A New Granular Particle Swarm Optimization Variant for Granular Optimization Problems

Guohua Wu
Science and Technology on Information Systems Engineering Laboratory National University of Defense Technology Changsha 410073, Hunan, China
guohuawu.nudt@gmail.com

Witold Pedrycz
Department of Electrical & Computer Engineering
University of Alberta
Edmonton, AB T6R 2V4 Canada
wpedrycz@ualberta.ca

Dishan Qiu
Science and Technology on Information Systems Engineering Laboratory National University of Defense Technology Changsha 410073, Hunan, China
ds_qiu@sina.com

Manhao Ma
Science and Technology on Information Systems Engineering Laboratory National University of Defense Technology Changsha 410073, Hunan, China
mhma@sina.com

Abstract **- As an emerging computing paradigm of information processing, Granular Computing exhibits great potential in human-centric decision problems such as feature selection and feature extraction, pattern recognition and knowledge discovery. Optimization plays an important role in these areas. The optimization problems arising in Granular Computing area are called granular optimization problems in which information granules are treated as information processing units and therefore granules denote the related solutions. Particle swarm optimization (PSO) has been demonstrated to be a very competitive algorithm in solving global optimization problems. In this paper, we develop a novel PSO variant called granular PSO to solve problems of granular optimization. Each granule in this study is expressed as a multi-dimension hyper-box with each coordinate being described by an interval. In the proposed granular PSO, the velocity and position of a particle is represented by intervals rather than single numerical values. The velocity and position update strategy is modified accordingly. In granular PSO, the solution space search behavior of a particle is realized in granule-to-granule manner rather than point-to-point format. We provide experimental simulations to demonstrate the effectiveness of the proposed granular PSO algorithm.**

Key Words: Particle swarm optimization, granular computing, granular optimization, granular PSO

I. INTRODUCTION

Granular Computing has emerged as an unified computing paradigm for constructing, describing, and processing information granules [1-3]. It concerns the processing of complex information entities called information granules, which arise in the process of data abstraction and derivation of knowledge from information. Generally speaking, information granules are collections of entities that usually originate at the numeric level and are arranged together due to their similarity, functional or physical adjacency, indistinguishability, coherency, or alike [4]. It represents an essential shift from machine-centric computing towards nature- and human-inspired computing [5].

Granular Computing has great potential in wide research and real-life application areas, such as decision making, feature selection and feature formation, pattern recognition and knowledge discovery. One of the important aspects existing in these areas is optimization. Most optimization problems are usually formulated in presence of numerical inputs and numerical outputs, and the optimization process is realized at the numeric level. However, from the perspective of Granular Computing, it could be helpful to consider information granules as processing units other than single numerical value. We call functions that map from granules to granules granular function. As a result, related granular optimization is aimed to determine an optimal granule that optimizes the objective *granular* function.

Optimization plays a primordial role in scientific research, management, industry, etc, given a fact that many problems in real world are essentially optimization problems. However, with the complexity of the optimization problems associated with multi-modality, noise and high dimensionality of problems, traditional optimization methods (such as gradient-based methods) are no longer effective to completely effective when searching for optimal or satisfactory solutions within the bounds of reasonable computation overhead. In light of these challenges, many bio-inspired algorithms, such as Genetic Algorithm (GA) and Ant Colony Optimization (ACO), have been emerged and show promising perspective in solving complicated global optimization problems. Particle swarm optimization (PSO), developed by Kennedy and Eberhart [6], is a competitive population-based algorithm being particularly efficient when dealing with continuous optimization problems. It is a swarm intelligence [7] algorithm that emulates swarm behaviours such as birds flocking and fish schooling [8]. Each particle in PSO adjusts its flying speed and direction by learning from its own past experience and its neighbours' experience, attempting to search for better position gradually [9]. Due to its powerful capability and relatively less number of parameters, PSO has drawn wide attention since its introduction. To strengthen the efficiency of the generic version of PSO, many variants have been proposed. These variants are realized through different augmentations of the generic method, generally including parameter tuning [10, 11], topology structure adjustment [12, 13], intelligent combination of various search strategy [14-16] and hybrid with other classical optimization techniques [17, 18].

