Università degli Studi di Brescia

Job Scheduling Optimization: Tabu Search implementation

Optimization Algorithms: Homework

Luca Pugnetti

A.Y.: 2024/2025

Problem Description

The problem involves determining the optimal sequencing of 6 jobs on a machine to minimize the global job tardiness. Each job has:

- p_j: the processing time of job j,
- d_j: the due date of job j,
- w_j: the penalty for tardiness of job j.

The objective is to minimize the total tardiness, calculated as:

$$T = \sum_{j=1}^{6} w_j [C_j - d_j]^+,$$

where C_j is the completion time of job j, and $[v]^+ = \max(0, v)$. The completion time C_j is the sum of the processing time p_j of job j, and the processing times of all jobs scheduled before job j. In the table (Table 1) the data for each job are collected.

Job	w_j	p_j	d_j
1	1	6	9
2	1	4	12
3	1	8	15
4	1	2	8
5	1	10	20
6	1	3	22

Table 1: Job data.

The goal is to determine the optimal job sequence that minimizes the total tardiness.

Optimal Solution with Gurobi

Before applying the Tabu Search algorithm, the optimal solution and job sequence were computed using the Gurobi solver, which provides an exact solution for this small instance of the job sequencing problem. The results from Gurobi are used as a benchmark to evaluate the performance of the Tabu Search algorithm.

Tabu Search: Key Concepts Implemented

Tabu Tenure

The tabu tenure is a parameter that determines how long a move remains in the tabu list. It is drawn randomly within a predefined range, for example:

$$\text{tabu\_tenure} \in [5, 30]$$

This value controls the memory of the algorithm, preventing the search from revisiting recently explored solutions.

Number of Iterations

The number of iterations is the total number of steps the algorithm will take. It can be set to a fixed value, such as 100:

$$\text{max\_iterations} = 100$$

At each iteration, the algorithm explores new solutions and updates the tabu list.

Objective Aspiration Criterion

The objective aspiration criterion allows a tabu move to be accepted if it improves the objective function. This criterion helps the algorithm escape local optima by relaxing the tabu restrictions:

If a tabu move improves the objective, it is accepted.

Default Aspiration Criterion

The default aspiration criterion is applied when all moves are tabu. In such cases, the algorithm will select the best tabu move:

If all moves are tabu, select the best tabu move (default aspiration).

This ensures the search continues even when no non-tabu moves are available.

Objective Aspiration Counter

The objective aspiration counter tracks how many iterations ago the objective aspiration criterion was last applied. It is decremented with each iteration and ensures that the objective aspiration criterion is applied periodically (to avoid oscillations):

$$\text{objective\_criteria\_counter} \leftarrow \text{tabu\_tenure}//2$$

Tabu Search Pseudocode

Algorithm 1 Tabu Search

Input: Jobs, Maximum iterations
Output: Best solution and its tardiness

Initialize a random *current_solution*
Set *best_solution* ← *current_solution*
Calculate *best_tardiness* for the initial solution
Initialize an empty *tabu_list*
Set *tabu_tenure* (randomly drawn from a range)
for each iteration **do**
    Generate a set of neighbors of *current_solution*
    Evaluate each neighbor (looking at the tardiness):
        Identify *best_non_tabu_move*
        Identify *best_tabu_move*
    if all moves are tabu **then**
        Select the best tabu move (default aspiration criterion applied)
    else
        if (t(*best_non_tabu_move*) < t(*best_tabu_move*)) or (objective aspiration recently applied) **then**
            Select *best_non_tabu_move*
        else
            Select *best_tabu_move* (objective aspiration criterion applied)
        end if
    end if
    Apply the selected move to update *current_solution*
    Update *tabu_list* with the selected move
    if *current_solution* improves the best tardiness **then**
        Update *best_solution* and *best_tardiness*
    end if
end for
return *best_solution*, *best_tardiness*

Results and Analysis

In this section, we present the results of 100 iterations of the tabu search algorithm applied to the given problem. Due to space limitations, are displaied only 18 textual iterations. The full exploration of 100 iterations is depicted in the graph, which visualizes the progression of the best and current tardiness over time. The graph includes up to 48 iterations, corresponding to three times the tabu tenure.

Gurobi Solution

The solution found by Gurobi is the following one:

Optimal sequence: [4, 1, 2, 3, 6, 5], Optimal tardiness: 19

Tabu Search results

The starting solution randomly selected is the following one:

Sequence: [6, 1, 5, 4, 2, 3], Tardiness: 44

The tabel below (Table 2) summarizes the first 18 iterations of the tabu search:

Iteration	Move Selected	Current Sequence	Tardiness	Tabu List Size	Objective aspiration criteria
1	[6, 4, 0, 3]	[4, 1, 5, 6, 2, 3]	31	1	No
2	[5, 2, 2, 4]	[4, 1, 2, 6, 5, 3]	23	2	No
3	[5, 3, 4, 5]	[4, 1, 2, 6, 3, 5]	21	3	No
4	[6, 3, 3, 4]	[4, 1, 2, 3, 6, 5]	19	4	No
5	[4, 1, 0, 1]	[1, 4, 2, 3, 6, 5]	19	5	No
6	[1, 4, 0, 1]	[4, 1, 2, 3, 6, 5]	19	6	No
7	[4, 1, 0, 1]	[1, 4, 2, 3, 6, 5]	19	6	No
8	[3, 6, 3, 4]	[1, 4, 2, 6, 3, 5]	21	7	Yes
9	[3, 5, 4, 5]	[1, 4, 2, 6, 5, 3]	23	8	No
10	[6, 5, 3, 4]	[1, 4, 2, 5, 6, 3]	23	9	No
11	[5, 3, 3, 5]	[1, 4, 2, 3, 6, 5]	19	10	No
12	[1, 2, 0, 2]	[2, 4, 1, 3, 6, 5]	22	11	No
13	[2, 1, 0, 2]	[1, 4, 2, 3, 6, 5]	19	12	No
14	[4, 2, 1, 2]	[1, 2, 4, 3, 6, 5]	23	13	No
15	[2, 4, 1, 2]	[1, 4, 2, 3, 6, 5]	19	14	No
16	[1, 4, 0, 1]	[4, 1, 2, 3, 6, 5]	19	15	No
17	[1, 2, 1, 2]	[4, 2, 1, 3, 6, 5]	22	15	Yes
18	[2, 1, 1, 2]	[4, 1, 2, 3, 6, 5]	19	16	No

