

A Survey on ANN Based Task Scheduling Strategies in Heterogeneous Distributed Computing Systems

Altaf Hussain, Faisal Azam, Muhammad Sharif, Mussarat Yasmin, Sajjad Mohsin
COMSATS Institute of Information Technology, Wah Cantt Pakistan
e-mail:altaf8031@yahoo.com

Abstract

Heterogeneous Distributed Computing Systems (HeDCS) efficiently utilize the heterogeneity of diverse computational resources which are interlinked through high speed networks for executing a group of computing intensive applications. Directed acyclic graphs (DAGs) are usually used to represent these parallel applications with varied computational requirements and constraints. The optimal scheduling of the given set of precedence constrained tasks to available resources is a core concern in HeDCS and is known to be NP Complete problem. Non deterministic nature of application programs and heterogeneous environment are the main challenges in designing, implementing and analyzing phases of task scheduling techniques. A myriad of heuristic and meta-heuristic approaches have been proposed in the literature to solve this complex problem. The basic purpose of this study is to cover ANN based task scheduling strategies in the distributed computing environment perspective. Further existing scheduling heuristics could be classified in a new state of art classification including the description of frequently used parameters in the mentioned scheduling strategies. The flexible and powerful nature of ANN for identifying the data patterns, underlying time and other constraints and learning capabilities have shown to be a promising candidate among other heuristics.

Key words: artificial neural networks, DAGs, high speed networks, NP Complete problem, scheduling

Introduction

Heterogeneous Distributed Computing System (HeDCS) is a computing platform with diverse set of interconnected processors via high speed network to execute parallel applications. Due to diverse computational resources, task scheduling mechanism in HeDCS has become an efficient tool for attaining reliable, effective and acceptable results. Maximum speedup gain through parallelization can be achieved by harnessing computational heterogeneity among the available processors with high quality schedules. The main objective of scheduling is to assign the tasks to available set of processing elements without violating task precedence requirements with a minimum execution time (make-span) (El Rewini & Abd El Barr 2005, Michael & David 1979)

An application with computational intensive tasks can be executed parallel in the HeDCS. The problem

becomes more complicated due to non deterministic nature of task application model and heterogeneous resource environments. A plethora of heuristics such as clustering algorithms (Palis *et al.* 1996, Topcuoglu *et al.* 2002), list scheduling algorithms (Augonnet *et al.* 2011, Topcuoglu *et al.* 1999), task duplication based algorithms (Hagras & Janeček 2005, Park & Choe, 2001, Ranaweera & Agrawal, 2000), genetic algorithms (Oh&Wu, 2004, Ulusoy 2004), simulated annealing (Braun *et al.* 2001, Kazem *et al.* 2008, Wanneng & Shijue 2006), tabu search (Porto *et al.* 2000, Porto & Ribeiro 1995) and particle swarm optimization (Jarboui *et al.* 2008, Salman *et al.* 2002) have been proposed in literature for the optimal solution of scheduling problem. Static (Shirazi *et al.* 1990) as well as dynamic scheduling (Page & Naughton 2005, Rotithor 1994) schemes are generally employed for the optimal solution. Static schemes augmented with different

techniques (Shin *et al.* 2001) provide an efficient and reliable performance profile on a given architecture.

The current study discusses some intelligent approaches employing different neural networks with various learning techniques for task scheduling problem in the HeDCS. Though the complexity of the problem increases with the application of these trends, the overall gain is promising for large tasks. Moreover, the dynamic and non-deterministic nature (Shi *et al.* 2006) of the scheduling problem demands some intelligent machine learning based scheduling strategies for the complex data patterns in the recent e-science and geo physics applications (Cannataro *et al.* 2004, Glatard *et al.* 2008, MacLaren *et al.* 2004). Thousands of jobs coming dynamically to the scheduler make it imperative to embed intelligent features in it. In the present scenario it is necessary to employ artificial neural network based scheduling strategies to obtain robust and optimal solutions with minimum make span and maximum throughput (Adeli & Karim 1997).

