IJICTR
International Journal of Information & Communication Technology Research

Volume 10- Number 3 – Summer 2018 (12 -18)

# PSO-Based Scheduling Algorithm for Time and Cost Optimization in Mobile Grid

Mehdi Golsorkhtabaramiri*
Department of Computer Engineering,
Babol Branch, Islamic Azad University,
Babol, Iran.
golesorkh@baboliau.ac.ir, golesorkh@gmail.com

Fahimeh Zakeri
Department of Computer Engineering,
Babol Branch, Islamic Azad University,
Babol, Iran.
fahimeh.zakeri@gmail.com

Received: 9 March, 2018 - Accepted: 15 Jun, 2018

*Abstract*— **Resource owners in mobile grids, with financial motivations in mind, provide their clients with the resources they possess. The clients can, on the other hand, obtain what they seek through the payments they make. Mobile grid users may also demand cost- and time-optimizing strategies as part of their service quality. As a result, the scheduling system, through making use of a proper algorithm, should allocate resources to users' independent tasks in such a way that would minimize time and cost parameters. Hence, in this paper, two scheduling algorithms are introduced with regard to user needs based upon Particle Swarm Optimization (PSO), which, through mutation and population expanding, are able to find the best scheduling scheme for user tasks, while preventing quick algorithm convergence.**

***Keywords- Mobile Grid; Resource Scheduling; Time Optimization; Cost Optimization; Particle Swarm Optimization; local Minimum; Mutation, Expanding Population***

## I. INTRODUCTION

A mobile grid is a combination of wire grids, gateways and a wireless grid that is setup based on mobile calculations and grid calculations [1,2]. In fact, users can send tasks to mobile grids, then a scheduler based on a scheduling algorithm distributes the tasks between mobile nodes, and after finishing the calculation, the results will be sent to the schedulerm and the scheduler transmits them to related the user [3,4]. Since the nodes are mobile, cases such as limitation of battery capacity, increased motions and low security can produce problems and new challenges in this context. A mobile grid is usually adopted the wireless connection between mobile devices and grids, this connection is unstable, thus the mobile device and grid system may disconnect at any time [5].

One of the most important part in a mobile grid is scheduling of resources [6]. A scheduler of resources, regarding user needs, should take best scheme of allocating work for reaching optimum scheduling. Traditional schedulers often concentrate on execution time of tasks, but other limitations may also be imposed.

Mobile grid users can request optimization of time or cost as a quality of services, and system scheduling must follow the strategy that minimizes time and cost parameters in allocating tasks to resources. Hence in this paper, in order to realize user needs, two new algorithms are presented.

## II. RELATED WORKS

Shengjun et al. [7] proposed a differential evolution algorithm that is similar to the majority of evolutionary algorithms. However, the difference is in the mutation operation, which is the most crucial part

* Corresponding Author

of the differential evolution algorithm. In differential evolution algorithm, the basic principle is to select randomly three individuals in a population and to carry out the vector differential operator after which comes a multiplication by the scaling factor. Finally, a new individual is obtained by adding the third individual. The best individual will be the next generation individual by comparison with the new individual and the best individual in contemporary population. The differential evolution algorithm takes the possibility of global random points into account in the process of evolving, and enhances the global search capability.

Lei et al. [8] proposed a heuristic approach that is adopted in solving scheduling problems in grid environment. Each particle represents a possible solution, and the position vector is transformed from the continuous values to the discrete values based on the small position value rules, accordingly, a permutation is formed. This approach aims to generate an optimal schedule so as to get the minimum makespan and maximum resource utilization while completing the tasks.

Zahra et al. [9] proposed a hybrid-scheduling algorithm to solve the independent task scheduling problem in grid computing. This algorithm uses particle swarm optimization (PSO) as the main search algorithm, while the gravitational emulation local search (GELS) algorithm is used to improve the population. There are two reasons for using both algorithms. First, we need an algorithm that is based on a population that can search the entire grid space for this problem. Second, the grid environment is dynamic, so the scheduling algorithm must be fast enough to adapt with the natural grid environment and must be able to converge faster than other algorithms. Moreover, although PSO is weak for local searches, this combination of PSO with an algorithm that is strong in local searches addresses this weakness. This algorithm minimizes makespan and the number of tasks that fail to meet their deadlines.

