MULTIPROCESSOR SCHEDULING AND PERFORMANCE EVALUATION USING ELITIST NON DOMINATED SORTING GENETIC ALGORITHM FOR INDEPENDENT TASK

D.Rajeswari[1] and V.Jawahar SenthilKumar[2]

[1]Department of Information Technology, R.M.K Engineering College,Thiruvallur,India
[2]Department of ECE, Anna University, Chennai, India

ABSTRACT

Task scheduling plays an important part in the improvement of parallel and distributed systems. The problem of task scheduling has been shown to be NP hard. The time consuming is more to solve the problem in deterministic techniques. There are algorithms developed to schedule tasks for distributed environment, which focus on single objective. The problem becomes more complex, while considering bi-objective. This paper presents bi-objective independent task scheduling algorithm using elitist Non-dominated sorting genetic algorithm (NSGA-II) to minimize the makespan and flowtime. This algorithm generates pareto global optimal solutions for this bi-objective task scheduling problem. NSGA-II is implemented by using the set of benchmark instances. The experimental result shows NSGA-II generates efficient optimal schedules.

KEYWORD

Task scheduling, bi-objective, pareto-optimal solution, NSGA-II , crowding distance, independent task.

1.INTRODUCTION

Parallel and distributed systems have collection of computing machines connected to meet different complex computational requirements. As the computing machines have different capacity, the time taken to complete the tasks varies on different processors. [1,2]. Proper scheduling of tasks helps to utilize the entire power of the different computing capacity machines.

Scheduling techniques are classified as static and dynamic [2]. Static scheduling is useful for different computing capacity machines like high power computational grids and clouds [3]. To utilize the entire power of computing machines properly, the parameters like throughput, communication cost, reliability cost, response time and network traffic should be optimized [4]. The makespan and flowtime minimizations are considered here. Whenever more than one parameters (which is contradicted with each other) are considered, this requires multi-objective formulation. Two different approaches for multi-objective optimization problems (MOOP) are there. First approach considers the multi-objective into single objective by using weight function. Next approach is used to find the set of pareto optimal solutions and it is preferable for real time applications [5].

Genetic algorithm(GA) is one of the best evolutionary algorithm for scheduling tasks to different computational machines [6]. In GA, the time taken to find the optimal solution is more, due to the challenge in characteristics of parent population by mutation. In this paper NSGA-II is used ,

DOI:10.5121/ijcsa.2015.5505

which minimize the time taken using elitism concept. New population is developed by adding few best solutions from current population. Crowding distance calculation is used to keep the pareto front size within the limit chosen. This optimization method is applied to minimize the makespan and flowtime.

The rest of the paper is structured as follows. Section 2 discussed about the related task scheduling algorithms. Section 3 defines the environment for task scheduling using NSGA-II. The procedure of NSGA-II is discussed in section 4. Section 5 reports the simulated results and section 6 presents the conclusion and future work.

2.RELATED WORK

A number of algorithms are developed to schedule independent tasks in different computing capacity machines. Munir et al used min-max algorithm to schedule independent tasks in heterogeneous algorithm [7]. Freund et al used min-min and max-min algorithm to develop the scheduler [8]. Abraham et al schedule the independent jobs to grid systems by using LJFR-SJFR [9]. Scheduling of task using simulated annealing by Daniil et al [10], ant colony optimization by Chun-Yan Liu et al [11], particle swarm optimization[6] by Salman et al and genetic algorithm by Page et al are more famous[12].

There are lot of works are carried out in the genetic algorithm itself. Scheduling non-pre emptive tasks using GA was proposed by Mitra et al [13]. Oh et al schedules the non- pre emptive tasks by using multi-objective GA[14]. Instead of single objective , multi-objective problems are considered here. Static scheduling using genetic algorithm was proposed by Theys et al[15]. The above papers considered single objective problem and converting multi-objective to a single objective problem. This project considers the multi-objective optimization technique to schedule the task using NSGA-II.

