# Scheduling to Minimize Makespan on Identical Parallel Machines

Yousef German*, Ibrahim Badi*, Ahmed Bakir*, Ali Shetwan**
* Faculty of Engineering, Misurata University, Misurata, Libya
** The college of Industrial Technology, Misurata, Libya

**Abstract-** Reducing production time is an important factor for companies which their main objectives are to maximize profits and minimize costs. To achieve these objectives one must follow the scientific methods for scheduling production time. This paper studies the Makespan Minimization for Identical Parallel Machines. The problem involves an assignment number of jobs ($N$) to a set of identical parallel machines ($m$), when the objective is to minimize the makespan (maximum completion time of the last job on the last machine of the system). In the literature the problem is denoted by ( $P_m||C_{\max}$ ). The objective of this study is to find the optimal schedule (solution) for identical parallel machines scheduling problems, by using hypothetical situation under defined assumptions and constraints. Algorithms are developed and used in this paper to represent and solve the problem. Integer Linear Programming model (ILP) is used to formulate the problem. Longest Processing Time algorithm (LPT) is used to find (generate) the initial solution, then the developed algorithm is used to improve the initial solution. The solution of the algorithms are coded in MATLAB. The results demonstrated that, the mathematical modeling and the algorithms are powerful tools and are more effective for these kinds of problems, compared with traditional methods.

**Keywords:** Makespan, Parallel machines, Integer Linear Programming

## 1 INTRODUCTION

In this study the problem deals with number of jobs ($\boldsymbol{N}$ jobs) to be processed on a number of identical parallel machines ($\boldsymbol{m}$ machines), when the objective is to minimize makespan (maximum completion time of the last job). The problem denoted by ( $P_m||C_{\max}$ ). Figure (1) shows ($N$) independent jobs on ($m$) parallel machines.

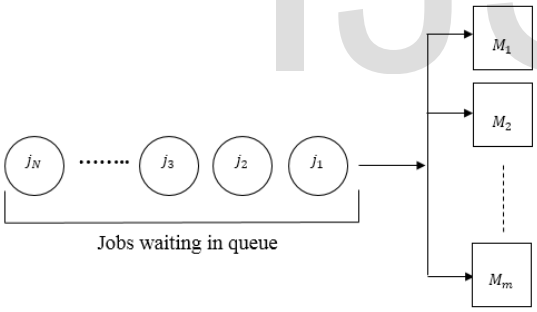


Fig. 1. ($m$) parallel machines with ($N$) jobs.

To solve the problem statement of this study, the following assumptions are used:

1. All machines are identical and are able to perform all operations.
2. Each machine can only process one task at a time.
3. Permission of a job on another machine is not allowed.
4. All jobs are available at time zero.

There are many studies in literature dealing with parallel machines scheduling problems. Although the relatively small sized of identical parallel machines problems can be solved by operational methods such as dynamic programming, branch and bond method. However, these methods still have limitations with a large number of jobs and when the number of machines are more than two.

## 2 LITERATUR REVIEW

Koulamas and Kyparisis proposed a modified longest processing time (MLPT) heuristic algorithm for the two uniform machine makespan minimization problem. The MLPT algorithm schedules the three longest jobs optimally first, followed by the remaining jobs sequenced according to the LPT rule. The obtained results showed that the MLPT rule has the tight worst-case bound of 1.22 [1], an improvement over the LPT bound of 1.28 [2].

Chaudhry et al presented a spreadsheet based on genetic algorithm approach to minimize the makespan for scheduling a set of tasks on identical parallel machines and worker assignment to the machines. The performance of the proposed approach is compared against two data sets of benchmark problems available on the internet. It was found that the proposed approach produces optimal solution for almost 95 percent of the problems demonstrating the effectiveness of the proposed approach [3].

Queiroz and Mendes addressed an identical parallel machines scheduling problem with release dates in which the total weighted tardiness has to be minimized. The obtained results showed that the instances of 10 jobs and 2 machines for which the metaheuristic achieved an optimal solution in a competitive amount of time, when compared with the exact approach. For the cases of 15 jobs and 3 machines and 20 jobs and 3 machines, the performance of the metaheuristic was similar [4].

