Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Comparative Study Between a Genetic Algorithm and Tabu Search for Scheduling Problems on a Single Machine

Nabila Lounissi[1], Omar Selt[2] ,Allaoua Hemmak[3]
[1]*Mohamed Boudiaf University, M'Sila, Algeria*
[2]*ziane achour university, djelfa, Algeria*
[3]*Mohamed Boudiaf University, M'Sila, Algeria*

ARTICLE INFO

Received: 26 Dec 2024

Revised: 14 Feb 2025

Accepted: 22 Feb 2025

ABSTRACT

In this paper, we will present a comparative study between a genetic algorithm and tabu search for solving scheduling problems on a single machine to minimize the weighted sum of the task's end dates; since this problem is NP-hard in the strong sense; exact methods require a computational effort that increases exponentially with the size of the problem. Approximate algorithms enable the solution of this problem reasonably. In this farm, we present a genetic algorithm and tabu search metaheuristics, aiming to find an approximate solution to the problem under consideration. The results obtained are applicable in economics and industry.

Keywords:NP hard -scheduling- genetic algorithm -tabu search- neighborhoods

1. INTRODUCTION

The scheduling problem involves organizing the time required to complete task'sconsidering time constraints (deadlines and sequencing constraints), as well as utilization and availability constraints .A schedule is solution to the scheduling problem; it describes the execution of task's and the allocation of resources over time and aims to satisfy one or more objectives [10].More precisely,we speak of scheduling when we reserve the term sequencing for the case where only the relative order of tasks is fixed independently of the execution dates .

Tasks

A task **i** is elementary work entity (operation or set of operation), located in time by a start date $\mathbf{r_i}$or end date $\mathbf{c_i}$,the completion of which requires a duration $\mathbf{p_i}$ and which consumes resoures (material , personnel,ect.)

The constraints

In most scheduling problems, the task to be excuted are subject to constraints that must be satisfied when searching for an optimal solution .There are there types of constraints .

Potential constraints

These are the time location constraints, for example, task **i** must precede task **j**,or task **i** must be completed before acertin date .With this type of constraint,the problem is called a central scheduling problem.

Disjunctive constraints

When two tasks cannot be performed at the same time, we say that there is a disjunctive constraint that both tasks must satisfy.

Cumulaive constraints

They concern the evolution over time of the total volume of human or material resources devoted tothe execution of the tasks.

Resources

Resources are the means required for the execution of tasks; they are of two types.

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A resources is consumable if, after being allocated to one task ,it is no longer available for other tasks[12].Similar to the case of a raw material or money .

A Resource is renewable if,after being allocated to a task,it becomes availableagain , upon completion of that task, for other tasks , such as a machine ,a process,etc .

The criteria

The criteria of a scheduling problemis the economic function that one aims to optimize.It is also called in another sense (objective function).

In general, the criterion is considered as an application **F**from the set $\sum \sigma f$ permutation of the tasks of theproblem posed to $\boldsymbol{R}^{+}$,which , at each permutation .

$(j_1, j_2,...j_n)$, associates a positive real number.

$F(j_1, j_2,...j_n)$,which expresses, in addition to the efficient use of resources, the overall execution time and compliance with the greatest number of constraint , the objective function associated with a scheduling problem is defined by : $F\ obj = \underset{\sigma\epsilon\Sigma}{\mathrm{OPT}} F(\sigma)$

Neighborhood Methods

Neighborhood methods are based on the notion of neighborhood.

Definition

Let X be the set of admissible configurations of a problem.A neighborhood is any applicationN: $x \rightarrow 2^x$,and a

Neighborhood exploration mechanism is any procedure that specifies how the search moves from a configuration $\boldsymbol{S} \in \boldsymbol{X}$ to a configuration.$\boldsymbol{S'} \in \boldsymbol{N(s)}$

Configuration S is said to be a local optimum (minimum) concerning the neighborhood N if

$\forall\, S' \in N(S): f(S) \leq f(S')$

A typical neighborhood matching method starts with an initial configuration and then performs an iterative process that consists of replacing the current configuration with one of its neighbors (Schmidt,G,2000),taking into account account the cost function .This process stops and returns the best configuration found when the stopping condition is met .This condition can generally be a limit for the iteration member.

Initialization: Generate an initial population P of solution of size $|P| = n$

Algorithm

Repeat:

Crossover: Combine the two parent solutions X and Y to form a child solution y

Conditional mutation of y

Choose an individual solution to be replaced in the population

Replace y with y in the population

Until the stopping criterion is met

2. METHODS

Proposed methods:

Genetic algorithm

This class of methods is based on an imitation of the adaptation phenomena of living beings.

