

HYBRID GENETIC ALGORITHM FOR BI-CRITERIA MULTIPROCESSOR TASK SCHEDULING WITH COMMUNICATION DELAY

Sunita Dhingra^{1*}, Ashwani K. Dhingra², Satinder Bal Gupta³ and Ranjit Biswas⁴

¹Department of Computer science & Engineering, ²Department of Mechanical Engineering,

University Institute of Engineering & Technology,

Maharshi Dayanand University Rohtak-124001 Haryana, India

³Department of Computer Science, Vaish College of Engineering Rohtak-124001 Haryana, India

⁴Department of Computer Science & Engineering, Jamia Hamdard University New Delhi-110062, India

ABSTRACT

Present work considers the minimization of the bi-criteria function including weighted sum of makespan and total completion time for a Multiprocessor task scheduling problem. Genetic algorithm is the most appealing choice for the different NP hard problems including multiprocessor task scheduling. Performance of genetic algorithm depends on the quality of initial solution as good initial solution provides the better results. Different list scheduling heuristics based hybrid genetic algorithms (HGAs) have been proposed and developed for the problem. Computational analysis with the help of defined performance index has been conducted on the standard task scheduling problems for evaluating the performance of the proposed HGAs. The analysis shows that the ETF-GA is quite efficient and best among the other heuristic based hybrid genetic algorithms in terms of solution quality especially for large and complex problems.

Keywords:

Multiprocessor task scheduling, heuristics, Hybrid Genetic algorithm, makespan, Total Completion time.

1. INTRODUCTION

Parallel processing is the most promising approach for meeting the increased computational requirements that has introduced a number of problems including multiprocessor task scheduling. Usually a large program is divided into smaller tasks having some dependencies representing the precedence constraints such that a task cannot be started until all its predecessors have finished. The goal of a task scheduling algorithm is to schedule all the tasks on the given number of available processors without violating precedence constraints so as to minimize the different performance measures in order to maximize the throughput and utilization of the system. The multiprocessor task scheduling problem with precedence constraints belongs to the class of NP hard problems and its importance led to several comparative studies. The conventional research on scheduling considers heuristic based algorithms which are divided into three categories: list scheduling, clustering based and duplication based. Among heuristic algorithms, List scheduling algorithms are commonly used and work by assigning priority to each task according to attributes such as t-level (top level), b-level (bottom level), static level (sl) and ALAP (As Late As Possible)

start time[1-5,10].The task with highest priority is assigned to the processor which gives the earliest start time. Task selection can be random or based on some rules for the task having the same priority. The heuristic algorithms follow the procedure that narrow the search to a very small portion of the solution space. Heuristic-based approaches have greedy nature due to which they are not likely to produce reliable results for a wide range of problems, particularly when the density of the scheduling problem increases.

Therefore, researchers go for metaheuristics such as Genetic Algorithm, Simulated annealing, Tabu search, Particle swarm optimization etc. for the sake of better quality schedules. Most of the metaheuristics especially Genetic Algorithm (GA) outperformed traditional heuristic based scheduling algorithms at the cost of extra time and computing effort [6]. It is therefore hybridization of metaheuristics with heuristics is the next choice for improving the solution quality.

In the present work, an attempt has been made for hybridization of different list scheduling heuristics with genetic algorithm for the multiprocessor task scheduling problem with precedence constraints on homogeneous processors for minimizing the considered bi-criteria objective function. The remaining part of the paper is organized as follows: section 2 is based on literature review in the field of hybrid algorithms related to task scheduling. Problem formulation along with assumptions is presented in section 3. Section 4 describes the different hybrid Genetic Algorithm (HGAs). Section 5 gives analysis of experimental results with discussions followed by the conclusion in section 6.

2. RELATED WORK

As the complexity of the scheduling problem increases the heuristic algorithms fail to provide the reliable results. Metaheuristic based scheduling algorithms obtain schedules of better quality but at the expense of more computing efforts due to limited exploration ability. Hou et al. [6] proposed the first and most important work that has used GA for multiprocessor tasks scheduling which uses the height of tasks in input DAG. Though the algorithm is very simple in terms of computational complexity, but it cannot guarantee that the search space is global due to which some viable schedules are not accessible [11]. Hou et al. [6] and Dhodhi & Ahmad [7] developed a new technique in combination with the ISH [2] and DSH [2] based list scheduling heuristics with GAs and called as problem-space genetic algorithms (PSGAs). The analysis of results showed that the GA when combined with heuristics could work efficiently and schedule the tasks on several processors. In literature, different methods have also been attempted to combine the heuristic and genetic approaches for the solution of the problem [7, 8, 23].

