



Additional Out-Group Search for JADE

Yuichi Miyahira^(✉) and Akira Notsu

Osaka Prefecture University, 1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
miyahira@hi.cs.osakafu-u.ac.jp

Abstract. JADE is a method to adaptively select parameters using probability distribution, and shows good searching accuracy and speed. However, the search outside the solution group is not considered as well as other methods. Though it is possible to forcibly increase the search outside the solution group by dividing the solution group or adding random search, the search speed lowers. In this study, referring to the Nelder-Mead method, the algorithm for improving the accuracy without reducing the search speed as much as possible by adding the solution group outside search to JADE is proposed. Concretely, when a fixed condition is satisfied, one point of the search point carries out the group outside search. In order to prevent the search speed from decreasing even in the case of a high dimension, the number of search points for out-of-group search is set not to increase even when the dimension increases. The effectiveness of the proposed method is confirmed by numerical experiments.

Keywords: Differential evolution · JADE · Nelder-Mead

1 Introduction

Difference Evolution [1] is an optimizing algorithm without gradients that has good search performance and is applicable to a variety of applications. DE is a relatively simple algorithm with the advantage that it has only three control parameters: mutation coefficient F , crossover rate CR , and population size N . However, due to the small number of parameters, the accuracy of the search is affected by the parameters. Therefore, it is necessary to set the appropriate parameter value according to the problem to be handled and the search situation. To solve this problem, SaDE [2], jDE [3], SHADE [4], JADE [5, 6] have been proposed.

JADE is a method for efficiently adapting the parameters of DE to the environment by sampling them from a Cauchy or Gaussian distribution adapted to the environment. Specifically, the mean value in the probability distribution used to generate the mutation coefficient F and the crossover rate CR is adjusted according to the success value of each parameter, allowing each parameter to

Supported by JSPS KAKENHI Grant Number JP18K11473.

© Springer Nature Switzerland AG 2022
K. Honda et al. (Eds.): IUKM 2022, LNAI 13199, pp. 105–116, 2022.
https://doi.org/10.1007/978-3-030-98018-4_9

be adjusted automatically. This allows the appropriate parameters to be used, increasing the accuracy of the search.

As with other methods, out-group search is not considered much in JADE, and performance degrades if a solution is missed outside the group after the search has progressed. In our basic experiment [7], we confirmed that searching can be improved by adding the probability of selecting $F = 1.5$, which is an enhanced parameter of out-group search that is not normally set in DE. Therefore, we believe that the performance can be further improved by adding out-group search to JADE.

In this study, referring to Nelder-Mead method [8], we propose an algorithm to add out-group search to JADE and to improve the accuracy without decreasing the search speed as much as possible. If the distance of the solution with the best and worst solutions is more than half of the distance from the end of the search point group, it is judged that there is a large bias in the direction of the update of the search point group, and the outside of the solution group is searched. The composition of this paper is as follows. In Sect. 2, JADE and Nelder-Mead method which are handled in this study are explained. In Sect. 3, we propose and explain JADE with the addition of out-group search. In Sect. 4, the results of numerical experiments using the proposed method are discussed, and in Sect. 5, future problems are summarized.

2 Base Algorithm

2.1 JADE

JADE is one of the improved methods of DE, and the mutation coefficient F and the crossover rate CR are sampled from the probability distribution and used. The Cauchy distribution is used for the mutation coefficient, and the Gaussian distribution is used for the crossover rate, and the adjustment of each distribution is carried out based on the proportion in which the solution was renewed in the past. In JADE, parameters are generated for each individual. As a mutation strategy, “DE/current-to- p best” is used. The generation method of each parameter is shown below.

For each generation, the mutation factor F_i of each individual x_i is generated according to the Cauchy distribution of the positional parameter μ_F and the scaling parameter $\sigma_F = 0.1$ as follows:

$$F_i \sim C(\mu_F, \sigma_F)$$

F_i is regenerated if $F_i \leq 0$, and truncated to 1 if $F_i \geq 1$. The positional parameter μ_F is initialized at 0.5 and updated for each generation as follows:

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}$$

c is a constant for $(0, 1]$ and the recommended value is 0.1. S_F is the set of mutation coefficients that have successfully updated the solution in that generation.

