

Cooperative Particle Swarm Optimization in Distance-Based Clustered Groups

Tomohiro Hayashida, Ichiro Nishizaki, Shinya Sekizaki, Shunsuke Koto

Faculty of Engineering, Hiroshima University, Hiroshima, Japan

Email: hayashida@hiroshima-u.ac.jp

How to cite this paper: Hayashida, T., Nishizaki, I., Sekizaki, S. and Koto, S. (2017) Cooperative Particle Swarm Optimization in Distance-Based Clustered Groups. *Journal of Software Engineering and Applications*, 10, 143-158.

<https://doi.org/10.4236/jsea.2017.102008>

Received: December 9, 2016

Accepted: February 4, 2017

Published: February 7, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

TCPSO (Two-swarm Cooperative Particle Swarm Optimization) has been proposed by Sun and Li in 2014. TCPSO divides the swarms into two groups with different migration rules, and it has higher performance for high-dimensional nonlinear optimization problems than traditional PSO and other modified method of PSO. This paper proposes a particle swarm optimization by modifying TCPSO to avoid inappropriate convergence onto local optima. The quite feature of the proposed method is that two kinds of subpopulations constructed based on the scheme of TCPSO are divided into some clusters based on distance measure, k -means clustering method, to maintain both diversity and centralization of search process are maintained. This paper conducts numerical experiments using several types of functions, and the experimental results indicate that the proposed method has higher performance than the TCPSO for large-scale optimization problems.

Keywords

Particle Swarm Optimization, Different Migration Rules, Clustering

1. Introduction

PSO (Particle Swarm Optimization) is one of the most famous evolutionary computation methods that emulate the swarm behavior of some kinds of birds [1]. This evolutionary computation method interprets that the feature of the corresponding problem is not independently recorded on each individual of the swarm, but the whole population share the information for smart search. In the search process of PSO, each individual gets closer to an individual with highest evaluation value, by updating the current solution of each individual. A variety of PSO are applied to a number of problems in engineering discipline, because the updating method of the solution is very simple and computationally cheap [2] [3] [4].

However, search process of PSO rarely converges a local optimum solution depending on the initial solution and the parameter settings due to the simplicity of the search procedure. To avoid the convergence problem several improved PSOs are proposed. For example, fully informed PSO (FIPSO) that several subgroups of particles (individuals) can communicate to each other by intermediary of the link between the subgroups [5] [6], linearly decreasing weight PSO (LDWPSO) such that the value of the parameters in regard to search process are adapted to actualize a global search in the first part of the search process and a local search toward the end of the process [7], FAPSO-ACO-K, a hybrid algorithm of fuzzy adaptive PSO (FAPSO), ant colony optimization (ACO), and k -means clustering, is combined search procedure of hybrid of FAPSO and ACO for local search and k -means clustering method for global search [8], two-swarm cooperative PSO (TCPSO) which maintains the diversity of the population by division of the population into two kinds of subgroups [9], and so forth.

The manner of division in TCPSO is not always suitable for the corresponding problems, thus the search process of TCPSO is not always appropriate. This study improves TCPSO by applying a statical clustering method for effective division of the population. The clustering is applied several times in process of search based on the degree of convergence of search. The experimental results indicate that the proposed method has higher performance for some high-dimensional problems than the existing methods relevant to PSO.

The rest of this paper is structured as follows: In Section 2, several works related to PSO, the original PSO, FIPSO, and TCPSO are briefly introduced. In Section 3, the proposed method using distance-based clustering method is constructed. In Section 4, numerical experiments using multiple benchmark problems are conducted, and finally Section 5 concludes this paper.

2. Related Works

In this section, some related works, PSO, FIPSO, and TCPSO are briefly described.

2.1. Particle Swarm Optimization (PSO)

Let n be the number of swarms (individuals), and a swarm i , $i = 1, 2, \dots, n$ retains its positional information vector at time t , $x_i(t)$, and its velocity vector, $v_i(t)$, where $x_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{ij}(t))$ and

$v_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{ij}(t))$, here J indicates number of dimension of the corresponding problem. Here, let \underline{x}_j and \bar{x}_j be the minimum and maximum values of the j -th dimensional value, respectively, *i.e.*, $x_{ij} \in [\underline{x}_j, \bar{x}_j], \forall i, j$ is satisfied. Additionally, a swarm i retains a positional information vector

$x_i^*(t) = (x_{i1}^*(t), x_{i2}^*(t), \dots, x_{ij}^*(t))$ of which has the highest evaluation value in which the swarm experienced from time 0 to t . In similar way, whole swarms in the population shares a positional information vector

$X^*(t) = (X_1^*(t), X_2^*(t), \dots, X_j^*(t))$ of which has the highest evaluation value in which whole swarms experienced from time 0 to t . The position information

vector $x_i(t)$ and velocity vector $v_i(t)$ of a swarm i is updated by using following equation.

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (1)$$

$$v_{ij}(t+1) = wv_{ij}(t) + C_1r_1(x_{ij}^*(t) - x_{ij}(t)) + C_2r_2(X_j^*(t) - x_{ij}(t)), \quad (2)$$

where C_1 and C_2 are parameters, w indicates inertia coefficient, and $r_1, r_2 \in [0,1]$ are random numbers which are determined before each updating. From (1) and (2), the performance of PSO depends in a large part on values of these parameters. The positions of all swarms are updated depending also on the common position information vector X^* , therefore it is difficult to deviate from a local optima if the search process converges near the local optima.

