Multi-Objective Optimization to Workflow Grid Scheduling using Reference Point based Evolutionary Algorithm

Ritu Garg
Assistant Professor
Computer Engineering Department
National Institute of Technology, Kurukshetra

Awadhesh Kumar Singh
Associate Professor
Computer Engineering Department
National Institute of Technology, Kurukshetra

ABSTRACT
Grid facilitates global computing infrastructure for user to consume the services over the network. To optimize the workflow grid execution, a robust multi-objective scheduling algorithm is needed. In this paper, we considered three conflicting objectives like execution time (makespan), total cost and reliability. We propose a multi-objective scheduling algorithm, using R-NSGA-II approach based on evolutionary computing paradigm. Simulation results shows that the proposed algorithm generates multiple scheduling solutions near the Pareto optimal front with small computation overhead.

General Terms
Distributed systems

Keywords
Workflow Grid Scheduling, Multi-objective Optimization, MOEA, Pareto dominance.

1. INTRODUCTION
Grid computing technologies primarily emerged to become the next generation of high performance computing by placing numerous heterogeneous resources of many organizations. Scheduling in Grid computing is the hot topic of research and challenging due to heterogeneity and dynamism of resources in grid. In this paper, we consider Directed Acyclic Graph (DAG), an application model for describing workflow. Scheduling of workflows in grid allows mapping of tasks on heterogeneous resources according to a set of procedural rules.

Dynamism of resources in grid is an important issue while making scheduling decisions, in which resources can fail inevitably. Failures of resources have adverse effects on performance of workflow application. Therefore, an effective scheduling algorithm should consider the failure rate of resources in order to make maximum reliability of the schedule. Scheduling is the NP-hard problem; so many heuristic approaches have been applied in the grid workflow. One of the primary motives of any grid system is to meet user requirements in an intuitive way by considering multiple objectives or criterion. Many different criterion can be considered in scheduling of complex workflow computational tasks, usually include execution time of the task, cost of the task to run on a resource, utilization of resources, reliability, turnaround time and many others.

In the recent years, many heuristics have been applied in scheduling of grid in the consideration of single criteria and pairs of certain criterion to generate single solution to the users but failed to fully satisfy users. For maximum satisfaction of user, it is necessary to produce multiple solutions with respect to minimization or maximization of objectives and selection of a solution from these solutions is further left to the user. Thereby, an optimization of conflicting multiple objectives is required to generate multiple tradeoff solutions. The Multi-Objective Evolutionary Algorithms (MOEAs) are the effective way to solve multi-objective optimization problem like scheduling in grid. An MOEA approach produces Pareto optimal set of solutions, which is the set consisted of all non-dominated solutions. A solution is called non-dominated solution if it is at least best in one objective with respect to others. This paper focuses on three major conflicting objectives namely execution time (makespan), cost and reliability of the schedule in order to generate schedules under deadline and budget constraints specified by the user. According to scheduling problem, both execution time and cost are minimization objectives while reliability is the maximization objective. But we consider reliability using reliability index as minimization objective.

Since, we are interested in the preference set of solutions near the user specified region of interest. Towards this goal we considered reference point based non dominated sort genetic algorithm (R-NSGA-II). Rest of the paper is organized as follows. Section 2 specifies some of the related work. In section 3, we introduced the Grid Workflow Scheduling problem definition. Section 4, describes the technique of multi objective optimization and different multi objective evolutionary algorithms used. Section 5 discusses the simulation analysis of MOEA approaches used. Finally section 6 gives the conclusion.

2. RELATED WORK
The Directed Acyclic Graph (DAG) based task graphs in parallel computing are reported already in literature [4] for scheduling problem. QoS aware heuristic has been proposed in [3] for grid independent task scheduling. In [5][10], Heterogeneous Earliest Finish Time(HEFT)[9] and Genetic Algorithms have been applied with extension for the ASKALON environment to solve scientific workflow applications in grid. E.Tsiakkouri et al. [8] suggested two scheduling algorithms LOSS and GAIN. LOSS makes adjustment in the schedule generated by a time optimized heuristic while GAIN in a cost optimized heuristic's schedule within the users' specified budget constraint. Wieczorek et al. [6] suggested, a bi-criteria workflow scheduling algorithm that performs optimization based on a flexible sliding constraint and the concept of dynamic programming is used in order to explore the search space effectively. Yu et al. [7] proposed time optimization and cost optimization algorithms based on the genetic algorithms within the budget and deadline constraints respectively.