3. PROBLEM FORMULATION

Parallel and distributed computing environment have collection of nodes. Let P be the collection of y processors and T be the collection of x tasks, which are to be assigned to y processors in parallel and distributed computing environment. Consider the tasks are independent in nature and a task can be assigned to a processor alone. It is also consider that the tasks are non- pre emptive. The capacities of computing resources, computational need of each task are estimated at first. Using all the details expected time to compute (ETC) matrix can be constructed. ETC[x][y], implies the time taken to complete task x in the processor y. Each row of the of ETC matrix indicates the time taken to complete a particular task in all y processors and each column indicates the time taken to complete all the x tasks in a particular processor. Table 1 shows the example for ETC [4][3] matrix.

	P1	P2	P3
Task 1	3	2	4
Task 2	5	4	2
Task 3	2	7	3
Task 4	4	5	6

Table 1. An example of ETC matrix

The objective is formulated to reduce makespan and mean flowtime. The time taken to complete the last task is makespan . Mean flowtime is the flowtime divided by number of systems, where the flowtime is the aggregation of completion time of all the tasks.

$$\text{makespan} = \min_{s_j\in\, sched}\{\max_{t\in tasks} F_t\} \qquad (1)$$

$$\text{flo}wtime = \min_{s_j\in sched}\{\textstyle\sum_{t\in tasks} F_t\} \qquad (2)$$

mean flowtime = flowtime/ number of systems (3)

4.Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II)

At first Goldberg [16] proposed the pareto-based objective assignment. It is used for assigning equal probability of reproduction to all non-dominated individuals in the population. NSGA-II has a fast procedure for non-dominated sorting with complexity O (kμ2) [18]. The steps are as follows:

4.1.Algorithm

Step 1: Initialize the population of size N randomly (Pt)
Step 2: Generate the offspring of size N (Qt) using selection, crossover and mutation.
Step 3: Combine parent and offspring of size 2N (Rt = Pt+ Qt).
Step 4: Find and sorting the fronts of Rt using Non-dominated sorting.
Step 5: Compare the individuals and break the tie using crowding distance to generate new individuals of size N.
Step 6: Stop if reach the stopping criteria, else go to step 2.

4.2.Non-dominated Sorting

Non-dominated sorting is to find the solution for the next generation after combining both parent and offspring. The procedure for non-dominated sorting is given below :
Step a: Take two solution x and y in population N.
Step b: If x and y are not equal compare x and y for all the objectives.
Step c: For any objective of x , y is dominated by x, mark solution y is dominated. Unmarked solution forms the first non-dominated front.
Step d: Repeat the procedure till the entire population is divided into fronts.

4.3.Crowding distance

To break the tie between the individuals, which are in the same front crowding distance is used.

Step a: Find the number of individuals (x) in the front (Fa).
Step b: set the distance di = 0, i = 1,2 ,…,x.
Step c: Sort the individuals (x) in front (Fa), depends on the objective function (Oj), j = 1,2,…,m. m is the number of objectives and S = sort (x,Oj).
step d: Set the distance of boundary individuals S(di)j = ∞ and S(dx)j = ∞.
Step e: Calculate S(d2)j to S(dx-1)j to individuals remains, l=2 to x-1.

$$\mathrm{S(d_l)j} \;=\; \mathrm{S(d_l)} + \frac{\mathrm{S(l+1)f_j - S(l-1)f_j}}{\mathrm{f_j^{max} - f_j^{min}}} \qquad (4)$$

$\mathrm{S(l)f_j}$ is the lth individual value in S for jth objective function.

4.4.Selection

Crowded tournament selection operator is used. An individual x wins the tournament with aother individual Y, if any of the following is true.

a) An individual x has better rank i.e., rankx < ranky .

b) Whenever the individual x and y have same rank (rankx = ranky), then the individual x has the better crowding distance (in less crowded area) than individual y (i.e., rankx = ranky and dx = dy).

4.5.Genetic Operators

To generate the new population from the existing one crossover and mutation operators are used. In this paper, fitness based crossover and swap mutation are used.

5.RESULTS AND DISCUSSIONS:

5.1.Simulation Results

The simulation model in [2] is based on expected time to compute (ETC) matrix for 512 tasks and 16 processors. The instances of the benchmark are classified into 12 different types of ETC matrices according to the three following metrics: job heterogeneity, machine heterogeneity, and consistency. In ETC matrix, task heterogeneity is defined as the amount of variance possible among the execution times of the tasks. Machine heterogeneity, represents the possible variation of the running time of a particular task across all the machines, both of which can be classified into low or high heterogeneity.