A new method has been developed by Navid Hashemian to schedule jobs on parallel machines with availability constraints. The objective of the problem was to minimize the

makespan of the total production schedule. It was concluded that the exact algorithm is able to solve large-scale problems which are not solvable by any other method including the current best ILP solver. Based on the results of the experiments for the ILP, it has been demonstrated that the proposed ILP can be used to solve the small-size problems effectively. The performance of the proposed algorithm is independent of the pattern of non-availability periods [5].

The aim of this study is to find the optimal schedule (solution) for identical parallel machines scheduling. This paper presented backtracking algorithm to find the optimal schedule (solution) for identical parallel machines scheduling. Also the paper suggested an algorithm to improve the initial solution.

## 3 PARALLEL MACHINES SCHEDULLING

The aim of machine scheduling is to assign jobs to the machines based on related objective function to minimize operating time and increase productivity [6]. Parallel machines scheduling is the task of determining when each operation has to start and finish on each machine and using available resources in efficient manners to execute (assign) jobs or tasks on machines.

The parallel machines can be identical (uniform) or unrelated. In this paper the case of study is identical parallel machines. Identical parallel machines are a set of machines which have the same speed factor and they can process all the jobs. Figure (2) shows Gantt chart for set of machines and jobs.

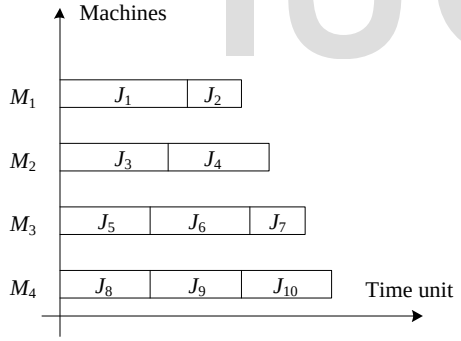


Fig. 2. Gantt chart for set of machines and jobs.

Assume that:

$C_{\max}$ : the makespan (maximum completion time).
$N$ : number of jobs (integer).
$m$ : number of machines. (integer).
$p_j$ :processing time of job ( $j$ ) (integer).
$x_{ij}$ : the assignment (decision) variable.

The mathematical model of the problem as follows:

$$Min \quad C_{\max} \tag{1}$$

subject to

$$\sum_{j=1}^{N} p_j x_{ij} \le C_{\max} \qquad i=1,\ldots,m \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad j=1,\ldots,N \tag{3}$$

$$x_{ij} \in \{0,1\} \qquad i=1,\ldots,m\ ,\ j=1,\ldots,N \tag{4}$$

$$C_{\max} \ge 0 \tag{5}$$

The first equation (1) in the model is the objective function $(C_{\max})$ makespan, which should be minimized.

Constraint (2) assures that the load on any machine is equal or less than ( $C_{\max}$ ).

Constraint (3) shows that each job must be assigned to exactly one machine.

Constraint (4) describes the type of the assignment decision variable ( $x_{ij}$ ):

$$x_{ij} = \begin{cases} 1 & \text{if the job } j \text{ is assigned to machine } i \\ 0 & \text{if the job } j \text{ is not assigned to machine } i \end{cases}$$

Constraint (5) shows that, the objective function $C_{\max}$ is integer variable.

## 4 Solution method

In this paper, the solution can be obtained in two steps: The LPT algorithm is used to find (generate) the initial solution, while the developed algorithm is used to improve the initial solution.

### 4-1 LPT algorithm

The (LPT) algorithm always puts the smaller jobs towards the end of schedule, that makes it easier to balance the machines loads. According to the (LPT) algorithm, whenever one of the ($m$) machines is freed, the longest job among number of jobs ($N$) in decreasing order waiting for processing is selected to be next **[16].**

The next job ($j$) will be scheduled on machine ( $i^*$ ) according to the equation:

$$i^* = argmin \left\{ L_i + p_j : i = 1, ..... , m \right\} \tag{6}$$

Where: $L_i$ is the load on machine ( $i$ ) , $p_j$ the processing time of job ( $j$ ).

The makespan ( $C_{\max}$ ) of any feasible solution is:

$$C_{max} = max \left\{ C_i : i = 1, ..... , m \right\} \tag{7}$$

The steps of LPT algorithm are as follows:

Step1: sort ($N$) jobs according to the non-increasing order of their processing time.

Step 2: set ( $j$=1 ).

