Multi-objective Workflow Grid Scheduling Based on Discrete Particle Swarm Optimization

Ritu Garg and Awadhesh Kumar Singh

Computer Engineering Department, National Institute of Technology,
Kurukshetra, Haryana, India
ritu.59@gmail.com, aksinreck@rediffmail.com

Abstract. Grid computing infrastructure emerged as a next generation of high performance computing by providing availability of vast heterogenous resources. In the dynamic envirnment of grid, a schedling decision is still challenging. In this paper, we present efficient scheduling scheme for workflow grid based on discrete particle swarm optimization. We attempt to create an optimized schedule by considering two conflicting objectives, namely the execution time (makespan) and total cost, for workflow execution. Multiple solutions have been produced using non dominated sort particle swarm optimization (NSPSO) [13]. Moreover, the selection of a solution out of multiple solutions has been left to the user. The effectiveness of the used algorithm is demostrated by comparing it with well known genetic algorithm NSGA-II. Simulation analysis manifests that NSPSO is able to find set of optimal solutions with better convergence and uniform diversity in small computation overhead.

1 Introduction

With the rapid development of networking technology, grid computing [4] has emerged as a promising distributed computing paradigm that enables large-scale resource sharing and collaboration. One of the key challenges of heterogeneous systems is the scheduling problem. Scheduling of computational tasks on the Grid is a complex optimization problem, which may require consideration of several scheduling criteria. Usually, the most important criterion is the task execution time, cost of running a task on a machine, reliability, resource utilization etc.

The optimization of scheduling problem is NP-complete, so numerous heuristic algorithms have been proposed in literature [3]. Many heuristics have also been proposed for workflow scheduling in order to optimize a single objective [9], [10], [16], [20]. To achieve better solution quality, many meta-heuristic methods have been presented for job scheduling such as simulated annealing (SA) [2], genetic algorithm (GA) [15], ant colony optimization (ACO) [14] and tabu search[6].

Defining the multiple objectives for the task scheduling problem for generating efficient schedules at reduced computational times are of research interest in the recent days. For multi objective optimization to independent task scheduling [1] used the fuzzy particle swarm optimization and [11] used the discrete particle swarm optimization. These methods combine the multiple objectives into scalar cost function

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 183–190, 2011.

using the weight factors, which convert the problem into single objective problem prior to optimization. Generally, it is very difficult to accurately select these weights as small perturbations in weights leads to different solutions.

Hence, in this paper we introduced the multi objective optimization approach based on discrete particle swarm optimization to generate Pareto optimal solutions, which is the set consisting of all non-dominated solutions. A solution is called non-dominated solution if it is best at least in one objective with respect to others. Pareto optimal solutions are preferred to single solution in real life applications. In this paper, we considers the two objectives for task scheduling keeping in view the tradeoff between two conflicting objectives of minimizing the makespan and total cost under the specified deadline and budget constraint. Towards the goal of obtaining Pareto optimal set, we applied non dominated sort PSO (NSPSO) [13] as it performs better against a set of well known test functions that are presumed difficult. Rest of the paper is organized as follows. Section 2 specifies the problem definition. In section 3, describes the formulation of discrete particle swarm optimization for multi objective workflow grid scheduling. In section 4, we explained the procedure of non dominated sort particle swarm optimization algorithm which is used to obtain Pareto optimal solutions. Section 5 discusses the simulation analysis and finally section 6 gives the conclusion.

2 Problem Definition: Workflow Grid Scheduling

We define workflow Grid scheduling as the problem of assigning different precedence constraint tasks in the workflow to different available grid resources. We model the application as a task graph: Let G = (V, E) be a directed acyclic graph (DAG), with V as the set of n tasks $t_i \in V$, $1 \leq i \leq n$ and E is the set of edges representing precedence constraint among the tasks $e_{ij} = (t_i, t_j) \in E$, $1 \leq i \leq n$, $1 \leq j \leq n$, $i \neq j$. Associated to each edge is the amount of data required to send from task t_i to t_j if they are not executed on the same resource. Let set R represent the m number of resources which are available in the grid and resource $r_j \in R$ is associated with two values: Time and cost of executing the task on resource r_j. Every task t_i has to be processed on resource r_j until completion. In our work, scheduling solution is represented as the task assignment string corresponding to the scheduling order string. Task assignment string is the allocation of each task to the available time slot of the resource capable of executing the task, and the scheduling order string encodes the order to schedule tasks. The ordering of tasks in the scheduling order string must satisfy the task dependencies. The execution optimization problem is to generate task assignment string S, which maps every t_i onto a suitable r_j to achieve the multi objective below:

$$\text{Minimize Time}(S) = \max \mathit{time}(t_i) \quad \text{where } t_i \in V \text{ and } 1 \leq i \leq n \tag{1}$$

$$\text{Minimize Cost}(S) = \sum \mathit{cost}(t_i) \quad \text{where } t_i \in V \text{ and } 1 \leq i \leq n \tag{2}$$

$$\text{Subject to} \quad \text{Cost}(S) < B \text{ and Time}(S) < D$$

Where B is the cost constraint (Budget) and D is the time constraint (Deadline) required by users for workflow execution.

3 Discrete Particle Swarm Optimization for Workflow Grid Scheduling

In this paper, we used the version of discrete particle swarm optimization (DPSO) [11] to solve the problem of workflow grid scheduling. PSO is a self adaptive global search optimization technique introduced by Kennedy and Eberhart [12] and it relies on the social behavior of the particles. In every generation, each particle adjusts its trajectory based on its best position (local best) and the position of the best particle (Global best) of the entire population. One of the key issues in designing successful PSO algorithm is the representation step, i.e. finding a suitable mapping between problem solution and PSO particle. For optimization of workflow grid scheduling problem, solution is represented as task assignment string (S) as mentioned in section 2. To represent S, we setup an n dimension search space corresponding to n number of tasks and each dimension represents the discrete value corresponding to m number of resources.

Here, solutions or task assignment strings are encoded as m×n matrix, called position matrix where m is the number of available resources and n is the number of tasks. Let X_k is the position matrix of k^{th} particle then

$$X_k(i,j) \in \{0,1\}\ (\forall\ i,j),\ i \in \{1,2,...m\},\ j \in \{1,2,..., n\}. \tag{3}$$

where X_k (i, j) = 1 means that j^{th} task is performed by i^{th} resource. Hence, in each column of the matrix only single element is 1 and others are 0. For example, Fig.1. shows the mapping between one possible task assignment strings to the particle position matrix in PSO domain.

Task Assignment String

{[T1 : R3], [T2 : R1], [T3 : R3], [T4 : R2], [T5 : R1], [T6 : R3]}

Particle Position Matrix

	T1	T2	T3	T4	T5	T6
R1	0	1	0	0	1	0
R2	0	0	0	1	0	0
R3	1	0	1	0	0	1

Fig. 1. Mapping of Task assignment string to Particle Position matrix

Velocity of each particle is again an m×n matrix whose elements are in range [$-V_{max}$, V_{max}]. If V_k is the velocity matrix of k^{th} particle, then:

$$V_k(i,j) \in [-V_{max}, V_{max}],\ (\forall\ i, j),\ i \in \{1,2,...m\},\ j \in \{1,2,..., n\}. \tag{4}$$

Also, Pbest and Gbest are m×n matrices and their elements assume value 0 or 1 as in the case of position matrices. $Pbest_k$ represents the best position that k^{th} particle has visited since the initial time step and $Gbest_k$ represents the best position that k^{th} particle and its neighbors have visited since the algorithm was initiated. For updating $Pbest_k$ and $Gbest_k$ in each time stamp we are using the non dominated sort multi objective PSO algorithm as mentioned by procedure of NSPSO in section 4.

For particle updating, we are first updating velocity matrix according to (5) and then finally position matrix is updated using (6).

$$V_k^{(t+1)}(i,j) = \omega .\ V_k^{(t)}(i,j) + c_1 r_1 (Pbest_k^{(t)}(i,j) - X_k^{(t)}(i,j)) + c_2 r_2 (Gbest_k^{(t)}(i,j) - X_k^{(t)}(i,j)) \tag{5}$$

$$X_k^{(t+1)}(i,j) = \begin{cases} 1, & if\ V_k^{(t+1)}(i,j) = \max\left\{V_k^{(t+1)}(i,j)\right\} \forall\ i \in \{1,2,\ldots m\} \\ 0, & otherwise \end{cases} \tag{6}$$

Using equation (6), each column of position matrix, the value 1 is assigned to the element whose corresponding element in velocity matrix has maximum value in its corresponding column. If in a column of velocity matrix there are more than one element with max value, then one of these elements is selected randomly and 1 is assigned to its corresponding element in the position matrix.