Also an ETC matrix is said to be consistent, if machine rx has a lower execution time than machine ry for job jk , then the same is true for any job ji . The ETC matrices that are not consistent are inconsistent ETC matrices. A semi consistent ETC matrix is characterized by an inconsistent matrix which has a consistent sub-matrix of a predefined size.All instances consist of 512 tasks and 16 processors and are labeled u- m-nn-oo that represents the following

u means uniform distribution (used in generating the matrix).
m means the type of inconsistency (c–consistent, i–inconsistent and s means semi-consistent).
nn indicates the heterogeneity of the jobs (hi–high, and lo–low).
oo indicates the heterogeneity of the resources (hi–high, and lo–low).
The following values for various parameters of GA yielded the best result on average, which is used in obtaining the results on all 12 static instances.

Population size	: 100
No of generations	: 200
Selection operator	: Crowded tournament selection
Cross-over operator	: Fitness based crossover
Mutation operator	: Swap mutation
Cross-over probability	: 0.8
Mutation probability	: 0.01
Weights w1	: 0.6
Weights w2	: 0.4

NSGA-II task scheduling algorithm was simulated for the above parameters settings. In the same setting the algorithm runs 5 times and pareto optimal solutions were calculated for all the consistency level with different task and machine heterogeneity, which are plotted in Fig 1 – 4.

