Journal of AI and Data Mining
Vol 4, No 1, 2016, 67-76 ***10.5829/idosi.JAIDM.2016.04.01.08***

PSO for multi-objective problems: Criteria for leader selection and uniformity distribution

H. Motameni

Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran.

Received 03 December 2014; Accepted 12 October 2015
**Corresponding author: motameni@iausari.ac.ir (H. Motameni).*

Abstract
This paper proposes a method to solve multi-objective problems using improved Particle Swarm Optimization. We propose leader particles which guide other particles inside the problem domain. Two techniques are suggested for selection and deletion of such particles to improve the optimal solutions. The first one is based on the mean of the *m* optimal particles and the second one is based on appointing a leader particle for any *n* founded particles. We used an intensity criterion to delete the particles in both techniques. The proposed techniques were evaluated based on three standard tests in multi-objective evolutionary optimization problems. The evaluation criterion in this paper is the number of particles in the optimal-Pareto set, error, and uniformity. The results show that the proposed method searches more number of optimal particles with higher intensity and less error in comparison with basic MOPSO and SIGMA and CMPSO and NSGA-II and microGA and PAES and can be used as proper techniques to solve multi-objective optimization problems.

Keywords: *Multi-objective Optimization, Particle Swarm Optimization, Intensity Distance, Mutation.*

1. Introduction

Optimization means finding one or more solution regarding one or more objectives. One multi-objective problem has more than one objective function which has to be minimized or maximized. The minimization and maximization of functions have a broad usage in scientific research as well as business applications. Multi-objective optimization comes from the real world decision making problems in which one should decide to select a set of solutions rather than a solution. For a set of finite solutions we can have a set of solutions in which two selected solutions have priority over each others. In other words, the solutions of this set are far better than other solutions. This is also called the optimal-Pareto set. In fact, there is one optimal solution in the problem domain but the first set of optimal solutions are selected. Then, user can select the optimal solution among the given best solution [1]. Multi-objective algorithms cannot find the best solutions themselves. Thus, a good technique could be the combination of such algorithms with PSO to find better solutions [2]. PSO is a population based stochastic optimization technique developed by Eberhart and Kennedy, 1995 [3], and inspired by social behavior of bird flocking or fish schooling. The system is initialized with a population of random solutions and searches for optimality by updating generations. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space associated with the best solution (fitness) achieved so far. (The fitness value is also stored.) Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. When a particle takes all the population as its topological neighbors, the best value is a global best. The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its personal best and local best locations (local version of PSO). Acceleration is weighted by a random term, with separate random

numbers being generated for acceleration toward personal best and local best locations [4].

2. Standard PSO algorithm

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest[5,6,32]. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest.

In order to establish a common terminology, in the following we provide some definitions of several technical terms commonly used:

Swarm: Population of the algorithm.

Particle: Member (individual) of the swarm. Each particle represents a potential solution to the problem being solved. The position of a particle is determined by the solution it currently represents.

pbest (personal best): Personal best position of a given particle, so far. That is, the position of the particle that has provided the greatest success (measured in terms of na scalar value analogous to the fitness adopted in evolutionary nalgorithms).

lbest (local best): Position of the best particle member of the neighborhood of a given particle.

gbest (global best): Position of the best particle of the entire swarm.

Leader: Particle that is used to guide another particle towards better regions of the search space.

After finding the two best values, the particle updates its velocity and positions with following equations (1) and (2). PSO, includes parallel search algorithms based on population, which with a group of random answers (particles) start, then the optimal solutions of the problem space by date particle location in the search continues. Each particle Multidimensional) depending on the problem (with the two vectors x_{id} and v_{id} represent the location and velocity of the i particle dimension d are to be determined. At each stage of the movement is the, location of each particle of the two values best on the day.

The first value, which is the best experience ever gotten particle by showing p_best the second value is the best experience of all particles obtained by. g_best shown [5,6]. In each iteration, the algorithm after finding two values, the new particle velocity and position according to (1) and (2) is updated.

$$v_{id}(t+1) = wv_{id}t + c_1.rand(p_{best_{id}(t)} - x_{id}(t)) \quad (1)$$
$$+c_2.rand(g_{best_{id}(t)} - x_{id}(t))$$

$$x_{id} = (t+1)x_{id}(t) + v_{id}(t+1) \quad (2)$$

In (1), W is a linear coefficient of inertia reduced, and is usually in the range [0,1], respectively and from which we in this paper considered equal to 0.2. C1 and C2 are coefficients of learning or acceleration in the interval [0,2] is selected and in most cases for g_best , and the second and third equation (1) will be zero. The particle motion in the previous ones will be moving. This is because, typically W has both the 1.49 level and 2 Use [6,7,8]. We in this paper considered c1 and c2 equal to 1 and 2 [15].

The right side of (1) is composed of three parts:, The first part of a multiple current speed of the particle is, the second part of the third rotation of a particle to the personal experience and basic variety rotate a bit to the experience is the best [2].w, seeking to establish a balance between local and global, for the first time in [9] proposed the specifies motion coefficient global search. In many cases, this leads to premature convergence and the algorithm will be a local optimum. To resolve this problem first in 2002, a new algorithm was presented with the name GCPSO [10] and in this method, a new parameter has been added to the algorithm. The question is if the answer queries about the random particle g_best optimized. The main problem for solving multi-objective optimization using PSO update equation is the speed, because it makes that all particles converge on one point to get a result of each run. To solve this problem, noted in the previous position, the new position stores them. The result of the selection of the initial population is doubled. The method is intended for particle and is selected from among Old and new particles in the initial population as regular and the other one is defeated [2]. To overcome the problem algorithm of premature convergence to a local optimum, in 2002, a new algorithm called GCPSO was presented [10] and in this method, a new parameter was added to the algorithm that would be random searches particle g_best around the optimal solution.