Previous studies on Granular Computing are generally focused on granule representation, information granulation,

978-1-4799-0348-1/13/$31.00 ©2013 IEEE

information granule encoding and decoding, optimal granularity allocation and others. In comparison, granular optimization problems have been seldom studied. In addition, although PSO has been extensively investigated, to the best of our knowledge, the application of PSO to granular optimization is never considered up to now.

In this study, we introduce a concept of granular optimization and briefly discuss its potential applications. We then redesign a particle swarm optimization variant called granular PSO to address problems of granular optimization. In the discussed granular PSO, the velocity and position of a particle are no longer treated as numeric entities. In contrast, each coordinate of the position and velocity is described by an interval. The position of a particle is essentially a multi-dimension hyper-box, namely information granule. Therefore, the solution space search behavior is of granule-to-granule manner rather than previously point-to-point manner. The position and velocity update strategy particles is modified accordingly. We also conduct experiments to demonstrate the effectiveness of the proposed granular PSO.

II. Potential applications of granular optimization

In Granular Computing, the data processing units are granules. Accordingly, in granular optimization problems, solutions come in the form of granules as well. Generally, granules could be represented as fuzzy sets, interval analysis, shadowed sets, rough sets and probability theory. In this study, granules are described by intervals, namely each variable of a granule is represented by an interval value.

One potential application of granular optimization is related to robust optimization. In some real-world applications, variables in an optimization problem may be uncertain because of the impact of noise (or errors), so the exact single numerical value of the variable is difficult or impossible to provide. For example, the value of variable x may fluctuate above and below a value x_0 within certain range ξ, i.e., $|x - x_0| \le \xi$. Then the value of x can be expressed as an interval $x = [\underline{x}, \overline{x}] = [x_0 - \xi, x_0 + \xi]$. Therefore, to find such a robust solution to an optimization problem, the objective is not to determine an optimal numerical value but an interval for each variable. In some real applications, providing an exact input value of a variable is impractical. For example, assume that x is such an uncertain variable and the objective is to find an interval (with fixed interval length) for x to the optimize objective function $f(x)$.

As shown in Fig. 1, it is clear that the function reaches its maximum at the point included in the interval $x_i = [\underline{x_i}, \overline{x_i}]$. However, obviously, the average value of $f(x)$ over interval $[\underline{x_i}, \overline{x_i}]$ is worse than that over interval $[\underline{x_j}, \overline{x_j}]$. For robustness, we would like to choose $[\underline{x_j}, \overline{x_j}]$ as the most proper interval solution of x. Assume the size of the required granule is predefined, i.e., for each dimension, $\overline{x_{id}} - \underline{x_{id}} = c_d$, where c_d is a constant numerical value, then the objective is to find an granule i, maximizing the average fitness $\tilde{f}(x_{i1}, x_{i2}, \ldots x_{iD})$ over interval $x_i = [\underline{x_i}, \overline{x_i}]$

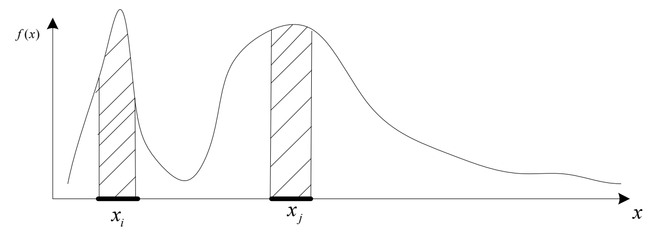


Fig. 1. Demonstration of search proper granule for a function with one dimension

Another possible application concerns the evaluation and decision that requires process information coming in the interval form (or information granule, in general). For instance, when a company searches for employees, it may claim some requirements, which would be naturally expressed by intervals instead of accurate numerical values. For instance, the age of a potential employee should be within interval [20, 35] and education level should range from bachelor degree to master degree, etc. We may then be interested in how to help companies to determine the reasonable requirements for finding potential employees. These requirements are granules.