Similarly, for each generation, the crossover rate CR_i of each individual x_i is generated according to a Gaussian distribution with a mean of μ_{CR} and a standard deviation of $\sigma_{CR} = 0.1$ as follows:

$$CR_i \sim N(\mu_{CR}, \sigma_{CR}^2)$$

CR_i is truncated to the interval $[0, 1]$. The average μ_{CR} is initialized at 0.5 and updated for each generation as follows:

$$CR_i \sim N(\mu_{CR}, \sigma_{CR}^2)$$

S_N is the number of times the solution is updated successfully in each generation, and S_{CR} is the set of crossover rates CR when the solution is updated successfully.

The following describes the mutation strategy “DE/current-to- p best” used in JADE. In this strategy, the mutation vector $\mathbf{v}_{i,g}$ for each individual $x_{i,g}$ of each generation g is generated by:

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_i \cdot (\mathbf{x}_{best,g}^p - \mathbf{x}_{i,g}) + F_i \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g})$$

F_i is the mutation coefficient of each individual x_i , and $\mathbf{x}_{best,g}^p$ is an individual selected from the top $100p\%$ individuals. Also, $\mathbf{x}_{r1,g}$ and $\mathbf{x}_{r2,g}$ are two points randomly selected from the search points other than x_i so that they do not overlap. Here p is a constant in $(0, 1)$, and the recommended value is 0.1. In addition, there is a way to use an archive in this strategy. In this case, the mutation vector $\mathbf{v}_{i,g}$ is generated by the following equation:

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_i \cdot (\mathbf{x}_{best,g}^p - \mathbf{x}_{i,g}) + F_i \cdot (\mathbf{x}_{r1,g} - \tilde{\mathbf{x}}_{r2,g})$$

$\tilde{\mathbf{x}}_{r2,g}$ is an individual selected at random from the aggregate of past failed individuals stored in the archive and the aggregate of current solution populations. The archive is initially empty and adds failed individuals at the end of each generation update.

The next search point is generated by crossing over this mutation vector with the individual as follows based on the crossover rate:

$$x_{i,g+1,j} = \begin{cases} v_{i,g,j} (w \leq CR) \\ x_{i,g,j} (otherwise) \end{cases}$$

w is uniform random number between 0 and 1, and $x_{i,g,j}$ means the j -th element of the i -th individual in generation g .

2.2 Nelder-Mead Method

The Nelder-Mead method is a kind of optimizing algorithm without using gradient information. By giving $D + 1$ search points in D dimension space and repeating reflection, expansion and contraction for them, the optimum solution can be searched. The algorithm of function minimization in the Nelder-Mead method is shown below.

Step 0. For $D + 1$ search points, $f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \dots \leq f(\mathbf{x}_{D+1})$ is set in the order of the values of the objective function f , and if the end condition is not satisfied, **Step1** is assumed, and if the end condition is satisfied, \mathbf{x}_1 is assumed as the solution.

Step 1. Using the centroids \mathbf{x}_c of $\mathbf{x}_1 \dots \mathbf{x}_n$, the reflection point \mathbf{x}_{ref} of \mathbf{x}_{D+1} is determined by the following equation:

$$\mathbf{x}_{ref} = \mathbf{x}_c + \alpha(\mathbf{x}_c - \mathbf{x}_{D+1})$$

Step 2.

case 1 If $f(\mathbf{x}_1) \leq f(\mathbf{x}_{ref}) < f(\mathbf{x}_D)$, replace \mathbf{x}_{D+1} with \mathbf{x}_{ref} and go to **Step0**.

