TP2 - The Quadratic Assignment Problem

Metaheuristics for Optimisation

[redacted]

October 15, 2020

1 Introduction

1.1 Quadratic Assignment Problem (QAP)

The Quadratic Assignment Problem is an NP-hard combinatorial optimization problem. An example of QAP is finding the best way to assign a set of n facilities to set of n locations given distances and flows where the distance-flow needs to be minimized. The research space S of such a problem is the set permutations which is of size $n!$. Given the problem of facilities and locations, in order to find the best location for each facility, we need to minimize the fitness $I(\psi)$:

$$I(\psi) = \sum_{i,j=0}^{n-1} w_{i,j} * d_{\psi_i} d_{\psi_j}$$

In our problem, we are given a set of 12 factories, to be placed in 12 locations. Our considered neighborhood is the set of permutations of two elements (two-swap) which yields $\frac{n(n-1)}{2}$ neighbors (66 neighbors).

1.2 Tabu Search

In our report, we study the effect of Tabu-Search in solving our Quadratic Assignment Problem of assigning 12 factories to 12 locations. Tabu-Search is a metaheuristic approach used to solve problems such as the QAP where the search space is explored by going from a neighborhood to another with an arbitrary starting state. The specificity of this concept is that there are "tabu" or forbidden states that we can't access until after a certain number of iterations (stated by the specified short-term memory). It is also possible to introduce a long-term memory (diversification) in order to explore more neighborhoods and finally reach our global fitness. Figure 1 shows the flow chart of Tabu-Search.

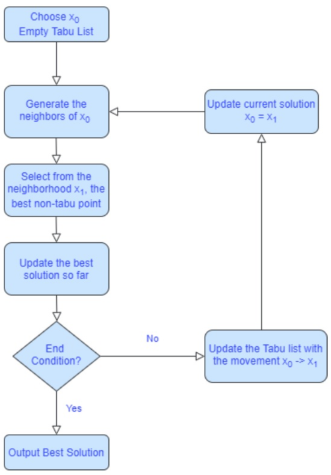


Figure 1: Tabu Search Flow Chart

2 Methodology

In our experiment, we study the effect of short term (using tenure) and long term (using diversification) memory of tabu-search on our QAP of 12 facilities assignment to 12 locations.

Our algorithm works as follows:

1. We start with a random assignment of 12 facilities over 12 locations.

2. We initialize the diversification matrix elements to n^2 and tabu-list elements to zeros. The diversification matrix shows if a facility i has been placed in location r during the past $u = n^2$ iterations. The tabu-list shows our tabu/forbidden permutations, i.e. a facility i is forbidden to be assigned to a location r if in the past l iterations, the facility had been removed from that location.

3. We compute the fitness $I(\psi)$ of our initial generated state x_0 and set it to the global fitness

$$I(\psi) = \sum_{i,j=0}^{n-1} w_{i,j} * d_{\psi_i} d_{\psi_j}$$

   where d and w are the distance and weight matrices, respectively.

4. The neighborhood of the state is the set of permutations of two elements (two-swap) which gives us $\frac{n(n-1)}{2}$ neighbors, which in our case is equal to 66. In order to choose a neighbor x_1, we calculate the cost of each swap (ψ_i, ψ_j) with respect to our x_0. We follow the following formula:

$$\Delta(\psi, i, j) = 2 \sum_{k \neq i,j} (w_{j,k} - w_{i,k}) * (d_{\psi_i,\psi_k} - d_{\psi_j,\psi_k})$$

5. In case of diversification mechanism, if the diversification matrix contains facilities that have not been assigned to certain location, we filter our possible neighbors to only those facilities/location permutations.

6. We select our best permutation (one with the least cost), and check if it abides by the tabu-search condition. We calculate the fitness (x_0 fitness + cost of swap), and check the following:

   (a) If the fitness is less than the global fitness, we choose this neighbor, even if is a forbidden state in the tabu list.

   (b) If the best x_1 is an allowed move in the tabu list, we choose it to be the next neighbor. Otherwise, we choose our next best neighbor and go back to (a).

7. After the selection of neighbor x_1, we set $x_0 = x_1$, and update the tabu list (set set (r, i) and (s, j) to l + iteration) and diversification matrix (decrement all elements by 1 and set (r, i) and (s, j) to n^2).