Step 3: assign job ( $j$ ) to machine ($i$) according to equation (6).

Step 4: if $j$=$N$ (all jobs are allocated) then go to the next step, otherwise set $j$=$j$+1 and go to the step 3.

Step 5: calculate $C_{max}^{LPT}$ by using equation (7).

### 4-2 Developing the algorithm

The developed algorithm, is used as a main algorithm and is based on two types of operations: (construction and

backtracking). The load of machines are determined in sequence, one after another. Therefore, the potential load of machine ($i$) with ($1< i \le m$) depends on the loads of the previous machines, then the loads on machines continue until assigning the last job [4].

Assume that, ($C_{\max}$): is the makespan of the current feasible solution. So if the optimal solution has not been explored yet, then its value is not greater than ($C_{\max}-1$) in the case of integer processing times. Therefore:

$$UB = C_{\max} - 1 \quad (8)$$

Where ($UB$) is the upper bound for the load of all machines in the feasible solution and still to be investigated.

And the lower bound ($LB$) for all the other loads in the same feasible solution can be found by the following equation:

$$LB = \max\left\{0\ ,\ \sum_{j=1}^{N} p_j - (m-1)\,UB\right\} \quad (9)$$

Equation (9) implies that if all machines except one have a total load equal to the upper bound, then the remaining load is the lower bound. After finding a new feasible solution, both bounds (upper and lower bound) tighten up. To ensure that the load on any machine is feasible, the total load on the machine, must be between the lower and upper bounds.

$$LB \le \sum_{j=1}^{n} p_j\ k_j \le UB \quad (10)$$

Where: ($k_j$) is integer, and $j$=1,…..,$n$.

**4-2-1 Construction phase**

There are many feasible solutions that can satisfy the equation (10). For an implicit enumeration procedure, the feasible solutions must be ordered somehow and enumerated in this order. One easy way to perform this task is to order them in lexicographical order also known as (the dictionary order) or (the alphabetic order). In the construction phase, and to load the machines one by one, the largest solution in the lexicographical order for ($n$) jobs types is given by equations (11) and (12). That is when the machine has no previous load.

$$k_1 = \min\left\{\left\lfloor \frac{UB}{p_1} \right\rfloor, r_1\right\} \quad (11)$$

$$k_j = \min\left\{\left\lfloor \frac{UB - \sum_{h=1}^{j-1} p_h\ k_h}{p_j} \right\rfloor, r_j\right\},\ j = 2,\ldots..,\ n. \quad (12)$$

The feasibility of the load is checked by equation (3-10).

If the construction is successful (all machines are loaded and all conditions are satisfied), both the upper bound and lower bound are updated.

**4-2-2 Backtracking phase**

The algorithm is always looking for the optimal solution. Therefore the backtracking is applied whenever any of the following two situations are accounted:

1- When the load of a machine is not feasible (cannot satisfy equation(10)).

2- When a new feasible solution has been found for all machines (updating makespan).

When the machines has no further feasible load ( the load of machines are not satisfy equation (10) ). Then there is no feasible solution for the current upper bound and optimal makespan ($C^*_{\max}$) is equal to pervious upper bound.

$$C^*_{\max} = UB + 1 \quad (13)$$

**4-3 Steps of the improvement algorithm:**

Step 1: set ($i=1$) and ($t=1$).

Step 2: use (LPT) algorithm to find the initial solution (upper bound).

Step 3: load machine ($i$) according to the equations (11) and (12).

Step 4: if machine ($i$) does not satisfy inequality (10) go to step (6).

Step 5: if ($i=m$) set ($UB = C_{\max}-1$) and ($t = t+1$) and go to step (3). otherwise set ($i = i+1$) and go to step (3).

Step 6: calculate $C^*_{\max}$ by using equation (13).

Step 7: print results and stop.

Figure (3) shows the flow chart of the suggested algorithm.

**5 Computer program**

The developed algorithms have been coded in MATLAB and executed on Intel (R) core (TM)i5 CPU 2.30 GHz, RAM 4.00 GH. System type (64bit) in order to remain comparable in terms of the computational time. The program is designed to solve variety of problems depending on number of machines (m) and number of jobs (N).

Many computational applications with different levels of difficulties have been carried out to evaluate the performance of the suggested algorithm.

