Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 14, Number 1, pages 18–21, January 2025

Multitask System Design by an Improved PSO Algorithm

Wei-Der Chang* Shan-Cheng Pan
Department of Computer and Communication
Shu-Te University
Kaohsiung 824, Taiwan
email: wdchang@stu.edu.tw

***Abstract*—This paper will propose a simple modified version of the particle swarm optimization (PSO) algorithm to solve multitasking problems. In the proposed scheme, a single population existed in the original PSO algorithm is firstly divided into several subpopulations and the number of subpopulations entirely depends on the number of the problems that will be solved. The related information regarding the best particle within each subpopulation needs to be recorded, and each of them is to deal with one corresponding problem. To show the applicability and efficiency of the developed method, some target searching problems with two dimensions are examined.**

***Keywords*—multitasking system, particle swarm optimization (PSO), subpopulations, target searching**

I. INTRODUCTION

Multitasking means that several tasks or problems can be solved on the same time. In recent years, this concept has been applied in a variety of practical applications such as media [1][2], object tracking [3][4], face identification [5] and robot systems [6][7]. To cope with the multitasking problem, one of commonly used methods is the multi-task learning (MTL) which is designed to train multiple tasks jointly and simultaneously and has been shown to be more effective than independently training each single task [8][9]. In [4], the authors used the collaborative MTL and appearance model updating to solve the object tracking problem. In [10], a unified single/multi-view human action recognition method was developed via regularized MTL and the final experiment results revealed that this method can significantly improve recognition performance over the general method. To tackle the multitasking problem, however, this paper will use a simple modified version of particle swarm optimization (PSO) algorithm rather than the MTL approach. Under the modified structure, a single population initially used in the algorithm has to be divided into several subpopulations, and each subpopulation then aims at one corresponding task. Once the algorithm is executed, multiple tasks can simultaneously be achieved.

In recent years, the PSO algorithm has become a popular and commonly used algorithm for solving optimization problems because of its simplicity and efficiency. To fulfill the system optimization, this kind of algorithm only utilizes two main updating formulas, i.e., the velocity and position updating formulas. In addition, all the evolutionary operations use a real-value representation. As a result, it is very suitable for most of physical engineering applications. Recently, for certain special purposes some variants of PSO algorithm have successively been presented; for instance, the algorithm with multiple subpopulations (or subswarms) [11]-[15]. Liao et al. developed an accurate sub-swarms PSO algorithm by adopting parallel and serial niching techniques. It locates optimal solutions by using sub-swarms searching grid cells where the density of feasible solutions is high [11]. In [15], a multi-swarm self-adaptive cooperative PSO algorithm was proposed based on four sub-swarms. Several strategies are utilized to avoid falling into local optimum and further to achieve better solution. Some well-known benchmarks are simulated to show the good performance of the proposed algorithm in solving complex multimodal functions.

In [12], an improved PSO algorithm with many subpopulations has been shown for multimodal function optimizations. The so-called multimodal function means having a lot of optimum points. At the beginning, the original single population is partitioned into several subpopulations simply according to the order of particles. A modified velocity updating formula where the global best particle is replaced by the best particle of each subpopulation is executed. Particle movements of each subpopulation are guided by its corresponding best particle. Under this architecture, each subpopulation can separately and individually catch one system optimum of the solved multimodal function. Thus, several optimums probably including the global optimums and local optimums can simultaneously be derived only when the algorithm is performed one time. Conversely, after executing the traditional scheme which contains a single population, only one system solution is caught. This paper will adopt and extend this kind of modified PSO algorithm for multitasking problem. Each subpopulation aims at one corresponding task or objective function, and there is no relation between the subpopulations. The proposed scheme can simultaneously deal with multiple task problems which the number of tasks solved is equal to that of subpopulations used in the developed algorithm. This is the main contribution of this paper.

II. THE GENERAL PSO ALGORITHM WITH A SINGLE POPULATION

The first version of PSO algorithm was proposed by Kennedy and Eberhart in 1995 [16]. Henceforth it becomes a very well-known optimization tool and has been successfully applied in solving various optimization problems. The initial motivation of this algorithm is to mimic the social behavior of fish school and bird flock. There is always a leader to guide the movement of the whole school or flock towards better directions. Consequently, the information of such the leader is very critical and needs to be enrolled. Following this concept, the general PSO algorithm begins with generating a single population that consists of a large number of particles or called individuals. Generally, a better particle is the one with lower objective value when the minimization problem is solved. In the PSO algorithm, two kinds of principle information have to be recorded during the evolution, i.e., the individual best for each particle and the global best for the whole population. The individual best stands for the best one for each particle from the beginning to present iteration and is denoted by *pbest*. In addition, the best one in the whole population is then referred to as the global best denoted by *gbest*.

