

IMPROVED WORKLOAD BALANCING OF THE SCHEDULING JOBS WITH THE RELEASE DATES IN IDENTICAL PARALLEL MACHINES

Ali Mohamed Hussien, Mohamed Eid Hussein, Moatamad Refaat Hassan

Computer Science, Mathematics Department.
Faculty of Science, Aswan University, Aswan 81511, Egypt.

ABSTRACT

This paper presents the identical parallel machine's scheduling problem when the jobs are submitted over time. This problem consists of assigning N various jobs to M identical parallel machines to reduce the workload imponderables among the different machines. We generalized the mixed-integer linear programming approach to decrease the workload imbalance between the different machines, and that is done by converting the problem to the mathematical model. The studied cases are presented for different problems, and it indicates to an online system, and this system does not know the arrival times of the jobs before and reduce Makespan criterion is not well appropriate to describe the utilization for this online problem. The obtained results proved good solutions for the scheduling problem compared with standard algorithms.

KEYWORDS

Identical Parallel Machine, Scheduling Jobs, Workload Balancing, Release Date, Linear Programming.

1. INTRODUCTION

Parallel manufacturing body is one of the solutions for enhancing the processing capability of a manufacturing system. Two decisions are taken at once in parallel machine scheduling. The first is to locate the assignment of tasks to the machines, and the second is the sequencing of the tasks assigned to each machine to optimize various objective functions. One of these objective functions is the workload imponderable's decrease which can be defined as assigning available workload among various machines as equally possible. Workload balancing is serious for both the service, and manufacturing industries. In the service industry, people should have a balanced workload to be equitable and provide a quality service. The goal of a manager is to assign tasks to each worker so; their workloads are as much as potential. In the manufacturing industry, balancing the workload among the machines is serious to minimize the idle times and work-in-process, Y. Ouazene et al. [16]. It helps also to remove the bottlenecks in manufacturing systems, Rajakumar et al. [13]. Addressed the workload balancing problem using various priorities basics such as random, the shortest processing times and longest processing times. The authors used the proportional variation of imbalance to evaluate the performances of these various strategies. In their next pamphlet, Rajakumar et al. [14], the authors proposed a genetic algorithm (GA) that outperforms these various strategies. Above classical priority rules, Raghavendra and Murthy [1] made overworks to decrease the imbalance in a random type of parallel machines addressing the loading problem in an elastic manufacturing system. Later, Raghavendra et al. [4] proposed a GA based approach with short processing time (SPT) and longest processing time (LPT) rules for a decrease in the imbalance

between the parallel machines. The authors have completed that their GA supplies better solutions than the strategies proposed, Raghavendra and Murthy [1]. Raghavendra et al. [3]. Applied this GA in a case study for ten various side styles with various batch quantum on two vertical machining centers. The itself genetic-based heuristic's algorithm was compared with other approximate approaches proposed in the literature. Presented a comparative study among various test examples to illustrate the competence of their algorithm. The authors have shown that their algorithm outperforms various heuristics proposed by Heinrich [9] and the GA and simulated boost algorithms proposed by Liu and Wu [12]. Caragiannis [7] presented an improved upper limit for the greedy algorithm to decrease the Lateness (LP) norm of the machine loads for the problem of scheduling permanent tasks on unrelated machines. As mentioned above, the workload balancing problem is present in various environments. For example, Moon et al. [5] considered this problem with diverse operators in a semi-automatic parallel machine shop with two kinds of machines.

Their objective was to assign the tasks to the machines and allocate the machines to operators in such a way to reduce the operator's workload imbalance under the chain of machine availability and operator times. The authors have proposed a nonlinear mathematical formulation of this problem. In the literature review, the workload balancing problem has been related to various scheduling criteria in various ways, even by considering the workload imbalance as a chain or as an objective. Ouazene et al. [16] addressed the identical parallel machine scheduling problem to decrease altogether total tardiness and workload imbalance. The authors proposed a mathematical form and GA based on the collecting of the two objective functions. Yildirim et al. [10] have studied the scheduling of semi-related machines with sequence-dependent setups and load balancing chains. The authors proposed a mathematical formulation of the problem, and a convergent resolution based on some heuristics and GA. Latterly, Keskinurk et al. [15] presented a non-linear mathematical model for a parallel machine problems with sequence-dependent setups with the objective of lower total relative imponderables.