Genetic algorithms operate on ananalogy with the reproduction of living beings.

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Generally speaking , genetic algorithms use the sum principle :a population of individuals(corresponding to solutions) evolves at the same time ,as in natural evolution in biology.For each individual ,its ability to adapt to the external environment is measured by fitness [8] .Genetic algorithms are then based on three functionalities.

Selection

Which allows us to favor individuals who have better fitness for us ,the fitness will be the most valuable, as it saves the value of the objective function of the solution associated with the individual.

Crossing

Which combines two parent solutions to form one or tow children (offspring), trying to keep the good characteristics of the parentsolution.

Mutation

Which allows diversity to be added to the population by mutating certain characteristics (genes) of a solution?

Therepresentation of solutions (coding) is a criticalpoint in the success of a geneticalgorithmIt must, of course adap as well as possible to the problem and the evaluation of a solution.

Algorithm

1: Initialization: generate an initial population **P**of solutions of size$|\boldsymbol{P}| = \boldsymbol{n}$

2: Repeat

3: Select: choose two solution X_1 and X_2 using a selection technique

4: Crossover: combine the two parent solutions X_1 and X_2 to form a child solution y

5: Mutation of y under conditions

6: Choose an individual solution yto be replaced in the population

7: Replace **y** with yin the stopping criterion is satisfied

Tabu Search

The Tabu method was introduced simultaneously by several researchers, P. Hansen, F.Glover, and B.Jaumard (1986).

Tabu search is used because it prohibits the retrieval of recently visited solutions.

At each iteration, the least –worst neighbor is chosen [3].

To avoid cycles, i.e.,the infinite repetition of a sequence of moves, the last L moves are considered forbidden,and L is the size of the Tabu list.

Ateachiteration,the move performed is therefore the least-worst non-Tabu move.

Tabu List

The Tabu list represents short-term memory; it contains the attributes of the most frequently performed moves.

This list is maintained to guide the search.

Aspiration Criterion

This criterion is often used to remove Taboo restrictions from a high- quality movie. Itallows bypassing certain forbidden cases.It main use is to override the prohibition of a move if it allows obtaining an element better than the solution found so far (consists of revoking the Taboo status of a move if the latter allows solving a higher quality than that of the best solution found so far).

Intensification Criterion

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Intensiification consists of returning to one of the best solutions found so far and then resuming the search for this solution.

The intensification strategy is embodied in the following algorithm by reinforcing the search in the list of best moves.

Diversification Criterion

This consists of generating a new solution, different from the one already explored, with the aim of moving in a new direction, to explore another region.

Stopping Criterion

Generally, to stop the algorithm,two criteria are taken into account or ignored .First,ateachiteration,the total iteration counter has not exceeded a maximum number since the beginning of the process.

We base our calculation on another threshold value, which corresponds to the maximum number of iterations we allow between two consecutive modifications (improvements) of the (best) solution [2].

But instead of considering the number of iterations, we could also base our calculation on the total time,which should be less than a maximum value [13].

General Tabu Search Algorithm

Below, we present the general Tabu search algorithm,withan emphasis on short-term memory and aggressive exploration :

Generate an initial solution S randomly

$S^* \leftarrow S$; $C^* \leftarrow F(S)$/S^* is the best solution encountered, C^* is its cost,and F is the objective function

Add Sto the tabu list; $k \leftarrow 0$

Repeat until an end criterion is met.

Choose from the neighborhood of S_k, V (S_k), the move that minimizes and that does not belong to the taboo list, best (S_k)

$S_{k+1} \leftarrow best(S_k)$

If the tabu list is full, then

Replace the last element with S_{k+1}

End if

AddS_{k+1} to the tabu list

End if

If $(C(S_{k+1}) < C^*)$ then

$S^* \leftarrow S_{k+1}, C^* \leftarrow C(S_{k+1})$

End if

End

In each availability period, jobs must be scheduled according to the WSPT rule in an optimal schedule [11].

$$\frac{P_h}{w_h} = min\left\{\frac{p_k}{w_k}\right\}$$

Illustrative example

Given the problem of 5 jobs, we consider the size of the Tabu List =2

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

All jobs are available at the start of the schedule.

job	1	2	3	4	5
P_i	2	5	6	4	8
w_i	3	4	2	1	2

Initialization:

