SENRA Academic Publishers, British Columbia
Vol. 8, No. 3, pp. 3147-3153, October 2014
Online ISSN: 1920-3853; Print ISSN: 1715-9997

Canadian Journal of pure&applied sciences

A COMPARATIVE STUDY OF HEURISTIC AND METAHEURISTIC FOR THREE IDENTICAL PARALLEL MACHINES

*Omar Selt[1] and Rachid Zitouni[2]
[1]Department of Mathematics, University of M'sila, Algeria
[2]Laboratory of Fundamental and Numerical Mathematics, University of Setif 1, Algeria

ABSTRACT

In this paper, we propose a comparative study between metaheuristic and a new heuristic for solving scheduling problems of *n* tasks on *m* identical parallel machines with unavailability periods. This problem is NP-complete in the strong sense of the expression and finding an optimal solution appears unlikely. In this frame, we suggested a new heuristic in which availability periods of each machine are filled with the highest weighted tasks. To improve the performance of this heuristic, we used three diversification strategies (T_1, T_2 and T_3) with the aim of exploring unvisited regions of the solution space and two well-known neighborhoods (neighborhood by swapping and neighborhood by insertion). The computational experiment was carried out on three identical parallel machines with different availability periods. It must be noted that tasks movement can be within one machine or between different machines. Note that all data in this problem are integer and deterministic. The weighted sum of the end dates of tasks constitutes the optimization performance criterion in the problem treated in this paper.

Keywords: Scheduling, parallel identical machines, unavailability periods, metaheuristic, tabu search.

INTRODUCTION

A scheduling problem consists in organizing tasks realization time with consideration of time constraints (time limits, tasks series character) and constraints related to using and availability of required resources. The scheduling constitutes a solution to the considered problem, describes the tasks execution and resources location during time and aims to satisfy one or many objectives.

A scheduling problem under machines availability constraints has been studied by many authors. For example $P_m // N-\boldsymbol{C} // C_{\max}$ has been studied by Lee (1996, 1997, 1999), Schmidt (2000) and Yun-Chia *et al*. (2013). The tabu search is a metaheuristic originally developed by Glover (1986), Glover and Hanafi (2002) and independently by Hancen (1986).

This method combines a local search procedure with a certain number of rules and mechanism which allows surmounting the obstacle of local optima without cycling. Toward furthermore, it proved hight efficiently in resolution of the problems NP-complet and approximate more the optimal solution.

The scheduling problem of a single machine with minimization of the weighted sum of the end dates of tasks. without unavailability constraint is optimally resolved by using the WSPT (weighted shortest processing time) rules. The case of several machines is studied by many authors like Belouadeh *et al*. (1992), Sadfi (2002) and Haouari and Ladhari (2003).

In 1984, Schmidt has studied the scheduling problem of parallel identical machines with different unavailability intervals and different tasks deadlines. He used the method of Branch and Bound based on two procedures: the first is the generation by decomposition and cut approach and the second is the hybridization of procedures of generation by cut. He also built an admissible preemptive scheduling of a complexity $O(n/m \ln n)$ where n is the number of tasks and m is the number of machines. Lee (1999) have studied the simultaneous scheduling of production works and maintenance activities in parallel identical machines to minimize the weighted sum of the end dates of tasks. They have studied two cases: The first, with sufficient number of resources, concerns the case where several machines can be checked up simultaneously (overlap of unavailability periods). The second case, with insufficient number of resources, concerns the case where only one machine can be checked up (overlap of unavailability periods not allowed). They could demonstrate that even if all tasks have the same weight, the problem is NP-hard. They proposed the method of Branch and Bound based on the approach of columns generation to solve the two cases. They have published an experimental study on average size instances.

Zribi *et al*. (2005) have studied the problem

*Corresponding author email: selt.omar@yahoo.fr

$1//N-\boldsymbol{C}//\sum_{j=1}^{n} w_j C_j$ and have compared two exact methods: one is the Branch and Bound, the other is the integer programming.They have concluded that Branch and Bound method have better performance and it allowed resolving instances of more than 1000 tasks.