Zhou et al. [10] proposed a novel cultural algorithm based on the PSO algorithm. In the process of PSO optimization, the particle traps two targets: one is the best-found position of the single particle up to the current iteration count, and the other is the global best-found position among all particles in the swarm up to the current iteration. The information of global best-found solutions is accepted to the belief space and then influences the update of population space in reverse. By the dual evolutionary mechanism of PSO space (the main population space) and belief space (the additional upper space), the algorithm attains a better capacity of global search. At each generation, individuals in the population space are first evaluated with the objective function so that the good-performing individuals are found. The acceptance function, accept (), is then used to determine which individuals will be allowed to update the belief space, which are normally elitists. The population space and the belief space of cultural algorithm are redesigned. The algorithm is stable and presents low variability.

Wang et al. [11] proposed an efficient task scheduling method based on an advanced no-velocity PSO. To standardize PSO, the particle modifies its position by velocity. However, they advance the no-velocity PSO using the subgroup best position's center instead of the swarm's center, using subgroup best position instead of the personal best position, and use it in the grid task scheduling. The initial particle groups can be divided into several subgroups. They use the personal best position for the best position of the subgroup, and use the global best position for the best position of the global group. In no-velocity PSO, the other leading node – the swarm's center is imported. But in the task-scheduling problem, the task response time requires more stringency, so it requests that the convergence time not be too long. In order to shorten the convergence time, they consider the value of all subgroup's best position as the center of the node. This can speed up the convergence. The subgroup's best position and their center information are taken into account to get minimum makespan and minimum mean task response time.

Bu et al. [12] proposed an improved algorithm with a discrete coding rule for the grid scheduling problem. In the grid environment, the scheduling problem is to schedule a stream of tasks to a set of nodes. During the execution, there are some communications between nodes. The function of the algorithm is to find the best tasks scheduling strategy and to obtain the optimal makespan.

Sara et al. [13] proposed a hybrid genetic algorithm (GA) and variable neighborhood search (VNS) which was presented to reduce the overall cost of task executions without a noticeable increment in system makespan. This paper addresses the scheduling problem of independent tasks in the market-based grid environment. In market-based grids, resource providers can charge users based on the amount of resource requested by them. In this case, scheduling algorithms should consider users' willingness to execute their applications in the most economical manner. GA-VNS runs the genetic as the main algorithm and uses the VNS procedure for improving individuals in the population. Each individual in the population is used to generate new offsprings by applying appropriate genetic operators such as selection, crossover and mutation.

## III. THE RESOURCES SCHEDULING ALGORITHM

Since users can request the time or cost parameter as quality of services, the aims of scheduling algorithm can be put forward as the following:

- Time Optimization
- Cost Optimization

In case of time optimization, the user follows the allocative scheme to do the task in the least possible time, and in case of cost optimization, also, the user wants an allocative scheme that does the task with the least possible cost. Consequently, the resource scheduler must use an algorithm which meets the needs of the user.

In this paper, two new scheduling algorithms are introduced that are in agreement with each of the aims. For the first case, we assume that people use laptops, Personal Digital Assistants (PDAs) or other mobile

devices for connecting to a network. They send the task to the scheduler, and the scheduler divides the task into some sub tasks that do not have any relation with each other. This case must be considered that each resource can do a task simultaneously in every moment. Regarding these cases, the proposed evolutionary algorithms are stated as in the following:

### *A. The PSO algorithm*

The PSO is a social search algorithm that is modeled from behavior of birds or fish groups [14]. In this method, particles are flowing in the search space, and the variation of particle positions in the search space is affected by their experiments and neighbors. Therefore, the position of other particles affects thesearch quality of a particle. The result of modeling such social behavior is a search process that leads particles toward the successful regions. The concept of PSO is not difficult, and the number of individuals is small, the calculations are simple, and it can be an advantage in the mobile telecommunications.

