International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume6(2014) pp.466–473

Grid Scheduling Heuristic Methods: State of the Art

Rajni Aron, Indreveer Chana

Thapar University, Computer Science and Engineering Department,
Patiala, 147004
rajni3387@gmail.com
inderveer@thapar.edu

Abstract: **Efficient management of the Grid resources require fair resource allocation and scheduling. The mapping of jobs to the resources in the Grid is a NP complete problem. NP-complete problems are often solved using heuristic techniques. Over the time, heuristics and meta-heuristics have proved to provide an optimum solution for the combinatorial optimization problems. In this paper, a survey of scheduling algorithms and heuristic approaches is done. The motivation of this survey is to encourage the amateur research of heuristics based scheduling in Grid computing, so that the researchers can understand the concept of heuristic approaches for resource scheduling in the Grid computing. The comparison of the heuristic has been shown and experimental result shows that the hyper-heuristics can be of significance importance in Grid scheduling.**
Keywords: Grid scheduling; Heuristic methods; Grid computing.

I. Introduction

The emergence of Grids which collaborate resources from multiple organizations to fulfill the computing needs of applications with varying resource requirements has grownup in the form that it satisfies the increasing demand of the scientific computing. Grid computing provides the facility of resource sharing of multiple virtual organizations [1]. Due to the heterogenous and dynamic nature of the resources in the Grid environment, resource management and resources scheduling are significantly challenging tasks. The task of mapping jobs to the available computing nodes or scheduling of the jobs on the grid is a NP-complete problem. NP-Complete problem is often solved using heuristic methods [2]. Various heuristic methods are used to produce solutions of acceptable quality in reasonable time for Grid scheduling problems. This paper addresses the various scheduling approaches in Grid computing. We have presented different heuristic approaches for Grid scheduling in this paper. Heuristics, Meta-heuristics and Hyper-heuristics have proven to be efficient in solving Grid resource scheduling problems. So, heuristic methods play an important role for getting the optimal solution of Grid scheduling problems.
The motivation behind this paper is to explore the existing scheduling approaches which are applicable in the Grid environment and schedule resources to the preferred jobs which will return optimum results to the Grid users.

This paper presents survey on heuristic methods for Grid scheduling. This paper presents existing scheduling approaches in section 2. In section 3, a description of heuristic approaches has been presented. Section 4 shows analysis and comparison between various heuristic approaches. We provide the conclusion in section 5.

II. Grid Scheduling

Grid scheduling is defined as the process of making scheduling decisions involving allocating jobs to resources over multiple administrative domains [3]. This process includes searching multi administrative domains to use available resources from the Grid infrastructure in order to satisfy the requirements of the user. Figure. 1 depicts a model of Grid scheduling systems in which functional components are connected. Grid users will submit the application through a portal. After this it will further contact the grid scheduler for user applications. The Grid scheduler via middleware consults local resource manager for the availability of the resources. Local resource manager consults with different resource providers and accordingly will see availability of the job. Then, the local resource manager will submit the job. After getting the result, it will be given to Grid scheduler through a local scheduler. Then Grid application gets the information and finally the user will collect the information.

A. *Grid Scheduling approaches*

1. **Local versus Global**
Scheduling can be either local or global. The local scheduling discipline determines how the processes resident on a single CPU are allocated and executed where as a global scheduling policy uses information about the system to allocate processes to multiple processors so as to optimize a system-wide performance objective. Grid scheduling should be done as global scheduling.

2. **Static versus Dynamic**
In Grid, both static and dynamic types of scheduling have been adopted. In static scheduling, information regarding all the resources in the Grid as well as all the tasks in an application are assumed to be available by the time, the application is scheduled but in the case of