3. Related works

Multi-objective optimization was done in research on transportation planning in which the proposed problem was resource distribution of products [11]. MOPSO was applied to solve the problem.

The problem was divided into sub-solutions and objective functions were described based on variable dependencies. The result showed the robustness and flexibility of the research.

A new technique was proposed in [12] to solve redundancy and reliability in which three functions were used simultaneously; objective function, cost function, and dynamic penalty function. The dynamic function controlled objective and cost function throughout iterations.

Deb et al proposed an elitist non dominated sorting genetic algorithm for multi-objective optimization called NSGA-II. NSGA-II used elitism. In different dimensions in contrast with NSGA, NSGA-II worked by composing parent and offspring population and creating Rt. Then, Rt was classified using elitism. If in each class, the population is less than N (population size), all of the class members are chosen for next generation. The rest of the solutions are selected from other elite classes. On the other hand, if the population is greater than N, the better solutions for next generation are chosen based on intensity operator. The intensity is calculated using congestion procedure in objective function space and it can also be calculated in parameter space too. The constraint of the proposed technique was that intensity could offend convergence inside the algorithm in some cases [12]. Both intensity and convergence were targeted in [13] with introducing a new algorithm called MOGA. MOGA can also be used for composite optimization problems. The objective function procedure never guaranteed that a solution with lower rank always has a better scaled fitting F specially when there are solutions with better ranks and higher congestion. This can also offend convergence. However, in NSGA, the mentioned deficiency does not exist since it used queues. Share σ is a parameter that should be initialized at the beginning of algorithm like other GAs.

Deb and his students in an elite category or sorting Non-dominated genetic algorithms) so-called (NSGA_II offered.

NSGA_II of one of the most common methods is EMO multiple Pareto optimal solutions for multi-objective optimization problem acquires [14].

Multi-objective optimization using PSO was used in multi-objective handling system in [15]. The problem had three objectives:

- Minimization of the algorithm's produced pareto in comparison with the main pareto.
- Maximization of the founded solutions' distribution which makes uniform distribution achievable.
- Maximization of solutions in the optimal-Pareto.

Firstly, this algorithm works by initializing the parameters. Then, better solutions are both identified and archived. Next, for each particle a leader is selected from the archive and that particle should move toward its leader. The intensity of particles is small throughout the search space in this algorithm. Likewise in [15] an MOPSO optimizer was introduced for integrated low-carbon distribution system for the demand side of a product distribution supply chain. The proposed MOPSO selected bad solutions for deletion. The optimization occurred based on priority, ranking, and scenario analysis. The optimization of CO2 production and its relevant cost had been targeted in this research.

A new algorithm was proposed in [16] in which particle swarm optimization has been used. Particles produced offsprings in order to apply comparison in optimal-Pareto set. The problem in basic PSO is that the optimal-Pareto set comparison is not originally done in updating the best particle form each particle. To overcome this problem and to increase the sharing level among particles in a group NSPSO composes the entire best particles population (N) with their offsprings (N) and creates a temporary population of size 2N. Then the comparison procedure starts within the entire 2N particles. To do so, the entire population should be sorted in different optimal-Pareto sets as NSGA-II. A special rank is assigned to each particle in accordance with the optimal-Pareto set the particle belongs and receives the particles in the first Pareto the fitness priority of 1 and those in the second Pareto receives the fitness priority of 2 and likewise particles' fitness in each Pareto should have been calculated. In addition to fitness priority, a cumulative distance for each particle must have been calculated as well to guarantee the distribution of optimal particles. Cumulative distance is also used to evaluate the distance of each particle with its neighbours.

In [17] Sigma method as a new way to find the best local guides for each particle of the population has been introduced.

In [18], the CMPSO method has been proposed which was the combination of basic PSO with cumulative distance to solve multi objective problems. Particles are kept in the archive based on the cumulative distance. If a non-optimal solution wants to enter the archive and the archive size is the predefined size, the particle with the smallest cumulative distance in the archive should be selected first and then compared with new

particle. The particle with the smaller cumulative distance is selected to be deleted from the archive. Keeping optimal particles in archive causes keeping good solutions and not missing them. Archives keep particles with greater cumulative distance and this leads to have diversity in solutions. Archive members in low density area have greater probability to be converted to optimal situations. This helps the algorithm to find the best optimal-Pareto set.

In [19], a task scheduling using multi-objective genetic algorithm with fuzzy adaptive operators for computational grids and compared with fixed rate of mutation and crossover was proposed. Fuzzy method with a more efficient solution set of values for load balancing, makespan and price.

We have to improve [19] proposed a method and with using experiments, all we show is more efficient and our method provides makespan, price and in some cases load balancing.

In [20], the scheduling job-shop with discrete solution spaces multi-objective problem is solved using an algorithm MOPSO.