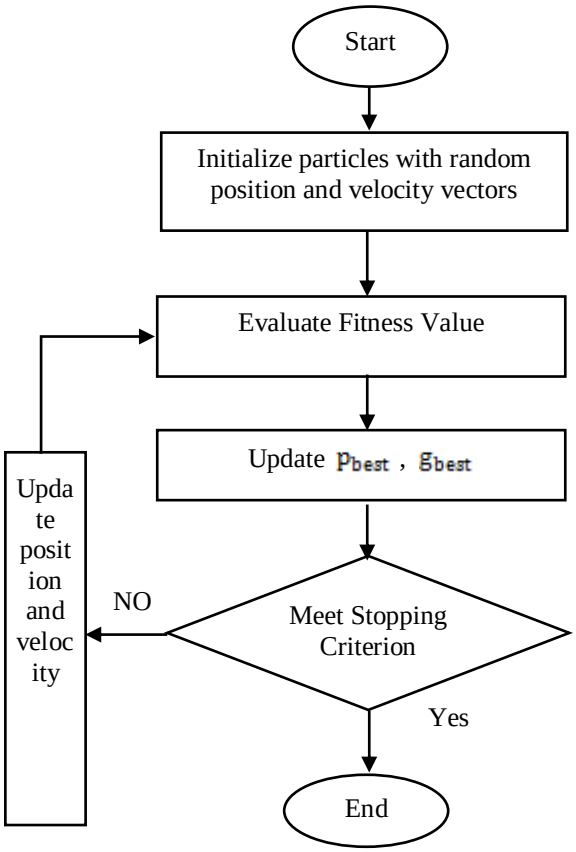



Fig 1. The flow of the PSO algorithm

In PSO, in every moment, every particle regulates its position in the search space regarding the best position that has been placed until now $(p_{best})$ and the best position that exists in its neighborhood $(g_{best})$ according to (1) and (2) .

(1) $v_i = wv_i + c_1r_1(p_{best} - x_i) + c_2r_2(g_{best} - x_i)$

(2) $x_i = v_i + x_i$

That $(i)$ is from 1 to $n$, and $(n)$ is equal to the number of population particles, $(v_i)$ is velocity of $i$ $i$Th particle, $(x_i)$ is the position of the $i$Th particle, $(r_1, r_2)$ are random numbers, $(c_1, c_2)$ are training factors, $(p_{best})$ is the best position of every particle, $(g_{best})$ is the best position between all particles and $(w)$ is the amount of weight.

In resource scheduling in mobile grids, every allocation scheme can be shown with a particle in PSO. For this, we assume that N is the number of mobile nodes in the network and the number of independent tasks is also equal to M. In the vector of particle position $X = (x_1, x_2, \dots, x_M)$ , the $i$Th task has been allocated to the node of the $x_i$ resource. There is a system in the network that obtains the time or cost of the task performance through the resource and, as a matrix, hands the result over to the PSO algorithm for being evaluated. Of course, here, the matrix is valued with random amounts.

The mathematical model for resource scheduling is as follows:

The set of tasks is T= $\{T_1, T_2, \dots, T_M\}$ and the set of resource nodes is R= $\{R_1, R_2, \dots, R_N\}$, where N can be less than or equal to M. the information matrixes of resources and tasks are as follows:

The expected execution time matrix (EET) returns the execution time of the nodes for doing the tasks.

$$EET = \begin{bmatrix} eet_{11} & \cdots & eet_{1N} \\ \vdots & \ddots & \vdots \\ eet_{M1} & \cdots & eet_{MN} \end{bmatrix}$$

The expected execution cost matrix (EEC) returns the execution costs of the nodes for doing the tasks.

$$EEC = \begin{bmatrix} eec_{11} & \cdots & eec_{1N} \\ \vdots & \ddots & \vdots \\ eec_{M1} & \cdots & eec_{MN} \end{bmatrix}$$

In the proposed methods, these matrixes are valued with random amounts. The proposed methods use these matrixes for evaluating the fitness values of particles. Therefore, the algorithms must find the best scheduling scheme through considering these matrixes. In fact, in this paper, the time optimization algorithm uses the EET matrix to find the optimal scheme with minimum time. The cost optimization algorithm uses the EEC matrix to find the optimal scheme with minimum cost.

