Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 8, No. 1, February 2017), Pages: 1-16
www.jacr.iausari.ac.ir

Task Scheduling Using Particle Swarm Optimization Algorithm with a Selection Guide and a Measure of Uniformity for Computational Grids

Mona Torabi✉
College of Computer Science, Tabari University of Babol, Iran
monatorabi@tabari.ac.ir

Received: 2015/08/27; Accepted: 2015/11/09

Abstract

In this paper, we proposed an algorithm for solving the problem of task scheduling using particle swarm optimization algorithm, with changes in the Selection and removing the guide and also using the technique to get away from the bad, to move away from local extreme and diversity. Scheduling algorithms play an important role in grid computing, parallel tasks Scheduling and sending them to appropriate resources. The proposed method has less Makespan and price. In addition to implementing a grid computing system, the proposed method which is using three standard test functions in evolutionary multi-objective optimization is evaluated. In this paper, the number of elements in the assessment of the Pareto optimizes set, uniformity and error. The results show that this Search method has more optimization in particle number density and high accuracy with less error than the MOPSO and can be replaced as an effective solution for solving multi-objective optimization.

Keywords: *Task scheduling, load balancing, multi-objective optimization, particle swarm optimization, guide select, guide remove, Distance density*

1. Introduction

Grid computing is a new computing environment that is as a leading technology in the field of parallel computing and is distributed to appear. [3,4] The idea of grid computing, was born in the early 90's and its aim is to use idle computing resources around the world. It was great for science and research.

The economic aspects of this technology has been done and it moves towards global business. Marketing based model is one of the most important existing models in the field. In the model resource have priced determine and proposed from their owners or based on supply, demand and value system Based on economic and grid system.

Load balancing discusses how to refers and divide resources. In this case there is no any extra overhead with resources. In addition to load balancing and price, in the market-based grid computing, task completion time is very important. Therefore, a proper timing, must offer for the lowest price to get things done in the possible time, with the highest load balancing [5].

In this paper, we use the particle swarm optimization algorithm (MOPSO) with changes in the Selection and removing the guide and also using the technique to get away from the bad solution and to move away from local extreme and diversity. to solve the problem three objectives were used to optimize the timing. We use a set of particles in which particles are responsible for conducting and choosing and removing the particles to improve the set of Pareto optimal solutions. In the first proposed method, optimization particle selection is done by averaging and optimizing particle. In the second proposed method each n-particle has a guide for moving towards Pareto optimal. In the third method a technique is used to get away from the bad solution and to move away from local extreme. For removing efficiency particle of the three methods, to replace the new optimal particles used of the density measure of the uniformity of the optimal particles did not me.

While we are motivated to find the answers that ultimately only one answer is that we need to decide on the most sometimes, the user is not aware of the exact relationship between the objectives. So it is better that the set of optimal solutions Pareto found among them, then the user can be based on a Some additional information and assumptions of their minds, the best Answer adopts [6].

Multi-objective optimization algorithm able to correct find answers are not optimal. We can combine these Search algorithm with PSO algorithm have a good we can find better solutions to various Non-dominated [7].

Particle swarm optimization a techniques Initiative is based on the working population. Main idea in 1995 this doctor by doctor Kennedy and Eberhart [8] It was proposed that the collective behavior of fish and birds Inspired food. A group of birds and fish Random space for food, there is only a piece of food And none of the birds of the food does not know and only distance Knows his food, one of the best strategies for the Bird food that is closer to the theoretical, PSO algorithm is the main strategy. Each bird is a possible solution to the problem space in which PSO particle called. Each particle has a lot of merit is the merit function is calculated. Particles with higher competence are closer to the answer. The algorithm Continuous nature and their performance in various applications have been demonstrated [2].

The rest of the particle is organized as follows. We begin with an overview of related works in Section 2. MOPSO and our approach are presented in Section 3. Experimental results and discussion are presented in Section 4. Finally the paper concludes in Section 5.

2. Related Works

The proposed method is for timing problem of tasks. The method Production Set of different solutions with different quality for and allows choosing a solution to users according to their needs and requests. For example, in [9,10] objectives, such as Makespan and load balancing and Prices are the main objectives and ignore the users interests and needs. In [11] proposed price and Makespan as the main objective, regardless the load balancing by using a GA algorithm for scheduling problem modeling.

In [12] proposed an economic model for network resource management and scheduling. In [13] presented two types of GA to improve the performance of the scheduling algorithm. Minimize the total execution time and meet load balancing. In [14] a method using particle swarm optimization (PSO) is proposed to reduce the communication overhead and reduce the time to complete the process and improve resource utilization of the computational grid. In [15], the balance is the net charge on the computational grid using genetic algorithms regardless of Makespan or fees for network resources represented. In [5], 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 [5.16] considered only load balancing without Makespan or price to users consider. In [17] presented a hierarchical architecture for grid computing. So that is a two-level adaptive algorithm to minimize Makespan and maximize system throughput. NSGA-II in [18] is used to optimize the scheduling problem in heterogeneous distributed computing systems with the goal, Makespan and flow without load balancing or price.

In [19] presented multi-objective particle swarm optimization in the problem Transportation planning. The problem addressed in this paper is distributed to multiple sources of products if you use classical optimization method's complexity, it goes up and the problem becomes difficult. As a result of MOPSO solve algorithm to the problem. In this way, is divided into several sub-solution with regard to the dependence Variables and the objective function is defined by a particle swarm optimization algorithm was solved. The results show that this algorithm is robust and scalable.

