Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/200314

Scalable Optimization of Line Maintenance Scheduling: Comparative Analysis of MILP and Constraint Programming Formulations

Bilal Bataineh[1], *Mohamed Heshmat*[2,3], *and Abdelmoty M. Ahmed*[2,*]

[1] Department of Computer Science, Jadara University, Irbid, Jordan
[2] Department of Computer Sciences, Faculty of Information and Technology, Ajloun National University, P.O. Box 43, Ajloun 26810, Jordan
[3] Faculty of Computers and Artificial Intelligence, Sohag University, Sohag 82524, Egypt

Received: 7 Jan. 2026, Revised: 28 Mar. 2026, Accepted: 10 Apr. 2026
Published online: 1 May 2026

Abstract: Aircraft line maintenance scheduling is a critical operational challenge for airlines, requiring the coordination of repetitive mandatory tasks, non-mandatory inspections, and time-varying resource constraints within narrow maintenance windows at multiple airports. This paper proposes a compact time-indexed mixed-integer linear programming (MILP) model that minimizes weighted earliness and tardiness penalties while respecting complex precedence networks, non-simultaneity constraints, airport restrictions, and dynamic resource capacities.To address scalability limitations of time-indexed formulations for large fleets, we develop a constraint programming (CP) alternative using interval variables and cumulative resource functions, eliminating the variable explosion inherent in fine-grained temporal discretization. We conduct a rigorous computational comparison between the MILP (solved with Gurobi) and CP-SAT (Google OR-Tools) formulations across 14 real-world instances spanning 5–11 aircraft, 10–30 planning days, and 51–331 maintenance tasks. Results demonstrate that while the MILP guarantees optimality for small-to-medium instances (< 150 tasks), the CP-SAT approach achieves solutions within 0.93% of optimality while reducing solution time by 43.6% on average and cutting memory usage by 42% for large-scale problems (> 200 tasks).We further extend both formulations to handle operational uncertainties—flight delays, unplanned corrective tasks, and resource absenteeism—through dynamic buffer management strategies. The proposed framework enables airlines to select the appropriate solver based on fleet size and time sensitivity: MILP for optimality-critical planning horizons and CP-SAT for rapid daily re-optimization under dynamic conditions. Our open-source implementation provides practitioners with a scalable decision support tool for robust line maintenance scheduling.

Keywords: Aircraft maintenance scheduling; Mixed-integer linear programming; Constraint programming; Time-indexed formulation; Operational robustness; Resource-constrained project scheduling.

2010 Mathematics Subject Classification. Primary 26A33, 35A20; Secondary 33B15.

1 Introduction

Aviation stands as one of the most globally integrated industries, facilitating over 100,000 daily flights that transport approximately 12 million passengers and $18 billion worth of cargo worldwide [1]. The exceptional safety record of modern aviation stems largely from rigorous maintenance practices mandated by international regulatory bodies. Aircraft maintenance—encompassing both preventive and corrective activities—is not merely a regulatory obligation but a strategic lever influencing airline profitability, operational reliability, and passenger trust. Globally, airlines allocate $50.3 billion annually to maintenance, repair, and overhaul (MRO) activities, representing nearly 10.5% of total operational expenditures [2]. Consequently, even marginal improvements in maintenance scheduling efficiency translate directly into enhanced fleet utilization, reduced operational disruptions, and significant cost savings.

Aircraft maintenance activities are formally categorized into two distinct types based on scope, resource requirements, and execution environment: base

* Corresponding author e-mail: A.Ahmed@anu.edu.jo

maintenance and line maintenance [2,3,4]. Base maintenance encompasses heavy inspections and major overhauls requiring hangar facilities, specialized tooling, and extended aircraft downtime—typically ranging from several days to 30 days. These activities include engine overhauls, structural inspections, corrosion treatments, and major system modifications that cannot be performed in operational ramp environments [3,4,5].

In contrast, line maintenance comprises tasks executable outside hangar facilities during routine ground operations, specifically within the constrained windows of turnaround times (TAT) between consecutive flights or overnight layovers at airports subject to night-flying restrictions [6,7]. As defined under CASR Part 145 regulations, line maintenance tasks do not require in-depth structural inspections and are typically performed at remote stands or ramp positions using portable equipment [6,8]. Representative activities include replacement of Line Replaceable Units (LRUs), minor system repairs, troubleshooting of reported defects, mandatory walk-around inspections, and cabin servicing. Although individually less resource-intensive than base maintenance activities, line maintenance tasks occur with substantially higher frequency—often daily per aircraft—and therefore exert a direct and continuous influence on operational costs, flight punctuality, and fleet availability.