8. We loop over the steps until we reach a stopping criterion (t_{max}), and return our global fitness $I(\psi)$

We run the algorithm 10 times for each tenure $l = \{1, 0.5n, 0.9n\}$ and observe the results.

3 Results and Discussion

After running the algorithm 10 times for each tenure l, we record the best global fitness $I(\psi)$, the mean, and standard deviation for each loop. We then observe the effect of tenure l, diversification mechanism, and t_{max} on the results.

Effect of Tenure l

We vary the tenure l, which is the short-term memory of the tabu search. In this setting, the tabu list contains the facility-location pairs that have been visited for the last l iterations. These solutions are excluded from the neighborhood selection.

As shown in Table 1 and Figure 2, as tenure l increases, we are more likely to reach the global fitness, and the solutions are closer to each other. Increasing the tenure permits to reach untouched points and not get stuck in a loop of neighbors which may yield a local minimum. Increasing l thus allows us to [redacted] where there may [redacted], even if the immediate marginal improvement in fitness is not optimal. We thus observe that, as l increases, [redacted] At 0.9n, we could reach the global minimum with a standard [redacted]

Table 1: t_{max}=20,000, without diversification

Tenure(l)	Best Fitness	Mean Fitness	Standard Deviation
1	586	608.2[redacted]	
0.5n	586	592.8	[redacted]
0.9n	578	586.2[redacted]	

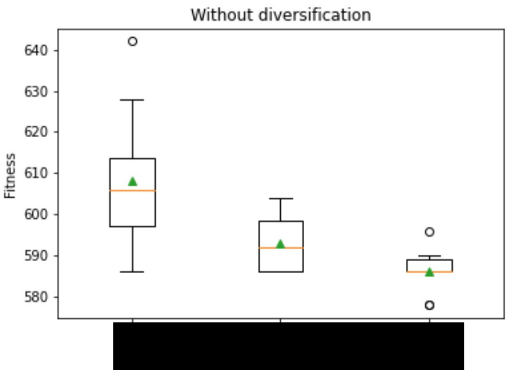


Figure 2: Tabu QAP Search Without Diversification

Effect of Diversification Mechanism

We later study the effect of diversification (long-term memory for tabu list). Diversification [redacted](exploration versus exploitation). Thus, it [redacted] the global minimum instead of getting stuck at a local one. As is clear in Table 2 and Figure 3, we reached the global minimum after [redacted] for all tenure l. Thus, with diversification, the search results are more consistent, and the search operation is hence *precise*. As we observe here and verify in the next section, diversification also leads to a faster convergence (finds the global minimum in less iterations).

Table 2: t_{max}=20,000, with diversification

Tenure(l)	Best Fitness	Mean Fitness	Standard Deviation
[redacted]			
[redacted]			
[redacted]			

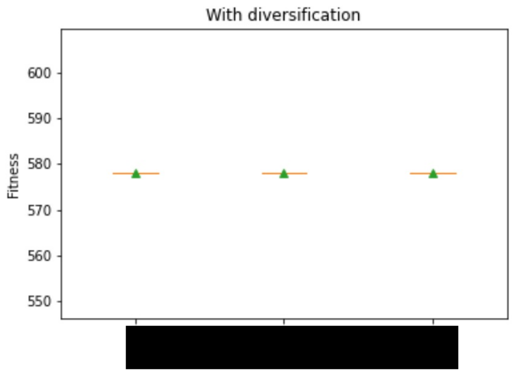


Figure 3: Tabu QAP Search With Diversification

Effect of Varying t_{max}

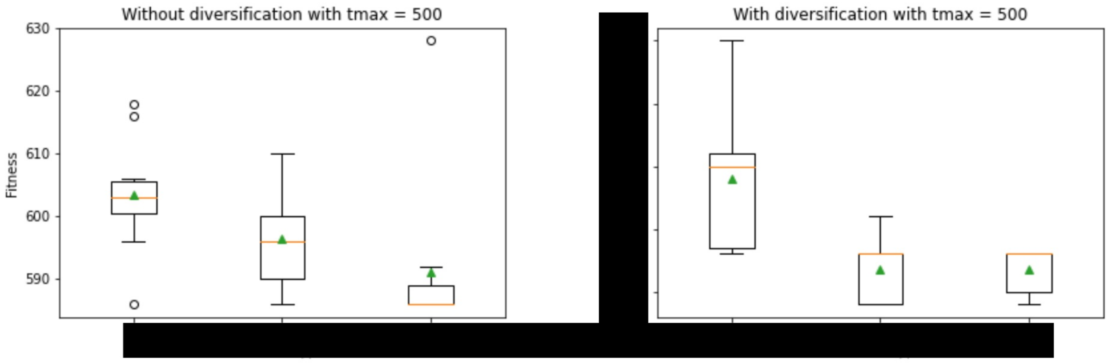
To better understand the convergence of tabu search, we study the effect of varying t_{max} on the obtained results. As shown in Figure 4, we observe that the higher t_{max} (number of iterations for each initial facility state), the more consistent and closely grouped the results are, and the more probable that we reach the global minimum. This effect is more pronounced in the absence of diversification, since the tabu search requires many iterations to escape local neighborhoods. We notice that, at t_{max} [redacted], the global minimum is not reached, whereas at t_{max} [redacted], it is. The tabu search does not converge quickly and yields inconsistent results.