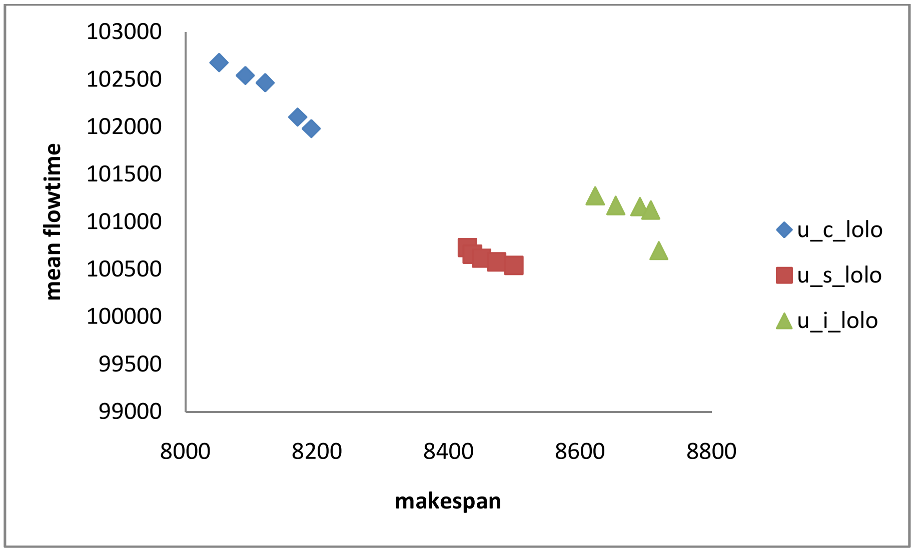

Fig 1. Pareto optimal solution for low task, low machine heterogeneity

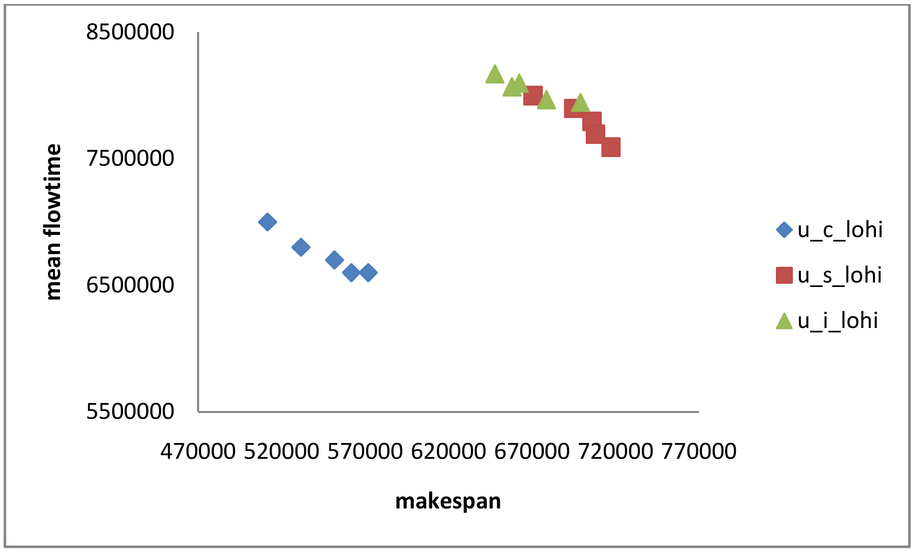

Fig 2. Pareto optimal solution for low task, high machine heterogeneity

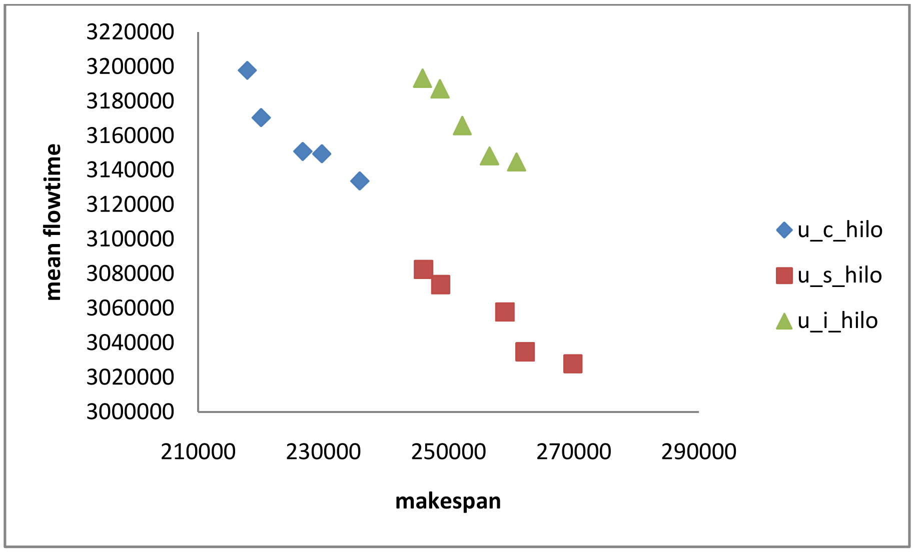

Fig 3. Pareto optimal solution for high task, low machine heterogeneity

Fig.1 represents for consistent with low job and resource heterogeneity values are deviated much from other two consistent types. For low job and high resource heterogeneity, high job and resource heterogeneity consistent matrix produces much better result than other two consistent types, which are present in Fig.2 and Fig.4. Fig.3 shows for high job and low resource heterogeneity, all consistent type gives the results, which are not that much deviated from each other.

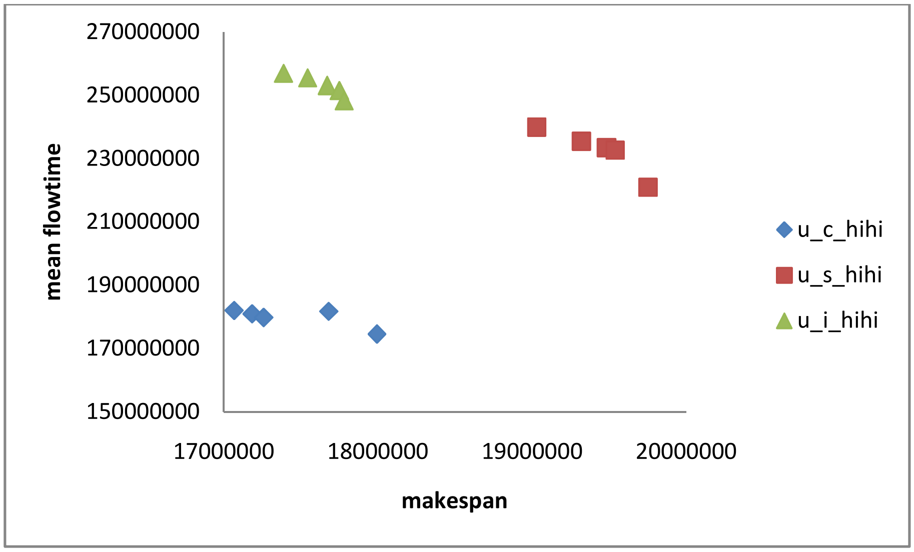

Fig 4. Pareto optimal solution for high task, high machine heterogeneity

Fig. 5 and 6 presents the non-dominated individuals in the first and second front. It shows that, the number of iterations are increased, the individual which occupy the first front too increased. It also shows that best schedules are obtained while the number of iterations is increased.