case 2 If $f(\mathbf{x}_{ref}) < f(\mathbf{x}_1)$, the expansion point, which is the point where the reflection point is further extended, is obtained as follows:

$$\mathbf{x}_{exp} = \mathbf{x}_c + \gamma(\mathbf{x}_{ref} - \mathbf{x}_c)$$

If $f(\mathbf{x}_{exp}) \leq f(\mathbf{x}_{ref})$, replace \mathbf{x}_{D+1} with \mathbf{x}_{exp} and go to **Step0**, otherwise replace \mathbf{x}_{D+1} with \mathbf{x}_{ref} and go to **Step0**.

case 3 If $f(\mathbf{x}_D) \leq f(\mathbf{x}_{ref})$

case 3-1 If $f(\mathbf{x}_{ref}) < f(\mathbf{x}_{D+1})$, the contraction point is obtained as follows:

$$\mathbf{x}_{con} = \mathbf{x}_c + \beta(\mathbf{x}_{ref} - \mathbf{x}_c)$$

Then go to **Step3**.

case 3-2 Otherwise, the contraction point is obtained as follows:

$$\mathbf{x}_{con} = \mathbf{x}_c + \beta(\mathbf{x}_{D+1} - \mathbf{x}_c)$$

Then go to **Step3**.

Step 3. If $f(\mathbf{x}_{con}) < \min\{f(\mathbf{x}_{ref}), f(\mathbf{x}_{D+1})\}$, replace \mathbf{x}_{D+1} with \mathbf{x}_{con} and go to **Step0**, otherwise go to **Step4**.

Step 4. Shrink all individual i to point \mathbf{x}_1 as follows:

$$\mathbf{x}_i = \mathbf{x}_1 + \delta(\mathbf{x}_i - \mathbf{x}_1)$$

Then go to **Step0**.

3 Proposed Method

In the proposed method, one of the search points of JADE is made to search outside the group according to the situation. By this, the search is made so that the solution is not missed, when the function in which there are multiple local solutions is searched. In addition, escape from the local solution by the search outside the group is expected, when it falls into the local solution. And, the lowering of the search speed by the increase of the outside group search is prevented by limiting the point of the outside group search to one point.

Concretely, if the distance between the search point with the best solution and the search point with the worst solution is more than half of the distance from the end to the end of the search point group, one search point searches the point by extending a 2^n vector from the search point with the worst solution to the search point with the best solution. The value of n is determined by sampling from a geometric distribution. The success probability of the geometric distribution p is defined as $1/G^{p_{update}}$ by the update rate of the solution p_{update} and the generation G . This p_{update} is updated in the same way as JADE with an initial value of 0 (Fig. 1).

$$\begin{aligned} &\text{if } \|\mathbf{x}_{\text{best}} - \mathbf{x}_{\text{worst}}\| > \|\mathbf{x}_{\text{max}} - \mathbf{x}_{\text{min}}\|/2 \\ &n = \text{GeometricDistribution}(1/G^{p_{update}}) \\ &\mathbf{x}_{\text{last}} = \mathbf{x}_{\text{worst}} + (\mathbf{x}_{\text{best}} - \mathbf{x}_{\text{worst}}) * 2^n \end{aligned}$$

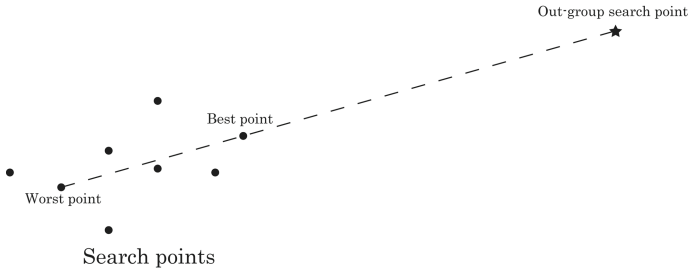


















Fig. 1. Out-group search

4 Numerical Experiments and Results

JADE and the proposed method (JADE+) are compared for 16 benchmark functions shown in Table 1. In the comparison experiment, calculation up to 1000 generations was tried 1000 times for each function at a search point of 10. Figure 2 through 24 show the average of 1000 times of the best solution for each generation, with the evaluation value on the vertical axis and the generation number on the horizontal axis. Figures 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17 show the results of the experiment with the dimension of 2, and Figs. 18, 19, 20, 21, 22, 23 and 24 show the results of the experiment with the dimension of 10. In order to deal with the minimization problem, we show that the lower the graph is, the better the solution is. Tables 2 and 3 summarize the mean \pm standard deviation of the final solutions for each function. Table 2 shows the case where the dimension is 2, and Table 3 shows the case where the dimension is 10. In Tables 2 and 3, the methods which obtained good solutions and obtained solutions are shown in bold type.

