ISSN 2250-3153

An Evolutionary Algorithm in Grid Scheduling by multi-objective Optimization using variants of NSGA

Shahista Navaz, Uzma Ansari

Department of Computer Science and Engg. RITEE, Raipur.

Abstract- With the rapid development of network technology, Grid Computing has emerged for satisfying the increasing demand of the computing power of scientific computing community. 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.

In this work, we proposed the use of epsilon dominance based MOEA approach for the purpose of solving workflow scheduling problems in Grid. In one scheduling problem, we addressed two major conflicting objectives called make span and cost.

Index Terms- Multi-objective Optimization, NSGA, Workflow Grid Scheduling, Pareto dominance.

I. Introduction

The technology advancements in high speed networks and wide popularity of the Internet have led to the possibility of using vast number of geographically distributed resources owned by multiple organizations. These advancements have led to the foundation of new paradigm known as Grid Computing [1, 2, 3, 4]. Grid Computing is a type of parallel and distributed system that involves the integrated and collaborative use of computers, networks, scientific instruments depending on their availability, capability, cost and user requirements. 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 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).

II. 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 constraint respectively.

Bi-objective workflow scheduling problem: In this problem, we have considered two major objectives makespan and cost which are conflicting in nature. So, we have tackled this scheduling problem by considering the minimization of both makespan and cost under the deadline and budget constraints. Tri-objective workflow scheduling problem: In dynamic environment of grid where resources can fail inevitably, a scheduling decision is still challenging area and it should consider reliability of resources while generating schedule (s) in addition to other objectives. Failures of resources can have adverse effects on the performance of workflow application. So, we have optimized another objective called reliability along with makespan and cost objectives, to incorporate the failure affect of resources in scheduling decision.

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 mentioned work, we have proposed workflow scheduling based on referenced Point based NSGA-II (RNSGA- II) [17] considering three objectives. Using this RNSGA- 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.

III. 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 ti 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 rj R is associated with three values: completion time, total cost and of executing the task ti on resource rj. Completion time of the task ti is denoted by time (ti) and total cost of the task ti which includes both communication and service cost for executing task ti is denoted by cost (ti). Reliability of the task is calculated with the help e-RI. Here RI is the reliability index. By minimizing 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.

Minimize makespan (S) = max time (ti) (1)

Minimize Cost (S) = cost (ti) (2)

Minimize Reliability(S)=RI= (time(ti)×γ_j) (3)

Subject to Time(S) < D and Cost(S) < B

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

IV. MULTIOBJECTIVE OPTIMIZATION SOLVING WORKFLOW GRID SCHEDULING

4.1 Problem with Multi-objective Optimization

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) fm(s)) consists of m objectives. Consider two vector solutions s1 and s2. Then solution s1 is said to dominate s2 if following two conditions are true:

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

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) [12] 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 [11]. 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. 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 NSGAII. The preference distance represents how the solutions are closest to the reference points. The NSGA-II algorithm is shown below.

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 strategy below.
7. Make new parent population by selecting individuals which are in better front and having least preference distance
8. until (maximum number of generations)

4.3 FORMULATION AND OPERATORS USED

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 F_{time} (S), F_{cost} (S) and F_{rel} (S) are formed in order to evaluate individuals according to makespan, cost and reliability of the schedule respectively.

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. 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.

V. EVALUATION AND DISCUSSION OF RESULTS

5.1 PERFORMANCE EVALUATION

There are two distinct goals of multi-objective optimization: (i) Convergence and (ii) Diversity. In particular, these two goals are orthogonal to each other. Convergence requires a search towards the Pareto optimal front, while the diversity requires a search along the Pareto optimal front. A multi-objective evolutionary algorithm is said to be good, if both goals are satisfied adequately. Since convergence to the Pareto optimal front and maintaining distribution among solutions are two distinct, no single metric can evaluate the performance of an evolutionary algorithm in an absolute sense. So, two performance metrics are required, each one corresponds to one of two distinct goals of multi-objective optimization. For the performance comparison between NSGA with one point crossover and NSGA with uniform crossover , we conducted our experiment over 10 runs. To measure the quality of both evolutionary algorithms, we have used two metrics Generational Distance (GD) and Spacing. Further, we have measured the Computation Time taken by each algorithm. All these performance measures are described in the following sections.

5.2 GENERATIONAL DISTANCE METRIC

Generational Distance 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 both algorithms over 10 runs i.e. P* contains all the non-dominated solutions obtained after combining outputs of 10 runs of both NSGA with one point crossover and NSGA with uniform crossover evolutionary algorithm. Let, Q is the set of non-dominated solutions obtained using algorithm for which we calculate GD metric, then mathematically GD is expressed as follows:

$$GD = \frac{\left(\sum_{i=1}^{|Q|} d_i^2\right)^{1/2}}{|Q|}$$

di is the Euclidean distance between the solution i of Q ($i \in Q$) and the nearest solution of P* and it is calculated as below:

$$d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{m=1}^{M} \left(f_m^{(i)} - f_m^{(k)}\right)^2}$$

Where, M is the number of objectives. $f_{m(i)}$ is the m-th objective value of the i-th solution of Q and $f_{m(k)}$ is the m-th objective value of the k-th member of P*. Intuitively, an algorithm having a small value of GD is better. Furthermore, we normalized distance measure specified in Equation before its calculation, because of differing magnitude of objectives.

