2011 International Conference on Future Information Technology
IPCSIT vol.13 (2011) © (2011) IACSIT Press, Singapore

Hybrid PSO for Independent Task scheduling in Grid Computing to Decrease Makespan

Z. Pooranian [1, a], A. Harounabadi [2,b], M. Shojafar [3,c] and J. Mirabedini [4,d]

[1] Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran

[2] PhD in Computer Eng- Software, Islamic Azad University, Central Tehran Branch, Department of computer, Tehran, Iran

[3] Department of Electrical and Computer Eng., Qazvin Islamic Azad University, Qazvin, Iran

[4] PhD in Computer Eng- Software, Islamic Azad University, Central Tehran Branch, Department of computer, Tehran, Iran

[a] Zahra.pooranian@gmail.com , [b] A.Harounabadi@gmail.com , [c] Shojafar@qiau.c.ir, [d]Jvd2205@yahoo.com

Abstract. Task assignment Problem in Grid Computing is a NP-Complete Problem that has been studied by several researchers. The broker undertake scheduling applications belonged to user should have an efficient scheduling algorithm. In this paper, a new optimized scheduling hybrid algorithm was presented named GPSO. GPSO is composed of both Particle Swarm Optimization and GELS algorithms. Since PSO is weak on local search, GELS was used for improve it and avoid becoming trapped in a local optimum. GPSO is done independent Task scheduling to decrease Makespan and minimize the missed tasks. The experimental results showed Effectiveness of the proposed combined approach in finding optimal solutions.

Keywords: Grid Computing, Particle Swarm Optimization (PSO), Scheduling, Independent Task.

1. Introduction

The main idea of grid is that participating machines resources are used through software layer as transparent and reliable. This software layer is responsible for resource virtualization, discovery, searching of resources and the management of running applications [1].Task scheduling is a challenging problem in grid computing environment.

Heuristics optimization algorithm is widely used to solve a variety of NP-complete problems. In [2], the author proposed a new method of scheduling in grid based on Heuristic Algorithms. Moreover, Abraham et al [3] presented three basic heuristics implied by nature for grid scheduling, namely Genetic Algorithm (GA) [4], Simulated Annealing (SA) [5] and Tabu Search (TS) [6], and heuristic derived by a combination of their three algorithms. Particle Swarm Optimization (PSO) [7] is one of the latest evolutionary optimization techniques inspired by nature. It also has fewer algorithm parameters than both GA and SA. Furthermore, PSO works well on most global optimal problems. But, since the ability of local search in PSO is weak and also the possibility of becoming trapped in the local optimum is high, in this paper, its combination GELS which is a local search algorithm is used to improve its performance in finding solution. The Proposed Hybrid algorithm (GPSO) is decreased makespan and minimized missed tasks.

2. Related Works

In 2004 [8], a combined optimization algorithm was presented for task scheduling problem in which the combination of PSO and SA is used. A particle consists of m segments and every segment has n different job numbers, representing the processing orders of n jobs on m machines.

Thus, we should have m machine and n job, to convert continuous optimization problem to the discrete optimization problem. Round off the real optimum values to its nearest integer number. In Proposed Algorithm, results of PSO are given to SA to avoid trapping in a local minimum. In [9], Combination of SA and PSO have been used for scheduling independent tasks in a way that it is used dynamically varying inertia. It will provide a balance between the global exploration and local exploration. And results to less iterations on average to find a sufficiently optimal solution. DPSO algorithm in [10] has presented for task scheduling in grid systems. The scheduler aims at minimizing makespan and flow time simultaneously. In position matrix each column represents a job allocation and each row represents allocated jobs in a node.

3. PSO Algorithms

Particle Swarm Optimization (PSO) introduced by Kennedy and Eberhart [7] in 1995. In a PSO system, a swarm of individuals (called particles) fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself i.e. its own experience and the position of the best particle in its neighborhood i.e. the experience of neighboring particles. When the neighborhood of a particle is the entire swarm, the best position (pbesti) in the neighborhood is referred to as the global best(gbesti) position of the particle, and the resulting algorithm is referred to as the PSO. When smaller neighborhoods are used, the algorithm is generally referred to as the best PSO.