Table 1. Test functions used in the experiment

Function	Equation · Domain · Optimum solution	Shape (D=2)
F1: Sphere	$f(x_1, \dots, x_D) = \sum_{j=1}^D x_j^2$ $-1 \leq x_j \leq 1$ $f_{\min}(0, \dots, 0) = 0$	
F2: Rosenbrock	$f(x_1, \dots, x_D) = \sum_{j=1}^{D-1} (100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2)$ $-5 \leq x_j \leq 5$ $f_{\min}(1, \dots, 1) = 0$	
F3: Booth	$f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$ $-10 \leq x_j \leq 10$ $f_{\min}(1, -3) = 0$	
F4: Matyas	$f(x_1, x_2) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$ $-10 \leq x_j \leq 10$ $f_{\min}(0, 0) = 0$	
F5: Easom	$f(x_1, x_2) = -\cos(x_1) \cos(x_2) \exp(-((x_1 - \pi)^2 + (x_2 - \pi)^2))$ $-100 \leq x_j \leq 100$ $f_{\min}(\pi, \pi) = -1$	
F6: Rastrigin	$f(x_1, \dots, x_D) = 10D + \sum_{j=1}^D (x_j^2 - 10 \cos(2\pi x_j))$ $-5 \leq x_j \leq 5$ $f_{\min}(0, \dots, 0) = 0$	
F7: Ackley	$f(x_1, \dots, x_D) = 20 - 20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{j=1}^D x_j^2}\right)$ $+ e - \exp\left(\frac{1}{D} \sum_{j=1}^D \cos(2\pi x_j)\right)$ $-32.768 \leq x_j \leq 32.768$ $f_{\min}(0, \dots, 0) = 0$	
F8: Levi N.13	$f(x_1, x_2) = \sin^2(3\pi x_1) + (x_1 - 1)^2 (1 + \sin^2(3\pi x_2))$ $+ (x_2 - 1)^2 (1 + \sin^2(2\pi x_2))$ $-10 \leq x_j \leq 10$ $f_{\min}(1, 1) = 0$	
F9: Bukin N.6	$f(x_1, x_2) = 100 \sqrt{ x_2 - 0.01x_1^2 } + 0.01 x_1 + 10 $ $-15 \leq x_1 \leq 5$ $-3 \leq x_2 \leq 3$ $f_{\min}(-10, 1) = 0$	
F10: Beale	$f(x_1, x_2) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2$ $+ (2.625 - x_1 + x_1x_2^3)^2$ $-4.5 \leq x_j \leq 4.5$ $f_{\min}(3, 0.5) = 0$	
F11: Goldstein-Price	$f(x_1, x_2) = (1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2))(30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$ $-2 \leq x_1, x_2 \leq 2$ $f_{\min}(0, -1) = 3$	
F12: Schaffer N.2	$f(x_1, x_2) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$ $-100 \leq x_1, x_2 \leq 100$ $f_{\min}(0, 0) = 0$	
F13: Five-well potential	$f(x_1, x_2) = \left(1 - \frac{1}{1 + 0.05(x_1^2 + (x_2 - 10)^2)} - \frac{1}{1 + 0.05((x_1 - 10)^2 + x_2^2)} - \frac{1}{1 + 0.03((x_1 + 10)^2 + x_2^2)} - \frac{1}{1 + 0.05((x_1 - 5)^2 + (x_2 + 10)^2)}\right) (1 + 0.0001(x_1^2 + x_2^2)^{1.2})$ $-20 \leq x_1 \leq 20$ $-20 \leq x_2 \leq 20$ $f_{\min}(4.92, -9 : 89) = -1.4616$	
F14: Griewank	$f(x_1, \dots, x_D) = 1 + \frac{1}{4000} \sum_{j=1}^D x_j^2 - \prod_{j=1}^D \cos\left(\frac{x_j}{\sqrt{j}}\right)$ $-600 \leq x_j \leq 600$ $f_{\min}(0, \dots, 0) = 0$	
F15: Xin-She Yang	$f(x_1, \dots, x_D) = \left(\sum_{j=1}^D x_j \right) \exp\left(-\sum_{j=1}^D \sin(x_j^2)\right)$ $-2\pi \leq x_j \leq 2\pi$ $f_{\min}(0, \dots, 0) = 0$	
F16: Styblinski-Tang	$f(x_1, \dots, x_D) = \frac{\sum_{j=1}^D x_j^4 - 16x_j^2 + 5x_j}{2}$ $-5 \leq x_1 \leq 4$ $-3 \leq x_2 \leq 4$ $-39.16617D \leq f(-2.903534, \dots, -2.903534) \leq -39.16616D$ $f_{\min}(-2.903534, \dots, -2.903534) \approx -39.166165D$	