Maximum number of iterations =6

$\sigma^* \leftarrow \sigma_0 \; and \; f^* \leftarrow f_0$

Tabu list =∅

$i \leftarrow 1$

iterations 01:the neighboods are :σ_1 :(1,2,3,4,5) (2,3,1,4,5) (2,1,4,3,5) (2,1,3,5,4).

f_i: 137 180 141 144

bestneighbor no tabu

σ_0 = (1,2,3,4,5)

Tabu list{(1 ,2,3,4,5)}

$i \leftarrow 2$

Iterations 02:the neighborhoods are :

σ_1:(2,1,3,4,5) (1,3,2,4,5) (1,2,4,3,5)(1,2,3,5,4).

f_i : 144 204 139 137

best neighbor tabu

σ_0 =(1,2,4,3,5)

Tabu list {(1,2,4,3,5)}

$i \leftarrow 3$

Iterations 03:the neighborhoods are :

σ_1 : (2,1,4,3,5) (1,4,2,3,5) (1,2,3,4,5) (1,2,4,5,3).

f_i: 146 164 137 143

best neighbor

σ_1 = (1,2,4,5,3)

Tabu list {(1,2,4,5,3)}

$i \leftarrow 4$

Iterations 04 :the neighborhoods are :

σ_1 : (2,1,4,5,3) (1,4,2,5,3) (1,2,5,4,3) (1,2,4,3,5).

f_i: 150 168 143 139

best neighbor no tabu

σ_0 =(1,2,4,3,5)

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Tabu list={(1,2 ,4,3,5),(1,2,4,5,3)}

$i \leftarrow 5$

Iterations 05 :the neighborhoods are :

σ_1 : (2,1,4,3,5) (1,4,2,3,5) (1,2,3,4,5) (1,2,4,5,3).

f_1: 146 164 137 143

best neighbor tabu

σ_0 =(2,1,4,3,5)

Tabu list← {(2,1 ,4,3,5),(1,2,4,3,5)}.

$i \leftarrow 6$

Iterations 06 :the neighborhoods are :

σ_1 : (1,2,4,3,5) (2,4,1,3,5) (2,1,3,4,5) (2,1,4,5,3).

f_1: 139 178 144 150

best neighbor tabu

σ_0 =(2,1,3,4,5)

Tabu list← {(2,1 ,4,5,3),(2,1,4,3,5)}.

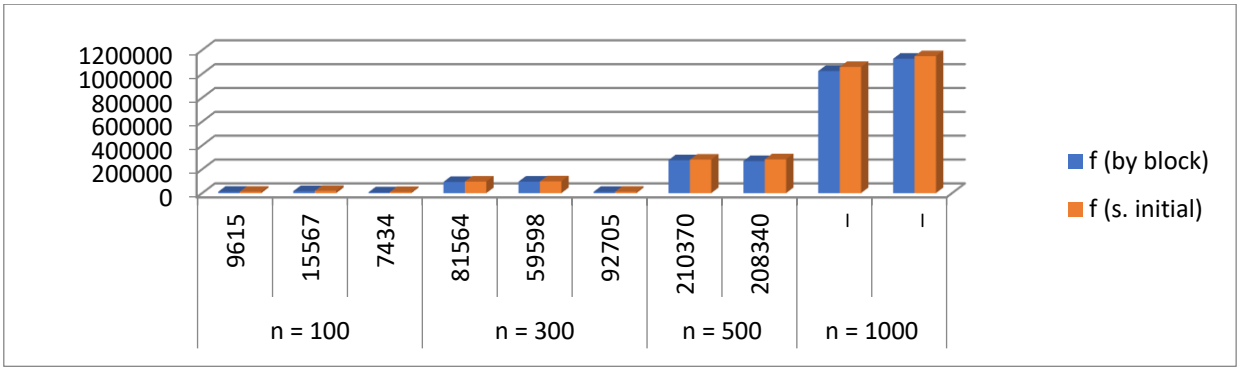
$I \leftarrow 7$ stop

So the search process stop, and the best solution found is :

σ^* =(1,2,3,4,5) and $f^* = \sum_{i=1}^{5} w_i c_i = 137$

jobs	f (swap)	f (by block)	f (s.initial)
N=10	9615	11161	11740
	15567	16482	18917
	7434	6077	9616
N=30	81564	93955	96735
	59598	96403	98680
	92705	10620	11620
N=50	210370	276280	280970
	208340	269140	283690
N=100	-	1024100	1058500
	-	1127300	1149000

Table 1.Results from two approaches



Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Fig1.Amelioration cost between two neighbourhoods