Sivanandam et al. [13] proposed a hybrid algorithm in support of particle swarm optimization/simulated annealing (PSO/SA) for static allocation of tasks in a heterogeneous distributed computing system with the objective of minimizing the cost. Implementation has been carried out on different PSO algorithm with Simulated Annealing. Different experiments have been performed on the benchmark problems and shows that the proposed hybrid method was effective and efficient in finding near optimal solutions. Yoo et al. [14] proposed a multi-objective hybrid genetic algorithm (MOHGA) for real-time tasks on heterogeneous multiprocessor environment with the purpose of minimizing the total tardiness and completion time simultaneously. The adaptive weight approach has been used for multiple objectives. The convergence of GA is enhanced by introducing the probability of SA as the measure for taking

new trial solution. The effectiveness of proposed algorithm has been checked by various experiments and it was concluded that the MOHGA provides improved results as compared to other algorithms without communication cost. Dahal et al. [15] proposed a hybrid algorithm for dynamic scheduling. The well known heuristics such as 'Earliest Deadline First (EDF)' and 'Shortest Computation Time First (SCTF)' has been hybrid with the Genetic Algorithm (GA). It was concluded that hybridization of GA with SCTF provides better performance as compared to the EDF based hybrid GA. Azghadi et al. [16] developed an immune genetic approach for multiprocessor task scheduling problem and proves to be more effective. Jouglet et al. [17] proposed a memetic algorithm (MA) for the hybrid flow shop scheduling with multiprocessor tasks. They hybridized the Genetic Algorithm (GA) and Constraint Programming (CP) for a Memetic Algorithm (MA) and concluded the superiority of proposed MA. Hwang et al. [18] proposed priority-based GA which uses a new encoding mechanism with a multi-functional chromosome that uses the priority representation— so-called priority-based multi-chromosome (PMC). They addressed the problem of multiprocessor task scheduling with communication cost. From results, it is concluded that proposed priority-based GA has effective performance in various parallel environments for scheduling methods.

Kim et al. [19] considered heuristic method for a deterministic scheduling problem where multiple jobs with s-precedence relations are processed on multiple identical parallel machines for minimization of the total completion time. Goh et al. [21] considered a heterogeneous multiprocessor scheduling problem with precedence constraint and proposed a hybrid evolutionary algorithm (HEA) for makespan minimization. The method considered the partial list scheduling and duplication scheduling heuristic for exploiting the intrinsic structure of the solution and specialized genetic operators for promoting the exploration of the search space. Experiments were carried on a set of benchmark problems and it was concluded that the proposed HEA provide better results. Wen et al. [22] incorporated GA with both Variable Neighborhood Search (VNS) and a heuristic extracted from traditional list scheduling algorithms for the minimization of makespan in the heterogeneous multiprocessor scheduling problem resulting into a heuristic based hybrid genetic variable neighborhood search algorithm. The performance of proposed approach was compared with four related algorithms, HEFT, AIS, VNS and IGA on standard benchmarks problems and it was concluded that proposed algorithm constantly outperforms the other four algorithms in terms of schedule quality.

Mohamed et al. [23] proposed the Modified List Scheduling Heuristic (MLSH) along with the hybridization with the Genetic Algorithm for multiprocessor task scheduling system and concluded the superiority when compared to others. Roy et al. [24] considered task scheduling in multiprocessor systems and proposed a heuristic based Genetic Algorithm by choosing the eligible processor on educated guess. Variation of HLFET algorithm with genetic algorithm was proposed and experiments were performed on Standard Task Graphs (STG). It was concluded that the algorithm has better average makespan than Elitism stepping method in lesser number of evaluations.

Hybrid methods of scheduling have been motivating by the fact that each type of scheduling technique has its own supremacy and restriction. Therefore, the present work considers the development of different Hybrid Genetic Algorithms (HGAs) with commonly used list heuristics for improving the quality of solution in the presence of communication cost.