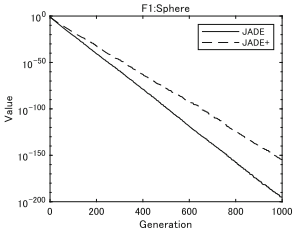


Fig. 2. F1 (D = 2)

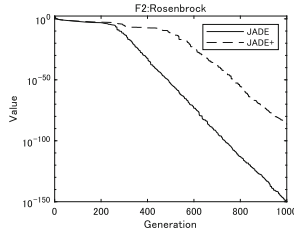


Fig. 3. F2 (D = 2)

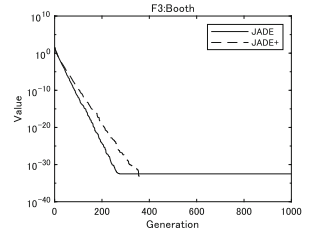


Fig. 4. F3 (D = 2)

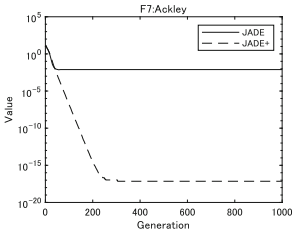


Fig. 8. F7 (D = 2)

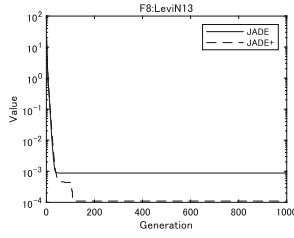


Fig. 9. F8 (D = 2)

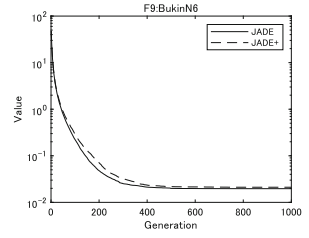


Fig. 10. F9 (D = 2)

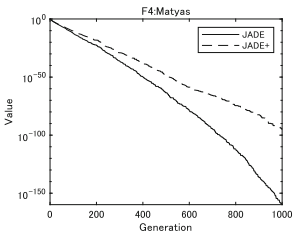


Fig. 5. F4 (D = 2)

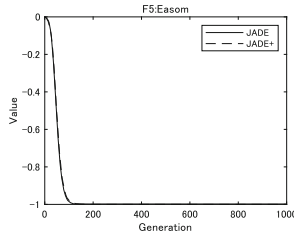


Fig. 6. F5 (D = 2)

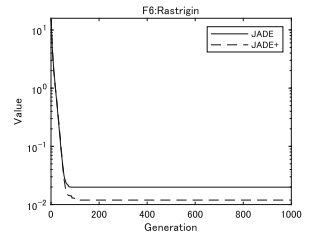


Fig. 7. F6 (D = 2)