In [20] presented a new method for multi-objective particle swarm optimization to solve Redundancy and reliability allocation problems. In this way, the profit function and a cost function and a function Dynamic penalty function which is used by a fine of the profit and cost controls.

In [21] presented multi-objective particle swarm optimization in systems handling is that are presented multi-objective particle swarm optimization in systems handling is stated that objectives: Minimize the Pareto fronts distance generated by the algorithm and the Pareto front, to maximize the development of solutions has been found, 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 [22] presented the integration of low-carbon distribution in EPA uses the optimization MOPSO algorithm. Done integration to distribute applicants will be done in the supply chain. MOPSO non-optimal set of solutions from the solution Desirable and practical search and remove them. Is done optimization and prioritization, rating and analysis scenario. Concerned is optimized of greenhouse gases CO_2 and cost optimization.

In [30], has proposed a new algorithm using the concepts of dominance and particle swarm optimization. Use particles and Children to overcome the lack of effectiveness. It is found in the original PSO non-dominated comparisons in the process of updating the particles of each particle is not fully exploited. The dominance comparisons among all of the 2N bit runs. Compare this with 2N bit non-dominated communication, the total population in different fronts non-dominated as NSGA-II ordered. Each bit in the front on the side that belongs to a grade is assigned. In addition to ranking, a parameter space for each particle density is calculated to give the best distribution of non-responses in the front to make sure. This parameter to assess how the particles are close to your neighbor goes to work.

In [33] proposed task scheduling using multi-objective genetic algorithm with fuzzy adaptive operators for computational grids and compared with fixed rate of mutation and crossover. Fuzzy method with a more efficient solution set of values for load balancing, makespan and price.

We have to improve [33] proposed a method and with using experiments, we show is more efficient our method makespan, price and in some cases load balancing.

3. Particle Swarm Optimization Algorithm and the Proposed Method

PSO, including 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, location 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 [25.24].In each iteration, the algorithm after finding two values, the new particle velocity and position according to the equations (1) and (2) is updated.

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

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

3.4 The Proposed Method

This section is divided into two sub-sections: the first section, Simulate and evaluate the effectiveness of the proposed method. The proposed method Evaluated by standard test function three in evaluated multi-objective optimization and Compare the particle swarm optimization. Evaluation criteria this thesis, the number of elements in the set of Pareto optimal, Uniformity and the error is. In the second part of the examined and evaluated proposed method in a grid computing network.

Section A. The proposed method first random value is given to each of the particles. Due to the non-dominant and recessive particles are separated and non-recessive bits are stored in an archive. Select for each particle, guide, and to move the selected guide.

After moving particle, a mutation [21] as well, since we in PSO Convergence is high. So you have to use the jump Convergence cut to insure the whole space we represent the problem. In the early stages with a high probability of mutation that decreases this value with increasing generation. Updated the best Personal memories of each particle is. The recessive non-members new people are added to the archive and remove members Non-dominated. When reach quorum the number of archive members, must remove some of the members, to replace the new Non-dominated members.

If the met termination conditions, ends the algorithm. Otherwise, continue selecting a guide and guides remove excess.

The proposed method uses three standard test functions Evolutionary multi-objective optimization is evaluated. In this paper, evaluation criteria is the number of elements in of the Pareto optimal set, uniformity and error. The results show that this method more optimal particle number density and high accuracy and error Less than MOPSO search method, and can As a solution for solving optimization problems Objective to be replaced. Figure 1 shows the general steps of the proposed algorithm.

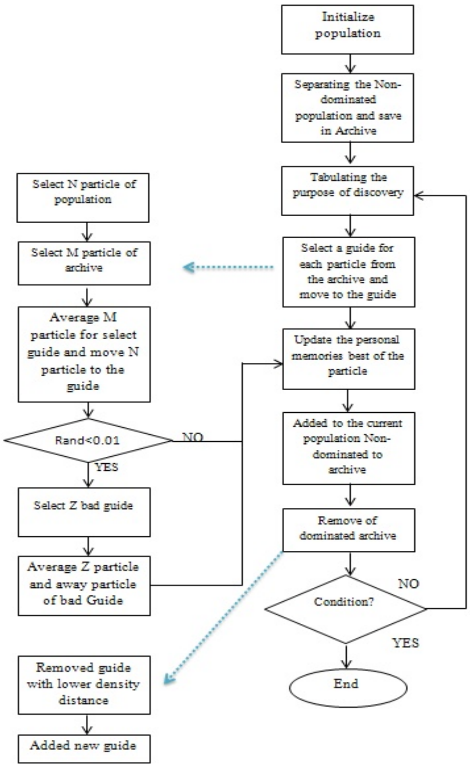

Figure 1. Overview of the Proposed Algorithm.

3.5 Guide Selection

How guide to selection has large impact on the optimum solution. That's why proposed a new method for selecting the guide to improve the MOPSO algorithm. Pareto optimal set members are selected by the guides. Proposed guide selects for each particle three methods.