Another study Adamu and Adewunmi (2012) have studied the problem $P_m//\sum_{j=1}^{n} w_j\left(U_j+V_j\right)$, they proposed some metaheuristics for scheduling on parallel identical machines to minimize weighted number of early and tardy jobs.

In 2013, they carried out a comparative study of different (a genetic algorithm, particle swarm optimization and simulated annealing with their hybrids) metaheuristics for identical machines. In this paper, the results of Adamu and Adewunmi (2012) research works are exploited to develop a different new metaheuristic to solve the scheduling problem under different constraints.

PROBLEM STATEMENT

This problem consists in scheduling n tasks for m parallel identical machines $\{M_1, M_2, ..., M_m\}$ where $n >> m \geq 2$, with unavailability periods.

We assume that the tasks $\{j_1, j_2, ..., j_n\}$ are all available at $t=0$ and their operation times are independent from the choice of machines performing these tasks. In the generic case of the problem, each one of the m machines can show some unavailability periods during scheduling horizon and each task must be executed onetime.

This problem noted by $P_m//N-\boldsymbol{C}//\sum_{j=1}^{n} w_j C_j$ consists in assigning n tasks to m machines over availability intervals in a manner to enforce the weighted sum of the end dates of tasks. referred to as $\sum_{j=1}^{n} w_j C_j$ to be minimal.

It must be noted that there is $(n!)^m$ possibility to assign n tasks to m machines (Sakarovitch, 1984).

NEIGHBORHOOD STRUCTURE

Neighborhood determination constitutes the most important stage in metaheurstic methods elaboration. In the following part, we use two Neighborhoods, (neighborhood by swapping) and (neighborhood by insertion).

It must be noted that tasks movement can be within one machine or between different machines.

NEIGHBORHOOD BY SWAPPING

Definition. *Consider a sequence σ composed of n tasks. A neighborhood σ' is obtained by permuting two tasks. j and j' of respectively k and k' positions σ with $k' = k+1, k+2, ..., n$.*

The set :

$N_1(\sigma) = \{\sigma, \sigma \text{ is obtained by permutation of two tasks}\}$

is called neighborhood of σ This set is consequently obtained by permutation of all tasks of σ two by two.

Proposition. *Consider a sequence σ, the set's cardinal of $N_1(\sigma)$ is $\frac{n(n-1)}{2}$.*

Proof . The permutation of all tasks. two by two consists in permuting each task of the sequence with all remained tasks. without identical ones. The number of possible permutations in a sequence σ composed of n tasks. is : $(n-1)+(n-2)+..+2+1=\frac{n(n-1)}{2}$.

NEIGHBORHOOD BY INSERTION

Definition. *Consider a sequence σ composed of n tasks.*

A neighborhood σ' is obtained by inserting one task j of a position k in a new position k' in the sequence σ. The set

$N_2(\sigma) = \{\sigma', \sigma' \text{ is obtained by inserting a task of position } k \text{ in } k'\}$

is a neighborhood of σ. This set is consequently obtained by realizing all possible insertions of all tasks of σ.

Proposition. *Consider a sequence σ, the set's cardinal of $N_2(\sigma)$ is $(n-1)^2$.*

Proof . Inserting a task j of position k in an other position k' in the sequence σ allows getting $n-1$ possible insertions. Hence, for n tasks, there is $n(n-1)$ insertions to be done. To avoid getting identical sequences, adjacent tasks insertions are counted once. Consequently $n-1$ insertions will be deleted. Finally, the number of obtained insertions is: $n(n-1)-(n-1)=(n-1)^2$.

TABU LIST STRUCTURE

The tabu method is based on the principle that consists in maintaining in memory the last visited solutions and in forbidding the return to them for a certain number of iterations. The aim is to provide sufficient time to the algorithm so it can leave the local optimum. In other words, the tabu method conserves in each stage a list L of solutions (Tabu's) which it is forbidden to pass-by temporarily. The necessary space for saving a set of solutions tabus in the memory is indispensable.