A particle represented as position matrix X_k is formulated from task assignment string (S). Initially, S representing resource on which a task will execute is defined randomly. The fitness functions Ftime(S) and Fcost(S) are formed in order to evaluate individuals according to makespan and cost of the schedule respectively. These fitness functions are calculated from Equation (1) and (2) by adding the penalty. On the violation of deadline and budget constraints, penalty is added respective to objective function, otherwise not.

4 Multi Objective Optimization Algorithm Used

To optimize workflow grid scheduling under two conflicting objective of makespan and total cost, we are using non dominated sort particle swarm optimization (NSPSO) [13] approach. NSPSO extends the basic form of PSO by making a better use of particle's personal bests and offspring for effective non-domination comparisons. The steps of basic NSPSO procedure are as follows:

NSPSO Procedure

1. Create and initialize m×n dimensional swarm with N particles randomly. The initial velocity for each particle is also initialized randomly but in the range [$-V_{max}$, V_{max}]. The personal best position of each particle ($Pbest_k$), is set to X_k.
2. Evaluate each particle in the swarm.
3. Apply non-dominated sorting on the particles.
4. Calculate crowding distance of each particle.
5. Sort the solutions based on decreasing order of crowding distance.

6. Select randomly $Gbest^{(t)}$ for each particle from a specified top part (e.g. top 5%) of the first front F1;
7. Calculate the new velocity $V^{(t+1)}$ for all particles based on Equation (5) and new position $X^{(t+1)}$ from Equation (6) using the determined $Gbest^{(t)}$ and $Pbest^{(t)}$.
8. Create a new population of size 2N by combining the new position and their personal best, $X^{(t+1)} \cup Pbest^{(t)}$.
9. Apply non-dominated sorting on 2N particles and calculate the crowding distance for each particle.
10. Generate a new set of N solutions by selecting solutions from non-dominated fronts F1, F2 and so on using the crowding distance. The N solutions form the personal best for the next iteration.
11. Go to step 2 till the termination criteria is met.

5 Simulation Results and Discussion

We used GridSim [5] toolkit in our experiment to simulate the scheduling of workflow tasks. GridSim is a java based toolkit for modeling and simulation of resource and application scheduling in large-scale parallel and distributed computing environment such as Grid.

In our test environment, we simulated the balanced workflow consisting of 20 tasks on 8 virtual resources and these resources are maintained by different organizations in the grid. Links between resources are established through a router so that direct communication can take place between resources. Computational rating (Million instructions per second) and computational cost (in dollars) of each resource is generated with non-uniform distribution. Number of data units required by one task from another task in the workflow is also generated non-uniformly.

In order to generate valid schedule which can meet both deadline and budget constraints specified by the user, two algorithms HEFT [10] and Greedy Cost were used to make deadline and budget effectively. HEFT is a time optimization scheduling algorithm in which workflow tasks are scheduled on minimum execution time heterogeneous resources irrespective of utility cost of resources. So HEFT gives minimum makespan ($Time_{min}$) and maximum total cost ($Cost_{max}$) of the workflow schedule. Greedy Cost is a cost optimization scheduling algorithm in which workflow tasks are scheduled on cheapest heterogeneous resources irrespective of the task execution time. Thus Greedy Cost gives maximum makespan ($Time_{max}$) and minimum total cost ($Cost_{min}$) of the workflow schedule Thus Deadline (D) and Budget (B) are specified as:

$$D = Time_{max} – 0.1(Time_{max} - Time_{min}) \quad (7)$$

$$B = Cost_{max} – 0.1(Cost_{max} - Cost_{min}) \quad (8)$$

To measure the effectiveness and validity of NSPSO algorithm for workflow grid scheduling problem, we have implemented a probabilistic GA based technique known as non dominated sort genetic algorithm (NSGA-II) [7]. To implement the NSGA-II we have taken binary tournament selection, two point crossover and replacing mutation.