3.5.1 First Method to Selection a Guide

In the first proposed method, M optimum particle randomly with Will we choose to use roulette. Homes that have little are more likely to have less choice, because this guide selection to the increase of efficiency. The mean particle M chosen as guide considered Be. How to choose the guide of Figure 2 is shown. In the Figure M = 2 is considered.

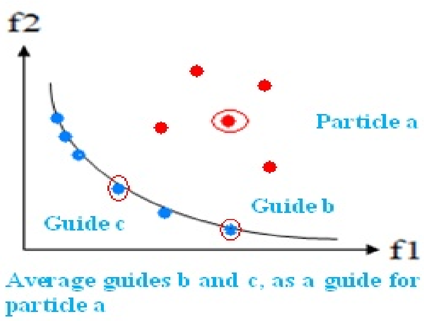

Figure 2: The First Method Guide Selection

3.5.2 Second Method to Selection Guide

In the proposed method, we considered the N particles in the population and select a guide for the N particles. For selecting the guide use the selection of the guide first method that we use M = 3 is considered. The particles move by the average guide Selected guide and eventually to the Pareto optimal front. Figure 3 shows the second method, the selection of guides.

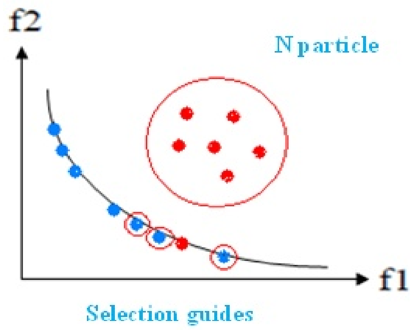

Figure 3: The Second Method is Select Guide

3.5.3 Third Method to Guide Selection

In addition to consider more particles nearest to the guide, try it some of the particles based on the number Generated random, with probability c3 away particle bad and are directed towards the Pareto front.

$$v_{id}(t+1) = wv_{id}t + c3.rand\left(x_{id}(t) - g_{worst_{id}(t)}\right) \quad (3)$$

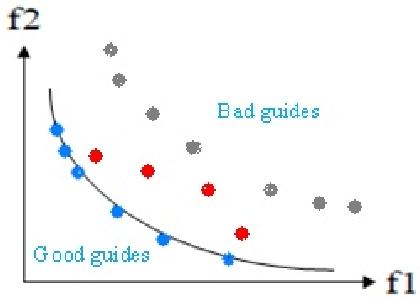

Figure 4: Third Method to Guide Selection

In addition to this proposed method to determine the number of selected leaders, and picking out the main leader of the movement to its particles, the fuzzy rules are used. Also the set of fuzzy rules for the fuzzy system is shown in Table (1). Obtained the value of M by equation (4). Figure 5 shows the membership function for the fuzzy system.

M=ceil ((p*(max-min)) + min); (4)

Table 1: the database fuzzy rules for determining M

If	GD is low	then	p is high
If	GD is high	then	p is low
If	GD is med	then	p is med

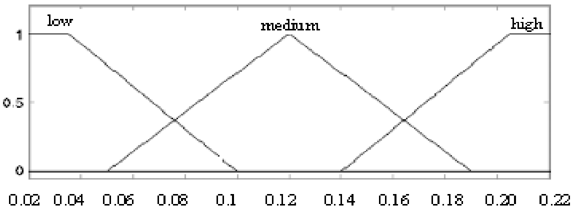

Figure 5: Input membership function to determine the parameters M phase

3.6 Guide Removes

When to reach a quorum the size of the archive, must be removed to some of the members be replaced by the new non-dominated.

To remove, we use a measure of uniformity and consistency Collection efficiency of particles and the particles do not crash with eachother away Picks Fewer dropped to remove the set the answer is low and distributed the. The density distance obtained by the equation (8).

$$d_i^1 = min\{d_{ij} \big| X^j \in Q, j \neq i\} \tag{5}$$

$$d_i^2 = min\,\{d_{ij}|d_{ij} > d_i^1, X^j \in Q, j \neq i\} \tag{6}$$

$$d_{ij} = \sqrt{\sum_{i=1}^{M}(\mu_k\,(f_k(X^i) - f_k(X^j)))^2} \tag{7}$$

$$c_{iQ} = \frac{d_i^1 + d_i^2}{2} \tag{8}$$

3.7 Objectives and functions

Three of the most important objectives in the optimization of the Grid scheduling problem has been found in research Include Makespan, price, load balancing. Due to load balancing Benefits such as resource efficiency and thus Grid system and it also decreases the response time, it is of paramount importance.

MOPSO algorithm, with three objectives in this price and Makespan And load balancing is normally two objectives and Makespan Price, are in conflict with each other. For example, when Price reduced, Makespan increases and vice versa. The reason it is that of the higher processing speed, more expensive Sources with low processing speed, and this has Conflicts and incompatibilities [33].

4. Experiments

This season is divided into two parts. The first part of the simulation and performance evaluation of our proposed methods. The proposed method uses standard test functions three in evolutionary multi-objective optimization, particle swarm optimization algorithm is evaluated and compared. Evaluation criteria in this paper, the number of elements in the set of Pareto optimal, and the error rate is uniform. In the second section, the proposed method is applied in grid computing networks, are examined and evaluated.