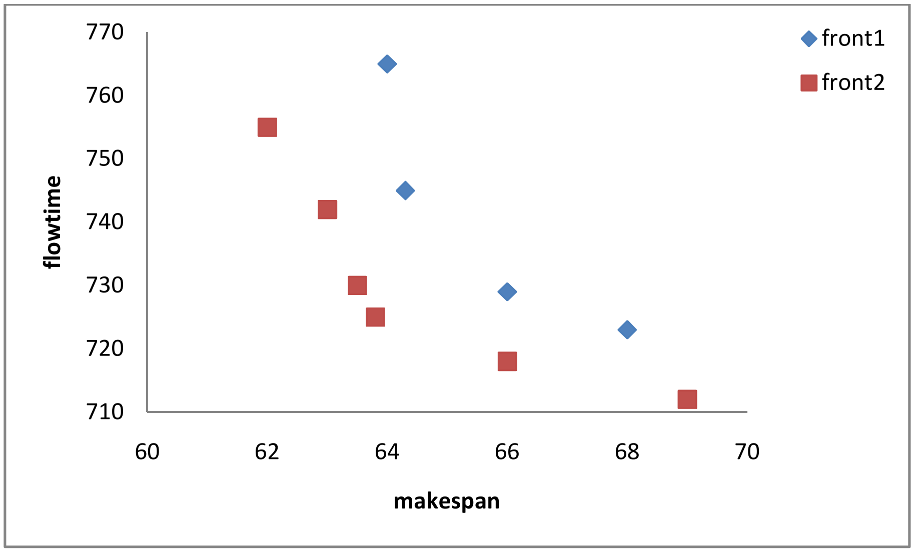

Fig 5. Pareto-optimal solutions after 100th iteration

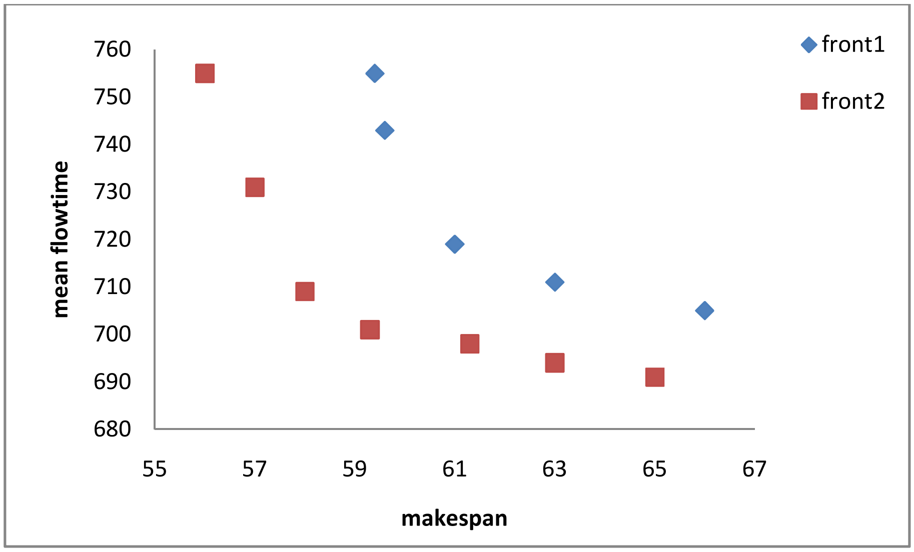

Fig 6. Pareto-optimal solutions after 200th iteration

Table.2 represents the makespan and mean flowtime values of best pareto optimal solution in 200 iteration.