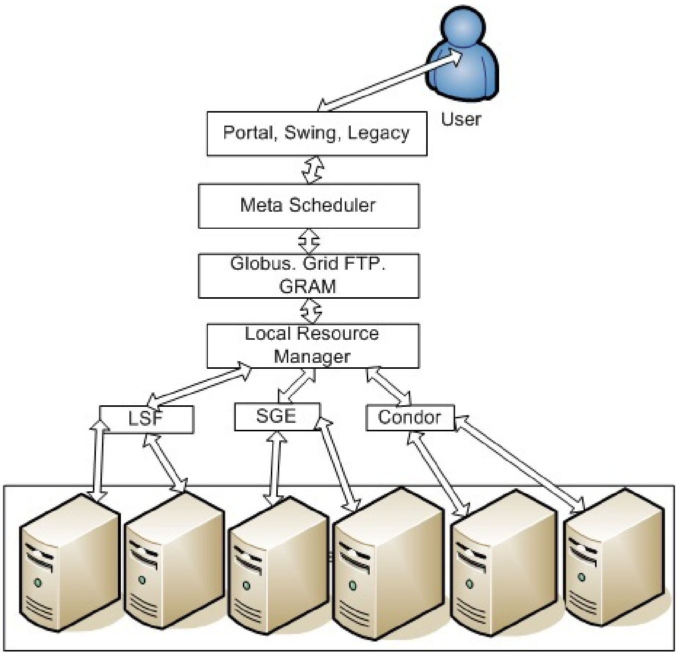

Figure. 1: A logical Grid scheduling architecture

dynamic scheduling, the basic idea is to perform task allocation on the fly as the application executes.

3. **Centralized versus Decentralized**
In Grid scheduling, the responsibility for making global scheduling decisions may lie with one centralized scheduler, or be shared by multiple distributed schedulers. In centralized scheduling, Grid schedular has more control on the resources and in this case efficient scheduler can be designed. Centralized Grid scheduling algorithm can be easily implemented but it suffers from lack of scalability, fault tolerance etc. Therefore, centralized scheduling is not useful for large scale Grids. In decentralized scheduling, Grid schedulers have no centralized control over the resources and to design an efficient scheduler is a challenging task. In this case local schedulers play an important role in scheduling and they also manage and monitor the status of the resources.

4. **Co-operative versus Non-cooperative**
In Co-operative scheduling, each Grid scheduler carries out its own scheduling tasks, but all schedulers are working toward a common system-wide goal. Scheduling is done through the cooperation of Grid users, rules and policies. In the non-cooperative case, each scheduler acts alone as an autonomous entity and arrives at decisions regarding their own optimum objects independent of the effects of the decision on the rest of the system [4].

5. **Approximation versus Heuristics**
The approximate algorithms use formal computational models, but instead of searching the entire solution space for an optimal solution, they are satisfied when a solution that is sufficiently "good" is found. Heuristic algorithms are more adaptive to the Grid scenarios where both resources and applications are highly diverse and dynamic, so heuristics are considerably a defacto approach for solving Grid scheduling problems.

III. Heuristic approaches

Heuristic approaches can be applied to Grid scheduling problem because Grid scheduling has various important issues that need to be addressed such as heterogeneity of the resources, dynamic and autonomous nature of Grid resources and finally resource providers and resource consumers have different policies for the execution of their applications.

A. Local based Heuristic approaches

Local search heuristic approaches is a family of methods that explore the solution space by starting at an initial solution, and constructs a path in solution space during the search process [6]. Local search heuristic approaches improve solutions through neighborhood search. The main objective of this local search based heuristic approach is to gain feasibility as soon as possible. They have been applied successfully to many industrial problems and performance of local search based heuristic approaches depending on construction of neighborhood.

Tabu Search Tabu Search (TS) is a high level heuristic procedure for solving optimization problems and was proposed by Glover in 1986. Tabu search is a meta-heuristic that guides a local search procedure to explore the solution space beyond local optimality [5].

Advanatges

- TS avoids entrapment in local minima and continues the search to give a near optimal final solution.
- TS is very general and conceptually much simpler than other meta heuristic algorithms such as genetic algorithm, simulated annealing and ant colony optimization algorithms.
- TS is very easy to implement.
- TS does not require special memory space.
- TS takes short searching time to solve combinatorial optimization problems.
- TS uses specific set of constraints, known as tabu conditions, in order to avoid blind search.

Disadvantages

- TS often gets locked in looping from one local optimum to another.
- TS has low global search capability.

Hill Climbing Hill Climbing (HC) is a graph search algorithm where the current path is extended with a successor node which is closer to the solution than the end of the current path. Hill climbing is logical and beneficial especially in situations where the search space is of simple nature with no more than a single maxima or minima [6].

Advantages

- HC is a local search heuristic technique.

- Hill climbing is simpler and straight forward in comparison to other heuristics.