### *B. Proposed algorithms of scheduling resources based on PSO*

The important issue is that the PSO algorithm is easily captured by the local minimum. That is, when one particle moves toward the local minimum, the capture will occur, because in PSO, particles learn from each other. Consequently other particles also move towards the local minimum, and the algorithm

gets convergent to a local optimum amount. Hence, for preventing this, the two following algorithms are proposed.

*1) PSO algorithm with mutation*

This method, much like the PSO algorithm, starts the search process with numerous particles as initial population. Particles are flowing in the search space with velocity and position. Fitness of every particle is calculated regarding the obtained information from the network. The best position of every particle is recognized as $p_{best}$ and the position of the particle that, among other particles, has less fitness is recognized as $g_{best}$ . Regarding the fact that the search process is a reiterative process, the solution of the problem will be a continued base on maximum iteration considered for it. For preventing trapping in local minimum in every iteration, uniform mutation is used[15]. For doing this in every iteration, half of the particles that have lower operation in population and change the position with a random amount are chosen. In case the new position is in lower fitness, the new position will be substituted with the particle former position, otherwise, substituting the position does not take place. In fact, the mutation results in the moving of particles in the search space and helps global searching. In addition to mutation, this procedure updates the position and velocity of particles according to (1) and (2) until reaching maximum iteration in every iteration. In updating, a great amount of $w$ helps global searching and a small amount helps local search. Hence with a linear decrease of $w$ from a great amount to a small amount we can obtain the best operation [16]. For better operation of this method, in every iteration, the amount of $w$ decreases based on the iteration. After updating, the fitness of particles is calculated in the new position. If fitness of the new position of the particle is lower than $p_{best}$ fitness of that particle in previous iteration, the new position is saved as $p_{best}$ . Also, if the fitness of best particle is lower than the fitness of $g_{best}$ in the previous iteration, the amount of $g_{best}$ is changed to that position.

**Algorithm1** PSO algorithm with mutation

Step 1: Initialize the size of the particles swarm $n$
Step 2: Initialize $x_i$ and $v_i$ for each particle $i$ in the
Population
Step 3: Calculate the fitness value of each particle
Step 4: Initialize $p_{best}$ for each particle
Step 5: Initialize $g_{best}$
Step 6: While the end criterion is not met do
Steps 7 to 11
Step 7: For each particle $i$ update $v_i$ and $x_i$ according to equations (1) and (2)
Step 8: Calculate the fitness value of each particle
Step 9: For each particle $i$ :
If fitness < Best Fitness Then
$p_{best} = x_i$
Step10: Mutation
Step11: Update $g_{best}$

Regarding the mentioned subjects, the algorithm of the new method is stated as follows:

*2) PSO algorithm with mutation and expanding population*

This algorithm, like the former algorithm with mutation, leads the particles to better points. Wherever the initial population is more, more solutions are discovered [17]. This method increases the existing population, and since increasing population results in increasing calculations, the considered population increases only in a number of iterations.

Steps of the proposed algorithm are defined as follows:

This method, like the PSO algorithm, starts the searching process with a number of particles as initial population. Particles are flowing in the search space with velocity and position. Fitness of every particle is calculated regarding the obtained information from the network. The best position of every particle is recognized as $p_{best}$, and the position of the particle that has less fitness compared to other particles is recognized as $g_{best}$ . Regarding the fact that the search process is a reiterative one, the solution of problem will be continued base on the maximum iteration that is considered for it. In order to detect more and better solutions, the population increases only in a number of iterations. For preventing trapping in local minimum, in every iteration, it uniform mutation is used. For doing so, in every iteration, half of the particles that have lower operation in population are chosen and change the position with a random amount. In case the new position is in lower fitness, the new position will be substituted by the particle former position, otherwise substituting of position does not take place. In fact, the mutation results in moving particles in the search space and helps global searching. In addition to mutation, this procedure updates the position and velocity of particles according to (1) and (2) until reaching maximum iteration in every iteration. In updating, the great amount of $w$ helps global searching and the small amount helps local search.