The authors proposed two Metaheuristic techniques for the approached resolution of the problem. The two Metaheuristic depend on an ant colony optimization algorithm and GA. Based on diverse randomly generated tests, the authors have concluded that the ant colony algorithm outperforms both heuristics and GA. This paper aims to generalize the mixed-integer linear programming approach to decrease the difference between the greats and smallest machine's workload in the definition of the appropriate functions of the approximate resolution algorithms proposed in the literature, and we selected actual the online problems where the times of jobs are not known in advance and minimizing the Makespan is not suitable away to evaluate the utilization of the online problems.

The rest of this paper is organized as follows: In the second section, Details the mathematical formulation of the problem considered. Various notations and decisions variables are described. Also, the conventional list-scheduling algorithm usually used in workload balancing on parallel machines problem be illustrated. Sections Three and four, Presents the comparative study based on various test examples reported in the literature. Section 5, summarizes the donating of this paper and gives some view for future possible extensions.

2. PROBLEM FORMULATION

In this paper, the problem deals with many jobs (N jobs) to be processed on a few of identical parallel machines (M machines) see Fig.1. The objective is to improve workload balancing of all jobs with the release dates. To solve the problem statement, the following assumptions are used:

- All machines are identical and are able to implement all operations.
- Each machine can only process one job at a time and each job needs only one machine for processing.
- Preemption of jobs are not allowed.
- Job submissions are not times known in advance.

The problem is formulated as a linear programming model to improve workload balancing of jobs with release dates.

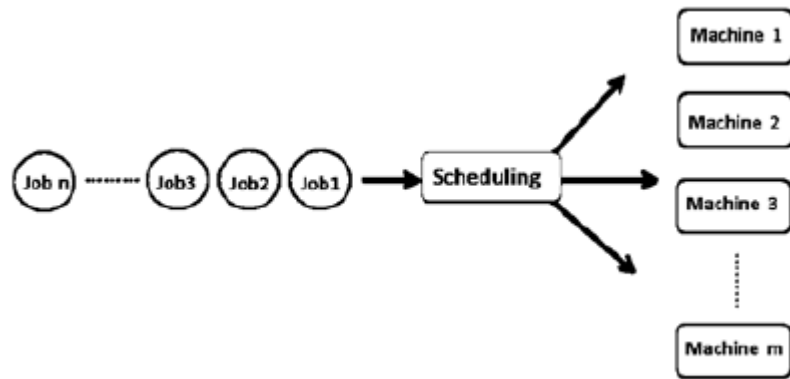


Fig.1. scheduling jobs in identical parallel machines

2.1. Notification

Table 1. Notations for scheduling problem that using in this paper

Symbol	Description
N	The number of jobs
M	The number of machines
$i, j \in \{0, 1, \dots, N\}$	Job indicator where job 0 is a fanciful one which is always sequenced at the first position on a machine
P_j	The processing time of job j
C_m	workload of machine m
S_m	Group of jobs scheduled on the machine m
C_j	The completion time of job j
r_j	The release time of job j
C_{max}	The last job's maximum completion time
C_{min}	The last job's minimum completion time

2.1. Integer Linear programming Model

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

2.2.1. Decision Variables

$$X_{ijm} = \begin{cases} 1, & \text{when machine } m \text{ executed job } j \\ & \text{directly after job } i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$y_{jm} = \begin{cases} 1, & \text{if job } j \text{ is processed on} \\ & \text{machine } m \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$C_j = (p_j + r_j)X_{0jm} \quad \forall j = 1 \dots N, m = 1 \dots M \quad (3)$$

$$C_j = \begin{cases} (C_i + p_j)X_{ijm}, & \text{if } r_j \leq C_i \\ (r_j + p_j)X_{ijm}, & \text{if } r_j > C_i \end{cases} \quad \forall i = 1 \dots N, j = \\ 1 \dots N, m = 1 \dots M \quad (4)$$

$$C_m = \max_{1 \leq j \leq N} \{C_j y_{jm}\} \quad \forall m = 1 \dots M \quad (5)$$

$$C_{max} = \max_{1 \leq m \leq M} \{C_m\} \quad (6)$$

$$C_{min} = \min_{1 \leq m \leq M} \{C_m\} \quad (7)$$

2.2.2. Mathematical model

$$Min C = C_{max} - C_{min} \quad (8)$$

subject to:

$$\sum_{j=1}^N x_{0jm} \leq 1, \quad \forall m = 1 \dots M \quad (9)$$

$$\sum_{i=0, i \neq j}^N \sum_{m=1}^M x_{ijm} = 1 \quad \forall j = 1 \dots N \quad (10)$$

$$\sum_{j=1, j \neq i}^N x_{ijm} \leq y_{jm} \quad \forall i = 1 \dots N, m = 1 \dots M \quad (11)$$

$$\sum_{i=0, i \neq j}^N x_{ijm} = y_{jm} \quad \forall j = 1 \dots N, m = 1 \dots M \quad (12)$$

$$\sum_{m=1}^M y_{jm} = 1 \quad \forall j = 1 \dots N \quad (13)$$

$$C_{max} \leq \max_{1 \dots M} (C_j \times y_{jm}) \quad \forall j = 1 \dots N, m = 1 \dots M \quad (14)$$

$$C_{min} \geq \min_{1 \dots M} (C_j \times y_{jm}) \quad \forall j = 1 \dots N, m = 1 \dots M \quad (15)$$

$$x_{ijm}, y_{jm} \in \{0,1\}; C_{max}, C_{min} \geq 0 \quad (16)$$

$$C_j - p_j \geq r_j \quad \forall j = 1 \dots N \quad (17)$$

In the model, Eq. (8) is the objective function which minimizes the workload balancing among the machines. Eq. (9) assures that for each machine, one real job follows the fanciful job 0. Eq. (10) states that a job must be processed at only one position on a machine and it will be preceded by exactly one job. Eq. (11) states that if job i is processed on machine m ; it will be followed by at most one other job on this machine. Eq. (12) states that job j should directly follow another job on machine m if it is placed on this machine. Eq. (13) guarantees that each job is assigned to exactly one machine. Eq. (14) and Eq. (15) represent workload constraints. The first one ensures that the maximum workload is larger than or equivalent to individual workloads, and the second one ensures that the minimum workload is the least than or equivalent to individual workloads. Eq. (16) states the properties of the decision variables. It states also that the fanciful job's completion time is equal to zero. Eq. (17) guarantees that any job cannot start before its arrival time (r_j), wherefore the difference between the completion time and processing time of job j is greater than or equal the release time of this job.

3. EXPERIMENTS

This section attends some numerical experiments with various well-known test configurations. These instances have been used by Liu and Wu [12] to compare GA and simulated annealing algorithms. Recently, Raghavendra and Murthy [2] have used the same instances to compare the performance of their GA with different heuristic methods proposed by Heinrich [9] for minimizing the imbalance of workload among identical parallel machines. The design of the various numerical examples is reported in table 1 [17]. These instances present small scale scheduling problems with 7 jobs to be processed by respectively 3 machines. They present also a large scale problem with 29 and 21 jobs to scheduling on respectively 3 and 6 machines and other cases mentioned in paper [16]. The goal of this computational study is to provide optimal solutions for these well-known instances formerly adopted in the literature. For every instance, the mathematical programming model has been solved using ILOG CPLEX 11.0 software [17]. These optimal solutions allow the exact evaluation of these various approached resolutions. The authors [17] 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 between identical parallel machines as a binary mixed integer program with the objective of decrease the maximum of workloads. Or, the numerical results reported in tables two and three illustrate that this formulation does not provide the optimal workload re-partition. We present for every instance, the numerical results obtained by various approximate algorithms in the literature, and the exact solutions obtained by the mathematical programming model (see tables 2, 3 and 4) [17]. The comparative study presented by Raghavendra and Murthy [10] concludes that their GA outperforms the different heuristics proposed by Heinrich [9] (STA, SYSR, and IE) and the GA and simulated annealing algorithms proposed by Liu and Wu [12]. So in paper [17], the authors are not especially interested in the comparison with these approximate methods because they provide the optimal solutions which can be used to evaluate the real performances of the best known approximate resolution approach (GA) [2]. They are also interested in the comparison between the mixed integer linear program based on the minimization of the maximum of workload (MILP 1) [17] and the proposed mathematical program (MILP 2) [17]. The authors note that for the small scale instances with seven jobs, both approximate methods and maximum completion time minimization model obtain the optimal solutions. 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 jobs (see tables 3 and 4)

[17] illustrate that the mathematical programming approach based on the decrease of the maximum completion time does not provide the optimal repartition of the workload among various 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. We note that the second model is more consuming in terms of computational times but it provides the optimal solutions.