**5-1 Case (1)**

Find the optimal schedule (solution) by using the (LPT) and the algorithms to minimize maximum completion time ($C_{\max}$) for (7) jobs their processing times ($p_j$) given bellow:

$p_j$ = [3, 3, 3, 4, 4, 5, 5 ]

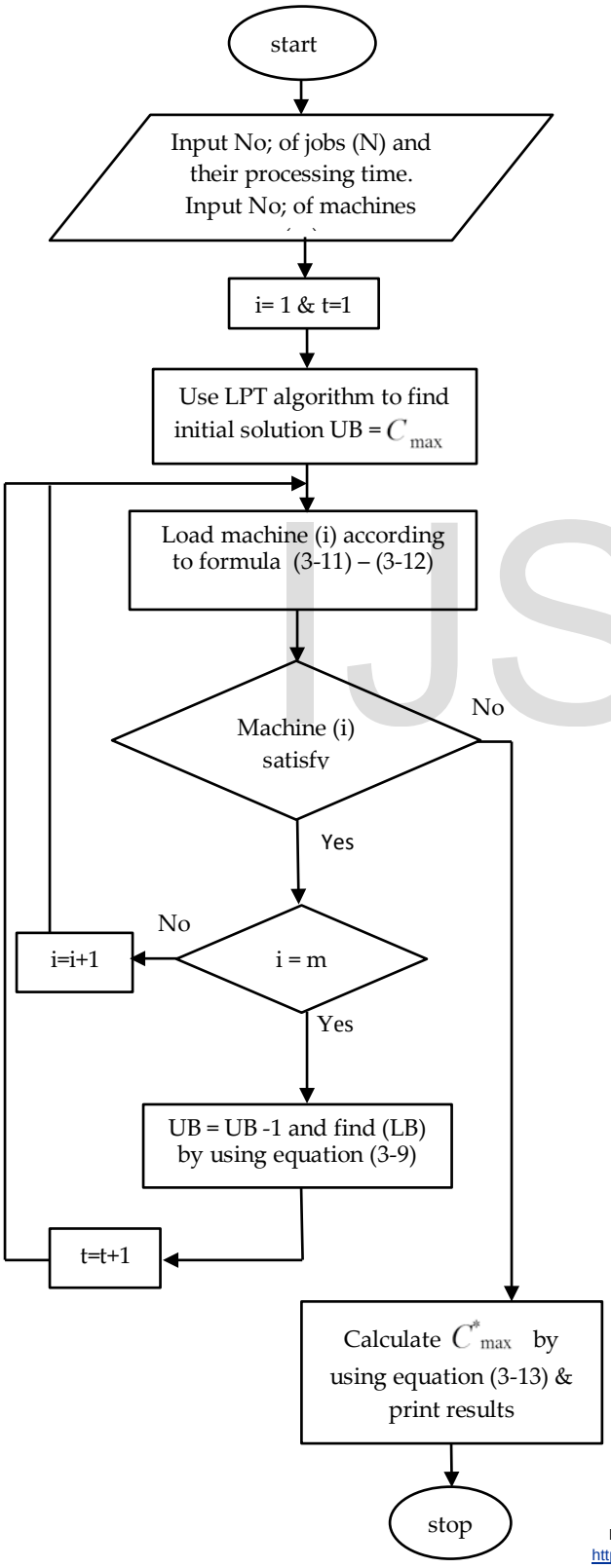


Fig. (3). The flow chart of the algorithm.

The jobs processed by (3) identical parallel machines, each machine can only process one job at any time and permission of a job on another machine is not allowed.

**5-1-1 Using LPT algorithm** to find the initial schedule (solution):

Step1: sort the jobs in non-increasing order (5, 5, 4, 4, 3, 3, 3)

Step 2: set $j$ =1

Step 3: assign job (1) to the machine ($i$) according to equation (6).

$$i = argmin\ \left\{ L_i + p_j : i = 1, ..... , m \right\}$$

The load on machines:

$C_1$ = P1 + P5 + P7 = 5 + 3 + 3 = 11

$C_2$ = P2 + P6 = 5+ 3 = 8

$C_3$ = P3 + P4 = 4+ 4 = 8

Step 4: $j = N = 7$ ( all jobs are assigned (allocated)).