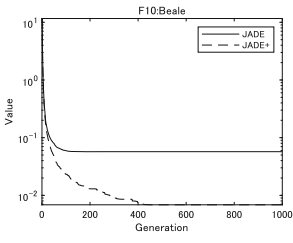


Fig. 11. F10 ($D = 2$)

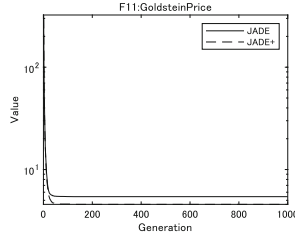


Fig. 12. F11 ($D = 2$)

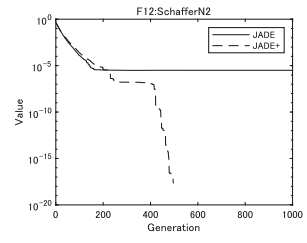


Fig. 13. F12 ($D = 2$)

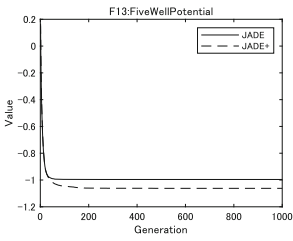


Fig. 14. F13 ($D = 2$)

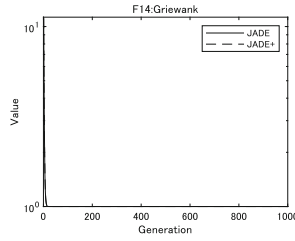


Fig. 15. F14 ($D = 2$)

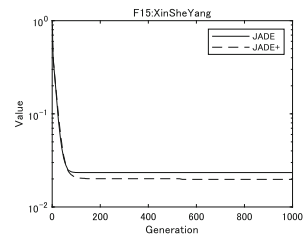


Fig. 16. F15 ($D = 2$)

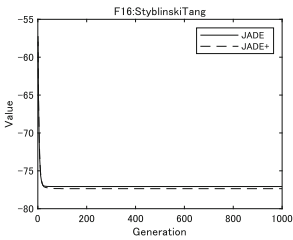


Fig. 17. F16 ($D = 2$)

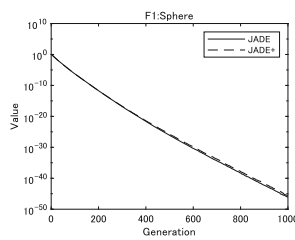


Fig. 18. F1 ($D = 10$)

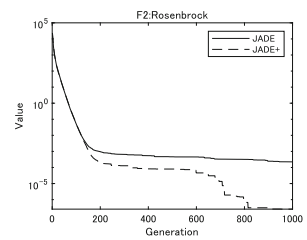


Fig. 19. F2 ($D = 10$)

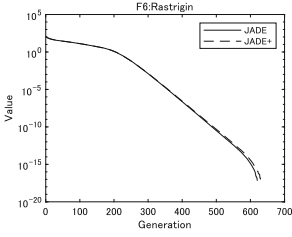


Fig. 20. F6 ($D = 10$)

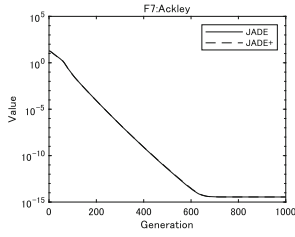


Fig. 21. F7 ($D = 10$)

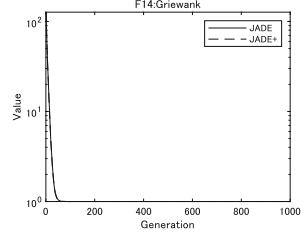


Fig. 22. F14 ($D = 10$)

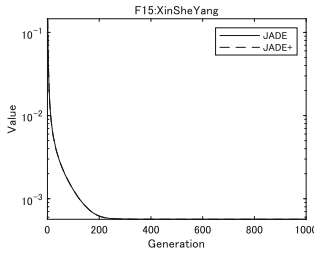


Fig. 23. F15 ($D = 10$)

