

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## Improving Scheduling of Scientific Workflows Using Tabu Search for Computational Grids

Shajulin Benedict and V. Vasudevan

Software Technologies Group, TIFAC Core in Network Engineering, AKCE, India

**Abstract:** In this study, a scheduling procedure is proposed to evaluate a heuristic search algorithm, namely, Tabu Search (TS) for Grid Scheduling problem. In addition, two contradicting objectives of the Grid environment are aimed to achieve simultaneously by the scheduling mechanism. Further, the evaluation of other scheduling mechanisms namely First Come First Serve (FCFS), Last Come First Serve (LCFS) and Earliest Deadline First (EDF) are compared. In the experiments, we show the effectiveness of the TS method.

**Key words:** Grid computing, scheduling, tabu search, workflow

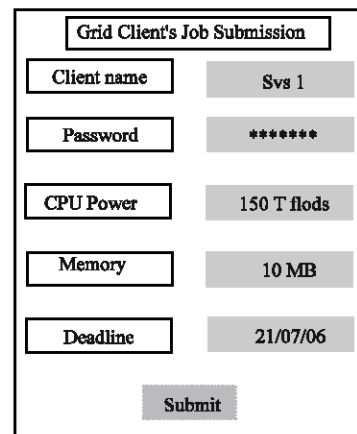
### INTRODUCTION

Many large scale applications such as scientific, engineering and business problems (Hai *et al.*, 2005; Cannataro *et al.*, 2002) are solved effectively using the logical amalgamation of geographically dispersed Grid resources (Berman *et al.*, 2002). Grid computing (Eva *et al.*, 2003; Minzhu *et al.*, 2005) is very much promising for performing complex computations involving many computation tasks and processing large amount of data as in image simulation, storage and processing (Eva *et al.*, 2003). Grids sub divide the large scale problem to smaller jobs and allot the work at different autonomous domain grid sites. The feasibility for reality is nearer owing to the exponential growth of the Internet, computers and low-cost of components.

But still, grid computing becomes a major problem owing to the challenges in security issues (Song *et al.*, 2005, 2006), protocol design, access control (Baker *et al.*, 2002), implementation models (Parashar *et al.*, 2005), resource management and grid scheduling. Following are the grid scheduling challenges:

- Unpredictable challenges in Grid resources (i.e., Availability, accessibility and so on).
- Need for a multiple resource types for completing a job (Andrieux *et al.*, 2004).
- Need for a parallel (Rewini *et al.*, 1990) or concurrent execution of tasks in any workflows.

Workflows submitted to the Computational Grids by resource consumers have a proper budget proposal (Buyya, 2002), client authentication and the requirements for its execution as shown in Fig. 1. The willingness to complete any job is given by resource providers. Hence the Grid schedulers (Berten *et al.*, 2006; Nabrzyski *et al.*,



The figure shows a web form titled "Grid Client's Job Submission". It contains several input fields and a submit button. The fields are: Client name (Svs 1), Password (\*\*\*\*\*), CPU Power (150 T flops), Memory (10 MB), and Deadline (21/07/06). A Submit button is located at the bottom right of the form.

Grid Client's Job Submission	
Client name	Svs 1
Password	*****
CPU Power	150 T flops
Memory	10 MB
Deadline	21/07/06
Submit	

Fig. 1: Job submission chart

2005) search for solutions in the state space aiming at achieving high performance, both in terms of solution quality and execution speed.

The traditional system namely advanced reservations for scheduling the workflows undergoes problems such as overloading and power failure. The overloading and the scheduler failure problem are overridden by a two level scheduling scheme where the first level is used for frequent small jobs and second level for large jobs. The market oriented approach algorithm succeeded in the distributed scheduling of workflows, but could not appease completion of more workflows within the deadline. The success ratio of the workflows allotted for mapping the Grid sites is 30% (Chien *et al.*, 2005) when 30 workflows are scheduled at a time.

The distributed resource discovery and algorithms suit appropriately when the scheduling is for small jobs. On considering the large periodical jobs that are scheduled in advance, iteration based scheduling

mechanisms provide optimal results. The Genetic algorithm is efficient on considering a single objective. Often, the real world problems require the simultaneous optimization (Shajulin and Vasudevan, 2006) of several incommensurable and competing objectives.

