Mixed Integer Linear Formulation for the Multiple Trip Aircraft Refueling Problem of a Brazilian Company

Karyne Alves Zampirolli[1(✉)] and André Renato Sales Amaral[2]

[1] Graduate School of Computer Science (PPGI), Federal University of Espírito Santo, Vitória ES 29075-910, Brazil
karyne.zampirolli@ufes.br

[2] Graduate School of Computer Science (PPGI), Federal University of Espírito Santo, Vitória ES 29075-910, Brazil
amaral@inf.ufes.br

Abstract. This work presents a study on a problem faced by a Brazilian fuel distribution company, which aims to develop a decision system to minimize the operational costs of tanker trucks that refuel aircraft at airports. This is called the Aircraft Refueling Problem (ARP). ARP is related to the vehicle routing problem but focuses on the practical aspects of actual operations. The ARP variant discussed allows trucks to make multiple trips, which is called the Multiple Trip Aircraft Refueling Problem (MTARP). A Mixed-Integer Linear Programming Model (MILP) was developed to address the problem. A diverse set of instances was generated based on real-world operational constraints, simulating refueling scenarios at Brazilian airports. The model was solved using the CPLEX solver, and computational experiments were conducted to evaluate solution quality under different time limits. Two key analyzes were performed: (1) identifying instance characteristics that increase problem complexity and (2) assessing solution quality within a 30-min time limit. For smaller instances, CPLEX quickly finds optimal solutions, while for larger, constrained cases, optimality gaps remain significant. These findings offer insights into MTARP's scalability and demonstrate that even under strict computational limits, CPLEX can provide useful solutions. However, for highly complex instances, hybrid approaches, such as heuristics or decomposition methods, may be necessary. These findings offer valuable guidance for optimizing aircraft refueling operations in real-world applications.

Keywords: Mixed-integer linear programming · Aircraft refueling problem · Vehicle routing

1 Introduction

At Brazilian airports, tanker trucks are typically used to refuel aircraft before they take off for their next destination. Poor management of this refueling service can lead to flight delays. A Brazilian company, which is the country's main

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15887, pp. 163–180, 2026.
https://doi.org/10.1007/978-3-031-97589-9_12

aviation fuel distributor and responsible for the refueling service, aims to plan routes for its tanker trucks, ensuring that aircraft receive fuel within predetermined time windows, minimizing the total distance traveled by the trucks. This is called the Aircraft Refueling Problem (ARP). ARP is a type of Vehicle Routing Problem (VRP) [5,10,11,17,23,27]. Finding the optimal solution to a real-size vehicle routing problem is a complex task since these problems are NP-hard [8]. Thus, depending on the size of the problem, no optimization algorithm can achieve the optimal solution of the problem in a feasible time.

Although the ARP is related to the vehicle routing problem, ARP differs by considering the unique characteristics of a real airport refueling operation. This distinction makes ARP a new and promising object of study. When reviewing the literature, no studies were found that completely represented the Brazilian company's aircraft refueling problem. A few authors have examined the use of tanker trucks for aircraft fuel supply service [3,9,12,15,18,20,21,24,25,28] but the problems they considered differ from the ARP in some objectives or by not including ARP assumptions, such as the aircraft refueling rate, the average speed of the trucks, the rate of reloading trucks at the depot, the fuel sedimentation time after truck reloading at the depot for a new trip, heterogeneous fleet, aircraft being serviced by a single truck without compatibility restrictions, and trucks without maximum travel distance restrictions, among other factors.

When dealing with the ARP, we can investigate two variants: the Single Trip Aircraft Refueling Problem (STARP) and the Multiple Trip Aircraft Refueling Problem (MTARP). STARP was introduced in [26], where the authors propose a mathematical model and heuristic methods to solve it. The STARP focuses on the refueling service of aircraft using a fleet of non-homogeneous tanker trucks with capacity and time window constraints, where each truck makes only one trip within a given time horizon. This problem is related to the Vehicle Routing Problem with Time Windows (VRPTW) [7]. On the other hand, the MTARP allows trucks to make multiple trips, requiring them to return to the depot to refuel before starting new trips, and is, therefore, closely related to the Multi-Trip Vehicle Routing Problem with Time Windows (MTVRPTW) [2]. However, STARP and MTARP add application-specific restrictions.

In this paper, we will address the MTARP variant. We present a Mixed-Integer Linear Programming Model (MILP) for the MTARP and test it on a set of instances constructed based on several airport scenarios for the study of the ARP. The instances span two distinct categories: one where aircraft are parked in a straight line and another where they are not.

The main contributions of this article are: the introduction of a practical problem of a fuel distribution company, called MTARP; the development of a MILP formulation tailored to the operational constraints and objectives of the company under study; the generation of a diverse set of instances to analyze problem complexity under different scenarios; a computational analysis identifying key instance characteristics that impact problem solvability; insights into the model's scalability, emphasizing its effectiveness for small- and medium-scale problems and its limitations in solving large, highly constrained instances within practical timeframes.

The paper is organized as follows: Sect. 2 presents a brief literature review of studies related to the aircraft fuel supply service and the MTVRPTW problem. Then, Sect. 3 presents the mathematical model proposed for the MTARP. Section 4 describes the characteristics of the instances considered in the study. Computational results are presented in Sect. 5. Finally, some conclusions and suggestions for future research are presented in Sect. 6.

2 Literature Review

This section presents a brief literature review on research related to aircraft fuel supply operations and MTVRPTW.

2.1 Aircraft Fuel Supply Service

The ARP is innovative because, after reviewing the literature, no previous studies have been found to address this problem fully. Some studies address the aircraft fuel supply service by tanker trucks [3,9,12,15,18,20,21,24,25,28], but they differ from the ARP as they do not include some of the ARP assumptions outlined in Sect. 1.