The execution schedule for all maintenance tasks is governed by the aircraft manufacturer's Maintenance Planning Document (MPD), which specifies for each task a unique identifier, required processing time, resource profile (including mechanics, avionics specialists, electricians, cabin technicians, and Ground Support Equipment such as power units and specialized tools), and a maximum execution interval expressed through one or more usage parameters: flight hours (FH), flight cycles (FC), calendar days (DY), or calendar months (MO) [4, 9]. Operators may convert these parameters to alternative units provided the conversion does not violate the minimum safety intervals established in the Maintenance Review Board Report (MRB) and MPD. Critically, failure to execute any mandatory task before its specified deadline results in immediate loss of airworthiness certification, grounding the aircraft until compliance is restored [4,10]. This regulatory imperative—combined with the operational reality that line maintenance opportunities are confined to brief, stochastic ground windows—creates the complex scheduling challenge addressed in this research.

Historically, airlines employed block-check packaging methodologies, grouping maintenance tasks with similar scopes and periodicities into letter checks (A, B, C, D) [7, 10]. This approach, however, systematically compromises component lifespan utilization: tasks with intervals shorter than the scheduled check (e.g., 15-month tasks within a 20-month C-check) are executed prematurely, accelerating maintenance cycles and creating artificial workload peaks. The paradigm shifted fundamentally with the second revision of the Maintenance Steering Group-3 (MSG-3) framework developed by Airlines for America (A4A), which decoupled maintenance tasks from mandatory letter-check packaging [11]. This enabled the emergence of task-oriented maintenance philosophies where individual tasks are scheduled at their optimal execution points—maximizing component utilization while respecting maximum allowable intervals. Under this philosophy, every ground time window becomes a potential maintenance opportunity, transforming previously idle aircraft downtime into productive maintenance capacity [12].

Despite its operational significance, line maintenance scheduling remains predominantly manual in practice, yielding suboptimal solutions that fail to exploit the full potential of task-oriented maintenance. The Line Maintenance Scheduling Problem (LMSP) requires coordinating repetitive mandatory tasks, opportunistic non-mandatory inspections, and time-varying resource constraints within narrow maintenance windows across multiple airports—all while respecting complex precedence networks, non-simultaneity constraints, airport-specific restrictions, and dynamic resource capacities. Prior optimization research has largely oversimplified this problem by neglecting critical real-world constraints. For instance, Katscher et al. [13] modeled LMSP with resource constraints only, ignoring precedence relationships, airport restrictions, and task repetitiveness.

Shaukat [14] incorporated airport restrictions and task inclusions/exclusions but omitted precedence constraints and time-dependent resource profiles. Van Buskirk et al. [28] addressed resource time-dependency for fighter aircraft maintenance but provided no formal objective function or mathematical model. Critically, none of these studies addressed operational uncertainties inherent to airline environments—flight delays altering maintenance windows, unplanned corrective tasks emerging dynamically, or resource absenteeism—nor did they evaluate alternative solver technologies beyond monolithic MILP approaches. This paper addresses these gaps through four principal contributions:

A compact time-indexed MILP formulation that minimizes the total weighted deviation (earliness plus tardiness) from task due dates while fully capturing realistic LMSP constraints—including task precedence networks with repetitions, non-simultaneity constraints, airport restrictions, release/deadline windows, and time-varying resource capacities—with minimal reliance on weak Big-M constraints that degrade solver performance [15].

The first constraint programming formulation for LMSP using Google OR-Tools CP-SAT solver, representing tasks as interval variables with native support for sequencing and cumulative resource constraints. This approach eliminates the variable explosion inherent in fine-grained temporal discretization, reducing model size