2.2. Fully Informed PSO (FIPSO)

Whereas whole swarms in the population share the common position information in PSO, a pair of swarms which located at nearest in whole swarms each other share personal best position information with the highest evaluation x_i^* in FIPSO. The nearest swarm of a swarm is called neighbor of her/him, and let $b(i)$ be a neighbor of a swarm i . FIPSO updates the position information vector $x_{ij}(t)$ by using following (1) and the velocity vector $v_{ij}(t)$ by using following equation.

$$v_{ij}(t+1) = \chi \left\{ v_{ij}(t) + \frac{1}{n_i} \sum_{i=1}^{n_i} C_1 r_1 (x_{ij}^*(t : b(i)) - x_{ij}(t)) \right\}, \quad (3)$$

where χ is a learning parameter.

2.3. Two-Swarm Cooperative PSO (TCPSO)

TCPSO divides population into two subgroups, master swarms and slave swarms. As shown in **Figure 1**, the master swarms are assigned to wide search and the slave swarms are assigned to intensive local search.

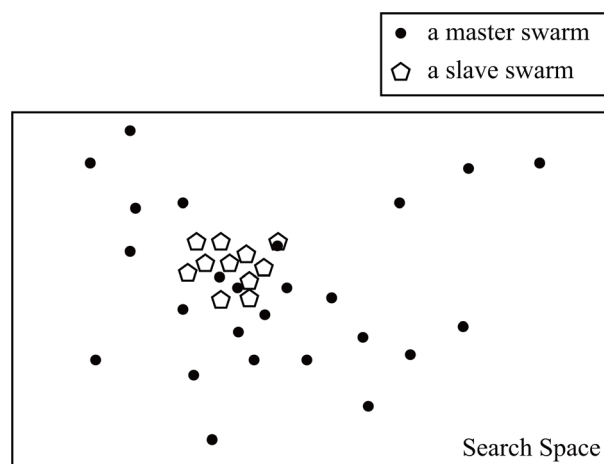


Figure 1. The distribution of swarms: TCPSO.

The position information vector $x_i^M(t)$ and the velocity vector $v_i^M(t)$ of swarm i belonging to master swarms are updated by following equations.

$$x_{ij}^M(t+1) = x_{ij}^M(t) + v_{ij}^M(t+1) \quad (4)$$

$$\begin{aligned} v_{ij}^M(t+1) = & wv_{ij}^M(t) + C_1^M r_1 (x_{ij}^{M*}(t) - x_{ij}^M(t)) \\ & + C_2^M r_2 (X_j^{S*}(t) - x_{ij}^M(t)) \\ & + C_3^M r_3 (X_j^*(t) - x_{ij}^M(t)) \end{aligned} \quad (5)$$

The position information vector $x_i^S(t)$ and the velocity vector $v_i^S(t)$ of swarm i belonging to slave swarms are updated by following equations.

$$x_{ij}^S(t+1) = x_{ij}^S(t) + v_{ij}^S(t+1) \quad (6)$$

$$\begin{aligned} v_{ij}^S(t+1) = & wv_{ij}^S(t) + C_1 r_1 (x_{ij}^{S*}(t) - x_{ij}^S(t)) \\ & + C_2 r_2 (X_j^*(t) - x_{ij}^S(t)) \end{aligned} \quad (7)$$

In Equations (4)-(7), “ M ” and “ S ” indicate the master and the slave swarms in order, respectively. From Equation (7), a slave swarm updates the velocity vector $v_i^S(t)$ based only on the position information, but the last velocity. This updating mechanism without inertia term leads that search regions of the slave swarms becomes more narrow than of the master swarms. The difference of search area between the master and the slave swarms are due to updating mechanism differences such whether it refers another kind of swarms or not. On the other hand, the term including $X_j^{S*}(t)$ in the updating mechanism of the velocity of a master slave refers the search efforts of the slave swarms. Several kinds of experimental results using TCPSO indicate that it has satisfying performance in many types of optimization problems. However, it does not have satisfying performance in high dimensional maps.

3. Distance-Based Divided Groups and Cooperative PSO

As described in the above section, TCPSO divides the population into two groups with different migration rules. However, the master swarms cannot always maintain the search area widely, because, a master swarm updates its velocity in refer not only to master swarms but also the best solution in the slave swarms as Equations (4) and (5). This is a reason of which the searching process of TCPSO rarely converges on the local optima of the target optimization problem which is not the global optima of it, depending on the future of target problems.

This paper revises TCPSO to avoid the convergence of the swarms on the local optima by dividing the master swarms into multiple groups based on Euclidean distance. This is the main feature of the proposed method. Similarly, the slave swarms are divided into same number of groups and each group is connected a group of divided master swarms. For distance-based clustering, k -means clustering method is applied for dividing the swarms.

3.1. k -Means Clustering

The proposed method applies k -means clustering method to the population.

k -means clustering is a non-hierarchical clustering algorithm, the whole swarms in the population are divided into k groups based on the distance measure. Here, let \bar{x}_κ be a median point of the positional information vector of the swarms belonging to a cluster κ . Revise the division of the population into cluster to satisfy following condition.

$$\arg \min_{V_1, \dots, V_k} \sum_{i=1}^n \min_{\kappa} \|x_i - \bar{x}_\kappa\| \quad (8)$$

3.2. Algorithms

Let $X^{M\kappa^*}$ and $X^{S\kappa^*}$ indicate the locational information vector of which has the highest evaluation value in which the master and slave swarms in a cluster κ experienced from time 0 to t . This study revises update procedure of the velocity vector of a swarm as follows, differing depending on the kind whether the swarm is master or slave swarm as following equations.

$$\begin{aligned} v_{ij}^M(t+1) = & wv_{ij}^M(t) + C_1^M r_1 (x_{ij}^{M^*}(t) - x_{ij}^M(t)) \\ & + C_2^M r_2 (X_j^{S^*}(t) - x_{ij}^M(t)) \\ & + C_3^M r_3 (X_j^{M\kappa^*}(t) - x_{ij}^M(t)) \\ & + C_4^M r_4 (X_j^*(t) - x_{ij}^M(t)) \end{aligned} \quad (9)$$

$$\begin{aligned} v_{ij}^S(t+1) = & C_1^S r_1 (x_{ij}^{S^*}(t) - x_{ij}^S(t)) \\ & + C_2^S r_2 (X_j^*(t) - x_{ij}^S(t)) \\ & + C_3^S r_3 \sum_{\kappa=1}^k (X_j^{M\kappa^*}(t) - x_{ij}^S(t)), \end{aligned} \quad (10)$$

where, $r_1, r_2, r_3, r_4 \in [0,1]$ are the random variables as employed in Equations (2), (5), and (7). The positional information vector is updated based on Equations (4) and (6). Equation (9) maintains or increases diversity of search process of TCPSO by referring the positional information vector $X^{M\kappa^*}(t)$ with the highest evaluation value in the belonging cluster, and Equation (10) avoids excessive convergence.

