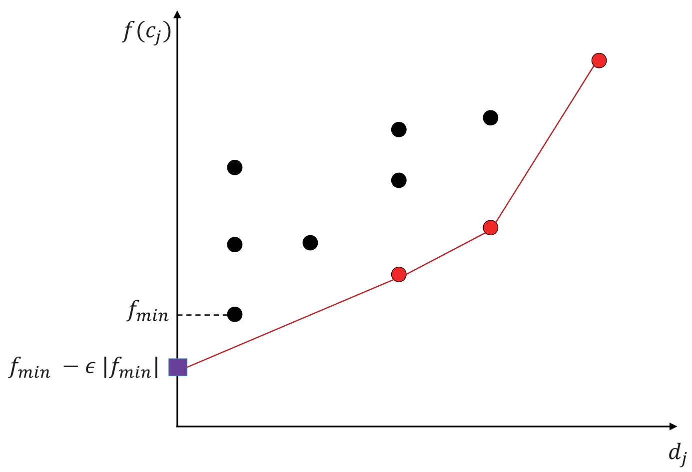


FIGURE 1. Identification of potentially optimal hyper-rectangles

Potentially optimal hyper-rectangles are subdivided along their long coordinate directions. This strategy ensures equal sampling in every dimension. The procedure of division is shown as follows.

Step 1: Let R be a potentially optimal hyper-rectangle with center c.
Step 2: Let ξ be the maximal side length of R.

Step 3: Let I be the set of coordinate directions corresponding to sides of R with length ξ.

Step 4: Evaluate the objective function at the points $c \pm \frac{1}{3}\xi e_j$, for all $j \in I$, where e_j is the jth unit vector. Increase of the number of evaluations ΔFE is the number of newly sampled points.

Step 5: Let $w_j = \min\left\{f\left(c \pm \frac{1}{3}\xi e_j\right)\right\}$.

Step 6: Divide the hyper-rectangle containing c into thirds along the dimensions in I, starting with the dimension with lowest w_j and continuing to the dimension with the highest w_j.

Figure 2 shows an example of division and selection of potential optimal hyper-rectangle. In Figure 2(a), the hypercube is cut in the direction perpendicular to x_2. Then, the hypercube is cut in the direction perpendicular to x_1. The hypercube, with a function value at the center point equal to 2, is the potential optimal hyper-rectangle after two divisions. Figure 2(b) is the next step in the algorithm. DIRECT will divide the shaded area. Figure 2(c) shows the third step in the algorithm for this example. In this step DIRECT will divide the two shaded rectangles. The rectangle in the lower area is a square; therefore, it is divided twice as Figure 2(a). Also, the top larger area is a rectangle and it is divided once.