The rest of the paper is organized as follows: In section 2 we define the task scheduling problem,

section 3 discusses various ANN based approaches for task scheduling problem, section 4 provides summary of ANN based techniques; section 5 concludes the paper with some final remarks.

Methodology

The task scheduling problem

The task scheduling problem can be visualized as a consumer-resource model (Ho *et al.* 2002, Wieczorek *et al.* 2009). It consists of a group of consumers represented by parallel tasks and a set of resources, generally the CPU time, to serve the consumers according to certain predefined policies (Muskens&Chaudron, 2004). Managing such resources access through an efficient policy for the consumer satisfaction is the basis of task scheduling problem. The prime consideration is performance optimization on some predefined criteria such as program execution time by the utilization of available resources for various consumers (Buyya 2002). Accordingly, a scheduling system may be considered as a group of consumers, a set of resources and a scheduling policy as illustrated in Figure 1.

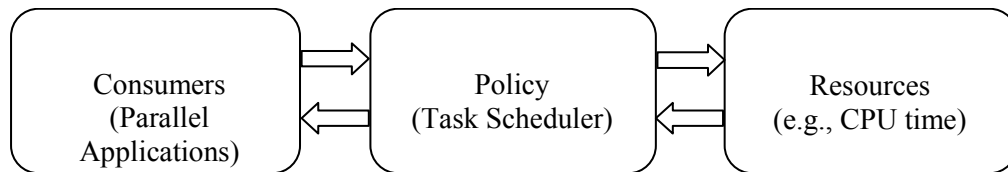


Fig.1. A general scheduling system

The consumer has to submit the constraints of an application such as data dependencies between the sub tasks as well as their timing constraints. Besides, the resource provider has to acknowledge the scheduler's explicit concerns about the available resources such as CPU time, memory, bandwidth etc. Using this information the scheduler can exercise its policy to arrange tasks for executing into the target system in order to meet the desired scheduling requirements.

The scheduling problem has been classified as one of the exigent problems in heterogeneous computing systems (Ilavarasan&Thambidurai, 2007). Design of a scheduling paradigm is based on three primary performance factors:

- Low complexity
- Minimum parallel execution time
- Maximum system efficiency

Conflict may arise in fulfilling all the above mentioned parameters. For example, when the objective is achievement of maximum efficiency with the minimum of parallel execution time, factor number one is given priority over factor number 2 and factor number 3. It is obvious that finding an optimum solution of scheduling problem is difficult and complex in its general form. In recent years, researchers are putting their efforts in designing algorithms that attain maximum efficiency with minimum execution cost and low complexity.

The parallel application model

It is assumed that an application with computationally intensive tasks can be simultaneously executed in heterogeneous computing system. These parallel applications are generally represented by directed acyclic graphs (DAGs) (Mehdiratta & Ghose 1994). The nodes of a DAG represent application tasks whereas inter task dependencies (Dick *et al.* 1998) are represented by the directed edges (Kwok & Ahmad 1997). This model is also largely known as task precedence graph model (Mak & Lundstrom 1990).

A DAG is generally defined by a tuple $T = [N, E, C, W]$, where N is the set of tasks shown by DAG nodes, E is the set of directed edges, C is the set communication cost among the interdependent nodes and W is the set of computation cost associated with each node on the available set of resources. The task and edge weights are assumed to be deterministic (Kafil & Ahmad 1998). The value $c_{ij} \in C$ is the communication cost in transferring data from node n_i to node n_j along the edge $e_{ij} = (n_i, n_j) \in E$. If both the nodes are mapped on the same processor then the communication cost among them is zero. The value $w_i \in W$ is execution cost of the node $n_i \in N$. A task without any predecessor is known as an entry task and a task without a successor is known as an exit task.

The width of a DAG is the maximum number of independent tasks that can be executed parallel. Figure 2 shows a sample DAG with ten nodes.