3. PROBLEM STATEMENT

The considered work includes the minimizing the weighted sum of makespan and total completion time for multiprocessor task scheduling problem. Various assumptions and the fitness function considered are illustrated below:

3.1 ASSUMPTIONS

- The problem is deterministic with known values of data communication time, task dependencies & execution time.
- A DAG is used to represent the dependencies along with execution time and communication cost.
- Communication cost is considered only when the tasks are scheduled on different processor otherwise it is taken as zero.
- All the processors are homogeneous. i.e same execution time for all the processors.
- Pre-emption of tasks is not allowed.
- Task duplication is not allowed.
- All processors & tasks are accessible at time $t = 0$.

3.2 FITNESS FUNCTION

Fitness function considered in the present work deals with minimizing weighted sum of makespan and total completion time. Makespan of a schedule is the time at which its last task completes. Total completion time of a schedule is the summation of completion times of all the tasks of that schedule. For the requirement of maximum utilization of resources, increase in throughput, load balancing etc, authors have proposed the bi-criteria decision making fitness function including the two performance measure for multiprocessor task scheduling which has been framed as:

$$F = Min \left[\alpha C_{max} + (1 - \alpha) \sum_{i=1}^n C_i \right]$$

Where, F is the bi-criteria fitness function, C_{max} is the makespan and C_i is the completion time of i_{th} task of a schedule with ' α ' the weight coefficient as per the priority of the performance measures to be minimised.

4. PROPOSED HYBRID GENETIC ALGORITHM (HGA)

It has been generally established that finding optimality to NP hard problems is not a feasible option since large amount of computational time is needed for finding of such solutions. In reality, a superior initial solution can be obtained by a heuristic in a rational computational time. A Genetic algorithm is a valuable population based approach for the multiprocessor task scheduling. Consequently many researchers [6, 12, 19] have reported success with genetic algorithm in achieving good solutions to combinatorial optimization problems. A genetic algorithm starts with an initial population which can be generated arbitrarily or based on some rules, heuristics and algorithms. Then in each generation the population follows the sequence of encoding, fitness evaluation, selection, crossover and mutation until some stopping criteria is met.

Performance of GA depends on the superior chromosomes in initial population for faster convergence [24]. So, in the present work different list scheduling heuristics (table 1) is used to obtain a seed sequence which is then combined with (ps-1) randomly generated sequences to give initial population of size ps. The proposed hybrid genetic algorithm is described below:

Step 1 Encoding

- a) Encoding give the illustration of a chromosome. In the present work, chromosome is taken as (T, P) pair where T is task sequence t_1, t_2, \dots, t_n & P is owed processor sequence p_1, p_2, \dots, p_n .
- b) Each task sequence is a variation of task numbers & each processor sequence is a variation of processor numbers (1, 2... m) with length equal to number of tasks.

Step 2 Initialization

Each task sequence is a variation of task numbers, so each task will be processed according to its emergence. Each task in the sequence should come out before all of its children and parents due to dependency. Some mechanism is needed for validating the invalid sequences. The steps for generation of initial population:

1. Use scheduling heuristic (table 1) for generating the seed sequence.
2. Generate (ps – 1) population arbitrarily using the following steps
 - a) Generate the suitable task sequences (TS) of (Ps-1) using the algorithm as stated by Bonyadi and Moghaddam[20].
 - b) Generate the processor sequences (PS) of ps-1 arbitrarily.
 - c) Each task sequence (T) is mapped from TS for arbitrarily selecting the processor sequence (P) giving each chromosome in the form (T, P) pairs such that the task sequence should be followed by mapped processor sequence.
3. Merge the seed sequence with randomly generated population as per population size (Ps).

Step 3 Reproduction

As the task and processor sequences have different nature, so, different reproduction operators are used for both sequences. Both the sequences are firstly separated from a chromosome and then used independently for performing crossover and mutation. Task sequences after reproduction may not follow dependency, so a mechanism stated by Bonyadi and Moghaddam [20] has been applied for validating the task sequences. The valid task sequences after reproduction (TS') have mapped to the processor sequences based on minimum fitness value.