The outline of the proposed method is briefly summarized as follows.

Step 0 Initialize the parameter $k^{\text{temp}} = k = 2$.

Step 3 Generate initial population of the master and the slave swarms.

Step 2 If $k > 1$, execute k -means clustering to the population of master and the slave swarms.

Step 3 Calculate the evaluation value of each swarm, based on the positional information vector.

Step 4 Update the positional information and velocity vector by Equations (4), (6), (9), and (10). If the largest evaluation value in the population is larger than predetermined value, it remains unchanged during for a given period, or number of updating of the positional information and the velocity vector approaches predetermined number, then terminate the search process. If a convergence condition satisfied, then go to Step 3, otherwise go to Step 5.

Step 5 If $k = 1$, then let $k = k^{\text{temp}} + 1$ and go to Step 2. If $k > 1$, then let

$k^{\text{temp}} = k, k = 1$ and go to Step 2.

A quite feature of the proposed method is applying k -means clustering to population of the master swarms and the slave swarms, respectively, with changing number of clusters k . Note that there exists a hybrid algorithm using k -means clustering, FAPSO-ACO-K [8], the algorithm divides the all swarms in the population into clusters. Our method has stronger tendency to avoid the convergence of swarms on the local optima, because the number of clusters k periodically changes in our method as 2,1,3,1,4,1,⋯. Whereas FAPSO-ACO-K uses predetermined number of clusters. The desired effect of the proposed method is that population repeats widespread migrations and convergences when the number of k changes at Step 5. It is expected to discover another better solutions by the widespread migration, and intensive local search by the convergence of the search process.

4. Numerical Experiments

In this section, numerical experiments using some kinds of nonlinear functions. The terminate conditions are set as that the function values $f(x)$ is less than 1.0×10^{-10} or the number of iterations without change of the largest evaluation values of solution. 10 discrete trail runs are executed for each problem by changing number of the dimensions of the problems as $D = 2, 10, 30, 50$. The values of parameters are set as **Table 1**.

The numerical experiments which conduct in this study use 11 kinds of functions shown in **Table 2** and **Table 3**.

Table 1. Parameter settings.

Number of dimensions D	2/10/30/50/100
Inertia coefficient w	0.9
Number of swarms n	60
Parameters $(C_1^S, C_2^S, C_1^M, C_2^M, C_3^M)$	(1.6, 1.6, 2.2, 2.2, 2.2)
Number of initial cluster	2
Maximum number of clusters	5

Table 2. Benchmark problems (unimodal functions).

Name	Function $f(x_1, x_2, \dots, x_D)$	Optimal solution
Sphere	$\sum_{i=1}^D x_i^2$	$f(0, 0, \dots, 0) = 0$
Quadric	$\sum_{i=1}^D ix_i^4$	$f(0, 0, \dots, 0) = 0$
Schwefel's Problem 1.2	$\sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$f(0, 0, \dots, 0) = 0$
Rosenbrock	$\sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$f(1, 1, \dots, 1) = 0$
Schwefel's problem 2.22	$\sum_{i=1}^D x_i^2 + \prod_{i=1}^D x_i $	$f(0, 0, \dots, 0) = 0$

Table 3. Benchmark problems (multi-modal functions).

Name	Function $f(x_1, x_2, \dots, x_D)$
Ackley	$-20 \exp \left\{ -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right\} - \exp \left\{ \frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i \right\} + 10 + e$ <p style="text-align: right;">Optimal solution: $f(0, 0, \dots, 0) = 0$</p>
Generalized Rastrigin	$\sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$ <p style="text-align: right;">Optimal solution: $f(0, 0, \dots, 0) = 0$</p>
Generalized Schwefel's problem 2.26	$\sum_{i=1}^D (x_i \sin \sqrt{ x_i })$ <p style="text-align: right;">Optimal solution: $f(a, a, \dots, a) = f_{GS}^*$, ($a = 420.9687, f_{GS}^* = -12569.5$)</p>
Generalized Griewank	$\frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$ <p style="text-align: right;">Optimal solution: $f(0, 0, \dots, 0) = 0$</p>
Generalized Penalized 1	$\frac{\pi}{D} \left[10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} \left\{ (y_i - 1)^2 (1 + 10 \sin^2(\pi y_{i+1})) \right\} + (y_D - 1)^2 \right]$ $+ \sum_{i=1}^D u_i(x_i, 10, 100, 4), y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$ <p style="text-align: right;">Optimal solution: $f(-1, -1, \dots, -1) = 0$</p>
Generalized Penalized 2	$0.1 \left\{ \sin^2(\pi 3x_1) \sum_{i=1}^{D-1} \left\{ (x_i - 1)^2 [1 + 10 \sin^2(3\pi x_{i+1})] \right\} \right\}$ $(x_D - 1)^2 [1 + \sin^2(2\pi x_D)] + \sum_{i=1}^D u_i(x_i, 5, 100, 4), y_i = 1 + \frac{1}{4}(x_i + 1),$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$ <p style="text-align: right;">Optimal solution: $f(1, 1, \dots, 1) = 0$</p>

The aim of these optimization problem is finding the solution vector $x = (x_1, x_2, \dots, x_D)$ which minimizes each target function. The functions are classified in terms of unimodal or multi-modal. D indicates number of dimensions. In **Appendix**, some functions with number of dimensions is 2 ($D = 2$) are shown in **Figures 6-15** for example.