For both independent and DAG tasks, Nguyen et al [24] proposed two failure-aware algorithms to generate a single schedule by considering the minimization of makespan while keeping maximum reliability. The work presented in [21] addresses tradeoff between execution time and reliability. In the paper [20], effectiveness of Evolutionary Algorithms over Simulated Annealing and Particle Swarm Optimization has been presented for scheduling jobs on Computational Grids. Furthermore, the Multi-Objective Evolutionary Algorithms (MOEAs) for workflow scheduling have been investigated to optimize two conflicting objectives simultaneously [11], [12] to generate Pareto optimal solutions. Yu et al [11] suggested and compared three major well known MOEA approaches NSGA-II [14], SPEA2 [15] and PAES [16] to solve the workflow scheduling problem in grid. Talukder et al [12] proposed a workflow execution planning approach using Multi-objective Differential Evolution (MODE) to generate a set of tradeoff schedules within the user specified constraints (deadline and budget). The ε-constraint classic Optimization method [18] has been applied in grid scheduling on independent tasks by considering makespan and flow-time objectives.

Unlike the aforementioned work, we have proposed workflow scheduling based on referenced Point based NSGA-II (R-NSGA-II) [17] considering three objectives. Using this R-NSGA-II approach we can get number of solutions in the multiple region of interest simultaneously. It generates multiple trade-off schedules, which minimize the time and cost along with the maximization of reliability in the given quality of service constraints.

3. PROBLEM DEFINITIONS
Grid workflow scheduling is defined as the problem of mapping tasks on different available grid resources according to workflow precedence constraints imposed on them. We use Directed Acyclic Graph (DAG) to model an application which consists of nodes and edges. A node in DAG represents a task t_i and an edge e represents the precedence constraint between two nodes. Amount of data is specified on each edge if two tasks are matched on the same resource. Let a set R represents the n resources available in the system and each resource $r_j \in R$ is associated with three values: completion time, total cost and reliability of executing the task t_i on resource r_j. Completion time of the task t_i is denoted by time (t_i) and total cost of the task t_i which includes both communication and service cost for executing task t_i is denoted by cost (t_i). Reliability of the task is calculated with the help e-RI. Here RI is the reliability index as denoted in the equation (3). As shown clearly, minimizing the RI, we can maximize the reliability of the task. Therefore, in this paper, we denote reliability in terms of RI. The equations (1) to (3) show incorporation of these three values in their respective objective.

$$\text{Minimize makespan}\,(S) = \max \text{time(ti)} \qquad (1)$$

$$\text{Minimize Cost}\,(S) = \sum \text{cost (ti)} \qquad (2)$$

$$\text{Minimize Reliability}\,(S) = RI = \sum (\text{time(ti)} \times \lambda j) \qquad (3)$$

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

In the equation (3), γ_j represents the failure rate of resource r_j. D is the Deadline constraint and B is the Budget constraint specified by the user for workflow application.

4. SOLVING WORKFLOW GRID SCHEDULING AS MULTI-OBJECTIVE OPTIMIZATION

4.1 Multi-objective Optimization problem
Conventionally, multi-objective optimization problem [13] can be defined as the problem of simultaneously minimization or maximization of multiple conflicting objectives. In this paper, all three objectives are considered as minimization objectives, so we present optimization accordingly. In the state of multi-objective optimization, multiple solutions are generated rather than a single solution. These multiple solutions form a set called Pareto optimal. In the Pareto optimal set all solutions are non-dominated with each and every solution of the set. A solution is said non-dominated if it is better in at least one objective with respect to all other solutions in the Pareto set. Therefore, finding Pareto optimal set of a problem is the main concern of multi-objective optimization.