In [20,21] flexible job-shop scheduling problem (FJSP), one of the classic problems of planning a multi-objective job-shop is inconsistent and contradictory, and is solved through using algorithm PSO and tabu search (TS).The computational results have proved that the proposed hybrid algorithm in [21] is an efficient and effective approach to solve the multi-objective FJSP, especially for the problems on a large scale. In [22] a method using particle swarm optimization (PSO) is proposed to reduce the communication overhead and reduces the time to complete the process and improves resource utilization of the computational grid. The representations of the position and velocity of the particles in conventional PSO is extended from the real vectors to fuzzy matrices. The proposed approach is to dynamically generate an optimal schedule so as to complete the tasks within a minimum period of time as well as utilizing the resources in an efficient way. In [23], a presented multi-objective particle swarm optimization in systems handling is that multi-objective particle swarm optimization in systems handling is stated these objectives: Minimize the Pareto fronts distance generated by the algorithm and the Pareto front, maximize the development of solutions, so that a smooth and uniform distribution maximize the number of elements found in optimal Pareto. In this algorithm, we first initialize the population and then Non-dominated members are isolated populations. Archives are stored. For each particle of the members of the leadership archive, select the particles move toward the guide. In this paper, it is proved that the algorithm optimization MOPSO algorithm Optimization NSGAII, PAES, Micro GA Better Performance and better solutions with greater density in more smoothly and with less error is generated. In [24], the integration of low-carbon distribution in EPA using the optimization MOPSO algorithm. Done integration to distribute applicants will be done in the supply chain is presented. MOPSO non-optimal is a set of solutions from the solution desirable and practical search which remove them. Optimization and prioritization, rating and analysis scenario are done. Optimized of greenhouse gases CO2 and cost optimization are concerned.

Scheduling algorithms plays an important role in grid computing, parallel distributed systems for scheduling tasks and deploys them to appropriate resources. Grid computing system has three objectives makespan, price and balance the load.

In [25] the problem of scheduling independent tasks in heterogeneous distributed systems such as grid using multi-objective optimization algorithm non-dominated density of particles is studied. This paper presents a scheduling algorithm based on multi-objective optimization offers free particles. The work optimized simultaneously two objectives makespan and circulation time.

In [26] takes advantages of genetic algorithm, brings forward a novel heuristic genetic load balancing algorithm and applied to solve grid computing load balancing problem.

In [27], the price and makespan as the main objective, regardless the load balancing by using a GA algorithm for scheduling problem modeling are proposed. In [28], the two types of GA to improve the performance of the scheduling algorithm are presented and minimize the total execution time and meet load balancing.

In [29], the balance is the net charge on the computational grid using genetic algorithms regardless of makespan or fees for network resources represented. In [30], the different load balancing strategy based on a tree representation of a network is studied. This enables conversion of any network architecture to a unique tree with a maximum of four levels. Task scheduling algorithm in [30,31] is considered only load balancing without makespan or price to users consider.

4. Proposed technique

In the proposed technique, each particle is dedicated a random amount. Particles are divided based on optimal or bad solutions archived. For

each article a leader is selected and each article should move toward that article. After movement of particles, mutation occurs. Mutation could decrease convergence [14] and this reduction is necessary if the uniform distribution should be achieved. At first, the probability of occurring mutation is high and gradually this probability is decreased. Next, the personal best of each particle should be updated. This continues iteratively and new optimal solutions are archived. Sometimes the replacement of archive elements is necessary specially when we have other candidate optimal solutions and the archive is full. This continues until stop criterion achieves otherwise the leader selection and deletion of extra leaders continue.

4.1. Leader selection

The way how the leaders are selected has great impact on the proposed technique. Thus, a new leader selection is proposed to improve a basic MOPSO algorithm. The selected leaders are the best elements in optimal-Pareto. Two ways are suggested to choose a leader for each particle.

4.1.1. Leader selection (1st approach)

In the first way, a roulette wheel is used to select the *M* optimal solutions randomly. The places with fewer elements have more probability to be selected because this increases the optimal set domain. The mean of M selected articles is appointed as a leader. This leader selection is shown in figure 1 under the assumption of M=2.
As it is shown in figure 1, particle D is the leader of particle A which is the mean of leaders B and C. In the proposed technique, more optimal solutions will be obtained for M = 3. This means that the main leader for each particle is found through calculating the average of three leaders.

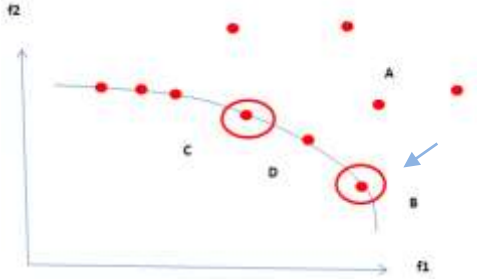

Figure 1. Leader selection (1st approach).

4.1.2. Leader selection (2nd approach)

In the second approach, a leader is selected for any of N particles in population. The first approach in leader selection is also applied in which M=3. All the particles should move toward the leader and consequently to the optimal-Pareto. Figure 2 depicted the proposed approach. Particle D is the selected leader for N particles which is the mean of leaders A and B.

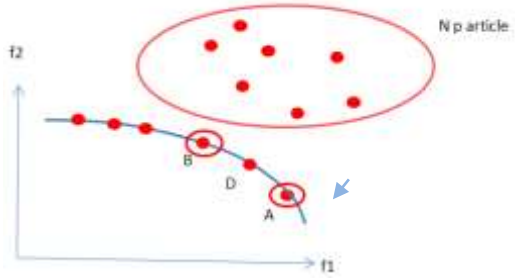

Figure 2. Leader selection (2nd approach).