To formulate the general PSO algorithm, firstly let $\Theta_i = [\theta_{i1}, \theta_{i2}, \cdots, \theta_{in}]$ be a representation of the *i*th particle where θ_{ij} is the designed parameter of the optimization problem for $i = 1, 2, \cdots, PS$ and $j = 1, 2, \cdots, n$, PS and n denote the number of particles (population size) and designed parameters, respectively. A population is then constructed by many such particles, and the general PSO algorithm has only one population during the evolution. For updating the movement of each particle at the *k*th iteration, the following two equations are employed including the velocity updating formula of (1) and the position updating formula of (2):

$$v_{ij}(k+1) \leftarrow w v_{ij}(k) + c_1 r_1 (pbest_{ij}(k) - \theta_{ij}(k)) + c_2 r_2 (gbest_j(k) - \theta_{ij}(k)) \quad (1)$$

$$\theta_{ij}(k+1) \leftarrow \theta_{ij}(k) + v_{ij}(k+1) \quad (2)$$

where $pbest_{ij}$ and $gbest_j$ are the *j*th position component of the *i*th individual best and the global best particle, respectively, v_{ij} is the *j*th velocity component of the *i*th particle, *w* represents the inertia weight to balance the global and local search, c_1 and c_2 are two positive constants which are given by the designer, r_1 and r_2 are two random numbers generated from the interval $[0, 1]$ uniformly. The PSO algorithm utilizes these two updating formulas to achieve the optimization. Again, it is noticed that the general PSO contains a single population and only one global best particle to tackle one objective function (one task).

III. A MODIFIED PSO VERSION AND ITS APPLICATION TO MULTITASKING PROBLEM

Due to only one population used in the general PSO algorithm, it can just solve a single task with some objective function. For solving the multitasking problem, the general version is not enough and unsuitable. This paper is to utilize a multi-subpopulation PSO algorithm to achieve the multitasking design. This modified version was initially proposed to seek for several system optimums for multimodal functions [12]. In the developed scheme, the original global best *gbest* in (1) is just replaced by the best particle of each subpopulation, and the others are same with the general PSO algorithm. Based on the simple modification, the proposed method with multiple subpopulations is able to simultaneously solve several global and/or local optimums of the multimodal function.

For the multimodal function problem mentioned above, it is to cope with a single objective function, i.e., the multimodal function. In this paper, the multi-subpopulations scheme is further applied to solving the multitasking problem. Each task has one corresponding objective function that will be minimized. The proposed method can synchronously tackle many tasks when the algorithm is executed only one time. Fig. 1 shows the diagram of developed method where *N* represents the number of subpopulations and the best particle within each subpopulation is marked by green. It can be seen that each of best particles separately guides the motion of other particles which are located at the same subpopulation in accordance with the modified velocity updating formula as mentioned above. Under this structure, each subpopulation consisting of PS/N particles tackles one assigned task with some objective function. In contrast to the general PSO with a single population, the number of tasks solved can be increased as *N* when the algorithm is performed one time as well. This is main contribution of the proposed method.

Based on the modified PSO version with multiple subpopulations, the design steps for the multitasking problem are listed below:

Step 1. Assign all variables used in the algorithm including the number of particles (population size) *PS*, number of iterations *G*, inertia weight *w* and positive constants c_1 and c_2 in (1), number of solved tasks *N* and corresponding objective function f_i for each task and $i = 1, 2, \cdots, N$.
Step 2. Generate an initial population which consists of *PS* particles from the search interval $[\theta_{\min}, \theta_{\max}]$ randomly and uniformly.
Step 3. Partition the original population into *N* subpopulations, and this implies that each subpopulation consists of PS/N particles.
Step 4. Check whether the number of iterations *G* is achieved. If yes, the algorithm stops; otherwise, Step 5 is executed.
Step 5. Evaluate the objective function f_i of each particle and enroll the individual best particle *pbest* for each particle and the best particle *gbest* for each subpopulation instead of the global best particle in the whole population, respectively.
Step 6. Execute the modified velocity updating formula of (1) for each particle, where the global best is replaced by the best particle of each subpopulation.
Step 7. Perform the position updating formula of (2) for each particle.
Step 8. Check the obtained particle position by (3)

$$\theta_{ij} = \begin{cases} \theta_{\min} & if\ \theta_{ij} < \theta_{\min} \\ \theta_{ij} & if\ \theta_{\min} \le \theta_{ij} \le \theta_{\max} \\ \theta_{\max} & if\ \theta_{ij} > \theta_{\max} \end{cases}, \text{ for } i = 1, 2, \cdots, PS \text{ and } j = 1, 2, \cdots, n. \quad (3)$$

Step 9. Go back to Step 4.