Definition 1: Pareto dominance

Let function $f(s) = (f1(s), f2(s) \ldots, fm(s))$ consists of m objectives. Consider two vector solutions s1 and s2. Then solution s1 is said to dominate s2 iff following two conditions are true:

1. $\forall i \in \{1,2,\ldots m\}: f_i(s_1) \leq f_i(s_2)$
2. $\exists j \in \{1,2,\ldots m\}: f_j(s_1) < f_j(s_2)$

4.2 Evolutionary Algorithms
Many evolutionary algorithms proposed to solve the multi-objective optimization problems effectively. In our work, we considered two major well known multi-objective algorithms NSGA-II and R-NSGA-II to solve the grid workflow scheduling problem.

K. Deb and his students suggested an elitist based non-dominated sorting genetic algorithm (NSGA-II) [14] which is the improved version of NSGA. NSGA-II obtains solutions by making non-dominated fronts and selecting solutions of the last unaffordable front in less crowded area to ensure diversity among the solutions. Solutions having quality of fitness are always kept in next generation, thus ensuring elitism. NSGA-II generates solutions over entire Pareto front but to obtain multiple solutions in user specified multiple regions simultaneously, we applied another evolutionary algorithm called R-NSGA-II [17]. In R-NSGA-II, a user or decision maker simply provides some clues in terms of reference directions or reference points which represent the region of interest of the user. The generic overview of R-NSGA-II procedure is shown in Fig. 1.

In order to incorporate the idea of reference point in NSGA-II, modified crowded operators called preference operator is used to select the subset of solutions from the last front which cannot be accommodated entirely to maintain the population size in the next population. This preference operator uses the preference

distance measurement instead of crowding distance as in NSGA-II. The preference distance represents how the solutions are closest to the reference points. The modification performed is shown in Fig 2. by R-NSGA-II nitching (diversity) strategy.

---

1 generate initial parent population
2 **repeat**
3 generate offspring population from parent population by applying Selection, Crossover and Mutation operators
4 combine parent and offspring population
5 place each individual in its respective front by applying fast non-dominated sort on combined population
6 calculate preference distance of each fronts' individual using nitching strategy specified in Fig. 2
7 make new parent population by selecting individuals which are in better front and having least preference distance
8 **until** (maximum number of generations)

Fig 1: Overview of R-NSGA-II Procedure

---

1. Assign rank to each solution with respect to each reference point according to calculated Euclidean distance (Equation 4).
2. Determine preference distance of each solution by selecting minimum rank with respect to all reference points.
3. Make groups of solutions by applying ε-clearing idea and retain one random solution from each group.

Fig 2: R-NSGA-II Nitching Strategy

$$d_{x,R} = \sqrt{\sum_{i=1}^{\text{Nobj}} \left(\frac{f_i(x) - f_i(R)}{f_i^{\max} - f_i^{\min}} \right)^2} \qquad (4)$$

Where $d_{x,R}$ - normalized Euclidean distance from solution x to reference point R, Nobj - number of objectives, $f_i^{\max}$ - maximum value of ith objective in the population, $f_i^{\min}$ - minimum value of ith objective in the population.

The ε-clearing idea specified in nitching strategy is similar to ε-dominance strategy used to control the spread of solutions near the preferred Pareto optimal regions. This spread (diversity) between solutions is maintained by ε-value which states the tolerance or precision specified by the user for objective values. The choice of ε-value is application specific and can be specified different for each objective.

4.3 Formulation and Genetic Operators

To solve the workflow scheduling problem, we formulated workflow elements in the population and development of fitness functions. Further, we applied genetic operators such as selection, crossover and mutation. The whole methodology is described in the following sub-sections clearly.

