INDEPENDENT JOURNAL OF MANAGEMENT & PRODUCTION (IJM&P)
http://www.ijmp.jor.br _v. 12, n. 7, September-October 2021_
ISSN: 2236-269X
DOI: 10.14807/ijmp.v12i7.1029

PERFORMANCE ENHANCEMENT OF NUMERICAL APPROACHES FOR SCHEDULING PROBLEM ON MACHINE SINGLE

Omar Selt
Laboratory of pure and applied mathematics,msila university, algeria
E-mail: selt.omar@yahoo.fr

Submission: 3/28/2019
Revision: 10/10/2020
Accept: 10/17/2020

ABSTRACT

In this paper, we consider a single-machine scheduling problem, with the aim of minimizing the weighted sum of the completion time. This problem is NP-hard, making the search for an optimal solution very difficult. In this frame, two heuristics (H_1), (H_2) and metaheuristic tabu search are suggested.

To improve the performance of this techniques, we used, on one hand, different diversification strategies (TES_1 and TES_2) with the aim of exploring unvisited regions of the solution space. On the other hand, we suggested three types of neighborhoods (neighborhood by swapping, neighborhood by insertion and neighborhood by blocks).It must be noted that tasks movement can be within one period or between different periods.

Keywords: Scheduling; Single machine; NP-hard; Tabu search

1. INTRODUCTION

The scheduling problem of a single machine with minimization of the weighted sum of the tasks' end-dates, 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 (Zribi et al., 2005; Zitouni& Selt ,2016).

Zribi et al., (2005) have studied the problem $1//N-\boldsymbol{C}//\sum_{j=1}^{n} w_j C_j$ and have compared two exact methods, the Branch and Bound method and the integer programming one. They have concluded that Branch and Bound method has better performance and it allows resolving instances of more than 1000 tasks. Chang et al. (2011) proposed a genetic algorithm (GA) enhanced by dominance properties for single machine scheduling problems to minimize the sum of the job’s setups and the cost of tardy or early jobs related to the common due date.

Selt and zitouni (2014) have studied the following problem ($P_m // \mathrm{N}-C // \sum_{j=1}^{n} w_j C_j$), carrying out a comparative study of heuristic and metaheuristic for three identical parallel machines.

In this paper, we propose an approach to solve tasks scheduling problem on machine single under unavailability periods.

2. PROBLEM DESCRIPTION

There are n tasks to schedule in a single machine. All the tasks are available at time zero.

- Each task j has associated a processing time pj and a weight Wj
- There is a time interval T_z between the completion times of thre consecutive maintenance activities.
- The tasks can not be interrupted.

Objective: To assign tasks to blocks between maintenance activities in such a way that the last task finishes as soon as possible,that is, to minimize the weighted sum of the completion time.

3. PROPOSED METHOD

3.1. Tabu Search

Tabu Search is a metaheuristic originally developed by Glover (1986). This method combines local search procedure with some rules and mechanism to surmount local optima obstacle avoiding the cycling trap. Tabu search is the metaheuristic that keeps track of the regions of the solution space that have already been searched in order to avoid repeating the search near these areas (Glover & Hanafi, 2002).

It starts from a random initial solution and successively moves to one of the neighbors of the current solution. The difference between tabu search and other Meta-heuristic approaches is based on the notion of the tabu list, which is a special short-term memory, storing of previously visited solutions including prohibited moves. In fact, short-term memory stores only some of the attributes of solutions instead of whole solutions. So, it gives no permission to revisit solutions, and then, avoids cycling and being stuck in local optima.

During the local search, only those moves that are not tabu will be examined, if the tabu move does not satisfy the predefined aspiration criteria. These aspiration criteria are used, because the attributes in the tabu list may also be shared by unvisited good quality solutions. A common aspiration criterion is better fitness, i.e. the tabu status of a move in the tabu list is overridden if the move produces a better solution.

The process of Tabu Search (TS) can be represented as follows:

3.2. Algorithm (TS)

Step 1 Generate initial solution x.