One of the first studies on aircraft fueling service was carried out in [3], where a model suitable for Belgrade airport was presented. The main objective in [3] was to minimize both the number of tanker trucks and the total distance traveled by these trucks. The problem was approached using the *branch-and-bound* technique. Reference [3] assumes that flights are served in increasing order of service start time, that all tanker trucks have the same capacity, and that each aircraft can be served by up to two vehicles, each providing half the required amount of fuel. However, in ARP, the flight service order is not fixed, tanker trucks have different capacities and finally, an aircraft cannot be serviced by more than one truck at the same time.

In [9] a mathematical formulation for the optimization problem known as Airport Ground Service Scheduling (AGSS) is presented. The objective is to minimize several factors, such as the number of tanker trucks used, the sum of the service start times for each flight, and the total travel time of the trucks. To illustrate the problem, the authors considered the Oil Tanker Truck Scheduling (OTTS) problem. The objective of OTTS is to obtain a feasible solution that assigns tanker trucks to fuel aircraft and provides the start time of each service operation. In this formulation, preference was given to fueling aircraft with earlier service start times, shorter travel times, shorter waiting times, shorter service times, and earlier service end times. In addition, the study considered the use of non-homogeneous vehicles, where each could serve up to $G(\in \mathbb{Z})$ aircraft. In ARP, unlike this work, a truck can serve as many aircraft as needed, as long as it does not violate the vehicle's capacity. As for the solution method, the authors used an Ant Colony Optimization (ACO) algorithm. A heuristic called Earliest Due Date First (EDD) was incorporated into ACO as a comparative ant to

improve the algorithm's performance. A case study was conducted considering data from a domestic airport.

The study in [21] investigated the Dynamic Capacitated Vehicle Routing Problem with Time Windows, (partial) Split Delivery and Heterogeneous fleets (DVRPTWSD), applied to the aircraft fuel supply activity using trucks at an airport. The objective is to optimize response capacity, that is, to complete the fuel supply as quickly as possible within the time window, allowing the vehicle to return to its activities as quickly as possible. In constructing the routes of the tanker trucks, the problem considers that each vehicle has a specific capacity, a service time and a list of incompatibilities according to which some trucks are not suitable to serve certain aircraft. Furthermore, in the DVRPTWSD, a single aircraft can be served by different trucks. In the ARP, the trucks do not have a list of incompatibilities and an aircraft cannot be served by more than one truck. The author proposed an algorithm based on ant colony optimization to solve DVRPTWSD. This approach was tested in a real case in the air transport area, demonstrating the effectiveness of the proposed algorithm.

Reference [15] conducted a study on route scheduling for aircraft fuel supply at Juanda Airport in Surabaya. This was called Airport Ground Service Scheduling (AGSS). The authors assume that an aircraft can only be serviced by one truck. They also considered the use of non-homogeneous vehicles with a maximum service restriction of G aircraft. The objective of the study was to determine the minimum number of trucks required for the refueling service. To solve the problem, they developed a solution approach based on the ACO algorithm.

In [20], a mathematical model is presented to address the dynamic scheduling problem of fueling vehicles at airports, formulating it as a variant of the Vehicle Routing Problem with Time Windows (VRPTW). In the study, the authors considered tanker trucks with maximum travel distance constraints. Unlike the assumption of this work, in ARP, trucks do not have a travel distance limitation. The objective is to minimize both the number of service routes for refueling vehicles and the number of vehicles needed to complete the aircraft refueling service. To solve this problem, the authors proposed a hybrid multi-strategy approach. This approach combines a Genetic Algorithm (GA) with a crossover method inspired by particle swarm optimization and a local search strategy based on Simulated Annealing (SA). Additionally, they developed a local replanning strategy based on a dynamic time window. The effectiveness of this method was validated through a real case study carried out at an airport.

2.2 Multi-Trip Vehicle Routing Problem with Time Windows

In the Multi-Trip Vehicle Routing Problem with Time Windows (MTVRPTW) vehicles can be scheduled to make more than one trip during a working day, and the customers must be served by these vehicles within specific time windows. The need for multiple trips arises when the vehicle fleet is limited. The problem consists of optimizing the working day of each vehicle, aiming at minimizing transportation costs. The working day of a vehicle is made up of a sequence of

customers to be visited. The vehicle can return to the distribution center to be reloaded and then start a new trip. The maximum number of trips for a vehicle is predefined. Each customer must be served only once, within a predetermined time [1,2,4,13,16,19].

According to [1], MTVRPTW has received little attention in the literature, despite its importance in practice. Most existing studies on solving this problem have addressed the use of meta-heuristics. In the literature, the research in [13] is considered the first study to address the concept of multiple trips in the vehicle routing problem without time windows. The author used an economy algorithm to construct the routes and a heuristic for the *bin-packing* problem to combine them into vehicles. The proposed approach was used to solve examples between 68 and 361 customers.

In [4], the authors divide MTVRPTW into two easier problems and develop two heuristics to solve them. The first heuristic deals with the creation of a set of routes, while the second deals with the *bin-packing* problem, in which routes are aggregated into multiple trips. The complete algorithm is iterative and is based on an adaptive guidance strategy, which directs the route heuristic to calculate only routes capable of improving the solution.

In [22], the MTVRPTW is addressed in the context of a company that distributes loaded pallets to designated customers. They proposed a mathematical model that considers heterogeneous fleet, total route time, and accessibility constraints between vehicles and customers. As a solution method, the authors used a constructive heuristic and a meta-heuristic based on Tabu Search (TS). To evaluate the performance of the heuristic, they presented a column generation algorithm, integrating a *branch-and-bound* procedure.

