International Journal of Computational Intelligence Systems, Vol. 7, Supplement 1 (2014), 58-67

Workload balancing in identical parallel machine scheduling using a mathematical programming method

Yassine Ouazene, Farouk Yalaoui, Hicham Chehade and Alice Yalaoui

Institut Charles Delaunay - Laboratoire d'Optimisation des Systèmes Industriels
UMR-STMR-CNRS 6279
Université de Technologie de Troyes
12 rue Marie Curie,CS 42060
10004, Troyes, France
E-mail: {yassine.ouazene, farouk.yalaoui, hicham.chehade, alice.yalaoui}@utt.fr

Received 15 December 2012
Accepted 20 July 2013

Abstract

This paper addresses the workload balancing problem in identical parallel machines context. The problem consists of assigning *n* different jobs to *m* identical parallel machines in order to minimize the workload imbalance among the different machines. This problem is formulated as a linear mixed integer program to minimize the difference between the greatest and smallest workload assigned to each machine. Based on some numerical examples reported in the literature, we establish that the classical formulation which consists of minimizing the greatest machine completion time does not provide the optimal workload repartition. That is why we consider a new mathematical formulation based on the minimization of the difference between the workload of the bottleneck machine and the workload of the fastest machine. The proposed programming method is also used to provide optimal solutions in reasonable computational times for different test problems presented in the literature by Raghavendra and Murthy [10] to test their genetic algorithm.

Keywords: Parallel machine, workload balancing, scheduling, mathematical programming.

1. Introduction

Parallel manufacturing structure is one of the solutions of increasing the processing capacity of a manufacturing system. Generally, in parallel machine scheduling two decisions are taken at once. The first one is to determine the assignment of jobs to the machines and the second one is sequencing of the jobs assigned to each machine in order to optimize different objective functions. One of these objective functions is the workload imbalance minimization which can be defined as assigning available workload among different machines as equally as possible. Workload balancing is important for both service and manufacturing industries. In service industry, human resources should have a balanced workload in order to be equitable and provide a quality service. The aim of a manager is to assign jobs to each worker in such a way that their workloads are as similar as possible. In manufacturing industry, balancing the workload among the machines is important to reduce the idle times and work-in-process [8]. It helps also to remove bottlenecks in manufacturing systems [13].

Rajakumar et al. [13] addressed the workload balancing problem using different priority rules such as: random, shortest processing times and longest processing times. The authors used the relative difference of imbalance to evaluate the performances of these different strategies. In their next publication [14], the authors proposed a genetic algorithm that outperforms these different strategies. Based on these classical priority rules, Raghavendra and Murthy [9] made an effort to reduce the imbalance in random type of parallel machines addressing the loading problem in flexible manufacturing system. Later, Raghavendra et al. [12] proposed a genetic algorithm based approach with SPT and LPT rules for reducing the imbalance between the parallel machines. The authors have concluded that their genetic algorithm provides better solutions than the strategies proposed in the first paper [9]. Raghavendra et al. [11] applied this genetic algorithm in a case study for ten different part styles with different batch quantity on two vertical machining centers. The same genetic based heuristics algorithm was compared with other approximate approaches proposed in the literature. Indeed, Raghavendra et al. [10] presented a comparative study among different test examples to illustrate the efficiency of their algorithm. The authors have shown that their algorithm outperforms the different heuristics proposed by Heinrich [2] and the genetic and simulated annealing algorithms proposed by Liu and Wu [5]. Caragiannis [1] presented an improved upper bound for the greedy algorithm to minimize the L_p norm of the machine loads for the problem of scheduling permanent jobs on unrelated machines. As mentioned above, workload balancing problem is present in different environments. For example Moon et al. [7] considered this problem with several operators in semi-automatic parallel machine shop with two types of machines. There objective was to assign the jobs to the machines and allocate the machines to operators in such way to minimize the operators workload imbalance under the constraint of machine availability and operator times. The authors have proposed a non linear mathematical formulation of this problem.