5.3: SPACING METRIC

On the other side, Spacing metric was also used to evaluate obtained solutions of NSGA with one point crossover and NSGA with uniform crossover in order to check diversity among solutions. Let, Q is the set of non-dominated solutions obtained using algorithm for which we calculate spacing metric, Spacing metric is calculated with a relative distance measure between solutions of Q, as follows:

$$Spacing = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} \left(d_i - \bar{d}\right)^2}$$

Where d_i is the distance between the solution i and its nearest solution of Q as specified. Thus, an algorithm obtaining a non-dominated set of solutions having a smaller spacing is better. Further, distance measure calculation is also normalized as normalization of distance measure in GD metric.

5.4 COMPUTATION TIME

Computation time is the time taken by an evolutionary algorithm to produce a set of non-dominated solutions. In order to achieve two distinct goals convergence and diversity of multi-objective optimization, an MOEA algorithm needs to be computationally fast enough. So that users can get non-dominated solutions quickly.

NSGA with one point crossover show that it is worst in seven runs as compared to NSGA with uniform crossover. We can observe that in extent of minimization and good spread between obtained solutions simultaneously, the NSGA with uniform crossover appears to be better in four runs as compared to NSGA with one point crossover which is better in only two runs. Again, the time taken by NSGA with one point crossover is significantly large as compared to NSGA with uniform crossover in ten runs of execution. Furthermore, in average case, again NSGA with uniform crossover is better than NSGA with one point crossover in terms of convergence and distribution among solutions.

NSGA with one point crossover	2820	3750	4460	5610	6320	9120
NSGA with uniform crossover	2789	3654	4340	5530	6360	9010

Table: NSGA with one point and uniform crossover

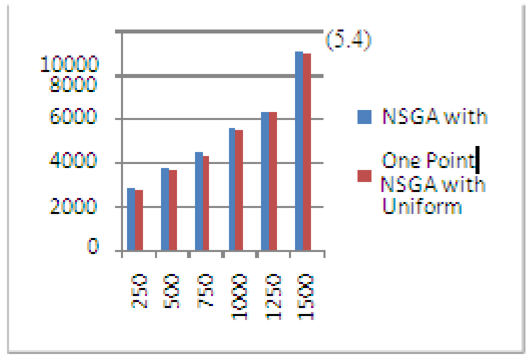

Fig: Graph between NSGA single point and uniform point crossover

VI. CONCLUSION AND FUTURE WORK

The work in this dissertation focused on both the variants of NSGA i.e. NSGA with one point crossover and NSGA with uniform crossover and also focused on optimizing the multiple objectives of the workflow scheduling problems in Utility Grid systems. In multi-objective optimization problem, multiple trade-off Pareto solutions are produced for the maximum satisfaction of user. In this work, we have compared which cross over is better when comparing the NSGA one point crossover with uniform crossover approach and to solve scheduling workflow problems in (bi-objective optimization evolutionary algorithm). 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. The Pareto solutions obtained by NSGA with uniform crossover are compared with the Pareto solutions obtained by another well known (NSGA-I) and a statistical analysis of their results has been presented to show the quality of each algorithm on different constraint levels. To measure the quality of these approaches, we selected two metrics called GD (convergence metric) and Spacing (diversity metric). A statistical analysis showed that NSGA with uniform crossover outperforms NSGA with one point crossover in terms of both convergence towards Pareto optimal front and maintaining good spread between obtained solutions with a small computation overhead. In future:-

- The workflow scheduling problems considered in this work focus on time, cost and reliability objectives. However, other objectives such as flow time, due date of each individual task, security levels and resource utilization could also be required for many work applications. For example, confidential applications must be executed on resources with high security levels. User can also impose deadline on each individual task of the workflow.

- Cost and failure of communication links can be taken into account in scheduling workflow applications.
- Capability of other MOEA approaches can be explored inworkflow grid scheduling to obtain multiple trade-off solutions for the users.
- Particle Swarm Optimization (PSO) and Bacterial Foraging Optimization (BFO) techniques can be used in order to find non-dominated solutions towards Pareto optimal front.

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,

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Young, L., McGough, S., Newhouse, S., Darlington, J.: Scheduling Architecture and Algorithms within the ICENI Grid Middleware, UK e-Science All Hands Meeting, IOP Publishing Ltd, Bristol, UK, Nottingham, UK,**ISBN:** 1-9044-2511-9, pp. 5-12, Sep(2003).

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

AUTHORS

First Author –Shahista Navaz, Department Of Computer Science And Engg, RITEE, Raipur, Email: shahista.navaz@gmail.com
Second Author – Uzma Ansari, Department Of Computer Science And Engg, RITEE, Raipur, Email: arshi.uzma@gmail.com