International Journal of the Physical Sciences Vol. 6(30), pp. 6972 - 6980, 23 November, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.858
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

Grid resources selection optimization with quality of service guarantee by a hybrid algorithm of genetic and particle swarm optimization

Hossein Shirgahi[1]* and Amir Masoud Rahmani[2]

[1]Young Researchers Club, Jouybar Branch, Islamic Azad University, Jouybar, Iran.
[2]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.

Accepted 10 August, 2011

When a large number of tasks request different resources in a grid system, resources allocation should have been done with proper planning and scheduling to guarantee a good quality of service (QOS). There are different ways to provide these requests by choosing appropriate resources allocation to optimize the total operation of the system. In this study, first some parameters, such as priority, delay, reliability and cost are determined for each task to maximize system performance and appropriate resources distribution. Then, a hybrid optimization algorithm for choosing grid resource based on genetic algorithm and particle swarm optimization (PSO) is presented. Based on the experimental results, this method's performance is 17.5% more than genetic and PSO algorithms in average.

Key words: Grid system, resources selection, quality of service, genetic algorithm, particle swarm optimization (PSO).

INTRODUCTION

Grid system that is developed in recent years is a network infrastructure. It can integrate distributed resources and it can provide a higher layer distributed resource sharing environment too. Grid system performance changed dynamically by competing resources and uncertainties, and some other factors. One of the three standards for providing quality of service (QOS) is grid systems estimation (Foster, 2005). Consequently, one of the key issues in grid computing is resources selection and allocation based on QOS. However, traditional scheduling methods, such as Backfilling and first come first serve (FCFS) (Hua et al., 2003) cannot be in accordance with grid resources features, such as resource allocation to the programs based on services priority, dynamic resources properties and parallelism of resources allocation, etc.

A scheduling algorithm based on genetic algorithm has been introduced in one study performed in this field (DiMartino and Mililotti, 2004). Its proposal improves the amount of resources usage and permittivity when compared with traditional methods.

Another research is suggested based on simulated annealing (SA) which has made tasks planning on the basis of the amount of latency and reliability (Abraham and Buyya, 2000). It solves some problems like local search and premature convergence well. So, it increases the total system's efficiency. In another study, an optimizing grid resources selection algorithm providing QOS by genetic algorithm simulated annealing (GASA) algorithm is suggested (Ning et al., 2009). This method allocates grid resources to the tasks according to the amount of delay, reliability and the amount of cost. It develops optimizing resource selection based on GASA to increase efficiency. This method is better than the previous ones, because it uses both GA and SA advantages.

In one of the latest studies, we improve the grid resources selection optimization with quality of service (QOS) guarantee by expressing QOS classes and creating priorities based on the amount of delay, reliability and the

*Corresponding author. E-mail: hossein.shirgahi@gmail.com.

amount of cost by using a hybrid of genetic and simulated annealing algorithms (Shirgahi et al., 2010).

This study uses the hybrid of genetic and PSO algorithms (GAPSO) to optimize grid resources selection. Also, it expands eight priority levels for the tasks. Subsequently, QOS parameters in grid system are presented, after which resources selection algorithm was presented. This was followed by the experiments and implementations. Lastly, the results and discussion were expressed.

QOS PARAMETERS IN THE GRID SYSTEM

Describing QOS parameters in the grid

In order to build the model easier, we only consider some important features in QOS and we do not express the rest of the QOS parameters. Of course, other parameters can also be extended as the same method. Considered parameters are as follow:

1. Latency: The time between asking for a request and receiving successful result of that request.
2. Reliability: The success rate of performing tasks by a resource.
3. Cost: The cost paid by a user when he/she needs a resource.
4. Priority: Tasks have priorities based on their importance and type of their works.

So, they need different services, the scheduler should allocate resources to tasks by the priority levels; therefore, QOS can be expressed with four parameter.

Measuring QOS parameters in the grid

Delay has three parts: network transfer time (ttran), system scheduling time (tsch) and computation time (tcom). Total delay is obtained based on Equation 1.

$$q_{delay} = t_{tran} + t_{sch} + t_{com} \qquad (1)$$

nth time reliability of a requested service may be computed based on n-1 previous times, of which a request has been performed successfully. According to Equation 2:

$$(q_{reliabiliy})_n = \frac{\sum_{i=1}^{n-1}(q_{reliabiliy})_i}{n-1} \qquad (2)$$

The reliability of each system is dependent on the failure rate (□). In this study, we suppose the failure distribution function is exponential. Therefore, the systems reliability is as Equation 3.