The new offspring's are generated with the following steps:

- a) Scores every member of the present population by computing the fitness function included weighted sum of makespan and total completion time.
- b) The individuals as per fitness in the current population act as elite and admit in the next population.
- c) Selects parents for reproduction based on the fitness value as per the selection function.

Step 4: Stopping criteria

The algorithm stops when maximum number of generations reaches 100.

4.1 List Scheduling heuristics

List scheduling heuristics are based on assigning priorities to tasks and then allocating the highest priority task to processor which gives the minimum start time. The present work uses five list scheduling heuristics i.e. HLFET, ISH, MCP, ETF, DLS to input one best sequence in initial population of GA resulting into five different hybrid genetic algorithms HLFET-GA, ISH-GA, MCP-GA, ETF-GA and DLS-GA. Different heuristics used for the initial seed sequence in the different hybrid genetic algorithmic are shown in table 1.

Table 1. Different Heuristics for initial feasible solution of different Hybrid GAs

S. No.	Heuristic	Description
1.	HLFET[1]	HLFET decides the priorities of tasks according to static level and allocates processor to highest priority task based on minimum start time.
2.	ISH [2]	ISH is similar to HLFET except it attempts to fill the idle time slots created in partial schedule by scheduling other nodes into them.
3.	MCP[4]	The MCP algorithm uses the ALAP of a node as scheduling priority and uses the ALAP of child nodes to break the ties. It allocates the processor to highest priority process based on minimum start time using insertion approach.
4.	ETF[3]	ETF algorithm computes the earliest start time of all the ready nodes by examining the start time of node on all processors exhaustively and selects the one with smallest start time.
5.	DLS[5]	DLS works by computing the DL (difference between static level of a node and its earliest start time) at each scheduling step for every node in the ready pool on all processors. The node-processor pair resulting into largest value of DL is selected for the scheduling.

5. RESULTS AND DISCUSSION

The proposed different Hybrid Genetic algorithms (HGA's) have been implemented in MATLAB environment. First, a sample multiprocessor task scheduling problem (T18) has been considered to set the parameters of the proposed GA for the bi-criteria multiprocessor task scheduling problem. The different parameters setting of GA have been optimised for a sample problem using design of experiments. Parameters fixed for different proposed and developed HGAs are shown in table 2.

The value of total completion time is very large as compared to makespan and values of weight coefficient (α) must be normalized in such a way that minimization of fitness values have negligible impact on the makespan and total completion time for equal priority. The normalized values eliminate the effects of certain gross influences.

Table 2. Parameters fixed for different HGAs.

Parameters	Values
Population size	75
Elite Count	2
Crossover fraction	0.5
Selection function	Tournament
Number of generation	100
Crossover function(Task)	Position based
Mutation function(Task)	Swap
Crossover function (processor)	One point crossover
Mutation function (Processor)	uniform mutation

There are different methods of normalization and every method requires the best value of the objectives. Therefore, normalization has been done with the help of different experiments from the heuristics on the different problems and a value of weight coefficient (α) for a particular problem are calculated from the relation (irrespective of number of processors) and is given below:

$$\text{Weight Coefficient } (\alpha) = \left[1 - \frac{\text{Makespan}_{\text{Best}}}{\text{TotalCompletiontime}_{\text{best}} + \text{Makespan}_{\text{Best}}} \right]$$

The comparative analysis has been done by computing the performance Index (PI) as:-

$$\text{Performance Index (PI) } (\%) = \left[1 - \frac{\text{Algorithm}_{\text{solution}} - \text{Best}_{\text{solution}}}{\text{Best}_{\text{solution}}} \right] \times 100$$

Algorithm_{solution} is the average solution obtained by the different algorithms and Best_{solution} is the best solution obtained from different algorithm in all the runs. Performance Index closer to 100% provides the good results. Some of the standard problems [12] along with their computed weight coefficient (α) have been used for the comparative analysis among proposed HGA's as shown in table 3.