From the previous studies, we generalize the mixed integer linear programming approach to decrease the difference between the greatest and smallest machine's workload in the definition of the appropriate functions of the approximate resolution algorithms and we selected the realistic of the online problems where the times of jobs are not known in advance. For every instance, the mathematical programming model has been solved using python 3.7 programming language and LINGO 18.0.44 software.

4. COMPUTATIONAL RESULT

In this section, we used mixed integer linear programming to find the optimal solution in the five cases in table 2 with 7,10,13,21 and 29 jobs with release dates to be processed by 3,2,4,6 and 3 identical parallel machines. In five cases for tasks and machines, we have selected release dates randomly. We compare our results by longest processing time (LPT) and the column header with the symbol (MILP2R) is our results.

5. DISCUSSION

This paper is presented and solved the optimal component's assignment concerning coexistence of one constraint: release dates. Previous studies focused on studying one objective is lower Makespan for each machine without release dates such as Heinrich Kuhn 1995 [9] and GA [12] and the study [17] focus the maximum workload and the minimum workload in the same time to balancing in identical parallel machine scheduling.

Table 2 shows all input data in all the studied cases, table 3, 4, 5, 6 and 7 show the results of applying the proposed approach in comparison with LPT algorithm from these results it is observed that the proposed approach leads to good values.

We note that for the small scale instances with 7 jobs in table 4.3, both LPT algorithm and MILP2R approach are similar in difference between maximum and minimum Makespan that it is equal to one unit time but in large scale instances that show in table 4.4,4.5, 4.6 and 4.7, the MILP2R approach is better than LPT algorithm, we obtain the optimal solution in the case 3 and in case 2,4 and 5 the maximum workload imbalance obtained by LPT algorithm is equal to 2,21 and 11 time units while MILP2R approach obtains an optimal solution with two time unit in case 2 and one time unit of workload imbalance in case 4 and 5

These results, we proved that LPT algorithm is not better than MILP2R because we used an online model, where in our online model, all jobs are submitted over time and minimizing Makespan criterion is not well suited to describe utilization for this online problem [11] show Fig.2, Fig.3, Fig.4, Fig.5, and Fig.6. While LPT processes the tasks with the largest time first and then the least according to their access to the system.

6. CONCLUSION

In this paper, integer linear programming for workload balancing on an identical parallel Machine problem is presented. The workload balancing problem is the decrease of the difference among the workload of the bottleneck machine and the workload of the fastest machine. After this generalized, mathematical programming model, we provide optimal solutions for various test instances presented in the literature in reasonable computational times. These solutions allow the exact evaluation of various approximate methods presented in the literature. As an addition, 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 re-partition. A direct extension of this paper is to consider generalize formulation of the workload imbalance minimization in the online problems, which are more important than offline problems because the Jobs submissions are not times known in advance and minimizing Makespan criterion is not well suited to describe utilization for the online problems.

Table 2. The input data for all studied cases

Case (1) (N=7,M=3)			Case (2) (N=10,M=2)			Case (3) (N=13,M=4)			Case (4) (N=21,M=6)			Case (5) (N=29,M=3)		
j	P _j	r _j	j	P _j	r _j	j	P _j	r _j	j	P _j	r _j	j	P _j	r _j
1	3	0	1	3	0	1	5	0	1	23	0	1	14	0
2	2	0	2	2	0	2	11	0	2	29	0	2	16	0
3	6	0	3	6	0	3	9	0	3	21	0	3	26	0
4	4	0	4	4	9	4	8	0	4	11	0	4	3	0
5	4	8	5	5	12	5	7	0	5	20	0	5	25	0
6	7	4	6	7	0	6	3	0	6	28	0	6	3	0
7	8	3	7	8	10	7	8	11	7	10	26	7	11	0
-	-	-	8	6	18	8	6	13	8	18	0	8	25	47
-	-	-	9	2	17	9	9	0	9	1	26	9	2	102
-	-	-	10	6	19	10	2	19	10	6	26	10	24	47
-	-	-	-	-	-	11	5	16	11	28	20	11	11	71
-	-	-	-	-	-	12	2	19	12	19	26	12	25	47
-	-	-	-	-	-	13	9	12	13	5	26	13	21	0
-	-	-	-	-	-	-	-	-	14	6	26	14	14	71
-	-	-	-	-	-	-	-	-	15	28	20	15	25	0
-	-	-	-	-	-	-	-	-	16	27	20	16	1	0
-	-	-	-	-	-	-	-	-	17	8	47	17	26	47
-	-	-	-	-	-	-	-	-	18	7	47	18	18	71
-	-	-	-	-	-	-	-	-	19	29	47	19	13	71
-	-	-	-	-	-	-	-	-	20	10	47	20	2	102
-	-	-	-	-	-	-	-	-	21	7	47	21	14	71
-	-	-	-	-	-	-	-	-	-	-	-	22	20	105
-	-	-	-	-	-	-	-	-	-	-	-	23	5	127
-	-	-	-	-	-	-	-	-	-	-	-	24	12	127
-	-	-	-	-	-	-	-	-	-	-	-	25	21	105
-	-	-	-	-	-	-	-	-	-	-	-	26	23	105
-	-	-	-	-	-	-	-	-	-	-	-	27	21	105
-	-	-	-	-	-	-	-	-	-	-	-	28	17	127
-	-	-	-	-	-	-	-	-	-	-	-	29	19	127