Each experiment was repeated 20 times with different random population initialization. Initial population was seeded with two solutions obtained by heuristics namely LOSS-II and modified GAIN-II. LOSS-II and modified GAIN-II were used to generate a solution with minimum makespan (total time) while meeting budget constraint and minimum total cost while meeting deadline constraint respectively. This is done because population with seeding which contains two already optimized boundary solutions gives good and fast convergence with better spread rather than generating all solutions randomly [20]. In order to compare the performance of algorithms, we have run the algorithms over 200 generations with the initial population size of 50. The Pareto optimal solutions obtained with NSPSO and NSGA-II for bi-objective workflow grid scheduling problem are shown in Fig. 2. From a typical run shown in Fig. 2 we can see that most of the solutions obtained with NSPSO are lie on the better front as compared to NSGA-II while preserving uniform diversity between solutions.

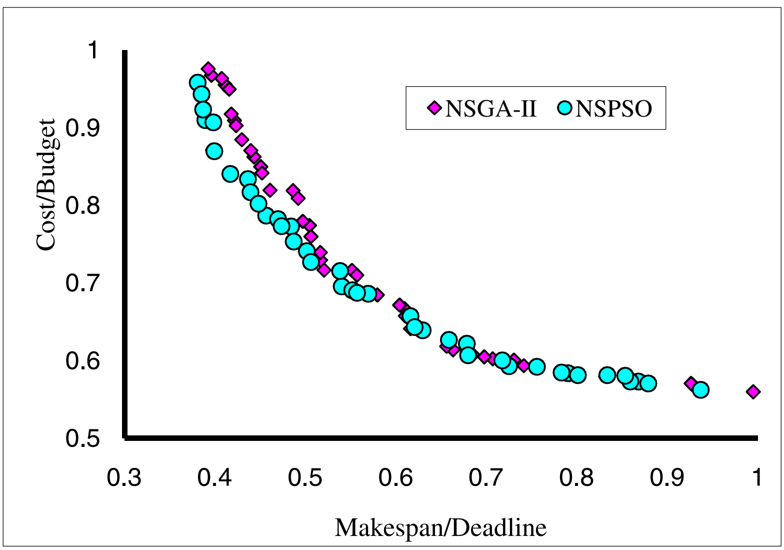

Fig. 2. Obtained Pareto Optimal Solutions with NSPSO and NSGA-II

5.1 Performance Evaluation: GD, Spacing

For the performance comparison between NSPSO and NSGA-II we conducted our experiment over 20 runs and then average of these runs has been taken for evaluation. To measure the quality of evolutionary algorithms, we used two metrics Generational Distance (GD) and Spacing [8]. GD is the well known convergence metric to evaluate the quality of an algorithm against the reference front P*. The reference front P* was obtained by merging solutions of algorithms considered. On the other side, Spacing metric was used to evaluate diversity among solutions. The results obtained with these metrics for each algorithm are depicted in Table 1. The average value of the GD and spacing metric corresponding to NSPSO is less as compared to other algorithm considered i.e, NSGA-II. The result confirms the better convergence towards the real Pareto optimal front. Further, the low value of standard deviation shows that algorithm converges almost in every execution.

Table 1. GD, Spacing Metric Results for the algorithms used

		NSGA-II	NSPSO
GD Metric	Avg	0.0231861	**0.021173**
	Std. Dev.	0.001567	0.001480
Spacing Metric	Avg	0.042911	**0.040233**
	Std. Dev.	0.003558	0.003324

6 Conclusion and Future Work

The current work emphasizes on the planning and optimizing the workflow scheduling in the grid. In this paper, we have used a version of discrete particle swarm optimization to represent the scheduling of workflow grid. The multi-objective non dominated sort particle swarm optimization (NSPSO) approach has been used to find solutions in the entire Pareto optimal front in order to minimize the two conflicting objectives of makespan and total cost. The performance of NSPSO algorithm has been evaluated in comparison with NSGA-II algorithm. The simulation results exhibit that NSPSO is a better compromised multi-objective optimization algorithm for workflow grid task scheduling in terms of convergence towards true Pareto optimal front and uniformly distributed solutions with small computation overhead. In future we plan to evaluate the scheduling approach using more than two objectives simultaneously. We will also apply more advanced algorithms like MOPSO [18], 2LB-MOPSO [17], fuzzy dominance based MOPSO [19] etc to obtain the optimal solutions for workflow grid scheduling problem.