In this Equation, $\Delta t = t_C - t_0$, tc is the current time and t0 is the initial run time of a system.

$$R(t) = e^{-\lambda \Delta t} \qquad (3)$$

Cost is a feature which is determined directly by resources considering different users.

Nowadays, with a fast growth of networks, some concepts such as increasing need of bandwidth and simultaneous support of different classes of services will be important. Therefore, QOS has turned to a key concept in network usage and services. Tasks have different priority levels of QOS according to their requested QOS. In this study, eight levels of priorities are considered (Table 1).

As the priority levels increases, delay will decrease, but reliability will increase to provide requested QOS. Therefore, the importance of low cost will be ignored.

QOS STANDARDIZING

The aforementioned criteria can be divided into three classes (Zeng et al., 2004). The delay that will decrease by increasing QOS can be standardized by Equation 4. Reliability has a direct relationship with QOS value. It can be standardized according to Equation 5 and the cost can be standardized according to Equation 6. In these Equations, i represents the ith request service and j represents the jth resource.

$$V(q_{delay})_{i,j} = \begin{cases} \dfrac{\max(q_{delay})_i - (q_{delay})_{i,j}}{\max(q_{delay})_i - \min(q_{delay})_i} & \text{if } \max(q_{delay})_i - \min(q_{delay})_i \neq 0 \\ 1 & \text{if } \max(q_{delay})_i - \min(q_{delay})_i = 0 \end{cases} \qquad (4)$$

$$V(q_{reliabiliy})_{i,j} = \begin{cases} \dfrac{(q_{reliabiliy})_{i,j} - \min(q_{reliabiliy})_i}{\max(q_{reliabiliy})_i - \min(q_{reliabiliy})_i} & \text{if } \max(q_{reliabiliy})_i - \min(q_{reliabiliy})_i \neq 0 \\ 1 & \text{if } \max(q_{reliabiliy})_i - \min(q_{reliabiliy})_i = 0 \end{cases} \qquad (5)$$

Table 1. QOS priority levels with the importance amounts of delay, reliability and cost parameters.

Priority level	Service type	delay parameter weight	Reliability parameter weight	Price parameter weight
0 0	Best Effort Best effort	0.2 0.2	0.2 0.2	0.6 0.6
1 1	Background Background	0.25 0.25	0.25 0.25	0.5 0.5
2 2	Standard Standard	0.3 0.3	0.3 0.3	0.4 0.4
3 3	Excellent Load Excellent load	0.35 0.35	0.35 0.35	0.3 0.3
4 4	Controlled Load Controlled load	0.38 0.38	0.42 0.42	0.2 0.2
5 5	Voice and Video Voice and video	0.4 0.4	0.45 0.45	0.15 0.15
6 6	Layer 3 Network Control Reserved Traffic Layer 3 network control reserved traffic	0.42 0.42	0.48 0.48	0.1 0.1
7 7	Layer 2 Network Control Reserved Traffic Layer 2 network control reserved traffic	0.44 0.44	0.51 0.51	0.05 0.05

$$V(q_{price})_{i,j} = q_j \tag{6}$$

Now, the importance of delay, reliability and cost parameters must be determined according to their weights based on QOS priority levels in Table 1.

So, the total cost of service which the ith service can receive from jth resource will be computed based on Equation 7.

$$V(q)_i = \omega_{delay} * V(q_{delay})_{i,j} + \omega_{reliability} * V(q_{reliability})_{i,j} + \frac{\omega_{price}}{V(q_{price})_{i,j}} \tag{7}$$

In Equation 7, the total weights for delay, reliability and the cost parameters are equals to 1.

DESCRIPTION OF THE RESEARCH STRUCTURES

We assume that there are n resources as R = {R1,R2,…,Rn} and m tasks as T = {T1,T2,…,Tm }. We make $\sum = (R,T,Q,A)$ quadric. R is resources set, T is tasks set, Q is a m × n matrix (m rows and n columns), qij represents a QOS which Ti is derived from Rj and A is a 1 × m task allocation matrix. The total services cost is computed according to Equation 7. The main goal of this model is to determine a resources allocation policy to maximize the evaluation function in the system.