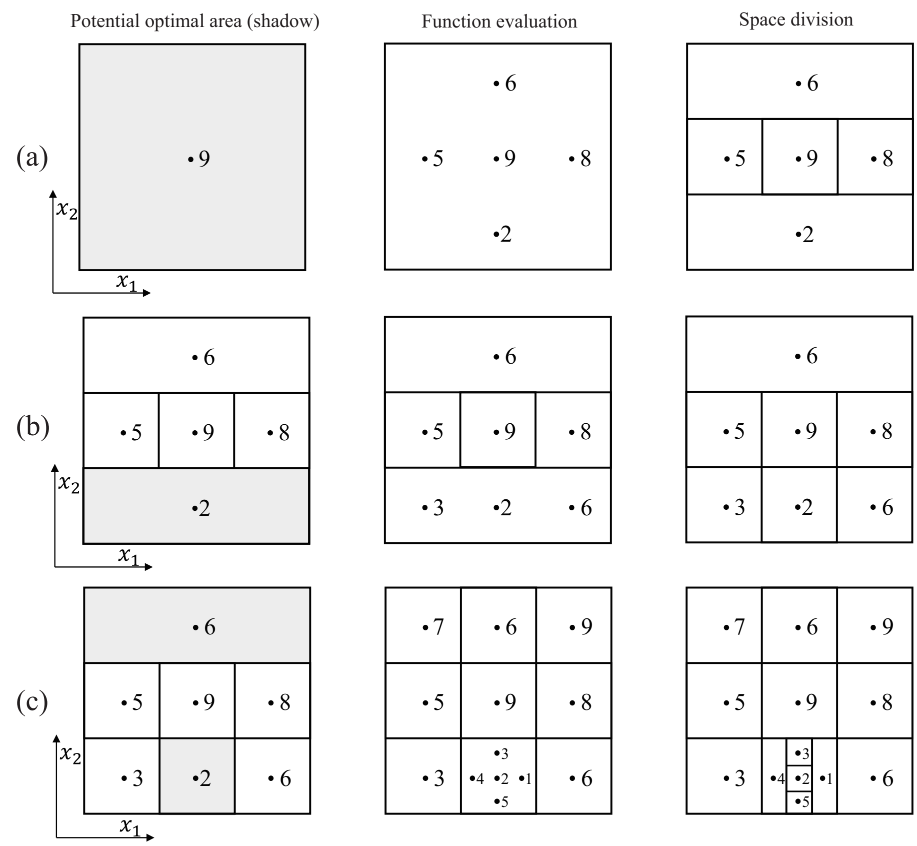


FIGURE 2. Division and selection of potential optimal hyper-rectangle

3. **Differential Evolution.** DE is one of the variants of evolutionary algorithms that use a population. An individual of DE is represented by vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$. There are some variants of DE that have been proposed. The variants are denoted as DE/*base*/*num*/*cross*, where "*base*" denotes the manner of constructing the mutant vector, "*num*" denotes the number of difference vectors, and "*cross*" indicates crossover

Algorithm 2 DE/rand/1/-

/*Initialize a population*/
$P = NP$ individuals $\{\boldsymbol{x}_i\}$ generated randomly in S;
Set scaling factor F and Crossover rate CR;
while $FE < FE_{\max}$ **do**
  for $i = 1$ **to** NP **do**
    /*DE operation*/
    $(\boldsymbol{x}_{r_1}, \boldsymbol{x}_{r_2}, \boldsymbol{x}_{r_3})$ = randomly selected from P s.t. $r_1 \neq r_2 \neq r_3 \neq i$;
    $\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3})$;
    $\boldsymbol{u}_i$ = trial vector generated from $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$ by a crossover;
  end for
  for $i = 1$ **to** NP **do**
    if $f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i)$ **then**
      $\boldsymbol{x}_i^{new} = \boldsymbol{u}_i$;
    else
      $\boldsymbol{x}_i^{new} = \boldsymbol{x}_i$;
    end if
  end for
  $P = \{\boldsymbol{x}_i^{new}, i = 1, 2, \ldots, NP\}$;
end while

method. The pseudo-code of DE/rand/1/- is presented in Algorithm 2, where FE is the current number of function evaluations and $FE_{\max}$ is the maximum number of evaluations.

In the initialization phase, NP individuals $P = \{\boldsymbol{x}_i, i = 1, 2, \ldots, NP\}$ are randomly generated in a given search space. Each individual contains D genes as decision variables. At each generation, DE creates a mutant vector $\boldsymbol{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ for each individual $\boldsymbol{x}_i$ (called a target vector) in the current population. In case of DE/rand/1 strategy, a mutant vector $\boldsymbol{v}_i$ is generated as follows:

$$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}) \tag{5}$$

The indices r_1, r_2 and r_3 are distinct integers uniformly chosen from the set $\{1, 2, \ldots, NP\} \backslash \{i\}$. The parameter F is called the scaling factor, which amplifies the difference vectors.

