BOHR International Journal of Operations Management Research and Practices
2024, Vol. 3, No. 1, pp. 1–7
DOI: 10.54646/bijomrp.2024.22
www.bohrpub.com

CASE STUDY

# Application of tabu search, TS, to solve a flow shop scheduling problem with changeover times in operations: A case study

**Phong Nguyen Nhu*** **and Thuy Nhi Nguyen Thi**

HCMC University of Technology, VNU HCM, Ho Chi Minh City, Vietnam

***Correspondence:**
Phong Nguyen Nhu,
nnphong@hcmut.edu.vn

**Received:** 12 October 2023; **Accepted:** 19 October 2023; **Published:** 05 January 2024

Flow Shop Scheduling (FSS) Problems are NP-hard combinatorial optimization problems. It is quite difficult to achieve an optimal solution for FSS problems with mathematical modeling approaches because of its NP-hard structure. Tabu search (TS), a mega-heuristic algorithm, plays a major role in searching for near-optimal solutions for NP-hard optimization problems. This paper develops a TS model for solving a FSS problem with the objective to reduce total weighted tardiness time, and constraint on changeover time in operations. The performance of the TS model is compared with that of traditional EDD heuristics, being used. The result shows that the objective value has been reduced by 43%, from 215.95 to 123.07 (h). It indicates that the TS model is a good approach for FSS problems.

**Keywords:** flow shop scheduling problems, tardiness time, changeover times, tabu search, mega heuristic.

## Introduction

Scheduling is the allocation of resources to perform a collection of tasks over a period of time. Scheduling problems determine the order or sequence for processing a set of jobs through several machines in an optimal manner. FSS problems consider m different machines and n jobs; each job consists of m operations and each operation requires a different machine and all the jobs are processed in the same processing order.

The problem to be solved is a FSS problem with the assumption that the orders are ready at the start of the scheduling process. The objective of the problem is to minimize the total weighted tardiness of orders. The constraints are on the sequence of orders, on the sequence of operations in the orders, and on machine changeover time.

The model of the problem is built based on the above assumptions, objectives, and constraints. A TS model is built. Based on the model of the problem, the TS model will find a good solution for the problem, which will be compared with the solution of the currently used heuristic model to evaluate the effectiveness of the TS model.

## Literature review and research methodology

### Flow shop scheduling FSS problems

Flow shop scheduling problems consider different machines and different jobs. Each job consists of different operations and each operation requires a different machine and all the jobs are processed in the same processing order (1).

Flow shop scheduling is categorized as an NP-hard problem, so it is difficult to develop algorithms to solve it (2). Jatinder N. D. Gupta, Nagarajan Palanim Uthu, and Chuen-Lung Chen designed a tabu search-based heuristic for the two-stage flow shop problem with the makespan minimization as the primary criterion and the minimization of total flow time as the secondary criterion (3). Moch Saiful Umam, Mustafid Mustafid, and Suryono Suryono combined the tabu search process with a genetic algorithm to solve flow shop scheduling problem with the objective of minimizing makespan (2). Anna Burduk, Kamil Musiał, Joanna Kochańska, Dagmara Górnicka, and Anastasia Stetsenko applied tabu Search and genetic algorithm to solve production process scheduling problems and found that intelligent methods can find, in relatively short time, the solution that is close to the optimal and acceptable from the problem point of view (4).

## Tabu search

According to Farhad Kolahan et al. (5), heuristic algorithms are widely used to solve very large and complicated optimization problems, in recent years, and Tabu Search (TS) is one of the most efficient neighborhood search algorithms.

Tabu search (TS), suggested by Glover and Laguna in 1997, is one of the most popular meta-heuristics to find solutions of various combinatorial optimization problems. TS guides a local search procedure to explore the solution space beyond local optimality.

In order to apply TS to a problem, generally the solution space of the problem is represented by a population of codes. An evaluation value is associated with each code. The evaluation function is a measure of the extent to which the objective of the problem is achieved. The Fbest value is the best evaluation value, found during the search.

TS allows intelligent problem solving by the incorporation of adaptive memory and responsive exploration. Key elements of the search path are selectively remembered and strategic choices are made to guide the search out of local optima and into diverse regions.