Step 2 Initialize the Tabu List.

Step 3 While a set of candidate solutions X“ is not complete.

Step 3.1 Generate candidate solution x“ from current solution x.

Step 3.2 Add x“ to X“ only if x“ is not tabu or if at least one

Step 4 Select the best candidate solution x* in X“.

Step 5 If fitness(x*) > fitness(x), then x = x*.

Step 6 Update Tabu List and Aspiration Criteria

Step 7 If the termination condition met, then finish;otherwise,go to Step 3.

3.3. Intensification

One memorizes the best-found solutions and tries to determine common proprieties to define interesting regions and orient the research towards these regions, by considering all the movement that leads to leaving these regions as tabu, for example.

The intensification allows to stop periodically the normal exploration process and to intensify her research effort within a region that seems promising. One of the methods of intensification application is to memorize the best-found solutions to go back to one of these solutions.

3.4. Diversification

This technique is the inverse of the intensification method. It directs the research towards the unexplored regions. Implementing this technique consists in memorizing the solutions the most frequently visited and imposing a penalty system, in order to favor the movement the less frequently used. In this paper, the first starting time is TES_1=25 minutes and the second restarting time is TES_2=20 minutes; these times are practically sufficiently enough for exploring the majority of regions.

3.5. Neighborhoods

Neighborhood determination constitutes the most important stage in metaheuristic methods elaboration. In the following part; we use three neighborhoods (neighborhood by swapping, neighborhood by insertion and neighborhood insertion by blocks).

Notations:

We denote by:

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

p_h : Execution time of the task h.

M : Single machine

k : Number of availability zones.

$Z = \{1, 2, ..., k\}$: Availability Zones.

E_z : Period of unavailability zones.

σ : Sequence assigned to the machine I.

w_h : Weight of the task h

C_h : Execution time of the task h by the machine I.

C_z $(z \in Z)$: Execution time of the task $j \in J_z$, allocated to the zone *z*.

f(σ): Objective function cost.

$f_{\mathbf{swapp}}$: Swapping algorithm cost.

$f_{\mathbf{insert}}$: Insertion algorithm cost.

f_{ins_bloc} : Insertion bloc algorithm cost

$f_{\mathbf{best}}$: Minimal cost.

T: Tabu List.

L: Tabu List Size.

4. HEURISTICS DESCRIBED

An initial solution is always necessary. For this reason, we suggest in this part the following heuristics based on two principles :

1-assigned the (best) task $_j$ where $\left(\frac{p_h}{w_h} = \min_{j \in J}\left\{\frac{p_j}{wj}\right\}\right)$ to machine M .

2-assigned the (best) task j where p_h= max(p_j) to machine M.

4.1. Formal statement

It is not useful to let the machine (idle) if a task can be assigned to this machine(smith,1956).

5. HEURISTICS

5.1. Heuristic (H1)

5.1.1. Initialization

```
Begin
    j={1, 2, …, n} ; $E_1$=0 ;σ=$\phi$, $f$ ($\phi$)=0; $p_j$ = random (1.99) ;$w_j$ = random (1.10) ; z=1
    Sort tasks $h \in J$ in increasing order according to the criterion $\frac{p_j}{w_j}$ in a list $U$
While ($U \neq \phi$ and $z_k \geq p_h$) do
Begin
```

```
Set $p_h = p_h / w_h$ from the top list of $U$ ;
End if
Assigned the task $h$ to the machine M;
Delete the task $h$ from the list $U$ ;
Compute $C_z = \sum p_j + E_z$;
Determine $\sigma = \sigma \cup \{h\}$ and $f_\sigma = f_\sigma + w_h C_z$;
End
Else
Begin
Set $z = z + 1$;
End
End if
End
```