Disadvantages

- In case of hill climbing, the solution is better than all of its neighbors, but it is not better than some other states far away.
- There is a flat area of the search space in which all the neighboring states have the same value.
- HC is a local method and it moves in many directions at a time.

Simulated Annealing Simulated Annealing (SA) heuristic approach was proposed by kirkpatrick et al in 1983. The simulated annealing process consists of first melting the system being optimized at high effective temperature, then lowering the temperature by slow stages untill the system freezes and no further changes occur [7]. SA is an iterative technique that considers only one possible solution (mapping) for each meta task at a time [8]. This solution uses the same representation as the chromosome for the genetic algorithm. The initial implementation of SA was evaluated and then it was modified and refined to give a better final version. SA uses a procedure that probabilistically allows poorer solutions to be accepted in an attempt to obtain a better search of the solution space.
Advantages

- SA is guaranteed to converge in asymptotic time.
- SA can deal with arbitrary systems and cost functions.
- SA statically guarantees to find an optimal solution.
- SA is relatively easy to code, even for complex problems.
- SA is a robust heuristic to implement and has an ability to provide reasonably good solutions for many combinatorial problems.

Disadvantages

- SA has a difficulty in defining a good cooling schedule which is important both in single and multi objective optimization.
- In case of SA, if there is a repeated annealing with 1/logk then the scheduling is very slow, especially if the cost function is expensive to compute.
- SA is often comparable to heuristics.
- The main drawback of simulated annealing is that there is a need for a great deal of computer time for many runs and carefully chosen turnable parameters.

B. Population based Heuristic approaches

Population-based heuristic is a large family of methods which are highly efficient for solving combinatorial optimization problems. However, when the objective is to find feasible solutions of good quality in short execution times, as in the case of Grid scheduling, the inherent mechanisms of these methods can be exploited to increase the convergence of the method [6].
Genetic Algorithms Genetic Algorithm (GA) was proposed by holland et al. Genetic algorithms are playing an increasingly important role in studies of complex adaptive systems, ranging from adaptive agents in economic theory to the use of machine learning techniques in the design of complex devices such as aircraft turbines and integrated circuits [9]. GA is a famous stochastic optimization algorithm which uses biologically inspired techniques such as genetic inheritance, natural selection, mutation and sexual reproduction (recombination, or crossover) [10]. Genetic algorithms are useful heuristics to find a near optimal solution in large search spaces [8]. In GA, a point in search space is represented by a set of parameters and these parameters are known as genes and a set of genes is known as string or a chromosome. A fitness function must be devised for each problem to be solved. Each chromosome is assigned a fitness value that indicates how closely it satisfies the desired objective. Given a particular chromosome, the fitness function returns a single numerical fitness or figure of merit, which will determine the ability of the individual, which that chromosome represents [11][26]. A set of chromosomes is called population. Reproduction is another critical attribute of GAs where two individuals selected from the population are allowed to mate to produce offspring, which will comprise the next generation. Having selected two parents, their chromosomes are recombined, typically using the mechanisms of crossover and mutation. Mutation provides a small amount of random search, and helps ensure that no point in the search space has a zero probability of being examined. If the GA has been correctly implemented, the population will evolve over successive generations so that the fitness of the best and the average individual in each generation increases towards the global optimum. The genetic algorithms have been found to be very powerful in finding out a global minima [12][13]. Genetic algorithms have been applied to many classification and performance tuning applications in the domain of Knowledge Discovery in Databases (KDD) [10].
Advantages

- In case of GA, there is no need of analytical knowledge.
- GA is easy to understand and implement.
- GA supports multiobjective optimization.
- GA is easy to parallelize and no derivatives are required.
- GA works on a wide range of problems and has better global capability.

Disadvantages

- Genetic algorithm requires much more evolution functions than linearized methods.
- There is no guaranty of convergence to a local minima.
- It converges to a local optima or an arbitrary point rather than the global optima of the problem.
- GA has a slow convergence rate and premature convergence.

- It can not use the feedback of a system.