In this study, an heuristic search procedure is developed using TS for generating better schedules for a computational Grid environment, by considering dual objectives namely maximization of Job completion ratio and minimization of the penalty incurred to the Grid scheduler on choosing the specific workflow sequence. The TS (Bland and Dawson, 1991; Fred and Manuel, 1993; Nowicki *et al.*, 2005; Buyya, 2002) is a heuristic and iterative scheduling mechanism that explores the solution space and make a better move finding its neighborhood within a limited amount of time. The evaluation of the Tabu Search scheduling mechanism with other algorithms such as First Come First Serve (FCFS), Last Come First Serve (LCFS) and Earliest Deadline First (EDF) are made.

As mentioned earlier, a typical example for a computational Grid is focused on the efficient utilization of Grid resources by Grid scheduling to appease the customer and provider satisfactions, such as Condor project (Condor-online (Frey *et al.*, 2002), which being developed for about fifteen years, aim for high throughput. Two-level scheduling for large periodical jobs is discussed using Genetic algorithm in (He *et al.*, 2004; Vincenzo *et al.*, 2002). Here an optimal solution is selected using the parameters such as job characteristics, Grid environment characteristics and data distribution characteristics. In (Wiecezok *et al.*, 2005), scheduling two scientific workflows with the algorithm namely Heterogeneous Earliest Finish Time (HEFT) is compared with other myopic and genetic algorithms. HEFT schedules workflows by creating an order for the tasks. The above approaches work well when the number of the workflows is small.

In the separated works (Eva *et al.*, 2003), the Grid workflows are mapped onto Grid resources based on existing planning technology. This work focuses on planning problem considering specific planning systems. The drawback towards this system is poor response times, time intensive computation and not inclined towards the Grid specific constraints. In (Song *et al.*, 2005), the simulated Grid performances of the Grid job scheduling to assure secure Grid job execution under different risky conditions is considered. However, the workflow assumed for the simulation is purely independent tasks and has single objective for each run.

The works related to (Halpern and Moses, 1990; Real *et al.*, 2005), considers scheduling using learning algorithm based approach, requiring more database for

efficient scheduling. In (Buyya, 2002), the tasks are scheduled using Tabu search, simulated annealing, Genetic algorithm and the hybrid mechanisms. The work in (Buyya, 2002; Song *et al.*, 2006; Vincenzo *et al.*, 2002) discuss the scheduling procedure for single objectives that is not sufficient for the real Grid environment.

Our approach, using Tabu Search with combined dual objective function based scheduling, is efficient when considering the other scheduling mechanisms. A few papers address the issues of Tabu Search based scheduling (Bland and Dawson, 1991; Nowicki *et al.*, 2005).

### PROBLEM DESCRIPTION

The Inter-Grid Scheduler Architecture (IGSA) described with a four node Grid environment example is shown in Fig. 2. This architecture can be utilized for any practical applications for the normal grid environments. The setup is experimented in TIFAC Core in Network Engineering under DST project.

The goal of the IGSA is to find the allocation sequence of workflows on each Grid site. Four major entities are involved in this architecture:

- The grid users submit their request for job completion to the local grid managers.
- All the tasks should be received by the grid managers and the decision for the scheduling is made on deploying the request to the Intra-Grid schedulers.