The work in [6] surveyed mathematical formulations and research exploring exact and heuristic approaches in the context of MTVRP and its extensions. The study also addresses other variants of VRP, where multiple trips are sometimes allowed. In this scenario, the authors highlight the motivations for linking multiple trips to vehicles and discuss the consequences of this approach on solution methods.

3 Problem Description and Modeling

In this section, we will present a Mixed Integer Linear Programming (MILP) formulation for the problem of the Brazilian fuel distribution company, called the Multiple-Trip Aircraft Refueling Problem (MTARP).

3.1 MTARP Description

Consider an airport with a depot (distribution center - DC) that houses tanker trucks responsible for supplying fuel to aircraft. The MTARP is defined on a complete directed graph $G(N, A)$ where $N = N_A \cup \{0, n+1\}$. $N_A = \{1, 2, ..., n\}$ is the set of aircraft. 0 and $n + 1$ are the origin and destination vertices, both

corresponding to the depot. The set A is the set of arcs that join two distinct vertices of G. No arc ends at vertex 0 and no arc originates at vertex $n+1$.

Thus, MTARP can be described as follows: let N_A be the set of aircraft waiting for refueling service, K be the set of non-homogeneous tanker trucks and R be the set of trips, where $n = |N_A|$, $\bar{k} = |K|$ e $\bar{r} = |R|$. For each aircraft i that requires a quantity of fuel c_i $(i \in N_A)$, the refueling service must occur in the time interval $[a_i, b_i]$. A truck k, with capacity Q_k $(k \in K)$, cannot arrive to refuel aircraft i after time b_i. Furthermore, if the truck arrives before time a_i, it must wait until a_i to start the service. Each aircraft i can only be served once and by a single truck.

During aircraft refueling service, all tanker trucks are initially located at the depot. A preparation time for reloading a truck is associated with each route. This preparation time is proportional to the sum of the service times on each aircraft along the route. In addition, a fuel sedimentation time after truck reloading at the depot is considered. Each truck can service several aircraft. The truck leaves the depot and heads to aircraft i to refuel it, after which it can travel to another aircraft j or return to the depot to be reloaded and start a new trip. Thus, a route consists of a feasible sequence of trips $r \in R$, where each trip starts and ends at the depot. The maximum number of trips is predefined. The fuel demand for each trip cannot exceed the tanker truck's maximum load capacity. Trucks that are not used follow a fictitious route, starting and ending at the depot, without visiting any aircraft. The objective is to minimize the total distance traveled by the trucks. To formulate the problem as a MILP model, indices, parameters and decision variables are described in Table 1. The conditions to be considered are the following: $a_0 = a_{n+1} = 0$; $b_0 = b_{n+1} = b_{CD}$, $D_{0,n+1} = 0$.

3.2 MTARP Mathematical Model

Our model is based on the one proposed in [22]. However, our model does not consider two constraints used in [22]: the total route time constraint and the accessibility constraints between vehicles and customers. Another difference is the way of calculating the preparation time for reloading a truck at the depot for a new trip. In [22], the time for reloading the truck at the depot corresponds to an average vehicle loading time, regardless of the customers to be served along the route. An additional difference is that, in our model, we consider the average speed of the vehicle, the rate of reloading the vehicle at the depot, and for the calculation of the service time at the customer, the rate to refueling customer, which are metrics appropriate to the matter of aircraft refueling operations (e.g., [3]).

The MILP model proposed here for MTARP is formulated as follows:

Minimize:

$$\sum_{k \in K} \sum_{r \in R} \sum_{i \in N} \sum_{j \in N, j \neq i} D_{i,j} X_{i,j,k,r} \tag{1}$$

Subject to:

Table 1. Indices, parameters and decision variables of the MTARP model.

Indices	Description
i, j	aircraft
k	tanker trucks
r	trips
Parameters	Description
n	total number of aircraft
$\bar{k}$	total number of trucks
$\bar{r}$	maximum number of trips allowed for each truck
c_i	amount of fuel requested by aircraft $i \in N_A$
Q_k	maximum fuel capacity of truck $k \in K$
$D_{i,j}$	distance associated with arc $(i, j) \in A$
a_i	earliest start time for refueling aircraft i, $i \in N$
b_i	latest start time for refueling aircraft i, $i \in N$
q_i	rate to refueling aircraft i, $i \in N_A$
V_k	average speed of the truck k, $k \in K$
u	rate to reloading truck k at the depot, $k \in K$
t_σ	fuel sedimentation time after reloading truck k, $k \in K$
M	large positive number for time window constraints
Variables	Description
$X_{i,j,k,r}$	binary variables. Equal to 1 if edge $(i, j) \in A$ is traversed by truck $k \in K$ on trip $r \in R$. Otherwise, equal to 0. Note that when $X_{0,n+1,k,r} = 0$ trip r is empty
$T_{i,k,r}$	non-negative variables. They indicate the time when truck $k \in K$ on trip $r \in R$ starts serving node $i \in N$

$$\sum_{k \in K} \sum_{r \in R} \sum_{j \in N, j \neq i} X_{i,j,k,r} = 1, \quad i \in N_A \tag{2}$$

$$\sum_{i \in N_A} \sum_{j \in N, j \neq i} c_i X_{i,j,k,r} \leq Q_k, \quad k \in K, r \in R \tag{3}$$

$$\sum_{j \in N} X_{0,j,k,r} = 1, \quad k \in K, r \in R \tag{4}$$

$$\sum_{i \in N} X_{i,n+1,k,r} = 1, \quad k \in K, r \in R \tag{5}$$

$$\sum_{i \in N, h \neq i} X_{i,h,k,r} - \sum_{j \in N, j \neq h} X_{h,j,k,r} = 0, \quad h \in N_A, k \in K, r \in R \tag{6}$$