After mutation, DE performs the crossover operator between target vector and mutant vector, and generates a trial vector $\boldsymbol{u}_i = (u_{i1}, u_{i2}, \ldots, u_{iD})$. In this paper, we use shuffled exponential crossover, which does not rely on arbitrary dependencies between adjacent variables [12]. In the crossover, CR is the crossover rate within the range $[0, 1)$ and presents the probability of generating genes for a trial vector $\boldsymbol{u_i}$ from a mutant vector $\boldsymbol{v_i}$. If the jth element u_{ij} of the trial vector $\boldsymbol{u}_i$ is infeasible (i.e., out of the boundary $[L_j, U_j]$), it is reset as follows:

$$u_{ij} = \begin{cases} 2L_j - x_{ij} & (u_j < L_j) \\ 2U_j - x_{ij} & (u_j > U_j) \end{cases} \tag{6}$$

After all of the trial vectors have been generated, the selection operator is performed to select a better one from the target vector $\boldsymbol{x}_i$ and its corresponding trial vector $\boldsymbol{u}_i$ according to their fitness values $f(\cdot)$. The selected vector is given by

$$\boldsymbol{x}_i^{new} = \begin{cases} \boldsymbol{u}_i & \text{if } f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i) \\ \boldsymbol{x}_i & \text{otherwise} \end{cases} \tag{7}$$

and $\boldsymbol{x}_i^{new}$ is used as a target vector in the next generation. The algorithm is terminated when the maximum number of function evaluations is reached.

4. **DE with DIRECT Algorithm.** In this section we propose a combination of DE and DIRECT to realize efficient and robust search in expensive optimization problems. In the proposed method, called DE-DIRECT, firstly a search by DIRECT is performed using a part of the given function evaluation budgets. The search of DIRECT is terminated when the number of function evaluations reaches FE_{direct}. After DIRECT, the remaining search is performed by DE until the number of function evaluations reaches $FE_{\max}$. FE_{direct} is given as follows:

$$FE_{direct} = FE_{\max} \times r_{fe} \tag{8}$$

where r_{fe} is a parameter for controlling function evaluation budgets of DIRECT.

After the search of DIRECT is finished, solution set P_{direct} is extracted from the set of hyper-rectangles H which are obtained by DIRECT. In the initialization phase, DE obtains NP_{direct} individuals from DIRECT, and the remaining $(NP - NP_{direct})$ individuals are generated randomly. NP_{direct} is given as follows:

$$NP_{direct} = NP \times r_{np} \tag{9}$$

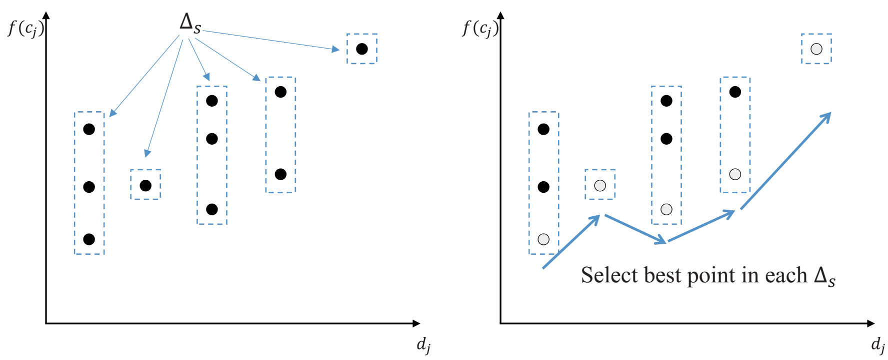
where r_{np} is the proportion of individuals that are selected from DIRECT.

To improve the convergence speed of DE, it is desirable to provide good solutions from DIRECT to DE. However, diversity of the initial population may be decreased if the extracted search points are concentrated in a specific area of solution space. Therefore, we introduce an extraction method considering diversity as well as accuracy of solution.

Algorithm 3 is the procedure of selection method of initial individuals from DIRECT. Firstly, the hyper-rectangles H are sorted in groups according to the size d as shown in Figure 3(a). Let Δ_s be the set of all hyper-rectangles of the same size d, for $s = 1, \ldots, S$.