However, considering every possible move from the current solution may be extremely time consuming and computationally expensive. In order to avoid cycling and becoming trapped in local optima, certain moves that lead to previously explored regions are forbidden. Attributes of recently visited solutions are set to be tabu for a certain number of iterations, and these moves are stored in the tabu list. Elements of tabu list define all tabu moves that cannot be applied to the current solution. The size of tabu list is bounded by a parameter. The tabu status of a move may be canceled, making it an allowable move if an aspiration criterion is satisfied. A tabu move can always be allowed to be chosen if it creates a solution better than the incumbent best solution.

A typical TS implementation starts from an initial solution and moves from the current solution to the best one among its neighborhoods at each iteration, even if this new solution is worse than the one available, until a pre-specified termination rule becomes true.

## Research methodology

FSS problem is a NP hard problem with a large size of solution space. The methodology, used in this research to solve the problem, includes 2 phases (1):

- Phase A: Construct the model of the problem.
- Phase B: Use TS model to solve the problem.

In phase A, the model of the problem is formulated with the objective of minimizing the total weighted tardiness time and constraint on operation changeover time.

Based on the model of the problem, a TS model is used to solve the problem in phase B. The TS procedure (1) is as follows:

Step 1: Initialize the TS model.

Step 2: Generate the initial solution $S_0$. Set k = 0.

Step 3: Generate the neighborhood population $P_N(k)$.

Step 4: Find the next solution $S_{k+1}$. Set k = k + 1,

Step 5: Check the termination rule. If No, return to step 3. If Yes, finish the loop.

**Step 1** initializes the TS model by setting up the structure and parameters of the TS model, including the method of coding, the TS factors, and the termination rule.

For coding, the orders are numbered from 1 to n (n is the total number of orders), each code is a string of n numbers. Each number corresponds to an order. The sequence of numbers Gi, i = 1÷n, represents the sequence of orders scheduled.

$$C = [G1, G2, \ldots, Gn]$$

The TS factors include the evaluation function, the current best value, the TS operators, and the tabu list. The evaluation function is defined according to the objective function of the problem, is used to evaluate scheduling codes. The current best value Tbest is the best objective value, updated after each iteration.

The TS operators include the neighborhood and search operators. Tabu list is defined by tabu members and tabu list size. Tabu members are moves in the previous loops. Tabu list size is a parameter of the algorithm. Tabu list is updated after each iteration.

The termination rule in this research: The best objective value Tbest does not improve after a specific number of consecutive iterations.

**Step 2** generates the initial solution S0. A good initial solution will be a good starting point for the search. The initial solution is often chosen by heuristic methods, SPT, LPT, EDD. In this research, the initial solution is generated by the best heuristic method that gives the best objective value. This steps also starts the algorithm by setting the iteration counting index k to 0.

**Step 3** uses neighborhood operator to generate the neighborhood population $P_N(k)$ from the current string $S_k$. This research uses SWAP as the neighborhood operator.

SWAP uses the method of permutation of adjacent numbers in the string to find the neighborhood. For a string with n numbers, SWAP will generate n-1 neighboring strings.

**Step 4** finds the next solution $S_{k+1}$ from the neighborhood population $P_N(k)$ by using the search operator. The search operator relies on the objective function and the tabu list to find the best solution in the neighborhood region, the chosen solution must not violate the tabu list. This solution will be selected for the next iteration, if any. This step also updates the tabu list and the current best Tbest if the objective value of the solution is better than the current Tbest. This step also increases the iteration counting index k by 1 to prepare for the next iteration, if any.

**Step 5** checks the termination rule. The algorithm usually terminates after a number of iterations if the objective function is not improved. If the termination rule is not met, the algorithm returns to step 3 for the next iteration. If the termination rule is met, the iteration loop is stopped.

The research methodology of the TS model is shown in the following sections.

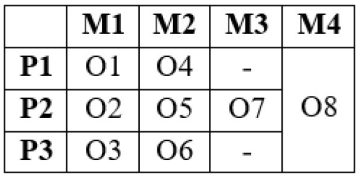
## The model flow shop scheduling problem

The problem to be solved is a FSS problem (1) with 10 orders, $O_i$, i = 1÷10, scheduling on 4 machines, $M_1$, $M_2$, $M_3$, $M_4$. Each order has 3 parts, P1, P2, P3, processed in 8 operations, $O_j$, j = 1÷8, distributed on the 4 machines as in Figure 1.