The error per number of dimensions D and termination term per D are shown in **Table 4** and **Table 5** as experimental result. **Figures 2-5** summarizes these results.

From the experimental result shown in **Table 4**, **Figure 2** and **Figure 3**, the proposed method has higher or approximately equivalent performance relative to TCPSO in the optimization problem of unimodal functions. In other words, the proposed method is more helpful than the comparative approach, TCPSO. In the case of Rosenbrock, it is a unimodal function, though the gradient around

Table 4. Experimental results (unimodal functions).

Target function	Proposed method			TCPSO [9]	
	D	Error/ D	Termination term/ D	Error/ D	Termination term/ D
Sphere (Unimodal)	2	0	11.0	0	10.7
	10	0	71.3	0	73.0
	30	0	50.8	0	53.1
	50	0	47.9	0	52.1
	100	0	53.6	0	64.4
Quadric (Unimodal)	2	0	6.7	0	6.6
	10	0	39.8	0	42.0
	30	0	40.7	0	44.3
	50	0	49.2	0	53.8
	100	0	68.3	0	74.0
Schwefel's Problem 1.2 (Unimodal)	2	0	11.6	0	10.9
	10	0	81.7	0	79.2
	30	0	59.9	0	62.8
	50	0	55.9	63.6	
	100	0	64.5	0	75.8
Rosenbrock (Unimodal)	2	0	228.4	0	235.8
	10	0	3827.8	0	3307.2
	30	0	3935.8	0	4801.0
	50	0	3152.2	0	3320.5
	100	0.00644	2032.0	0.00399	2125.6
Schwefel's problem 2.22 (Unimodal)	2	0	17.9	0	20.8
	10	0	561.4	0	592.4
	30	0.627	1276.0	0.693	1102.4
	50	1.228	894.6	1.216	730.0
	100	1.719	1028.0	1.708	679.0

the optimal solution is very small, and it is very difficult to find the optimal solution of such problems by heuristic approaches.

As shown in **Table 5**, **Figure 4** and **Figure 5**, the proposed method has higher performance relative to TCPSO also in almost types of multi-modal functions. However, the function “Generalized penalized 2” with the number of dimensions is $D = 100$, the amount of error by the proposed method is larger than of TCPSO. This function is nearly discrete type function as shown in **Figure 15**. From the experimental result, the proposed method is very effective for a lot of types of optimization problems, however, it should be revised to improve the performance also in high-dimensional discrete type functions such as “Generalized penalized 2”.

Table 5. Experimental results (multi-modal functions).

Target function	Proposed method			TCPSO [9]	
	D	Error/ D	Termination term/ D	Error/ D	Termination term/ D
Ackley (Multi-modal)	2	0	21.7	0	20.6
	10	0	149.1	0	226.3
	30	0	178.1	0	364.6
	50	0.667	194.8	3.97	350.8
	100	0	146.5	0.710	202.3
Generalized Rastrigin (Multi-modal)	2	0	13.0	0	11.6
	10	0	71.9	0	71.1
	30	0	68.9	0	70.6
	50	0	97.9	0.002	107.0
	100	0.058	354.6	0.516	399.6
Generalized Schewefel's problem 2.26 (Multi-modal)	2	74.4	2571.0	137.5	4042.1
	10	1198.2	1702.7	1448.3	1343.3
	30	5383.2	920.7	5607.6	506.6
	50	9389.2	761.5	9772.9	355.0
	100	19138.3	1245.8	21225.8	288.4
Generalized Griewank (Multi-modal)	2	0	102.1	0	112.0
	10	0.034	1271.1	0.045	1427.5
	30	0.0066	237.9	0.011	290.4
	50	0.0034	139.6	0.0032	124.2
	100	0.0019	79.6	0.00057	62.9
Generalized penalized 1 (Multi-modal)	2	0	11.0	0	10.8
	10	0	132.6	0	141.1
	30	213.3	808.5	408.7	671.1
	50	699.4	596.8	752.7	443.1
	100	1578.7	291.9	1654.2	264.0
Generalized penalized 2 (Multi-modal)	2	0	32.1	0	14.7
	10	0	80.8	0	83.5
	30	0	167.1	0	153.3
	50	0	217.9	0	198.9
	100	0.152	380.2	0.124	351.9

5. Conclusions

This paper proposed a procedure of particle swarm optimization (PSO), which is constructed based on two-swarm cooperative PSO (TCPSO) [9] and includes the procedure of distance-based clustering. The main idea of the proposed method is dividing whole swarms into multiple subgroups by using k -means clustering, and the divisions are executed several times with periodical change of k during the search process. This mechanism maintains the diversity and centralization of the search, and resolves the optimization problem of several kinds of functions.