4.1 Section 1

The proposed methods are evaluated from three aspects:

1. Generational distance (GD): The concept of generational distance was introduced by Van Veldhuizen and Lamont as a way of estimating how far the elements are in the set of non-dominated vectors found so far from those in the Pareto optimal set and is defined as [21].

$$GD = \frac{\sqrt{\sum_{i=1}^{n} {d_i}^2}}{n} \tag{9}$$

2. Spacing (SP): Here, one desires to measure the spread (distribution) of vectors throughout the non-dominated vectors found so far. Since the “beginning” and “end” of the current Pareto front found are known, a suitably defined metric judges how well the solutions in such front are distributed. Schott proposed such a metric measuring the range (distance) variance of neighboring vectors in the non-dominated vectors found so far. This metric is defined as [21].

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d-d_i)^2} \tag{10}$$

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

$$j = 1, \dots, n\ ,\ \mathrm{d} = \text{mean of all } \mathrm{d_i} \tag{11}$$

And n is the number of non-dominated vectors found so far. A value of zero for this metric indicates all members of the Pareto front currently available are equidistantly spaced. This metric addresses the second problem from the list previously provided.

3. Error ratio (ER): This metric was proposed by Van Veldhuizen to indicate the percentage of solutions (from the non-dominated vectors found so far) that are not members of the true Pareto optimal set [21].

$$ER = \frac{\sum_{i=1}^{n} e_i}{n} \tag{12}$$

Where n is the number of vectors in the current set of non-dominated vectors available$\mathrm{e_i} = 0$, if vector i is a member of the Pareto optimal set, and$\mathrm{e_i} = 1$, otherwise. It should then be clear that ER=1 indicates an ideal behavior, since it would mean that all the vectors generated by our algorithm belong to the Pareto optimal set of the problem. This metric addresses the third problem from the list previously provided.

4.1.1 Test function 1

First test function in [23] is expressed as follows:

$$\mathrm{f_1(x) = x^2}$$

$$\mathrm{f_2(x) = (x - 2^2)} \tag{13}$$

The test function used of the archive 1 to 100 and the number of the initial population is 50 and was the number of iterations to 20.

In Table 2 can be seen that the proposed method is GD less much than the MOPSO algorithm, and this is Means that the number of particles found in the algorithm. The Pareto optimal set are more members.

In Table 3, the proposed algorithm is also reduced SP. As a result, more uniform particles with higher density dispersed.

The error is displayed in Table 4 of the MOPSO algorithm and CMPSO error by more than 3 of other algorithms there.

Table 2: Results of the GD metric for the first test function

GD	MOPSO	CMPSO	MOPSO1	MOPSO2	MOPSO3
Best	0.2159	0.10925	0.1142	0.090054	0.083412
Worst	0.3274	0.62396	0.31341	0.20463	0.291
Average	0.250712	0.19627	0.18849	0.145853	0.190952

Table 3: Results of the SP metric for the first test function

SP	MOPSO	CMPSO	MOPSO1	MOPSO2	MOPSO3
Best	0.096278	0.12946	0.12964	0.087337	0.09129
Worst	1.2454	0.37104	0.87629	0.20392	0.19821
Average	0.374938	0.294364	0.35284	0.141489	0.15518

Table 4: Results of the ERROR metric for the first test function

ERROR	MOPSO	CMPSO	MOPSO1	MOPSO2	MOPSO3
Best	0.0826	0.0826	0.0741	0.0741	0.0741
Worst	0.1304	0.1304	0.115	0.1071	0.1304
Average	0.11082	0.10834	0.09718	0.09566	0.10354

4.1.2 Test function 2

The second test in [23] is expressed as follows:

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

$$f_2(x) = \sum_{i=1}^{n}\left(|x_i|^{0.8} + 5sinx_i^3\right) \quad (14)$$

In the test 2 were considered the size of the archive 200 and the number of the initial population of 100 and the number of iterations of 200.

In evaluating the second test function in Table 5 seen two proposed algorithms have a mount GD less than the MOPSO algorithm and CMPSO and this means that the three algorithms Pareto optimal set is the number of particles found more members are. In Table 6, thee proposed algorithm is also SP less. As a result, more uniform particles with higher density dispersed.

Table 7 shows the percentage of error is shown that the MOPSO algorithm and CMPSO error by more than 3 of other algorithms there.

Table 5: Results of the GD metric for the second test function

GD	MOPSO	CMPSO	MOPSO1	MOPSO2	MOPSO3
Best	2.03851	2.5685	1.7619	1.8274	1.7425
Worst	2.4601	2.0385	2.6521	2.3863	2.4771
Average	2.180802	2.26366	2.02984	2.101662	1.99306

Table 6: Results of the SP metric for the second test function

SP	MOPSO	CMPSO	MOPSO1	MOPSO2	MOPSO3
Best	0.045384	0.04189	0.02801	0.014392	0.028383
Worst	0.23174	0.13083	0.19154	0.16151	0.051085
Average	0.115702	0.073292	0.07101	0.07814	0.04226

Table 7: Results of the ERROR metric for the second test function

ERROR	MOPSO	CMPSO	MOPSO1	MOPSO2	MOPSO3
Best	0.0395	0.0339	0.0366	0.0303	0.022
Worst	0.0769	0.1129	0.0926	0.0822	0.043
Average	0.06012	0.05918	0.05726	0.05332	0.0367

4.1.3 Test function 3

The third test function [23] is expressed as follows:

$$f1(x) = 1 - exp\left(-\sum\left(\frac{x-1}{\sqrt{n}}\right)^2\right)$$

$$f2(x) = 1 - exp\left(-\sum\left(\frac{x+1}{\sqrt{n}}\right)^2\right) \qquad (15)$$

In the test 3 were considered the size of the archive 200 and the number of the initial population of 100 and the number of iterations of 200.