by 40–60% for large instances.
A rigorous computational comparison between MILP (Gurobi) and CP-SAT formulations across 14 real-world instances spanning 5–11 aircraft, 10–30 planning days, and 51–331 maintenance tasks. Results demonstrate that while MILP guarantees optimality for small-to-medium instances ($<$ 150 tasks), CP-SAT achieves solutions within 0.93% of optimality while reducing solution time by 43.6% on average and cutting memory usage by 42% for large-scale problems ($>$ 200 tasks).
A stochastic extension with dynamic buffer management that proactively allocates front-log buffers and resource slack to accommodate operational uncertainties—flight delays, unplanned corrective tasks, and workforce variability—without requiring complete schedule regeneration. This extension increases schedule robustness by 22% under 15% stochastic disruption while maintaining solution quality within 3% of the deterministic optimum.
Due to the significant impact of the scheduling approach on the operational cost of airlines, several works were published, generally focusing on finding optimal solutions for different types of maintenance planning and routing problems. However, previous studies have oversimplified the problem by ignoring important considerations and making unrealistic assumptions. Under realistic constraints, given a flight schedule and the set of line maintenance activities to be performed, this work focuses on deciding when and where tasks should be carried out to minimize the total deviation (the total earliness plus tardiness of the scheduled jobs) relative to tasks due time. This problem is known as the Line maintenance scheduling Problem (LMSP) [9]. This results in a higher task yield, which measures the scheduled maintenance task interval compared to the maximum allowable interval expressed as a percentage. Consequently, optimal or near-optimal scheduling of the LMSP can help improve maintenance efficiency and help balance the workload throughout peak periods.
Moreover, the problem aims to reduce the costs of leaving non-mandatory tasks unscheduled, which may require hiring additional staff or outsourcing to a third-party company [9]. This problem presents unique challenges due to several requirements for a wide range of ground time windows and tasks. The conditions that will be applied to the LMSP and discussed in this research are as follows:

1. Tasks must be done within their time window (after their release time and before their deadline) and as close to their due time as possible.
2. Depending on their interval, tasks could be executed more than once during the planning horizon.
3. Tasks cannot be interrupted once started (i.e., pre-emption is not allowed); therefore, a task's starting time and finish time should be within the upper bound and lower bounds of the TAT or the overnight stay period.
4. The LMSP considers multiple aircraft, airports, and different types of time-dependent resource capacity.
5. The LMSP includes both mandatory and non-mandatory tasks.
6. Tasks vary in importance, resource requirements, and processing times.
7. Non-simultaneity and precedence constraints apply between tasks.
8. Certain tasks are banned at specific airports.
9. Task deadlines are expressed in different parameters like flight hours, flight cycles, days, and months.

A time-indexed MILP mathematical model was developed to fully capture the LMSP constraints while minimizing the model size (number of variables and constraints) and contributing to a faster solution time by decreasing the dependency on the big-M constraints type. To test the model, various instances of different sizes were generated and solved using a commercial MILP solver.

2 Literature review

Aircraft maintenance planning, and scheduling, have been extensively studied in many works of literature, as shown by the literature reviews of Jorne Van den Bergh et al. [16,17]and Abdelrahman E.E. Eltoukhy et al. [18], mainly in the aircraft routing problem or any integrated study that includes the aircraft routing problem. Aircraft routing and maintenance planning could be studied in integration with one or more of the first three main stages of airline operations (flight schedules, fleet assignment, and crew scheduling). It is mainly incorporated in these phases in the form of constraints such as [18,19,20,21, 22,23]; however, several studies tackle aircraft maintenance scheduling as their primary objective [23, 24,25] Aircraft maintenance planning typically includes two stages of a top-down approach: aircraft maintenance check scheduling (AMCS) and maintenance task allocation (allocation of maintenance tasks to letter checks resulting from stage one) [25,26].
The aircraft maintenance check schedule determines maintenance check times while considering various operational constraints such as aircraft utilization, route, and maintenance capacity. Pedro Andrade et al. [27] recently studied this problem using Reinforcement Learning. They created a Deep Q-reinforcement learning algorithm to optimize the long-term scheduling of A and C-checks to maximize flight hours. A limited amount of hangar space each day and a higher priority for scheduling C-checks were considered in this study [27]. The same problem was studied by Nima Safaei et al[28] to minimize the sum of non-revenue flights and the misalignment between the maintenance opportunities and due maintenance workload [28]. using an integer programming model for the aircraft maintenance routing problem.