Memetic Algorithm Memetic Algorithm (MA) is an extension of genetic algorithm. Memetic algorithms are evolutionary algorithms that can be applied on a local search process to refine solutions for hard problems. Memetic algorithms are the subject of intense scientific research and have been successfully applied to a multitude of real-world problems ranging from the construction of optimal university exam timetables, to the prediction of protein structures and the optimal design of space-craft trajectories [14].
Advanatges

- Memetic algorithm can handle complex objective functions.
- It combines the advantages of local search and genetic algorithm for optimization problems.
- MA can be used for global search.
- It is based on a genetic algorithm and extended by a search technique to further improve individual fitness that may keep with the population diversity and reduce the likelihood premature convergence.

Disadvanatges

- MA requires a considerable amount of time and memory needed for improvement of its performance.
- MA can be used only in non-linear continuous multiobjective combinatorial optimization problems.

Ant Colony Optimization Ant Colony Optimization (ACO) was proposed by Marco Dorigo in 1992 [16].The real power of ants resides in their colony brain. The self-organization of those individuals is very similar to the organization found in brain-like structures. Like neurons, ants use mainly chemical agents to communicate. One ant releases a molecule of pheromone that will influence the behavior of other ants [15]. Ant algorithms are often compared with other evolutionary approaches such as Genetic Algorithms, Evolutionary Programming and Simulated Annealing. It is important to remember that Ant algorithms are non-deterministic and rely on heuristics to approximate to a sub-optimal solution in cases where the number of combinations is extremely huge and is impossible to calculate using a deterministic algorithm [16].
Advantages
Following advantages have been identified in [17]:

- ACO is versatile and can be applied to similar versions of the same problem; for example, there is a straightforward extension from the Traveling Salesman Problem (TSP) to the Asymmetric Traveling Salesman Problem (ATSP).
- It is robust and can be applied with only minimal changes to other combinatorial optimization problems such as the Quadratic Assignment Problem (QAP) and the Job-Shop Scheduling Problem (JSP).
- It is a population based approach. This is interesting because it allows the exploitation of positive feedback as a search mechanism. It also makes the system amenable to parallel implementations.
- It can be used for static and dynamic combinatorial optimization problems.
- ACO convergence is guaranteed and it can be used for solving constrained discrete problems.
- ACO has the powerful feedback capability which can increase the speed of evolution of algorithm to make algorithm convergence possible in the end.

Disadvantages

- ACO's convergence rate is slow in comparison to other heuristics.
- ACO performs poorly for larger city in Traveling salesman problems.
- In ACO, there is no centralized control to guide and provide good solutions.
- ACO can be applicable to only discrete problems and theoretical analysis in ACO is difficult.

Particle Swarm Optimization Particle Swarm Optimization (PSO) is a method for performing numerical optimization without explicit knowledge of the gradient of the problem to be optimized. PSO is one of the latest evolutionary optimization techniques inspired by nature and was introduced in 1995 by Kennedy and Elberhart [18]. It simulates the process of a swarm of birds preying. It has the better ability of global searching and has been successfully applied to many areas [27][28]. A flock or swarm of particles is randomly generated. Initially, each particle position represents a possible solution point in the problem space. The fitness value of each particle is evaluated by the objective function to be optimized. Each particle remembers the coordinates of the best solution (gbest) achieved so far. The coordinates of current global best (pbest) are also stored.
Advantages

- PSO is a robust stochastic optimization based on the movement and intelligence of swarms.
- There is no selection and crossover parameter like genetic algorithm
- PSO is easy to implement, few parameters to adjust, computationally efficient etc.
- PSO is efficient for global search algorithm.

Disadvantages

- PSO has a weak local search.
- PSO has a slow convergence rate in refined search strategy.

Bacterial Foraging Optimization Bacterial foraging optimization algorithm was proposed by Passino [24]. It is population based numerical optimization algorithm based on foraging behavior of Escherichia coli bacteria. In the foraging theory, the objective of the animal is to search and obtain nutrients in a fashion that energy intake per unit time (E/T) is maximized. Foraging is a process in which a group of

bacteria moves in search of food in a region, they decide whether or not to enter into a possible food region and then search for a new food region so as to get high quality of nutrients. The bacterial foraging process consists of three main mechanisms: Chemotactic, Swarming, Reproduction and Elimination-dispersal event. Chemotactic is the process of simulating the movement of E.coli bacteria, which is carried in a flagella, through swimming and tumbling. The cell also repels a nearby cell in the sense that it consumes nearby nutrients and so it is not physically possible to have two cells at the same location. A bacterium in times of stress releases attractants to signal the bacteria to swarm together. After chemotactic steps, a reproduction step is taken. Fitness value of bacteria is sorted in an ascending order.The least healthy bacteria eventually dies while each of the healthier bacteria (those yielding lower value of the objective function) asexually splits into two bacteria, which are then placed in the same location. This keeps the swarm size constant. Elimination event may occur due to sudden changes like a significant local rise of temperature or a part of them may move to other regions in the environment that will effect the behavior of bacteria heavily. The elimination and dispersal event destroys the performance of chemotactic event but dispersal may place bacteria near good sources of food [25].
Advantages