$$T_{j,k,r} \geq T_{i,k,r} + \frac{c_i}{q_i} + \frac{D_{i,j}}{V_k} - M(1 - X_{i,j,k,r}), \quad i,j \in N, i \neq j, k \in K, r \in R \quad (7)$$

$$a_i \leq T_{i,k,r} \leq b_i, \quad i \in N, k \in K, r \in R \quad (8)$$

$$T_{0,k,r} \geq \sum_{i \in N_A} \sum_{j \in N, j \neq i} \frac{c_i}{u} X_{i,j,k,r} + t_\sigma, \quad k \in K, r \in R \quad (9)$$

$$T_{0,k,r} \geq T_{n+1,k,r-1} + \sum_{i \in N_A} \sum_{j \in N, j \neq i} \frac{c_i}{u} X_{i,j,k,r} + t_\sigma, \quad k \in K, r = 2, ..., |R| \quad (10)$$

$$X_{0,n+1,k,r} \leq X_{0,n+1,k,r+1}, \quad k \in K, r = 1, ..., |R| - 1 \quad (11)$$

$$X_{i,j,k,r} \in \{0, 1\}, \quad i,j \in N, i \neq j, k \in K, r \in R \quad (12)$$

$$T_{i,k,s} \geq 0, \quad i \in N, k \in K, r \in R \quad (13)$$

The objective function (1) aims to minimize the total distance traveled by tanker trucks. Constraints (2) impose that an aircraft i can only be served by a truck k on a trip r. Constraints (3) state that the total demand on a route cannot exceed the vehicle capacity. In (4), it is imposed that every truck k must depart from each origin on the route. Similarly, constraints (5) impose that every truck k must arrive at each destination on the route.

Constraints (6) ensure that if a truck arrives at an aircraft, it must also depart from it. Constraints (4) to (6) ensure flow conservation, that is, they ensure that routes have their proper sequencing, starting and ending at the depot and visiting all aircraft. Constraints (8) impose that time windows are respected. Constraints (7) establish the relationship between the time a truck leaves an aircraft and the time the next aircraft is serviced. Note that these constraints ensure that when a truck k travels from aircraft i to aircraft j on a trip r, the arrival time at aircraft j must be greater than or equal to the time the truck began servicing aircraft i, plus the service time to refuel aircraft i (c_i/q_i), plus the time to travel from i to j ($D_{i,j}/V_k$). M is a large constant value.

Constraints (9) impose that the departure time of truck k on trip r is greater than or equal to the time required to reload the truck for it to make trip r plus the fuel sedimentation time after reloading the truck (t_σ). Furthermore, constraints (10) impose that the departure time of truck k at the origin node of trip r must be greater than or equal to the arrival time of truck k at the destination node of trip $r-1$, plus the time it takes to reload the truck for it to make trip r and the fuel sedimentation time, applicable only when $r \geq 2$. These constraints ensure proper route sequencing.

Constraint (11) ensures that trip $r+1$ is not taken if trip r has not been taken. Trips that service aircraft are always placed at the beginning of the trucks'

route, and empty trips are always taken at the end. This constraint only applies when $|R| \geq 2$, because if a truck k traverses the fictitious arc $(0, n+1)$ during trip r, then the fictitious arcs of subsequent trips must also be traversed. Finally, constraints (12) and (13) define the domain of the variables.

4 Instance Generation

To simulate real scenarios of Brazilian airports, sets of adapted instances were created. The creation of these instances was motivated by the fact that the ARP is a new problem and, thus, there is no instances available. The construction of these sets was based on premises identified through dialogues and questionnaires conducted with employees of the company studied and through data from the FlightAware platform [14]. FlightAware operates the largest flight data and tracking platform in the world, ensuring the quality of the data used.

The company studied does not have data on the location of aircraft parking lots. For this reason, it was necessary to seek this data from other sources to determine the distances. Due to the unavailability of real data, the fuel demand and time windows were estimated based on information collected on the platform FlightAware [14].

The constructed instance sets provide two scenarios for the MTARP investigation: the first set, called *APSL-MT type set*, has aircraft parked in a straight line. The second set, *ANPSL-MT type set*, introduces a scenario of aircraft not parked in a straight line. For the sets of instances with aircraft parked in a straight line, the positions were generated using the following methodology: the distance between the depot and the first aircraft parking position was considered. For each subsequent parking position, the distance of 40 m was added to the previous parking position, determining its position. This process resulted in a set of distances d_i, where $i \in N$, represents the distance between aircraft i and the depot. From d_i, the distance matrix $D_{i,j}$ was calculated as $D_{i,j} = |d_i - d_j|$. This calculation of $D_{i,j}$ follows the methodology described in the work of [3]. For the instance sets with aircraft not parked in a straight line, a distance matrix $D_{i,j}$ was constructed with random values between 99 and 1299.

The time windows $[a_i, b_i]$ represent the first and last moments in which the fueling operation of each aircraft i can occur. These intervals were defined based on data from Brazilian airport operations collected on the FlightAware platform, considering the scheduled departure time of each flight. The upper limit b_i of each window was defined as 10 min before the scheduled departure time, as per the company guidance. The lower limit a_i assumes values between 30 and 40 min before b_i, varying according to the airline, the boarding and baggage handling processes. For this study, a_i was set at 30 min before b_i.

The truck's service time to refuel each aircraft varies according to the amount of fuel requested and the refueling rate. This rate varies according to the type of aircraft, which can be 350, 500 or 700 liters per minute, according to the data from the company studied. However, in this study, only the values of 350 and 500 liters per minute were considered.

The average speed of the trucks was set at 20 km/h, although the maximum permitted speed is 30 km/h. The trucks used have different capacities— 7,500 liters, 13,000 liters, 20,000 liters, 40,000 liters and 50,000 liters— but availability differs across airports. The average amount of fuel requested by a given flight ranges from 800 liters to 24,000 liters.