The second stage of maintenance planning involves solving the task allocation problem. Tasks with different intervals from letter checks must be allocated to more frequent checks, increasing maintenance. The task allocation problem aims to optimally allocate tasks in predefined maintenance checks to execute them as close to their due dates as possible while accounting for sharing available resources at maintenance stations. Huaiyuan Li et al. [29]developed a model for this using fuzzy C-means clustering with reduced dimensions and variables. They solved it using a nonlinear simplex optimization method to optimally combine maintenance tasks into work packages. The model assigns different weights to similarities between tasks, such as the ATA code, tasks interval, zone, and possible check types, then gives each type of similarity a different weight where the task interval had the highest importance.

To dynamically respond to changes in the number and duration of maintenance activities and to cope with uncertainties in aircraft daily utilization and maintenance check processing time, Qichen Deng et al. [31] proposed a lookahead approximate dynamic programming (ADP) algorithm. Their objective was to minimize total unused flight hours between checks while considering operational constraints like hangar capacity and maximum duration between letter checks. Carlos Lagos et al [32] tackled only one type of uncertainty (the stochastic disclosure of tasks over time) in the task-based maintenance scheduling problem by designing a dynamic decision support tool based on different Approximate Dynamic Programming policies. In addition, they integrated the tail assignment problem (i.e., dynamically assigning aircraft to itineraries) with their maintenance scheduling model to increase the problem's flexibility and further reduce maintenance costs. Limitations of these studies include assuming that additional resources can always reduce the processing time of a task and considering only overnight stays as maintenance opportunities, ignoring turnaround times. The analysis also assumes that resources are constant over time and that tasks can be scheduled in any order.

Task allocation problem has been studied on fighter aircraft as well by Van Buskirk, C et al. [30], who produced a two stages system that aids ground-attack aircraft maintenance planners in allocating mission flights to aircraft and scheduling maintenance activities down to a fifteen minutes resolution. For the maintenance scheduling part, the authors represented the problem as a mixed scheduling and resource allocation constraint satisfaction-based problem solved with a search engine built using the Mozart system's Oz programming language. The system's task is to decide the starting time of all maintenance jobs within specified ground time windows, considering the time-dependent availability of human resources and tools and other operational constraints. Precedence constraints, task release time and deadline constraints, and task conflict constraints are some operational constraints considered in their model. Although Van Buskirk, C et al. [30] study is closely related to our work, it does not provide an objective function to be optimized or a mathematical model. In their finite domain constraint problem, the objective was to find a solution that satisfies all constraints imposed by the task properties and resource capacity.

To dynamically respond to changes in the number and duration of maintenance activities and to cope with uncertainties in aircraft daily utilization and maintenance check processing time, Qichen Deng et al. [31] proposed a lookahead approximate dynamic programming (ADP) algorithm. Their objective was to minimize total unused flight hours between checks while considering operational constraints like hangar capacity and maximum duration between letter checks. Carlos Lagos et al [32] tackled only one type of uncertainty (the stochastic disclosure of tasks over time) in the task-based maintenance scheduling problem by designing a dynamic decision support tool based on different Approximate Dynamic Programming policies. In addition, they integrated the tail assignment problem (i.e., dynamically assigning aircraft to itineraries) with their maintenance scheduling model to increase the problem's flexibility and further reduce maintenance costs. Limitations of these studies include assuming that additional resources can always reduce the processing time of a task and considering only overnight stays as maintenance opportunities, ignoring turnaround times. The analysis also assumes that resources are constant over time and that tasks can be scheduled in any order.

The drawbacks of the letter check philosophy in aircraft maintenance regarding task yield have led to an interest in task-based scheduling for airline operation optimization models. For example, Ozkol and Senturk [33,34]studied an Airbus A340 fleet case where the total maintenance downtime over ten years where 15 days using the individual-task-driven philosophy compared to 87 days using the traditional letter check packaging concept. Their results were achieved by utilizing every moment the aircraft was on the ground in executing maintenance jobs. In addition, they constructed software that produces more efficient letter check alternatives in a manner that decreases the earlier or later accomplishment of some tasks]cite12. As for ensuring that the most suitable staff members perform many tasks, the authors used a fuzzy analytical hierarchy process method that deals with decision-making uncertainties [12].Nevertheless, their research does not provide a mathematical formulation of the scheduling problem and does not consider most of the problem parameters and constraints.

The study of Ozkol and Senturk [27] was further extended in detail by Steenkamp in[11], where he compared the results of an A-check-based scheduling approach and an individual task-based scheduling approach over two months for a fleet of 15 aircraft. The author in [11] showed that using a rolling horizon metaheuristic algorithm with one day of accurate forecasting could remove the need for aircraft to be taken

out of operations, which decreased the lost flying hour by nearly 98%. This is done by individually planning maintenance tasks within overnight ground times while solving the tail assignment problem [11].