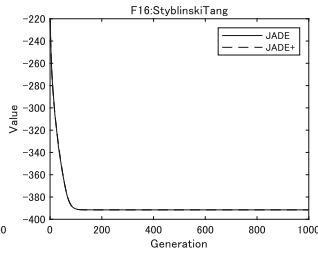


Fig. 24. F16 ($D = 10$)

We begin with a discussion of the two-dimensional case, Figs. 2 through 17. The graphs show that the proposed method has a better solution than JADE in Fig. 7 (Rastrigin function), Fig. 8 (Ackley function), Fig. 9 (Levi N. 13 function), Fig. 11 (Beale function), Fig. 12 (Goldstein-Price function), Fig. 13 (SchafferN2 function), Fig. 14 (Five-well potential function), Fig. 16 (Xin-She Yang function), and Fig. 17 (Styblinski-Tang function), and the convergence speed is almost the same. From this fact, it is proven that the accuracy heightens without lowering the search speed by carrying out the group outside search in one search point. In Fig. 9 (Levi N. 13 function) and Fig. 13 (SchafferN2 function), it can be seen that the proposed method escaped from the point where it almost converged once, and reached a better solution. This suggests that the search outside the group works well to escape from the local solution. However, in Fig. 2 (Sphere function), Fig. 3 (Rosenbrock function) and Fig. 5 (Matyas function), the search speed of the solution is inferior to JADE, and it can be read that the useless group outside search leads to the lowering of the search speed in the simple unimodal function.

We then discuss the 10 dimensional case, Figs. 18 through 24. First, as shown in Fig. 19, the result of F2 was improved. The results of other functions were not

Table 2. Comparison of mean and standard deviation ($D = 2$)

Function	Method	Mean \pm Standard deviation
F1:Sphere	JADE	5.7656e-196 \pm 0
	JADE+	8.7776e-156 \pm 1.9808e-154
F2:Rosenbrock	JADE	1.3679e-150 \pm 4.3256e-149
	JADE+	2.9785e-87 \pm 9.4188e-86
F3:Booth	JADE	3.1554e-33 \pm 9.9784e-32
	JADE+	0 \pm 0
F4:Matyas	JADE	1.3784e-160 \pm 4.0084e-159
	JADE+	7.5967e-96 \pm 2.4023e-94
F5:Easom	JADE	-0.998 \pm 0.044695
	JADE+	-1 \pm 0
F6:Rastrigin	JADE	0.019899 \pm 0.1463
	JADE+	0.01194 \pm 0.10839
F7:Ackley	JADE	0.0077398 \pm 0.14117
	JADE+	7.1054e-18 \pm 1.588e-16
F8:LeviN13	JADE	0.00087899 \pm 0.0097929
	JADE+	0.00010987 \pm 0.0034745
F9:BukinN6	JADE	0.019676 \pm 0.014354
	JADE+	0.021124 \pm 0.026538
F10:Beale	JADE	0.057155 \pm 0.20082
	JADE+	0.0068586 \pm 0.072006
F11:GoldsteinPrice	JADE	5.43 \pm 12.4936
	JADE+	4.566 \pm 10.2735
F12:SchafferN2	JADE	3.1266e-06 \pm 9.8872e-05
	JADE+	0 \pm 0
F13:FiveWellPotential	JADE	-0.99589 \pm 0.32489
	JADE+	-1.0621 \pm 0.29633
F14:Griewank	JADE	0.99975 \pm 2.2327e-14
	JADE+	0.99975 \pm 2.2327e-14
F15:XinSheYang	JADE	0.023408 \pm 0.084715
	JADE+	0.019781 \pm 0.078336
F16:StyblinskiTang	JADE	-77.0742 \pm 4.1255
	JADE+	-77.3569 \pm 3.6402

Table 3. Comparison of mean and standard deviation ($D = 10$)