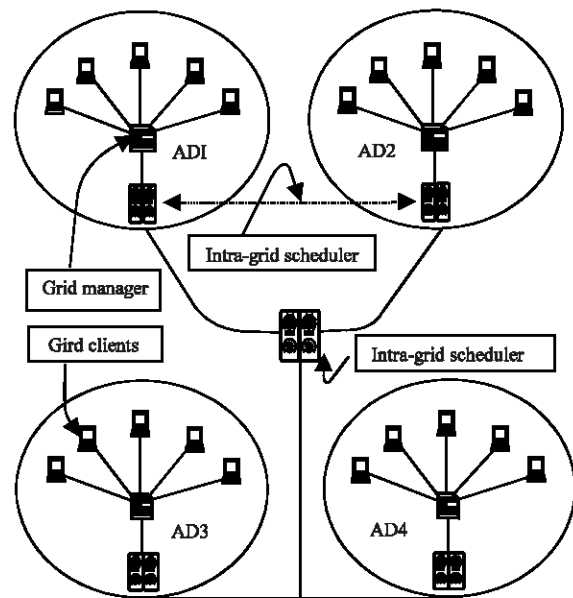


Fig. 2: Inter-grid scheduler architecture

Table 1: Experimental work flows

Work flow	Tasks/(duration)	Penalty cost	Grid
W1	T <sub>1</sub> (3),T <sub>2</sub> (2),T <sub>3</sub> (3),T <sub>4</sub> (4), T <sub>5</sub> (3),T <sub>6</sub> (1).	100.00	Any
W2	T <sub>1</sub> (2),T <sub>2</sub> (4),T <sub>3</sub> (3),T <sub>4</sub> (1), T <sub>5</sub> (3),T <sub>6</sub> (2),T <sub>7</sub> (3).	50.00	Any
W3	T <sub>1</sub> (2),T <sub>2</sub> (4),T <sub>3</sub> (1),T <sub>4</sub> (2), T <sub>5</sub> (3),T <sub>6</sub> (4),T <sub>7</sub> (5).	75.00	Any
W4	T <sub>1</sub> (2),T <sub>2</sub> (3),T <sub>3</sub> (1),T <sub>4</sub> (4), T <sub>5</sub> (5),T <sub>6</sub> (1),T <sub>7</sub> (2), T <sub>8</sub> (3).	80.00	Any
W5	T <sub>1</sub> (1),T <sub>2</sub> (3),T <sub>3</sub> (2),T <sub>4</sub> (4), T <sub>5</sub> (5),T <sub>6</sub> (6).	100.00	Any
W6	T <sub>1</sub> (1),T <sub>2</sub> (3),T <sub>3</sub> (1),T <sub>4</sub> (2), T <sub>5</sub> (4),T <sub>6</sub> (2).	99.00	Any
W7	T <sub>1</sub> (1),T <sub>2</sub> (4),T <sub>3</sub> (2),T <sub>4</sub> (5), T <sub>5</sub> (3),T <sub>6</sub> (2).	150.00	Any
W8	T <sub>1</sub> (1),T <sub>2</sub> (3),T <sub>3</sub> (2),T <sub>4</sub> (3), T <sub>5</sub> (1),T <sub>6</sub> (1),T <sub>7</sub> (2), T <sub>8</sub> (3).	100.00	Any
W9	T <sub>1</sub> (3),T <sub>2</sub> (3),T <sub>3</sub> (4),T <sub>4</sub> (6), T <sub>5</sub> (2),T <sub>6</sub> (4),T <sub>7</sub> (2),T <sub>8</sub> (1), T <sub>9</sub> (5)	10.00	Any
W10	T <sub>1</sub> (4),T <sub>2</sub> (5),T <sub>3</sub> (4),T <sub>4</sub> (4), T <sub>5</sub> (5),T <sub>6</sub> (1),T <sub>7</sub> (2).	5.00	Any

- The Intra-Grid schedulers have the updated information of the grid resources that are idle during time t. This information is frequently updated. The smaller jobs can be scheduled within their deadlines by the Intra-Grid schedulers in their respective Administrative Domains (AND). Here scheduling is often dynamic.
- For data intensive applications where the jobs are larger require the necessity of the resources world wide. At that moment, there is a necessity of Inter-Grid schedulers which is static often.

The workflow allocation strategy in a Grid environment differs as in (Hamscher *et al.*, 2000). The goal of the Inter-Grid scheduler is to receive the request from different intra-grid schedulers and make an optimistic scheduling such that it accommodates many workflows completing within its deadline. The following DAG workflows and the penalty cost for each workflow are considered for experimental purpose.

The duration for any workflow, penalty cost incurred and the required grid resources are shown in the Table 1. The tasks taken for experiment have their predecessors and successors, such as T<sub>1</sub> follows T<sub>2</sub> or T<sub>2</sub>, T<sub>3</sub> are parallel computation once when the task T<sub>1</sub> is executed.

The workflow model for W1, W2, W3 are shown in Fig. 3. The FCFS map tasks to the idle Grid sites based on first task arrival to serve first. The EDF algorithm executes the tasks whose absolute deadline is the earliest. Hence it estimates the execution deadline of the individual workflow for any standalone system and schedules such that the workflows that require greater completion time is served first. In EDF, the task priorities are not fixed but change depending on the closeness of their absolute deadline.