Even though developments in line maintenance have a great potential to improve an airline maintenance operation cost because of its high recurrence rate, there is little research on optimized solutions and modeling of the LMSP. One of the recent and most relevant literature on this topic is the work of M. Katscher et al [13]. This work modeled the LMSP as a mixed integer programming problem to determine the best starting time for each maintenance task to minimize the total deviation of tasks from their due dates. The primary constraint applied to this model was the resource constraint. Two solution methods were compared: a sequential heuristic algorithm and an exact integrated solution algorithm. The heuristic approach was 50 times faster but produced sub-optimal solutions with a 3.5% optimality gap, while the exact algorithm provided optimal solutions for all instances [13].

Moreover, they conducted a sensitivity analysis, concluding that having more maintenance bases with a relatively small resource capacity is more effective than having fewer maintenance bases with a higher resource capacity. The study assumed tasks could be scheduled in any order and that resource capacity was constant along the planning horizon. However, in reality, there are constraints like precedence and non-simultaneity constraints, and some tasks cannot be executed at all airports due to facility, noise, policy, regulation restrictions, etc. It was also assumed that a task would not be performed more than once within the planning horizon, which does not align with the repetitive nature of the tasks.

Some of these assumptions (maintenance airport restrictions, simultaneous job inclusions, and exclusions) in M. Katscher et al [13]work were addressed in earlier literature by S.Shaukat [14]that solved the LMSP using a hybrid combination of greedy search Earliest Deadline First (EDF), Shortest Processing Time (SPT) and Largest Processing Time (LPT) heuristic techniques. The aim was to maximize task yield by introducing compactness and a relaxing factor that controls the task's release time. His algorithm was tested on a range of 64 cases and produced schedules with 43% higher task yields under 45 seconds. S. Shaukat [14] indicated that it is better to add an overall 'safety buffer' to the ground time of the aircraft rather than adding it to each task's completion times. This literature would have been more realistic to the line maintenance environment if the precedence relationship of jobs and man-hours time dependency were considered. Moreover, it lacks an exact solution to their case instances which would help determine how close their results are to the optimal ones.

To deal with emerging unplanned tasks over time in the context of line maintenance scheduling, Mikael Öhman et al. [35] proposed two types of safety buffers along resource capacity and time dimensions (similar to the due time buffers adopted by[13] that helps in dynamic and responsive scheduling. The first is a front-log buffer that schedules routine maintenance tasks earlier than their deadlines, dynamically creating maintenance opportunities for later use. This allows planned tasks to be postponed by giving more future maintenance opportunities to free up capacity and accommodate unplanned tasks when they emerge just in time. The second is maintaining a slack resource capacity buffer (e.g., ample labor and space supply) that helps accommodate unplanned demand. The paper focuses more on front-log scheduling using an empirically grounded discrete event simulation to find the best balance between maximizing the task's maintenance intervals and allowing planned tasks to be postponed without violating its deadline to accommodate emergent demand.

There are several scheduling famous problems in the field of operation research similar to a great extent to the LMSP. These include the Job Shop Scheduling Problem (JSSP), the bin packing problem (BPP), and the Resource Constrained Project Scheduling Problem (RCPSP).

The Job Shop Scheduling Problem involves assigning a sequence of operations to machines to minimize the completion time of all jobs. The Flexible Job Shop Problem is a variant where operations can be processed on any available machine. Extensions and solutions to this problem have been extensively studied, as shown in [36] Maintenance schedules can be viewed as a similar problem, with maintenance opportunities as machines and tasks as operations. This context was implemented in an Aircraft Engine Maintenance scheduling study in [37] to assign subcomponents of aircraft engines (fan, turbine, core, nozzle, and augmenter) to specialized work areas using a Multi-objective Evolution Algorithm to minimize two objectives: the time for engines to return to missions and the total number of times an engine needs maintenance [37].