The list, that we propose, contains the found solutions sequences. After many tests, a dynamic size list, which varies according to the search amelioration state, is conceived. The initial size of this list is considered to be $\frac{3\sqrt{n}}{2}$ where n is the tasks number. After that, during the search, when 5 successive iterations pass without amelioration of solution, the list is reduced to a number inferior or equal to $\sqrt{n}$. On the other hand, when 5 successive iterations pass and the solution is ameliorated, the list is increased to a number superior or equal to $2\sqrt{n}$. The Tabu list is consequently dynamic and its size varies within the interval $[\sqrt{n}, 2\sqrt{n}]$. The decrease or the increase of list size must always be done at the end of the list.

HEURISTIC FOR THE PROBLEM (P).

An initial solution is always necessary. For this reason, we suggest in this part the following heuristic: assigne the (best) task h where $\left(\frac{p_h}{w_h} = \min_{j \in J}\left\{\frac{p_j}{wj}\right\}\right)$ to the best machine (the most available[1]) based on two principles justified by the two following propositions :

Proposition 3. In an optimal scheduling, it is necessary to schedule the tasks. in each availability period of the machine according to the order SWPT.

Proof . It results directly by adjacent task exchange like used by Smith (1956) for the corresponding periods.

Proposition 4. It is not useful to let the machine (idle) if a task can be assigned to this machine.

Notations:

We denote by :

$J = \{1, 2, ..., n\}$: The set of tasks.

p_h : Execution time of the task h.

$p_h^{(i)}$: Execution time of the task h assigned to the machine $i \in I$.

$I = \{1, 2, ..., m\}$: The set of machines

I_{NA} : The set of non-assigned machines.

α : Number of availability zones.

$Z = \{1, 2, ..., \alpha\}$: Availability zones.

$S_z^{(i)}$ ($z \in Z$) : The beginning of the unavailability time of the machine $i \in I$.

T_F : Final time.

$T_z^{(i)}$ $(z \in Z)$: The end of the unavailability time of the machine $i \in I$.

$\sigma_z^{(i)}(z \in Z)$: The set of partial sequences assigned to the machine $i \in I$.

$\sigma_z = \sigma_z^{(1)} \cup \sigma_z^{(2)} \cup ... \cup \sigma_z^{(m)}$.

$$J_z^{(i)}(z \in Z) = \left\{ \begin{matrix} j \,/\, j \text{ task assigned to the machine } i \\ \text{with } T_z^{(i)} \le C_j^{(i)} \le S_z^{(i)} \end{matrix} \right\}.$$

$C_z^{(i)}$ $(z \in Z)$: Execution time of the task $j \in J_z^{(i)}$.

We assume that for each task $h \in J$, there is at least one machine $i \in I$ such that $P_h \le S_z^{(i)} - T_z^{(i)}$.

FORMULATIONS MATHEMATICS

Consider $I = \{1, 2, ..., m\}$, $J = \{1, 2, ..., n\}$,
$\sigma = (\sigma(1), \sigma(2), ..., \sigma(n))$ a sequence of n tasks. and **P** the set of all possible sequences permutations.
This problem is formulated as an integer linear programming model:

$$f = \min_{\sigma \in \mathbf{P}} \sum_{j=1}^{n} w_{\sigma(j)} C_{\sigma(j)}$$

$$(P) \quad \sum_{j \in \sigma_{(j)}^{(i)}} p_{\sigma(j)}^{(i)} \le S_z^{(i)} \quad , \tag{1}$$

$$S_z^{(i)} \in \mathrm{N};\ z = 1, 2, ..., \alpha;\ \sigma(j) = \sigma_{(j)}^{(1)} \cup \sigma_{(j)}^{(2)} \cup ... \cup \sigma_{(j)}^{(i)}$$

$$C_{\sigma(j)}^{(i)} = \sum_{j \in \sigma_{(j)}^{(i)}} p_{\sigma(j)}^{(i)} + T_z^{(i)} \quad , T_z^{(i)} \in \mathrm{N}; \tag{2}$$

$$i = 1, 2, ..., m;\ j = 1, 2, ..., n \tag{3}$$

$$\sum_{\substack{j=1,2,..,n \\ z=1,2,..,\alpha}} X_{j_z} = 1 \text{ and } X_{j_z} \in \{0, 1\}$$

ALGORITHM

Initialization

$J = \{1, 2, ..., n\}$, $I = \{1, 2, ..., m\}$; $Z = \{1, 2, ..., \alpha\}$; $I_{NA} = I$;

$S_\alpha^{(i)} = T_F$ (given), $\sigma = \phi$, $f_\sigma = 0$, $z = 1, C_z^{(i)} = 0$ and $T_1^{(i)} = 0$.

