

### 0.1. Complexity analysis

These are the steps to solve the ACLP+RPDP with the complexity analysis:

#### 0.1.1. Input data

Load  $K$  lines of text to compose a distances matrix for  $K$  airports, the  $m$  packed contents from previous nodes, and  $n$  items parameters from text files are of complexities  $O(K)$ ,  $O(m)$ , and  $O(n)$ , respectively:

- $O(K + m + n)$

#### 0.1.2. Number of tested tours ( $T$ )

- If  $K \leq 6$ , determine the number of tours through permutation. To get permutation of nodes as tours has a complexity of  $O(T = K! \cdot K)$ , as the number of nodes is permuted (factorial) to generate tours.

–  $O(K + m + n) + O(T = K! \cdot K)$  or

–  $O(K + m + n) + O(T = 2)$ , if the we are solving the two optimal TSP tours.

- If  $K > 6$ , we obtain 100 tours from a heuristic TSP solution (see its time complexity analysis in subsubsection 0.1.6):

–  $O(K + m + n) + O(T = 100)$

#### 0.1.3. Solving $T$ tours:

- $O(K + m + n) + O(T)$

a) each tour with  $K$  nodes:

- $O(K + m + n) + O(T \cdot K)$

b) in each node we have to reset (to eliminate the risk of previous destinations assigned to a pallet)  $O(K \cdot m)$  and set the empty pallets destinations:  $O(K \cdot m + K \cdot m) = O(2 \cdot K \cdot m) = O(K \cdot m)$

- $O(K + m + n) + O(T \cdot K \cdot (K \cdot m))$

c) To calculate the time limit to solve each node, it is necessary to determine the potential total volume for each node.:  $O(K \cdot n)$

- $O(K + m + n) + O(T \cdot K \cdot (K \cdot m + K \cdot n))$

d) The time complexity to put the packet contents destined to the next nodes in the tour on board:  $O(m)$ .

- $O(K + m + n) + O(T \cdot K \cdot (K \cdot m + K \cdot n + m))$

d) The time complexity to optimize the pallets positions to minimize the CG deviation:  $O(m^2)$  ( $m$  pallets in  $m$  positions)

- $O(K + m + n) + O(T \cdot K \cdot (K \cdot m + K \cdot n + m + m^2))$

e) Plus the time complexity of the method used to solve an ACLPP with the aircraft partially loaded:  $O(C)$

- $O(K + m + n) + O(T \cdot K \cdot (K \cdot m + K \cdot n + m + m^2 + C))$

#### 0.1.4. Considering the Shims algorithm: $O(C) = O(Shims)$

a) the time complexity for sorting the pallets in non-descending order of distances to the CG:

- $O(Shims) = m \cdot \log(m)$

b) As each pallet is solved at a time, the overall Shims complexity is multiplied by  $m$ .

- $O(Shims) = m \cdot (m \cdot \log(m))$

c) The greedy phase has  $O(n)$ .

- $O(Shims) = m \cdot (m \cdot \log(m) + n)$

d) The composition phase has  $O(n)$  added to the First-Fit Decreasing algorithm time complexity  $O(m \cdot \log(m))$

- $O(\text{Shims}) = m \cdot (m \cdot \log(m) + n + n + m \cdot \log(m))$
- $O(\text{Shims}) = m \cdot (2n + 2 \cdot m \cdot \log(m)) = m \cdot (n + m \cdot \log(m))$

The overall time complexity to solve the ACLP+RPDP with Shims is:

- $O(K + m + n) + O(T \cdot K \cdot (K \cdot m + K \cdot n + m + m^2 + m \cdot (n + m \cdot \log(m))))$

The overall time complexity provided combines two distinct parts:  $O(K + m + n)$  and  $O(T \cdot K \cdot (K \cdot m + K \cdot n + m + m^2 + m \cdot (n + m \cdot \log(m))))$ . Analysing each part separately and then considering their combined implications provides insight into the algorithm's performance characteristics.