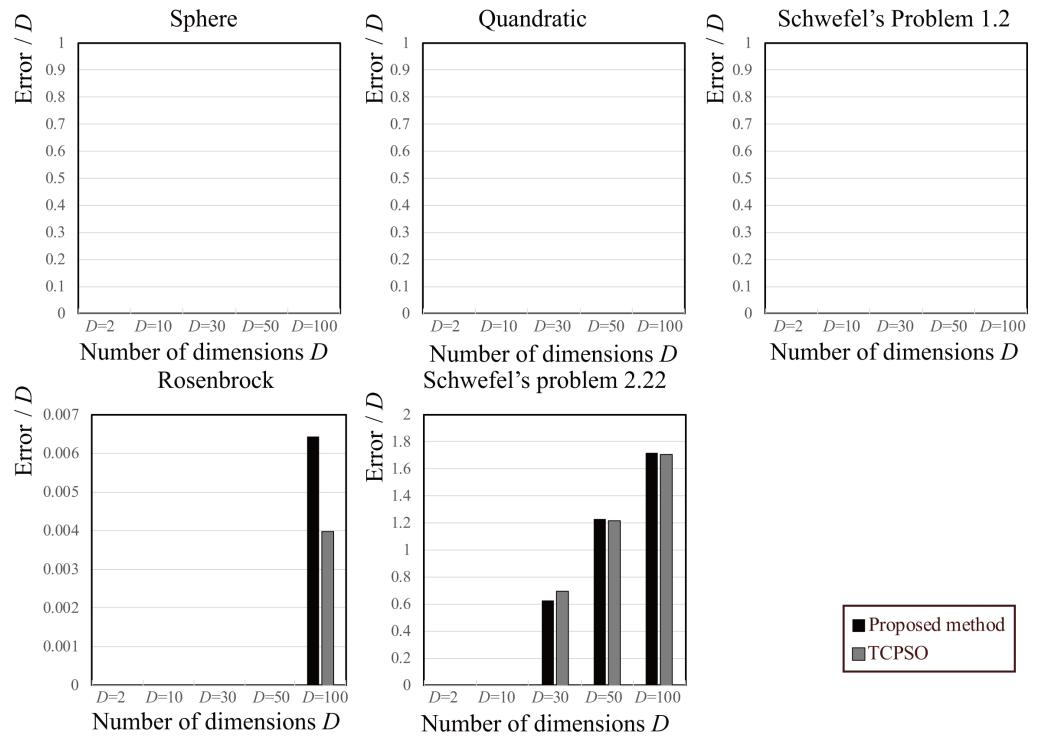


Figure 2. Experimental result: error/D (unimodal functions).

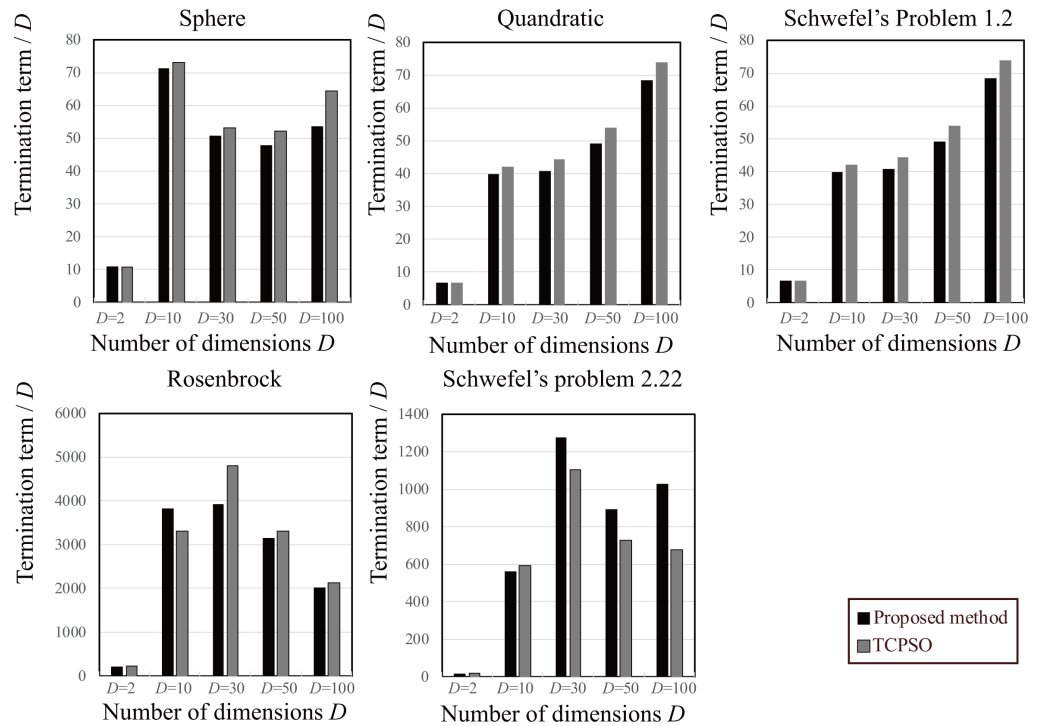


Figure 3. Experimental result: termination term/D (unimodal functions).

This paper conducts numerical experiments using some benchmark problems of unimodal and multi-modal functions, and the experimental results indicate that the proposed method succeed to discover better solutions of some problems compared to the existing method (TCPSO).