The performance of each particle is measured using a fitness function that varies depending on the optimization problem [11]. Each particle in the swarm is represented by the following characteristics. xi is the current position of the particle, vi is the current velocity of the particle and yi is the personal best position of the particle. During each PSO iteration, particle i adjusts its velocity Vi+1 and position vector particleij through each dimension j by referring the random multipliers, either the personal best vector (pbesti) and the swarm's best vector (gbesti, if the global version is adopted). If global version is adopted, the equations 1 and 2 are used.

$$V_{i+1} = W\,V_i + C1\,rand1(pbest_i - X_i) + C2\,rand2(gbest_i - X_i) \tag{1}$$

$$X_{i+1} = X_i + V_{i+1} \tag{2}$$

Where C1 and C2 are the cognitive coefficients and rand1 and rand2 are random real numbers drawn from U (0, 1). The inertia weight can be dynamically varied by applying an annealing scheme for the w-setting of the PSO, where w decreases from W = 0.9 to W = 0.1 over the whole run. In general the inertia weight W is set according to the following equation 3.

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} * iter \tag{3}$$

Gravitational Emulation Local Search (GELS) Algorithm

Voudouris and his colleagues [12] for the first time in 1995 were suggested that GLS algorithm for searching in a searching space and NP-hard solution. In 2004, Vebster [13] presented it as a strong algorithm and called it as GELS algorithm. This algorithm based on gravitational attraction and by using of this process is mimic for searching within a searching space. Obtained neighbors in each neighbor group called dimension. For each dimension, was defined a primary velocity and each dimension has much primary velocity and more apparent response for problem. GELS algorithm accounted gravitation force within responses in a searching space in two methods. In the first method, a response selected from local neighbor space of current response and gravitation force accounted within these two response. In the second method, gravitation force was accounted within all of the neighbor responses in a neighbor space of current response and it is not limited to one response. GELS algorithm uses the formula for the gravitational force between the two solutions as

$$F = G\left(\frac{CU - CA}{R^2}\right) \tag{4}$$

CA and CU are candidate response and current response alternatively. G is constant amount 6.672 and R is the neighbor radius of two things parameters in searching space.

PSO Algorithm for Solving Task Scheduling Problem

The Encoding Scheme and Initial Swarm. One of the key issues in applying PSO successfully to job scheduling is how to encode a schedule to a search solution, i.e. finding a suitable mapping between problem solution and PSO particle. In the proposed method each particle represents a feasible solution for task assignment using a vector of n elements, and each element is an integer value between 1 to m, that are produced randomly, Fig. 1 illustrates how 4 tasks are allocated to 4 resources, as in particle 1, task 4,T4, is assigned to resource 4.

	T1	T2	T3	T4
Particle 1	1	3	2	4
Particle 2	3	1	4	2

Fig. 1. Particle presentation.

Fitness Evaluation. Initial population generate randomly, then, it should be define velocity vector between *[-Vmax, Vmax]* for each particle randomly, after that, it use fitness value for evaluating. Basic purpose of task scheduling is that it could minimize makespan. It is noted that this time always should be parallel with or smaller than *Max Deadline (MD)* within all tasks. In suggested method, the solution is move appropriate for task scheduling problem that in addition to decreasing makespan, also the number of missed tasks are minimized in it. Equation 5 showed the accounting of *fitness* function. Here, *miss_task* is the number of tasks which was missing in $particle_i$

$$Fitness = \frac{1}{makespan} + \frac{1}{miss\_task * MD} \tag{5}$$

Modifying Particles. particles velocity and position are updated using equation 1 and 2. After generating new positions may achieve real values for particle positions such as 2.25. It is meaningless for resource number. Therefore, in the algorithm we usually round off the real values to its nearest number. By this way, we convert a continuous optimization problem to a discrete optimization problem.