4.3.1 Population Formulation and Fitness Assignment

In an evolutionary algorithm, the population consists of number of individuals. An individual is formulated with two strings called task matching string (TMS) and scheduling order string (SOS). Initially, on which resource a task will execute is defined in the TMS randomly. Tasks' ordering is described by the SOS if they are matched on the same resource. SOS is also randomly generated while preservation of precedence constraints between workflow tasks. The fitness functions Ftime(S), Fcost(S) and Frel(S) are formed in order to evaluate individuals according to makespan, cost and reliability of the schedule respectively. These fitness functions are calculated from Equation (1) and (2) by adding the penalty value. On the violation of deadline and budget constraints, penalty is added respectively to objective functions otherwise not.

4.3.2 Selection Operator

Selection of individuals plays very important role in evolutionary algorithm by which unfitted individuals are rejected. We used binary tournament selection due to it's widely use in the past. In binary tournament selection one of two randomly individuals is selected based on their fitness value. Thus individual having good fitness value get more chance to be survive in the next generation.

4.3.3 Crossover and Mutation Operators

Crossover produces new individuals from the existing ones by interchanging machines (resources) of them. We have used one point crossover, which showed good performance for workflow scheduling problem. Mutation operator is used to explore new things which could not be exploited by crossover operator. In mutation, a task of the individual is reassigned on another resource randomly. Mutation operator used here is replacing mutation. We have applied crossover and mutation only on matching string.

5. EVALUATION AND DISCUSSION OF RESULTS

We used GridSim [22] toolkit to simulate workflow task scheduling for multi-objective optimization. GridSim is a java based toolkit for modeling and simulation of resource and application scheduling in large-scale parallel and high performance distributed computing environment such as Grid.

In our experiments, we simulated complex workflow applications consisting of 20 tasks on 8 resources and these resources are maintained by different organizations in the grid. Each resource's computational power rate in million instructions per second and their cost in dollars are generated non-uniformly. In the workflow, data units transferred between tasks are also generated non-uniformly. A router is used to make connections between available resources. Communication baud rate between resources is specified in terms of Mbps.

Two single objective optimization algorithms HEFT [9] and Greedy Cost are used to make deadline and budget effectively. HEFT is a time optimization workflow scheduling in which tasks are matched and scheduled based on minimum execution time of resources irrespective of resource's cost. Greedy Cost is

a cost optimization workflow scheduling algorithm in order to match and schedule tasks on cheapest resources. Thereby, HEFT gives minimum makespan Mmin and maximum total cost TCmax of the workflow schedule. And Greedy Cost gives maximum makespan Mmax and minimum total cost TCmin of the workflow schedule. Thus Deadline and Budget are formulated as:

$$\text{Deadline} = \text{Mmax} - C\,(\text{Mmax} - \text{Mmin}) \qquad (5)$$

$$\text{Budget} = \text{TCmax} - C\,(\text{TCmax} - \text{TCmin}) \qquad (6)$$

The value of parameter C can lies between 0.1 and 0.7. For both deadline and budget, we used 0.1, 0.4 and 0.7 to make loose, intermediate and stiff constraints respectively. The default parameter setting for our simulated MOEA approaches is mentioned in Table 1

Table 1. Default Setting for Evolutionary Algorithms

Parameter/Operation	Value/Type
Population size	10
Number of generations	200
Population initialization	Random
Crossover rate	0.9
Mutation rate	0.5
Selection operator	Binary tournament
Crossover operator	One Point crossover
Mutation operator	Replacing mutation