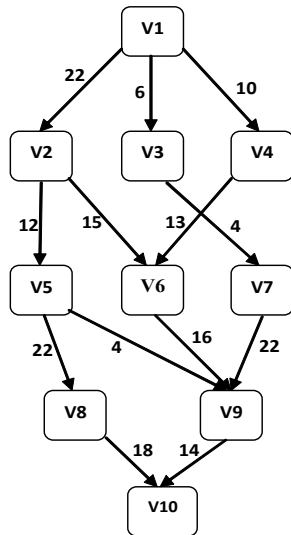


Fig. 2. A DAG with 10 nodes; the edges representing the communication costs

The directed edges exhibit the communication costs among the dependent tasks. The computation costs for three different heterogeneous processors are given in Table 1.

Table 1. Computation cost matrix

Tasks	Processors		
	P1	P2	P3
1	14	8	10
2	14	14	10
3	9	11	10
4	11	19	9
5	10	10	18
6	7	8	5
7	38	19	23
8	11	11	14
9	28	27	24
10	11	13	19

Each DAG generates different levels of parallelism among the tasks. The directed edges describe the dependency constraints for the flow of application program. Each node in a DAG is associated with a task computation cost and each edge is associated with a data communication cost among the nodes. The scheduling cost and performance of a task scheduling algorithm is dependent on these computation and communication overheads among the tasks.

Edges in a DAG start from a parent task and end in one of its child task. A task cannot start execution before all the parents finish their execution and send all messages to the resource assigned to the task. Communication to computation ratio (CCR) of DAG measures its granularity and is defined as ratio between average of all communication costs and average of all computation costs. The longest path from entry node to the exit node of DAG including non-zero communication edge cost and node weights is known as critical path (CP) (Kwok&Ahmad, 1996). The parallel execution time in scheduling a given DAG strongly depends on the critical path of the scheduled DAG.

The heterogeneous distributed computing system model

Heterogeneous distributed computing system model refers the use of different kinds of processors in a single system to enhance the capability and performance of the system. Further the main processor bears different architecture as compared to the rest of processors in the system.

The efficient scheduling of the task graph demands a group of heterogeneous processing elements interconnected with high speed network. These diverse set of machines have different processing capabilities such as speed, memory and bandwidth. It is assumed that this pool of target machines has enough memories and storage spaces; it is independent from external interference. It is also taken as granted that the communication links are uniform and deterministic (B³a_çewicz & Drozdowski 1997). Communication among the machines is through a message passing mechanism (Abdelzaher & Shin 1999). The message transmission takes place simultaneously with equal speed on all outgoing links. The topology of the distributed heterogeneous network may vary from application to application. For n nodes and p processing elements, the computation cost of each processing element for a given task is defined by an $(n \times p)$ matrix (Lo 1988). Furthermore, it is also assumed that all the processing elements are equally capable of executing all the tasks. The data communication time between two machines has two components; data transfer time and data receive time.

Neural network based task scheduling strategies

An artificial neural network (ANN) comprises of a large number of distributed processing elements with parallel information processing among the connected elements. Each processing element has multiple inputs and a single output (Chen&Huang 2001). The information is processed locally and then distributed in the form of signals on the available channels. These networks have the learning capabilities using dynamic weight adjustments which make them suitable for intelligent task scheduling strategies.

A myriad of neural network models have been designed specifically for the heterogeneous task scheduling problem and are embedded in the grid scheduler (Tang *et al.* 2010). Though these models are somewhat complex in nature as compared to the existing heuristics, still they show promising results in a large problem space. Below mentioned are some most popular models adopted for the task scheduling problem.

Fuzzy neural network (FNN)

With the usage of neural networks techniques of approximation, a learning machine named as fuzzy

neural network is developed to find fuzzy system parameters.