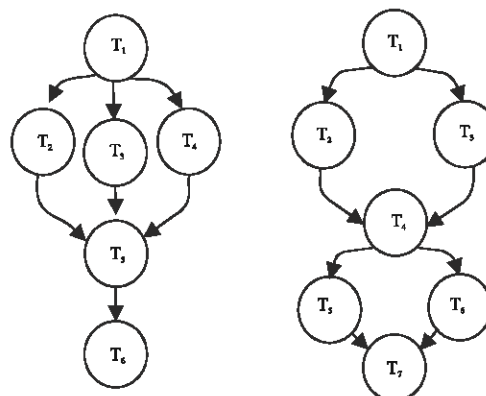


Fig. 3: DAG workflow model

The setting of the experiment consists of workflows with following assumptions:

- Each workflow received in the Inter-Grid scheduler consists of a set of Tasks T<sub>1</sub>, T<sub>2</sub>, T<sub>3</sub> and so on.
- The task in each workflow is a Directed Acyclic Graph (DAG) model (Fig. 3).
- The output from a task can be transferred to other tasks as per the DAG graph model and all jobs are available at time zero.
- At any time, a task can be executed only on a Grid site which is reported to the Inter-Grid scheduler as idle via Intra-Grid scheduler.
- There is no pre-emption of tasks or workflows.
- The sequential order of workflow allotment changes.

Here we present a scheduling approach for the wide area problem. Since it is a wide area problem, the resources and jobs are dispersed geographically on a wide area network.

### PROPOSED METHOD OF TS

In this study, TS heuristic to solve scientific workflow scheduling problem in Grid is discussed. The basic idea of the method is described by (Fred and Manuel, 1993; Bland and Dawson, 1991).

**TS Algorithm:** The TS introduces flexible memory structures articulating strategic restrictions and aspiration levels as a mean for exploiting search spaces. TS has the ability to generate solutions of notably high quality such as to escape from the local minima and to implement an explorative strategy. TS are an iterative procedure for searching a global optimum for discrete combinatorial problem. The basic concept of TS is meta-heuristic superimposed on another heuristic. The philosophy of TS is to avoid entrainment in cycles by forbidding or

penalizing moves, which take the solution in the next iteration, to points in the solution space previously visited.

The TS methods operate under the assumption that a neighborhood can be constructed to identify adjacent solutions that can be reached from any current solution. Pair wise exchanges (swaps) are frequently used to define neighborhoods in permutation problems, identifying moves that lead from one solution to the next. Associated with each swap is a move value, which represents the change on the objective function value as a result of the proposed exchange.

The algorithm checks for the best individual starting from the initial solution. The best neighbor that does not violate the Tabu rule is sought to replace the current individual. A chief mechanism for exploiting memory in TS is to classify a subset of the moves in a neighborhood as forbidden (or Tabu). The classification depends on the history of the search, particularly as manifested in the recently or frequency that certain move or solution components, called attributes, have participated in generating past solution. As a basis for preventing the search from repeating swap combinations tried in the recent past, potentially reversing the effects of previous moves by interchanges that might return to previous positions, we will classify Tabu all swaps composed of any of the most recent pairs of such modules. Also, Tabu restrictions are not inviolable under all circumstances. When a Tabu would result in a solution better than any visited so far, its Tabu classification may be overridden. A condition that allows such an override to occur is called an aspiration criterion.

Hence TS may be conveniently characterized as a special form of neighborhood search.

**General Scheme of TS:** The structure of TS algorithm is as shown below:

**Step 1: Initialization**

- Select a starting solution  $s \in S$ .
- Set a history record H empty.
- Add the current best solution by setting  $s\_best = s$  and define  $best\_cost = c(H, s\_best)$ .

**Step 2: Choice and Termination**

- Determine Individual\_N(s) as a subset of N(H,s).
- Select  $s\_next$  from Individual\_N(s) to minimize  $c(H,s)$  over this set.
- Terminate by the chosen iteration cutoff rule.