The time required to reload a truck for a new trip is influenced by the amount of fuel to be loaded and the rate of reloading the truck, which is a pump with an average flow rate of 1,000 liters per minute. After the truck is finished loading, an additional 10 min is considered for the fuel sedimentation. The company reported that, on average, each truck makes three trips per day, which may vary during peak seasons. The value of three was set as the maximum limit for trips in this study.

Tables 2 and 3 provide details about the instance characteristics for MTARP. The first column indicates the name of the instances followed by the number of aircraft (n), the number of available trucks ($\bar{k}$), the number of allowed trips ($\bar{r}$), total demand (liters), demand characteristics, time window characteristics and characteristics of simultaneous arrivals of flights.

Table 2. Characteristics of the APSL-MT type set.

Instance	n	$\bar{k}$	$\bar{r}$	Total Demand (liters)	Demand Characteristics	Time Windows	Simultaneous Arrivals
A1	6	6	3	18,650	Less than truck capacity	Spaced	No
A2	11	6	3	33,650	Less than truck capacity	Spaced	No
A3	15	6	3	41,650	Less than truck capacity	Spaced	Yes
A4	18	6	3	50,100	Less than truck capacity	Some close	Yes
A5	18	6	3	106,850	Close to or exceeding capacity	Close	Yes
A6	25	7	3	172,700	Close to or exceeding capacity	Close	Yes
A7	30	8	3	192,800	Close to or exceeding capacity	Close	Yes
A8	40	10	3	212,500	Close to or exceeding capacity	Close	Yes

Table 3. Characteristics of the ANPSL-MT type set.

Instance	n	$\bar{k}$	$\bar{r}$	Total Demand (liters)	Demand Characteristics	Time Windows	Simultaneous Arrivals
B1	6	10	3	18,650	Less than truck capacity	Spaced	No
B2	15	10	3	69,730	Less than truck capacity	Close	Yes
B3	18	10	3	84,250	Close to truck capacity	Spaced	Yes
B4	25	10	3	109,250	Less than truck capacity	Spaced	Yes
B5	30	10	3	131,600	Close to truck capacity	Close	Yes
B6	45	10	3	182,800	Close to truck capacity	Close	Yes
B7	50	10	3	183,000	Close to or exceeding capacity	Spaced	Yes

5 Computational Results

The MILP model was coded in *Optimization Programming Language* (OPL) and solved by CPLEX 12.8 with default settings and some predefined maximum execution time, running on an Intel Core i5-8250U processor at 1.8 GHz and 12 GB of RAM.

5.1 Computational Performance of the MILP Model for a Time Limit of 18,000 s

In this subsection, we analyze the performance of the MILP model by running the CPLEX solver with a 5-h. time limit (18,000 s). The objective is to assess the ability to find high-quality solutions with the MILP model and to determine which instance characteristics contribute most to problem difficulty. Table 4 presents the results for the APSL-MT instance set, detailing the solutions obtained within the time limit. The first column identifies the name of the instance (I) and the second column presents the number of aircraft that requested service (n). The third and fourth columns show the *lower bound* (LB) and *upper bound* (UB) determined by CPLEX. The fifth column shows the gap (in %) $GAP = (UB - LB)/UB$. The sixth and seventh columns provide the CPLEX execution time in seconds ($Time$) and the number of trucks used in the solution (w), respectively.

Table 4. Experimental results of the application of the MILP proposed for MTARP in the APSL-MT type set.

		CPLEX 18,000				
I	n	LB	UB	$GAP(\%)$	$Time$	w
A1	6	740.00	740.00	0.00	<1.0	1
A2	11	2,000.00	2,000.00	0.00	<1.0	2
A3	15	1,720.00	1,720.00	0.00	<1.0	1
A4	18	2,880.00	2,880.00	0.00	10,562.45	2
A5	18	2,951.78	3,380.00	12.67	18,000.00	3
A6	25	2,978.09	4,960.00	39.96	18,000.00	5
A7	30	3,209.13	5,760.00	44.29	18,000.00	4
A8	40	3,663.51	8,540.00	57.10	18,000.00	6

From the results for the APSL-MT set in Table 4 it can be seen that smaller instances (6, 11, and 15 aircraft) were solved to optimality in under one second, while larger instances presented progressively greater challenges. Instance A4, containing 18 aircraft, required substantially more computational effort (10,562.45 s) despite reaching optimality. Notably, Instance A5 (also with 18 aircraft) proved more challenging, yielding a 12.67% gap after the 5-h time limit

(18,000 s). This increased difficulty likely stems from several key characteristics: fuel demands exceeding individual tanker truck capacities, a substantially higher total fuel requirement (106,850 liters), and numerous flights with overlapping time windows (see Table 2). The remaining instances (25, 30, and 40 aircraft) exhibited these same challenging characteristics, resulting in progressively larger optimality gaps of 39.96%, 44.29%, and 57.10% respectively. This pattern suggests that problem difficulty scales non-linearly with instance size when combined with these complicating factors.

The experimental results on Table 5 show that for small-scale instances (e.g., B1 with 6 aircraft), the model could be optimally solved within 1.25 s. However, solution quality degraded significantly for larger instances, with optimality gaps ranging from 7.18% to 37.96% when reaching the 18,000-sec time limit. Notably, Instance B5 (30 aircraft) exhibited the worst performance (35.9% gap) among mid-sized instances, suggesting a critical complexity threshold when combining close time windows with demand near truck capacity. Analysis of instance characteristics (Table 3) reveals three primary difficulty factors: (i) simultaneous aircraft arrivals, (ii) tight time windows, and (iii) demand-to-capacity ratios approaching or exceeding 100%. The counterintuitive performance of Instance B3 (18 aircraft), which achieved a relatively low 7.18% gap despite high demand, suggests that spaced time windows can partially mitigate challenges posed by increased fuel requirements. The results particularly emphasize how the convergence of multiple challenging characteristics - rather than just problem size alone - creates particularly difficult cases where exact optimization methods struggle to produce high-quality solutions within reasonable timeframes.