Sort task $h \in J$ in increasing order according to the criterion p_h / w_h in a list L_1

Sort task $h \in J$ in increasing order according to the criterion p_h in a list L_2

While

$(L_1 \neq \phi \text{ and } z \le \alpha)$ **do**

Begin

[1] A machine is supposed to be the most available if it has an availability period the most close to t=0 and it is able to realize the required task.

Set $p_{h_1} = p_h / w_h$ from the top list of L_1.
$p_{h_2} = p_h$ from the top list of L_2.

Determine the machine $k \in I_{NA}$ and the task $h \in J$ such that
$S_z^{(k)} - C_z^{(k)} = \max_{i \in I_{NA}} \{S_z^{(i)} - C_z^{(i)}\} >= \min(p_{h_1}, p_{h_2})$
If $\{k\} = \phi$ **then**

Determine the machine $k \in I$ and the task $h \in J$ such that
$S_z^{(k)} - C_z^{(k)} = \max_{i \in I} \{S_z^{(i)} - C_z^{(i)}\} >= \min(p_{h_1}, p_{h_2})$

Endif
If $\{k\} \neq \phi$ **then**
Begin
Assigned the task h to the machine k
Delete the task h from the two lists L_1 and L_2
Compute $C_z^{(k)} = \sum_{j \in J_z^{(k)}} p_j + T_z^{(k)}$;
Determine $\sigma_z^{(k)} = \sigma_z^{(k)} \cup \{h\}$ and $f_\sigma = f_\sigma + w_h C_z^{(k)}$;
Set $I_{NA} = I_{NA} \setminus \{k\}$
End
Else
Begin
Set $z = z + 1$; $I_{NA} = I$;
End
Endif
End

COMPUTATIONAL ANALYSIS

Data generation

The heuristic were tested on problems generated with 500 tasks similar to that used in previous studies (Adamu and Abass, 2010; Ho and Chang, 1995; M'Hallah and Bulfin, 2005) for each task j an integer processing time p_j was randomly generated in the interval $(1,99)$ with a weight randomly w_j chosen in interval $(1,10)$.

The search time to define a neighborhood and to determine minimal cost is chosen equal to $90s$.

The number of machines fixed (3 machines) with (3 availability zones for each machine).

Diversification strategies

The final time to execute this problem is chosen as $T_F = 1200s$. It is divided according to diversification strategy to three times T_1, T_2 and T_3. After many experiments, these periods are chosen as follows:

$T_1 = 700s$ Initial starting time: uses long term memory to store the frequency of the moves executed throught of the search.

$T_2 = 300s$ First restarting time makes use of influential moves.

$T_3 = 200s$ Second restarting time also makes use of influential moves.

The table 1 below presents:

1- The initial mean values of objective function corresponding to initial sequence.
2- The initial mean values of objective function obtained by using on one hand, the neighborhood by swapping and on the other hand, the neighborhood by insertion.
3- The average times corresponding to the two neighborhoods.
4- The percentage of cost improvement.
5- The best costs.

RESULTS

The results listed in table 1 shows clearly that the tabu method based on neighborhood by insertion presents best costs compared with tabu method based on neighborhood by swapping. This is due to the fact that the first neighborhood ensures a faster tasks movement besides that the search space is richer with optimals partials sequences in each availability periods. This can also be explained by the nature of used neighborhoods, besides the left shifting of other tasks in the swapping neighborhood. The results show that execution time obtained by the first neighborhood is acceptable.