4.2. Leader deletion

Some of the elements of the archive should be deleted provided that the archive queue becomes full. Through this way new optimal candidate solutions can be inserted to the archive. To delete an element uniformity criterion has been used to keep the set of optimal particles uniform. We want to delete those particles with less intensity distance. The process of leader deletion has been shown in figure 3. Intensity distance is calculated using (6). As it is shown in figure 2, particle A is deleted because of having smaller cumulative distance.

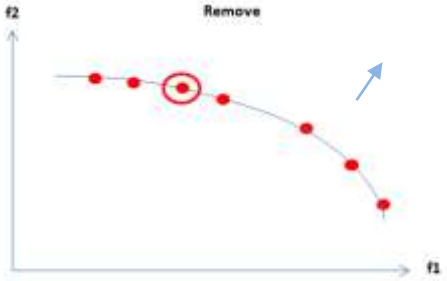

Figure 3. Leader deletion.

$$d_i^{\,2} = \min\{d_{ij} \,|\, x^j \in Q, j \neq i\} \quad (3)$$

$$d_i^{\,2} = \min\{d_{ij} \,|\, d_{ij} > d_i^{\,2}, x^j \in Q, j \neq i\} \quad (4)$$

$$d_{ij} = \sqrt{\sum_{i=1}^{M} (\mu_k (f_k(x^i) - f_k(x^j)))^2} \quad (5)$$

$$c_{iQ} = (d_i^{\,2} + d_j^{\,2})/2 \quad (6)$$

Generally, the proposed technique has been tested using four standard tests in multi-objective optimization. The evaluation criteria in this paper are: the number of elements in optimal-Pareto set, uniformity, and error. The results show that the proposed technique searches more numbers of optimal particles with higher intensity and less error in comparison with basic MOPSO. The proposed technique can be selected as a good replacement in solving multi-objective optimization problem. The proposed algorithm is shown in figure 4.

1	Initialization of the population
2	Separation and archiving optimal particles
3	Tabulating the detected objective search space
4	Leader selection for each particle from archive set and moving toward leader
5	Updating the personal best of each particle
6	◌Addition of current optimal particles to the archive
7	Deletion of non-optimal elements in archive
8	Archive elements => dedicated capacity => deletion of extra elements
9	Finish if stopping criterion satisfied otherwise goto step 3

Figure 4. Proposed algorithm.

5. Evaluation

The proposed technique has been evaluated based on three facts:

- The number of elements in optimal-Pareto set: the summation of each particle distance from nearest particle in optimal-Pareto set over the number of detected solutions as in (7). The less GD is, the more elements belonging to the optimal-Pareto set[14].

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \quad (7)$$

- Uniformity: assuming zero for this parameters means that the element of pareto set has been distributed uniformly as shown in (8)[14].

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d - d_i)^2} \quad (8)$$

$$d_i = \min(f_1^i(x) - f_1^j(x) + f_2^i(x) - f_1^j(x))$$

$$d = mean d_i$$

- Error: assuming zero for this parameter means that all produced solutions from the proposed algorithm belong to optimal-Pareto set as (9) [14].

$$ER = \frac{\sum_{i=1}^{n} e_i}{n} \quad (9)$$

5.1. First test function

We use the first test function as in [12] and (10).

$$f_1(x) = x^2 \quad (10)$$

$$f_2(x) = (x - 2^2)$$

For the first test function, the initial population was assumed to be 50, the iteration number was assumed to be 20 and the repository capacity was assumed to be 100.

In tables 1 to 4 the results of the comparison proposed technique by MOPSO and SIGMA and CMPSO is to for first test function. This table has three values of the best, worst and average values of GD, SP, are error. GD in the SIGMA less than other techniques, and it is for this reason that the number of elements found in less optimal and most repetitive elements is found. However, in the above aspects, the proposed technique is more than GD. As a result of using a less optimal number of elements can be found Sigma.

The sigma error rate in comparison to the proposed high you can conclude that fewer elements have been found to be the optimal set.

In the first part of table 1 shows that the two proposed algorithms have less GD than the basic MOPSO and SIGMA and CMPSO. This means that more detected particles belong to the optimal-Pareto set. In the second part of table 1 shows that the two proposed algorithms have less SP than the basic MOPSO. As a result the distribution is more uniform. Also In the third part of table 1 shows that the error in basic MOPSO is more than the two proposed algorithms. In the Fourth part of table 1 compares the result based on time execution.

Table 1: GD/SP/ERROR/TIME result for first function.

GD/SP/ ERROR	MOPSO	SIGMA	CMOPSO	MOPSO1	MOPSO2
Best	0.2159	0.013288	0.10925	0.1142	0.090054
Worst	0.3274	0.028923	0.62396	0.31341	0.20463
Average	0.25072	0.018854	0.19627	0.18849	0.145853
Best	0.09628	0.11698	0.12946	0.12964	0.090545
Worst	1.2454	0.8231	0.37104	0.87629	0.99389
Average	0.37498	0.364742	0.294364	0.35284	0.433475
Best	0.0826	0.1071	0.0826	0.0741	0.0741
Worst	0.1304	0.1453	0.1304	0.115	0.1071
Average	0.11082	0.12248	0.10834	0.09718	0.09566
TIME		**MOPSO1**			**MOPSO2**
Best		15.1353			4.982
worst		50.0331			38.7575
average		31.68764			25.70032

5.2. Second test function

We use the second test function as in [12] and (11).

$$f_1(x) = \sum_{i=1}^{n-1} (-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2})) \quad (11)$$

$$f_2(x) = \sum_{i=1}^{n} (|x_i|^{0.8} + 5\sin x_i^2)$$

For the second test function, the initial population was assumed to be 100, the iteration number was assumed to be 200 and the repository capacity was assumed to be 200.