Step 5: calculate $C_{max}$ by using equation (7).

$$C_{max} = max \left\{ C_i : i = 1, 2, 3 \right\} \rightarrow C_{max} = max \left\{ C_1, C_2, C_3 \right\}$$

$$C_{max} = max \left\{ 11, 8, 8 \right\} \rightarrow C_{max} = 11$$

The machines are loaded and the initial solution obtained by LPT algorithm are illustrated in Figure (4) by Gantt chart.

**5-1-2 Using the algorithm** to improve the initial solution:

Step 1: set ( $i = 1$ ) and ( $t = 1$ ).

Step 2: use (LPT) algorithm to find the initial solution (upper bound).

At first iteration $UB = C_{max} = 11$

The lower bound ( $LB$ ) can be found by equation (9) as follows:

$$LB = \max \left\{ 0, \sum_{j=1}^{N} p_j - (m-1)\, UB \right\}$$

$$LB = \max \left\{ 0, \sum_{j=1}^{7} p_j - (3-1)\, 11 \right\} \rightarrow LB = \max \left\{ 0, 27 - (2)\, 11 \right\} \rightarrow$$

$$LB = \max \left\{ 0, 5 \right\} \rightarrow LB = 5$$

Step3: load machine(1) according to the equations (11) &(12).

$$k_1 = \min \left\{ \left\lfloor \frac{UB}{p_1} \right\rfloor, r_1 \right\}$$

$$k_j = \min \left\{ \left\lfloor \frac{UB - \sum_{h=1}^{j-1} p_h\, k_h}{p_j} \right\rfloor, r_j \right\},\ j = 2, ....., n.$$

Therefore:

$k_1 = \min\left\{\left\lfloor \frac{11}{5} \right\rfloor, 2\right\} \rightarrow k_1 = \min\{\lfloor 2.2 \rfloor, 2\} \rightarrow k_1 = 2$

$k_2 = \min\left\{\left\lfloor \frac{UB - \sum_{h=1}^{2-1} p_h k_h}{p_2} \right\rfloor, r_2\right\}, j = 2 \rightarrow$

$k_2 = \min\left\{\left\lfloor \frac{11 - (p_1 k_1)}{4} \right\rfloor, 2\right\}, j = 2$

$k_2 = \min\left\{\left\lfloor \frac{11 - (5 \times 2)}{4} \right\rfloor, 2\right\} \rightarrow k_2 = \min\{\lfloor 0.25 \rfloor, 2\} \rightarrow k_2 = 0$

$k_3 = \min\left\{\left\lfloor \frac{UB - \sum_{h=1}^{3-1} p_h k_h}{p_3} \right\rfloor, r_3\right\}, j = 3 \rightarrow$

$k_3 = \min\left\{\left\lfloor \frac{11 - (5 \times 2 + 4 \times 0)}{3} \right\rfloor, 3\right\}, j = 3$

$k_3 = \min\{\lfloor 0.33 \rfloor, 3\}, j = 3 \rightarrow k_3 = 0$ Thus the load on machine(1) is ($C_1$):

M1 : $C_1 = p_1 k_1 + p_2 k_2 + p_3 k_3 \rightarrow C_1 = 5 \times 2 + 4 \times 0 + 3 \times 0 \rightarrow C_1 = 10$

Step 4:

$LB \le \sum_{j=1}^{n} p_j k_j \le UB$ ……. (3-10) $\rightarrow 5 \le \sum_{j=1}^{n} p_j k_j \le 11$

Where the load on machine(1) is: $C_1 = \sum_{j=1}^{3} p_j k_j = 10$

then the machine(1) satisfy inequality (10)

$i = i + 1 \rightarrow i = 1 + 1 \rightarrow i = 2 \neq m$ . then the next step is (3).

Step 3: load machine (2) according to the equations (11)&(12). With the same procedure for loading machine (1). The load on machine (2) is($C_2$) as follows:

M2 : $C_2 = p_1 k_1 + p_2 k_2 + p_3 k_3 \rightarrow C_2 = 5 \times 0 + 4 \times 2 + 3 \times 1 \rightarrow C_2 = 11$

Where the load on machine (2) is: $C_2 = \sum_{j=1}^{3} p_j k_j = 11$

then the machine 2 satisfy inequality (10).