Table 8, is shown that the proposed method three is much GD less than the MOPSO algorithm and this is Means that the number of particles found in the three algorithms more members are set Pareto optimal.

In Table 9, the proposed algorithm is also reduced SP. As a result, particle dispersed more uniform with higher density.

In Table 10, is shown the error that the MOPSO algorithm and CMPSO, the error is greater than 3 other algorithms.

Table 8: Results of the GD metric for the third test function

GD	MOPSO	CMPSO	MOPSO1	MOPSO2	MOPSO3
Best	0.047057	0.046153	0.03973	0.039172	0.035626
Worst	0.054195	0.076365	0.055144	0.044375	0.041065
Average	0.050508	0.055487	0.043991	0.042329	0.038362

Table 9: Results of the SP metric for the third test function

SP	MOPSO	CMPSO	MOPSO1	MOPSO2	MOPSO3
Best	0.096128	0.073575	0.082234	0.051319	0.030568
Worst	0.11224	0.11286	0.10116	0.090021	0.046101
Average	0.104914	0.100004	0.092765	0.068794	0.039237

Table 10: Results of the ERROR metric for the third test function

ERROR	MOPSO	CMPSO	MOPSO1	MOPSO2	MOPSO3
Best	0.0253	0.0238	0.0229	0.0196	0.0182
Worst	0.0319	0.0476	0.0268	0.028	0.0244
Average	0.02836	0.0316	0.02506	0.0245	0.02148

4.2 Section 2: implementation Improved particle swarm optimization algorithm

In a scheduling problem with n task and m source, every particle has three features, location, cost and speed. The position of each particle is obtained regarding the resources and tasks. Length of the array is considered for the position the number of tasks in the task is learned. The content of each house of the array, which represents a number between 1 and m is the reference number assigned to complete the task. Figure 6 schema is shown an array to the problem of scheduling tasks.

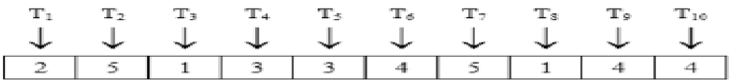

Figure 6: scheme an array for tasks scheduling

A Grid system with three objective Makespan, price and load balancing the target price Makespan and scheduling tasks, an important objective and Grid computing environment for users that are essential for economic and those resources are important. In addition to these two objectives, load balancing, it has many advantages, such as Reduce response time and increase productivity and system resources.

Detailed simulation data in Table 11 are presented.

Table 11: Parameter problem

value	parameter
100	Number of generations
40	Number Guides
c1=1,c2=2	The value of c1, c2
0.2	The value of w
500	Number of tasks
50	Number of sources
20-100 (MI)	The size range of tasks
1-5 (G$/sec)	Price Range Resources
2-10 (MI/sec)	CPU speed range
3 objectives	Optimization objective

In Figures 7 and 8, can be seen the simulation results of the proposed method and fuzzy method that the proposed method have more efficient and cost and makespan.

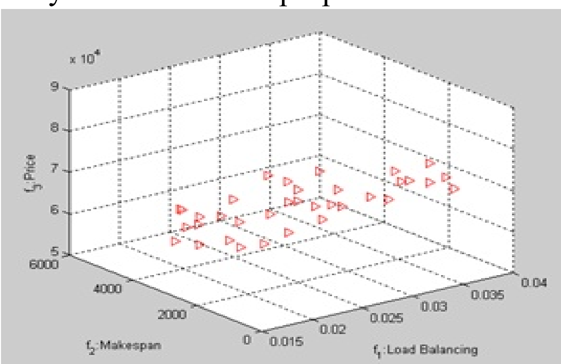

Fig 7: Fuzzy mutation and cross over method

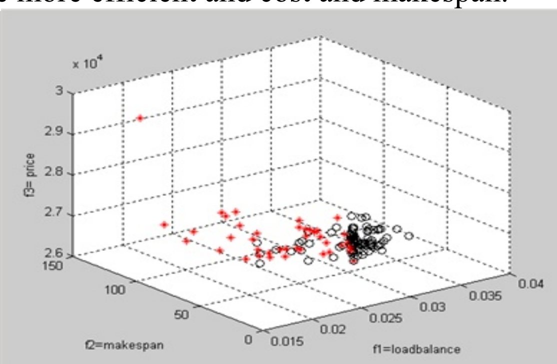

Fig 8: Proposed method

The method proposed by [33] compared the results in table (12, 13) states.