In tables 5 to 8 the results of the comparison proposed technique by MOPSO and SIGMA and CMPSO is to for second test function. This table has three values of the best, worst and average values of GD, SP, are error. GD in the SIGMA less than other techniques, and it is for this reason that the number of elements found in less optimal and most repetitive elements is found. However, in the above aspects, the proposed technique is more than GD. As a result of using a less optimal number of elements can be found Sigma. The

sigma error rate in comparison to the proposed high you can conclude that fewer elements have been found to be the optimal set.

In the first part of table 2 shows that the two proposed algorithms have less GD than the basic MOPSO and SIGMA and CMPSO. This means that more detected particles belong to the optimal-Pareto set. In the second part of table 2 shows that the two proposed algorithms have less SP than the basic MOPSO. As a result the distribution is more uniform. Also In the third part of table 2 shows that the error in basic MOPSO is more than the two proposed algorithms. In the fourth part of table 2 compares the result based on time execution.

Table 2: GD/SP/ERROR/TIME result for second function.

GD/SP/ ERROR	MOPSO	SIGMA	CMOPSO	MOPSO1	MOPSO2
Best	2.03851	0.2247	2.5685	1.7619	1.8274
Worst	2.4601	0.4519	2.0385	2.6521	2.3863
Average	2.180802	0.364894	2.26366	2.02984	2.101662
Best	0.0395	0.0494	0.0339	0.0366	0.0303
Worst	0.0769	0.0962	0.1129	0.0926	0.0822
Average	0.06012	0.07812	0.05918	0.05726	0.05332
TIME		**MOPSO1**		**MOPSO2**	
Best		40.6679		29.8304	
worst		77.8366		56.6297	
average		55.24176		41.643	

In [14] MOPSO was compared with NSGA-II, microGA, and PAES and the results are shown in table 3. As it is shown in table 3, MOPSO has the smallest GD in comparison with other techniques. Tables 3 are in [14].

Table 3: GD/SP/ERROR result for third function.

GD	MOPSO	NSGA-II	microGA	PAES
Best	0.00745	0.006905	0.006803	0.01467
Worst	0.00960	0.103095	0.010344	0.157191
Average	0.008450	0.029255	0.008456	0.54914
Medium	0.00845	0.017357	0.008489	0.049358
Std. Dev.	0.000051	0.02717	0.000987	0.030744
SP	**MOPSO**	**NSGA-II**	**microGA**	**PAES**
Best	0.06187	0.018418	0.071686	0.064114
Worst	0.118445	0.065712	0.203127	0.340955
Average	0.09747	0.036136	0.128895	0.197532
Medium	0.10396	0.036085	0.126655	0.186632
Std. Dev.	0.01675	0.010977	0.029932	0.064114
ER	**MOPSO**	**NSGA-II**	**microGA**	**PAES**
Best	0.18	0.06	0.18	0.10
Worst	0.37	1.01	0.36	0.68
Average	0.2535	0.56	0.27	0.27
Medium	0.255	0.495	0.245	0.245
Std. Dev.	0.04082	0.384516	0.053947	0.10489

5.3. Third test function

We use the third test function as in [12] and (12).

$$f_1(x) = (1 - \exp(-\sum(\frac{x-1}{\sqrt{n}})^2)) \quad (12)$$

$$f_2(x) = (1 - \exp(-\sum(\frac{x+1}{\sqrt{n}})^2))$$

For the third test function, the initial population was assumed to be 100, the iteration number was assumed to be 200 and the repository capacity was assumed to be 200.

In table 4 the results of the comparison proposed technique by MOPSO and CMPSO is to for third test function. This table has three values of the best, worst and average values of GD, SP, are error. GD in the SIGMA less than other techniques, and it is for this reason that the number of elements found in less optimal and most repetitive elements is found.

However, in the above aspects, the proposed technique is more than GD. As a result of using a less optimal number of elements can be found Sigma. The sigma error rate in comparison to the proposed high you can conclude that fewer elements have been found to be the optimal set.

In the first part of table 4 shows that the two proposed algorithms have less GD than the basic MOPSO and SIGMA and CMPSO.

This means that more detected particles belong to the optimal-Pareto set. In the second part of table 4 shows that the two proposed algorithms have less SP than the basic MOPSO and SIGMA and CMPSO. As a result the distribution is more uniform. Also In the third part of table 4 shows that the error in basic MOPSO is more than the two proposed algorithms. In the fourth part of table 4 compares the result based on time execution.

Table 4: GD/SP/ERROR/TIME result for third function.

GD/SP/ ERROR	MOPSO	SIGMA	CMPSO	MOPSO1	MOPSO2
Best	0.047057	0.003502	0.046153	0.03973	0.039172
Worst	0.054195	0.016435	0.076365	0.055144	0.044375
Average	0.050508	0.008953	0.055487	0.043991	0.042329
Best	0.096128	0.10691	0.073575	0.082234	0.051319
Worst	0.11224	0.12798	0.11286	0.10116	0.090021
Average	0.104914	0.119148	0.100004	0.092765	0.068794
Best	0.0253	0.0678	0.0238	0.0229	0.0196
Worst	0.0319	0.0833	0.0476	0.0268	0.028
Average	0.02836	0.07502	0.0316	0.02506	0.0245
TIME		**MOPSO1**		**MOPSO2**	
Best		65.8227		42.8899	
worst		99.5306		85.4931	
average		756.458		73.79252	

Table 5. GD/SP/ERROR analysis on first test function.