- The BFOA is more adaptive.
- Its performance is high with respect to speed of convergence, quality of solution and rate of success.

Disadvantages

- It's major disadvantage is it's premature convergence.

C. Meta-heuristics

Meta-heuristics support in decision-making with robust tools that provide high-quality solutions to important applications in business, engineering, economics and science in reasonable time horizons [19] .
Advanatges

- Meta-heuristic is an iterative master process that guides and modifies the operations of subordinate heuristics in order to produce high quality solutions.
- Meta-heuristics are very flexible to solve real problems.
- Meta-heuristics are often used for global optimizers.

Disadvantages

- Meta-heuristic approaches perform well on a particular real-world problem but may not work on all problems.
- Meta-heuristics may produce very poor solutions for other problems or even for other instances of the same problem.
- It requires extensive knowledge in both problem domain and appropriate heuristic techniques.
- Meta-heuristic is quite expensive to implement.
- Meta-heuristics are not suitable in those situations where problems data and business requirements change frequently over time.
- Meta-heuristics are heuristics which control the search in a space of solutions performed by a single low level heuristic.
- Optimality may not be guaranteed in meta-heuristics.
- There is a lack of theoretic basis and it requires multiple search parameters.
- Meta-heuristic algorithms like tabu search, ant colony etc have different searches but sometimes different searches may yield different solutions to the same problem.

D. Hybrid Heuristics

Hybrid strategies have been constructed to exploit the meta-heuristic techniques. To get a better result of genetic algorithm, it has been hybridized with local search methods as tabu search and simulated annealing etc. The major advantage of parallel hybrids implemented on shared-memory parallel architectures is their simplicity [20].
Advantages

- Hybrid-heuristics have a better convergence.
- hybrid-heuristics are more efficient in comparison to genetic algorithm.

Disadvantages

- Hybrid-heuristics are not easy to implement.
- Hybrid-heuristics are time consuming.

E. Hyper-heuristic

The term Hyper-heuristic describes heuristics to choose heuristics in the context of combinatorial optimization. A hyper-heuristic can be seen as a high level methodology which when given a particular problem instance or a class of instances and a number of low-level heuristics, automatically produces an adequate combination of the provided components to effectively solve the given problems [21].
Advantages

- Hyper-heuristics operate in a space of heuristics, choosing and applying one low-level heuristic from a given set at each decision point.
- Hyper-heuristics do not require knowledge of each low-level heuristic.
- Hyper-heuristics are robustness and re-applicability heuristics.

Disadvantages
Some of the disadvantages of hyper-heuristics have been identified by Chakhlevitch K. [22].

- Some hyper-heuristic techniques make use of additional problem specific knowledge. Such knowledge can be used to describe the current state of the problem in order to select a suitable low-level heuristic in hyper-heuristics employing learning classifier systems. In indirect GAs, a portion of problem-specific in formation is often injected into the chromosome.

- For many hyper-heuristics, a significant amount of parameter tuning is required in order to find good parameter settings for a given problem.

- A large number of problem instances may be required for training and testing of the method in order to accumulate enough knowledge to make the right choice of low-level heuristics. However, from any real-world problems the problem data are not easily available and randomly generated instances may not adequately represent the real distribution.

- Many hyper-heuristic methods are only tested on a relatively simple bench- mark problems for which the best solutions (often optimal) as well as an effective low-level heuristics are known in advance. There is no evidence that such hyper-heuristics would be effective in more complex real-world situations.

IV. Analysis and comparison between various heuristic approaches

Different heuristic techniques were evaluated using GridSim toolkit [23]. GridSim toolkit provides facilities for modeling and simulation of resources and network connectivity with different capabilities, configurations and domain. It also supports primitives for application composition, information services for resource discovery and interfaces for assigning application tasks to resources and managing their execution. The following are the reasons for the GridSim toolkit to be used for evaluation.