The weight $W_i$, i = 1÷10, and the due date $D_i$, i = 1÷10, of order i are estimated in Table 1.

The processing time $P_{ij}$ of order i, i = 1÷10, on operation j, j = 1÷8, are estimated in Table 2.

The changeover times in hours on operation j, j = 4÷8 are equal to 0, $S_j$ = 0, j = 4÷8. The changeover times in hours on operation j, j = 1÷3 are the same and depend on the current order i = 1÷10, and the next order, i' = 1÷10, as shown in Table 3.



| | M1 | M2 | M3 | M4 |
|---|---|---|---|---|
| P1 | O1 | O4 | - | O8 |
| P2 | O2 | O5 | O7 | |
| P3 | O3 | O6 | - | |

**FIGURE 1 |** The production process.

**TABLE 1 |** The weight $W_i$, i = 1÷10, and the due date $D_i$, i = 1÷10, of order i.

| i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $W_i$ | 3.70 | 3.40 | 3.30 | 4.65 | 3.90 | 2.35 | 2.70 | 4.55 | 4.65 | 4.30 |
| $D_i$ (h) | 24 | 36 | 40 | 60 | 68 | 80 | 88 | 88 | 96 | 96 |

The model is set up with variables $TS_{ij}$ being the start time, $TE_{ij}$ being the completion time of order i at operation j, and $T_i$ being the tardiness time of order i. The constraints on the sequence of operation on each order are as follows.

$$TS_{i4} \geq TE_{i1},$$
$$TS_{i5} \geq TE_{i2},$$
$$TS_{i6} \geq TE_{i3},$$
$$TS_{i7} \geq TE_{i5},$$
$$TS_{i8} \geq \max(TE_{i4}, TE_{i6}, TE_{i7}).$$

The start time of order i at operation j, $TS_{ij}$ depends on the end time of the previous order i', $TE_{i'j}$ and the changeover time between the orders on operation j.

$$TS_{ij} = TC_{i'j} + S_j$$

The end time of order i on operation j, $TE_{ij}$, is determined by the start time and processing time of the order.

$$TC_{ij} = TS_{ij} + P_{ij}$$

The tardiness time of order i, $T_i$, is determined by the end time in the last operation and due time of the order.

$$T_i = \text{Max}\,(0, TE_{i8} - D_i)$$

**TABLE 2 |** The processing time $P_{ij}$ of order i, i = 1÷10, on operation j.

| j | $P_{1j}$ | $P_{2j}$ | $P_{3j}$ | $P_{4j}$ | $P_{5j}$ | $P_{6j}$ | $P_{7j}$ | $P_{8j}$ | $P_{9j}$ | $P_{10j}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2.34 | 6.17 | 6.20 | 7.09 | 2.47 | 9.56 | 3.44 | 14.74 | 5.26 | 1.39 |
| 2 | 2.25 | 0.94 | 7.05 | 3.22 | 1.07 | 5.26 | 1.81 | 7.49 | 5.26 | 0.66 |
| 3 | 0.00 | 3.99 | 4.34 | 4.38 | 2.60 | 9.52 | 0.00 | 16.75 | 14.74 | 0.44 |
| 4 | 2.63 | 1.33 | 1.08 | 9.00 | 2.60 | 12.56 | 1.32 | 17.54 | 16.52 | 1.29 |
| 5 | 1.06 | 1.00 | 6.58 | 9.00 | 4.21 | 12.64 | 0.65 | 17.54 | 16.52 | 1.29 |
| 6 | 0.00 | 8.33 | 6.58 | 3.60 | 3.95 | 12.64 | 0.00 | 13.33 | 15.79 | 0.40 |
| 7 | 2.67 | 3.29 | 1.13 | 2.21 | 0.68 | 4.00 | 1.11 | 5.26 | 1.60 | 0.44 |
| 8 | 4.08 | 3.31 | 3.42 | 12.00 | 3.95 | 6.12 | 2.21 | 8.22 | 5.26 | 1.32 |