5.2. Heuristic (H2)

```
Initialization
j={1, 2, …, n}; $E_1$=0 ;σ=$\phi$, $f$ ($\phi$)=0; $p_j$ = random (1.99); $w_j$ = random (1.10); z=1
Begin
Sort tasks $h \in J$ in increasing order according to the criterion $\frac{p_j}{w_j}$ in a list $U_1$
Sort tasks $h \in J$ in decreasing order according to the criterion $p_j$ in a list $U_2$
While ($U \neq \phi$ and $z_k \geq p_h$) do
Begin
Set $p_{h_1} = p_h / w_h$ from the top list of $U_1$
Set $p_{h_2} = \max p_h$ from the top list of $U_2$
```

```
End if
Assigned the task $h$ to the machine M;
Delete the task $h$ from the two lists $U_1 and U_2$ ;
Compute $C_z = \sum p_j + E_z$;
Determine $\sigma = \sigma \cup \{h\}$ and $f_\sigma = f_\sigma + w_h C_z$;
End
Else
Begin
Set $z = z + 1$;
End
End if
End
```

5.3. Algorithm

```
Step 1 Get an initial solution σ and T[1]=0 ;
Step 2 Do permutation by swapping
Step 3 Do permutation by insertion
Step 4 Do permutation by insertion by a bloc
Step 5 Compute: $f_1=f_{swapp}$ ;$f_2=f_{insert}$ ;$f_3=f_{ins\_bloc}$
Step 6 Consider L=$\sqrt{N}$ ( Tabu list size)
Step 7for k=1 to 3 Do
If $f_{init}<f_k$
Do:T[1]=$f_{init}$ ;
else T[1]=$f_{k;}$
End if
$T_k$=T [1];
```

End

Step 7.1 $f_{best}=\min(T_1, T_2, T_3)$

End if

Step 7.2 Display $\sigma(f_{best})$

5.3.1. Example1

Consider the problem P_1 with the following data:

Table 1: problem P_1

j	1	2	3	4	5	6
P_j	11	36	88	10	91	31
W_j	3	6	8	7	4	1
P_j/W_j	3.67	6	11	1,42	22.75	31

Results of heuristic (H_1)are : f= 2666 ; execution time = 0,156 s

Results of tabu (swapping) are :f= 2145 ; execution time =0,991 s

Results of tabu (insertion) are : f= 2431 ; execution time =1,024 s

Results of tabu (insertion by bloc) are : f= 2567 ; execution time =0 ,306 s

The best results are obtained by using tabu by swapping for f=2145.

5.3.2. Example2

Consider the problem P_2 with the following data:

Table 2: problem P_2

j	*1*	*2*	*3*	*4*	*5*	*6*
Pj	*11*	*36*	*88*	*10*	*91*	*31*
Wj	*3*	*6*	*8*	*7*	*4*	*1*
Pj/Wj	*3.67*	*6*	*11*	*1,42*	*22.75*	*31*
Pj(MAX)	*91*	*88*	*36*	*31*	*11*	*10*

Results of heuristic (H2)are : f= 2548 ; execution time = 0,650 s

Results of tabu (swapping) are :f= 1986 ; execution time =0,991 s

Results of tabu (insertion) are : f= 2367 ; execution time =1,542 s

Results of tabu (insert. by bloc) are: f= 2410; execution time =0 ,945 s

The best results are obtained by using by tabu (swapping) for f=1986.

6. COMPUTATIONAL ANALYSIS

6.1. Data generation

The proposed approaches were tested on problems generated with 1000 tasks similar to those used in previous studies (M'Hallah & Bulfin, 2005). For each task *j*, an integer processing time p_j was randomly generated in the interval $(1,99)$ with a weight w_j randomly chosen in the interval $(1,10)$.

The tables 1and 2 below presents:

1) The initial mean values of the objective function corresponding to the initial sequence.
2) The initial mean values of the objective function
3) The average times corresponding to the three neighborhoods.
4) The best costs.

AC: Average costs.

AT: Average time.

7. RESULTS

The results listed in tables(III and IV) show clearly that the tabu method based on neighborhood by swapping presents the best costs compared with tabu method based on neighborhood by insertion and by blocks.