Hence with linear decrease of $w$ from the great amount to small, we can obtain the best operation. For better operation of this method in every iteration, the amount of $w$ decreases based on the iteration. After updating, the fitness of particles is calculated in the new position. If fitness of the new position of the particle is lower than $p_{best}$ fitness of that particle in previous iteration, the new position is saved as $p_{best}$ . Also, if the fitness of the best particle is lower than that of $g_{best}$ in the previous iteration, the amount of $g_{best}$ is changed to that position. Regarding the mentioned subjects, the algorithm of the new method is stated as follows:

## IV. SIMULATION RESULTS

The performance of the PSO algorithm, PSO algorithm with mutation (MPSO) and PSO algorithm with mutation and expanding population (MPSO with expanding population) in equal conditions have been studied for time and cost optimization.

Accordingly, it is assumed that the number of mobile resources in the mobile grid is 8, the number of initial population is 100, maximum iteration of algorithms is 100, $C_1$and $C_2$ are 2 in MPSO and MPSO with expanding population algorithms and $r_1$, $r_2$, EET and EEC are valued with random amounts.

**Algorithm2** PSO algorithm with mutation and expanding population

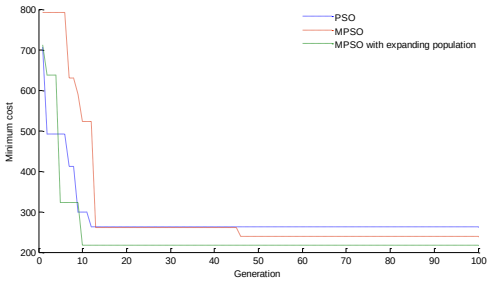
Step 1: Initialize the size of the particles swarm $n$
Step 2: Initialize $x_i$ and $v_i$ for each particle $i$ in the population
Step 3: Calculate the fitness value of each particle
Step 4: Initialize $p_{best}$ for each particle
Step 5: Initialize $g_{best}$
Step 6: While the end criterion is not met do
Steps 7 to 18
Step 7: While (expanding iterations) do
Steps 8 to 13
Step 8: Initialize a new Swarm
Step 9: Initialize $x_i$ and $v_i$ for the new particles
Step10: Calculate the fitness value of particles
Step11: Initialize $p_{best}$ for the new particles
Step12: Add the new particles to the population
Step13: Update $g_{best}$
Step14: For each particle $i$ update $v_i$ and $x_i$ according to equations (1) and (2)
Step15: Calculate the fitness value of each particle
Step16: For each particle $i$ :
If fitness < Best Fitness Then
$p_{best} = x_i$
Step17: Mutation
Step18: Update $g_{best}$

The PSO algorithm has been compared with MPSO and MPSO with expanding population algorithms per numbers of 5, 10, 15 and 20 tasks. These comparison results with the aim of time optimization have been shown in "Fig. 2" and "Table.1".

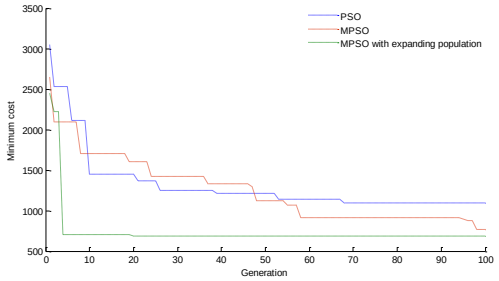
Table.1 Comparison of execution time

| Number of task | Minimum execution time (S) | | |
|---|---|---|---|
| | PSO | MPSO | MPSO with expanding population |
| 5 | 672 | 646 | 508 |
| 10 | 3158 | 2948 | 2540 |
| 15 | 7529 | 5981 | 3978 |
| 20 | 12950 | 11330 | 9140 |