Table 5. Experimental results of the application of the MILP proposed for MTARP in the ANPSL-MT type set.

		CPLEX 18,000				
I	n	LB	UB	$GAP(\%)$	$Time$	w
B1	6	3,335.00	3,335.00	0.00	1.25	2
B2	15	7,763.30	9,926.00	21.79	18,000.00	4
B3	18	7,438.44	8,014.00	7.18	18,000.00	5
B4	25	10,208.79	11,566.00	11.73	18,000.00	6
B5	30	7,493.46	11,690.00	35.90	18,000.00	7
B6	45	14,413.26	19,369.00	25.59	18,000.00	8
B7	50	14,123.00	22,763.00	37.96	18,000.00	9

The initial tests of the MTARP model on both APSL-MT and ANPSL-MT instance sets (Tables 4 and 5) revealed substantial optimization times for larger instances. While these extended computation periods (up to 18,000 s) were valuable for establishing baseline performance, real-world operational scenarios

typically demand faster decision-making. To evaluate the model's practical applicability, we conducted additional experiments with a strict 30-min (1,800-s) time limit, as presented in Subsect. 5.2.

5.2 Experimental Analysis with Reduced Time Limits (30 Min)

Let UB(1,800) represent the upper bounds obtained with time limits of 1,800 s. In this subsection, we wish to assess how effective UB(1,800) is in approximating a good solution in practical settings, where only 30 min (1,800 s) are available to solve the problem. Table 6 compares results of the MTARP model on the APSL-MT instance set under time limits of 1,800 s and 18,000 s. The table shows that for instances (A1 to A5), UB(1,800) matches the best-known values or comes very close, making it a reliable guide. For simpler instances (A1 to A3), optimal solutions are found quickly—often within seconds. These instances feature relatively few customers, low total demand (well below truck capacity), and time windows that are well spaced. The number of vehicles used is minimal (usually one or two), and the solver can readily verify optimality. For moderately complex instances A4 and A5, the solution found within 30 min matches the one found in a 5-h run in terms of objective value and number of vehicles used. However, the reported optimality gaps remain above 20%, not because the solutions are of poor quality, but because the solver struggles to improve the lower bounds. These results highlight that as problem complexity grows (more aircraft, tighter constraints, simultaneous arrivals), the solver requires significantly more time to close the optimality gap.This trend becomes more pronounced for more complex instances (A6 to A8), where the number of aircraft, vehicles, and the total demand increase substantially. For Instances A6 and A7, UB(1,800) overestimates the cost by 4.84% and 7.29%, respectively—still under 10%, which is relatively acceptable for operational use. Moreover, the number of vehicles suggested is only one more than in the longer run, which may not be a critical issue in practice. Thus, UB(1,800) appears to remain useful in these cases. Notwithstanding, its limitations should be acknowledged in tightly constrained or highly complex instances. For example, in A8, CPLEX could not produce any feasible solution within 30 min, highlighting that UB(1,800) may not provide guidance at all in very large instances. In such cases, relying solely on a general-purpose MILP solver might not be sufficient, and alternative strategies, such as problem decomposition, matheuristics, or using tailored constructive heuristics, may be necessary to support decision-making. Furthermore, it would be useful to employ methods aimed at improving the lower bound. A tighter lower bound reduces the optimality gap and increases confidence in the quality of feasible solutions found. Techniques such as problem relaxations, dual formulations, or bounding procedures tailored to the problem structure could be explored to strengthen the model's performance under strict time constraints.

The experimental results shown in Table 7 reveal distinct computational phases based on instance characteristics. For small-scale instances (e.g., B1 with 6 aircraft), UB(1,800) consistently achieves optimal solutions (0% gap) in under two seconds while utilizing only two trucks (w = 2). This strong performance

aligns with the instance's simple structure: low fuel demand relative to truck capacity, widely spaced time windows, and no simultaneous aircraft arrivals. Such cases demonstrate UB(1,800)'s effectiveness for basic refueling scenarios where operational constraints are minimal. For medium-sized instances (B2-B4, 15-25 aircraft), UB(1,800) becomes increasingly sensitive to time constraints. Instance B2, despite having manageable demand levels (69,730 liters), exhibits a 27.4% gap under 30-min limits most likely due to its challenging combination of close time windows and simultaneous arrivals. Notably, while UB(1,800) remains identical for B2 (9,926.00), both B3 and B4 show modest increases of 3.2% and 1.3% respectively. This confirms that the degraded gaps primarily reflect insufficient time for lower bound improvement rather than substantially poorer solutions. The pattern suggests that UB(1,800) may be operationally acceptable for moderately complex cases despite suboptimal gaps, particularly when spaced time windows (B3, B4) help maintain solution stability. The most significant challenges appear in large-scale instances (B5-B7, 30-50 aircraft),

Table 6. Computational performance of the MILP model for MTARP using APSL-MT instances under different CPLEX time limits (1,800 s and 18,000 s)