Resources selection algorithm

An introduction on genetic algorithm (GA)

GA is a tool helping us to simulate the evolutionary development mechanism. The simulation has been developed by searching in the problem space to find a suitable answer, but no optimum one is necessarily (Dianati and Song, 2002). GA contains following steps:

1. Encoding: This process is an important part of solving problems by GA. Often, when GA is used to solve the problems, it is difficult to find a suitable answer which solves all problems respective to solving the problem by GA. This process leads to generation of the population, searching for the response and chromosome presentation of problem answers like binary string, alphabetic strings and permutation (Danesh et al., 2010). In this study, an indirect encoding method is used. This method determines the resource, which each task receives services. In this structure, the length of chromosome is equal to tasks frequency in the system. The value of each chromosome bit determines the issue that serviced the resource. The chromosome structure is shown in Figure 1.
2. Start: In this step k chromosomes are generated. K is chromosomes frequency depending on the problem nature. The production of initial chromosomes is randomized.
3. Valuation: After producing an initial population, a value should be assigned to each member of the population. It is performed by an evaluation function. It is very important to find a suitable evaluation method and a proper evaluation function. The evaluation function is based on formula of Equation 8.

$$F = \sum_{i=1}^{m} V(q_i) \tag{8}$$

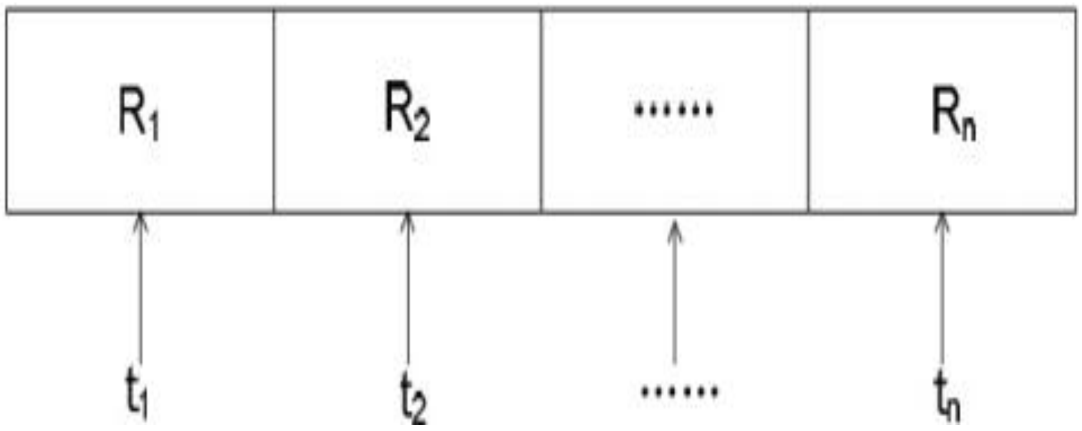


Figure 1. Overall structure of a chromosome.

4. New population production: This is one of the most important steps of solving a problem by GA. Since the selection of initial population and its dispersal in the problem searching space cannot find a new point in the searching space and it cannot lead the searching space to the optimum answer, by using some genetic operations, such as mutation and crossover based on evolutional system in nature, a new population using the initial population is developed in this stage, then it is placed in the searching space. The major operation in this important stage is to create some new population with k initial population and relevant value of each of them by using following steps:

a. Selection operation: According to Darwin's theory that the best will survive, reproduce and generate new individuals, this theory will be applied to solve problems through GA. Therefore, two chromosomes from the population are chosen to do crossover operation and to generate new children. Selection is an important operation. There are different methods to do this operation like random selection, roulette wheel and rank power (Shirgahi et al., 2010). In this study, we calculated the probability of each chromosome being a parent based on each chromosome value and rank power method using Formulas 8 and 9. It is supposed that the population is equal to k. In rank power method, value 1 is assigned to the worst chromosome; value 2 is assigned to the second worst one, up to value k which is assigned to the best chromosome. Finally the probability of selecting ith chromosome from Equations 9 and 10 may be estimated. Therefore, a few numbers from 1 to F will be assigned to each chromosome based on its probability. Now a number between 1 and F can be accidentally selected and the chromosome which has this number will be chosen.