**TABLE 3 |** Changeover time (h) $S_j$, j = 1÷3.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.0 | 0.5 | 2.0 | 0.0 | 0.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 |
| 2 | 0.5 | 0.0 | 0.0 | 0.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 |
| 3 | 2.0 | 0.0 | 0.0 | 2.0 | 0.5 | 2.0 | 0.5 | 0.5 | 2.0 | 0.5 |
| 4 | 0.0 | 0.0 | 2.0 | 0.0 | 0.0 | 0.0 | 2.0 | 2.0 | 2.0 | 2.0 |
| 5 | 0.0 | 2.0 | 0.5 | 0.0 | 0.0 | 2.0 | 0.5 | 2.0 | 0.5 | 0.5 |
| 6 | 2.0 | 2.0 | 2.0 | 0.0 | 2.0 | 0.0 | 0.5 | 2.0 | 0.0 | 0.0 |
| 7 | 2.0 | 2.0 | 0.5 | 2.0 | 0.5 | 0.5 | 0.0 | 0.5 | 2.0 | 0.0 |
| 8 | 2.0 | 2.0 | 0.5 | 2.0 | 2.0 | 2.0 | 0.5 | 0.0 | 0.0 | 0.0 |
| 9 | 2.0 | 2.0 | 2.0 | 2.0 | 0.5 | 0.0 | 2.0 | 0.0 | 0.0 | 2.0 |
| 10 | 2.0 | 2.0 | 0.5 | 2.0 | 0.5 | 0.0 | 0.0 | 0.0 | 2.0 | 0.0 |

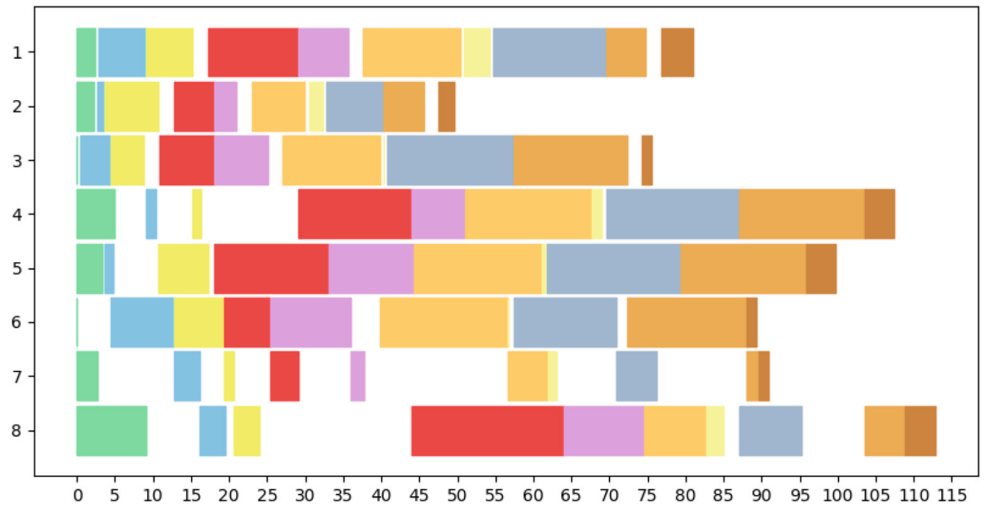


**FIGURE 2 |** The Gantt Chart by EDD dispatching method.

The objective function that minimizes the total tardiness is defined as follows.

$$\text{Tbest} = \text{MinT}, \text{T} = \Sigma(\text{W}_i^* \text{T}_i, \text{i} = 1 \div 11)$$

The company is currently using the EDD dispatching method. The sequence of dispatching S and the value of the objective function are as follows:

$$\text{S} = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10);$$
$$\text{T} = 215.95 \text{ (h)}$$

The Gantt Chart is as in Figure 2.

## The TS model for the flow shop scheduling problem

The above FSS problem is a NP hard problem with the size of the solution space of 10! or 3,628,800. The above TS model is used to solve the problem with following steps.

**Step 1: Initialize the TS model**

This step sets up the method of coding, the TS factors, and the termination rule.

***The method of coding:*** The orders are numbered from 1 to 10. Each code is a string of 10 numbers. Each number corresponds to an order. The sequence of numbers $G_i$, i = 1÷10 represents the sequence of order scheduled. For example, the string of EDD scheduling method is as follows.

$$\text{C} = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$$

**The TS factors** include the evaluation function, the current best value, the parameters of the TS operators, and the Tabu List as shown in Table 4.