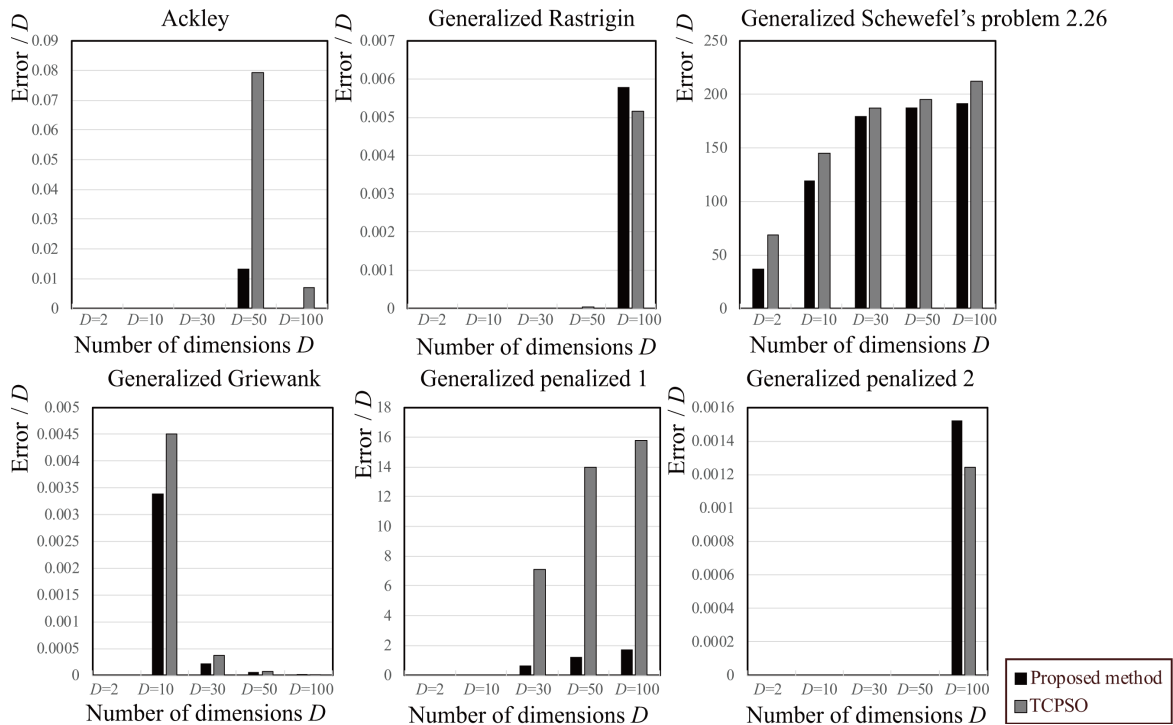


Figure 4. Experimental result: error/ D (multi-modal functions).

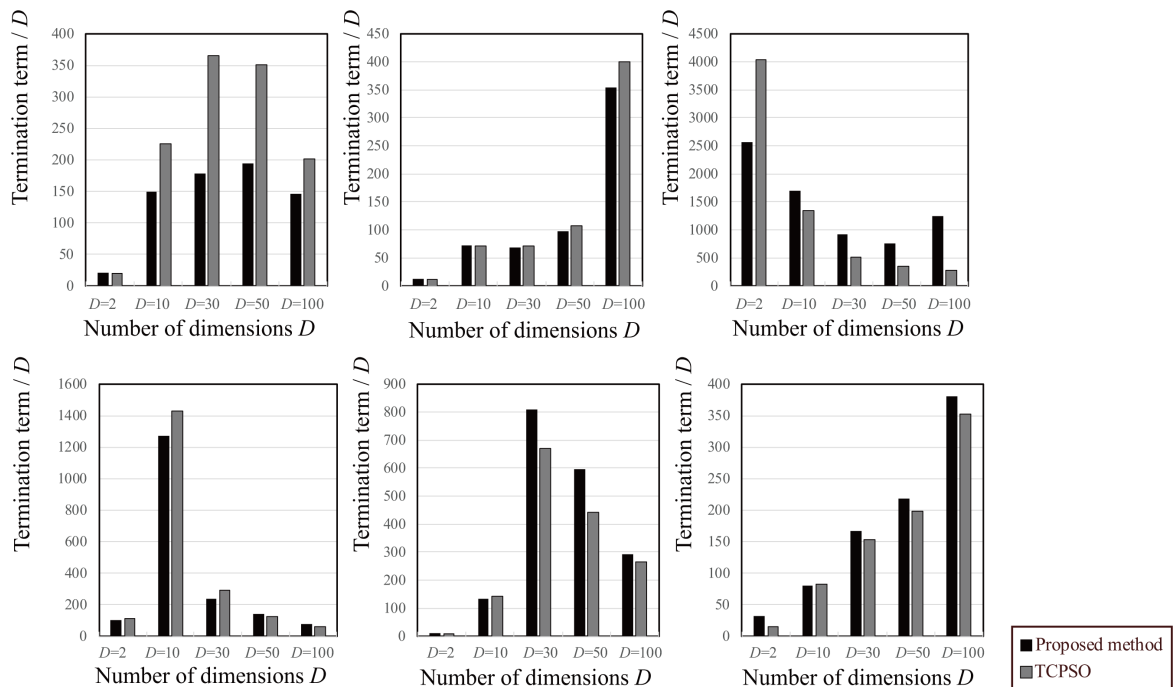


Figure 5. Experimental result: termination term/ D (multi-modal functions).

As shown in Table 4, Figure 2 and Figure 3, only in a case of high-dimensional function of Rosenbrock ($D=100$), the proposed method is obviously defeated by TCPSO, we should explain the reason and propose an improvement of the performance, for example, by revising the condition of change of k is improved.

References

- [1] Kennedy, J. and Eberhart, R.C. (1995) Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, 27 November-1 December 1995, 1942-1948. <https://doi.org/10.1109/ICNN.1995.488968>
- [2] Babazadeha, A., Poorzahedyb, H. and Nikoosokhana, S. (2011) Application of Particle Swarm Optimization to Transportation Network Design Problem. *Journal of King Saud University—Special Issue on “Advances in Transportation Science”*, **23**, 293-300.
- [3] Esmin, A.A.A. and Lambert-Torres, G. (2012) Application of Particle Swarm Optimization to Optimal Power Systems. *International Journal of Innovative Computing*, **8**, 1705-1716.
- [4] Li, W. and Wang, G.-Y. (2010) Application of Improved PSO in Mobile Robotic Path Planning. *Proceedings of International Conference on Intelligent Computing and Integrated Systems (ICISS)*, Guilin, October 2010, 45-48.
- [5] Kennedy, J. and Mendes, R. (2002) Population Structure and Particle Swarm Performance. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, Honolulu, May 2002, 1671-1676.
- [6] Kennedy, J. and Mendes, R. (2006) Neighborhood Topologies in Fully Informed and Best of Neighborhood Particle Swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C, Applications and Reviews*, **36**, 515-519. <https://doi.org/10.1109/TSMCC.2006.875410>
- [7] Yang, C.-H., Hsiao, C.-H. and Chuang, L.-Y. (2010) Linearly Decreasing Weight Particle Swarm Optimization with Accelerated Strategy for Data Clustering. *IAENG International Journal of Computer Science*, **37**. (Online Available)
- [8] Niknam, T. and Amiri, B. (2010) An Efficient Hybrid Approach Based on PSO, ACO and k -Means for Cluster Analysis. *Applied Soft Computing*, **10**, 183-187. <https://doi.org/10.1016/j.asoc.2009.07.001>
- [9] Sun, S. and Li, J. (2014) A Two-Swarm Cooperative Particle Swarms Optimizations. *Swarm and Evolutionary Computation*, **15**, 1-18. <https://doi.org/10.1016/j.swevo.2013.10.003>