- It allows modeling of heterogeneous type of resources.

- Resources capability can be defined in the form of Millions instructions Per Second (MIPS) as per Standard Performance Evaluation Corporation (SPEC) benchmark.

- There is no limit on the number of application jobs that can be simulated.

- Multiple user entities can submit tasks for execution simultaneously.

- Statistics of all or selected operations can be recorded and they can be analyzed using GridSim statistics analysis methods.

- It supports simulation of both static and dynamic schedulers.

- Application tasks can be heterogeneous, and they can be CPU or I/O intensive.

We simulated the Grid with heterogenous and dynamic nature of resources having different Multiple Instruction Per Second (MIPS). Each resource had different number of Processing Element (PE) ranging from 3 to 10. The cost per second of each resource is varied between G$4 to G$5. We performed the scheduling experiment by setting the value of jobs varying from 100 to 300. The execution time is recorded to analyze the feasibility of the algorithm. Figure 2 shows the comparison of heuristic approaches for scheduling 100 jobs on 50 resources. Figure 3 shows the comparison of heuristic approaches with varying number of jobs. It is experimentally shown that hyper heuristic experimentally gives better result than the individual hybrid heuristics in test cases. Hyper heuristic performs on the search of heuristic instead of directly performing on solution of problems.

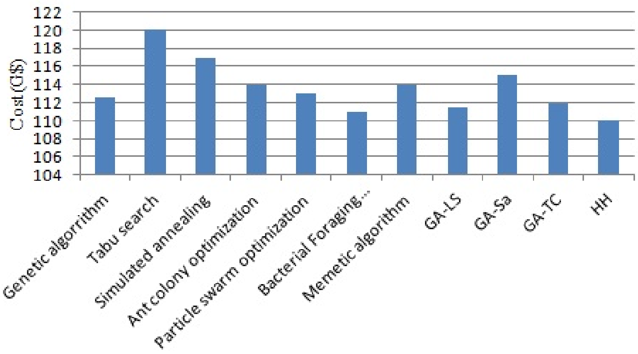

Figure. 2: comparison of heuristics approaches with parameter makesapn

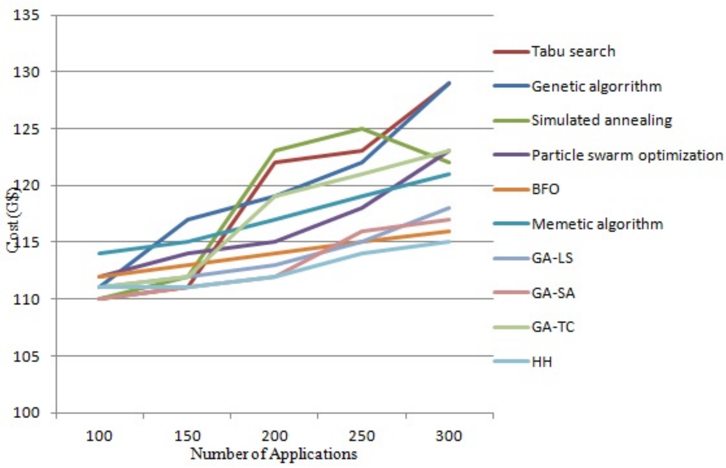

Figure. 3: Cost comparison of heuristics approaches for scheduling 300 jobs on 70 resources

Grid computing has emerged for solving scientific, engineering and large scale problems. It can be concluded that Grid scheduling is one of the main challenging issues of Grid computing. Meta-heuristics are highly adaptive in Grid computing environment but it does not provide good solutions for more number of jobs in heterogeneous environment. Considering all these criteria and simulation results, it is found