Max Witteman et al. [38] and in studies [39,40,41] showed how the maintenance task allocation to letter checks problem is analogous to the time-constrained variable-sized bin packing problem (TC-VS-BPP), an extension of the basic Bin Packaging Problem (BPP). In the TC-VS-BPP, items (analogous to maintenance tasks) have associated ready time and allocation deadlines, whereby bins (comparable to pre-scheduled maintenance checks) have variable multidimensional resource capacities. When non-routine tasks (i.e., items) are disclosed dynamically over time, this resembles the online extension of the BPP problem where the complete set of items is not fully known before packaging. The aim of [38]is to minimize the de-escalation costs due to the scheduling of tasks much earlier than their due time. Max Witteman et al. [25] modeled the multi-year task allocation problem for a long-term aircraft A- and C-check schedules as a 0–1 MILP formulation and then designed a constructive worst-fit decreasing (WFD)

heuristic algorithm. in studies [39,40,41] greedily allocated tasks to bins (in [38], bins are considered to be the turnaround times and overnight stays) that will lead to the least de-escalation cost. One impressive feature of the Max Witteman et al [38] model is that they consider that since tasks are of the routine task type, one task can be scheduled and conducted more than once for the same aircraft over the time planning horizon. To do this, they estimate the maximum possible number of repetitions for each task and consider each occurrence of the same tasks to be an item in our TC-VS-BPP [38].

As indicated in [9], the closest related problem to the LMSP is the Resource Constrained Project Scheduling Problem (RCPSP) in the field of operation research. The RCPSP assigns activities to machines (i.e., maintenance opportunities) and produces a non-preemptive schedule with a minimum makespan for a set of tasks, each with its predefined processing time, subject to precedence and resource constraints. RCPSP comes in many variants, as shown in [42,43] but those most similar to the LMSP in this research assume time-varying resource capacities with just-in-time objectives. In studies [44,45] proposed an exact algorithm for RCPSP with weighted earliness/tardiness costs under the assumption of zero-lag finish–start precedence constraints. Jafar Bagherinejad [46] used a two-phase genetic algorithm to solve a similar problem with additional restrictions. They considered more than one executive mode is possible for each task, along with minimum and maximum time lags between activities.

Tackling time-dependent resource capacities in the RCPSP, in studies. [47,48] used a branch-and-bound approach and showed that time-dependent resource capacities could be converted to constant capacities. They propose that the constant capacity will be the maximum time-varying capacity over time. Then for each time slot with a smaller capacity, a dummy activity will be scheduled covering the whole slot to decrease the capacity appropriately. This conversion method was used in aircraft maintenance applications in [30] and process industries in [49]. They also showed that the same methodology can be applied to time-varying resource demand. Resource capacity having a value for each point in time was the time–indexed MIP formulations of this constraint in the work of Sonke Hartmann [50]. The proposed solution was a serial SGS algorithm solving a medical and pharmacological research project with time-varying resources and demand.

3 Problem Definition

This section defines the LMSP variant of this work, and the main terminologies and mathematical notations used. Also, it provides a formal definition using a time-indexed MILP formulation of the problem.

3.1 Problem Statement

This problem involves mandatory $i \in O$ and non-mandatory tasks $i \in N$. Mandatory tasks are the ones that an airline should perform by their own maintenance team within the planning horizon L satisfying the full range of constraints. Non-mandatory tasks, on the other hand, are scheduled and executed opportunistically and can be left unscheduled at a penalty cost c_i added to the objective function. This work assumes that non-mandatory tasks do not obey precedence, non-simultaneity, release time, and deadline constraints. Furthermore, unlike mandatory tasks, every non-mandatory task cannot be executed more than once within the planning horizon. The problem aims to schedule as many non-mandatory tasks as possible, avoiding the penalty cost c_i by trying to set the start time S_i of non-mandatory tasks to be somewhere feasible within the panning horizon L. A feasible starting time S_i for a non-mandatory task is the one that guarantees available resources and time window for the whole duration of the task execution time T_i. For unexecuted non-mandatory tasks $i \in N$, a fictitious starting time $t = 1 + \Delta t\,((L\ div\ \Delta t) - 1)$ at a fictitious airport $p = 0$ is defined to have unlimited resources of every type and unlimited time duration to guarantee that any non-mandatory task can be assigned to it without any constraint's violations.