I	n	CPLEX 18,000				CPLEX 1,800			
		UB	$GAP(\%)$	w	$Time$	UB	$GAP(\%)$	w	$Time$
A1	6	740.00	0.00	1	<1.0	740.00	0.00	1	<1.0
A2	11	2,000.00	0.00	2	<1.0	2,000.00	0.00	2	<1.0
A3	15	1,720.00	0.00	1	<1.0	1,720.00	0.00	1	<1.0
A4	18	2,880.00	0.00	2	10,562.45	2,880.00	20.77	2	1,800.00
A5	18	3,380.00	12.67	3	18,000.00	3,380.00	20.02	3	1,800.00
A6	25	4,960.00	39.96	5	18,000.00	5,200.00	48.05	6	1,800.00
A7	30	5,760.00	44.29	4	18,000.00	6,180.00	51.74	5	1,800.00
A8	40	8,540.00	57.10	6	18,000.00	-	-	-	1,800.00

Table 7. Computational performance of the MILP model for MTARP using ANPSL-MT instances under different CPLEX time limits (1,800 s and 18,000 s)

I	n	CPLEX 18,000				CPLEX 1,800			
		UB	$GAP(\%)$	w	$Time$	UB	$GAP(\%)$	w	$Time$
B1	6	3,335.00	0.00	2	1.25	3,335.00	0.00	2	1.28
B2	15	9,926.00	21.79	4	18,000.00	9,926.00	27.40	4	1,800.00
B3	18	8,014.00	7.18	5	18,000.00	8,268.00	22.99	5	1,800.00
B4	25	11,566.00	11.73	6	18,000.00	11,719.00	16.03	6	1,800.00
B5	30	11,690.00	35.90	7	18,000.00	14,651.00	50.17	7	1,800.00
B6	45	19,369.00	25.59	8	18,000.00	28,746.00	49.90	9	1,800.00
B7	50	22,763.00	37.96	9	18,000.00	26,961.00	48.54	10	1,800.00

where the interaction of multiple constraints creates compounding complexity. Instance B5's 50.17% gap - accompanied by a 25.3% cost increase in UB(1,800) - highlights how tight scheduling windows combined with near-capacity demand (131,600 liters) strain the solution quality. The effect intensifies for B6, showing a dramatic 48.4% increase in UB(1,800). Both bound quality and feasibility suffer, with B7's UB(1,800) degrading by 18.4% while requiring an additional truck (w goes from 9 to 10). These results underscore that under stringent operational constraints a hybrid solution approach may be necessary.

6 Conclusion

This paper considers a fuel supply problem faced by a Brazilian fuel distribution company, called the Multiple Trip Aircraft Refueling Problem (MTARP). A Mixed-Integer Linear Programming (MILP) model was presented to solve the problem. This model effectively represents the specific characteristics of the aircraft refueling service at Brazilian airports, meeting this crucial activity's operational and logistical needs. Sets of instances were created to provide various scenarios for problem analysis. Two key experiments were conducted: identifying the characteristics that make certain instances more challenging, and evaluating the quality of solutions obtained under a restricted 30-min computational time.

The first experiment analyzed the factors contributing to problem difficulty across different instance sizes and structures. The results showed that problem complexity does not scale linearly with the number of aircraft but is significantly influenced by the interaction of complicating factors. For example, while smaller instances (6, 11, and 15 aircraft) were solved optimally in under one second, instances with similar aircraft counts (e.g., A5 with 18 aircraft) exhibited substantial optimality gaps (12.67% after 18,000 s). This difficulty was attributed to high fuel demand, overlapping time windows, and demand exceeding individual truck capacities. Larger instances (25, 30, and 40 aircraft) exhibited even greater optimality gaps (up to 57.10%), confirming that complexity is compounded by multiple interacting constraints rather than just problem size alone. Furthermore, some mid-sized instances (e.g., B3) performed unexpectedly well due to spaced time windows, highlighting the importance of problem structure in determining computational tractability.

The second experiment evaluated how well solutions obtained in just 30 min (UB(1,800)) approximated those from extended runs. For small and moderately complex instances, UB(1,800) closely matched the best-known solutions, often achieving optimality within seconds (e.g., A1-A3, B1). Even for mid-sized instances like A4, A5, B3, and B4, UB(1,800) produced solutions with minimal deviations, making it a reliable approximation in many operational cases. However, as complexity increased, the solver struggled to refine lower bounds within the time limit, leading to larger optimality gaps, with feasibility also suffering in the most constrained scenarios (e.g., A8, where no feasible solution was found). These results indicate that while UB(1,800) remains useful in many practical settings, it is less reliable for highly constrained, large-scale instances.

It should be noted that large-scale and highly constrained instances may require alternative approaches such as decomposition techniques, matheuristics, or specialized heuristics to improve both solution feasibility and bound quality. Future work should explore these methods while also investigating ways to strengthen lower bounds, ensuring that optimization models remain practical for real-world decision-making.

