CHAPTER 6

CONCLUSION

6.1 INTRODUCTION

Grid computing is still in the development stage, and many challenges are to be addressed. Among these, improving its efficiency is a key issue. The question is: How do we make use of a large number of computers worldwide, ranging from simple laptops, to clusters of computers and supercomputers connected through heterogeneous networks in an efficient, secure and reliable manner?

For the majority of Grid systems, scheduling is a very important mechanism. In the simplest of cases, scheduling of jobs can be done in a blind way by simply assigning the incoming tasks to the available compatible resources. Nevertheless, it is a lot more profitable to use more advanced and sophisticated schedulers. Moreover, the schedulers would generally be expected to react to the dynamics of the Grid system, typically by evaluating the present load of the resources, and notifying when new resources join or drop from the system. Additionally, schedulers can be organized in a hierarchical form or can be distributed in order to deal with the large scale of the Grid.
6.2 SUMMARY OF CONTRIBUTIONS

This thesis starts with Chapter 1, by reviewing the state-of-the-art survey of problems pertaining to the Grid computing with respect to optimization. Also introduces about the present motivations of this research and explains the background of this research work and also summarizes the organization of this research in detail.

Chapter 2 reviews the thorough literature of grid task scheduling using various heuristic task scheduling algorithms.

Chapter 3 proposes a method of scheduling called as residual fair sharing algorithm for scheduling of tasks in a heterogeneous grid environment and the results are compared with the conventional max-min scheduling algorithm. The experimental study revealed that the residual sharing algorithm performs better with respect to max min algorithm especially; there is a noticeable reduction in the time complexity between the two algorithms. The simulations have been conducted by thousands of tasks of varying size and workload variance, submitted to a multiprocessor computing system comprising of hundreds of processors of varying capacity. Experimental results and comparisons of the two scheduling schemes indicate that our proposed Residual Fair Sharing scheduling scheme is fairer and exploit the available multiprocessor Grid resources. The experiments also indicate that the Residual fair sharing algorithm is less sensitive to processor capacity variations than max-min sharing scheme. However, in all conditions, the proposed Residual algorithm is more effective and outperforms all the traditional scheduling algorithms. Studies are under way to apply Heuristic concepts to resolve interdependencies and further optimize the residual fair share scheduling.
The problem of scheduling of tasks to be executed on a multiprocessor system is one of the most challenging problems in parallel computing. In chapter 4, DE-VNS with a modified mutation operator for the multiprocessor scheduling problem is used to get a reasonable finishing time or makespan of the schedule. The tasks have precedence relationship and are represented by using Directed Acyclic Graph. The individuals are represented in the form of vectors having a number of cells. The precedence relations of the tasks are considered. The initial population is generated randomly and fitness values are calculated for each individual. The crossover and mutation operations are performed until algorithm converges. The crossover of the tasks also guarantees that the new strings generated are legal. Furthermore, a tabu list is used for avoiding repetition of the parents whose children have been already searched in previous generations. The proposed algorithm will give a reduced finish time or makespan for the schedule of tasks represented by using the DAG with a reasonable execution time. The experimental result indicated that better solutions with less execution time are obtained by a combination of Genetic Algorithm.

In chapter 5, this research analyzed the job scheduling problems on computational grids. For scheduling problems, we consider several nature-inspired metaheuristics including evolutionary algorithms, simulated annealing, ant colony optimization and TLBO algorithms. We proposed a novel approach based on fuzzy TLBO. The representations of the position and velocity of the candidates in the conventional TLBO is extended from the real vectors to fuzzy matrices. The proposed approach is to dynamically generate an optimal schedule so as to complete the tasks within a minimum period of time as well as utilizing the resources in an efficient way. We also evaluated the performance of a fuzzy TLBO for grid job scheduling and compared the performance with GA and SA. Empirical results revealed that the proposed approach can be applied for job scheduling. When compared to GA and
SA, an important advantage of the TLBO algorithm is its speed of convergence and the ability to obtain faster and feasible schedules.

6.3 FUTURE DIRECTIONS

Besides the many aspects and facets of the Grid scheduling problem presented in the previous sections, there still remain other issues to be considered. We briefly mention some of them here.

Security is an important aspect to be considered in Grid scheduling. It can be seen as a two-fold objective: tasks could have security requirements to be allocated in secure nodes, while the node itself could have security requirements; that is, the tasks running in the resource will not “watch” or access other data in the node. It should be noted that current security approaches are treated at different levels of Grid systems and independently of the Grid schedulers. It is challenging to incorporate the security/trust level as one of the objectives of the scheduling by using trust values that range from very untrustworthy to very trustworthy scale. Moreover, the objective is to reduce the possible overhead to the Grid scheduler that would introduce a secure scheduling approach.

Other important issues are related to data-aware scheduling T. Kosar, et al. (2009). Most current Grid approaches are task-oriented or resource-oriented approaches. For instance, tasks are assumed to include all data needed for its computation or tasks are just the processes and data is assumed to be available in Grid nodes. However, with the ever-increasing complexity of large-scale problems in which both tasks and data are to be scheduled, an integrated scheduling approach that would optimize allocation of both the task and the data is required.