***The termination rule:*** The best objective value Tbest does not improve after 10 consecutive iterations.

**Step 2: Generate the initial solution $S_0$, set k = 0**

A good initial solution will be a good starting point for the search. With the objective of minimizing tardiness time, the initial solution is selected from the EDD method. The strings and its corresponding objective values are as follows.

$$\text{S}_0 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$
$$\text{T} = 215.95(h)$$

The initial tabu list TL and the current best value are as follows.

$$\text{TL} = \emptyset$$
$$\text{Tbest} = 215.95 \text{ (h)}$$

From the initial solution $S_0$, iteration 1 is executed with steps 3, 4, and 5.

**Iteration 1**

**Step 3: Generate the neighborhood population $P_N{}^{(0)}$**

This step uses the neighborhood operator SWAP to generate the neighborhood population $\mathbf{P}_N{}^{(0)}$ from the current string $S_0$.

$$\text{P}_\text{N}{}^{(0)} = \{\text{N1, N2, N3, N4, N5, N6, N7, N8, N9}\}$$

The strings in neighborhood population $\mathbf{P}_N{}^{(0)}$ and their objective values are shown in Table 5.

**TABLE 4 |** The TS factors.

| Factors | Values |
|---|---|
| The evaluation function | The objective function T |
| The current best value Tbest | The best objective value |
| Neighborhood operator | SWAP |
| Neighborhood region size | 9 |
| Tabu members | 2 swap numbers in the previous chosen string |
| Tabu list size | 5 |

**TABLE 5 |** The neighborhood population $P_N^{(0)}$.

| String | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 | $T_i$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_0$ | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | 215.95 |
| N1 | **2** | **1** | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 264.75 |
| N2 | 1 | **3** | **2** | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 203.75 |
| N3 | 1 | 2 | **4** | **3** | 5 | 6 | 7 | 8 | 9 | 10 | 242.49 |
| N4 | 1 | 2 | 3 | **5** | **4** | 6 | 7 | 8 | 9 | 10 | 175.90 |
| N5 | 1 | 2 | 3 | 4 | **6** | **5** | 7 | 8 | 9 | 10 | 241.39 |
| N6 | 1 | 2 | 3 | 4 | 5 | **7** | **6** | 8 | 9 | 10 | 221.26 |
| N7 | 1 | 2 | 3 | 4 | 5 | 6 | **8** | **7** | 9 | 10 | 224.11 |
| N8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **9** | **8** | 10 | 239.71 |
| N9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | **10** | **9** | 175.65 |

**Step 4. Find the next solution $S_1$**

This step uses the search operator to find the next solution $S_1$ that is the best solution in the neighborhood population $\mathbf{P}_N{}^{(0)}$, and does not violate the current tabu list. The strings in the neighborhood population $\mathbf{P}_N{}^{(0)}$, and their corresponding tabu members TM are shown in Table 6.

From Table 6, the best string in the neighboring population $\mathbf{P}_N{}^{(0)}$ without violating tabu list and its objective value are as follows.

$$\begin{aligned} \text{N9} &= (1, 2, 3, 4, 5, 6, 7, 8, 10, 9) \\ \text{T} &= 175.65 \text{ (h)} \end{aligned}$$

This string is selected as the next solution for the next iteration, if any.

$$S_1 = (1, 2, 3, 4, 5, 6, 7, 8, 10, 9)$$

The tabu list and the current best Tbest are updated as follows.

$$\begin{aligned} \text{TL} &= \{(9, 10)\} \\ \text{Tbest} &= 175.65 \text{ (h)}. \end{aligned}$$

**Step 5. Check the termination rule**

After iteration 1, $S_1$ is the best string with the best objective value of 175.65, appearing only once. The termination rule is not satisfied, so iteration 2 is executed.

**Iteration 2**

**Step 3: Generate the neighborhood population $P_N{}^{(1)}$**

This step uses the neighborhood operator SWAP to generate the neighborhood population $\mathbf{P}_N{}^{(1)}$ from the current string $S_1$.

$$P_N{}^{(1)} = \{\text{N1, N2, N3, N4, N5, N6, N7, N8, N9}\}$$

The strings in neighborhood population $\mathbf{P}_N{}^{(1)}$ and their objective values are shown in Table 7.

**Step 4. Find the next solution $S_2$**

This step uses the search operator to find the next solution $S_2$ that is the best solution in the neighborhood population $\mathbf{P}_N{}^{(1)}$, and does not violate the current tabu list. The strings in the neighborhood population $\mathbf{P}_N{}^{(1)}$, and their corresponding tabu members TM are shown in Table 8.