Yu *et al.* 2007, Zhou *et al.* 2007 attempted to solve the scheduling problem through the implementation of intelligent fuzzy neural network approach. A Fuzzy Neural Network (FNN) was designed to evaluate the loosely coupled Grid system parameters such as load information, time and initial start up constraints. The FNN tunes itself automatically to the required parameters necessary for finding an optimal solution. The dynamic nature of a large number of factors influencing the system performance in different circumstances makes it imperative to design an intelligent scheduler. The performance of Fuzzy Neural Network scheduler was verified by implementing the so designed model on the well known Globus Toolkit 4. The neuro fuzzy technique has shown more promising results for task scheduling in terms of minimum make span and higher speed-up ratio than the traditional strategies.

Huo *et al.* utilized FNN technique for the selection of Grid resource based on different Quality of Service (QoS) criteria. Their work was only limited to the resource identification and prioritization through intelligent Fuzzy logic selection procedures. The authors proved that the implementation of FNN based logics resulted in better performance and throughput gain.

Hop field neural network (HNN)

Among the different forms of artificial neural networks, Hopfield is a recurrent model. These networks having binary thresholds rely on the systems which carry content-addressable memory. Further the Hopfield networks possess a guarantee of local minimum convergence. These networks are also helpful in better human memory understanding.

The Hop field neural network owing to its potential capability for parallel implementation has also been employed extensively to solve optimization problems such as heterogeneous task scheduling. Though the HNN are usually employed along with some genetic algorithms, nevertheless they have shown promising results in solving NP Complete problem such as task scheduling.

Huang and Chen (1999) employed an energy based Hop field neural network approach for limitations of

HNN models i.e., slow convergence mechanism of HNN, they benefited from the converging features of simulated annealing to get a near optimal solution. A new function called mean field annealing (MFA) was proposed which incorporated the convergence merits of simulated annealing and HNN approaches along with the mean field approximation technique. For the training purpose, two sets of resource and timing constraints and a number of different initial neuron states were applied for the simulations. A comparison was made among already existing HNN techniques and simulated annealing algorithms with the new MFA algorithm. The results indicate that the MFA outperformed the original HNN and simulated annealing strategies for solving complex task scheduling problems.

Wang *et al.* 2008 used the Hop field neural network for solving task scheduling problem in grid environments. A computational mathematical model for energy function calculation was designed for various constraints on sub tasks, task starting time and resource availability so that the conflicts among the tasks and resources may be avoided. The main disadvantage in this scheme is the convergence of energy function to a local optimum solution rather than achieving a global one. In order to avoid such an undesirable convergence, simulated annealing algorithm was applied to the hop field neural network. The main drawback in their paper is the lack of comparison with the existing algorithms. The authors have only demonstrated their work with a simple example.

Chaotic neural network (CNN)

In the field of neural networks, chaotic behavior bears a very important place. Up till now different artificial neural networks have been developed having chaotic dynamic behavior based on the theory of stochastic models. Cao *et al.* (2005) proposed Chaotic Neural Network Algorithm for Task Scheduling (CNNTS). The chaotic neural network model is constructed using complex chaotic dynamics generated by deterministic nonlinear dynamical systems. These networks are similar to the conventional HNN with a slight modification in energy function to control the dynamic behavior of CNN. The chaotic neural network model is constructed by the introduction of an analog sigmoidal function at the output of each neuron and by designing an efficient objective function. The chaotic neural

network is an extension of chaotic neural model proposed by Aihara *et al.* (Aihara *et al.*, 1990). The task agent determines the computational power needed to process a specific task. This predicted or demanded computational power is then searched among the available set of processors. The main objective is to compute the difference between the calculated solution and the requested condition. For large combinatorial problems, this approach performs much better than the conventional searching methods such as tabu search and simulated annealing. The optimal solution is dependent on the minimization of this objective function. The study shows that the use of proposed chaotic neural network approach depicts outstanding results as compared to the conventional techniques such as genetic algorithms and simulated annealing. For small problem space, all algorithms can actually find a 100% optimum solution. But when the problem becomes large, these results show that only CNNTS can find an optimum solution.