	GD/SP/ERROR		
M	**2**	**5**	**15**
Best	0.1142	0.11619	0.08122
worst	0.31341	0.36649	0.32001
average	0.18849	0.21468	0.179382
Best	0.12964	0.11313	0.16972
worst	0.87629	0.54794	0.3692
average	0.35284	0.23272	0.243934
Best	0.0741	0.0654	0.0741
worst	0.115	0.1228	0.1525
Average	0.09718	0.08546	0.10474

Table 6. GD/SP/ERROR analysis on second test function.

	GD/SP/ERROR		
M	**2**	**5**	**15**
Best	1.7619	1.7713	1.6021
worst	2.6521	2.2464	3.398
Average	2.02984	2.06912	2.42
Best	0.02801	0.03613	0.034087
worst	0.19154	0.058174	0.29466
Average	0.07101	0.046731	0.174293
Best	0.0366	0.0375	0.0328
worst	0.0926	0.0649	0.1111
Average	0.05726	0.04274	0.0686

6. Experimental result

6.1.1. First analysis

This analysis is done to determine the best M, and to calculate the mean of M leaders for the first proposed technique.

Assuming 2, 5, and 15 for M, the results are calculated. Tables 5 to 7 show that the best assumption is M=2 that is few errors, the SP and GD.

Table 7. GD/SP/ERROR analysis on third test function.

	GD/SP/ERROR		
M	**2**	**5**	**15**
Best	0.03973	0.034591	0.035432
worst	0.055144	0.050634	0.063236
Average	0.043991	0.039307	0.048697
Best	0.082234	0.053142	0.037104
worst	0.10116	0.086248	0.057453
Average	0.092765	0.067503	0.047034
Best	0.0229	0.017	0.0152
worst	0.0268	0.0244	0.0227
Average	0.02506	0.02092	0.01962

6.1.2. Second analysis

This analysis is done to determine the best N, for the second proposed technique. This analysis also helps to determine how many particles need a certain leader to achieve better optimal-Pareto solutions. Assuming 5, 20, and 50 for N, the results are calculated. Tables 8 to 10 show that the best assumption is N=50 that is few errors, the SP and GD.

Table 8. GD/SP/ERROR analysis on the first test function.

	GD/SP/ERROR		
N	**5**	**20**	**50**
Best	0.066882	0.17949	0.090054
worst	0.25464	0.36342	0.20463
Average	0.156502	0.272078	0.145853
Best	0.16708	0.1387	0.20104
worst	1.1823	1.0292	0.99389
Average	0.409728	0.349194	0.596456
Best	0.0654	0.0741	0.0741
worst	0.1379	0.0991	0.1071
Average	0.10332	0.0842	0.09566

Table 9. GD/SP/ERROR analysis on the second test function.

	GD/SP/ERROR		
N	**5**	**20**	**50**
Best	1.7194	1.7304	1.8274
worst	2.7931	2.5637	2.3863
Average	2.094	2.05442	2.101662
Best	0.035883	0.038456	0.014392
worst	0.19398	0.20685	0.16151
Average	0.098005	0.12092	0.07814
Best	0.0274	0.0377	0.0303
worst	0.0577	0.0561	0.0822
Average	0.04244	0.04376	0.0533

6.1.3. Third analysis

This analysis is done to determine the dimensions of the problem, for both techniques. The consequences in tables 11 to 13 show that lower dimensions have better results and the proposed techniques is better in comparison with basic MOPSO.

Table 10. GD/SP/ERROR analysis on the third test function.

	GD/SP/ERROR		
N	**5**	**20**	**50**
Best	0.042976	0.03617	0.039172
worst	0.064321	0.047743	0.044375
Average	0.051529	0.043034	0.042329
Best	0.052436	0.070877	0.051319
worst	0.081611	0.075822	0.090021
Average	0.065549	0.073381	0.068794
Best	0.0167	0.0182	0.0196
worst	0.0226	0.0261	0.028
Average	0.01964	0.02348	0.0245

Table 11. GD/SP/ERROR analysis on the first proposed technique.

	GD/SP/ERROR		
NVAR	**3**	**10**	**30**
Best	1.7619	8.1567	33.367
worst	2.6521	13.6528	45.3985
Average	2.02984	10.9251	37.68592
Best	0.02801	0.020605	0.032489
worst	0.19154	0.83062	0.3059
Average	0.07101	0.231264	0.181066
Best	0.0366	0.027	0.1064
worst	0.0926	0.1892	0.1304
Average	0.05726	0.07266	0.11872

Table 12. GD/SP/ERROR analysis on the second proposed technique.

	GD/SP/ERROR		
NVAR	**3**	**10**	**30**
Best	1.8274	9.4872	26.9047
worst	2.3863	11.3522	37.5597
Average	2.101662	10.2596	33.13216
Best	0.014392	0.018486	0.050401
worst	0.16151	0.99951	0.46794
Average	0.07814	0.267326	0.256166
Best	0.0303	0.06	0.0357
worst	0.0822	0.0862	0.1304
Average	0.05332	0.06914	0.0793

As it is shown in table 12, with higher dimensions MOPSO does not work on second test function. It only works for dimensions 3 and 10.