From Table 8, string N9 in $\mathbf{P}_N{}^{(1)}$ violates tabu list. The best string in the neighboring population $\mathbf{P}_N{}^{(1)}$ without violating

**TABLE 6 |** Find the next solution $S_1$.

| String | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 | $T_i$ | TM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N1 | **2** | **1** | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 264.75 | (1,2) |
| N2 | 1 | **3** | **2** | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 203.75 | (2,3) |
| N3 | 1 | 2 | **4** | **3** | 5 | 6 | 7 | 8 | 9 | 10 | 242.49 | (3,4) |
| N4 | 1 | 2 | 3 | **5** | **4** | 6 | 7 | 8 | 9 | 10 | 175.90 | (4,5) |
| N5 | 1 | 2 | 3 | 4 | **6** | **5** | 7 | 8 | 9 | 10 | 241.39 | (5,6) |
| N6 | 1 | 2 | 3 | 4 | 5 | **7** | **6** | 8 | 9 | 10 | 221.26 | (6,7) |
| N7 | 1 | 2 | 3 | 4 | 5 | 6 | **8** | **7** | 9 | 10 | 224.11 | (7,8) |
| N8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **9** | **8** | 10 | 239.71 | (8,9) |
| **N9** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | **10** | **9** | **175.65** | **(9, 10)** |

**TABLE 7 |** The neighborhood population $P_N^{(1)}$.

| String | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 | $T_i$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *10* | *9* | 175.65 |
| N1 | **2** | **1** | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 9 | 224.45 |
| N2 | 1 | **3** | **2** | 4 | 5 | 6 | 7 | 8 | 10 | 9 | 163.45 |
| N3 | 1 | 2 | **4** | **3** | 5 | 6 | 7 | 8 | 10 | 9 | 202.19 |
| N4 | 1 | 2 | 3 | **5** | **4** | 6 | 7 | 8 | 10 | 9 | 135.60 |
| N5 | 1 | 2 | 3 | 4 | **6** | **5** | 7 | 8 | 10 | 9 | 201.09 |
| N6 | 1 | 2 | 3 | 4 | 5 | **7** | **6** | 8 | 10 | 9 | 180.96 |
| N7 | 1 | 2 | 3 | 4 | 5 | 6 | **8** | **7** | 10 | 9 | 187.61 |
| N8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | **10** | 9 | 177.58 |
| N9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | **9** | **10** | 215.95 |

**TABLE 8 |** Find the next solution $S_2$.

| String | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 | $T_i$ | TM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *10* | *9* | 175.65 | - |
| N1 | **2** | **1** | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 9 | 224.45 | (1,2) |
| N2 | 1 | **3** | **2** | 4 | 5 | 6 | 7 | 8 | 10 | 9 | 163.45 | (2,3) |
| N3 | 1 | 2 | **4** | **3** | 5 | 6 | 7 | 8 | 10 | 9 | 202.19 | (3,4) |
| **N4** | 1 | 2 | 3 | **5** | **4** | 6 | 7 | 8 | 10 | 9 | **135.60** | **(4,5)** |
| N5 | 1 | 2 | 3 | 4 | **6** | **5** | 7 | 8 | 10 | 9 | 201.09 | (5,6) |
| N6 | 1 | 2 | 3 | 4 | 5 | **7** | **6** | 8 | 10 | 9 | 180.96 | (6,7) |
| N7 | 1 | 2 | 3 | 4 | 5 | 6 | **8** | **7** | 10 | 9 | 187.61 | (7,8) |
| N8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **10** | **8** | 9 | 177.58 | (8,10) |
| N9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | **9** | **10** | 215.95 | (10,9) |

tabu list and its objective value are as follows.

$$N4 = (1, 2, 3, 5, 4, 6, 7, 8, 10, 9)$$
$$T = 135.60 \text{ (h)}$$

This string is selected as the next solution for the next iteration, if any.

$$S_2 = N4 = (1, 2, 3, 5, 4, 6, 7, 8, 10, 9)$$

The tabu list is updated as follows.

$$TL = \{(9, 10), (4, 5)\}$$

The objective value of the best string in this iteration is 135.60, smaller than the current best value, so Tbest is updated as follows.

$$\text{Tbest} = 135.60 \text{ (h)}$$

**Step 5. Check the termination rule**

After iteration 2, $S_2$ is the best string with the best objective value of 135.60, appearing only once. The termination rule is not satisfied, so iteration 3 is executed.