Appendix

Benchmark problems (unimodal functions)

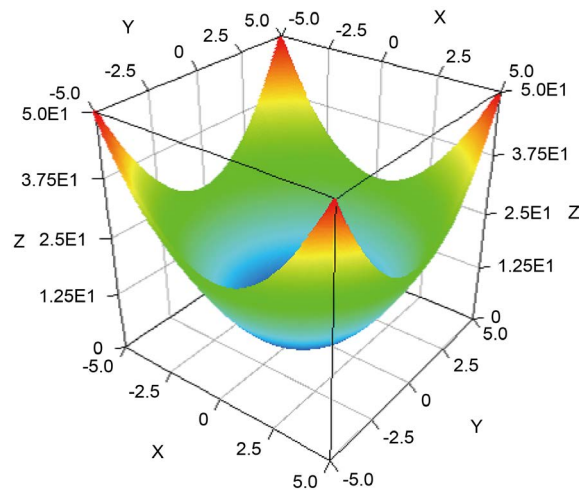


Figure 6. Sphere function ($D = 2$).

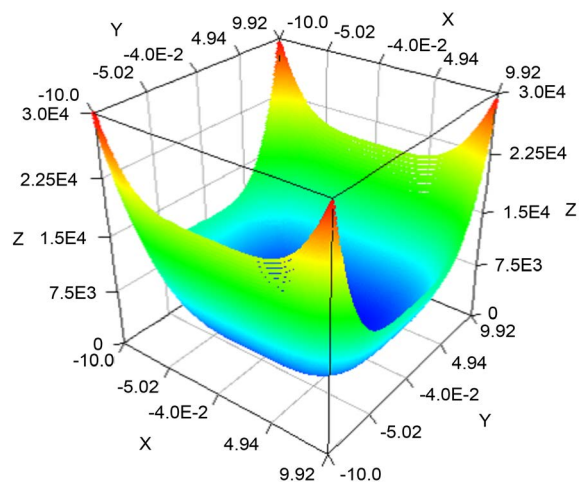


Figure 7. Quadratic function ($D = 2$).

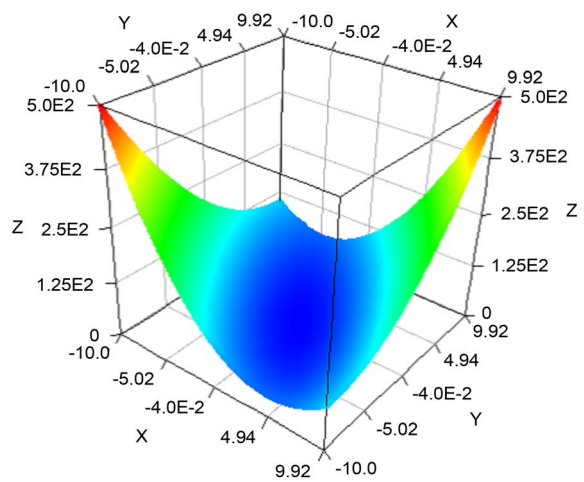


Figure 8. Schwefel's problem 1.2 function ($D = 2$).

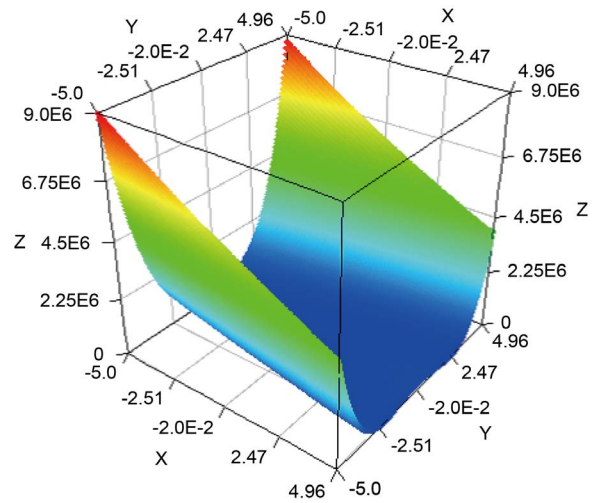


Figure 9. Rosenbrock function ($D = 2$).

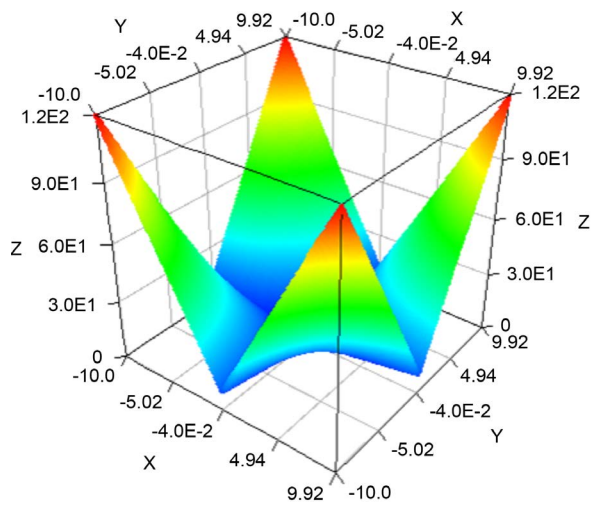


Figure 10. Schwefel's problem 2.22 function ($D = 2$).