The Pareto optimal solutions of NSGA-II and R-NSGA-II, obtained after 200 generations at loose, intermediate and stiff constraints are shown in Fig. 3, 4 and 5. Fig. 3 and 4 shows, non-dominated solutions generated by R-NSGA-II are only in region of user interest rather than generation of wide spread non-dominated solution over the entire Pareto front as produced by NSGA-II at loose and intermediate constraints. But in Fig. 5 on stiff constraint, solutions obtained by both algorithms are in same region with slight difference because only a few non-dominated solutions exist with very small value of budget and deadline constraint.
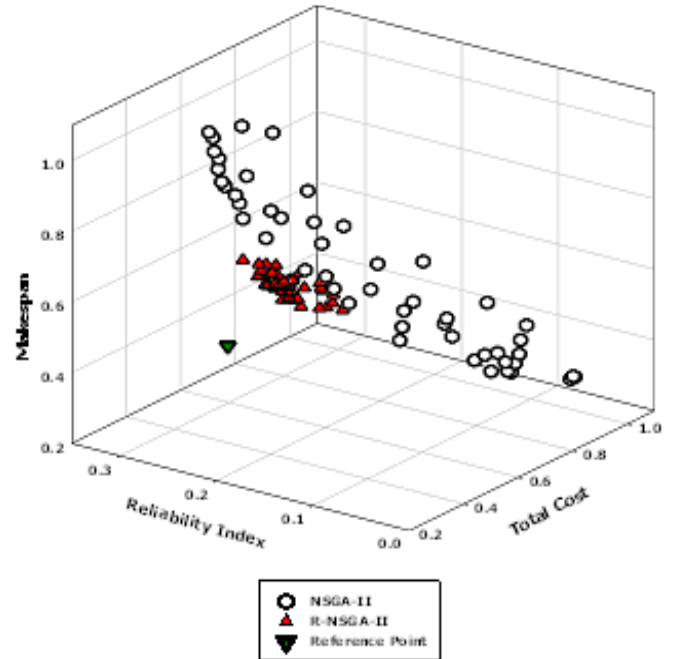

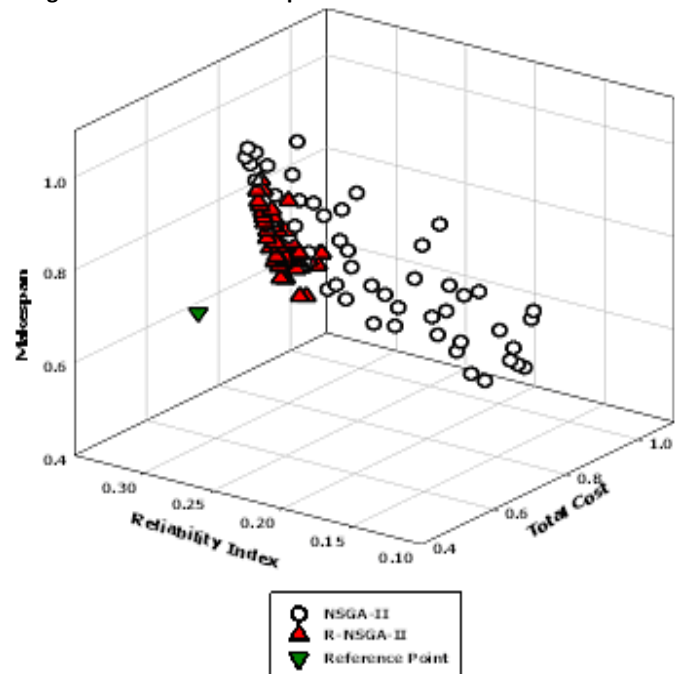**Fig. 3: Obtained Pareto Optimal front on loose constraint**.

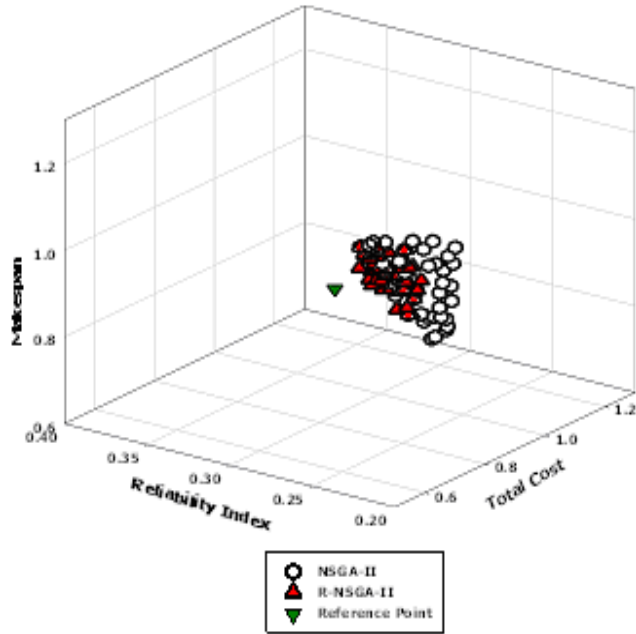**Fig. 4: Obtained Pareto Optimal front on intermediate constraint**

Fig. 5: Obtained Pareto Optimal front on stiff constraint.