**Iteration 3**

**Step 3: Generate the neighborhood population $P_N^{(2)}$**

This step uses the neighborhood operator SWAP to generate the neighborhood population $\mathbf{P}_N^{(2)}$ from the current string $S_2$.

$$P_N^{(2)} = \{N1, N2, N3, N4, N5, N6, N7, N8, N9\}$$

The strings in neighborhood population $\mathbf{P}_N^{(2)}$ and their objective values are shown in Table 9.

**Step 4. Find the next solution $S_3$**

This step uses the search operator to find the next solution $S_3$ that is the best solution in the neighborhood population $\mathbf{P}_N^{(2)}$, and does not violate the current tabu list. The strings in the neighborhood population $\mathbf{P}_N^{(2)}$, and their corresponding tabu member are shown in Table 10.

**TABLE 9 |** The neighborhood population $P_N^{(1)}$.

| String | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 | $T_i$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_2$ | *1* | *2* | *3* | *5* | *4* | *6* | *7* | *8* | *10* | *9* | 135.60 |
| N1 | **2** | **1** | 3 | 5 | 4 | 6 | 7 | 8 | 10 | 9 | 171.90 |
| N2 | 1 | **3** | **2** | 5 | 4 | 6 | 7 | 8 | 10 | 9 | 190.71 |
| N3 | 1 | 2 | **5** | **3** | 5 | 6 | 7 | 8 | 10 | 9 | 211.21 |
| N4 | 1 | 2 | 3 | **4** | **5** | 6 | 7 | 8 | 10 | 9 | 175.65 |
| N5 | 1 | 2 | 3 | 5 | **6** | **4** | 7 | 8 | 10 | 9 | 304.22 |
| N6 | 1 | 2 | 3 | 5 | 4 | **7** | **6** | 8 | 10 | 9 | 167.45 |
| N7 | 1 | 2 | 3 | 5 | 4 | 6 | **8** | **7** | 10 | 9 | 149.12 |
| N8 | 1 | 2 | 3 | 5 | 4 | 6 | 7 | **8** | **10** | 9 | 149.39 |
| N9 | 1 | 2 | 3 | 5 | 4 | 6 | 7 | 8 | **9** | **10** | 175.90 |

**TABLE 10 |** Find the next solution $S_3$.

| String | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 | $T_i$ | Tabu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N1 | **2** | **1** | 3 | 5 | 4 | 6 | 7 | 8 | 10 | 9 | 171.90 | (1,2) |
| N2 | 1 | **3** | **2** | 5 | 4 | 6 | 7 | 8 | 10 | 9 | 190.71 | (2,3) |
| N3 | 1 | 2 | **5** | **3** | 5 | 6 | 7 | 8 | 10 | 9 | 211.21 | (3,5) |
| N4 | 1 | 2 | 3 | **4** | **5** | 6 | 7 | 8 | 10 | 9 | 175.65 | (5,4) |
| N5 | 1 | 2 | 3 | 5 | **6** | **4** | 7 | 8 | 10 | 9 | 304.22 | (4,6) |
| N6 | 1 | 2 | 3 | 5 | 4 | **7** | **6** | 8 | 10 | 9 | 167.45 | (6,7) |
| **N7** | 1 | 2 | 3 | 5 | 4 | 6 | **8** | **7** | 10 | 9 | **149.12** | **(7,8)** |
| N8 | 1 | 2 | 3 | 5 | 4 | 6 | 7 | **10** | **8** | 9 | 149.39 | (8,10) |
| N9 | 1 | 2 | 3 | 5 | 4 | 6 | 7 | 8 | **9** | **10** | 175.90 | (10,9) |

**TABLE 11 |** The result after 27 iterations.

| Iteration | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 | T | Tbest |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 215.95 | 215.95 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 9 | 175.65 | 175.65 |
| 2 | 1 | 2 | 3 | 5 | 4 | 6 | 7 | 8 | 10 | 9 | 135.60 | 135.60 |
| 3 | 1 | 2 | 3 | 5 | 4 | 6 | 8 | 7 | 10 | 9 | 149.12 | 135.60 |
| ... | | | | | | | | | | | | |
| 17 | 1 | 3 | 2 | 4 | 5 | 10 | 8 | 7 | 9 | 6 | 123.07 | 123.07 |
| ... | | | | | | | | | | | | |
| 27 | 2 | 1 | 3 | 5 | 4 | 7 | 8 | 10 | 9 | 6 | 163.36 | 123.07 |