On the other hand, the heuristic amelioration rate between the two neighborhoods is remarkable (Figs. 1 and 2). It is also noted that the cost amelioration rate of the proposed tabu search heuristic is situated between 0.4% and 14%.

RECAPITULATED TABLE

In table 1 below one use the following abbreviations:
AC: Average costs
AT: Average time
PIIC: Percentage of improvement of the initial costs

Table 1. Percentage of heuristic cost amelioration based on metaheuristic.

n	Initial cost by heuristic (AC of 5 instances)	Tabu search by Swapping			Tabu search by Insertion			Best Cost
		AC	AT (second)	PIIC	AC	AT (second)	PIIC	
50	46290	41142	131	11%	41012	81	12%	41012
	54046	50550	179	6%	50444	139	7%	50444
	44648	43134	146	3%	43030	107	4%	43030
100	198110	179808	197	9%	179640	252	10%	179640
	176650	169064	314	4%	168868	480	5%	168868
	202408	186198	256	8%	195082	418	4%	186198
200	735058	633146	431	14%	733038	403	0.5%	633146
	692042	688068	327	1%	688962	498	1%	688068
	700578	699438	535	0.4%	699360	434	3.8%	699360
400	2773407	2771998	255	0.5%	2671898	583	4%	2671898
	2807046	2693834	325	6%	2793712	298	5%	2693834
	2756238	2638554	470	4%	2628452	428	5%	2628450
500	4289054	4273518	249	1%	4273400	378	1%	4273400
	4422156	4386046	360	1%	4384956	354	1%	4384956
	4649564	4497934	486	4%	4497822	707	4%	4497822