In the literature, the workload balancing problem has been associated with different scheduling criteria in different ways, even by considering the workload imbalance as a constraint or as an objective. As examples, Ouazene et al. [8] addressed the identical parallel machine scheduling problem to minimize simultaneously total tardiness and workload imbalance. The authors proposed a mathematical formulation and a genetic algorithm based on the aggregation of the two objective functions. Yildirim et al. [15] have studied the scheduling of semi-related machines with sequence dependent setups and load balancing constraints. The authors proposed a mathematical formulation of the problem and an approximate resolution based on some heuristics and a genetic algorithm. Recently, Keskinturk et al. [4] presented a non linear mathematical model for a parallel machine problem with sequence-dependent setups with the objective of minimizing the total relative imbalance. The authors proposed two metaheuristic methods for the approached resolution of the problem. The two metaheuristics consist of an ant colony optimization algorithm and a genetic algorithm. Based on various randomly generated tests, the authors have concluded that the ant colony algorithm outperforms both heuristics and genetic algorithm.

The remainder of this paper is organized as follows: section 2 details the mathematical formulation of the problem considered. The different notations and decision variables are described. Also, the classical list-scheduling algorithm usually used in workload balancing on parallel machines problem is explained. Section 3 presents the comparative study based on different test examples reported in the literature. Finally, section 4 summarizes the contribution of this paper and gives some perspectives for future possible extensions.

2. Problem formulation

The problem considered in this paper can be formally described as follows: a set of N independent jobs $\{J_1, J_2, ..., J_N\}$ are scheduled on M identical parallel machines. We assume that each job J_j has a

deterministic processing time p_j. The jobs may be assigned to any one of the machines. A machine can process only one job at once and no preemption is allowed. All jobs are available at time zero.
Based on these assumptions and the following notations, the problem is formulated as a mixed integer mathematical model in order to minimize the workload imbalance amon the machines.

2.1. Notations

N	Total number of jobs
M	Total number of machines
$i, j \in \{0,1,...,N\}$	Job index where job 0 is a fictitious one which is always sequenced at the first position on a machine
m	Machine index
p_j	Processing time of job j
C_m	Completion time of machine m
S_m	Set of jobs scheduled on the machine m
$C_m = \sum_{j \in S_m} p_j$	Completion time of machine m

2.2. Analysis of the existing formulations

The first performance measure or criterion used to deal with workload imbalance minimization problem is the total relative percentage of workloads imbalances (*RPI*) introduced by [13,14]. This performance measure is also called percentage of deviation from upper bound and defined as:

$$RPI = \frac{1}{M} \times \sum_{m=1}^{M} \frac{C_{max} - C_m}{C_{max}}.$$

In the case of identical parallel machines, this criterion depends solely of the maximum completion time criterion which is not optimal in workload balancing problem optimization. So, minimizing the relative percentage of imbalances criterion is the same thing as minimizing the maximum of completion times.

Remark 1. In the case of identical parallel machine, the relative percentage of imbalances *RPI* depends solely of the maximum of completion times C_{max}.

Considering the definition of *RPI* criterion, we have:

$$RPI = \frac{1}{M} \times \sum_{m=1}^{M} \frac{C_{max} - C_m}{C_{max}}$$

$$= \frac{1}{M \times C_{max}} \times \sum_{m=1}^{M} (C_{max} - C_m)$$

$$= \frac{\sum_{m=1}^{M} C_{max}}{M \times C_{max}} - \frac{\sum_{m=1}^{M} C_m}{M \times C_{max}}$$

$$= 1 - \frac{\sum_{m=1}^{M} C_m}{M} \times \frac{1}{C_{max}}$$

Known that,

$\frac{\sum_{m=1}^{M} C_m}{M} = \frac{\sum_{j=1}^{N} p_j}{M} = \mu$, μ is a constant.
So, the relative percentage of imbalances *RPI* can be written as a function of C_{max} as follows.

$$RPI = 1 - \frac{\beta}{C_{max}}$$

The second criterion called normalized sum of Square for workload deviations (*NSSWD*) has been introduced by Ho et al. [3]. This criterion is based on the sum of squares principle known in measuring variability in statistics and serves as a precise measurement criterion. *NSSWD* is defined as follows.
$NSSWD = \frac{\sqrt{\sum_{m=1}^{M} (C_m - \mu)^2}}{\mu}$
such as:
$\mu = \frac{\sum_{m=1}^{M} C_m}{M} = \frac{\sum_{j=1}^{N} p_j}{M}$
For each machine m the square error is given by $(C_m - \mu)^2$. It is easy to establish that the sum of square for workloads deviations $\sum_{m=1}^{M} (C_m - \mu)^2$ depends directly on the sum of square completion times.