Table 13. GD/SP/ERROR analysis on the second test function.

	GD/SP/ERROR	
NVAR	**3**	**10**
Best	2.03851	9.4417
worst	2.4601	13.7426
average	2.180802	11.78792
Best	0.045384	0.089812
worst	0.23174	1.5549
average	0.115702	0.611626
Best	0.0395	0.0588
worst	0.0769	0.119
average	0.06012	0.09424

CMPSO is not suitable for higher dimensions with second test function. It only has appropriate results for three and ten dimensions. Based on the comparisons made in table 14 even this result is not optimal.

SIGMA technique is not suitable for higher dimensions with second test function. It only has appropriate results for three and ten dimensions. Based on the comparisons made in table 15 even this result is not optimal.

Table 14: Test results for second test function.

	GD/SP/ERROR	
NVAR	**3**	**10**
Best	2.5685	12.0271
worst	2.0385	15.5735
average	2.26366	13.45778
Best	0.04189	0.016527
worst	0.13083	1.0559
average	0.073292	0.299453
Best	0.0339	0.0625
worst	0.1129	0.1429
average	0.05918	0.11088

Table 15: Test results for second test function.

	GD/SP/ERROR		
NVAR	**3**	**10**	**30**
Best	0.2247	3.6339	16.3493
worst	0.4519	8.3863	42.0749
Average	0.364894	5.32646	32.44846
Best	0.035788	0.11267	1.905
worst	0.15884	2.7703	15.3253
Average	0.083683	1.140006	6.2378
Best	0.0494	0.1875	0.1875
worst	0.0962	0.25	0.5455
Average	0.07812	0.21804	0.38548

7. Discussion

We have changed the leader selection and leader deletion in basic MOPSO to have more uniform distribution in the set of optimal-Pareto particles. The proposed approach also leads to have fewer errors and to detect more optimal particles. In the first leader selection approach, since the mean of M leader is calculated the results solutions will have more uniform optimal-Pareto particles. Likewise the result SP parameter is also smaller than the basic MOPSO and SIGMA and CMPSO. In the second leader selection approach, one leader is considered for N particles. This yields to have more solutions with uniform distribution. Also the time to achieve optimal solutions decreased. For leader deletion the proposed technique helps to increase the optimal-Pareto solutions or in other words, GD will be decreased. Finally, error was also declined.

8. Conclusion

In this paper a technique was proposed to enhance the basic MOPSO. The results prove the enhancement in comparison with the basic MOPSO and SIGMA and CMPSO and NSGA-II and microGA and PAES. In fact the proposed MOPSO has considerable results to detect optimal solutions based on the test functions. Moreover, the proposed MOPSO keeps optimal-Pareto solutions more uniform and with more intensity. As a future work we can make the algorithm and its parameters more accurate to achieve acceptable results for dynamic functions.

References

[1] Deb, K. (2014). Multi-objective optimization. In search methodologies, pp. 403–449. Springer.

[2] Coello Coello, C. A., & Lechuga, M. S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, vol. 2, pp. 1051–1056. IEEE.

[3] Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. Proceeding of the sixth international symposium on micro machine and human science, vol. 1, pp. 39–43. New York, NY.

[4] Sheibani, M. & Meybodi, M. R. (2007). PSO-LA: A new model for Optimization. 12rd Iranian Conference on Computer Engineering. pp:1162-1169, In persian.

[5] Hu, X., Shi, Y. & Eberhart, R. C. (2004). Recent advances in particle swarm. IEEE congress on evolutionary computation, vol. 1, pp. 90–97.

[6] Kennedy, J. & Mendes, R. (2002). Population structure and particle swarm performance.

[7] Cui, X., Potok, T. E. & Palathingal, P. (2005). Document clustering using particle swarm optimization. Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE, pp. 185–191. IEEE.

[8] Liu, H. & Abraham, A. (2005). Fuzzy adaptive turbulent particle swarm optimization. Hybrid Intelligent Systems, 2005. HIS '05. Fifth International Conference on. doi:10.1109/ICHIS.2005.49

[9] Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, pp. 69–73. IEEE.

[10] Van den Bergh, F. & Engelbrecht, A. P. (2002). A new locally convergent particle swarm optimizer. Proceedings of the IEEE international conference on systems, man, and cybernetics, vol. 7, pp. 6–9.

[11] Zheng, Y.-J. & Chen, S.-Y. (2013). Cooperative particle swarm optimization for multiobjective transportation planning. Applied intelligence, vol. 39, no. 1, pp. 202–216.

[12] Khalili-Damghani, K., Abtahi, A.-R. & Tavana, M. (2013). A new multi-objective particle swarm optimization method for solving reliability redundancy allocation problems. Reliability Engineering & System Safety, vol. 111, pp. 58–75.

[13] Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. Neural Networks, 1995. Proceedings., IEEE International Conference on. http://doi.org/10.1109/ICNN.1995.488968.

[14] DEH, K., Agrawal, S., PRATAP, A. & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Lecture notes in computer science, pp. 849–858.

[15] Coello, C. A. C., Pulido, G. T. & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. Evolutionary Computation, IEEE Transactions on, vol. 8, no. 3, pp. 256–279.

[16] Liu, Y. (2008). A fast and elitist multi-objective particle swarm algorithm: NSPSO. Granular Computing, 2008. GrC 2008. IEEE International Conference on, pp. 470–475.

[17] IEEE. Mostaghim, S. & Teich, J. (2003, April). Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE, pp. 26-33. IEEE.