$$F = 1^2 + 2^2 + 3^2 + ... + K^2 \tag{9}$$

$$\frac{Rank_i^2}{F} \tag{10}$$

b. Crossover operation: This operation leads to generating new population and distributing it in the problem space. This process is derived from individuals' genetic and it causes variety of features and appearances among them. Crossover operation leads to population dispersion in the searching space in GA. This operation is done by a special probability like α. It means individuals take part in crossover operation with the probability of α. In crossover operation, two chromosomes are selected by some methods mentioned in earlier. Now, these two chromosomes are crossed with each other to generate two new children (chromosomes) and they are put in the population. In Figure 2, initial searching population and secondary population after applying crossover operation are shown. There are different methods to perform crossover, including one-point, two-point, multiple-point and steady (Danesh et al., 2010). The sample of two-point crossover is as shown in Figure 3.
c. Mutation operation: It is an unwanted change in a DNA string coding that can be useful most of the time, but sometimes harmful too. The searching population can lead to a local minimum in GA, where mutation operation helps in achieving global optimum answer and it keeps the population far from the local minimum through a mutation in the chromosome. This method is a random way to save the chromosome from an undesirable situation (Figure 4). There are different methods to do this operation, such as one-point, two-point and multiple-point mutation (Danesh et al., 2010). The sample of multiple-point mutation is as shown in Figure 5.
5. Substitution: The initial population could be generated and put in the problem space by using GA in previous steps. If this operation lasts too long, the population will be highly increased. Therefore, the population individuals

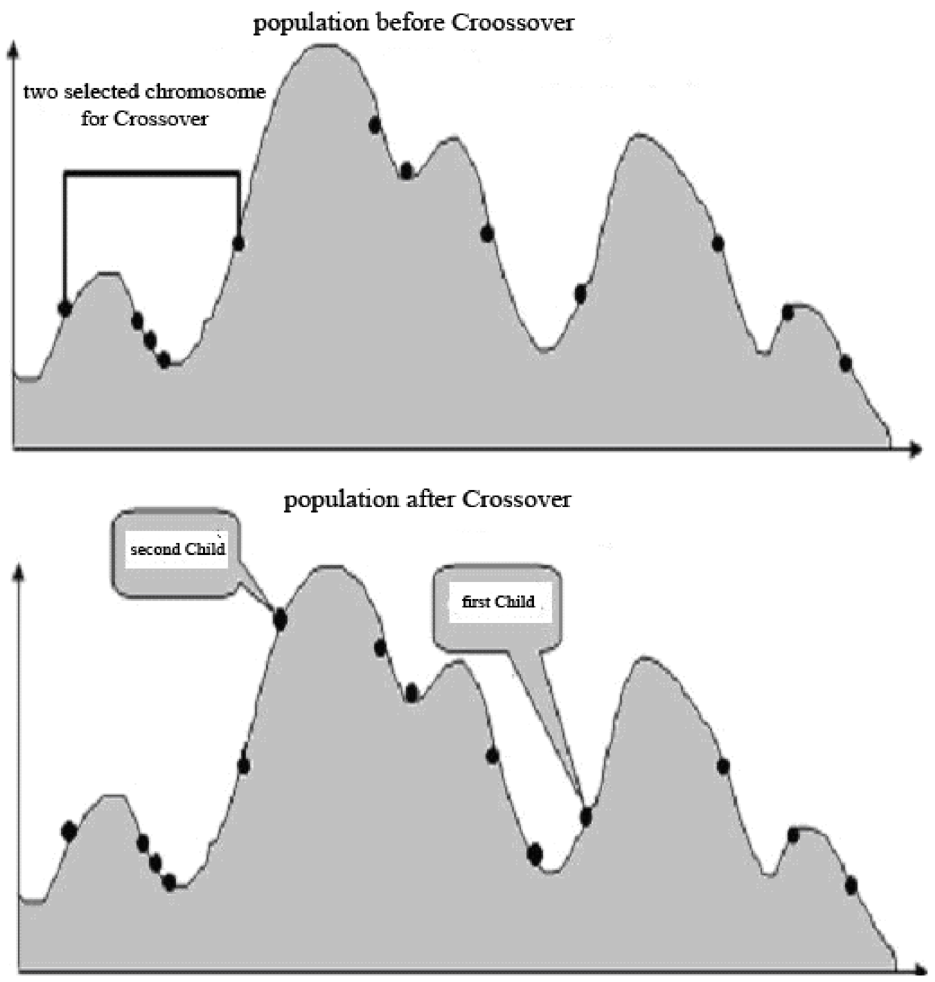

population before Croossover
two selected chromosome for Crossover
population after Croossover
second Child
first Child

Figure 2. A view of crossover effect on population dispersion.