Instance	Makespan	Mean flowtime
u_c_lolo	8050.74389	101982.560124
u_c_lohi	511428.38160	6599823.38623
u_c_hilo	217831.42096	3133868.49732
u_c_hihi	17067563.71978	174494967.32
u_s_lolo	8428.57225	100541.96233
u_s_lohi	670508.94835	7591225.32972
u_s_hilo	245978.15837	3027893.05322
u_s_hihi	19029310.59428	220939142.23143
u_i_lolo	8623.09586	100697.14745
u_i_lohi	647623.85438	7944532.7942
u_i_hilo	245823.45505	3144851.29854
u_i_hihi	17386751.81584	248287933.39277

Table.2 Makespan and mean flowtime for best pareto optimal solution

5.2.Performance metrics

5.2.1.Error Rate

It implies the percentage of obtained non dominated solutions, which are not the members of global pareto optimal set [17].

$$\mathrm{ErrR} = \frac{\sum_{x=1}^{m} e_x}{m} \qquad (5)$$

x is the number of solutions in the obtained non dominated solution set. e_x=0 if the solution x is the member of global pareto optimal set, otherwise e_x=1.

5.2.2.Generational Distance

It finds how far obtained non dominated solutions from global pareto optimal solutions [17].

$$\mathrm{G.Dist} = \frac{\sqrt{\sum_{x=1}^{m} d_x^2}}{m} \qquad (6)$$

d_x is the Euclidean distance between each of the obtained non dominated solution set and the nearest solution of global pareto optimal set.

5.2.3.Spacing :

It calculates the relative distance measure between the consecutive solutions in the obtained non dominated solution set [18].

$$\mathrm{Spac} = \sqrt{\frac{1}{m}\sum_{x=1}^{m}(d_x - \bar{d})^2} \qquad (7)$$

d_x is the minimum distance from the xth solution to all other solution in the obtained non dominated solution set.

5.2.4.Spread

It finds the spread of obtained non dominated solutions [18].

$$S = \frac{\sum_{n=1}^{ob} d_n^e + \sum_{x=1}^{m} |d_x - \bar{d}|}{\sum_{n=1}^{ob} d_n^e + m\,\bar{d}} \qquad (8)$$

n is the number of objective function. d_x is the distance between neighboring solutions. $\bar{d}$ is the mean of these distance value. d^e is the distance between the extreme solution of global pareto optimal set and obtained non dominated solutions for nth objective.

The values of performance metrics for all the 12 instances are indicated in table.3. Inconsistent with high job and resource heterogeneity has less Error Rate value. Low Generational Distance occurs for c_lo_hi. Semi consistent with low job and resource heterogeneity gives low Spacing. Minimum Spread value is produced by semi consistent with low job and high resource heterogeneity.

Instance	Error Rate (ErrR)	Generational distance (G.Dist)	Spacing (Spac)	Spread (S)
u_c_lolo	0.25	0.22	0.43	0.16
u_c_lohi	0.29	**0.19**	0.52	0.18
u_c_hilo	0.32	0.24	0.38	0.22
u_c_hihi	0.29	0.29	0.42	0.13
u_s_lolo	0.19	0.38	0.53	0.13
u_s_lohi	0.26	0.23	0.46	**0.1**
u_s_hilo	0.25	0.3	0.49	0.15
u_s_hihi	0.30	0.26	**0.3**	0.28
u_i_lolo	0.27	**0.19**	0.37	0.15
u_i_lohi	0.28	0.26	0.40	0.17
u_i_hilo	0.28	0.22	0.33	0.13
u_i_hihi	**0.18**	0.2	0.44	0.16

Table 3.Performance metrics

6.CONCLUSION

This paper presents the task scheduling using NSGA-II for multi-objective problem with performance evaluation. NSGA-II is used to find the pareto-optimal solution, which minimize the makespan and flowtime to independent tasks in static environment. The simulated result shows that NSGA-II minimizes the makespan and mean flowtime, if the number of iterations is increased. The number of individuals in the first front is increased in the higher iterations. NSGA-II can also be implemented for dependent tasks and dynamic environment.

7. REFERENCES

[1] S.Ali, T.D.Braun, H.J.Siegel, A.A. Maciejewski, “Heterogeneous computing”, Encyclopedia of Distributed Computing, Kluwer Academic, 2001.

[2] Braun T.D, sigel H J, Beck N, Boloni L L, Maheswaran M, Reuther A I, Roberston J P , Theys M D and Yao B, “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(6), 2001, pp 810- 837.

[3] Foster I and Kesselman C ,”The grid : Blueprint for a new computing infrastructure”, 2nd ed, New York:Morgan Kaufman, 2003.