Table 12: Pareto optimal solutions for fuzzy mutation method

Best load balancing			Best price			Best Makspan			Factors / method
The mean square deviation of productivity	price	makespan	The mean square deviation of productivity	price	makespan	The mean square deviation of productivity	price	makespan	
0.0186	90323	798.89	0.0198	40426	5498	0.0196	101625	187.5	Fuzzy mutation method [33]
0.0194	27449	140.998	0.0278	23704	38.25	0.0268	26671.08	38.25	Proposed method

Table 13: Pareto optimal solutions for takeoff and method

Best load balancing	Best price			Best Makespan			Factors / method
The mean square deviation of productivity	The mean square deviation of productivity	price	makespan	The mean square deviation of productivity	price	makespan	
0.0191	0.0196	27565	3768	0.0192	65564	176	Fuzzy mutation and cross over method[33]
0.0194	0.0278	23704	38.25	0.0268	26671.08	38.25	Proposed method

As the table (12, 13) we can see, the proposed method has Makespan price jumps fuzzy method have more efficient than the quality of the Pareto optimal solutions generated by the proposed method is better than the mutation fuzzy method.

We have three goals to balance the load charts, Max Penn and priced separately, we observed during the optimization. For comparison of M, Z, c3 various did simulations in accordance with Table (11). M number of good guides intended to select Ultimate Guide for each particle, Z number of bad guides intended to keep out the part of it is a guide. c3 is a constant value, which implies that each particle in the amount out there of bad Guides.

As previously mentioned, the proposed method has Makespan and optimal price than the rate of mutation and crossover fuzzy. According to the results in the form of fig (9) to (11) indicates the superiority of the proposed method is the value of M = 3. Although the diagram (10), M = 6 has a more optimal solution, but our main objective improvement Makespan quantity and price. In this simulation, we consider the value of Z is equal to 2 and the value of w is equal to 0.2.

The average price in the diagram (11), M=3, while the first generation is contains price high amount, but over time it can be improved in the last generation.

According to the results in the form of (12) to (14) indicates the superiority of the proposed method is the value of Z = 2. In this simulation, considered as the previous test value of M = 3.

According to the results in the form of (15) to (17) indicates the superiority of the proposed method is much c3 = 0.8. In this simulation, the previous experiments, the amount of M = 3 and Z=2.

Figure (16), although c3 = 0.2 in the last generation, Contains amount more efficient makespan, but no have amount optimal load balancing and price.