Bayesian neural networks (BNN)

An entire distribution of answers is considered in B neural networks rather than a single answer. These networks are helpful in the model selection, model comparison and regularization problems. Further there is no requirement of separate cross validation set.

Jiadong *et al.* (Yang *et al.* 2011) made use of Bayesian optimization algorithm with structural learning capabilities to solve the task scheduling problem. The innate structure of a Bayesian network resembles that of directed acyclic graph, with nodes representing to some known variable such as the computation cost in the final schedule length and the directed arcs represent the inter task communication dependencies. The number of nodes of the network is equal to the number of tasks. The probability assigned to each variable corresponds to the priority of given task to the set of available processor pool. In this way a network structure is built in which each node has a unique association with the corresponding heterogeneous processor in the scheduling problem.

The learning of the Bayesian network for task scheduling is twofold. At the first stage the whole structure of the network is learnt by specifying the communication paths and precedence constraints of the resulting DAG. The quality of the network structure is calculated using some scoring metrics such as

Bayesian Dirichlet (BD) or Bayesian Information criterion (BIC) (Huo *et al.* 2007). In the second stage conditional probability tables (CPTs) are prepared to identify the conditional probabilities of each variable in the DAG. These CPTs are served as the training sets.

In the present study, list scheduling heuristics along with the Bayesian optimization technique have been used. The traditional list scheduling heuristic is employed to make a priority queue of the ready task. After ranking of the corresponding tasks in a non increasing order the execution sequence of each task on the available set of processors is determined. The main comparison metric is the minimization of the final schedule length.

Augmented neural networks (AugNN)

Being highly sensitive in the boundary domain, augmented neurons are added to the output and input layers of neural networks. Accurate input/output mapping is achieved through these neurons in the boundary domain. Further high level of interaction exists between input, hidden and output augmented neurons.

Agarwal *et al.* (2006) proposed the Augmented Neural Network model for solving the task scheduling problem in the grid environments. The computation capabilities of each processing element is calculated and stored in the computation cost matrix. Furthermore, the interdependencies of tasks represented by the DAG edges are stored in the communication cost matrix. These two matrices are given as an input to the AugNN. Some lower bounds such as the make span and processor load are defined at the initial phase of

construction of the network model. The objective is to achieve better solutions with small number of iterations (Agarwal *et al.* 2006). This type of hybrid techniques has been proposed for other heuristics also to find optimal solutions with better convergence. The results indicated that the new formulation proved to be very robust in that the results were not very sensitive to the type of greedy heuristic chosen. The new approach was able to find solutions within the lower bound compared to the greedy AugNN approach. This improved efficiency in achieving better results but not at the cost of additional computational complexity (Jejurikar & Gupta 2006). The AugNN converges to the final solution with minimum number of iterations.

Summary of different ANN based techniques

In the current study we have presented a brief overview of various task scheduling techniques based on artificial neural networks for heterogeneous environments designed for computational intensive applications. Table 2 provides a summary of different ANN based techniques for solving the task scheduling problem. Various parameters such as network model, learning technique used, hybrid nature of the algorithm etc. have been analyzed for various strategies employed. The results indicate that most of the approaches used the hybrid strategies for obtaining more robust and optimal results. Though the list scheduling heuristics have shown promising results for heterogeneous task scheduling but ANN based approaches usually adopted the genetic algorithms for better convergence of the system. Furthermore, the complexity of ANN based approaches makes it more suitable for automated and large scale scheduling problems where the overhead of computation and communication cost functions calculation is hefty.