Given a collection of factors needing to be considered in the evaluation or decision process, we can take each factor as a variable. Then the variable vector can be $x = (x_1, x_2, \ldots x_D)$, where D is the number of variables (dimensions) given by intervals. Different factors may be related. Our objective is to ascertain an interval value for each factor to maximize the overall objective function $f(x) = f(x_1, x_2, \ldots x_D)$. Sometimes since the decision is human-centric, we may need to use the fuzzy sets theory. For example, there are m families of fuzzy sets, $\{A^1{}_1, \ldots A^1{}_{n_1}\}$, $\{A^2{}_1, \ldots A^2{}_{n_2}\}$, $\{A^m{}_1, \ldots A^m{}_{n_m}\}$, where n_i denotes the number of fuzzy sets in the i th ($i = 1, \ldots m$) family. The objective function can be formulated as $f(A_i^j(x_d) \mid 0 < j < m, 0 < i < n_j, 0 < d < D)$.

III. Granular PSO

Typically, each particle in the "standard" PSO attempts to search better solutions by moving from one point to another point in a continuous space, as illustrated in Fig. 2. In comparison, to effectively search required optimal granules, the position of each particle is viewed as a granule with certain level of granularity (i.e. the length of the interval). Then, unlike the typical way, each particle in PSO will search the solution space with the new manner of from one granule to another granule, as shown in Fig. 3.

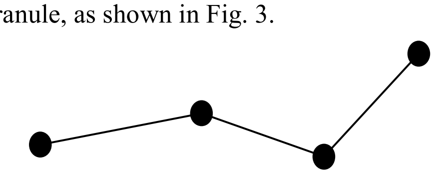


Fig. 2 The typical space search manner of point-to-point

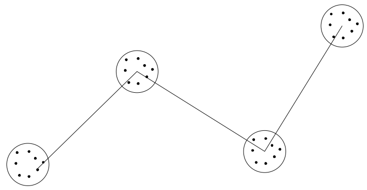


Fig. 3 The space search manner of granule-to-granule

A. *Calculation the interval function fitness*

For a particle i with D variables, let x_i describe its granular position. Then we have $x_i = [\underline{x_i}, \overline{x_i}]$, $x_i = (x_{i1}, x_{i2}, \ldots x_{iD})$ and the d th dimension of the position is a numerical interval $x_{id} = [\underline{x_{id}}, \overline{x_{id}}]$. Then we have $\underline{x_i} = (\underline{x_{i1}}, \underline{x_{i2}}, \ldots \underline{x_{iD}})$ and $\overline{x_i} = (\overline{x_{i1}}, \overline{x_{i2}}, \ldots \overline{x_{iD}})$.

One form of the granular function we intend to investigate in this study is the function that realizes mapping from granules to intervals. That is to say, over a granule variable, the function output is an interval. It is difficult or usually quite impossible to calculate the exact bounds of the interval function fitness if the objective optimization function is complex. A simple and straightforward idea is to first randomly sample a certain number of points contained the granule; then calculate the interval function fitness based on these sampled points. Therefore, through the sampling strategy, we can obtain a collection of sampled points, denoted by $S = \{y_j \mid j = 1, 2, \ldots m\}$, where m stands for the number of the samples. Afterwards, we use a numeric interval to represent the function fitness $f(x_i)$ of granule i:

$$\begin{aligned} f(x_i) &= [\min\{f(y_j) \mid j = 1, \ldots m\}, \max\{f(y_j) \mid j = 1, \ldots m\}] \\ &= [\underline{f}(x_i), \overline{f}(x_i)] \end{aligned} \quad (1)$$

We would like the interval function fitness to exhibit the minimum average value as well as shortest length. Therefore, the optimization objective for such granular optimization functions is transformed into:

$$\min\ 0.5(\underline{f}(x_i) + \overline{f}(x_i))(|\overline{f}(x_i) - \underline{f}(x_i)| + 1) \quad (2)$$

Another kind of granular optimization function of interest is the function mapping from granules to numerical value. In essence, such functions take the attributes and parameters of granules as dependent variables and output numerical value. So the function fitness over each granule is numerical and can be computed directly. In the experimental simulation section, we will apply the proposed granular PSO to these two kinds of granular optimization functions, respectively.