IV. SEVERAL TARGET SEARCHING EXAMPLES

In this study, the software of Borland C++ Builder 6 is utilized to implement the modified PSO algorithm on multiple target searching problems. Here, N searching targets with two dimensions are considered and represented by $\{(x_{t1}, y_{t1}), (x_{t2}, y_{t2}), \cdots, (x_{tN}, y_{tN})\}$, respectively, where (x_{ti}, y_{ti}) means the ith searching target for $i = 1, 2, \cdots, N$. Thus, to deal with the ith searching target (x_{ti}, y_{ti}), it can simply be formulated as a minimizing problem as (4)

$$\min f_i(x, y) = (x - x_{ti})^2 + (y - y_{ti})^2 . \tag{4}$$

Equation (4) also represents the ith objective function that will be solved using the proposed algorithm. For example, if there are two different points $\{(x_{t1}, y_{t1}), (x_{t2}, y_{t2})\}$ that will be solved, the number of subpopulations used in the algorithm should be given by $N = 2$ to match these two target points. In addition, the first subpopulation is to deal with the first objective function f_1 and the second subpopulation is then for another objective function f_2. After correctly executing the proposed scheme, it can be seen that the fist searching target (x_{t1}, y_{t1}) can be caught by the first-subpopulation particles and the second target (x_{t2}, y_{t2}) is by the second-subpopulation particles, separately and synchronously.

For the use of the PSO algorithm on target searching problems, let the particle be $\Theta = [\theta_1, \theta_2] = [x, y]$. In the following simulations, the search interval is constrained by $-10 \le x \le 10$ and $-10 \le y \le 10$, respectively, i.e., $[\theta_{\min}, \theta_{\max}] = [-10, 10]$ used in (3). The related variables employed in the algorithm are given by $PS = 60$, $w = 0.5$, $c_1 = 0.5$, and $c_2 = 0.5$, respectively, for any simulation cases. The four searching targets $(x_{t1}, y_{t1}) = (4, 4)$, $(x_{t2}, y_{t2}) = (-4, 4)$, $(x_{t3}, y_{t3}) = (-4, -4)$, and $(x_{t4}, y_{t4}) = (4, -4)$ are considered to demonstrate the efficiency of the proposed scheme. In the case, there are four assigned searching targets forming a square located at four quadrants, respectively, as displayed in Fig. 2, and correspondingly there are $N = 4$ subpopulations used in the developed algorithm. Each subpopulation equally contains $PS / N = 60 / 4 = 15$ particles, and particles in the fourth subpopulation are then marked by the purple for discrimination. As the above cases, Fig. 2(a) shows the initially mixed random distribution of particles. After executing some algorithm iterations, simulation results are shown in Figs. 2(b)(c)(d), respectively. These four searching targets can successfully and synchronously be solved by different subpopulation particles after about 20 iterations.

V. CONCLUSIONS

This paper has successfully proposed a modified PSO algorithm for solving the multitasking problem. In the general PSO, it only consists of a single population with a global best particle. Normally, the general algorithm can only deal with one design task, not with many tasks simultaneously. The proposed PSO algorithm, however, is to utilize multiple subpopulation structures consisting of multiple best particles which are necessarily recorded. Under the developed scheme, the number of subpopulations is equal to that of the solved tasks. Each subpopulation can separately tackle one corresponding task according to the modified velocity updating formula in which the global best particle is replaced by the best particle of each subpopulation. As a result, the proposed scheme can synchronously solve many design tasks when the algorithm is only executed one time. The searching target examples are successfully illustrated to confirm the applicability and efficiency of the proposed method.

REFERENCES

[1] W. A. Schuur, S. E. Baumgartner, S. R. Sumter, and P. M. Valkenburg, “The consequences of media multitasking for youth: a review,” Computers in Human Behavior, vol. 53, pp. 204-215, 2015.

[2] W. Ran, M. Yamamoto, and S. Xu, “Media multitasking during political news consumption: a relationship with factual and subjective political knowledge,” Computers in Human Behavior, vol. 56, pp. 352-359, 2016.