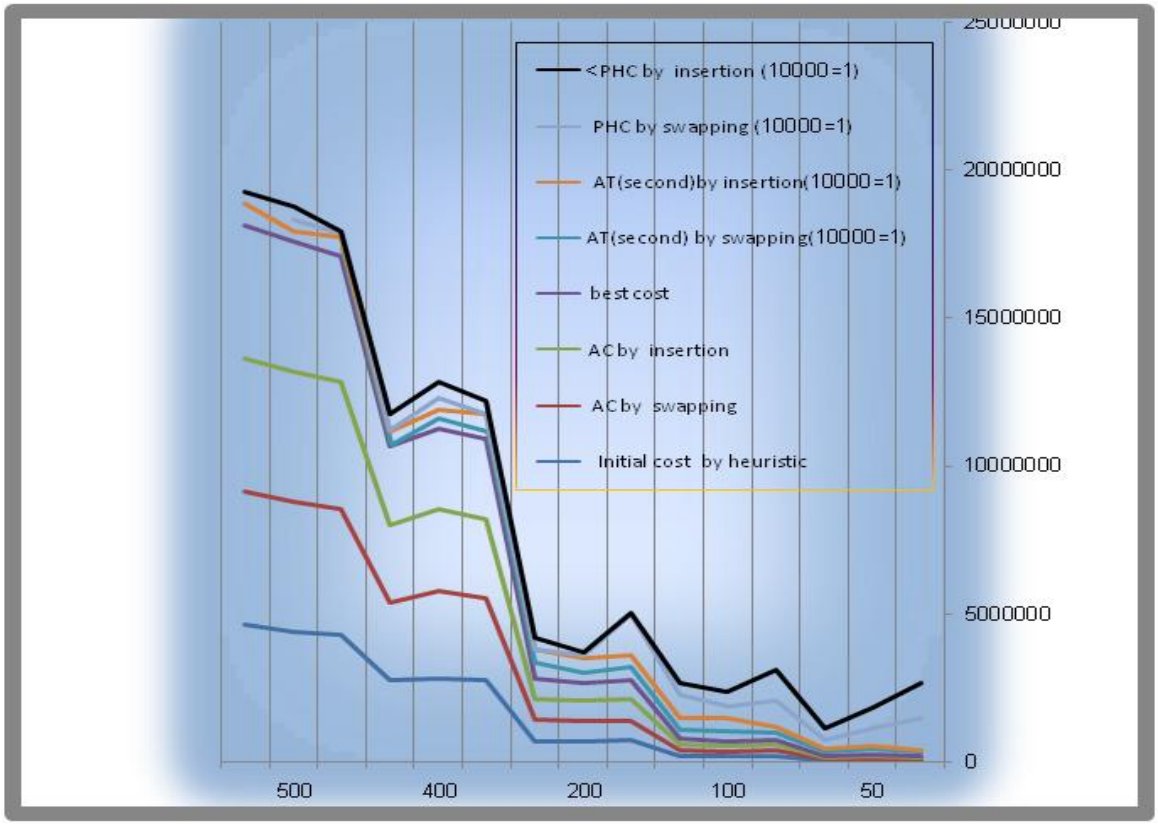

Fig 1. Comparison of heuristic and metaheuristic for n=500.

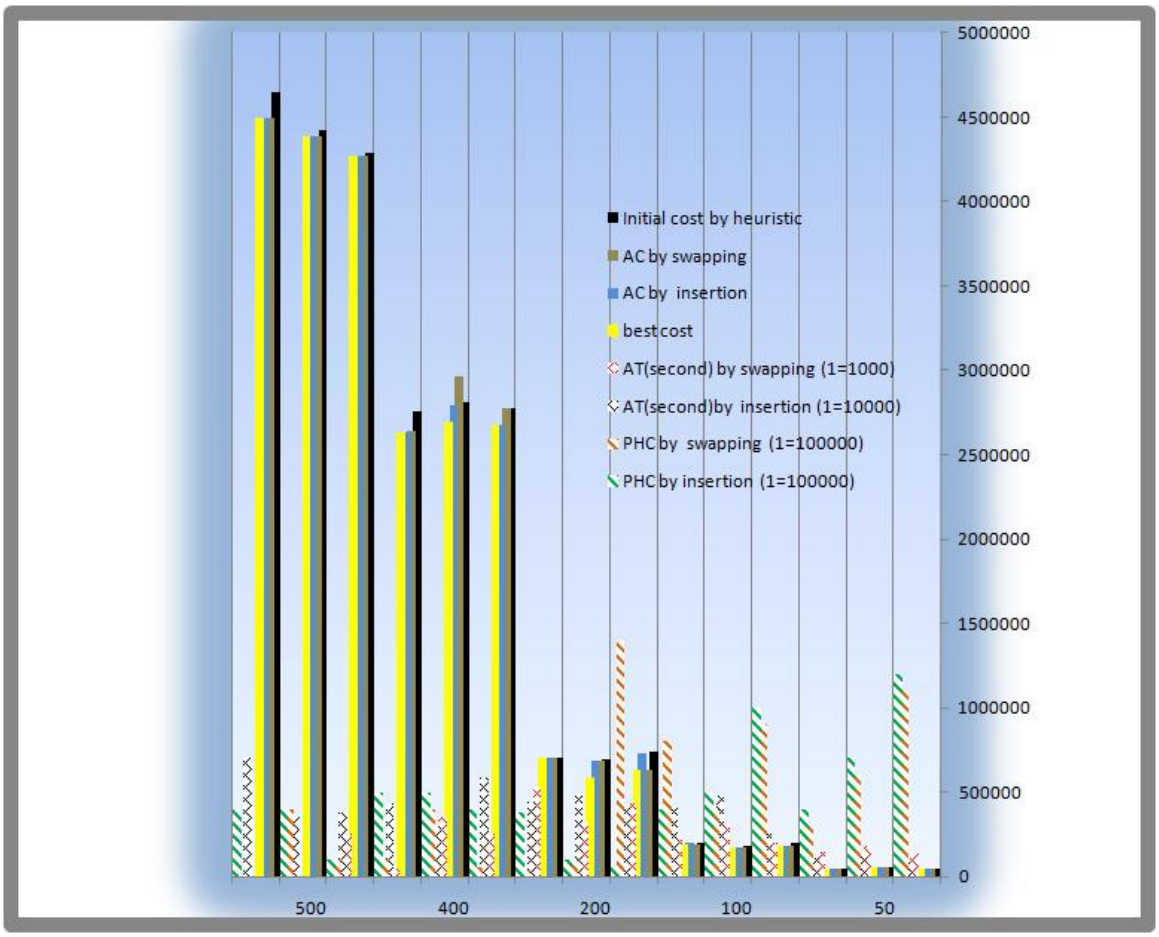

Fig 2. Comparison of heuristic and metaheuristic for n=500.