B. *The implementation of the granular PSO*

Although many PSO variants have been proposed to improve the performance of the basic version of PSO, at this stage, we develop the granular PSO just based on the modification of the basic PSO [19]. Each coordinate of the velocity and position of a particle is represented by an interval.

Algorithm 1: The procedure of granular PSO

Initialize the position all particles $X = \{x_1, x_2, \ldots, x_{ps}\}$;
Evaluate the fitness value $F = \{f_1, f_2, \ldots, f_{ps}\}$ for all particles;
Set X to be $pbest = \{pbest_1, pbest_2, \ldots, pbest_{ps}\}$ for each particle;
Set the x_i with the best fitness value to be $gbest$;
Initialize $t = 0$, ps, $maxGen$
While ($t < maxGen$)
  For i =1: ps
  // Velocity and position update according to (3),(4), (5) and (6)
    $\underline{v_{i+1,d}} = w\times\underline{v_{id}} + c_1\times r_1\times(\underline{pbest_{id}} - \underline{x_{id}}) + c_2\times r_2\times(\underline{gbest_d} - \underline{x_{id}})$;
    $\overline{v_{i+1,d}} = w\times\overline{v_{id}} + c_1\times r_1\times(\overline{pbest_{id}} - \overline{x_{id}}) + c_2\times r_2\times(\overline{gbest_d} - \overline{x_{id}})$;
    $\underline{x_{i+1,d}} = \underline{x_{id}} + \underline{v_{i+1,d}}$;
    $\overline{x_{i+1,d}} = \overline{x_{id}} + \overline{v_{i+1,d}}$;
    Evaluate the fitness f_i particle i in terms Section 3.1;
    If x_i is superior to $pbest_i$
      Set x_i to be $pbest_i$;
    End if
    If x_i is superior to $gbest$
      Set x_i to be $gbest$;
    End if
  End for
  Set $t = t + 1$;
  Tune the inertia weight $w = 0.9 - 0.5\cdot t/maxGen$;
End while

As a result, the velocity v_{id} and position x_{id} of the d th coordinate of the i th particle are updated as follows:

$$\overline{v_{i+1,d}} = w\times\overline{v_{id}} + c_1\times r_1\times(\overline{pbest_{id}} - \overline{x_{id}}) + c_2\times r_2\times(\overline{gbest_d} - \overline{x_{id}}) \quad (3)$$

$$\underline{v_{i+1,d}} = w\times\underline{v_{id}} + c_1\times r_1\times(\underline{pbest_{id}} - \underline{x_{id}}) + c_2\times r_2\times(\underline{gbest_d} - \underline{x_{id}}) \quad (4)$$

$$\overline{x_{i+1,d}} = \overline{x_{id}} + \overline{v_{i+1,d}} \quad (5)$$

$$\underline{x_{i+1,d}} = \underline{x_{id}} + \underline{v_{i+1,d}} \quad (6)$$

where $v_{id} = [\underline{v_{i+1,d}}, \overline{v_{i+1,d}}]$ stands for the velocity of particle i towards to the d th dimension; $x_{id} = [\underline{x_{id}}, \overline{x_{id}}]$ stands for the d th dimension of the position of particle i; $pbest_{id} = [\underline{pbest_{id}}, \overline{pbest_{id}}]$ represents the d th dimension of the best position in the search space ever visited by particle i; and $gbest_d = [\underline{gbest_d}, \overline{gbest_d}]$ denotes the d th dimension of the best position discovered by particles so far. Above all, we provide the procedure of the granular PSO in **Algorithm 1**.

IV. EXPERIMENTAL STUDIES

To test the performance of the granular PSO, we applied it to some granular optimization functions which are derived or modified from some classical benchmark numerical optimization functions. In our tests, parameters of the granular PSO is set to $ps = 20$ and $maxGen = 8000$. The number of samples for calculating the interval fitness of the granular mapping is set to 20. We test the granular PSO algorithm on two types of granular functions. The first one is the functions that map from granules to intervals, and the second is the functions that map from granules to numerical values.