$i = i + 1 \rightarrow i = 2 + 1 \rightarrow i = 3$

The load on machine(3) is ($C_3$):

M3 : $C_3 = p_1 k_1 + p_2 k_2 + p_3 k_3 \rightarrow C_3 = 5 \times 0 + 4 \times 0 + 3 \times 2 \rightarrow C_3 = 6$

Step 4:

$LB \le \sum_{j=1}^{n} p_j k_j \le UB$ ……. (3-10) $\rightarrow 5 \le \sum_{j=1}^{n} p_j k_j \le 11$

Where the load on machine (3) is: $C_3 = \sum_{j=1}^{3} p_j k_j = 6$

Then the machine (3) satisfy inequality (10).

Step 5: $i = m = 3$ then Calculate $C_{max}$ by using equation (7).

$C_{max} = max\{C_i : i = 1, ....., m\}$ (7)

$C_{max} = max\{C_i : i = 1, 2, 3\}$

$C_{max} = max\{C_1, C_2, C_3\} \rightarrow C_{max} = max\{10, 11, 6\} \rightarrow C_{max} = 11$

($UB = C_{\max} - 1$) and ($t = t + 1$).

$UB = C_{\max} - 1 \rightarrow UB = 11 - 1 \rightarrow UB = 10$ and $t = t + 1 \rightarrow t = 2$

This means: the new upper bound is $UB = 10$ and the next iteration is $t = 2$

And the new lower bound ($LB$)can be found by equation (9): $LB = 7$

With the same above procedure the new load on machines are:

$C_1 = 10$

$C_2 = 8$

$C_3 = 9$

Calculate $C_{max}$ by using equation (3-7).

$C_{max} = max\{10, 8, 9\} \rightarrow C_{max} = 10$

$UB = C_{\max} - 1 \rightarrow UB = 10 - 1 \rightarrow UB = 9$ and $t = t + 1 \rightarrow t = 3$

This means: the new upper bound is $UB = 9$ and the next iteration is $t = 3$

And the new lower bound ($LB$) can be found by equation (9): $LB = 9$

the new load on machines are:

$C_1 = 9$

$C_2 = 9$

$C_3 = 9$

$i = 3$ , $m = 3 \rightarrow i = m$

Calculate $C_{max}$ by using equation (7).

$C_{max} = max\{C_1, C_2, C_3\}$

$C_{max} = max\{9, 9, 9\} \rightarrow C_{max} = 9$

($UB = C_{\max} - 1$) and ($t = t + 1$).

$UB = C_{\max} - 1$

$UB = 9 - 1 \rightarrow UB = 8$ and $t = t + 1 \rightarrow t = 4$

This means: the new upper bound is $UB = 8$ and the next iteration is $t = 4$

Then the new lower bound ($LB$) can be found by equation (9): $LB = 11$

the new load on machines are:

$C_1 = 8$ ; $C_2 = 8$ , $C_3 = 8$

Where the load on machines does not satisfy inequality (10). Then go to step (6)

Step 6: set $C^*_{\max} = UB + 1$

$C^*_{\max} = 8 + 1 \rightarrow C^*_{\max} = 9$

The optimal solution $C^*_{\max} = 9$ .

Figure (4) shows Gantt chart for the solution and the load on machines .

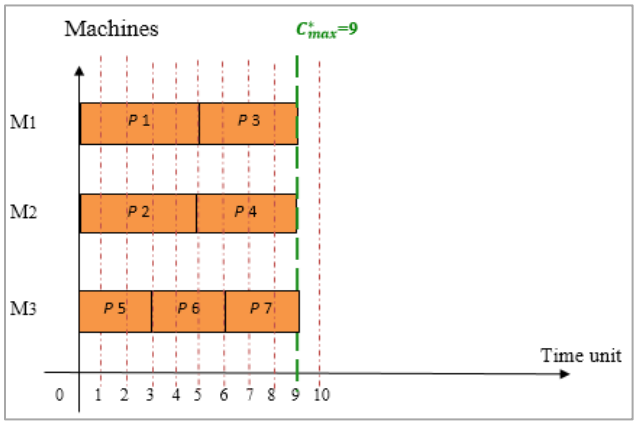


Fig. (4). The optimal solution (schedule) for Case (1).