$$\sum_{m=1}^{M} (C_m - \mu)^2 = \sum_{m=1}^{M} (C_m^2 - 2 \times \mu \times C_m + \mu^2)$$

$$= \sum_{m=1}^{M} C_m^2 - 2 \times \sum_{m=1}^{M} C_m + \sum_{m=1}^{M} \mu^2$$

By considering that $\sum_{m=1}^{M} C_m = M \times \mu$, we obtain:

$$\sum_{m=1}^{M} (C_m - \mu)^2 = \sum_{m=1}^{M} C_m^2 - 2 \times M \times \mu^2 + M \times \mu^2$$

So, the sum of square for workloads deviations can be written as follows:

$$\sum_{m=1}^{M} (C_m - \mu)^2 = \sum_{m=1}^{M} C_m^2 - M \times \mu^2$$

Since $M \times \mu^2$ is a constant, then minimizing the sum of square for workloads deviations is the same as minimizing the sum of squares of completion times C_m^2.

2.3. *A new mixed integer linear programming model*

The mathematical formulations based on the minimization of the maximum completion time C_{max} do not provide the optimal utilization of machines as mentioned in Rajakumar et al. [14]. In fact, minimizing the maximum workload is not sufficient to obtain the optimal repartition of the workload. This will be illustrated by some counters examples presented in computational experiments section. Therefore, the optimal formulation should considers simultaneously the maximum and the minimum workloads. So, workload balancing problem should be defined as the minimization of the difference between the maximum and the minimum workloads. In other words, it is defined as minimizing the difference between the workload of the bottleneck machine and the workload of the fastest machine. Based on this definition, we propose the following mathematical model.

2.3.1. *Decision variables*

The mathematical model detailed below includes both assignment and precedence variables. Generally assignment variables are sufficient for computing machine workloads. The precedence variables are used here in order to determine the schedule or the sequence of the jobs on each machine.

$$x_{ijm} = \begin{cases} 1, & \text{if job } j \text{ immediately follows job } i \text{ in a sequence on machine } m \\ 0, & \text{otherwise} \end{cases}$$

$$y_{jm} = \begin{cases} 1, & \text{if job } j \text{ is assigned to machine } m \\ 0, & \text{otherwise} \end{cases}$$

$$C_m = \sum_{j \in S_m} p_j = \sum_{j=1}^{N} (p_j \times y_{jm}), m = 1...M$$

$$C_{max} = max_{1 \leqslant m \leqslant M} \{C_m\}$$

$$C_{min} = min_{1 \leqslant m \leqslant M} \{C_m\}$$

2.3.2. *Mathematical formulation*

$$\min Z = C_{max} - C_{min} \tag{1}$$

$$\sum_{j=1}^{N} x_{0jm} \leqslant 1, m = 1...M \tag{2}$$

$$\sum_{i=0, i \neq j}^{N} \sum_{m=1}^{M} x_{ijm} = 1, j = 1...N \tag{3}$$

$$\sum_{j=1, j \neq i}^{N} x_{ijm} \leqslant y_{jm}, i = 1...N, m = 1...M \tag{4}$$

$$\sum_{i=0, i \neq j}^{N} x_{ijm} = y_{jm}, j = 1...N, m = 1...M \tag{5}$$

$$\sum_{m=1}^{M} y_{jm} = 1, j = 1...N \tag{6}$$

$$C_{max} \geqslant \sum_{j=1}^{N} (p_j \times y_{jm}), m = 1...M \tag{7}$$

$$C_{min} \leqslant \sum_{j=1}^{N} (p_j \times y_{jm}), m = 1...M \tag{8}$$

$$x_{ijm}, y_{jm} \in \{0,1\}; C_{max}, C_{min} \geqslant 0 \qquad (9)$$

In the above model, Equation 1 is the objective function that minimizes the workload imbalance among the machines. Equation 2 assures that for each machine, only one real job follows the fictitious job 0. Equation 3 states that a job must be processed at one and only one position on a machine and it will be immediately preceded by exactly one job.Equation 4 states that if job i is processed on machine m, it will be immediately followed by at most one other job on this machine. Equation 5 states that job j should immediately follow another job on machine m if it is placed on this machine. Equation 6 guaranties that each job is assigned to exactly one machine. Equations 7 and 8 represent workload-balancing constraints. The first one ensures that the maximum workload is greater than or equivalent to individual workloads and the second one ensures that the minimum workload is smaller or equivalent to individual workloads. Equation 9 states the properties of the decision variables. It states also that the completion time of the fictitious job is equal to zero.