[3] B. Xu, M. Lu, Y. Ren, P. Zhu, J. Shi, and D. Cheng, “Multi-task ant system for multi-object parameter estimation and its application in cell tracking,” Applied Soft Computing, vol. 35, pp. 449-469, 2015.

[4] X. Cheng, N. Li, T. Zhou, L. Zhou, and Z. Wu, “Object tracking via collaborative multi-task learning and appearance model updating,” Applied Soft Computing, vol. 31, pp. 81-90, 2015.

[5] H. Zheng, X. Geng, D. Tao, and Z. Jin, “A multi-task model for simultaneous face identification and facial expression recognition,” Neurocomputing, vol. 171, pp. 515-523, 2016.

[6] J. Lope, D. Maravall, and Y. Quinonez, “Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems,” Neurocomputing, vol. 163, pp. 47-55, 2015.

[7] M. Li, H. Wu, H. Handroos, G. Yang, and Y. Wang, “Software protocol design: communication and control in a multi-task robot machine for ITER vacuum vessel assembly and maintenance,” Fusion Engineering and Design, vol. 98-99, pp. 1532-1537, 2015.

[8] J. Pu, J. Wang, Y. G. Jiang, and X. Xue, “Multiple task learning with flexible structure regulation,” Neurocomputing, vol. 177, pp. 242-256, 2016.

[9] S. Zhong, J. Pu, Y. G. Jiang, R. Feng, and X. Xue, “Flexible multi-task learning with latent task grouping,” Neurocomputing, vol. 189, pp. 179-188, 2016.

[10] A. A. Liu, N. Xu, Y. T. Su, H. Lin, T. Hao, and Z. X. Yang, Single/multi-view human action recognition via regularized multi-task learning,” Neurocomputing, vol. 151, pp. 544-553, 2015.

[11] J. Liao, Y. Liu, X. Zhu, and J. Wang, “Accurate sub-swarms particle swarm optimization algorithm for service composition,” The Journal of Systems and Software, vol. 90, pp. 191-203, 2014.

[12] W. D. Chang, “A modified particle swarm optimization with multiple subpopulations for multimodal function optimization problems,” Applied Soft Computing, vol. 33, pp. 170-182, 2015.

[13] T. Ma, Y. Wang, and X. Li “Convex combination multiple populations competitive swarm optimization for moving target search using UAVs,” Information Sciences, vol. 641, 119104, 2023.

[14] N. Bazurto-Gomez. C. A. Martínez-Morales, and H. E. Espitia-Cuchango, “Multiple swarm particles simulation algorithm applied to coffee berry borer proliferation,” Journal of Computational Science, vol. 48, 101263, 2021.

[15] J. Zhang and X. Ding, “A multi-swarm self-adaptive and cooperative particle swarm optimization,” Engineering Applications of Artificial Intelligence, vol. 24, pp. 958-967, 2011.

[16] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. IEEE Int. Conf. Neural Networks, vol. IV, pp. 1942-1948, Perth, Australia, 1995.

Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 14, Number 1, pages 18–21, January 2025

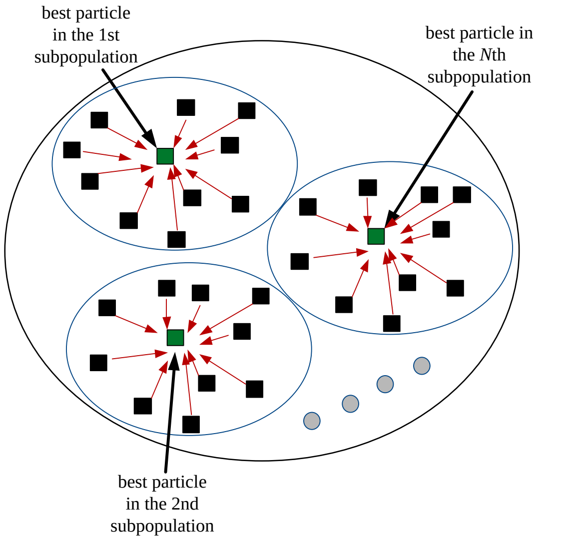

Fig. 1. An illustration of each particle's moving for the modified PSO version with *N* subpopulations.