Algorithm 3 Selection procedure of initial individuals from DIRECT

Set index $s = 1$;
Set $P_{direct} \leftarrow \emptyset$;
while $(|P_{direct}| < NP_{direct})$ **do**
  if $\Delta_s \neq 0$ **then**
    Choose best point $best_p \in \Delta_s$;
    Set $P_{direct} \leftarrow P_{direct} \cup best_s$;
    Set $\Delta_s \leftarrow \Delta_s \setminus \{best_s\}$;
  end if
  $s = (s+1)\%S$;
end while



(a) Grouping (b) Extraction

FIGURE 3. Grouping and extraction of hyper-rectangle set H

Next, a phase of extracting search points is performed as shown in Figure 3(b). Here, index s is set to 1 and the best search point $best_1$ in the smallest Δ_1 is added to P_{direct}. After that, while increasing s, the best points from each Δ_s are added to P_{direct}. If s exceeds S, s resets to 1. Also, individuals to be selected do not allow duplication. By considering not only the function value but also the size of hyper-rectangle, it is possible to select the good search points evenly from the whole solution domain searched by DIRECT.

5. Experiment.

5.1. Setup.

In this section, we investigate the performance of DE-DIRECT using the benchmark functions as shown in Table 1, where dimension $D = 100$. In summary, functions F_1-F_4 are unimodal and functions F_5-F_8 are multimodal. All functions have an optimal value 0. For functions where the optimal solution is located at the origin, the position of its optimal solution is shifted by random vector $\boldsymbol{o}$. The maximum number of function evaluations $FE_{\max}$ is set to 2×10^5. In DE and DE-DIRECT, independent 10 runs are performed for each function. Since DIRECT is a deterministic optimization algorithm, only one trial was performed. Each algorithm stops when the number of function evaluations exceeds the $FE_{\max}$.

Table 1. Benchmark functions

Name	Expression	Domain
F_1: Sphere	$f(\boldsymbol{z}) = \sum_{i=1}^{D} z_i^2,\ \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o}$	$[-5.12, 5.12]^D$
F_2: Ridge	$f(\boldsymbol{z}) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} z_j \right)^2,\ \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o}$	$[-100, 100]^D$
F_3: Rosenbrock(chain)	$f(\boldsymbol{x}) = \sum_{i=1}^{D-1} \left\{ 100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right\}$	$[-2.048, 2.048]^D$
F_4: Rosenbrock(star)	$f(\boldsymbol{x}) = \sum_{i=2}^{D} \left\{ 100 \left(x_1 - x_i^2 \right)^2 + (x_i - 1)^2 \right\}$	$[-2.048, 2.048]^D$
F_5: Griewank	$f(\boldsymbol{z}) = 1 + \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \left(\cos \left(\frac{z_i}{\sqrt{i}} \right) \right),\ \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o}$	$[-512, 512]^D$
F_6: Schaffer	$f(\boldsymbol{z}) = \sum_{i=1}^{D-1} \left(z_i^2 + z_{i+1}^2 \right)^{0.25} \times \left\{ \sin^2 \left(50 \left(z_i^2 + z_{i+1}^2 \right)^{0.1} \right) + 1 \right\},$ $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o}$	$[-100, 100]^D$
F_7: Schwefel	$f(\boldsymbol{x}) = 418.9829D - \sum_{i=1}^{D} x_i \sin \left(\sqrt{\lvert x_i \rvert} \right)$	$[-500, 500]^D$
F_8: Rastrigin	$f(\boldsymbol{z}) = 10D + \sum_{i=1}^{D} z_i^2 - 10\cos(2\pi z_i),\ \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o}$	$[-5.12, 5.12]^D$

The strategy of DE is DE/rand/1/exp and the parameters settings for DE are as follows – the population size $NP = 200$, $F = 0.5$, $CR = 0.95$. In DIRECT, the balance parameter ϵ is set to 10^{-6}. The performance of the DE-DIRECT has been evaluated for $r_{np} \in \{0.5, 0.7\}$, $r_{fe} \in \{0.1, 0.3, 0.5\}$.