#### 0.1.5. Complexity analysis considerations

**First Part:**  $O(K + m + n)$

This component of the time complexity suggests a linear dependency on the sizes of three separate inputs or parameters, denoted by  $K$ ,  $m$ , and  $n$ . These are sequential steps, each influenced by each of these parameters size. This implies that as any of  $K$ ,  $m$ , or  $n$  increases, the time required by the algorithm increases linearly with respect to the largest of these parameters, the number of items to be transported ( $n$ ).

**Second Part:**  $O(T \cdot K \cdot (K \cdot m + K \cdot n + m + m^2 + m \cdot (n + m \cdot \log(m))))$

This portion is considerably more complex, indicating an algorithm with nested dependencies on several variables ( $T$ ,  $K$ ,  $m$ , and  $n$ ):

- $T \cdot K$ : repeated iterations  $T$  for each of  $K$  nodes, nested loops where inner operations are contingent upon  $K$ .
- $K \cdot m + K \cdot n$ : These operations scale with both  $K$  and each of  $m$  and  $n$ , indicating processes that involve all pairs of  $K$  with  $m$  and  $n$ .
- $m + m^2$ : operations with linear and quadratic dependencies on the number of pallets  $m$ , indicating that as  $m$  grows, the complexity increases more significantly due to the quadratic term  $m^2$ .
- $m \cdot (n + m \cdot \log(m))$ : Combines linear scaling with  $m$  and  $n$  and a logarithmic term  $m \cdot \log(m)$ , because the algorithm uses divide-and-conquer strategies on  $m$  elements, the Shims heuristics.

#### Combined Implications

When combining these parts, the linear terms in the first part ( $O(K + m + n)$ ) are overshadowed by the more complex and potentially rapidly increasing terms of the second part, especially due to the presence of  $m^2$  and the product  $T \cdot K \cdot (...)$ . The quadratic ( $m^2$ ) and logarithmic ( $m \cdot \log(m)$ ) terms indicate that the algorithm's efficiency decreases significantly as the number of pallets  $m$  increases.

The overall complexity suggests an algorithm that performs efficiently with small values of  $K$ ,  $m$ , and  $n$  but may become substantially more resource-intensive as these parameters increase. The number tours  $T$  indicates that the entire complex operation is executed  $T$  times, further amplifying the effect of the inner complexities.

Optimizing such an algorithm involves minimizing the repetition count ( $T$ ), reducing the complexity of operations dependent on  $K$ ,  $m$ , and  $n$ .

#### 0.1.6. Time Complexity Analysis of the Genetic Algorithm for TSP

When  $K > 6$ , we obtain 100 tours from a heuristic TSP solution with the Genetic Algorithm (GA), which addresses optimization problems by evolving a population of candidate solutions through selection, crossover (recombination), and mutation over multiple generations, aiming to improve the fitness of solutions.

The time complexity of applying Genetic Algorithms to solve the TSP is influenced by several key factors, including the number of nodes ( $K$ ), population size ( $P$ ), number of generations ( $G$ ), and the complexities of the crossover ( $C(K)$ ), mutation ( $M(K)$ ), and selection ( $S(K)$ ) operations.

Given these parameters, we can express the overall time complexity as:

- $O(G \cdot (P \cdot (S(K) + P \cdot (C(K) + M(K))))))$

#### Detailed Explanation

- The selection operation's complexity,  $S(K)$ , may depend on evaluating the fitness of each solution, which in turn relies on computing the total distance of a path through  $K$  nodes.
- The complexities of the crossover and mutation operations,  $C(K)$  and  $M(K)$ , are functions of  $K$  because they involve manipulating sequences representing paths through the  $K$  nodes.
- Each generation involves selecting solutions based on fitness, generating new solutions through crossover, and applying mutations, leading to the multiplication by  $G$ .

#### **Impact of the number of nodes ( $K$ )**

- The direct involvement of  $K$  in the complexities of  $C(K)$ ,  $M(K)$ , and  $S(K)$  highlights how the number of nodes affects the algorithm's performance. Specifically, operations that manipulate or evaluate paths become more complex as  $K$  increases.
- The encoding of solutions and the calculation of fitness values, essential for the selection process, inherently depend on  $K$ , emphasizing its role in the GA's computational demands.