Table 3. Standard Multiprocessor Task Scheduling Problems along with weight coefficient

Problems	No. of Tasks	No. of processors considered	Communication cost	Reference	Remarks	Weight Coefficient
T9	9	2,3,4	Variable	Bonyadi and Moghaddam [20]	-----	0.82
T14_1	14	2,3,4	Fixed(20)	Tsuchiya et al. [9]	LU decomposition	0.89
T14_2	14	2,3,4	Fixed(80)	Tsuchiya et al. [9]	LU decomposition	0.89
T16_1	16	2,3,4	Fixed(40)	Wu and Gajski [4]	Laplace	0.90
T16_2	16	2,3,4	Fixed(160)	Wu and Gajski [4]	Laplace	0.90
T18	18	2,3,4	Variable	Bonyadi and Moghaddam [20]	Gaussian Elimination	0.91

All the Hybrid GAs have been run five times for taking final average and comparative analysis for makespan and total completion time criteria has been shown in fig. 1 and 2 respectively. The problem T9 and T18 having the variable communication cost and the problems T14_1, T14_2, T16_1 & T16_2 have the fixed communication cost. It can also be seen that the communication cost has great impact on the quality of solution. For the variable communication cost, ISH-GA and

ETF-GA provides the best results for the problem T9 and as the task size increase i.e. T18, the ETF-GA provides the superior results as comparison to others with all the processors for the makespan and total completion time.

For the problem having fixed communication cost, all the HGA's are comparable to each others for all the processors having lesser communication cost. As the communication cost increases along with task size, all the HGA's (Except HLFET-GA) provides the similar results. Therefore, in order to compare the different HGAs, Average Performance Index for all the problems has been calculated for 2, 3 and 4 processors and shown in table 4.

Average performance index shows the ETF-GA algorithm provides better results for all the problems considered with average PI of 95.55% and 95.57% for 2 processors, 98.58% and 98.09% for 3 processors and 99.47% and 98.44% for 4 processors for minimization of makespan and total completion time respectively. Therefore, ETF based HGA when compared to others provides the best compromise results for the makespan and total completion time criteria for the larger size multiprocessor task scheduling problems with 2, 3 and 4 processors.

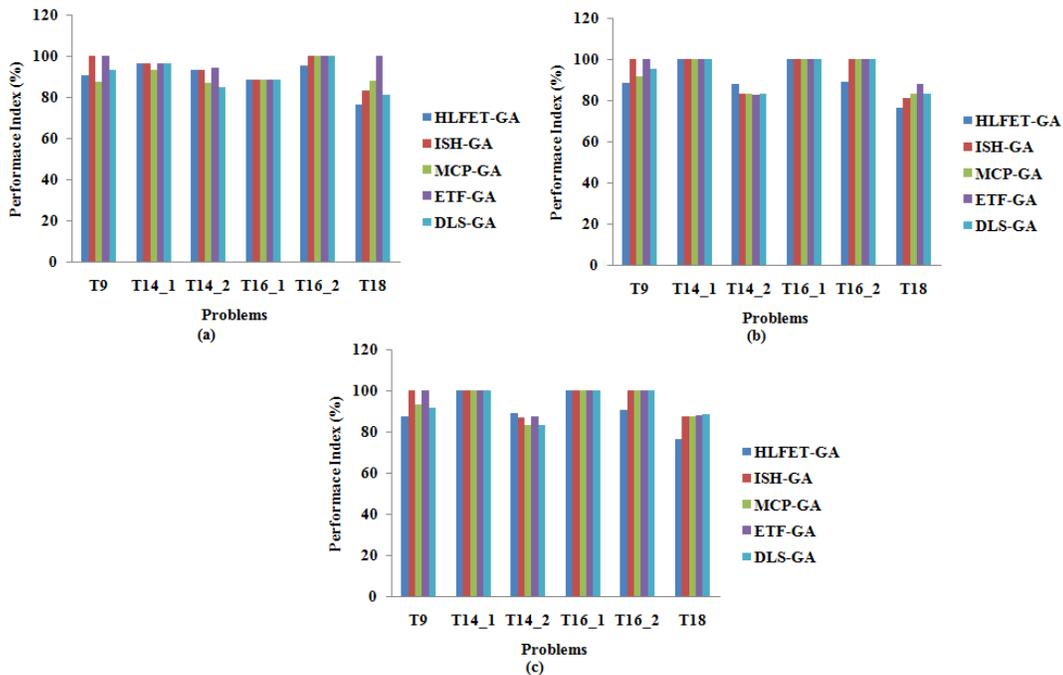


Figure 1. Performance Index (%) of different Hybrid GA for makespan (a) Two Processors (b) Three Processors (c) Four Processors