Table 2. Overview of different ANN based techniques

Neural Network Model	Main function	Hybrid with	Learning Strategy
FNN	Fuzzy Logics	Fuzzy logics	Reinforcement back propagation
HNN	Energy Function	Simulated annealing	Un Supervised
CNN	Objective function	--	Hebbian
BNN	Bayesian function	List based heuristics	Ensemble
AugNN	AugNN function	Greedy algorithms	Breadth first search

Results and Discussion

An application is partitioned into a number of independent tasks to use the available distributed

resources with their underlying parallelism capability. A term called task scheduling is used to explain such a scenario. Tasks refer to application schedulable units

and resources are referred to a number of processors available. In order to get distributed systems having high degree of performance, it is critical to schedule tasks to the processors operating in parallel manner. Neural networks based task scheduling strategies have emerged as a promising candidate for obtaining optimal solutions when dealing with large data sets task graphs. For small data sets the complexity overhead may suppress their intelligent behavior. Though the ANN based classifiers are more robust and accurate in determining the data dependencies and have been used for resource identification, yet their incorporation into the real environments have not been extensively studied. Furthermore, the intelligent behavior of an ANN based technique has not been compared properly with the other heuristics such as list based and task duplication based heuristics.

Future areas for research include further evaluating the ANN proposed algorithms in the real heterogeneous computing environments. An in depth analysis of the above mentioned heuristics using random task graphs and task graphs of some real world application problems such as Gaussian elimination, Fast Fourier transform and molecular dynamics can highlight the importance of ANN based strategies in the heterogeneous distributed task scheduling domain.

References

- Abdelzaher, T. F. and K. G. Shin. 1999. Combined task and message scheduling in distributed real-time systems. *Parallel and Distributed Systems, IEEE Transactions on* **10**: 1179-1191.
- Adeli, H. and A. Karim. 1997. Scheduling/cost optimization and neural dynamics model for construction. *Journal of Construction Engineering and Management* **123**: 450-458.
- Agarwal, A., S. Colak, V. S. Jacob and H. Pirkul. 2006. Heuristics and augmented neural networks for task scheduling with non-identical machines. *European Journal of Operational Research* **175**: 296-317.
- Agarwal, A., V. S. Jacob and H. Pirkul. 2006. An improved augmented neural-network approach for scheduling problems. *INFORMS Journal on Computing* **18**: 119-128.
- Aihara, K., T. Takabe and M. Toyoda. 1990. Chaotic neural networks. *Physics letters A* **144**: 333-340.
- Augonnet, C., S. Thibault, R. Namyst and P. A. Wacrenier. 2011. StarPU: a unified platform for task scheduling on heterogeneous multicore architectures. *Concurrency and Computation: Practice and Experience* **23**: 187-198.
- Bajewicz, J. and M. Drozdowski. 1997. Distributed processing of divisible jobs with communication startup costs. *Discrete Applied Mathematics* **76**: 21-41.
- Braun, T. D., H. J. Siegel, N. Beck, L. L. Bölöni, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, B. Yao and D. Hensgen. 2001. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed computing* **61**: 810-837.
- Buyya, R. 2002. Economic-based distributed resource management and scheduling for grid computing. *arXiv preprint cs/0204048*
- Cannataro, M., A. Congiusta, A. Pugliese, D. Talia and P. Trunfio. 2004. Distributed data mining on grids: services, tools, and applications. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **34**: 2451-2465.
- Cao, H., Z. Yu and Y. Wang. 2005. A Chaotic Neural Network Algorithm for Task Scheduling in Overlay Grid. In: *Semantics, Knowledge and Grid, 2005. SKG'05. First International Conference on*, pp. 38-38.
- Chen, R.-M. and Y.-M. Huang. 2001. Competitive neural network to solve scheduling problems. *Neurocomputing* **37**: 177-196.
- Dick, R. P., D. L. Rhodes and W. Wolf. 1998. TGFF: task graphs for free. In: *Proceedings of the 6th international workshop on Hardware/software codesign*, pp. 97-101.
- El-Rewini, H. and M. Abd El Barr. 2005. Scheduling and task allocation. *Advanced Computer Architecture and Parallel Processing* 235-265.
- Glatard, T., J. Montagnat, D. Lingrand and X. Pennec. 2008. Flexible and efficient workflow deployment of data-intensive applications on grids with moteur. *International Journal of High Performance Computing Applications* **22**: 347-360.
- Hagras, T. and J. Janeček. 2005. A high performance, low complexity algorithm for compile-time task scheduling in heterogeneous systems. *Parallel Computing* **31**: 653-670.
- Ho, D. P., Y. M. Teo and J. P. Gozali. 2002. Solving the N-body problem with the ALiCE Grid System. *Advances in Computing Science—ASIAN 2002. Springer* Pp. 87-97.
- Huang, Y.-M. and R.-M. Chen. 1999. Scheduling multiprocessor job with resource and timing constraints using neural networks. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **29**: 490-502.
- Huo, J., L. Liu, L. Liu, Y. Yang and L. Li. 2007. Selection of the order of autoregressive models for host load prediction in grid. In: *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed*

- Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, pp. 516-521.
- Ilavarasan, E. and P. Thambidurai. 2007. Low complexity performance effective task scheduling algorithm for heterogeneous computing environments. *Journal of Computer Science* **3**: 94.
- Jarboui, B., N. Damak, P. Siarry and A. Rebai. 2008. A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation* **195**: 299-308.
- Jejurikar, R. and R. Gupta. 2006. Energy-aware task scheduling with task synchronization for embedded real-time systems. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* **25**: 1024-1037.
- Kafil, M. and I. Ahmad. 1998. Optimal task assignment in heterogeneous distributed computing systems. *Concurrency, IEEE* **6**: 42-50.
- Kazem, A., A. M. Rahmani and H. H. Aghdam. 2008. A modified simulated annealing algorithm for static task scheduling in grid computing. In: *Computer Science and Information Technology, 2008. ICCSIT'08. International Conference on*, pp. 623-627.
- Kwok, Y.-K. and I. Ahmad. 1996. Dynamic critical-path scheduling: An effective technique for allocating task graphs to multiprocessors. *Parallel and Distributed Systems, IEEE Transactions on* **7**: 506-521.
- Kwok, Y.-K. and I. Ahmad. 1997. Efficient scheduling of arbitrary task graphs to multiprocessors using a parallel genetic algorithm. *Journal of Parallel and Distributed computing* **47**: 58-77.
- Lo, V. M. 1988. Heuristic algorithms for task assignment in distributed systems. *Computers, IEEE Transactions on* **37**: 1384-1397.
- MacLaren, J., R. Sakellariou, K. T. Krishnakumar, J. Garibaldi and D. Ouelhadj. 2004. Towards service level agreement based scheduling on the grid. In: *Workshop on Planning and Scheduling for Web and Grid Services*, pp. 3-7.
- Mak, V. W. and S. F. Lundstrom. 1990. Predicting performance of parallel computations. *Parallel and Distributed Systems, IEEE Transactions on* **1**: 257-270.
- Mehdiratta, N. and K. Ghose. 1994. A bottom-up approach to task scheduling on distributed memory multiprocessors. In: *Parallel Processing, 1994. Vol. 1. ICPP 1994. International Conference on*, pp. 151-154.
- Michael, R. G. and S. J. David. 1979. *Computers and intractability: a guide to the theory of NP-completeness. WH Freeman & Co., San Francisco*
- Muskens, J. and M. Chaudron. 2004. Prediction of run-time resource consumption in multi-task component-based software systems. *Component-Based Software Engineering. Springer* Pp. 162-177.
- Oh, J. and C. Wu. 2004. Genetic-algorithm-based real-time task scheduling with multiple goals. *Journal of systems and software* **71**: 245-258.
- Page, A. J. and T. J. Naughton. 2005. Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing. In: *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, pp. 189a-189a.
- Palis, M. A., J.-C. Liou and D. S. L. Wei. 1996. Task clustering and scheduling for distributed memory parallel architectures. *Parallel and Distributed Systems, IEEE Transactions on* **7**: 46-55.
- Park, C.-I. and T.-Y. Choe. 2001. An optimal scheduling algorithm based on task duplication. In: *Parallel and Distributed Systems, 2001. ICPADS 2001. Proceedings. Eighth International Conference on*, pp. 9-14.
- Porto, S., J. P. F. Kitajima and C. C. Ribeiro. 2000. Performance evaluation of a parallel tabu search task scheduling algorithm. *Parallel Computing* **26**: 73-90.
- Porto, S. C. and C. C. Ribeiro. 1995. A tabu search approach to task scheduling on heterogeneous processors under precedence constraints. *International Journal of high speed computing* **7**: 45-71.
- Ranaweera, S. and D. P. Agrawal. 2000. A task duplication based scheduling algorithm for heterogeneous systems. In: *Parallel and Distributed Processing Symposium, 2000. IPDPS 2000. Proceedings. 14th International*, pp. 445-450.
- Rotithor, H. G. 1994. Taxonomy of dynamic task scheduling schemes in distributed computing systems. *IEE Proceedings-Computers and Digital Techniques* **141**: 1-10.
- Salman, A., I. Ahmad and S. Al-Madani. 2002. Particle swarm optimization for task assignment problem. *Microprocessors and Microsystems* **26**: 363-371.
- Shi, Z., E. Jeannot and J. J. Dongarra. 2006. Robust task scheduling in non-deterministic heterogeneous computing systems. In: *Cluster Computing, 2006 IEEE International Conference on*, pp. 1-10.
- Shin, D., J. Kim and S. Lee. 2001. Intra-task voltage scheduling for low-energy, hard real-time applications. *IEEE Design & Test of Computers* **18**: 20-30.
- Shirazi, B., M. Wang and G. Pathak. 1990. Analysis and evaluation of heuristic methods for static task scheduling. *Journal of Parallel and Distributed computing* **10**: 222-232.
- Tang, X., K. Li, G. Liao and R. Li. 2010. List scheduling with duplication for heterogeneous computing systems. *Journal of Parallel and Distributed computing* **70**: 323-329.
- Topcuoglu, H., S. Hariri and M.-Y. Wu. 1999. Task scheduling algorithms for heterogeneous processors. In: *Heterogeneous Computing Workshop, 1999.(HCW'99) Proceedings. Eighth*, pp. 3-14.