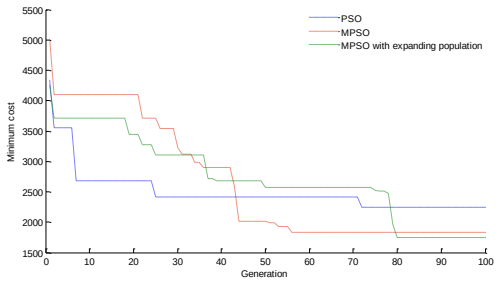
As can be seen, the MPSO and MPSO with expanding population algorithms are better than the



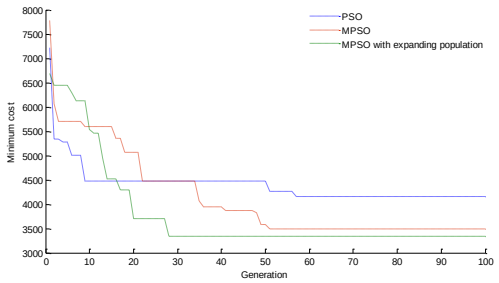

(a) The number of task=5




(b) The number of task=10




(c) The number of task=15




(d) The number of task=20

Fig.3 Comparison of minimum execution cost of algorithms

PSO algorithm and find better schemes with lower time. In addition, the MPSO with expanding population algorithm is better than the PSO algorithm. Actually, the PSO algorithm is easily captured to local minimum time.

These algorithms have been run for 50 times, and the means of results have been computed. The means of minimum time of each algorithm have been shown in "Table.2".

Also, the comparison of these algorithms with the aim of cost optimization have been computed. The

comparison of results per numbers 5, 10, 15 and 20 tasks have been shown in "Fig. 3" and "Table.3".

As can be seen, the proposed algorithms operate better than the PSO algorithm, and also, the MPSO with expanding population is better than the MPSO algorithm. The proposed algorithms find better schemes with lower cost and avoid quick convergence.

Table.2 Mean of minimum execution time

| Number of task | Minimum execution time (S) | | |
|---|---|---|---|
| | PSO | MPSO | MPSO with expanding population |
| 5 | 2.2436e+003 | 2.0848e+003 | 2.0322e+003 |
| 10 | 6.4811e+003 | 6.0589e+003 | 5.3191e+003 |
| 15 | 1.2681e+004 | 1.1268e+004 | 1.0663e+004 |
| 20 | 1.7750e+004 | 1.6947e+004 | 1.5689e+004 |

Table.3 Comparison of execution cost

| Number of task | Minimum execution time (S) | | |
|---|---|---|---|
| | PSO | MPSO | MPSO with expanding population |
| 5 | 262 | 240 | 218 |
| 10 | 1094 | 768 | 686 |
| 15 | 2246 | 1833 | 1752 |
| 20 | 4163 | 3497 | 3338 |

These algorithms were run for 50 times, and the means of minimum cost of each algorithm have been shown in "Table.4".

Table.4 Mean of minimum execution cost

| Number of task | Minimum execution time (S) | | |
|---|---|---|---|
| | PSO | MPSO | MPSO with expanding population |
| 5 | 644.2600 | 556.3600 | 532.3800 |
| 10 | 1.8287e+003 | 1.6155e+003 | 1.5788e+003 |
| 15 | 3.3014e+003 | 3.1106e+003 | 3.0020e+003 |
| 20 | 5.1339e+003 | 4.9388e+003 | 4.6684e+003 |

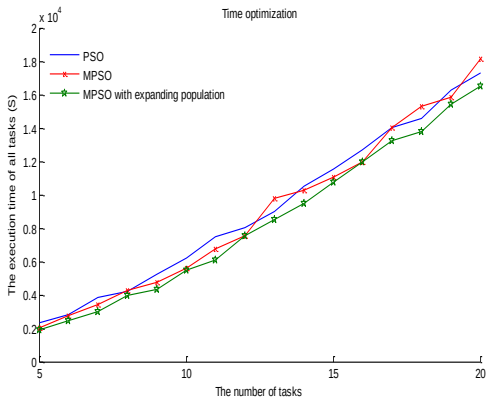



Fig.4 Comparison of PSO algorithm with proposed algorithm of MPSO, MPSO with expanding population with aim of time optimization

The PSO algorithm has been compared with MPSO and MPSO with expanding population algorithms when the number of tasks increases from 5 to 20. The mean results of 50 runs with the aims of time and cost optimization have been shown in "Fig. 4" and "Fig.5," respectively.