[18] Fan, J., Zhao, L., Du, L. & Zheng, Y. (2010). Crowding-distance-based multi-objective particle swarm optimization. Computational Intelligence and Intelligent Systems, pp. 218–225. Springer.

[19] Salimi, R., Motameni, H. & Omranpour, H. (2014). Task scheduling using NSGA II with fuzzy adaptive operators for computational grids. Journal of Parallel and Distributed Computing, vol. 74, no. 5, pp. 2333-2350.

[20] Sha D. Y. & Lin, H. H. (2010). A multi-objective PSO for job-shop scheduling problems. Expert Systems with Applications, vol. 37, no. 2, pp. 1065-1070.

[21] Zhang, G., Shao, X., Li, P. & Gao, L. (2009). An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. Computers & Industrial Engineering, vol. 56, no. 4, pp. 1309-1318.

[22] Liu, H., Abraham, A. & Hassanien, A. E. (2010). Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm. Future Generation Computer Systems, vol. 26, no. 8, pp. 1336-1343.

[23] Carlos A, Coello, Coello, Gregorio Toscano Pulido, M, Salazar Lechuga., (2004). Handling multiple objectives with paper swarm optimization, Evolutionary Computation. IEEE Transactions on, pp. 256-279.

[24] Validi, S., Bhattacharya, A., & Byrne, P. J. (2014). Integrated low-carbon distribution system for the demand side of a product distribution supply chain: a DoE-guided MOPSO optimiser-based solution method. International Journal of Production Research, vol. 52, no. 10, pp. 3074-3096.

[25] Subashini, G., Bhuvaneswari, M. C. (2011). Non dominated particle swarm optimization for scheduling independent tasks on heterogeneous distributed environments, vol. 3, no. 1, pp. 1-17.

[26] J. Ma, Lanzhou. (2010). A novel heuristic genetic load balancing algorithm in grid computing, in: Second International Conference on Intelligent Human–Machine Systems and Cybernetics, pp. 166-169

[27] Kardani-Moghaddam, S., Khodadadi, F., Entezari-Maleki, R. & Movaghar, A. (2012). A hybrid genetic algorithm and variable neighborhood search for task scheduling problem in grid environment. Procedia Engineering, vol. 29, pp. 3808-3814.

[28] Omara, F. A., Arafa, M. M. (2010). Genetic algorithms for task scheduling problem, J. Parallel Distrib. Comput, vol. 70, no. 1, pp. 13–22.

[29] Prakash, S., Vidyarthi, D. P. (2011). Load balancing in computational grid using genetic algorithm, Advances in Computing. vol. 1, no. 1, pp. 8–17. http://dx.doi.org/10.5923/j.ac.02.

[30] Patni, J. C., Aswal, M. S., Pal, O. P. & Gupta, A. (2011). Load balancing strategies for grid computing. In Electronics Computer Technology (ICECT), 3rd International Conference on, vol. 3, pp. 239-243. IEEE.

[31] Peng, H., Li, Q. (2011). One kind of improved load balancing algorithm in grid computing, in: International Conference on Network Computing and Information Security, vol. 1, pp. 347-351. IEEE.

[32] Izakian, Z., Mesgari, M. S. (2015). Fuzzy clustering of time series data: A particle swarm optimization approach. Journal of Artificial Intelligence and Data Mining, vol. 3, no. 1, pp. 39–46. doi:10.5829/idosi.JAIDM.2015.03.01.05.

نشریه هوش مصنوعی و داده کاوی

دانشگاه شاهرود

الگوریتم بهینه سازی گروه ذرات برای مسائل چند هدفه: معیاری برای انتخاب راهنما و توزیع یکنواخت

همایون موتمنی

گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد ساری، ساری، ایران.

تاریخ دریافت: ۲۰۱۴/۱۲/۰۳؛ تاریخ پذیرش: ۲۰۱۵/۱۰/۱۲

چکیده:

در این مقاله روشی برای حل مسائل چند هدفه با استفاده از بهبود الگوریتم بهینه سازی گروه ذرات پیشنهاد شده است، که در آن ذرات راهنما برای هدایت ذرات موجود در دامنه مسئله بکار گرفته شده است. دو روش برای انتخاب و حذف ذرات برای بهبود راه حل های بهینه پیشنهاد شده است. در راه اول انتخاب راهنما بر اساس میانگین m ذره بهینه و در راه دوم انتخاب یک راهنما برای هر n ذرات صورت می‌گیرد. یک معیار تراکم برای حذف ذرات در هر دو روش استفاده شده است. روش ارائه شده بر اساس سه آزمون استاندارد در مسائل بهینه سازی تکاملی چند هدفه مورد ارزیابی قرار گرفت. معیارها و پارامترهای ارزیابی شده در این مقاله، تعداد عناصر موجود در مجموعه بهینه پارتو، یکنواختی و خطا می‌باشد. نتایج نشان می‌دهد که روش ارائه شده تعداد ذرات بهینه بیشتری را به درستی و با تراکم بالا و خطای کمتر در مقایسه با روشPSO پایه و CMPSO و SIGMA و NSGA-II و microGA و PAES جستجو می‌کند و می‌توان از این روش به عنوان یک روش مناسب برای حل مسائل بهینه سازی چند هدفه استفاده نمود.

کلمات کلیدی: بهینه سازی چند هدفه، بهینه سازی گروه ذرات، فاصله تراکم، جهش.