5.2. Results.

Table 2 shows the results (mean function value and standard deviation) of DE, DIRECT, and DE-DIRECT. For each function, mean is shown in the top row and standard deviation is shown in the bottom row. For Sphere (F_1) and Griewank (F_5) functions, DIRECT obtains better results than other methods. In the DIRECT method,

Table 2. Experimental results on DE, DIECT, and DE-DIRECT. The top row is mean function value and the bottom row is standard deviation.

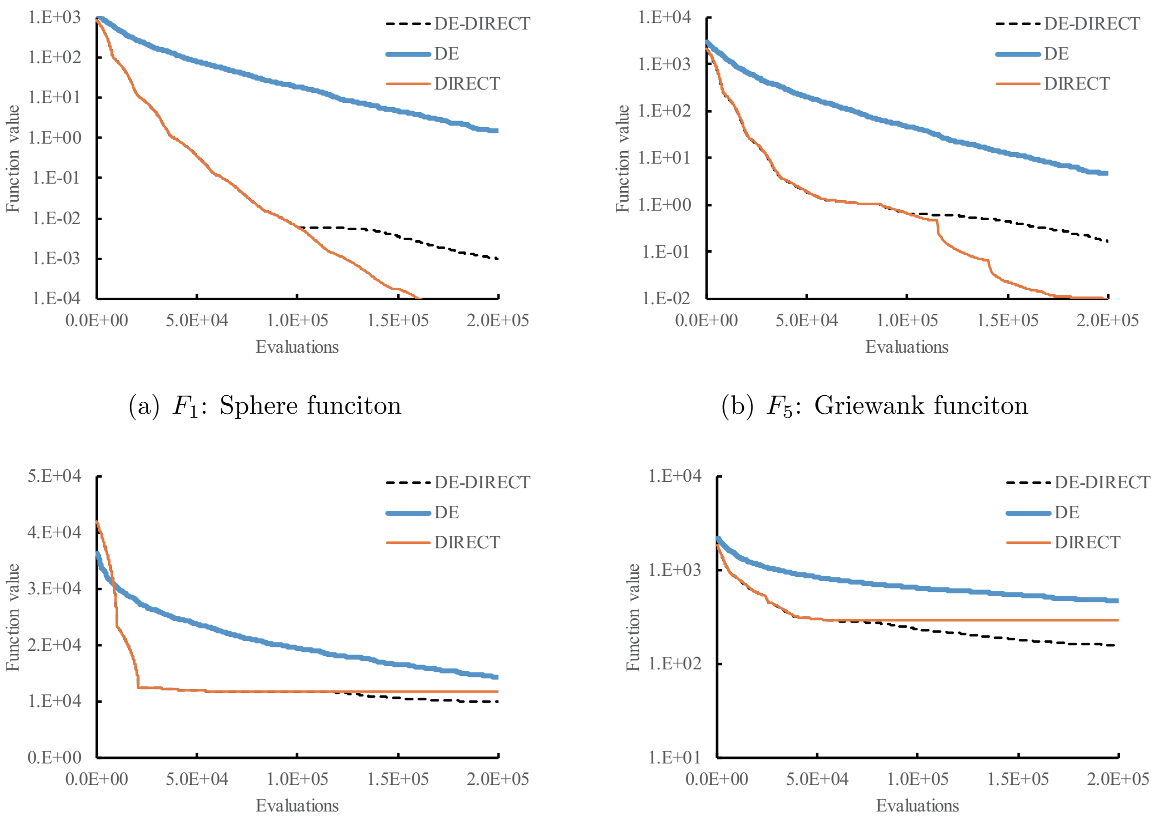
Function	DE	DIRECT	DE-DIRECT (r_{np}, r_{fe})					
			$(0.5, 0.1)$	$(0.5, 0.3)$	$(0.5, 0.5)$	$(0.7, 0.1)$	$(0.7, 0.3)$	$(0.7, 0.5)$
F_1	1.47E+00	**1.08E−05**	1.60E−01	1.02E−02	1.88E−03	1.83E−01	6.68E−03	9.99E−04
	1.19E−01		1.65E−02	3.93E−04	2.82E−04	8.57E−02	1.08E−03	7.89E−05
F_2	1.30E+05	1.65E+04	2.43E+04	1.77E+04	1.55E+04	2.31E+04	1.64E+04	**1.45E+04**
	8.11E+03		5.83E+02	5.54E+02	4.04E+02	8.19E+02	4.97E+02	**4.92E+02**
F_3	9.76E+02	9.68E+01	9.47E+01	9.54E+01	9.58E+01	**9.41E+01**	9.51E+01	9.56E+01
	7.59E+01		1.34E−01	2.04E−01	1.02E−01	**1.62E−01**	2.71E−01	1.33E−01
F_4	1.02E+03	6.76E+01	5.52E+01	5.27E+01	5.62E+01	7.57E+01	**5.16E+01**	5.47E+01
	8.77E+01		3.73E−01	2.02E−01	2.59E−01	5.64E+00	**2.51E−01**	4.09E−01
F_5	4.68E+00	**1.01E−02**	1.40E+00	8.64E−01	3.09E−01	1.46E+00	6.45E−01	1.71E−01
	2.96E−01		4.13E−02	2.15E−02	3.10E−02	2.14E−01	4.75E−02	3.02E−02
F_6	3.95E+02	2.14E+02	2.13E+02	**1.51E+02**	1.62E+02	2.14E+02	1.59E+02	1.69E+02
	6.88E+00		4.82E+00	**4.28E+00**	6.40E+00	1.13E+01	1.09E+01	6.89E+00
F_7	1.42E+04	1.18E+04	9.76E+03	1.00E+04	**9.90E+03**	1.03E+04	1.13E+04	1.08E+04
	4.28E+02		1.53E+02	6.24E+02	3.47E+02	4.39E+02	2.49E+02	2.82E+02
F_8	4.72E+02	2.89E+02	2.66E+02	**1.57E+02**	1.68E+02	2.55E+02	1.65E+02	1.80E+02
	1.50E+01		1.37E+01	**4.29E+00**	5.42E+00	1.00E+01	6.84E+00	1.00E+01