Acknowledgments. The first author would like to acknowledge the doctoral scholarship provided by FAPES - Fundação de Amparo Pesquisa e Inovação do Espírito Santo (FAPES) (Grant Number 2020-LGJV3). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors also acknowledge support from PROAPEM/FAPES - project numbers: 368/2022 - P: 2022-NGKM5; and PDPG/FAPES - project numbers: 129/2021 - P: 2021-GL60J.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Azi, N., Gendreau, M., Potvin, J.Y.: An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. Eur. J. Oper. Res. **178**(3), 755–766 (2007). https://doi.org/10.1016/j.ejor.2006.02.019
2. Azi, N., Gendreau, M., Potvin, J.Y.: An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. Eur. J. Oper. Res. **202**(3), 756–763 (2010). https://doi.org/10.1016/j.ejor.2009.06.034
3. Babić, O.: Optimization of refuelling truck fleets at an airport. Transport. Res. Part B: Methodologica **21**(6), 479–487 (1987). https://doi.org/10.1016/0191-2615(87)90042-7
4. Battarra, M., Monaci, M., Vigo, D.: An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem. Comput. Oper. Res. **36**(11), 3041–3050 (2009). https://doi.org/10.1016/j.cor.2009.02.008
5. Braekers, K., Ramaekers, K., Van Nieuwenhuyse, I.: The vehicle routing problem: state of the art classification and review. Comput. Ind. Eng. **99**, 300–313 (2016). https://doi.org/10.1016/j.cie.2015.12.007
6. Cattaruzza, D., Absi, N., Feillet, D.: Vehicle routing problems with multiple trips. 4OR **14**, 223–259 (2016). https://doi.org/10.1007/s10288-016-0306-2
7. Cordeau, J.F., Groupe d'études et de recherche en analyse des décisions Montréal, Q.: The VRP with time windows. Citeseer (2000)
8. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Manage. Sci. **6**(1), 80–91 (1959). https://doi.org/10.1287/mnsc.6.1.80
9. Du, Y., Zhang, Q., Chen, Q.: ACO-IH: an improved ant colony optimization algorithm for airport ground service scheduling. In: IEEE International Conference on Industrial Technology, pp. 1–6 (2008). https://doi.org/10.1109/ICIT.2008.4608674
10. Eksioglu, B., Vural, A.V., Reisman, A.: The vehicle routing problem: a taxonomic review. Comput. Ind. Eng. **57**(4), 1472–1483 (2009)
11. Elatar, S., Abouelmehdi, K., Riffi, M.E.: The vehicle routing problem in the last decade: variants, taxonomy and metaheuristics. Procedia Comput. Sci. **220**, 398–404 (2023). https://doi.org/10.1016/j.procs.2023.03.051, the 14th International

Conference on Ambient Systems, Networks and Technologies Networks and The 6th International Conference on Emerging Data and Industry 4.0 (EDI40)
12. Feng, X., Zuo, H., Sun, Q.: Research on collaborative scheduling of aircraft ground service vehicles based on simple temporal network. In: 2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT), pp. 263–269 (2021). https://doi.org/10.1109/ICCASIT53235.2021.9633418
13. Fleischmann, B.: The vehicle routing problem with multiple use of vehicles (1990)
14. FlightAware, E.: Flightaware (2005). https://pt.flightaware.com/
15. Gamayanti, N., Sahal, M., Wibisono, A.: Optimization of vehicle routing problem with tight time windows, short travel time and re-used vehicles (VRPTSR) for aircraft refueling in airport using ant colony optimization algorithm. J. Adv. Res. Electr. Eng. **2**(1), 43–46 (2018). https://doi.org/10.12962/j25796216.v2.i1.40
16. Hernandez, F., Feillet, D., Giroudeau, R., Naud, O.: A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. 4OR **12**(3), 235–259 (2013). https://doi.org/10.1007/s10288-013-0238-z
17. Konstantakopoulos, G.D., Gayialis, S.P., Kechagias, E.P.: Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. Oper. Res. Int. J. **22**(3), 2033–2062 (2022). https://doi.org/10.1007/s12351-020-00600-7
18. Liu, Y., Wu, J., Tang, J., Wang, W., Wang, X.: Scheduling optimisation of multi-type special vehicles in an airport. Transportmetrica B: Transp. Dyn. **10**(1), 954–970 (2022). https://doi.org/10.1080/21680566.2021.1983484
19. Macedo, R., Alves, C., Valério de Carvalho, J., Clautiaux, F., Hanafi, S.: Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. Eur. J. Oper. Res. **214**(3), 536–545 (2011). https://doi.org/10.1016/j.ejor.2011.04.037
20. Quan, W., Chen, C., Shao, Z., Meng, K.: Dynamic scheduling for airport special vehicles based on a multi-strategy hybrid algorithm. In: 2022 41st Chinese Control Conference (CCC), pp. 1916–1921 (2022).https://doi.org/10.23919/CCC55666.2022.9901875
21. Schyns, M.: An ant colony system for responsive dynamic vehicle routing. Eur. J. Oper. Res. **245**(3), 704–718 (2015). https://doi.org/10.1016/j.ejor.2015.04.009
22. Seixas, M.P., Mendes, A.B.: Column generation for a multitrip vehicle routing problem with time windows, driver work hours, and heterogeneous fleet. Math. Probl. Eng. **2013** (2013).https://doi.org/10.1155/2013/824961
23. Toth, P., Vigo, D.: The vehicle routing problem. Soc. Ind. Appl. Math. (2002). https://doi.org/10.1137/1.9780898718515
24. Wang, Y., Sun, J., Sun, B., Feng, M.: Research on application of airport tanker truck scheduling based on particle swarm optimization. In: 2021 6th International Conference on Control, Robotics and Cybernetics (CRC), pp. 278–282 (2021). https://doi.org/10.1109/CRC52766.2021.9620157
25. Wang, Z., Li, Y., Hei, X., Meng, H.: Research on airport refueling vehicle scheduling problem based on greedy algorithm. In: Huang, D.-S., Bevilacqua, V., Premaratne, P., Gupta, P. (eds.) ICIC 2018. LNCS, vol. 10954, pp. 717–728. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95930-6_73
26. Zampirolli, K.A., Amaral, A.R.S.: Simulated annealing and iterated local search approaches to the aircraft refueling problem. In: Gervasi, O., et al. (eds.) ICCSA 2021. LNCS, vol. 12952, pp. 422–438. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86973-1_30

27. Zhang, H., Ge, H., Yang, J., Tong, Y.: Review of vehicle routing problems: models, classification and solving algorithms. Arch. Comput. Methods Eng. **29**(1), 195–221 (2021). https://doi.org/10.1007/s11831-021-09574-x
28. Zhaohua, Z., Liqi, J.: Research on airport special vehicle scheduling problem. In: 2020 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), pp. 1076–1078 (2020). https://doi.org/10.1109/ICVRIS51417.2020.00262