References

1. Abraham, A., Liu, H., Zhang, W., Chang, T.G.: Scheduling Jobs on Computational Grids Using Fuzzy Particle Swarm Algorithm, pp. 500–507. Springer, Heidelberg (2006)
2. Attiya, G., Hamam, Y.: Task allocation for maximizing reliability of distributed systems: A simulated annealing approach. Journal of Parallel and Distributed Computing 66, 1259–1266 (2006)
3. Braun, T.D., Siegal, H.J., Beck, N.: A comparision of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems. Journal of Parallel and Distributed Computing 61, 810–837 (2001)
4. Buyya, R., Venugopal, S.: A Gentle Introduction to Grid Computing and Technologies. CSI Communications 29, 9–19 (2005)

5. Buyya, R., Murshed, M.: GridSim: A Toolkit for Modeling and Simulation of Grid Resource Management and Scheduling, vol. 14, pp. 1175–1220 (2002), http://www.buyya.com/gridsim
6. Chen, W.H., Lin, C.S.: A hybrid heuristic to solve a task allocation problem. Computers & Operations Research 27, 287–303 (2000)
7. Deb, K., Pratap, A., Aggarwal, S., Meyarivan, T.: A Fast Elitist Multi-Objective Genetic Algorithm. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
8. Deb, K., Jain, S.: Running Performance Metrics for Evolutionary Multi-objective Optimization. In: Proceedings of Simulated Evolution and Learning (SEAL 2002), pp. 13–20 (2002)
9. Wieczorek, M., Prodan, R., Fahringer, T.: Scheduling of Scientific Workflows in the ASKALON Grid Environment. SIGMOD 34(3), 56–62 (2005)
10. Haluk, T., Hariri, S., Wu, M.Y.: Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. IEEE Transactions on Parallel and Distributed Systems 13, 260–274 (2002)
11. Izakian, H., Tork Ladani, B., Zamanifar, K., Abraham, A.: A Novel Particle Swarm Optimization Approach for Grid Job Scheduling. In: Prasad, S.K., Routray, S., Khurana, R., Sahni, S. (eds.) ICISTM 2009. CCIS, vol. 31, pp. 100–109. Springer, Heidelberg (2009)
12. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
13. Li, X.: A Non-dominated Sorting Particle Swarm Optimizer for Multi- objective Optimization. In: Proceeding of Genetic and Evolutionary Computation Conference 2003 (GECCO 20003), Chicago, USA (2003)
14. Ritchie, G., Levine, J.: A fast, effective local search for scheduling independent jobs in heterogeneous computing environments, Technical report, Centre for Intelligent Systems and their Applications, School of Informatics, University of Edinburgh (2003)
15. Subrata, R., Zomaya, Y.A., Landfeldt, B.: Artificial life techniques for load balancing in computational grids. Journal of Computer and System Sciences 73, 1176–1190 (2007)
16. Tsiakkouri, E., Sakellariou, R., Zhao, H., Dikaiakos, M.D.: Scheduling Workflows with Budget Constraints. In: CoreGRID Integration Workshop Pisa, Italy (2005)
17. Zhao, S.Z., Zhao, P.N.: Two-lbests Based Multi-objective Particle Swarm Optimizer. Engineering Optimization 43, 1–17 (2011)
18. Coello, C.A.C., Pulido, G., Lechuga, M.: Handling multi-objective with particle swarm optimization. IEEE Trans. Evol. Comput. 8(3), 256–279 (2004)
19. Praveen, K., Das, S., Welch, S.M.: Multi-Objective Hybrid PSO Using ε-Fuzzy Dominance. In: Proceeding of Genetic and Evolutionary Computation Conference (GECCO 2007), London, UK (2007)
20. Yu, J., Buyya, R.: Scheduling Scientific Workflow Applications with Deadline and Budget constraints using Genetic Algorithms. Scientific Programming Journal 14(1), 217–230 (2006)