Benchmark problems (multi-modal functions)

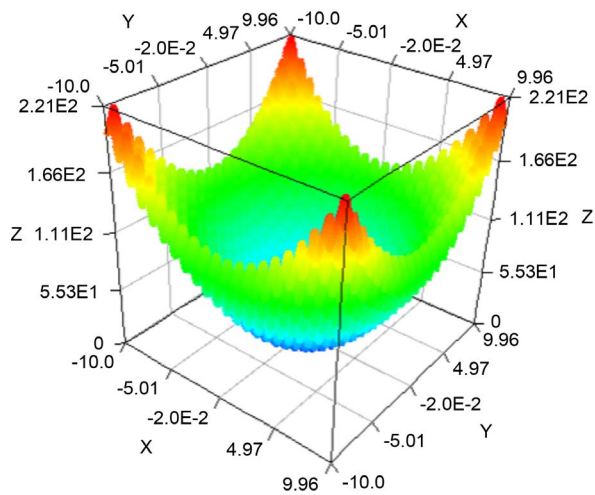


Figure 11. Generalized Rastrigin function ($D = 2$).

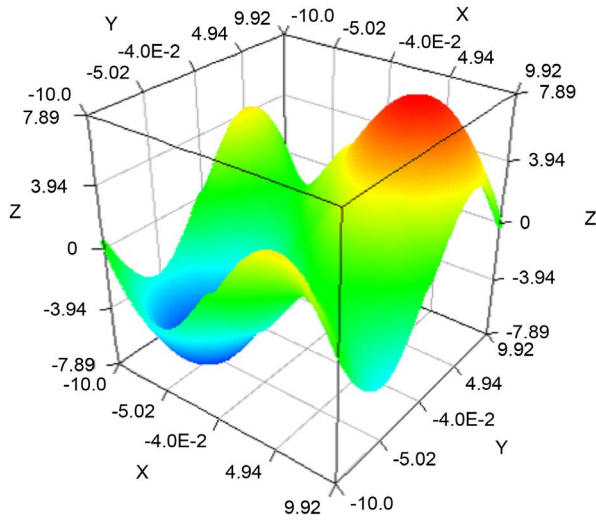


Figure 12. Generalized Schwefel's problem 2.26 function ($D = 2$).

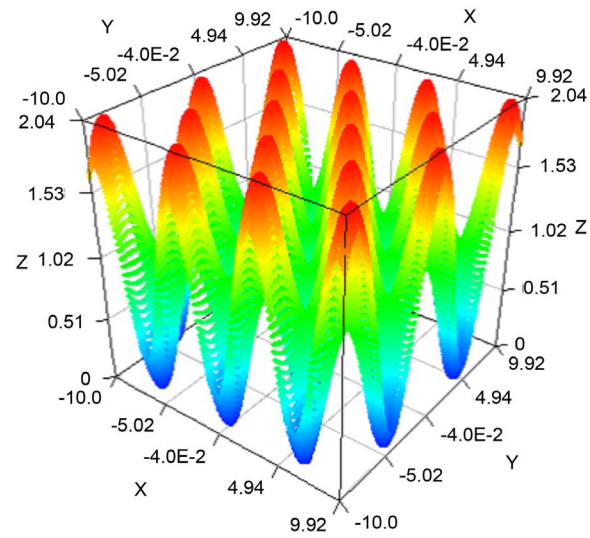


Figure 13. Generalized Griewank function ($D = 2$).

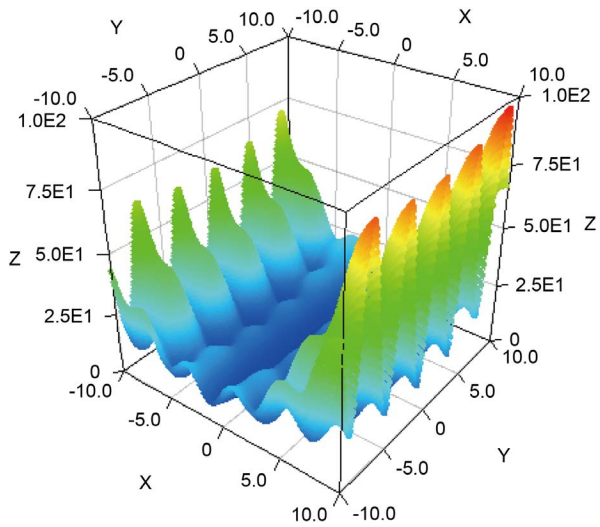


Figure 14. Generalized penalized 1 function ($D = 2$).

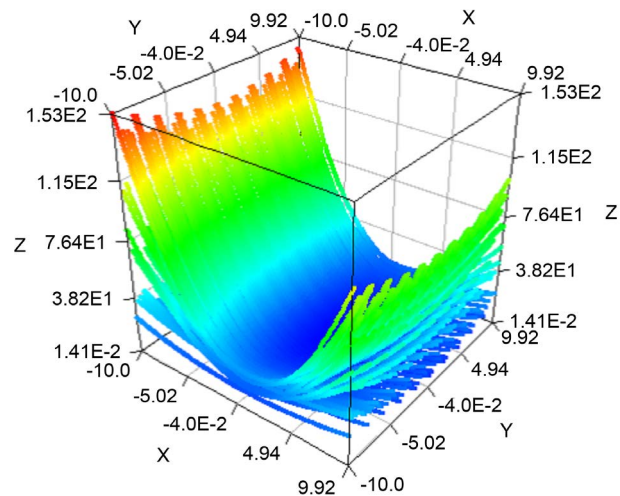


Figure 15. Generalized penalized 2 function ($D = 2$).



Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact jsea@scirp.org