Improving the initial solution to obtain the optimal solution (schedule) of the problem shown in figure (5).

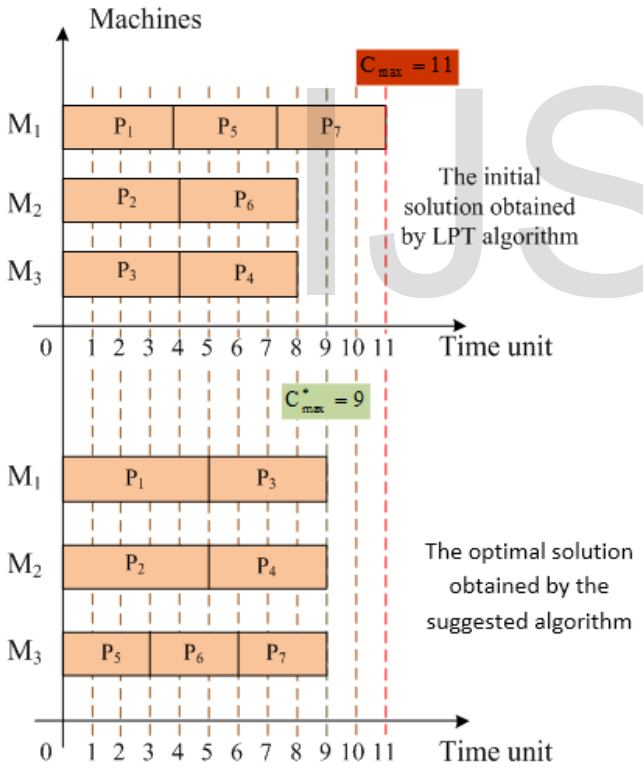


Fig. (5). Using the suggested algorithm to Improve the initial solution.

According to the results in table (4-1) the solution of the problem ($C^*_{max}$) which obtained by LPT algorithm ($C^*_{max}=11$).

TABLE 1

The solution for case(1) by (LINGO, LPT and the suggested algorithm)

| LINGO | | | | |
|---|---|---|---|---|
| Machine | Jobs | C | $C^*_{max}$ | Iteration |
| M1 | P5 , P6 , P7 | 9 | | |
| M2 | P2 , P4 | 9 | **9** | 21 |
| M3 | P1 , P4 | 9 | | |
| LPT algorithm | | | | |
| Machine | Jobs | C | $C^*_{max}$ | Iteration |
| M1 | P1 , P5 , P7 | 11 | | |
| M2 | P2 , P6 | 8 | **11** | 1 |
| M3 | P3 , P4 | 8 | | |
| the suggested algorithm | | | | |
| Machine | Jobs | C | $C^*_{max}$ | Iteration |
| M1 | P5 , P6 , P7 | 9 | | |
| M2 | P2 , P4 | 9 | **9** | 4 |
| M3 | P1 , P4 | 9 | | |

And as can be observed the solution by LINGO and the algorithm program is identical ($C^*_{max}=9$) with different distribution of jobs on machines and number of solution iterations.

**4-4 Problems with random number of machines and jobs**

Number of problems with random number of machines and jobs have been generated to ensure and demonstrate effectiveness of the algorithm. Table (4-3) represents the results.

TABLE 2

The solution for number of (m) Machines and (N) Jobs

| No | Mach-ines | Jobs | $C^*_{max}$ | | Iteration | | |
|---|---|---|---|---|---|---|---|
| | (m) | (N) | LINGO | LPT | Alg. | LINGO | Alg. |
| 1 | 2 | 8 | 42 | 42 | 42 | 17 | 2 |
| 2 | 3 | 11 | 26 | 28 | 26 | 72 | 5 |
| 3 | 4 | 9 | 12 | 15 | 12 | 88 | 5 |
| 4 | 5 | 10 | 25 | 25 | 25 | 260 | 2 |

* Alg.: the suggested algorithm

Table (2) represents the results for random number of machines and jobs, the makespan ($C^*_{max}$) which obtained by LINGO and the algorithm for problem 1 and 2 in the table are equal. On the other hand there is a big difference, in the iterations to reach the solution, the algorithm is always faster than LINGO.