Table 1: Comparison of different heuristics approaches

Heuristics/ Features	Parameters	Convergence	Premature Convergence	Services	Local/Global Search	Optimization Problems
Tabu Search	Less parameters	Guaranteed convergence	Prevent premature convergence	conceptual, Simpler, easy to implement, no special memory requirement	Low global search	Combinatorial optimization
Hill climbing	Less functions	No guaranteed	Prevent premature convergence	Simpler and straight forward	Local search	Simple Optimization Problem
Simulated annealing	Less Functions	Converge In asymptotic time	Premature convergence	Easy to code, robust heuristic	Local search	Combinatorial optimization Problems
Genetic Algorithm	More functions	No guaranteed	premature Convergence	No need analytical knowledge, easy to run and implement	Global search capability	Multi objective optimization
Memetic Algorithm	More functions	Guaranteed convergence	Less chance of premature Convergence	Flexible	Global search	Complex objective functions, non-linear multi objective Combinatorial optimization Problems
Ant Colony Optimization	Less functions	Guaranteed convergence	avoid the premature Convergence	Versatile, robust	Global search	Static and dynamic Combinatorial optimization Problems
Particle Swarm Optimization	No function like genetic algorithm	Slow convergence rate	Less chance of premature Convergence	Robust	Global search	Stochastic optimization
Bacterial Foraging Optimization	No function like genetic algorithm	Better Convergence	premature Convergence	Flexible, Robust	Global search	Real world optimization problems

that hyper-heuristic provides a better solution and near optimal solution for Grid scheduling problems.

V. Conclusion

In this paper, various scheduling approaches in Grid computing have been surveyed. A comparison of various parameters like multiple functions, parameters and services has been done. These facts can be used to develop better optimal algorithms. Simulation results show the variation in makespan with respect to the number of jobs using different heuristic methods. Hyper-heuristics provide a better solution and near optimal solution for Grid scheduling problems. Our future work will be based on the above findings to develop a more efficient algorithm for resource scheduling and resource provisioning that will further reduce the makespan.

References

[1] I. Foster and C. Kesselman, ”The Grid: Blueprint for a Future Computing Infrastructure”, Morgan Kaufmann Publishers, USA, 2004.

[2] S. M. S. Bhanu and N. P. Gopalan, ”A Hyper-Heuristic Approach for Efcient Resource Scheduling in Grid”, Int. J. of Computers, Communications Control, Vol. III, No.3, 2008, pp. 249-258.

[3] A. A. Khateeb, R. Abdullah and N. A. Rashid, ”Job type approach for deciding job scheduling in Grid computing systems”, Journal of Computer Science, Vol. 5, No. 10,2009, pp:745-750.

[4] F. Dong, S. G. AKI, ” Scheduling Algorithms for Grid computing: state of the art and open problems”, technical reportno, 2006.

[5] F. Glover and M. Laguna, ”Tabu Search”, Kluwer, Boston, 1997.

[6] F. Xhafa and A. Abraham, ”Computational models and heuristics methods for Grid scheduling problems”, FGCS, vol 26,, 2010, pp:608-621.

[7] S. Kirkpatrick, C. D. Gelatt and P.M. Vechhi, ”Optimization by Simulated Annealing, Science 220, 1983, pp 671-680.

[8] M. D. Theys, T. D. Braun, H. J. Siegal, A. A. Maciejewski and Y. K. Kwok, ”Mapping Tasks onto Distributed Heterogeneous Computing Systems Using a Genetic Algorithm Approach”, chapter 6, John Wiley and Sons, New York, USA, 2001, pp 135-178.

[9] J. H. Holland, ”Adaptation in Natural and ArtificialSystems”,The University of MichiganPress,AnnArbor,1975

[10] W. H. Hsu, ”Genetic Algorithms”, Department of Computing and Information Sciences, Kansas State University,234 Nichols Hall, Manhattan, KS 66506-2302, USA

[11] A. Abraham, R. Buyya, B. Nath, ”Nature’s Heuristics for Scheduling Jobs on Computational Grids”, The 8th IEEE Conference on Advanced Computing and Communications (ADCOM 2000). Cochin, India, 2000.

[12] T. Kokilavani, Dr. D.I. George Amalarethinam, ”Applying Non-Traditional Optimization Techniques to Task Scheduling In Grid Computing An Overview”,International Journal of Research and Reviews in Computer Science(IJRRCS) ,Vol. 1, No. 4, December 2010.

[13] T. D. Braun, J. H. Siegel, N. Beck, L. L. Boloni, M. Maheswaran, I. A. Reuther, P. J. Robertson, M. D.Theys and B. Yao, ”A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems”, Journal of Parallel and Distributed Computing, Vol. 61, No. 6, 2001, pp.810837.

[14] W. E. Hart, N. Krasnogor, J. E. Smith, ”Recent advance ment in Memtic algorithm”, Springer, Heidelberg New York, 2004.