This is due to the fact that the first neighborhoods ensures a faster tasks movement; besides that, the search space is richer with optimal partial sequences in each availability zones. This can also be explained by the nature of adopted neighborhoods.

The results show that execution time obtained by the proposed neighborhoods is acceptable. On the other hand, the heuristics amelioration rate between the three neighborhoods is remarkable (Figure 1 and 2, Graphic 1 and 2).

8. 7. CONCLUSION

In this paper, *n* tasks scheduling problem for machine sinle under availability constraint is discussed, with the aim of minimizing the weighted sum of the completion time. The approach based on tabu search allowed solving this problem, with the enhancement of initial solution obtained by a heuristics (H_1 and H_2) of complexity o(nlogn).

By considering three types of neighborhoods, tabu list and diversification strategy, the results of tabu search method were encouraging, and they will be more encouraging if good neighborhood based on problem's data is defined.

More encouraging if good neighborhood based on problem's data is defined.

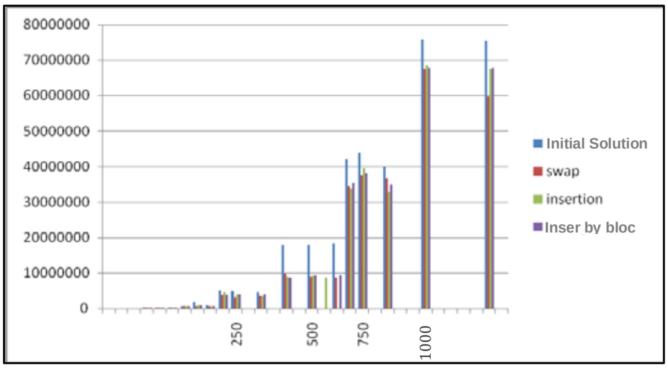
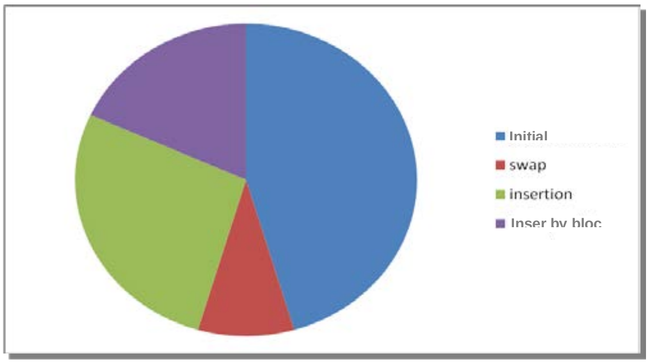


Figure1: Histogram of heuristic (H_1) cost amelioration based on tabu search for different N values.



Graphic 1: Circle graph of heuristic (H_1) cost amelioration

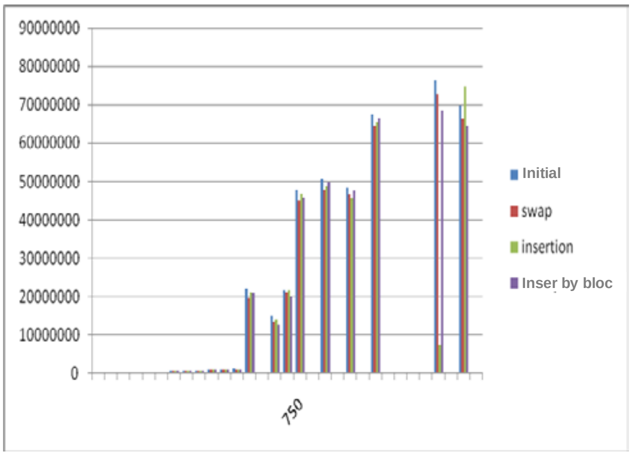
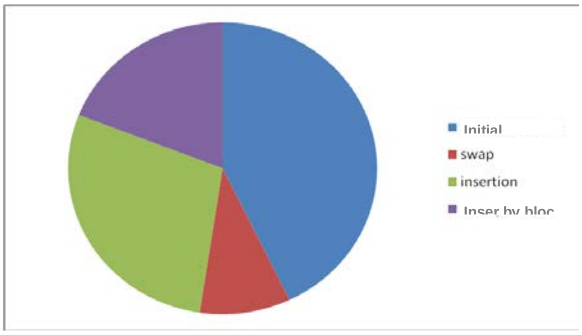


Figure 2: Histogram of heuristic (H_2) cost amelioration based on tabu search for different N values.