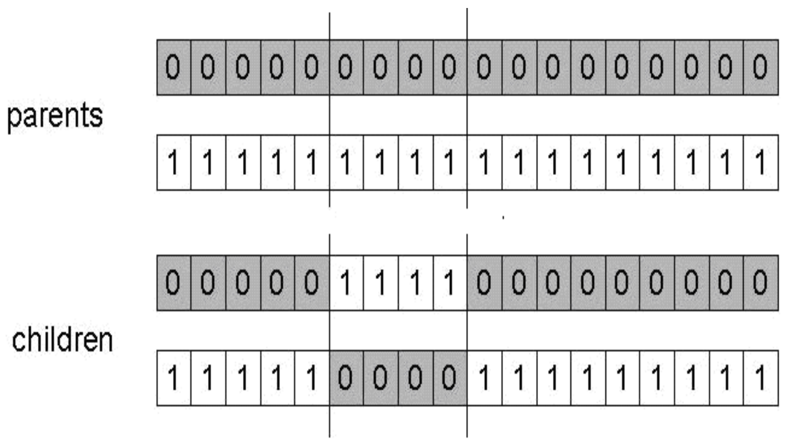


Figure 3. Two-point crossover.

will be reduced to initial population K after creating a new generation. The current population is sorted by its fitness value and K most valuable ones will be selected to be used as an initial population for the later generation

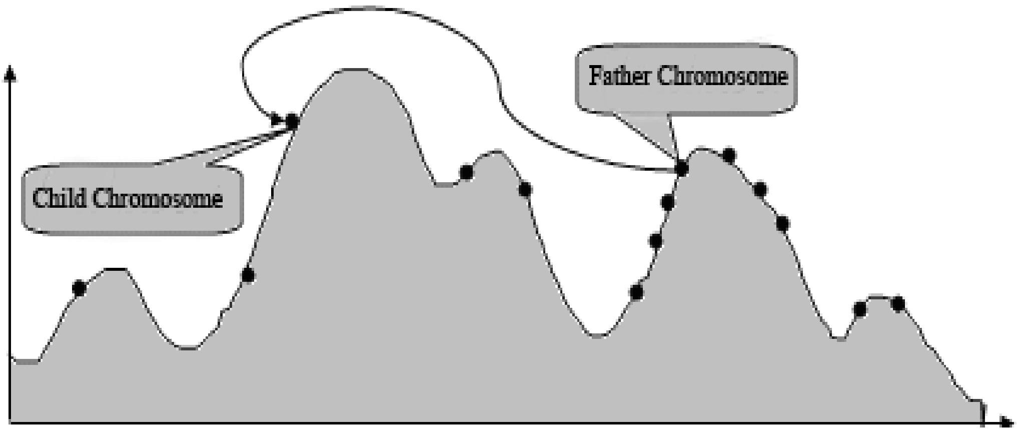


Figure 4. An overall view of the mutation operation.

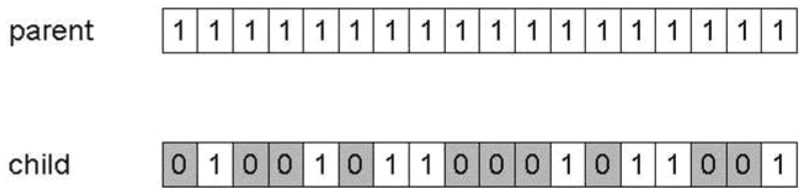


Figure 5. Multiple-point mutation.

and searching space.

6. Evaluation: The way of finding answers in GA differs from other algorithms, because GA tries to suggest some answers, even though they are not optimal, and it continues working instead of trying to find the best answer at first. If the answers could satisfy user's requests and problem needs, the algorithm will be stopped, otherwise it continues working again. In this study we determined the running iterations at first. So, the problem could be iterated for the particular times.