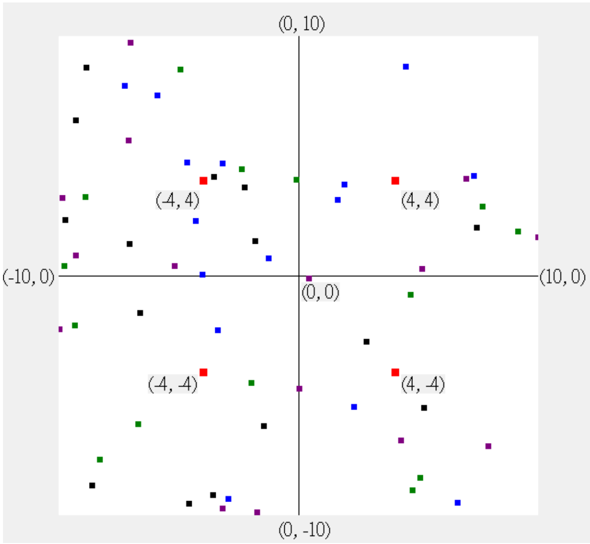

Fig. 2(a). iteration 0

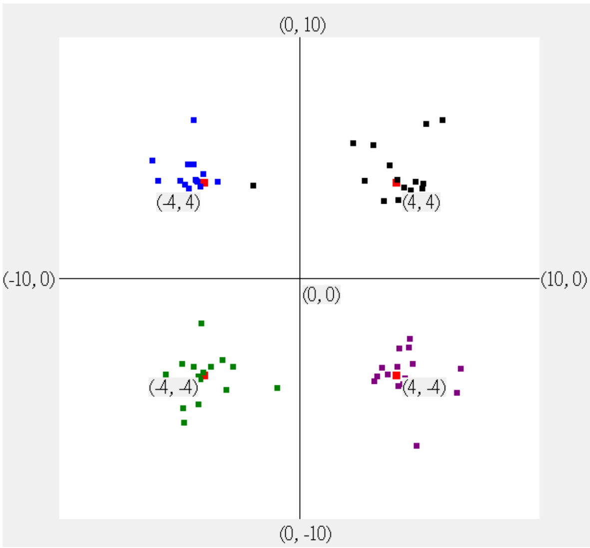

Fig. 2(b). iteration 5

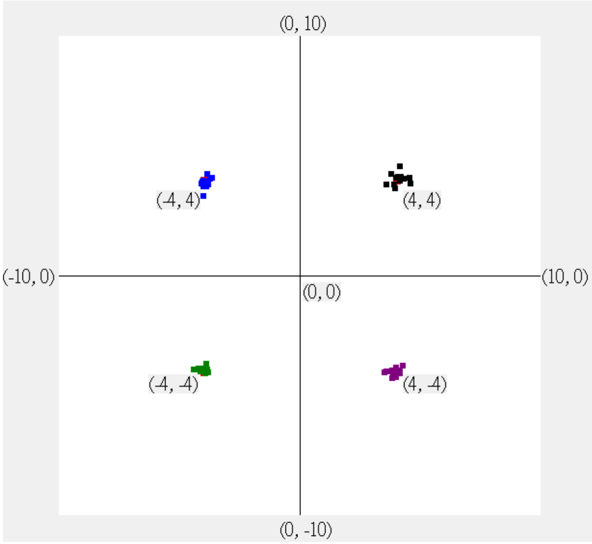

Fig. 2(c). iteration 10

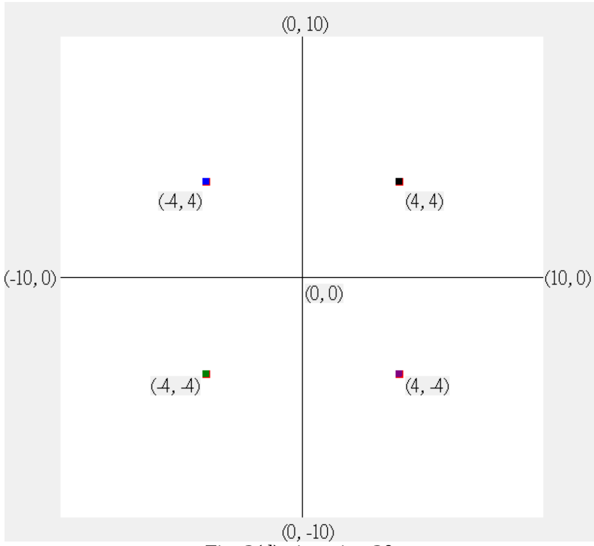

Fig. 2(d). iteration 20

Fig. 2. Simulation results.