A. Granular optimization of mappings from granules to intervals

We form this kind of functions from some classical optimization functions commonly used as test benchmarks in the literature. The difference is that the variables in the granular optimization functions in our test are no longer numerical variables but interval variables. The interval fitness value of the functions is computed by the sampling method. The selected functions are listed as below. Note that the interval search range of each dimension is restricted in [-20, 20]. The number of variables (dimensions) of each function is set to 10.

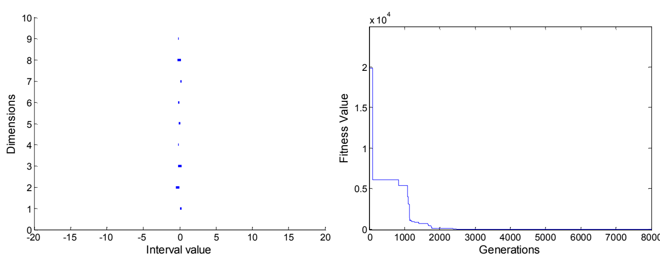
Granular Sphere: $f_1(x) = \sum_{i=1}^{n} x_i^2$

Granular Rastrigin: $f_2(\mathbf{x}) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$

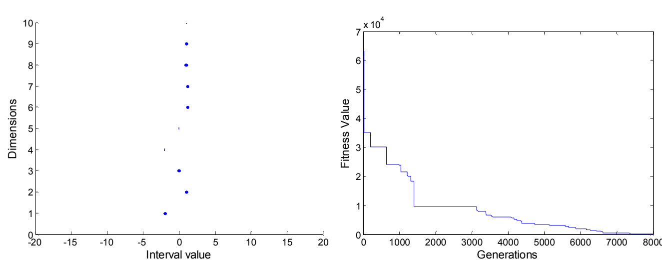
Granular Rosenbrock : $f_3(\mathbf{x}) = \sum_{i=1}^{n} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$

Granular Griewank: $f_4(\mathbf{z}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 + 1 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$

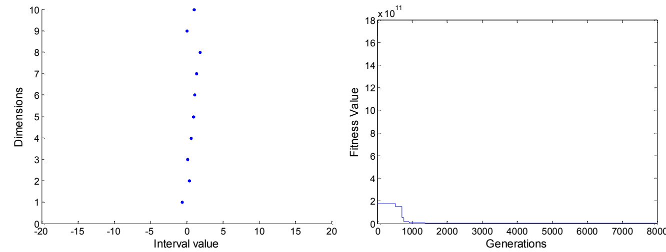
The Corresponding computational results obtained by running the granular PSO for one time are displayed in Fig. 4. The left sub figures demonstrate the final interval value of each variable (dimension). And the right sub figures show the function fitness evolutionary process. From the results, we can observe that each variable are converged to a very short interval or even some are converged to a point. The interval value of each variable is basically distributed closely around zero which is the theoretical optimal value of each variable. In addition, we can find that, from the perspective of the difference between the obtained experimental interval value and the theoretical optimal interval value of each variable, granular PSO performs better on Granular Sphere and Granular Griewank functions than on Granular Rastrigin and Granular Rosenbrock functions. The granular PSO algorithm yields good convergence performance for each function.



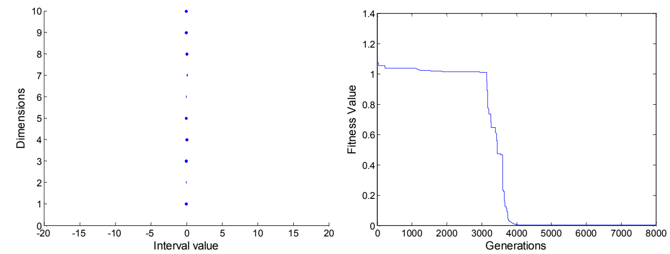
(a) Granular Sphere function



(b) Granular Rastrigin function



(c) Granular Rosenbrock function



(d) Granular Griewank function

Fig. 4 Interval value of each dimension and the fitness evolution with regard to each Granular functions

B. granular optimization of mappings from granules to numerical values

We design this type of granular functions through the modification of some classical optimization functions, meaning that we add the parameters and attributes of granules as factors (independent variables) into the objective granular optimization functions. The modified functions are listed below. Note that the interval search range of each variable is restricted in [-20, 20] as well. The dimensionality of each test function is set to 30. Theoretically, the optimal interval of each dimension in terms of the function fitness should be [0, 1].