Hybrid PSO algorithm

PSO Algorithm use several search points that these points are close to the optimum point with their pbests and gbest. PSO can be used for continues and discrete problems and it has good ability for global searching in problem space. But since its ability is weak in local search and there is the probability of becoming trapped in a local optimum.The combined PSO and GELS is used to resolve PSO disadvantages in the proposed method. A new hybrid algorithm of PSO and GELS, named GPSO, is presented in Fig. 2 it can be seen that PSO provides initial solution for GELS during the hybrid search process.

Implementation experimental results

For simulation has been used of java software on hardware with these characteristics: CPU 2.66GHZ, 4GB RAM, Win XP operating system. In suggested algorithm has been assumed $C_1=C_2=1$ and maximum velocity V_{max} is set to m(number of resources). Simulation conclusions about comparison within suggested algorithm (GPSO) and GELS, PSO algorithm has been shown in figures 3, 4. In Figure 3, shown a diagram which was scheduled the number of tasks within 20 and 60 on 20 resources by using of these algorithms. As shown, if the number of tasks increased, Makespan is increased too. Within scheduled algorithm was showed that the suggested algorithm produced less Makespan than the other.

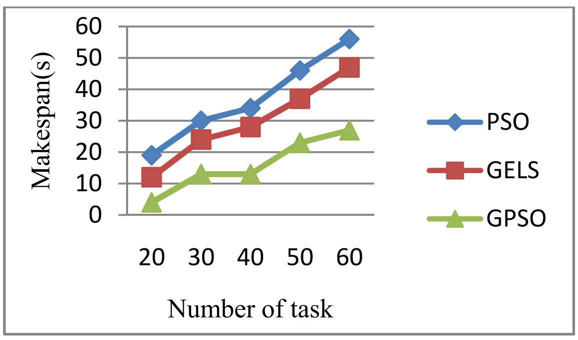

Fig.3. Comparison of Makespan

In Fig. 4, algorithms have been comprised together with a view to become miss rate of tasks. In designed diagram, the fitness amounts correspond to the rate of missed tasks. Since diagram is shown, whatever fitness amount increased, the rate of missed tasks decreased. It means that the number of missed tasks decreased as a result of completion their Makespan. The figure shown that GPSO have less missed tasks than above mentioned algorithms.

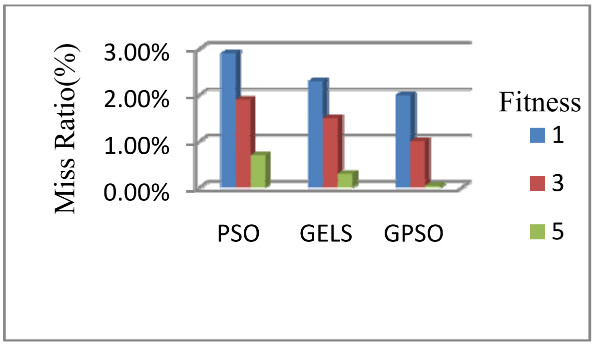

Fig. 4. Comparison the average miss ratio of different fitness functions algorithms.

4. Conclusion

This paper presents a novel task scheduling method based on hybrid PSO and GELS (GPSO) algorithm to solve grid scheduling problem to minimize makespan and missed task. Each particle represents a feasible solution. The position vector is transformed from the continuous values to the discrete values based on round off real values. The hybrid PSO performs better the local search. Because GELS algorithm rather than the other local searching algorithms such as hill-climbing, SA,...is searching space problem well and find better solutions. The performance of the proposed method is compared with existing method. From the simulated experiment, the result of GPSO algorithm is better than other algorithms.

5. References

[1] B. Yan-ping, ZH. Wei and Y. Jin-shou, "An Improved PSO Algorithm and Its Application to Grid Scheduling Problem", International Symposium on Computer Science and Computational Technology, (2008).