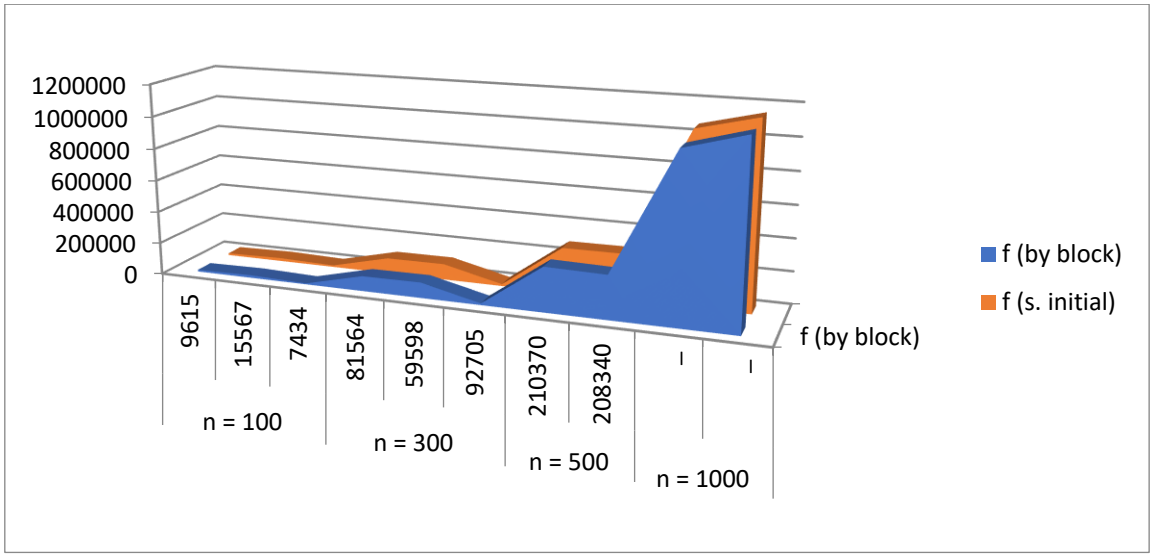


Fig2.The best cost

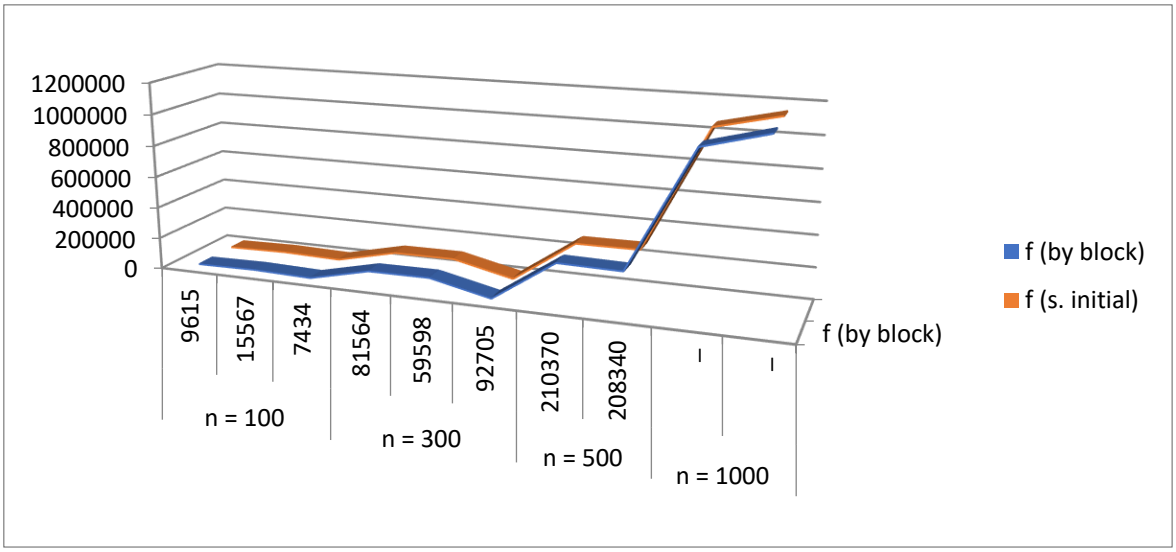


Fig3.Amelioration cost between two neighborhoods.