2.4. Scheduling algorithms

In the workload balancing problem, the different scheduling procedures are based on list-scheduling algorithm described by algorithm 1. These algorithms are known to be the best for solving the parallel machine scheduling problems to optimize utilization criteria [14], [6].
The main idea consists of selecting iteratively the machine with the smallest workload the the assignment of a new job from the list of unfinished jobs. Different strategies are adopted for the selection of jobs list such as: shortest processing times, longest processing times and random processing times. Based on a priority queue, the implementation complexity of a list-scheduling algorithm is of the order of $O(n\log(m))$.

Algorithm 1: List-scheduling algorithm

Input data

M	number of machines
N	number of jobs
$(p_1, ..., p_N)$	list of jobs processing times

for $m = 1$ to M
{
$W_m = 0$ workload of machine m
$S_m = \emptyset$ set of jobs assigned to machine m
}
for $i = 1$ to N
{
$m = argmin_k W_k$ machine with the smallest workload
$S_m = S_m \cup \{i\}$ assign job i to machine m
$W_m = W_m + p_i$ update workload of machine m
}
Return $S = [S_1, ..., S_M]$

3. Computational experiments

This section presents some numerical experiments based on different well known test configurations. These instances have been used by Liu and Wu [5] to compare between genetic and simulated annealing algorithms. Recently, Raghavendra and Murthy [10] have used the same instances to compare the performance of their genetic algorithm with different heuristic methods proposed by Heinrich [2] for minimizing the imbalance of workload among identical parallel machines. The design of the different numerical examples is reported in table 3. These instances present small scale scheduling problems with 7, 10 and 21 jobs to be processed by respectively 3, 2 and 6 machines. They present also a larger scale problem with 29, 30 and 33 jobs to scheduling on respectively 3, 10 and 5 machines.
The aim of this computational study is to provide the optimal solutions of these well known instances previously adopted in the literature. For each instance, the mathematical programming model has been solved using ILOG CPLEX 11.0 software. These optimal solutions allow the exact evaluation

of these different approached resolutions.
We compare also this mathematical formulation with the classical formulation proposed by Rajakumar et al. [14]. In their model, the authors have formulated the workload balancing problem among identical parallel machines as a binary mixed integer program with the objective of minimizing the maximum of workloads. Or, the numerical results reported in tables 6 and 7 illustrate that this formulation does not provide the optimal workload repartition.
We present for each instance, the numerical results obtained by the different approximate algorithms given in the literature and the exact solutions obtained by the mathematical programming model (see tables 2, 3, 4, 5, 6 and 7). The comparative study presented by Raghavendra and Murthy [10] concludes that their genetic algorithm outperforms the different heuristics proposed by Heinrich [2] (STA, SYSR and IE) and the genetic and simulated annealing algorithms proposed by Liu and Wu [5]. So in this paper, we are not especially interested by the comparison with these approximate methods because we provide the optimal solutions which can be used to evaluate the real performances of the best known approximate resolution approach (genetic algorithm [10]).
We are also interested on comparison between the mixed integer linear program based on the minimization of the maximum of workload (MILP 1) and the proposed mathematical program (MILP 2).
We note that for the small scale instances with 7 and 10 jobs, both approximate methods and maximum completion time minimization model obtain the optimal solutions. Which is predictable. In the case of the third small instance with 21 jobs, the best approached method fails to obtain the optimal solution with a maximum completion time of 58 time units and a maximum workload imbalance of 2 time units. The two large instances with respectively 29 and 30 jobs (see tables 6 and 7), illustrate that the mathematical programming approach based on the minimization of the maximum completion time is does not provide the optimal repartition of the workload among the different machines. In fact for both examples, the maximum workload imbalance obtained by this model is equal to 2 time units while the second formulation obtains an optimal solution with one time unit of workload imbalance.
Table 8 presents the computational times of the two mathematical programming methods (MILP 1 and MILP 2). We not that the second model is more consuming in term of computational times but it provides the optimal solutions. However, the computational times still reasonable (less than one minute).

4. Conclusion

In this paper, a mixed linear integer mathematical program for workload balancing on identical parallel machine problem is presented. The workload balancing problem is defined as the minimization of the difference between the workload of the bottleneck machine and the workload of the fastest machine. Based on this mathematical programming model, we provide optimal solutions for different test instances presented in the literature in reasonable computational times. These solutions allow the exact evaluation of the different approximate methods presented in the literature. As a second contribution, we have illustrated based on some numerical examples that the classical formulation of workload balancing problem using the minimization of the maximum machine completion time does not provide the optimal workload repartition.
A direct extension of this research is to consider this new formulation of the workload imbalance minimization problem in the definition of the fitness functions of the approximate resolution algorithms proposed in the literature. Another possible extension of this contribution would be the combination of workload imbalance criterion as defined here with other scheduling criteria for multiobjective formulations and resolutions.