**Step 3: Updation**

- Reset  $s=s\_next$
- If  $c(H,s) < best\_cost$ , perform step 1.3 and then return to step 2.
- Update the history record H.

In TS, each solution  $s \in S$  has its neighbor's (H,s) belongs to S, called the neighborhood of s. The solutions  $s_1, s_2, s_3$  and so on can be reached directly from s by a move. The solution that has reached by the move is determined by the history.

The importance of the TS method depends on the following points:

- The definition and utilization of the history record H.
- Determination of neighborhood Individual\_N and the evaluation function  $c(H,s)$ .

**Combined dual objective TS procedure:** Using Eq 1, the objective function is obtained and for every solution move in the TS procedure, the neighborhood solution will be evaluated for a Combined Dual Objective Function (CDOF) of minimizing the total penalty cost on choosing the workflow sequence and maximizing the number of workflows completed within deadline (Job Completion Ratio). In our proposed method, the workflows are created based on DAG model and the deadline  $DT_1$  is fixed to be at  $1.5 \times \text{Execution time}$  eg:  $DT_1 = 1.5 \times 16$ . If the workflows are completed early, the penalty cost as given in Eq. 2 should not be balanced.

**Combined Dual Objective Function (CDOF):**

Minimize

$$CDOF = (E_{F1}) \times (T_p \div MPP) + (E_{F2}) \times (1 \div JCR) \quad (1)$$

Where:

- $E_{F1}$  = Evaluation factor for grid client satisfaction
- $E_{F2}$  = Evaluation factor for job completion ratio
- JCR = Job Completion Ratio
- $T_p$  = Total penalty cost
- MPP = Maximum Permissible Penalty

$$T_p = \sum_i (JCT_i - DT_i) \times (PC_i)$$

$$\forall (JCT_i \geq DT_i) = 0 \quad (2)$$

Otherwise

i = Workflow number

$JCT_i$  = Job Completion Time of workflow  $i$   
 $Dt_i$  = Deadline time of workflow  $i$

$$JCR = \sum_i xw_i \quad (3)$$

Where:

$xw = 1 \forall (JCT_i \leq Dt_i)$   
 $= 0$  Otherwise

Each evaluation Factors are given equal weightings of 0.5. However different weightings can be applied as per the situations.

The TS parameters used for experimental purpose are given as follows:

- The pair of the workflow sequence numbers is swapped for the move attribute. For example Before move

W6, W2, W7, W5, W1, W4, W10, W9, W3, W8

After move

W6, W9, W7, W5, W1, W4, W10, W2, W3, W8

- History record indicates the list of workflow sequence numbers of the jobs swapped in the previous iterations each of them classified as Tabu. This record is more crucial for making the elimination of the looping on considering the same solution for the next iterations.
- After a consecutive ten iterations, the Tabu restriction of a sequence number is hiked termed as Tabu Tenure.
- The choice criterion determines the solution with the minimum CDOF among the neighboring solutions of the current solution.
- The termination criterion terminates the solution that does not reach the predefined minimum value of CDOF. The choice of termination criterion for the problem is taken as 1000 iterations.

The methodology is such that an initial job sequence is selected at random among the set of job sequence (S) and the CDOF for the solution  $s$  is defined as a best cost. The obtained solution is recorded as a initial step for the Tabu Scheduling mechanism. Later, the set of neighborhood solutions of  $s$  is generated and again the CDOF is calculated and replaced if necessary finding the best cost among the history record.

The updating of the history record is done and the termination is followed as a last step if the optimal results are obtained. Otherwise the procedure is iterated until optimal results.

## RESULTS AND DISCUSSION

This schedule for the Inter-Grid computation of workflows is obtained by the procedure using Tabu Search scheduling mechanism. This is compared with sequences obtained by different scheduling rules viz. FCFS and EDF. For the experimental problem the workflow sequences obtained by TS gives minimum penalty cost on choosing the scheduling sequence received in the Inter-Grid schedulers and maximize the job completion ratio. This is performed as dual objective satisfaction criteria and minimizes two contradicting objectives simultaneously during the same search time. Hence it outperforms all the previous procedures.

The comparative increase in the completion of workflows by TS dual objective scheduling mechanism considering other algorithms such as FCFS, EDF are shown in Fig. 4.