[4] Liu H, Abraham A, Hassanien A, “ Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm”, Future Generation Computer systems 26(8), 2010, pp 1336-1343.

[5] Chankong V, Haimes Y, “Multiobjective decision making theory and methodology”, Mineola, NY:Dover Publications, 2008.

[6] Salman A, Ahmad I, Al-Madani S, “ Particle swarm optimization for task assignment problem”, Microprocessor and Microsystems 26(8), 2002, pp 363-371.

[7] Munir E U, Li J-Z, Shi S-F, Rasool Q, “Performance analysis of task scheduling heuristics in grid:, In Proc. of the International Conference on Machine Learning and Cybernetics, 6, 2007, pp 3093-3098.

[8] R.F.Freund, M.Gherrity, S.Ambrosius, M.Campbell, M.Halderman, D.HEnsgen, E.Keith, T.Kidd, M.Kussow, J.D.Lima, F.Mirabile, L.Moore, B.Rust, and H.J.Siegel,” Scheduling resources in multiuser, heterogeneous, computing environments with SmartNet”, 7th IEEE Heterogeneous Computing Workshop (HCW’98), 1998, pp 184-199.

[9] Abraham, R.Buyya, B.Nath, “Nature’s heuristic for scheduling tasks on computational grids”, 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000), 2000.

[10] Daniil A.Zorin, Valery A.Kostenko,” Simulated Annealing algorithm for Job Shop Scheduling on Reliable Real-Time Systems”, Communications in Computer and Information Science, 2015,pp31-46.

[11] Chun-Yan Liu, Cheng-MingZou, Pei Wu, “ A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization in Cloud Computing”, In. 13th International Symposium on Distributed Computing and Applications to Business, engineering and Science(DCABES), 2014.

[12] Page A, Naughton J, “Framework for task scheduling in heterogeneous distributed computing using genetic algorithm”, Artificial Intelligence Rev.24:415-429, 2005.

[13] Mitra H, Ramanathan P.A, “genetic approach for scheduling non-preemptive tasks with precedence and deadline constraints”, in Proc. of the 26th Hawaii international conference on system science, 1993, pp.556-564.

[14] Oh J, Wu C, “Genetic algorithm based real-time task scheduling with multiple goals”. Journal of Systems and Software, 71(3), 2004, pp 245-58.

[15] Theys MD, Braun TD, Siegel HJ, Maciejewski AA, Kwok YK, “ Mapping tasks onto distributed heterogeneous computing systems using a genetic algorithm approach”, in Zomaya AY, Ercal F, Olariu S, editors, “Solution to parallel and distributed computing problems”,New York: Wiley, 2001, pp 135-78.

[16] Tan K C, Lee T H, Khor E F, “Evolutionary Algorithms for Multi-objective Optimization : Performance Assessments and Comparisions”, Artificial Intelligence Rev:17,2002,pp.251-290.

[17] Sarker, R , Coello, C. ,” Assessment Methodologies for Multiobjective Evolutionary Algorithms In Evolutionary Optimization” , R. Sarker, M. Mohammadian and X. Yao (edited), Kluwer Academic Publishers, Boston,2002, pp.177-195.

[18] Deb K,”Multi-objective optimization using evolutionary algorithms”, Singapore:John Wiley & Sons Ltd, 2001.

Authors

D.Rajeswari is an assistant professor at the Department of Information Technology, RMK Engineering College, Chennai. She received a master degree from PSG College of Technology, Coimbatore in 2010 and pursuing Ph.D. from Anna University. Her current research interests include evolutionary a lgorithm, distributed systems, cloud computing and Big Data.

Dr.V. Jawahar Senthilkumar is an Associate professor at the Department of Electronics and Communication Engineering, College of Engineering- Guindy, Anna University Chennai. He received the Bachelor of Engineering Degree in Electrical and Electronics Engineering from Hindustan College of Engineering & Technology, Madras University, Chennai .He did his post graduation in Applied Electronics from Bharatiyar University, Coimbatore and Ph.D Degree in Electronics and Communication Engineering from Anna University Chennai. He has contributed around 40 technical papers and in various j ournals & conferences. His main research interests are in the field of parallel & distributed algorithms, VLSI design, Network design and management and scientific computing.