Table 3. Solution for case (1) (N=7, M=3)

Method	LPT	MILP2R
Machine 1	11	11
Machine 2	12	12
Machine3	12	11
$C_{\max} - C_{\min}$	1	1

Table 4. Solution for case (2) (N=10, M=2)

Method	LPT	MILP2R
Machine 1	24	24
Machine 2	26	25
$C_{\max} - C_{\min}$	2	1

Table 5. Solution for case (3) (N=13, M=4)

Method	LPT	MILP2R
Machine 1	21	21
Machine 2	22	21
Machine 3	21	21
Machine 4	21	21
$C_{\max} - C_{\min}$	1	0

Table 6. Solution for case (4) (N=21, M=6)

Method	LPT	MILP2R
Machine 1	54	57
Machine 2	55	56
Machine 3	50	57
Machine 4	56	57
Machine 5	55	57
Machine 6	71	57
$C_{\max} - C_{\min}$	26	1

Table 7. Solution for case (5) (N=29, M=3)

Method	LPT	MILP2R
Machine 1	158	153
Machine 2	152	152
Machine 3	147	152
$C_{\max} - C_{\min}$	11	1

Fig.2. The Gantt chart for case (N=7, M=3): LPT Algorithm and MILP2R Approach

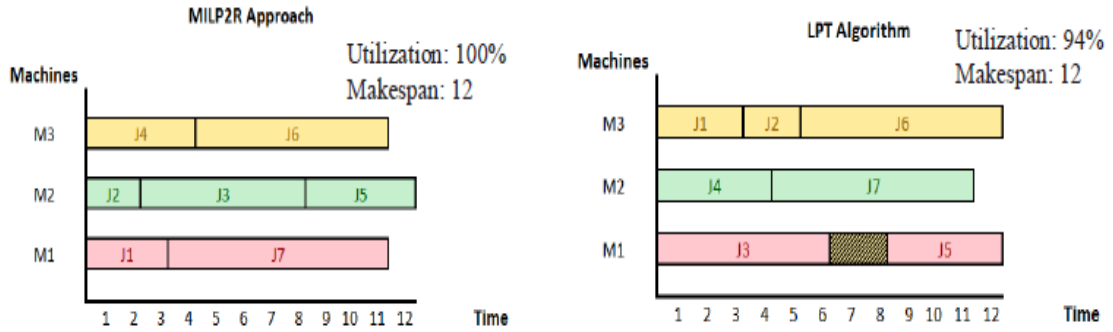


Fig.3. The Gantt chart for case (N=10, M=2): LPT Algorithm and MILP2R Approach

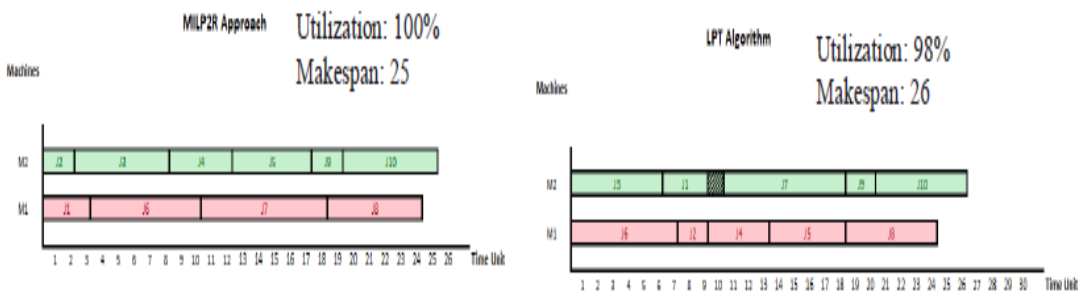


Fig.4. The Gantt chart for case (N=13, M=4): LPT Algorithm and MILP2R Approach

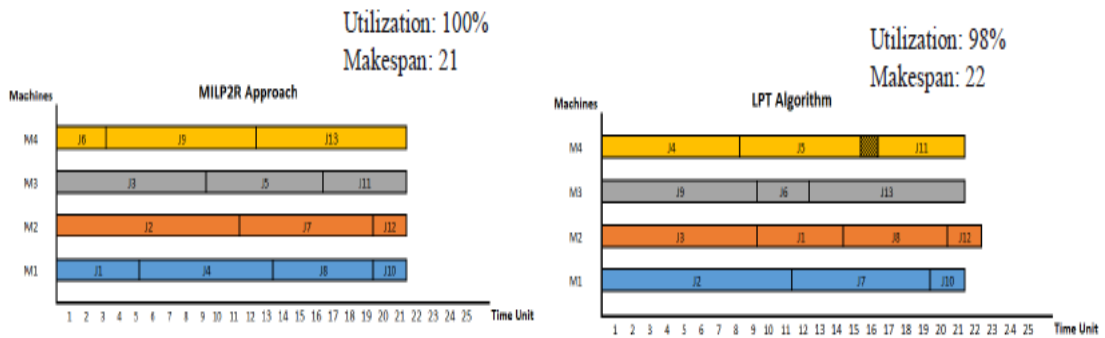
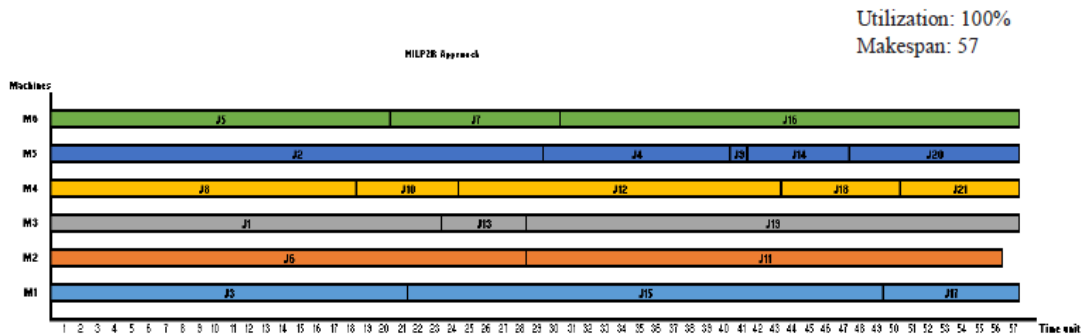
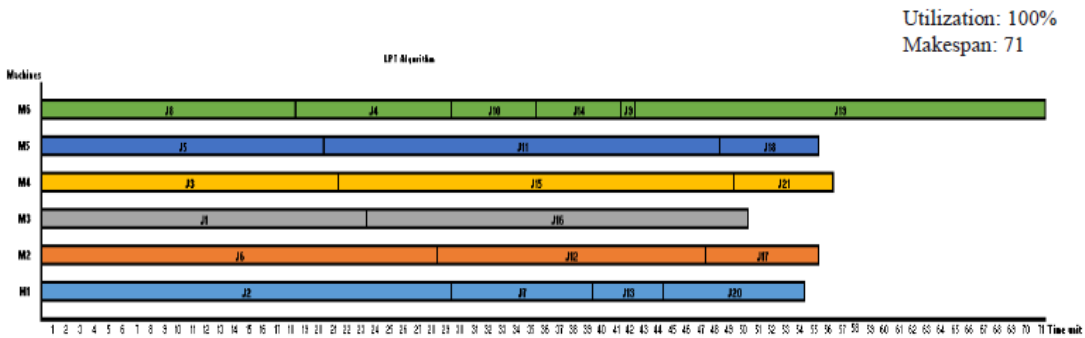