An introduction to the particle swarm optimization (PSO)

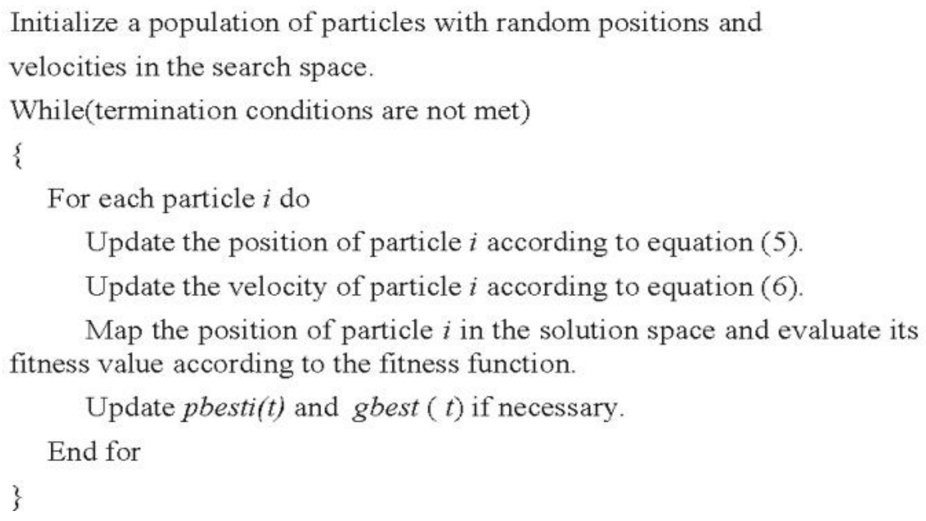
PSO method is expanded by Kennedy and Eberhart in 1995. Also, it is used successfully in many fields of science and functional so far. PSO is an optimization algorithm based on the population in which every individual is considered as a particle and every population composed of some of these particles.

Problem solving space is considered as searching space in PSO and every location in the searching space is a solution related to the problem. The particles of the population try together to find the best position (the best solution) in the searching space (solution space). Also, each particle moves in accordance with its speed. Each particle's movement in each iteration is calculated according to Equation 11.

$$x_i(t+1) \leftarrow x_i(t) + v_i(t) \qquad (11)$$

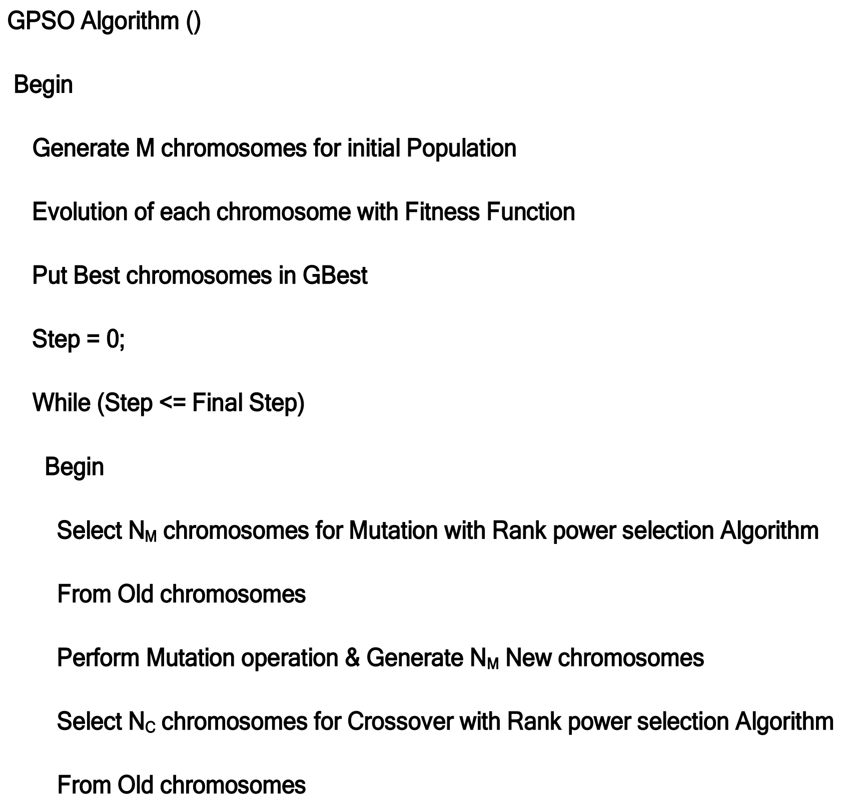
$$v_i(t+1) \leftarrow \omega v_i(t) + C_1 rand_1 (Pbest_i(t) - x_i(t)) + C_2 rand_2 (Gbest(t) - x(t)) \qquad (12)$$