[2] M. Shojafar, S. Barzegar, M. R. Meybodi, "A new Method on Resource Scheduling in grid systems based on Hierarchical Stochastic Petri net", First International Conference on Information, Networking and Automation (ICINA 2010), Vol. 9, No 2,pp. V9-175-180, China, 2010.

[3] Abraham, R. Buyya and B. Nath, "Nature's Heuristics for Scheduling Jobs on Computational Grids", The 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000), Cochin, India, December, (2000), pp. 45-52.

[4] Y. Gao, H.Q Rong and J.Z. Huang, "Adaptive grid job scheduling with genetic algorithms", Future Generation Computer Systems, Elsevier, (2005), pp.1510-161.

[5] J.E. Orosz and S.H. Jacobson, "Analysis of static simulated annealing algorithm", Journal of Optimization theory and Applications, Springer, 115 (1) (2002) , pp. 165- 182.

[6] R. Braun, H. Siegel, N. Beck, L. Boloni, M. Maheswaran, A. Reuther, J. Robertson, M. Theys, B. Yao, D. Hensgen and R. Freund, "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", J. of Parallel and Distributed Computing, vol.61, No. 6, (2001), pp. 810-837.

[7] Kennedy J. and Eberhart R. Swarm Intelligence, Morgan Kaufmann, (1995).

[8] X. Weijun, W. Zhiming, Zh. Wei, and Y. Genke, “A New Hybrid Optimization Algorithm for the Job-shop Scheduling Problem”, Proceeding of the 2004 American Control Conference Boston, Massachusetts June 30 - July 2, (2004).

[9] S. N. Sivanandam, P.Visalakshi, “Multiprocessor Scheduling Using Hybrid Particle Swarm Optimization with Dynamically Varying Inertia”, International Journal of Computer Science & Applications Vol. 4 Issue 3, (2007), pp 95-106.

[10] A. Izakian, B. Tork Ladani2, K. Zamanifar2, A. Abraham, “A Novel Particle Swarm Optimization Approach for Grid Job Scheduling”, Communications in Computer and Information Science, Volume31, (2009), pp. 100-109.

[11] G. Ritchie, “Static Multi-processor scheduling with Ant Colony Optimization and Local search”, Master of Science thesis, University of Edinburgh, (2003).

[12] Voudouris, chris, Edward Tsang, Guided Local Search. Technical Report CSM-247, Department of Computer Science, University of Essex, UK, August (1995).

[13] Barry Lynn Webster, “Solving Combinatorial Optimization Problems Using a New Algorithm Based on Gravitational Attraction”, Ph.D. Thesis, Florida Institute of Technology Melbourne, FL, USA, May 2004.

```
Begin
  Step 1. Initialization
     1)      PSO
               •   Initialize swarm size, each particle’s position and velocity;
               •   Evaluate each particle’s fitness;
               •   Initialize gbest position with the lowest fitness particle in the swarm;
               •   Initialize pbest position with a copy of particle itself;
               •   Initialize W_max, W_min, C1=C2=1, max generation and generation=0;
     2)      GELS
               •   Initialize R; V= Set initial Velocity_Vector;
  Step 2. Computation
     1)      PSO
             Do while(the maximum of generation is not met){
                Generate new swarm by equation (1) and (2);
                Evaluation swarm{
                      Find new gbest and pbest;
                      Update gbest of swarm and pbest of particle;   }
     2)      GELS
             Current_Solution=gbest;
             Best_Solution =Current_Solution;
             Direction= select Direction(V);
             While( i< max generation or Velocity_Vector[…]≠0){
                 Candidate_Solution= make neighbor(direction)
                Force=6.672* (fitness(Candidate_Solution)−fitness(Current_Solution)) / R^2
                If( fitness(Candidate_Solution)> fitness(Current _Solution))
                      Best_Solution = Candidate _Solution;
                Update Velocity_Vector(V,Force)
                 Direction= select Direction(V);  }}

   Step 3. Return Best_Solution;
End
```

Fig. 2. The hybrid optimization algorithm GPSO