Acknowledgments

The authors would like to thank the editors and anonymous reviewers for their valuable remarks, comments and suggestions that helped to improve this paper.

Table 1: Design of experiment instances

$(N=7,M=3)$		$(N=10,M=2)$		$(N=21,M=6)$		$(N=29,M=3)$		$(N=30,M=10)$		$(N=33,M=5)$	
job i	p_i	job i	p_i	job i	p_i	job i	p_i	job i	p_i	job i	p_i
1	6	1	3	1	23	1	14	1	3	1	23
2	6	2	2	2	29	2	16	2	2	2	29
3	4	3	6	3	21	3	26	3	6	3	7
4	4	4	4	4	11	4	3	4	4	4	2
5	4	5	5	5	20	5	25	5	5	5	25
6	3	6	7	6	28	6	3	6	7	6	10
7	3	7	8	7	10	7	11	7	9	7	14
-	-	8	6	8	18	8	25	8	13	8	3
-	-	9	2	9	1	9	2	9	4	9	26
-	-	10	6	10	6	10	24	10	12	10	23
-	-	-	-	11	28	11	11	11	10	11	10
-	-	-	-	12	19	12	25	12	8	12	7
-	-	-	-	13	5	13	21	13	22	13	30
-	-	-	-	14	6	14	14	14	11	14	25
-	-	-	-	15	28	15	25	15	8	15	1
-	-	-	-	16	27	16	1	16	26	16	8
-	-	-	-	17	8	17	26	17	14	17	18
-	-	-	-	18	7	18	18	18	6	18	13
-	-	-	-	19	29	19	13	19	17	19	15
-	-	-	-	20	10	20	2	20	27	20	28
-	-	-	-	21	7	21	14	21	11	21	30
-	-	-	-	-	-	22	20	22	17	22	14
-	-	-	-	-	-	23	5	23	26	23	28
-	-	-	-	-	-	24	12	24	16	24	14
-	-	-	-	-	-	25	21	25	7	25	20
-	-	-	-	-	-	26	23	26	23	26	18
-	-	-	-	-	-	27	21	27	15	27	27
-	-	-	-	-	-	28	17	28	18	28	19
-	-	-	-	-	-	29	19	29	15	29	11
-	-	-	-	-	-	-	-	30	13	30	25
-	-	-	-	-	-	-	-	-	-	31	5
-	-	-	-	-	-	-	-	-	-	32	22
-	-	-	-	-	-	-	-	-	-	33	30

Table 2: solution for the second instance $(N=10,M=2)$

Machine load	Heinrich Kuhn 1995			GA	MILP 1	MILP 2
	STA	SYSR	IE			
Machine 1	11	10	10	10	10	10
Machine 2	10	10	10	10	10	10
Workload Imbalance	2	0	0	0	0	0

Table 3: solution for instance $(N = 7, M = 3)$

Machine load	Heinrich Kuhn 1995			GA	MILP 1	MILP 2 2
	STA	SYSR	IE			
Machine 1	11	10	10	10	10	10
Machine 2	10	10	10	10	10	10
Machine 3	9	10	10	10	10	10
Workload Imbalance	2	0	0	0	0	0

Table 4: solution for instance $(N = 33, M = 5)$

Method	Heinrich Kuhn 1995			GA	MILP 1	MILP 2
Machine load	STA	SYSR	IE			
Machine 1	99	115	116	116	116	116
Machine 2	151	118	116	116	116	116
Machine 3	98	110	116	116	116	116
Machine 4	89	119	116	116	116	116
Machine 5	143	118	116	116	116	116
Workload Imbalance	62	9	0	0	0	0

Table 5: solution for instance $(N = 21, M = 6)$

Method	Heinrich Kuhn 1995			GA	MILP 1	MILP 2
Machine load	STA	SYSR	IE			
Machine 1	74	61	57	58	56	57
Machine 2	42	49	58	58	57	57
Machine 3	60	61	58	57	57	57
Machine 4	33	59	58	56	57	56
Machine 5	48	49	53	56	57	57
Machine 6	84	62	57	56	57	57
Workload Imbalance	51	13	5	2	1	1