From Table 10, strings N4, and N9 in $\mathbf{P}_N^{(2)}$ violate the tabu list. The best string in the neighboring population $\mathbf{P}_N^{(2)}$ without violating the tabu list and its objective value are as follows.

$$N7 = (1, 2, 3, 5, 4, 6, 8, 7, 10, 9)$$
$$T = 149.12 \text{ (h)}$$

This string is selected as the next solution for the next iteration, if any.

$$S_3 = N7 = (1, 2, 3, 5, 4, 6, 8, 7, 10, 9)$$

The tabu list is updated as follows.

$$TL = \{(9, 10), (4, 5), (7, 8)\}$$

The objective value of the best string in this iteration is 149.12 bigger than the current best value 135.60, then Tbest remains the same.

$$\text{Tbest} = 135.60 \text{ (h)}.$$

**Step 5. Check the termination rule**

After iteration 3, $S_3$ is the best string in $\mathbf{P}_N^{(2)}$ with the objective value of 149.12, but $S_2$ is still the best string with the best objective value of 135.60, appearing only twice. The termination rule is not satisfied, so iteration 4 is executed.

The result after 27 iterations is as in Table 11.

The best objective value Tbest remains the same from the 17th iteration to the 27th iteration, the termination rule is satisfied, the algorithm ends. The best scheduling string and its objective value are as follows.

$$\begin{aligned} S &= (1, 3, 2, 4, 5, 10, 8, 7, 9, 6), \\ T &= 123.07\ (h) \end{aligned}$$

The objective value according to TS model (123.07 h) is better than the objective value according to the EDD heuristic currently used (215.95 h).

## Conclusion

The Flow Shop Scheduling Problem with 10 orders on 4 machines has been formulated with the objective of minimizing the total weighted tardiness time and constraint on changeover time in operations. The TS model has been developed and used to solve the problem. The results show that the TS model is better than the heuristic EDD method being used. The objective value has reduced by 43% from 215.95 (h) to 123.07 (h).

However, the factors of the model, including the method for defining the initial solution, factors of neighborhood operator, search operator, tabu list size, the method and parameter of the termination rule, are only selected empirically, so the results are not very good. Future research is to use experimental design DOE to determine the model factors to get better results. Further research is to use the model for larger numbers of orders.

Moreover, TS is a local search method, if it combines with another global search method like Genetic Algorithm, the result would be better in terms of quality, better objective value, and cost, smaller number of iterations.

## Author contributions

Both authors have made a substantial, direct, and intellectual contribution to the work, and approved it for publication.

## References

1. Nhu PN, Nhi T, Thi N. Application of TSGA, a hybrid mega heuristic model, to solve flow shop scheduling problems with changeover times in operations. A case study. *Proceeding of the 4 th International Conference on Advanced Convergence Engineering (ICACE 2023) August 14th – 16th, 2023, Ho Chi Minh City University of Technology, VNUHCM*, Ho Chi Minh City: (2023).
2. Umam MS, Mustafid M, Suryono S. A hybrid genetic algorithm and Tabu search for minimizing makespan in flow shop scheduling problem. *J King Saud Univ Comp Inf Sci.* (2022) 34:7459–67.
3. Gupta JND. Designing a tabu search algorithm for the two-stage flow shop problem with secondary criterion. *Prod Plann Cont Manage Oper.* (1999) 10:251–65. doi: 10.1080/095372899233217
4. Burduk A, Musiał K, Kochańska J, Górnicka D, Stetsenko A. Tabu Search and genetic algorithm for production process scheduling problem. *LogForum.* (2019) 15:181–9. doi: 10.17270/J.LOG.2019.315
5. Kolahan F, Tavakoli A, Tajdin B, Hosayni M. Analysis of neighborhood generation and move selection strategies on the performance of Tabu Search. *Proceedings of the 6th WSEAS International Conference on Applied Computer Science, Tenerife, Canary Islands, Spain, December 16-18*, Tenerife (2006). p. 503–7.