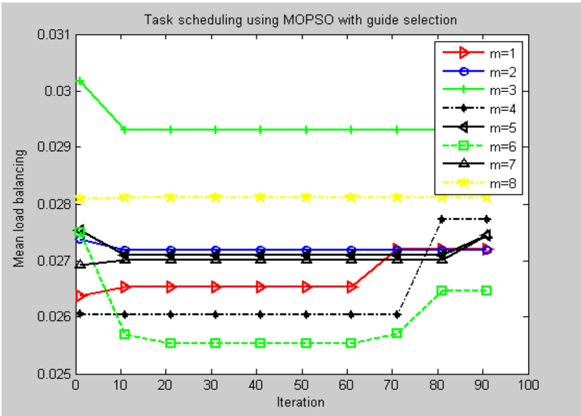

Fig (9). Diagram of mean values load balancing

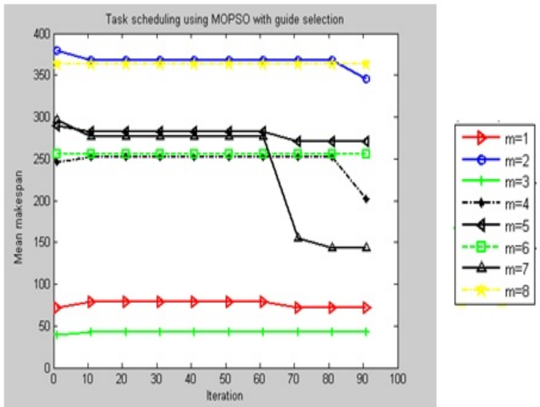

Fig (10). Diagram of mean the values makespan

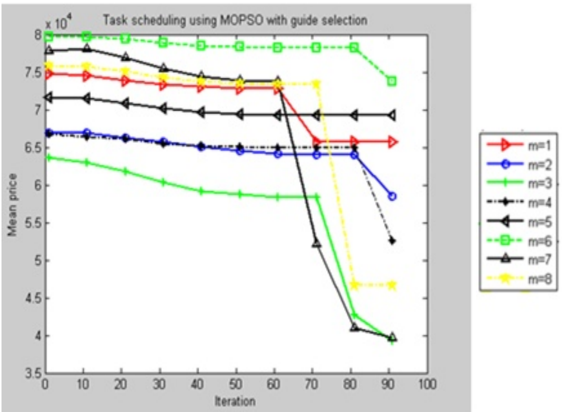

Fig (11). Diagram of mean the values price

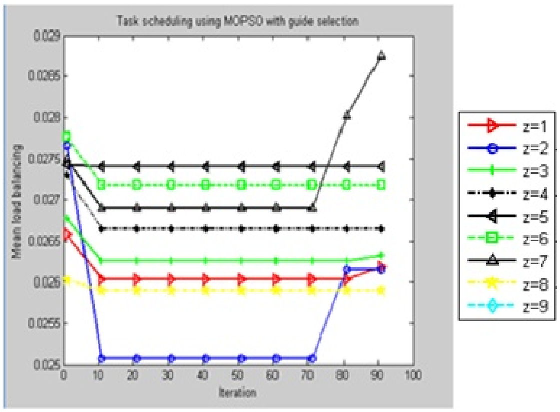

Fig (12). Diagram of mean the values load balancing

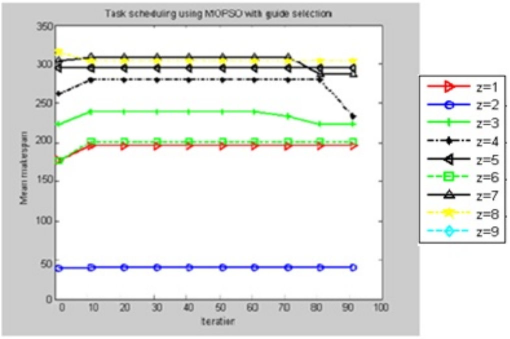

Fig (13). Diagram of mean the values makespan

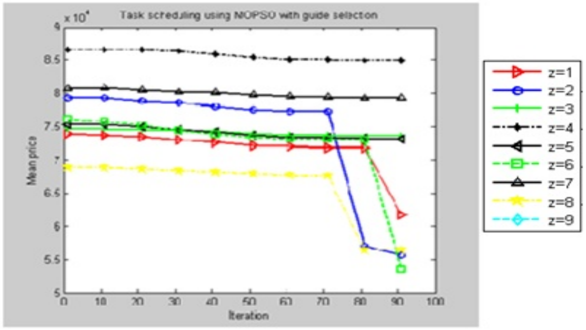

Fig (14). Diagram of mean the values price

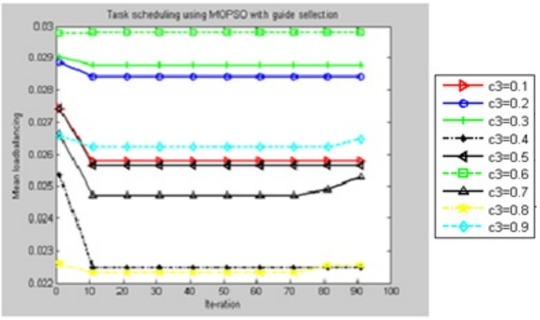

Fig (15). Diagram of mean the values load balancing

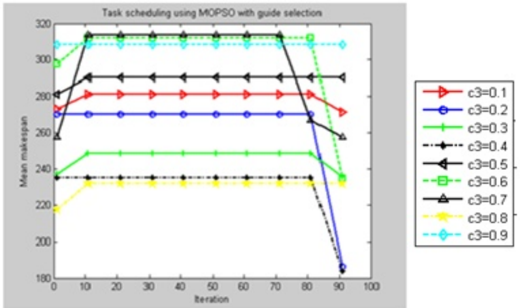

Fig (16). Diagram of mean the values makespan

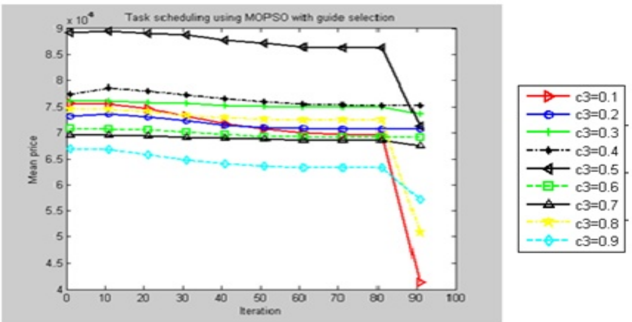

Fig (17). Diagram of mean the values price with different c3

5. Conclusions

In this paper, we use the particle swarm optimization algorithm, with Changes in the selection and removal of the guide and the guide of use a technique to get away from the bad, to move away from Local extrema and more variety, to solve the problem planning work, we use market by leaps and Phase crossover [33] compared.

As the results of testing the quality of answers the proposed method have better price and Makespan.

The proposed algorithm performs better than the basic PSO method for multi-objective optimization. The results show that our approach is a viable alternative for solving multi-objective problems. Because having very competitive performance compared to the average of evolutionary algorithms for multiple days. In fact, proposed MOPSO are able to cover the complete Pareto front in using test functions.

In addition, the proposed method is accountable on issues to thirty dimensions, if the particle swarm algorithm is able to respond to ten. The proposed method on problems with up to ten dimensions than the PSO method gives a more optimal solution.

Also spread set of Pareto optimal solutions in uniform more and more- dense than the algorithms studied.

Finally, the future work would be more accurate to refer to an algorithm using parameters fuzzy and the parameters and select and remove the guide or the mutation rate fuzzy, will obtain the desired response on dynamic functions.

References

[1] H. Nezamabadi, M. rostami. 1385. Generalization of the algorithm GCBPSO ", Twelfth International Conference of Computer Engineering in iran, esfand, page 29-35.(in persian).

[2] M. Sheibani, MR Meybodi. 1385. PSO-LA : A new model for optimization, Twelfth International Conference of Computer Engineering in iran, esfand, pages 1162-1169 (in persian)

[3] Li, Y., Yang, Y., Ma, M., & Zhou, L. (2009). A hybrid load balancing strategy of sequential tasks for grid computing environments. Future Generation Computer Systems, 25(8), 819-828.

[4] Brown, M., Fukui, K., & Trivedi, N. (2005). Introduction to grid computing. IBM, International Technical Support Organization.

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

[6] Kalyanmoy, Deb.,"Multi-objective optimization", Search methodologies. Springer US, 403-449, 2014.