Table 6: solution for instance $(N = 29, M = 3)$

Method	Heinrich Kuhn 1995			GA	MILP 1	MILP 2
Machine load	STA	SYSR	IE			
Machine 1	161	150	153	152	153	152
Machine 2	148	153	153	153	153	152
Machine 3	148	154	151	152	151	153
Workload Imbalance	13	4	2	1	2	1

Table 7: solution for instance $(N = 30, M = 10)$

Method	Liu and Wu 1999	GA	MILP 1	MILP 2
Machine load				
Machine 1	30	37	38	37
Machine 2	40	38	37	37
Machine 3	37	39	37	37
Machine 4	40	38	38	38
Machine 5	33	39	37	38
Machine 6	39	36	38	38
Machine 7	41	37	36	37
Machine 8	34	36	38	38
Machine 9	41	39	38	38
Machine 10	40	36	38	37
Workload Imbalance	11	3	2	1

Table 8: Computational times of exact resolution

-	MILP 1 CPU time (seconds)	MILP 2 CPU time (seconds)
Instance $(N = 7, M = 3)$	1.23	1.31
Instance $(N = 10, M = 2)$	0.85	1.43
Instance $(N = 21, M = 6)$	1.58	3.36
Instance $(N = 29, M = 3)$	1.73	1.17
Instance $(N = 30, M = 10)$	20.43	45.05
Instance $(N = 33, M = 5)$	20.21	26.25

References

1. I. Caragiannis. Better bounds for online load balancing on unrelated machines. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 972–981, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.
2. K. Heinrich. A heuristic algorithm for the loading problem in flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 7:229–254, 1995.
3. J.C. Ho, T.L. Tseng, A.J. Ruiz-Torres, and F. J. López. Minimizing the normalized sum of square for workload deviations on m parallel processors. *Computers and Industriel Engineering.*, 56(1):186–192, 2009.
4. T. Keskinturk, M.B. Yildirim, and M. Barut. An ant colony optimization algorithm for load balancing in parallel machines with sequence-dependent setup times. *Computers & Operations Research*, 39(6):1225 – 1235, 2012.
5. M. Liu and C. Wu. A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines. *Artificial Intelligence in Engineering*, 13(4):399 – 403, 1999.
6. E. Mokotoff. Parallel machine scheduling problems: A survey. *Asia - Pacific Journal of Operational Research*, 18:193 – 242, 2001.
7. D. H. Moon, D. K. Kim, and J. Y. Jung. An operator load-balancing problem in a semi-automatic parallel machine shop. *Computers and Industrial Engineering*, 46(2):355–362, 2004.
8. Y. Ouazene, F. Hnaien, F. Yalaoui, and L. Amodeo. The Joint Load Balancing and Parallel Machine Scheduling Problem. Operations Research Proceedings, pages 497–502. Springer Berlin Heidelberg, 2011.
9. B. V. Raghavendra and A. N. N. Murthy. Some solution approaches to reduce the imbalance of workload in parallel machines while planning in flexible manufacturing system. *International Journal of Engineering Science and Technology*, 2(5), 2010.
10. B. V. Raghavendra and A. N. N. Murthy. Workload balancing in identical parallel machine scheduling while planning in flexible manufacturing system using genetic algorithm. *ARPN Journal of Engineering and Applied Sciences*, 6(1), 2011.
11. B. V. Raghavendra, A. N. N. Murthy, and M. Jayaram. Job sequence to minimize the workload imbalance on parallel machines through genetic algorithm. *International Journal of Engineering Science and Technology*, 3(1), 2011.
12. B. V. Raghavendra, A. N. N. Murthy, and N. B. Rao. Some solution approaches to reduce the imbalance of workload in parallel machines while planning in flexible manufacturing system through genetic algorithm. *International Journal of Engineering Science and Technology*, 2(1), 2010.
13. S. Rajakumar, V.P. Arunachalam, and V. Selladurai. Workflow balancing strategies in parallel machine scheduling. *International Jornal of Advanced Manufacturing*, 23:366–374, 2004.
14. S. Rajakumar, V.P. Arunachalam, and V. Selladurai. Workflow balancing in parallel machines through genetic algorithm. *International Journal of Advanced Manufacturing Technology*, 33:1212–1221, 2007.
15. M.B. Yildirim, E. Duman, K. Krishna, and K. Senniappan. Parallel machine scheduling with load balancing and sequence dependent setups. *International Journal of Operations Research*, 4(1):42–49, 2007.