Graphic 2: Circle graph of heuristic (H_2) cost amelioration based on tabu search.

Table 3: Results obtained by heuristic (H1) and tabu search

N	**Initial Solution (H_1) (average of 3 instances)**		**Tabu search by swap**		**Tabu search by insertion**		**Tabu search by blocs**		**Best costs**
	AC	**AT (second)**	**AC**	**AT (second)**	**AC**	**AT (second)**	**AC**	**AT (second)**	
N =20	29190	0,185	41647	0,654	31779	0,307	35466	0,13	29190
	45768	0,201	40772	0,578	29096	0,224	32259	0,166	29096
	30731	0,194	29720	0,576	37763	0,447	33526	0,161	29720
N=50	205130	0,583	189551	0,634	223921	1,524	176484	0,328	176484
	207358	0,429	218017	0,603	219091	1,019	229906	0,437	218017
	214071	0,919	209126	0,645	194182	0,973	208584	0,437	194182
N=100	734976	2,212	682309	3,682	593040	5,179	707554	1,617	593040
	1994099	2,191	702905	3,385	786843	5,356	761648	1,472	702905
	839684	1,997	707977	3,297	685365	5,808	703850	1,700	685365
N=250	5090272	5,90	3845619	7,73	4659201	9,18	3697427	6,56	3697427
	4897215	6,14	3201037	7,61	4061839	9,63	3982569	6,49	3201037
	4658617	6,142	3597109	6,95	3312510	8,89	4045917	6,92	3012010
N=500	17866722	8,120	9730402	9,789	8964538	10,876	8763401	7,871	8763401
	17986067	7,128	9056321	9,562	9254175	10,501	9342104	7,549	9056321
	18410307	7,298	8657831	9,861	8765109	10,612	9297364	7,724	8657831
N=750	41931982	12,20	34537327	15,87	33762437	16,87	35318023	14,73	33762437
	43858415	13,01	37612943	16,08	39576182	16,49	38003183	13,98	37612943
	39895222	12,44	36578139	15,83	32789193	16,81	34987710	14,01	32789193
N=1000	75763079	16,80	67451280	20,563	68673189	25,675	67645329	17,977	67451280
	75377034	17,12	59672815	20,986	67563821	25,560	67832961	18,286	59672815
	74190765	17,44	66347819	20,926	58976897	25,241	68936103	18 ,672	58976897