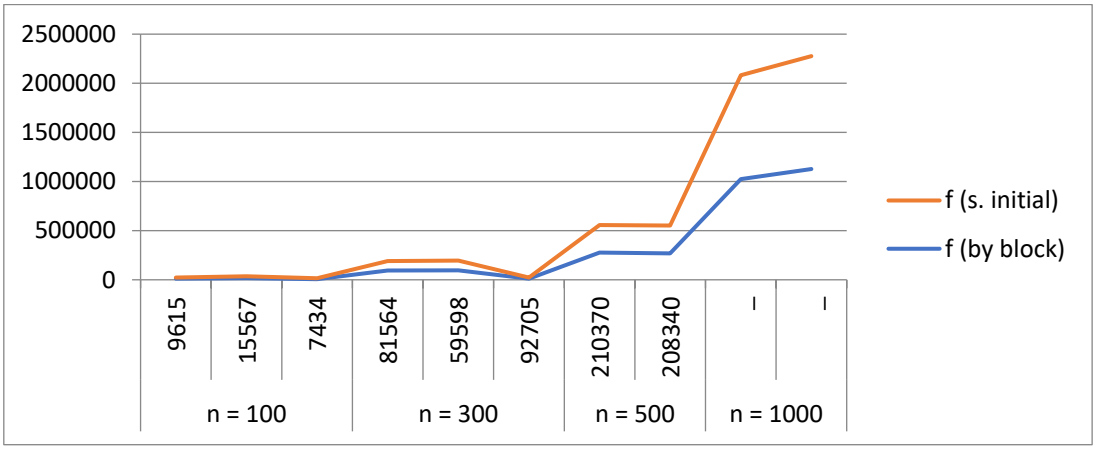


Fig4.The best cost

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

3. RESULTS AND DISCUSSION

Genetic algorithms are based on:

A chromosomal representation of the problem's solutions.

A method for generating an initial population of solutions.

An evaluation method that ranks the solutions according to their suitability.

4. CONCLUSION

In this work, we presented agenetic algorithm and tabu search method forsolvingthe scheduling problem on asingle machine with unavailability periods,and we suggested different types of neighborhoods.Each time we noticed the improvement in the solutions for our problem and the the best solution based on neighborhood swapping .

Therefore, the neighborhood improvement strategy (swapping two adjacent jobs)is better than the neighborhood by block ;when the number of jobs is less than 50,but for jobs exceeeding 50, neighborhood by block gives better results with minimal execution time.In practice, we have noticed that tabu search presents some better solutions, but theoretically, the genetic algorithm presents all possible effective solutions.

REFERENCES

[1] Adamu, M. O., and Adewunmi, A. (2012). Metaheuristics for scheduling on the parallel machine to minimize the weighted number of early and tardy jobs. Int. J. Phys. Sci. 7(10): 1641-1652.
[2] 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.
[3] Glover, F. (1986) Future paths for integer programming and links to artificial intelligence, Comput. Open Res. 13: 533-549.
[4] Hansen, P. (1986). The steepest ascent, mildest descent heuristic for combinatorial programming. In: Proceedings of the Congress on Numerical Methods.
[5] Ho, J.C., and Chang, YL.(1995). Minimizing the number of tardy jobs from parallel machines. Eur. J. Oper. Res. 84: 334-355.
[6 Lee, C. Y. (1996). Machine scheduling with an availability constraint. J. Global Optim. 9:395-416.
[7] Lee, C.Y.(1997). Minimizing the makespan in the machine flow shop scheduling problem with availability constraints. Oper. Res. Lett. 20:129-139.
[8] Lee, C.Y.(1999). Two-machine flow shop scheduling problem with availability constraints. European J. Oper. Res. 114:420-429.
[9] M'Hallah, R., and Bulfin, RL.(2005). Minimizing the weighted number of tardy jobs of parallel processors. Eur. J. Oper. Res. .160: 471-484.
[10] Schmidt, G.(2000). Scheduling with limited machine availability. European J. Oper. Res. 121:1-15.
[11] Smith, WE. (1956) Various optimizers for single-stage production. Naval Res. Logistics. 3:59-66. Yun-Chia, L., H., Yu-Ming. and Chia-Yun, T.(2013) Taiwan PCB industries. Int. J. Prod. Econ. 141(1):189-198.
[12] Zribi. I, Kacem. I, and Borne. P. (2005) Minimisation de la Somme des retards dans un job shop flexible. Revue e-STA (SEE). 2(2):2-11
[13] Zitouni. R and Selt. O (2016) Metaheuristics to solve task scheduling problems in parallel identical machines with unavailability periods, RAIRO. Oper. Res,50(1), pp. 83-90.