[15] P. E. Merloti, ”Optimization Algorithms Inspired by Biological Ants and Swarm Behavior”, San Diego State University, Artificial Intelligence Technical Report CS550, San Diego, 2004.

[16] M. Dorigo, L. M. Gambardella, ”Ant Colonies for the traveling salesman problem”, BioSystems, 43, 1997, pp 73-81.

[17] M. Dorigo and A. Colorni, ”The Ant System: Optimization by a colony of cooperating agents”, IEEE Transactions on Systems, Man, and CyberneticsPart B, Vol.26, No.1, 1996, pp.1-13.

[18] J. Kennedy and R. Eberhart, ”Particle Swarm Optimization”, Proceedings of IEEE International Conference on Neural Networks. IV.(1995), pp. 19421948.

[19] S. Vo, ”Meta-heuristics:The State of the Art”, A.Nareyek(Ed.):Local Search for Planning and Scheduling,LNAI 2148, Springer-Verlag Berlin Heidelberg 2001, pp.123.

[20] E.-G.TALBI, ”A taxonomy of hybrid heuristics”, Journal of Heuristics,Vol. 8, 2002, Kluwer Academic Publishers.Manufactured in The Netherlands, pp:541564.

[21] E. K. Burke, M.Hyde, G.Kendall, G. Ochoa, E. Ozcan and R.Qu, ”Hyperheuristics: A survey of the State of the Art”, Technical report, University of Nottingham,2009.

[22] K. Chakhlevitch and P. Cowling, ”Hyperheurictics: Recent Developments”, In:Cotta C,Sevaux M, Sorensen K(eds) Adaptive and Multilevel Metaheuristics,Studiesin Studies in Computational Intelligence, vol.136, Springer, 2008, pp3-29.

[23] R. Buyya and M. Murshed, ” GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing”, Concurrency and Computation: Practice and Experience (CCPE), **14**(13-15), ISSN: 1532-0626, Wiley Press, New York, USA, November - December 2002, pp. 1175-1220.

[24] K. M. Passino, ”Biomimicry of bacterial foraging for distributed optimization and control”,IEEE Control Systems Magazine, 2002, pp:52-67.

[25] S. Dasgupta, S. Das, A.Abraham, A.Biswas, ”Adaptive Computational Chemotaxis in Bacterial Forgaing Optimization: An Analysis.IEEE Transon Evolutionary Comp”, vol.13, 2009, pp.919-941.

[26] R. Armenise, C. Birtolo, E. Sangianantoni and L. Troiano, ”Optimizing ATM Cash Management by Genetic Algorithms”,International Journal of Computer Information Systems and Industrial Management Applications, vol.4, pp. 598-608, 2012.

[27] A. I. S. Nascimento and C. J. A. Bastos-Filho, ”Designing Cellular Networks using Particle Swarm Optimization and Genetic Algorithms”, International Journal of Computer Information Systems and Industrial Management Applications, vol.4,pp. 496-505,2012.

[28] A. B. de Carvalho and A. Pozo, ”Using Different Many-Objective Techniques in Particle Swarm Optimization for Many Objective Problems: An Empirical Study”, International Journal of Computer Information Systems and Industrial Management Applications,vol 3,pp. 96107,2011.

Author Biographies

Rajni received her Bachelors degree (mathematics, computer application and economics) from Punjab University, India in 2007, and obtained her Masters degree in mathematics and computing from Thapar University, India in 2009. At present she is a Ph.D. Candidate in the Computer Science and Engineering Department at Thapar University. Her research interests lie in algorithm aspects of resource provisioning, and scheduling in Grid computing and other areas of interest are heuristic algorithms and formal methods. She is a member of ACM.

Inderveer Chana has a Ph.D. in computer science with specialization in Grid computing and an M.E. in software engineering from Thapar University and a B.E. in computer science and engineering. She joined Thapar University in 1997 as Lecturer and has over fourteen years of experience. She is presently working as Associate Professor in the Computer Science and Engineering Department of Thapar University. Her research interests include Grid computing and Cloud computing and other areas of interest are software engineering and software project management. She has more than 50 research publications in reputed journals and conferences. She is currently supervising eight Ph.D. candidates in the area of Grid and Cloud computing. More than 25 Masters theses have been completed so far under her supervision