As observed, the two proposed algorithms perform better than PSO. Also, the algorithm of MPSO with expanding population performs better than MPSO. In fact, MPSO, through mutation, can obtain the best scheme with the least possible time for time optimization as well as the least cost for cost optimization in comparison to the PSO algorithm. The MPSO with Expanding Population, through its expanding of the population searches more points in the space of possible answers and as a result, obtains better schemes than do the PSO and MPSO algorithms for resource scheduling with time and cost optimization goals.

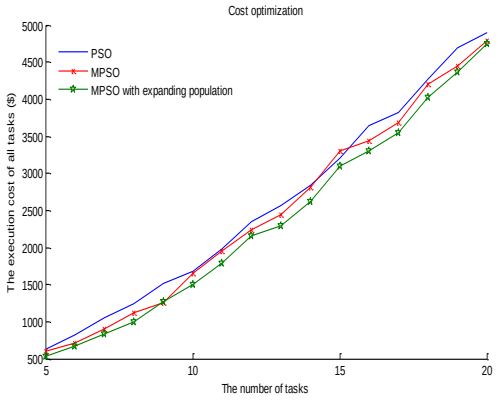



Fig.5 Comparison of PSO algorithm with proposed algorithm of MPSO, MPSO with expanding population with aim of cost optimization

## V. CONCLUSION AND FUTURE WORK

Since users can request time or cost optimization, scheduling systems are obliged to adopt a method for allocating the tasks to resources for realization of demanded parameters of users. Therefore, for minimizing time and cost parameters, two new algorithms have been introduced in this paper. The MPSO leads particles with high fitness toward a better allocation by mutation, and in MPSO with Expanding Population, in order to find better and more answers, through a gradual expansion of population, the existing population gradually increases as a result of which better schemes for scheduling are discovered. It is necessary to note that increasing population leads to increasing calculations and therefore leads to increasing the time of processing. However, both algorithms find better schemes than PSO.

As previously said, since the resource nodes are mobile, limitation of battery capacity is put forth as a new challenge for solving this problem, thus we propose combining Radio frequency identification (RFID) with mobile grid through which nodes of sensors are combined with RFID tags [18]. The integrated tag listens to the RFID reader radio of neighboring nodes. If channel activity is detected, the tag awakens the sensor to listen to the channel and

then receives data through the RF sensor radio. Otherwise, the sensor node can stay in sleep mode.

In case of cost optimization, we hereby suggest that the user specify a price for doing the task [19], so that the proposed algorithm evaluates the particles with regard to it. In future work, we will lay stress on resource load balance of dynamic task scheduling in mobile grid, and consider other resource in mobile grid computing, like, how network quality of service and workflow distribution influence task scheduling.

## REFERENCES


[1] W. Runze, Z. Dandan and C. zhicong, "An Improved Resource Scheduling Algorithm Based on the PSO in Mobile Grid", The 7th International Conference on Wireless Communications, Networking and Mobile Computing(WiCOM) , ISBN: 978-1-4244-6252-0, 2011.

[2] Navimipour, N. J., Rahmani, A. M., Navin, A. H., & Hosseinzadeh, M. (2014). Resource discovery mechanisms in grid systems: A survey. Journal of Network and Computer Applications, 41, 389-410.

[3] D. Li-juan and Y. Zhen-wei, "Scheduling Algorithm with respect to resourceintermittence in Mobile Grid", The 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM) , ISBN: 978-1-4244-3709-2, 2010.

[4] Milani, B. A., & Navimipour, N. J. (2017). A systematic literature review of the data replication techniques in the cloud environments. Big Data Research.