**6 Conclusion:**

In this study the ILP model introduced to represent and formulate the problem( $P_m||C_{\max}$ ), U-1 algorithm designed and coded in MATLAB to solve the ILP model, also LINGO software has been used to test and evaluate performance of the suggested algorithm.

Many experiments and problems are used with random number of machines (m) and jobs (N) to demonstrate effectiveness of the algorithm solution.

In this study, the ILP model can optimally be solved by ILP solvers for small size of problems. And as can be observed from tables (1) and (2) , LINGO can solve the problems with small size of number of machines and jobs. While the suggested algorithm can solve large sizes of parallel machines scheduling problem optimally in small number of iteration.

The results of the study demonstrated that:

1- Efficiency and performance of the algorithm is evaluated by the number of iterations required to find the optimal solution. This good performance can be observed by varying the number of machines.

2- The number of feasible solution(s) which are generated is equal to number of iteration required to find optimal solution minus one (*t*-1) and each one of them is better than the previous one.

3- If the number of iteration (*t*=1) the initial solution obtained by LPT algorithm is optimal. It was found that computations required by LPT will escalate considerably as the number of machines increases. However, its capability is limited in terms of solving a large-scale problem.



## REFERENCES

[1 ] C. Koulamas and G. J. Kyparisis "A modified LPT Algorithm for the Two Uniform Parallel Machine," *European Journal of Operational Research*, vol.196, pp.61–68, 2009.

[2] T. Gonzalez, T, O.H , Ibarra and S. Sahni " Bounds for LPT Schedules on Uniform Processors", *SIAM Journal on Computing* vol. 6, pp. 155–166.

[3] I. Chaudhry, S. Mahmood and R. Ahmad "Minimizing Makespan for Machine Scheduling and Worker Assignment Problem in Identical Parallel Machine Models Using GA", Proc. of the World Congress on Engineering, WCE 2010, June 30 - July 2, 2010, London, U.K

[4] M. Queiroz and A. Mendes" Strategies for Scheduling Jobs in an Identical Parallel Machine Production Environment, *African Journal of Business Management,* Vol. 7(16), pp. 1553-1559, 28 April, 2013.

[5] N. Hashemian" Makespan "Minimization for Parallel Machines Scheduling with Availability Constraints", MSc dissertation, Dept. Industrial Engineering, Dalhousie Uni.,2010.

[6] B. V Reghavendra and A. N. Murthy "Workload Balancing in Identical Parallel Machines Scheduling Using Genetic Algorithm", paper, *ARPN Journal of Engineering and Applied Sciences*. Vol.6, No.1, 2011.

[7] N. Hashemian" Makespan "Minimization for Parallel Machines Scheduling with Availability Constraints", MSc dissertation, Dept. Industrial Engineering, Dalhousie Uni., 2010.

[8] C. Koulamas and G. J. Kyparisis "A modified LPT Algorithm for the Two Uniform Parallel Machine," *European Journal of Operational Research*, vol.196, pp.61–68, 2009.

[9] C. H. Lin and C. Liao "Minimizing Makespan on Parallel Machines with Machine Eligibility Restrictions", paper, *The Open Operational Research Journal*, 2, pp. 18-24, 2008.

[10] S. Kaya, "Parallel Machine Scheduling Subject to Machine Availability Constraints", MSc Thesis, Bilkent Uni., 2006.

[11] G. ND. Jatinder , J. HO, and A. Ruiz-Torres "Makespan Minimization on Identical Parallel Machines Subject to Mminimum Total Flow-Time." *Journal of the Chinese Institute of Industrial Engineers* Vol. 21. No.3, pp. 220-229, 2004.

[12] M. L. Pinedo, "Scheduling (Theory, Algorithms and Systems)", New York University- USA, Fourth Edition, 2011.

[13] K. Jihene and Y. Harrath "A Survey of Parallel Machine Scheduling Under Availability Constraints", paper, *International Journal of Computer and Information Technology*, Vol. 3, issue 02, 2014.

[14] I. Chaudhry, S. Mahmood and R. Ahmad "Minimizing Makespan for Machine Scheduling and Worker Assignment Problem in Identical Parallel Machine Models Using GA", Proc. of the World Congress on Engineering, WCE 2010, June 30 - July 2, 2010, London, U.K.

[15] M. L. Pinedo, "Planning and scheduling in manufacturing and services", New York University- USA, 2005.