It can be analysed that the FCFS, EDF is not successful in the number of workflow completion. In Table 2, the penalty cost incurred by the Inter-Grid scheduler on not completing the job is plotted. As per the methodology, TS succeeds the other scheduling mechanisms in consideration.

The cruciality of the TS algorithm stands on the combined dual objective solution by minimization of the total penalty cost as well as comparatively maximization of the workflows in a single run as shown in Fig. 5. The maximal and the minimal points for each number of workflows are expressed as vertical lines. It can be easily drawn to a conclusion that the TS algorithm is the best scheduling mechanism compared to the other scheduling mechanisms.

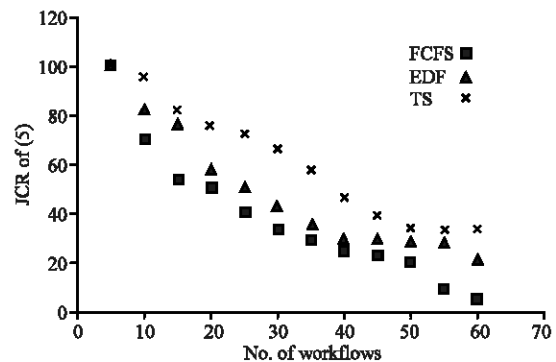


Fig. 4: Job completion ratio

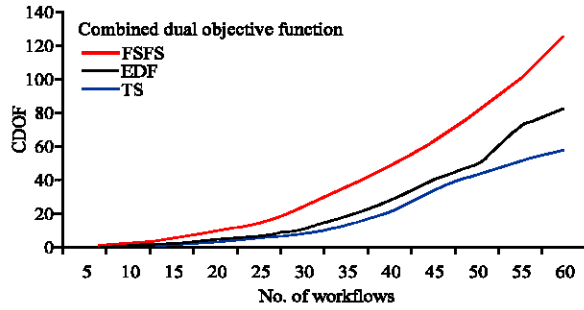


Fig. 5: CDOF for TS, EDF and FCFS

Table 2: Penalty cost incurred for the workflow sequence

No. of work flows	FCFS	EDF	TS
5	0.00	0.00	0.00
10	3.15	0.95	0.55
15	8.70	4.20	1.20
20	18.80	9.55	4.50
25	28.10	12.95	10.05
30	47.65	21.45	15.55
35	71.65	35.95	25.85
40	97.55	56.15	42.55
45	125.55	80.25	67.85
50	160.15	97.25	85.65
55	200.85	145.55	103.50
60	252.40	165.00	115.55

Although the main focus of this research is to devise a mechanism to produce a better schedule, it has been found that there is also ample scope of choice of selecting other parameters of TS namely cross over rate, mutation rate, population size and so on.

### CONCLUSION AND OUTLOOKS

This study has presented the application of a scheduling procedure for a specific Grid environment to maintain its flexibility and thereby the intended performance measures. The mechanism operates based on TS and minimizes two contradicting objectives simultaneously. The solution obtained by TS scheduling mechanism is compared with other different scheduling rules such as FCFS, EDF and is found to be superior. This research work leads to a conclusion that the developed procedure can suitably be modified to any kind of Grid environment and can be applied for different objectives separately as well as in combination.

In the near future we combine Tabu Search along with the sharing method to increase the efficiency. Similarly, the ant colony properties can be included for scalability in the existing algorithm. The procedure can also suitably be modified and applied to any kind of Grid scheduling with different problem environment and any number of objectives concurrently.

### REFERENCES

Andrieux, A. *et al.*, 2004. Open Issues for Grid scheduling, [uk/technicalpapers/UKeS-2004-03pdf](http://uk/technicalpapers/UKeS-2004-03pdf).

Baker, M., R. Buyya and D. Laforenza 2002. Grids and Grid Technologies for Wide-Area Distributed Computing. Software: Practice and Experience (SPE) Journal, Wiley Press, pp: 1437-1466.

Berman, F., G. Fox and T. Hey, 2002. Grid Computing: Making the Global Infrastructure a Reality. Wiley Publ., pp: 110-115.

Berten, V., J. Goossens and E. Jeannot, 2006. On the Distribution of sequential jobs in random brokering for Heterogeneous Computational Grids. IEEE Trans. parallel Distributed Syst., 17 (2): 113-124.