CONCLUSION

In this paper, a metaheuristic polynomial approach (Tabu search) as solution for tasks scheduling problem with parallel identical machines and availability periods is presented. The developed approach uses a diversification technique based on search restarting from the point of the solution that was chosen among the earlier best unmaintained found solutions, by considering that the tabu list is dynamic and its size varies according to amelioration state of the solution. According to the curried out tests, it can be concluded that the proposed approach ensure better results (heuristic amelioration cost up to14%). It must be noted that the neighborhood by insertion presents the best costs with an acceptable execution time.

ACKNOWLEDGEMENT

My special thanks to Professor Hocine Belouadah, Mohamed Said Hamani and Tarak Benslimane for their advises to achieved this work.

REFERENCES

Adamu, MO. and Abass, O. 2010. Paralell machine sheduling to maximize the weighted number of just-in-time jobs. J. App. Sci. Technol. 15(1&2):12-34.

Adamu, MO. and Adewunmi, A. 2013. Comparative study of metaheuristics for identical parallel machines. J. Eng.Technol. Res. 5(7):207-216.

Adamu, MO. and Adewunmi, A. 2012. Metaheuristics for scheduling on parallel machine to minimize weighted number of early and tardy jobs. Int. J. Phys. Sci. 7(10): 1641-1652.

Belouadah, H., Posner, ME. and Potts, CN. 1992. Scheduling with relates dates on a single machine to minimize total weighted completion time. Discrete Appl. Math. 36:213-231.

Glover, F. and Hanafi, S. 2002. Tabu Search and Finite Convergence, Special Issue on Foundations of heuristics in Combinatorial Optimization. Discrete Appl. Math. 119:3-36.

Glover, F. 1986. Futurepathsfo rinteger programming and

links to artificial intelligence, Comput. Open Res. 13: 533-549.

Hansen, P. 1986. The steepest ascent mildest descent heuristic for combinatorial programming. In: Proceedings of the Congress on Numerical Methods.

Haouari, M. and Ladhari, T. 2003. Branch and bound-based local search method for the flow shop problème. J. Oper. Res. Soc. 54:1076-1084.

Ho, JC. and Chang, YL. 1995. Minimizing the number of tardy jobs for m paralell machines. Eur. J. Oper. Res. 84: 334-355.

Lee, CY. 1996. Machine scheduling with an availability constraints. J. Global Optim. 9:395-416.

Lee, CY. 1997. Minimising the makespan in two machines flow shop scheduling problem with availability constraints. Oper. Res. Lett. 20:129-139.

Lee, CY. 1999. Two machines flow shop scheduling problem with availability constraints. European J. Oper. Res. 114:420-429.

M'Hallah, R. and Bulfin, RL. 2005. Minimizing the weighted number of tardy jobs of paralell processors. Eur. J. Oper. Res.160 :471-484.

Sadfi, C. 2002. Problèmes d'ordonnancement avec minimisation des encours Thèse PhD. Institut National Polytechnique de Grenoble, France.

Schmidt, G. 2000. Scheduling with limited machine availability. European J. Oper. Res. 121:1-15.

Smith, WE. 1956. Various optimizes for single-stage production. Naval Res. Logist. 3:59-66.

Yun-Chia, L., Yu-Ming, H. and Chia-Yun, T. 2013 Metaheuristics for drilling operation scheduling in Taiwan PCB industries. Int. J. Prod. Econ. 141(1):189-198.

Zribi, N., Kacem, I., El-Kamel, A. and Borne. P. 2005. Minimisation de la somme des retards dans un jobshop flexible. Revue e-STA (SEE). 2(2):2.

Received: April 11, 2014; Revised and Accepted: Sept 11, 2014