Table 2: First 18 iterations of the Tabu Search algorithm

- In iterations 7 and 16, the objective aspiration criteria are applied. These iterations involve a move that would normally be considered tabu, but it is accepted because it results in an improvement. It is important to note that in these cases, the size of the Tabu List does not increase due to the selection of a tabu move.
- In iteration 18, the Tabu List is filled up (the length of the Tabu List reaches the tabu tenure value), so the first move selected in iteration 1 ([6, 4, 0, 3]) will become available again (the Tabu List is built from a FIFO perspective).
- It can be observed that the best tardiness value found by the metaheuristic is 19 (as found by Gurobi), given by the optimal sequences: [4, 1, 2, 3, 6, 5] and [1, 4, 2, 3, 6, 5] (as found by Gurobi).

Graphical Representation

The graphical representation below (Figure 1) displays the evolution of the best tardiness (best solution) and current tardiness (current solution) over the course of 48 iterations, which is three times the tabu tenure. The graph helps to visualize the structure of the solution, showing the behavior of the algorithm as it searches for better solutions over time.

The graph shows two curves: one for the best tardiness found during the search and another for the current tardiness at each iteration. This comparison allows us to see how the algorithm is improving its solution, with the best tardiness steadily decreasing over time. However, the current tardiness may fluctuate, reflecting the dynamic nature of the tabu search process, especially when certain moves are tabu or involve applying the aspiration criteria.

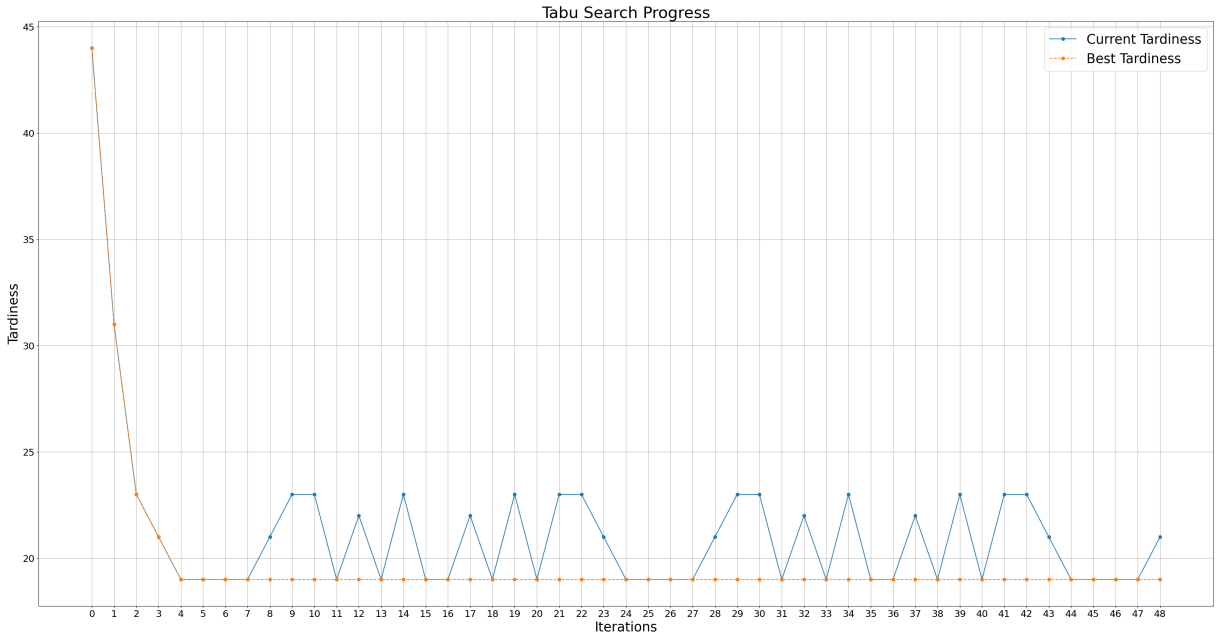

Figure 1: Progression of best and current tardiness over 48 iterations

Objective Aspiration Criteria in Iterations

In some iterations, such as iteration 7 and 16, the objective aspiration criteria are applied. This means that the algorithm is willing to accept a move that would typically be tabu if it leads to a better solution. This mechanism is crucial in escaping local optima, allowing the algorithm to explore potentially better solutions even when it involves revisiting previously considered solutions. The textual iterations show when this criterion is applied, as indicated in the respective moves.

Conclusion

From the analysis of both the textual and graphical results, it is evident that the tabu search algorithm is progressively improving the tardiness by applying moves strategically. . The application of the aspiration criteria in specific iterations ensures that the search process remains robust and able to escape local minima.