Bland, J.A. and G.P. Dawson, 1991. Tabu Search and Design optimization. Comput. Aided Des., 23: 3, 195-202.

Buyya, R., 2002. Economic based distributed resource management and scheduling for grid computing. Ph. D Thesis, School of Computer Science and software Engineering, Monash University, Melbourne, Australia.

Cannataro, M., D. Talia and P. Trunfio, 2002, Distributed data mining on the grid. Future Generation Comput. Syst., 18: 1101-1112.

Chien, C.H., P.H.M. Chang and V.W. Soo, 2005. Market Oriented Multiple Resource Scheduling in Grid Computing Environments. Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA). IEEE Comput. Soc., pp: 867-872.

Eva, D., J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vazhi, K. Blackburn, A. Lazzarini, A. Arbre, R. Cavanaugh and S. Koranda, 2003. Mapping abstract complex workflows onto grid environments. J. Grid Comput., pp: 9-23.

Fred, G. and L. Manuel, 1993. Tabu Search-Modern Heuristic Techniques for Combinatorial Problems Colin, Reeves, R. (Ed.). Blackwell Scientific Publications, pp: 70-150.

Frey, J. and T. Tannenbaum *et al.*, 2002. Condor-G: A computation management agent for multi-institutional grids. J. Cluster Comput. 5: 237-246.

Hai, J., X. Shi, W. Qiang and D. Zou 2005. An adaptive meta-scheduler for data intensive applications. Int. J. Grid and Utility Comput., 1. No (1): 32-38.

Halpern, J.Y. and Y. Moses, 1990. Knowledge and common knowledge in a distributed environment. J. ACM, 37 (3): 549-587.

Hamscher, V., U. Schwiegelshohn, A. Streit and R. Yahyapour, 2000. Evaluation of job scheduling strategies for grid computing, grid first ieee/acm international workshop. Proceedings, Springer, pp: 191-202.

- He, L., S.A. Jarvis, S.P. Spooner, X. Chen and G.R. Nudd, 2004. Hybrid performance based workload management for multiclustres and grids. *IEEE Proc. Softw.*, 151 (5): 224-231.
- Minzhu, L., H. Liu, F. Tang and F. Hong, 2005. Shanghai grid in action: The first stage projects towards digital city and grid. *Int. J. Grid Utility Comput.*, 1 (1): 22-31.
- Nabrzyski J., J.M. Schopf and J. Weglarz, 2005. *Grid Resource Management*. Kluwer Academic Publishers, pp: 201-207.
- Nowicki, E. *et al.*, 2005. An advanced tabo search algorithm for the job shop problem. *J. Scheduling*, pp: 145-159.
- Parashar, M., C. James and Browne, 2005. Conceptual and Implementation models for the grids. *Proceedings of The IEEE*, 93 (3): 653-668.
- Real, R., A. Yamin, L. de Silva, G. Frainer, I. Augustin, J. Barbosa and C. Geyer, 2005. Resource scheduling on grid: Handling uncertainty. *Proceedings of the 4th International Workshop on Grid Computing*, pp: 23-30.
- Rewini, H.E. *et al.*, 1990. Scheduling parallel program tasks onto arbitrary target machines. *J. Parallel Distrib. Comput.*, 9 (2): 138-153.
- Shajulin, B. and V. Vasudevan, 2006. Scheduling of scientific workflows using Niche Pareto GA for Grids. *Proceedings of IEEE Services, Operations, Logistics (SOLI)*, pp: 942-949.
- Song, S., K. Hwang and Y.K. Kwok, 2005. Trusted Grid computing with security binding and Trust Integration. *J. Grid Computing*.
- Song, S., K. Hwang and Y.K. Kwok, 2006. Risk resilient heuristic and genetic algorithms for security assured grid job scheduling. *IEEE Trans. Comput.*, 55: 6.
- Vincenzo, Di Martino and M. Marco, 2002. Scheduling in a Grid computing environment using Genetic Algorithms. *Proceedings of the Parallel and Distributed Processing Symposium, IEEE*.
- Wieczok, M., R. Prodan and T. Fahringer, 2005. Scheduling of scientific workflows in the askalon grid environment. *SIGMOD Record*, 34 (3): 56-62.