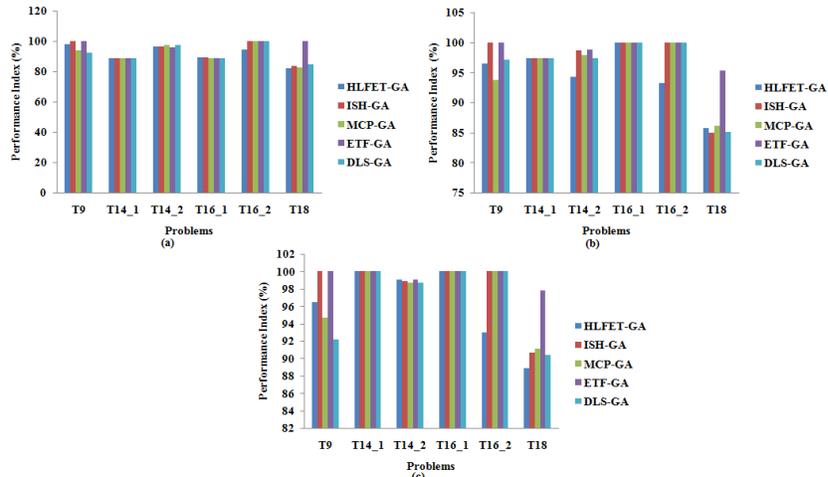


Figure 2. Performance Index (%) of different Hybrid GA for Total Completion Time (a) Two Processors (b) Three Processors (c) Four Processors

Table 4. Average Performance Index (%) for different HGAs on different processors

No. of (→) Processors	2		3		4	
	MS	TCT	MS	TCT	MS	TCT
HLFET-GA	91.39	90.81	94.53	95.34	96.23	96.66
ISH-GA	92.96	92.39	96.84	96.04	98.26	97.66
MCP-GA	91.89	91.66	95.86	95.99	97.42	97.34
ETF-GA	95.55	95.57	98.58	98.09	99.47	98.44
DLS-GA	74.76	91.97	96.15	96.89	96.88	97.25

MS: Makespan & TCT: Total Completion Time

6. CONCLUSIONS

Present work considers the bi-criteria multiprocessor task scheduling problem on homogeneous processors with objective of minimizing the weighted sum of makespan and total completion time. Initial solution from the well known list heuristics have been obtained and combined with initial population of genetic algorithm to form five different Hybrid Genetic Algorithms (HGA's) for the problem. The computation analysis has been done on the some standard benchmark multiprocessor task scheduling problem with variable and fixed communication cost. Performance analysis is done with the help of defined performance index and reveals that ETF based hybrid GA (ETF-GA) provides finer results for the makespan and total completion time criteria particularly for larger and complex problems.