when applied to a high-dimensional problem, an area to be divided is enlarged, making exhaustive search impossible. Therefore, the DIRECT method is not used much for the high dimensional problems [11]. However, if the objective function is simple unimodal landscape like the Sphere function, DIRECT can easily identify promising areas and perform effective search even in higher dimensions. Also, Griewank function is multimodal landscape but it is not so rugged. Due to this, it is considered that DIRECT showed good results.

For other functions which have dependencies between variables (F_2, F_3, and F_4) and multimodal landscape (F_6, F_7, and F_8), DE-DIRECT shows superior performance in almost all combinations of r_{fe} and r_{np}. However, the combination of the best parameters differs according to the objective function. Also, as the number of individuals selected from DIRECT is larger, the search efficiency of DE-DIRECT is not always improved. Except for F_2, F_3, and F_4, when the proportion of DIRECT individuals is large, population diversity decreases and the search efficiency of DE stagnates.

Next, Figure 4 shows the convergence graphs of function F_1, F_5, F_7, and F_8 where the mean best objective values of DE, DIRECT, DE-DIRECT are plotted over the number of function evaluations. Here, the plot of DE-DIRECT is the best parameter combination. As can be seen from these figures, DIRECT shows faster convergence than DE in F_1 and F_5. However, for F_7 and F_8 which are strong multimodal landscape, the objective value of DIRECT stagnates at the beginning of the search. On the other hand, in DE-DIRECT, population diversity can be preserved by combining the solutions obtained by DIRECT with random individuals, and consequently it is able to eventually find the good solution. From the above results, we conclude that DE-DIRECT has the advantages of both high search efficiency of DIRECT and robustness of DE, and consequently realize more effective search than DE.