Further Fig. 6, shows solutions generated with NSGA-II and R-NSGA-II when user have more than one preference region i.e. two reference points. Here we get multiple solutions in each region of interest simultaneously with same computation time. Thus R-NSGA-II is the better choice to have multiple tradeoff solutions when the user is having multiple regions of interest.
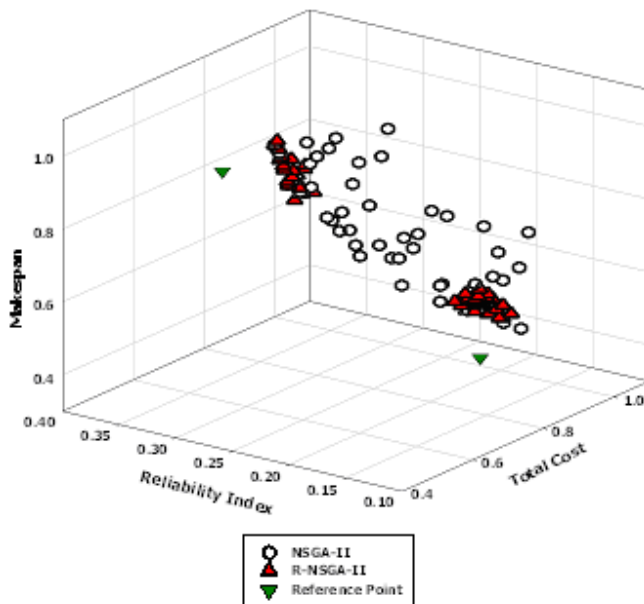

Fig. 6: Obtained Pareto Optimal front with Two reference Points

6. CONCLUSION AND FUTURE WORK

In this paper, our focus has been on the optimization of the workflow scheduling in a grid. Here, we exploit the multi objective evolutionary algorithms (MOEA's) approach to find more than one solutions not in the entire Pareto optimal front, but in the regions of Pareto optimality which are of interest to decision maker. With reference point based non dominated sorting algorithm we have been able to get the preferred set of scheduling solutions in the multiple regions of interest parally and multiple trade off solutions are generated that minimize the total execution time and cost along with the maximization of reliability in the close vicinity of each desired region of interest. In future work we will also enhance the approach by using hybrid Multi objective optimization.

7. REFERENCES

[1] Foster, I., Kesselman, C. 1999. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, ISBN 1-55860-475-8, San Francisco.

[2] Braun, T., Siegal, H., Beck, N. 2001. A comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems. In: Journal of Parallel and Distributed Computing, vol. 61, 810-837.

[3] X, He., XH, Sun., GV, Laszewski. 2003. QoS Guided min–min Heuristic for Grid Task Scheduling. In: Journal of Computer Science and Technology, vol. 18(4), 442-451.

[4] Wang, L., Siegel, H., Roychowdhury, V., Maciejewski, A. 1997. Task Matching and Scheduling in Heterogeneous Computing Environments using a Genetic-Algorithm-Based Approach. In: Journal of Parallel Distributed Computing, vol. 47, 9–22.

[5] Wieczorek, M., Prodan, R., Fahringer, T. 2005. Scheduling of Scientific Workflows in the ASKALON Grid Environment. SIGMOD Rec., vol. 34, 56-62, ACM

[6] Wieczorek, M., Podlipning, S., Prodan, R., Fahringer, T. 2008. Bi-criteria Scheduling of Scientific Workflows for the Grid. 978-0-7675-3156-4/08, IEEE.

[7] Yu, J., Buyya, R. 2006. Scheduling Scientific Workflow Applications with Deadline and Budget Constraints using Genetic Algorithms. Scientific Programming, vol. 14, 217-230.