REFERENCES

- [1] Adam, T.L., Candy, K.M. and Dickson, J., "A Comparison of list scheduling for parallel processing system", *Communication ACM*, Vol. 17, No. 12, pp. 685-690, 1974.
- [2] Kruatrachue, B. and Lewis, T.G., "Duplication Scheduling Heuristic, a New Precedence Task Scheduler for Parallel Systems", Technical Report, 87-60-3, Oregon State University, 1987.
- [3] Hwang, J.J., Chow, Y.C., Anger, F. D. and Lee, C.Y., "Scheduling precedence graphs in systems with interprocessor communication times", *SIAM Journal of Computing*, Vol. 18, No.2, pp. 244-257, 1989.
- [4] Wu, M.Y., and Gajski, D.D., "Hypertool A programming aid for message-passing systems", *IEEE Transactions on Parallel and Distributing System*, Vol. 1, No. 3, pp. 330-343, 1990.
- [5] Sih, G. C. and Lee, E. A., "A compile-time scheduling heuristic for interconnection- constrained heterogeneous processor architectures", *IEEE Transactions on Parallel and Distributing System*, Vol. 4, No. 2, pp.75-87, 1993.
- [6] Hou, E.S.H., Ansari, N. and Hong, R. "A Genetic Algorithm for Multiprocessor Scheduling", *IEEE Transactions on Parallel and Distributed Systems*. Vol. 5, No. 2, pp. 113 - 120, 1994.
- [7] Dhodhi, M.K., and Ahmad, I., " A multiprocessor scheduling scheme using problem-space genetic algorithms", *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 214-219, 1995.
- [8] Ahmad, I., and Dhodhi, M.K., "Multiprocessor scheduling in a genetic paradigm", *Parallel computing*, Vol. 22, pp. 395-406, 1996.
- [9] Tsuchiya, T., Osada, T., and Kikuno, T., "Genetics-based multiprocessor scheduling using task duplication", *Journal of Microprocessors and Microsystems*, Vol. 22, Vol. 3-4, pp. 197-207, 1998.
- [10] Kwok, Y.K., and Ahmad, I., " Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors", *ACM computing surveys*, Vol. 31, No. 4, pp. 406-471, 1999.
- [11] Zhong, Y.W., and Yang, J.G., "A genetic algorithm for tasks scheduling in parallel multiprocessor systems", *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, pp.1785-1790, 2003.
- [12] Wu, A.S., Yu, H., Jin,S., Lin, K.C and Schiavone,G., "An incremental genetic algorithm approach to multiprocessor scheduling", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 9, pp. 824-834, 2004.
- [13] Sivanandam, S.N., Visalakshi, P., and Bhuvanawari, A., "Multiprocessor Scheduling Using Hybrid Particle Swarm Optimization with Dynamically Varying Inertia", *International Journal of Computer Science & Applications*, Vol. 4, No.3, pp 95-106, 2007.
- [14] Yoo,M., and Gen, M., "Scheduling algorithm for real-time tasks using multiobjective hybrid genetic algorithm in heterogeneous multiprocessors system", *Computers and Operations Research*, Vol. 34, pp. 3084 - 3098, 2007.
- [15] Dahal, K., Hossain, A. , Varghese, B. , and Abraham, A., " Scheduling in Multiprocessor System Using Genetic Algorithms" *Proceedings Of 7th Conference on Computer Information Systems and Industrial Management Applications(CISIM-2008)*, pp. 281 - 286, 2008.
- [16] Azghadi M. R., Bonyadi, M.R., Hashemi, S. and Moghadam, M.E, "A Hybrid Multiprocessor Task Scheduling Method Based on Immune Genetic Algorithm", *Proceedings of 5th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness Hong Kong, Hong Kong , July 28 - 31, 2008.*
- [17] Jouglet, C. O'guz, and M. Sevaux, "Hybrid Flow-Shop: a Memetic Algorithm Using Constraint-Based Scheduling for Efficient Search", *Journal of mathematical modeling and algorithms*, Vol. 8, pp. 271 -292, 2008.
- [18] Hwang, R., Gen, M. and Katayam, H., "A comparison of multiprocessor task scheduling algorithms with communication costs," *The Journal of Computers & Operations Research*, vol. 35, pp. 976 - 993, 2008.
- [19] Kim, E.S., Sung, C.S. and Lee, I. S., "Scheduling of parallel machines to minimize total completion time subject to s-precedence constraints," *The Journal of Computers & Operations Research*, Vol. 36, pp. 698 - 710,2009.

- [20] Bonyadi, M.R and Moghaddam, M. E, “A bipartite genetic algorithm for multi-processor task scheduling” , International Journal of Parallel Programming, Vol. 37, No. 5, pp. 462- 487, 2009.
- [21] Goh, C. K. , Teoh, E. J., and Tan, K. C., “A hybrid evolutionary approach for heterogeneous multiprocessor scheduling”, Soft Computing, Vol.13, pp. 833–846, 2009.
- [22] Wen,Y., Xu, H., and Yang, J., “A heuristic-based hybrid genetic-variable neighborhood search algorithm for task scheduling in heterogeneous multiprocessor system”, Information Sciences Vol. 181, pp. 567–581, 2011.
- [23] Mohamed, M. R., and Awadalla, M. H. A., “Hybrid Algorithm for Multiprocessor Task Scheduling”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, 2011
- [24] Roy, P., Alam, M.M., and Das, N., “Heuristic Based Task Scheduling in Multiprocessor Systems With Genetic Algorithm By Choosing The Eligible Processor”, International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.4, 2012.