Modified Granular Sphere: $f_5(x) = \sum_{i=1}^{n} ((\overline{x_i} - \underline{x_i} - 1)^2 + \underline{x_i}^2)$

Modified Granular Rastrigin:

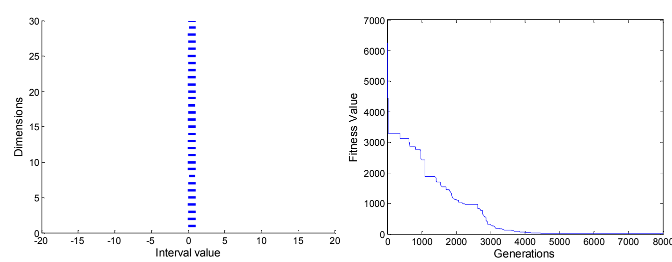
$$f_6(\mathbf{x}) = \sum_{i=1}^{n} ((\overline{x_i} - \underline{x_i} - 1)^2 - 10\cos(2\pi(\overline{x_i} - 1)) - 10\cos(2\pi \underline{x_i}) + 20 + (\overline{x_i} - 1)^2 + \underline{x_i}^2)$$

Modified Granular Rosenbrock :

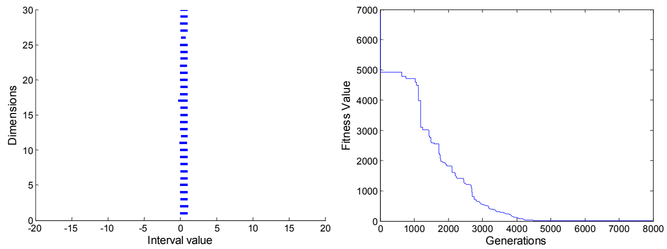
$$f_7(\mathbf{x}) = \sum_{i=1}^{n} (100((\overline{x_i} - \underline{x_i} - 1)^2 - \underline{x_{i+1}})^2 + (\overline{x_i} - 1)^2)$$

Modified Granular Griewank:

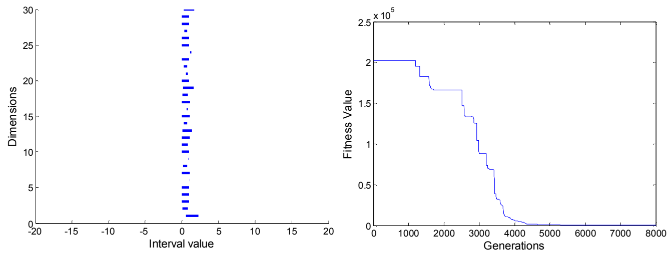
$$f_8(\mathbf{z}) = \frac{1}{4000}\sum_{i=1}^{n} ((\overline{x_i} - 1)^2 + \underline{x_i}^2) + 1 - \prod_{i=1}^{n} \cos(\frac{\overline{x_i} - \underline{x_i} - 1}{\sqrt{i}})$$



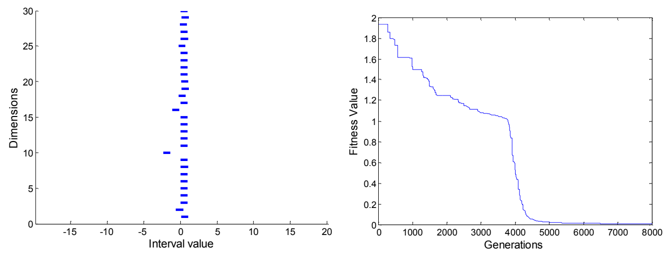
(a) Granular Sphere function



(b) Modified Granular Rastrigin function



(c) Modified Granular Rosenbrock function



(d) Modified Granular Griewank function

Fig. 5 Interval value of each dimension and the fitness evolution with regard to each modified Granular functions

Fig. 5 demonstrates results of the granular optimization.. Granular PSO obtains the optimal solution for the Modified Granular Sphere. It has generated suboptimal solutions extremely close to the optimal solution for the Modified Granular Rastrigin function. In comparison, the obtained results of the Modified Granular Rosenbrock function and the Modified Granular Griewank function are relatively worse, however they are still fairly good as being close to the theoretical optimal interval value. The granular PSO algorithm exhibits good performance on all granular optimization functions.