[8] Tsiakkouri, E., Sakellariou, R., Zhao, H., Dikaiakos, M. 2005. Scheduling Workflows with Budget Constraints. In: CoreGRID Integration Workshop, 347-357.

[9] Haluk, T., Hariri, S., Wu, M. 2002. Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. In: IEEE Transactions on Parallel and Distributed Systems, vol. 13, 260-274.

[10] Prodan, R., Fahringer, T. 2005. Dynamic scheduling of Scientific Workflow Applications on the Grid: A case study. In: SAC 05: Proceedings of the 2005 ACM Symposium on Applied Computing, 687–694.

[11] Yu, J., Kirley, M., Buyya, R. 2007. Multi-objective Planning for Workflow Execution on Grids. In: Proceedings of the 8th IEEE/ACM International conference on Grid Computing, ISBN:978-1-4244-1559-5, doi.10.1109/GRID.2007. 4354110.

[12] Talukder, A., Kirley, M., Buyya, R. 2009. Multiobjective Differential Evolution for Scheduling Workflow Applications on Global Grids. doi: 10.1002/cpe.1417, John Wiley & Sons, Ltd.

[13] Deb, K. 2001 Multi-Objective Optimization using Evolutionary Algorithms. Wiley and Sons, England.

[14] Deb, K., Pratap, A., Aggarwal, S., Meyarivan, T. 2000. A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II. In: Parallel Problems Solving from Nature VI, 849-858.

[15] Zitzler, E., Laumanns, M., Thiele, L. 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: K.C. Giannakoglou, D.T. Tshalis, J. Periaux, K. D. Papailion, T. Fogarty (eds), Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, 95-100, Athens, Greece.

[16] Knowles, J., Corne, D. 1999. The Pareto Archive Evolution Strategy: A New Baseline Algorithm for Multi-Objective Optimization, In: The congress on Evolutionary Computation, pp. 98-105.

[17] K. Deb, J. Sundar, U. R. Rao, S. Choudhuri. 2006. "Reference Point Based Multi-Objective Optimization Using Evolutionary algorithms", International Journal of Computational Intelligence Research, ISSN 0973-1873 vol.2, No.3, 273–286.

[18] Camelo, M., Donoso, Y., Castro, H. 2010. A Multi-Objective Performance Evaluation in Grid Task Scheduling using Evolutionary Algorithms. In: Applied Mathematics and Informatics, ISBN: 978-960-474-260-8.

[19] Navimipour, N., Es-hagi, S. LGR. 2009. The New Genetic Based Scheduler for Grid Computing Systems. In: International Journal of Computer and Electrical Engineering, Vol. 1, No. 5, 610-614.

[20] Grosan, C., Abraham, A., Helvik, B. 2007. Multiobjective Evolutionary Algorithms for Scheduling Jobs on Computational Grids. In: International Conference on Applied Computing, Spain. ISBN 978-972-8924-30-0, 459-463.

[21] Dogan, A., Ozguner, F. 2005. Biobjective Scheduling Algorithms for Execution Time-Reliability trade-off in Heterogeneous Computing Systems. Comput. J., vol. 48, no. 3, 300-314.

[22] Buyya, R. GridSim: A Toolkit for Modeling and Simulation of Grid Resource Management and Scheduling, http://www.buyya.com/gridsim.

[23] Deb, K., Jain, S. 2002. Running Performance Metrics for Evolutionary Multi-objective Optimization. In: Simulated Evolution and Learning (SEAL-02), 13-20

[24] Do, T., Nguyen, T., Nguyen, D., Nguyen H., Le, T. Failure-aware Scheduling in Grid Computing Environments.

[25] Chitra, P., Revathi, S., Venkatesh, P., Rajaram, R. 2010. Evolutionary Algorithmic Approaches for Solving Three Objective Task Scheduling Problem on Heterogeneous Systems. In: 2nd International IEEE Conference on Advance Computing, ISBN 978-1-4244-4791-6/10, IEEE.