Function	Method	Mean \pm Standard deviation
F1:Sphere	JADE	7.3367e-47 \pm 1.3492e-45
	JADE+	1.9139e-46 \pm 3.7947e-45
F2:Rosenbrock	JADE	0.00022373 \pm 0.0097558
	JADE+	2.5623e-07 \pm 1.1459e-05
F6:Rastrigin	JADE	0 \pm 0
	JADE+	0 \pm 0
F7:Ackley	JADE	3.503e-15 \pm 4.1751e-16
	JADE+	3.5136e-15 \pm 3.7065e-16
F14:Griewank	JADE	0.99975 \pm 3.6647e-14
	JADE+	0.99975 \pm 3.6647e-14
F15:XinSheYang	JADE	0.00056607 \pm 1.4686e-14
	JADE+	0.00056607 \pm 2.2173e-14
F16:StyblinskiTang	JADE	-391.5415 \pm 1.2981
	JADE+	-391.5274 \pm 1.3717

so different from those of the conventional method. In this experiment, it can be said that the result as intended was obtained, because the search frequency of the outside region did not increase, even if the dimension increased, so that the performance in the high dimension would not be lowered. And, it was proven that it was good to add outer region search like the proposed method, because the region near the optimum solution was lined in F2.

Table 2 shows that the proposed method improved the accuracy of the solution to 11 functions out of 16. For these 11 functions, the proposed method gives better values for both mean and standard deviation. Since all the functions whose accuracy is improved are multimodal functions, the proposed method is more stable and gives better solution in multimodal functions, and it is proven that the search outside the group is effective for the multimodal function. In the sphere function, Rosenbrock function and Matyas function which are the unimodal function, the reason why the accuracy is inferior to JADE is the lowering of the search speed by the out-of-group search, and there seems to be a large room of the improvement on the judging method of whether to carry out the out-group search or not.

Next, we discuss Table 3. In the proposed method, one of the search points searches outside the population. Therefore, as the dimension increases, the effect of out-of-group search decreases and the difference between JADE and JADE+ disappears. In Table 3, we can see that the results are similar between JADE and JADE+.

5 Conclusion

In this study, we propose a method to add search outside the group referring to the Nelder-Mead method to JADE which is adaptive differential evolution. A comparison experiment between the proposed method and the conventional method was carried out using 16 benchmark functions, and it succeeded in improving the accuracy of the solution in many functions. Especially, search outside the group works effectively in the multimodal function, and escape from the local solution is also observed. However, since there were some cases in which the search speed was inferior to JADE in the unimodal function, it is also necessary to examine how to decide whether to search outside the group.

Acknowledgment. This work was supported in part by JSPS KAKENHI Grant Number JP18K11473.

References

1. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob.Optim.* **11**, 341–359 (1997)
2. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. *IEEE Congr. Evol. Comput.* **2**, 1785–1791 (2005)
3. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **10**(6), 646–657 (2006)
4. Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for differential evolution. In: *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, pp. 71–78 (2013)
5. Zhang, J., Sanderson, A.C.: JADE: adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **13**(5), 945–958 (2009)
6. Zhang, J., Sanderson, A.C.: JADE: self-adaptive differential evolution with fast and reliable convergence performance. In: *Proceedings IEEE Congress on Evolutionary Computation*, Singapore, pp. 2251–2258 (2007)
7. Miyahira, Y., Notsu, A., Honda, K., Ubukata, S.: Adaptive selection of parameters in differential evolution by bandit algorithm. In: *Proceeding of the 65th Annual Conference of the Institute of Systems, Control and Information Engineers*, pp. 1020–1026 (2021)
8. Nelder, J.A., Mead, R.: A simplex method for function minimization. *Comput. J.* **7**, 308–313 (1965)
9. Singer, S., Singer, S.: Efficient implementation of the Nelder-Mead search algorithm. *Appl. Numer. Anal. Comput. Math.* **1**(2), 524–534 (2004)
10. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* **9**(2), 159–195 (2001)
11. Storn, R., Price, K.: Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. *J. Glob. Optim.* **23**(1) (1995)