[5] Q. Jiang, X. Wu and H. Yang, "Task Scheduling based on Genetic Algorithm in Mobile Grid", International Conference on Computer Science and Service System(CSSS), Pages 719-722, 2012.

[6] Hirsch, M., Rodríguez, J. M., Mateos, C., & Zunino, A. (2017). A two-phase energy-aware scheduling approach for cpu-intensive jobs in mobile grids. Journal of Grid Computing, 15(1), 55-80.

[7] S. Xue, C. Li, M. Yang and J. Nie, "Grid Resource Scheduling Based on Improved Differential Evolution Algorithms", Sixth International Conference on Natural Computation (ICNC 2010), Volume 8, Pages 4030 – 4034, 2010.

[8] L. Zhang, Y. Chen and B. Yang, "Task Scheduling Based on PSO Algorithm in Computational Grid", Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06), Volume 2, Pages 696 – 704, 2006.

[9] Z. Pooranian, M. Shojafar, J. H. Abawajy and A. Abraham, "An efficient meta-heuristic algorithm for grid computing", Journal of Combinatorial Optimization, DOI 10.1007/s10878-013-9644-6, 2013.

[10] Z. Wei, B. Yan-ping and Z. Ye-qing, "The Application of An Improved Cultural Algorithm in Grid Computing", Control and Decision Conference (CCDC), Pages 4565 – 4570, 2013.

[11] W. Meihong, Z. Wenhua and W. Keqing, "Grid task scheduling based on advanced no velocity PSO", International Conference on Internet Technology and Applications, Pages 1 – 4, 2010.

[12] B. Yan-ping, Z. Wei and Y. Jin-shou, "An Improved PSO Algorithm and Its Application to Grid Scheduling Problem", International Symposium on Computer Science and Computational Technology, Volume 1, Pages 352 – 355, 2008.

[13] 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", International Workshop on Information and Electronics Engineering (IWIEE), DOI 10.1016/j.proeng.2012.01.575,Pages 3808 – 3814, 2012.

[14] J. Kennedy and R. Eberhart, "Particle swarm optimization", IEEE Conference on Neural Networks, Pages 1942-1948, 1995.

[15] J.H. Holland. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1975.

[16] Y. Shi and R.C. Eberhart, "A Modified Particle Swarm Optimizer", In Proceedings of the IEEE Congress on Evolutionary Computation, pages 69– 73, 1998.

[17] B. Soudan and M. Saad, "An Evolutionary Dynamic Population Size PSO Impelementation", In Proc. International Conference on Information and Communication Technologies: From Theory to Applications (ICCTA 2008), Damascus, March 2008.

[18] H. Liu, M. Bolic and I. Stojmenovic, "Taxonomy and Challenges of the Integration of RFID and Wireless Sensor Networks", IEEE Network Magazine, Volume 22, Issue 6, Pages 26-35, 2008.

[19] Q. Xia, W. Sun, Z. Xu and M. Li, "A Novel Grid Resource Scheduling Model Based on Extended Second Price Sealed Auction", Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), Pages 305-310, 2010.


## AUTHOR'S INFORMATION

Mehdi Golsorkhtabaramiri received his M.Sc. in Computer Systems Architecture Engineering. He received his Ph.D. in Computer Engineering from Islamic Azad University Science and Research Branch, , Tehran, Iran. From 2011 until now, he has been working as an Assistant Professor in the Department of Computer Engineering of the Islamic Azad University, Babol Branch, Iran. His current research interests include Radio Frequency Identification (RFID) Systems, Wireless Sensor Network (WSN), Wireless Body Area Network (WBAN) and Named Data Networking (NDN). He has authored and co-authored many high-cited scientific articles published in peer-reviewed conference proceedings and international journals.

Fahimeh Zakeri received her M.Sc. degree in computer software engineering from Islamic Azad University, Babol Branch, Babol, Iran, in 2013. Currently, she is a software engineer in an Accounting software group. Her research interests include intelligent optimization methods, evolutionary algorithms, swarm intelligence, Radio Frequency Identification (RFID) Systems and its application in supply chain and the control systems.