- Topcuoglu, H., S. Hariri and M.-y. Wu. 2002. Performance-effective and low-complexity task scheduling for heterogeneous computing. *Parallel and Distributed Systems, IEEE Transactions on* **13**: 260-274.
- Ulusoy, G. 2004. Multiprocessor task scheduling in multistage hybrid flow-shops: a genetic algorithm approach. *Journal of the Operational Research Society* **55**: 504-512.
- Wang, C., H. Wang and F. Sun. 2008. Hopfield neural network approach for task scheduling in a grid environment. In: *Computer Science and Software Engineering, 2008 International Conference on*, pp. 811-814.
- Wanneng, S. and Z. Shijue. 2006. A parallel genetic simulated annealing hybrid algorithm for task scheduling. *Wuhan University Journal of Natural Sciences* **11**: 1378-1382.
- Wieczorek, M., A. Hoheisel and R. Prodan. 2009. Towards a general model of the multi-criteria workflow scheduling on the grid. *Future Generation Computer Systems* **25**: 237-256.
- Yang, J., H. Xu, L. Pan, P. Jia, F. Long and M. Jie. 2011. Task scheduling using Bayesian optimization algorithm for heterogeneous computing environments. *Applied Soft Computing* **11**: 3297-3310.
- Yu, K.-M., Z.-J. Luo, C.-H. Chou, C.-K. Chen and J. Zhou. 2007. A fuzzy neural network based scheduling algorithm for job assignment on Computational Grids. *Network-Based Information Systems. Springer* Pp. 533-542.
- Zhou, J., K.-M. Yu, C.-H. Chou, L.-A. Yang and Z.-J. Luo. 2007. A dynamic resource broker and fuzzy logic based scheduling algorithm in Grid environment. *Adaptive and natural computing algorithms. Springer* Pp. 604-613.