V. Conclusions

In this paper, we have introduced the concept of granular optimization, which is based on the fact that in many real-world optimization problems (e.g., human-centric decision problems), the naturally required solution is not a numerical value but an information granule. Aiming to solve the granular optimization problems, we develop a novel PSO variant called granular PSO which is designed through the modification of the basic version of PSO. In the granular PSO, the position and velocity of each particle are represented in the form of information granules (i.e., intervals) and the search behavior of each particle is realized in the granule-to-granule fashion rather than typical point-to-point manner. The granular PSO algorithm takes advantage of the strong optimization performance of the PSO itself and the essential features of Granular Computing. Experimental studies completed for several granular optimization functions demonstrated the effectiveness of granular PSO.

Acknowledgment

This work was supported by the National Nature Science Foundation of China under Grant No. 71271213. The Author Guohua Wu is supported by the China Scholarship Council under Grant No.201206110082.

References

[1] W. Pedrycz, "Granular computing-the emerging paradigm," *J. Uncertain Syst.,* vol. 1, pp. 38-61, 2007.

[2] W. Pedrycz, A. Skowron, and V. Kreinovich, *Handbook of granular computing*: Wiley-Interscience, 2008.

[3] W. Pedrycz, "Allocation of information granularity in optimization and decision-making models: towards building the foundations of granular computing," *Eur. J. Oper. Res.,* 2012.

[4] W. Pedrycz, "Granular computing: an introduction," in *Joint the 9th the IFSA World Congress and 20th NAFIPS International Conference*, 2001, pp. 1349-1354.

[5] W. Pedrycz, "Concepts and Design of Granular Models: Emerging Constructs of Computational Intelligence," in *Computational Intelligence*, ed: Springer, 2013, pp. 15-29.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.

[7] R. Eberhart, Y. Shi, and J. Kennedy, *Swarm intelligence*: Morgan Kaufmann, 2001.

[8] Z. H. Zhan, J. Zhang, Y. Li, and Y. H. Shi, "Orthogonal learning particle swarm optimization," *Ieee. T. Evolut. Comput.,* vol. 15, pp. 832-847, 2011.

[9] R. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Evolutionary Programming VII*, 1998, pp. 611-616.

[10] A. Ismail and A. Engelbrecht, "The Self-adaptive Comprehensive Learning Particle Swarm Optimizer," *Swarm Intell.,* pp. 156-167, 2012.

[11] K. Parsopoulos and M. Vrahatis, "Parameter selection and adaptation in unified particle swarm optimization," *Math. Comput. Model.,* vol. 46, pp. 198-213, 2007.

[12] J. Liang and P. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proceedings of the 2005 IEEE Swarm Intelligence Symposium*, 2005, pp. 124-129.

[13] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the 2002 Congress on Evolutionary Computation*, 2002, pp. 1671-1676.

[14] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimization," *Ieee. T. Syst. Man. Cy. B,* vol. 39, pp. 1362-1381, 2009.

[15] M. Hu, T. Wu, and J. D. Weir, "An intelligent augmentation of particle swarm optimization with multiple adaptive methods," *Information Sciences,* vol. 213, pp. 68-83, 2012.

[16] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *Ieee. T. Syst. Man. Cy. B,* vol. 42, pp. 627-646, 2012.

[17] C. Wei, Z. He, Y. Zhang, and W. Pei, "Swarm directions embedded in fast evolutionary programming," in *Proceedings of the 2002 Congress on Evolutionary Computation*, 2002, pp. 1278-1283.

[18] R. Poli, C. Di Chio, and W. B. Langdon, "Exploring extended particle swarms: a genetic programming approach," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*, 2005, pp. 169-176.

[19] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *The 1998 IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, 1998, pp. 69-73.