[7] Coello Coello, C. A, Lechuga, M. S., "MOPSO: A proposal for multiple objective paper swarm optimization", In Evolutionary Computation, IEEE, 1051-1056, 2002.

[8] Eberhart, R.C, Kennedy, J., "A new optimizer using paper swarm theory", In Proceedings of the sixth international symposium on micro machine and human science, 39-43, 1995.

[9] Y. Li, Y. Yang, R. Zhu, A hybrid load balancing strategy of sequential tasks for computational grids, in: International Conference on Networking and Digital Society, IEEE, 2009.

[10] M.I. Daoud, N. Kharma, A high performance algorithm for static task scheduling in heterogeneous distributed computing systems, J. Parallel Distrib.Comput. 68 (4) (2008) 399–409 (ELS).

[11] S. Kardani-Moghaddam, F. Khodadadi, R. Entezari-Maleki, A. Movaghar, A hybrid genetic algorithm and variable neighborhood search for task scheduling problem in grid environment, in: International Workshop on Information and Electronics Engineering, IWIEE, Procedia Engineering 29, 2012,pp. 3808–3814

[12] R. Buyya, J. Giddy, D. Abramson, An evaluation of economy-based resource trading and scheduling on computational power grids for parameter sweep applications, in: Proceedings of the 2nd International Workshop on Active Middleware Services (AMS 2000), August 1, 2000, Kluwer Academic Press, Pittsburgh, USA, 2000.

[13] F.A. Omara, M.M. Arafa, Genetic algorithms for task scheduling problem, J. Parallel Distrib. Comput. 70 (1) (2010) 13–22 (ELS).

[14] J. Ma, Lanzhou, A novel heuristic genetic load balancing algorithm in grid computing, in: Second International Conference on Intelligent Human–Machine Systems and Cybernetics, 2010

[15] S. Prakash, D.P. Vidyarthi, Load balancing in computational grid using genetic algorithm, Adv. Comput. 1 (1) (2011) 8–17. http://dx.doi.org/10.5923/j.ac.02.

[16] H. Peng, Q. Li, One kind of improved load balancing algorithm in grid computing, in: International Conference on Network Computing and Information Security, 2011.

[17] A.Touzene, S. Al-Yahai, H. AlMuqbali, A. Bouabdallah, Y. Challal, Performance evaluation of load balancing in hierarchical architecture for grid computing service middleware, IJCSI Int. J. Comput. Sci. Problems (ISSN: 1694-0814) 8 (2) (2011)

[18] . Subashini, M.C. Bhuvaneswari, NSGA-II with controlled elitism for scheduling tasks in heterogeneous computing systems, Int. J. Open Probl. Compt. Math. (ISSN: 1998-6262) 4 (1) (2011).

[19] Zheng, Y. J., & Chen, S. Y. (2013). Cooperative particle swarm optimization for multiobjective transportation planning. Applied intelligence, 39(1), 202-216.

[20] Khalili-Damghani, Kaveh, Amir-Reza Abtahi, and Madjid Tavana. "A new multi-objective paper swarm optimization method for solving reliability redundancy allocation problems." Reliability Engineering & System Safety, 58-75, 2013.

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

[22] Validi, Sahar, Arijit Bhattacharya, and P. J. Byrne. "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, 3074-3096, 2014.

[23] Deb, K, Agrawal, S, Pratap, A, Meyarivan, T., "A fast elitist nondominated sorting genetic algorithm for multi-objective optimization:NSGA-II"., in Proc. 6th Int. Conf. Parallel Problem Solving from Nature, pp. 849–858, 2000.

[24] X. Hu and Y. Shi and R. Eberhart, "Recent advances in paper swarm optimization", IEEE, 2004.

[25] J.Kennedy and R. Mendes, "Population structure and paper swarm performance," in Proc. IEEE Congr. Evol. Comput., Honolulu, HI, pp. 1671–1676, 2002.

[26] X. Cui, "Document clustering using paper swarm optimization", IEEE, pp. 185-191, 2005.

[27] L. Hongbo and M. Abraham, "Fuzzy adaptive turbulent paper swarm optimization", IEEE, pp. 39-47, 2005.

[28] Y. Shi and R. Eberhart, "A modified paper swarm optimizer",IEEE International Conference on Evolutionary Computation, USA, pp. 255- 262, 1998.

[29] F. Bergh and A. Engelbrecht, "A new locally convergent paper swarm optimizer", IEEE, 2002

[30] Liu, Yang, A fast and elitist multi-objective particle swarm algorithm: NSPSO, Granular Computing, IEEE International Conference on. IEEE, pp. 470-475, 2008.

[31] A.Y. Zomaya, Yee-Hwei Teh, Observations on using genetic algorithms for dynamic load-balancing, IEEE Trans. Parallel Distrib. Syst. 12 (9) ,2001.

[32] B.T.B. Khoo, B. Veeravalli, T. Hung, C.W.S. See, A multi-dimensional scheduling scheme in a grid computing environment, J. Parallel Distrib. Comput. 67 (6), 659–673 (ELS), 2007.

[33] Salimi, Reza, Homayun Motameni, and Hesam Omranpour. "Task scheduling using NSGA II with fuzzy adaptive operators for computational grids." Journal of Parallel and Distributed Computing 74.5, 2333-2350 2014.