In Equations 11 and 12, xi(t) is the position of particle i and vi(t) is its velocity at time t. Pbesti(t) is the best position that has been found by particle i itself so far and Gbest(t) is the best position that has been found by the total population so far. ω is an inertia weight that is a ratio of previous speed and c1 and c2 are acceleration coefficients which determine the effect of best position of any particle and best global position. Also, rand1 and rand2 are random variables between 0 and 1. PSO algorithm routine is as shown in Figure 6. In this article, we apply PSO algorithm in crossover and mutation operations of GA algorithm to avoid local optimum and local premature end. The details are expressed in subsequently.



```
Initialize a population of particles with random positions and
velocities in the search space.
While(termination conditions are not met)
{
    For each particle i do
        Update the position of particle i according to equation (5).
        Update the velocity of particle i according to equation (6).
        Map the position of particle i in the solution space and evaluate its
fitness value according to the fitness function.
        Update pbesti(t) and gbest ( t) if necessary.
    End for
}
```

Figure 6. Pseudo code of PSO algorithm.



```
GPSO Algorithm ()
 Begin
   Generate M chromosomes for initial Population
   Evolution of each chromosome with Fitness Function
   Put Best chromosomes in GBest
   Step = 0;
   While (Step <= Final Step)
     Begin
       Select $N_M$ chromosomes for Mutation with Rank power selection Algorithm
       From Old chromosomes
       Perform Mutation operation & Generate $N_M$ New chromosomes
       Select $N_C$ chromosomes for Crossover with Rank power selection Algorithm
       From Old chromosomes
```

Figure 7. Pseudo code of GPSO algorithm.

Selection optimization designing under GAPSO

The overall structure of selection optimization designing under GPSO is as shown in Figure 7. The following steps express selection optimization designing under GAPSO based on GPSO algorithm:

1. We consider an initial population K randomly.
2. ith cell structure of a chromosome which is equivalent to ith task in the system shows a particle. We should find

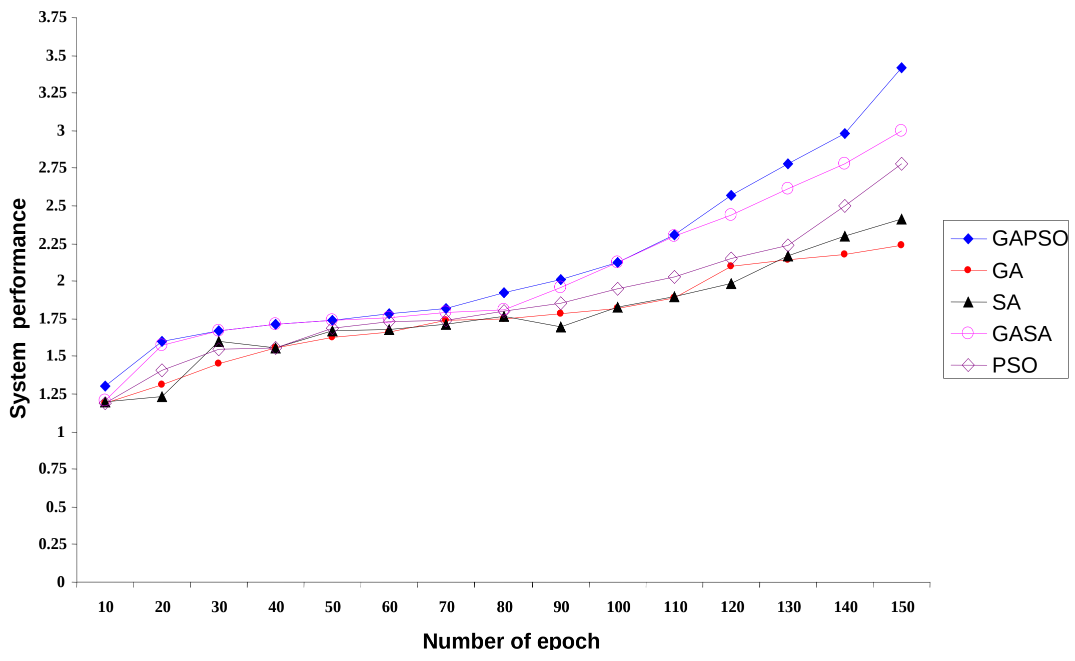


Figure 8. The relationship between iterations and system performance.

the best position which task i has a better position in that system among all K initial chromosomes for each cell i, and then we place it in Pbesti(0).