However, once we enable diversification, we notice that the performance of the search quickly improves, and by t_{max} [redacted] it almost always finds the global minimum. We deduce that diversification results in faster convergence.

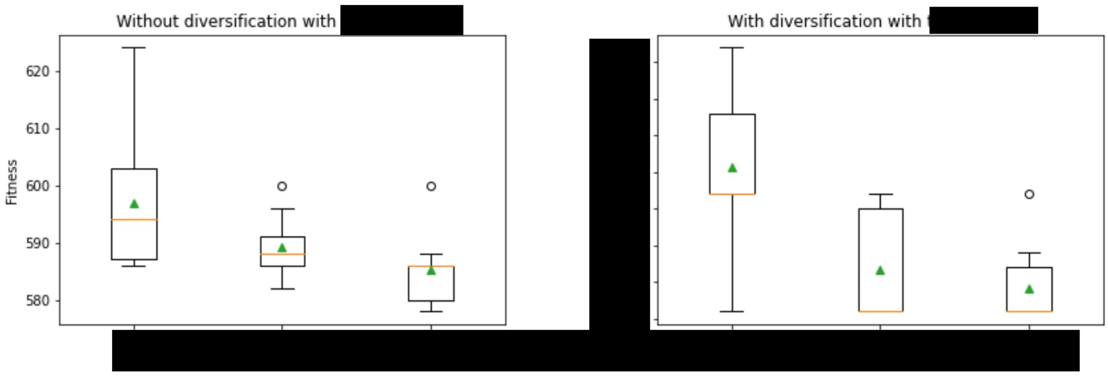
4 Conclusion

The Tabu Search is a metaheuristic approach that has been proved to help solve the Quadratic Assignment Problem, with the use of a tabu list with tenure l and diversification mechanism. The short-term memory allows the exploration and re-visiting of near neighbors which can be described as an intensification mechanism. On the other hand, [redacted] The tabu search also follows an aspiration criteria [redacted]

(a) $t_{max} = 500$



(b) t_{max} [redacted]



(c) t_{max} [redacted]

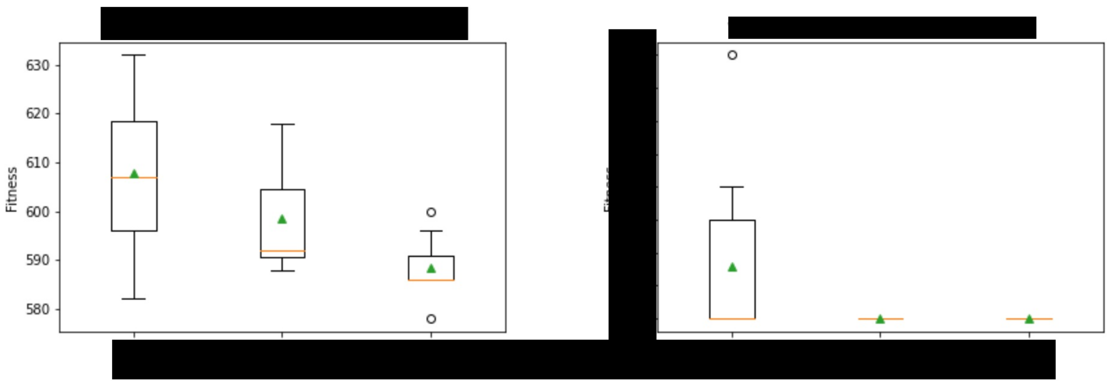


Figure 4: Tabu QAP Search with Variable t_{max}