Fig.5. The Gantt chart for case (N=21, M=6): MILP2R Approach (a) and LPT Algorithm (b)

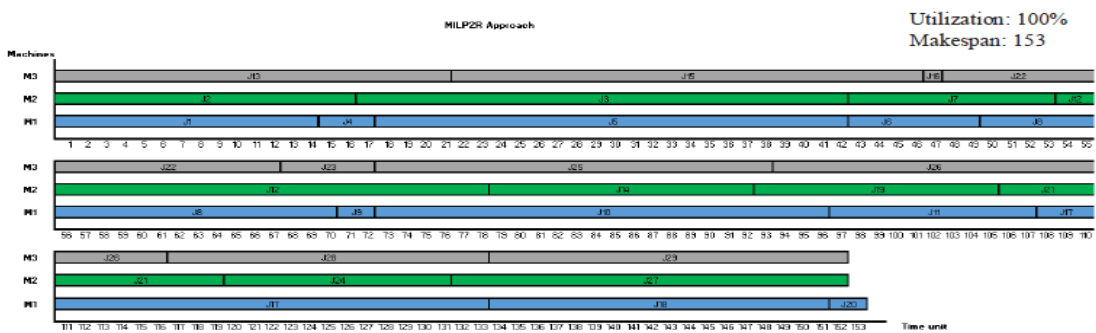


(a)

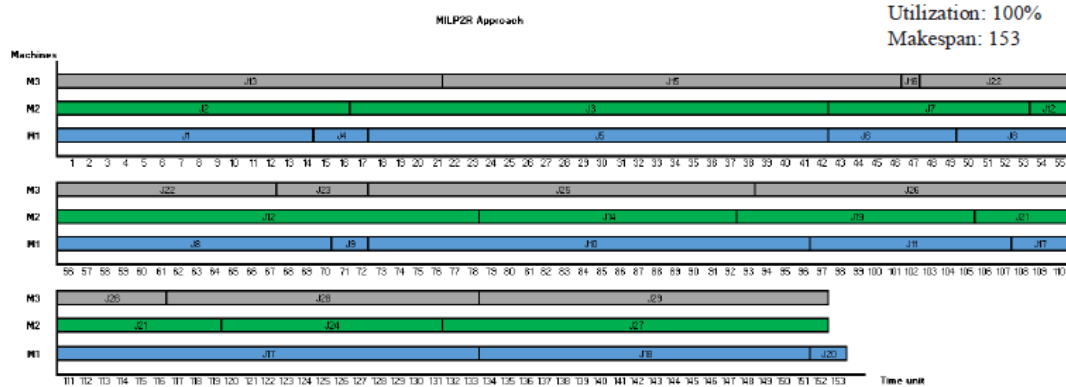


(b)

Fig.6. The Gantt chart for case (N=29, M=3): MILP2R Approach (a) and LPT Algorithm (b)



(a)



(b)

Acknowledgement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

- [1] 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.
- [2] B. V. Raghavendra and A. N. N. Murthy. Workload balancing in identical parallel machine scheduling while planning in flexible manufacturing system using genetic algorithm. *ARNP Journal of Engineering and Applied Sciences*, 6(1), 2011.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] E. Mokotoff. Parallel machine scheduling problems: A survey. *Asia - Pacific Journal of Operational Research*, 18:193 – 242, 2001.
- [7] 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.
- [8] J.C. Ho, T.L. Tseng, A.J. Ruiz-Torres, and F. J. L’opez. Minimizing the normalized sum of square for workload deviations on m parallel processors. *Computers and Industrial Engineering*, 56(1):186–192, 2009.

- [9] K. Heinrich. A heuristic algorithm for the loading problem in flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 7:229–254, 1995.
- [10] 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.
- [11] M. Eid and U. Schwegelshohn. Utilization of nonclairvoyant online schedules. *Theoretical Computer Science* 362: 238 – 247, 1-3(2006).
- [12] 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.
- [13] S. Rajakumar, V.P. Arunachalam, and V. Selladurai. Workflow balancing strategies in parallel machine scheduling. *International Journal 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] T. Keskinurk, 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.
- [16] 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.
- [17] Y. Ouazene, F. Yalaoui, H. Chehade, and A. Yalaoui. Workload balancing in identical parallel machine scheduling using a mathematical programming method. *International Journal of Computational Intelligence Systems*, Vol. 7, Supl: 58-67, 2014.