3. We calculate all chromosomes ‘value based on the evaluation function according to Equation 7 at this step. Then, we sort them in descending order. We put the chromosome with the highest value in Gbest(0). Also, we assign vi(0) values randomly at first.
4. We choose chromosome for mutation and crossover operations and PSO algorithm according to rank power method.
5. We do mutation and crossover operations and we compare Pbesti and Gbest values of all newly created population. If the result is better, we register them.
6. In addition to mutation and crossover operations, we consider the chromosomes which were selected for PSO algorithm, and then we calculate new chromosomes based on Equation 11 and we compare Pbesti and Gbest values for all newly created population. If the result is better, we register them.
7. We select K population with more valuable evaluation function at the end of each step for the next step. We continue these steps as much as the iterations which were specified first.
8. Finally, the Gbest value will be the desirable answer of the proposed algorithm.

IMPLEMENTATIONS AND EXPERIMENTS

The method of this study is implemented by Delphi 7 in windows XP. This application was able to achieve up to 200 resources and 1000 tasks independently. The tasks, resources and their parameters were achieved randomly. The randomly considered ranges for these parameters are as follow: the resources reliability is determined by a random determination of and t0 and tc times of the systems.

- Tasks priority levels: 0 to 7
- Resources cost: 10 to 500.
- Tasks running time: 10 to 250.
- Delay: 1 to 200.

We test the expressed method by using GA, SA, PSO, GASA and GAPSO algorithms with different numbers of implementation. The test results can be observed in Figure 8.

RESULTS AND DISCUSSION

Based on the experimental results, performance of GAPSO algorithm is more than GA, SA and PSO algorithms. It has 18, 11 and 8.5% more optimal answers than GA, SA and PSO algorithms, respectively. Also, the performance of GAPSO algorithm is 3.5% more than GASA algorithm. It should be noted that whatever the number of proposed algorithm iterations is, the desirable answers will be more than other methods. A proposal for future work in this area is suggested as follow:

1. Combining some other evolutionary development methods, such as GELS and Ant Colony with the presented method to develop the proposed method.
2. Considering more parameters to determine QOS, such as security, deadline and usage of the proposed method.

ACKNOWLEDGEMENTS

This work was supported by a research funding from the

Islamic Azad University, Jouybar Branch, Iran. We thank the Islamic Azad University, Jouybar Branch for their financial support.

REFERENCES

Foster I (2005). What is The Grid? A Three Point Checklist.http://www.fp.mcs.anl.gov/ Foster [EB/OL].

Hua YQ, Yi L, Dan M (2003). A Simplified Backfilling Algorithm Based on First Fit-RB-FIFT[J], Comput. Eng. Appl., 39(2): 70-74.

DiMartino V, Mililotti M (2004). Sub-optimal scheduling in a grid using genetic algorithms, Parallel Comput., 30: 553-565.

Abraham A, Buyya R (2000). Nature's heuristics for scheduling jobs on computational grids, In The 8th Int'l Conf on Advanced Computing and Communications (A-DCOM 2000), Cochin, India, pp. 45-52.

Ning Q, Binqiang W, Shuai W (2009). QoS-aware GASA:A QoS-aware Grid Resource Selection Optimization Algorithm, In The 15th Int' Conf on Advanced Computing and Communications, China.

Zeng LZ, Benatallah BA, Ngu HH (2004). QoS- aware middleware for web services composition, IEEE Trans. Software Eng., 30(5): 311-327.

Dianati M, Song I (2002). An Introduction to Genetic Algorithms and Evolution Strategies, University of Waterloo, Canada.

Danesh N, Shirgahi H, Motameni H (2010). Optimizing N relations join queries by genetic algorithm. Sci. Res. Essays, 5(13): 1576-1582.

Shirgahi H, Danesh M, Danesh N (2010). Grid resource selection optimization with guarantee quality of service by hybrid of genetic and Simulated Annealing algorithms, Proceedings of the World Congress on Engineering and Computer Science 2010, WCECS 2010, San Francisco, USA, pp. 1104-1108.