6. **Conclusion.** The selection of the initial population in DE is very important, since it affects the search efficiency and the goodness of the obtained solution. In this study, we focused on the method of generating initial individuals of DE. In order to provide a good initial individual to DE, we proposed a search method combining DE and DIRECT. In



(a) F_1: Sphere funciton

(b) F_5: Griewank funciton

(c) F_7: Schwefel function

(d) F_8: Rastrigin function

FIGURE 4. Convergence graphs

the proposed method, called DE-DIRECT, a search by DIRECT is performed first using a part of the given function evaluation budgets. Next, some individuals are selected from the search points obtained by DIRECT as a part of initial population of DE. In order to extract effective individuals from the solution set obtained by DIRECT, we introduced a selection method considering diversity as well as accuracy of solution. To evaluate the performance of the proposed DE-DIRECT, we conducted experiments using large-scale benchmark functions. From the experimental results, we confirmed that DE-DIRECT showed faster convergence than DE and realized efficient search in multimodal functions where DIRECT stagnates in the early stage of search. However, it was found that the setting of appropriate parameters in DE-DIRECT is dependent on the function landscape. In the future, we intend to conduct more detailed sensitivity analysis about r_{fe} and r_{np} on various expensive optimization problems [13]. Furthermore, we will combine the DE-DIRECT with landscape detection technique [14] and aim to develop dynamic parameter adaptation of r_{fe} and r_{np}.

REFERENCES

[1] J. H. Holland, Adaptation in natural and artificial systems: An introductory analysis with applications to biology, *Control, and Artificial Intelligence*, MIT Press, 1992.

[2] J. Kennedy, Particle swarm optimization, *Encyclopedia of Machine Learning*, Springer US, pp.760-766, 2011.

[3] R. Storn and K. V. Price, Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol.11, no.4, pp.341-359, 1997.

[4] I. Zelinka, A survey on evolutionary algorithms dynamics and its complexity – Mutual relations, past, present and future, *Swarm and Evolutionary Computation*, vol.25, pp.2-14, 2015.
[5] S. Das and P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Trans. Evolutionary Computation*, vol.15, no.1, pp.4-31, 2011.
[6] S. Das, S. S. Mullick and P. N. Suganthan, Recent advances in differential evolution – An updated survey, *Swarm and Evolutionary Computation*, vol.27, pp.1-30, 2016.
[7] B. Kazimipour, X. Li and A. K. Qin, A review of population initialization techniques for evolutionary algorithms, *2014 IEEE Congress on Evolutionary Computation (CEC)*, pp.2585-2592, 2014.
[8] H. Maaranen, K. Miettinen and M. M. Mäkelä, Quasi-random initial population for genetic algorithms, *Computers and Mathematics with Applications*, vol.47, no.12, pp.1885-1895, 2004.
[9] D. Jones, C. Perttunen and B. Stuckman, Lipschitzian optimization without the Lipschitz constant, *Journal of Optimization Theory and Application*, vol.79, no.1, pp.157-181, 1993.
[10] J. M. Gablonsky and C. T. Kelley, A locally-biased form of the direct algorithm, *Journal of Global Optimization*, vol.21, no.1, pp.27-37, 2001.
[11] T. Kaihatsu and S. Watanabe, A proposal of a low-dimensional approach based on DIRECT method and t-SNE for single optimization problems with many variables, *Joint the 17th World Congress of International Fuzzy Systems Association and the 9th International Conference on Soft Computing and Intelligent Systems*, pp.1-6, 2017.
[12] R. Tanabe and F. Alex, Reevaluating exponential crossover in differential evolution, *Parallel Problem Solving from Nature – PPSN XIII*, pp.201-210, 2014.
[13] J. J. Liang, B. Y. Qu and P. N. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, *Technical Report*, Zhengzhou University and Nanyang Technological University, 2013.
[14] T. Takahama and S. Sakai, Improving an adaptive differential evolution using hill-valley detection, *International Journal of Hybrid Intelligent Systems*, vol.13, no.3, pp.1-13, 2016.