Table 4: Results obtained by heuristic (H2) and tabu search

N	**Initial Solution (H_2) (average of 3 instances)**		Tabu search by swap		Tabu search by insertion		**Tabu search by blocs**		**Best costs**
	AC	**AT (second)**	AC	AT (second)	AC	AT (second)	**AC**	**AT (second)**	
N =20	49635	0,897	44957	1,83	**44845,66**	2,76	45738	1,75	49635
	41544	1,01	41235	1,72	41094	2,64	**40763**	1,70	**40763**
	34220	**0,99**	**33020**	1,97	34008	2,38	33872	1,84	**33020**
N=50	202482	4,43	200848	9,25	201830	14,40	**199673**	9,09	**199673**
	220786	6,65	220364	10,34	**210600**	16,03	219643	9,87	**210600**
	230501	3,98	197233	9,16	**226582**	14,19	219846	10,56	**226582**
N=100	903625	**7,65**	**834570**	14,25	856713	28,09	840163	13,98	**834570**
	863040	**9,86**	791284	15,18	**786173**	30,53	788527	15,76	**786173**
N=250	989476	**20,14**	**970528**	45,78	985476	35,80	980035	39,73	**970528**
	1098437	28,76	925376	39,73	**906718**	42 ,71	1028731	38,32	**906718**
	1287142	27,97	1001583	43,91	**973027**	40,93	1056738	40,03	**973027**
N=500	22206778	**40,12**	**19765183**	69,03	21165329	71,63	21067482	51,98	**19765183**
	15016104	**45,67**	13489345	72,87	13987543	79,51	**12692549**	67,90	**12692549**
	21646539	42,56	21135631	61,42	21598743	67,91	**20068251**	60,43	**20068251**
N=750	47951562	**60,87**	**45056218**	97,56	46875128	102,67	45978537	89,93	**45056218**
	50652915	**70,87**	**47872361**	95,54	48953672	112,03	49835872	92,89	**47872361**
	48501267	75,96	46734529	91,42	**45548271**	106,48	47629349	90,62	**45548271**
N=1000	67544328	**91,56**	**64672384**	120,76	65621738	129,36	66527183	113,67	**64672384**
	76248934	**87,53**	72789368	127,62	7472893	133,69	**68638624**	122,36	**68638624**
	69842351	84,06	66423946	124,30	74728194	131,42	**64673892**	123,72	**64673892**

REFERENCES

Abdenacer, G. (2014). metaheuristic algorithms for solving quadratic assignment problem, **IJACSA**, 5, 15-21.

Adamu M. O., & Adewunmi. A. (2013).Comparative study of meta-heuristics for identical parallel machines, **J. Eng. Technol. Res**., 5(7), 207-216.

Chang, P.-C., Chen, S.-H., Lie, T., & Liu, J. Y.-C. (2011). A genetic algorithm enhanced by dominance properties for single machine scheduling problems with setup costs, **International Journal of Innovational Computing Information and Control**, 7(5A), 2323–2344.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence, **Comput. Oper. Res**., 13, 533-549.

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

Hansen, P. (2006). The steepest ascent mildest descent heuristic for combinatorial programming, **Proceedings** of the Congress on Numerical Methods, Capri, Italy.

Haouari. M., & Ladhari, T. (2003). A branch-and-bound-based local search method for the flow shop problem, **J. Oper. Res. Soc**., 54, 1076-1084. DOI: 10.1057/palgrave.jors.2601612

Lee. C. Y. (1996). Machine scheduling with an availability constraints, **J. Global Optim**., 9, 395-416.

Lee C. Y. (1997). Minimising the makespan in two machines flow shop scheduling problem with availability constraints, **Oper. Res. Lett**., 20, 129-139.

M'Hallah, R., & Bulfin, R. L. (2005). Minimizing the weighted number of tardy jobs of parallel processors, **Eur. J. Oper. Res**., 160, 471-4847.

Sakarovitch, M. (1984). **Optimisation combinatoire**: Programmation discrete, Hermann, France.

Schmidt, G. (2000). Scheduling with limited machine availability, **European J. Oper**, 121, 1-15.

Selt, O., & Zitouni, R. (2014). A comparative study of heuristic and metaheuristic for three identical parallel machines, **Cjpas**, 3147-3153.

Smith, W. E. (1956). Various optimizes for single-stage production, **NavaRes.Logist**, 3, 59-66.

Liang, Y.-C., Hsiao, Y.-M., & Tien, C.-Y. (2013) Metaheuristics for drilling operation scheduling in Taiwan PCB industries, **International Journal of Production Economics,** 141(1), 189-198. DOI: 10.1016/j.ijpe.2012.04.014

Zribi, N., Kacem, I., El-Kamel, A., & Borne, P. (2005). Minimisation de la somme des retards dans un job shop flexible, **Revue e-STA (SEE)**, 6(6), 21-25.

Zitouni, R., & Selt, O. (2016). Metaheuristics to solve tasks scheduling problem in parallel identical machines with unavailability periods, **RAIRORes**, 50(1), 90.