This model also distinguishes between the 1st execution of a mandatory jobs $i \in O$ and the following executions of the same task within the planning duration L. Every mandatory task $i \in O$ is characterized by a deadline D_i, a ready time Q_i and a due time C_i. Mandatory tasks should start and finish sometime feasible between Q_i and D_i while having a finish time (starting time S_i + task execution duration T_i) as close as possible to C_i to minimize deviation penalty. Mandatory task i deviation penalty is either the product of the task's earliness Y_i and its cost per unit of earliness a_i or the product of the task's tardiness Z_i and its cost per unit of tardiness b_i. Q_i, C_i and D_i could be represented by flight hours (FH), flight cycles (FC), calendar days (DY), or months (MO); however, using historical data and predefined flight schedules, operators can convert them to common desired units (HH:MM:SS time format in this work). For both mandatory and non-mandatory tasks, there is a defined resource consumption r_i^k for every resource type $k \in K$. A feasible solution should ensure enough airport resource capacity at all times $R_k^{t,p}$ for all resource types. Furthermore, the starting point S_i of task i and the finish time should always be within a valid task i maintenance opportunity $m \in M_i$ starting time l_m and finish time u_m.

For non-simultaneity and airport restriction considerations, we define for each mandatory task $i \in O$ set U_i of tasks that cannot be executed while task i is being processed and set P_i which represents the airports where task i is not allowed to be conducted. To accommodate the dependency relations between different

tasks and the several occurrences of the same task (i.e., the activity precedence network such as the one shown in Figure 1) due to the technical origin or the logic of maintenance execution, set B_i such as the ones in Table 1 are defined for each mandatory task based on the maximum possible occurrences Max_i of this task within the planning horizon. For task i to be executed, all tasks in set B_i should have been already finished.

The objective of the problem is to set the finish times of maintenance tasks as close to the tasks' due times, minimizing the total deviation penalty and making the most of the lifetime of the aircraft components. The solution to the LMSP in this work is defined by the starting time S_i of each mandatory and non-mandatory task and whether the task is scheduled within the planning horizon. This model utilizes a time-indexed formulation of the problem by introducing a time parameter $t \in T_0$ where $T_0 = \{1, 1+\Delta t, 1+2\Delta t, \ldots, 1+\Delta t((L+\max_i(T_i)) \text{ div } \Delta t)-1)\} \ \forall \ i \in N$ representing all time points within the planning horizon and extra fictitious times outside the planning horizon L to deal with non-mandatory tasks. This discretizes the planning duration into equidistant points in time depending on the temporal resolution or the time step Δt of the discretization. Each point in time in T_0 is considered a potential event for a task to either start at this time or be in an execution process. In this formulation, dummy tasks are not needed since time-dependent resource capacity can be represented by a resource capacity value $R_k^{(t,p)}$ for each point in time. The model's output is to decide which of these discretized points in time is the best starting time for each task, resulting in an overall minimization of the total deviation of mandatory tasks plus the penalty cost of leaving unscheduled non-mandatory tasks.
This time-indexed model deals with the time-dependent resource constraints without the use of permutation-based variables representing whether tasks are overlapping or not, thus requiring fewer variables, constraints, and nonzero elements in the constraint matrix, which significantly reduces the size of the problem, and it facilitates the implementation of a pre-processing stage and non-simultaneity considerations. The formulation also considered the minimal usage of the big-M constraint type. Big-M-based constraints do not provide a close lower bound estimation on the objective function; therefore, many iterations will be required to prune nodes based on objective function value, slowing the solution process [15].

3.1.1 **Mathematical notation**

The sets, parameters, and functions used in this formulation are summarized as follows: Indices:

–i, u: index for task $i \in J$
–k: index for resource type $k \in K$
–p: index for an airport or a maintenance base $p \in P$
–m: index for a maintenance opportunity $m \in M_i$
–t: time instant $t \in T_0$
–e, g: represent execution number of task i where $e, g \in \mathbb{Z}^+ \mid e, g \leq \text{Max}_i$ or $e, g \in E_i$

Sets:

–P: set containing all airports, including the fictitious airport $p = 0$
–O: set of all mandatory maintenance tasks to be scheduled for the whole fleet
–N: set of the fleet's non-mandatory maintenance tasks to be scheduled or left unscheduled at a cost penalty
–J: set of all mandatory and non-mandatory tasks (i.e., $J = O \cup N$)
–K: set containing all types of resources (technicians, avionics Engineers, and Airframe and Engine Engineers) used in the resource demand and supply.
–M: set of all maintenance opportunities for all tasks, including the one at the fictitious airport $p = 0$.
–$M_{i,e}$: set of all maintenance opportunities of the e^{th} execution of task i where $i \in J$, and $e \in E_i$.
–$M_{i,e}^p$: set of all airport p maintenance opportunities of the e^{th} execution of task i where $i \in J$, and $e \in E_i$.
–$B_{i,e}$: set of mandatory tasks that should precede execution number e of task i (i.e., set of tasks whose finish time should be before the start time of execution number e of task i) where $i \in O$ and $e \in E_i$.
–$U_{i,e}$: Set of mandatory tasks that cannot be executed while execution number e of task i is being processed where $i \in O$ and $e \in E_i$.
–P_i: Set of airports that task i is allowed to be conducted at.
–E_i: set containing all possible numbers of executions of task i within the planning horizon starting with 1 (i.e $E_i = \{1, 2, \ldots, \text{Max}_i\}$ where $i \in O$ and Max_i is the maximum number of executions possible for task i within the planning horizon.
–E_i': a set containing all possible numbers of executions of task i except 1 (i.e., starting from 2 $E_i' = \{2, 3, \ldots, \text{Max}_i\}$)
–T_0: set of all time points within the planning horizon plus additional fictitious processing times for non-mandatory jobs. $T_0 = \{1, 1+\Delta t, 1+2\Delta t, \ldots, 1+\Delta t((L+\max_i(T_i)) \text{ div } \Delta t)-1)\} \ \forall \ i \in N$
–$Ps_{i,e}$: set of all possible starting times of the e^{th} execution of task i including $t = 1+\Delta t((L \text{ div } \Delta t)-1)$ if $i \in N$.
–$Pe_{i,e}$: set of all possible processing times of the several occurrences of task $i \in J$, including fictitious times if $i \in N$.
–$J_{t,p}$: set of all tasks $i \in J$ that their associated aircraft will be at airport p at time t where time t is a possible execution time (i.e., $t \in Pe_i$)

Functions: $p_i(t)$: outputs the airport that the aircraft corresponding to task i will be at during time t

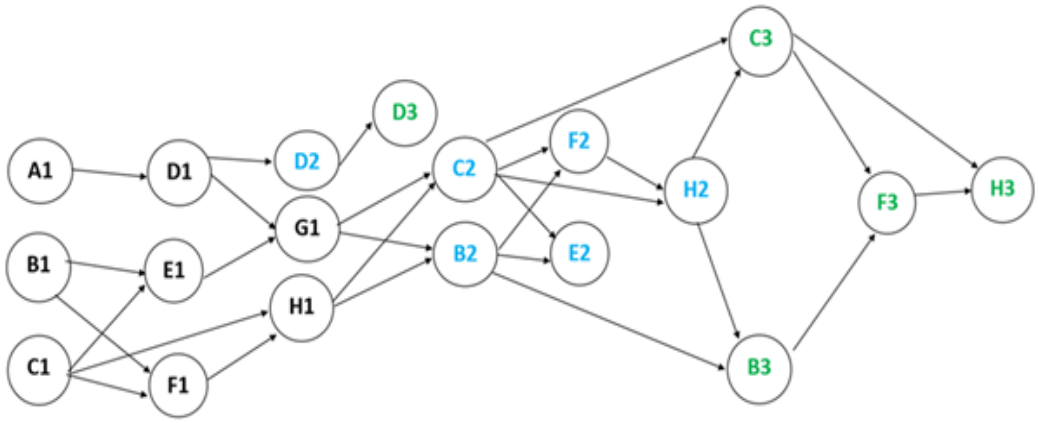

Fig. 1: Example of a sample of task precedence network including several executions of the same task.

Table 1: Precedence sets $\mathbf{B_{i,e}}$ of Figure 1 task precedence network.

Task i	$\mathbf{Max_i}$	1st Execution	2nd Execution	3rd Execution
A	1	$B_{A,1} = \{\ \}$	A2 does not exist	A3 does not exist
B	3	$B_{B,1} = \{\ \}$	$B_{B,2} = \{B1, G1, H1\ \}$	$B_{B,3} = \{B2, H2\}$
C	3	$B_{C,1} = \{\ \}$	$B_{C,2} = \{C1, G1, H1\ \}$	$B_{C,3} = \{C2, H2\ \}$
D	3	$B_{D,1} = \{A1\ \}$	$B_{D,2} = \{D1\ \}$	$B_{D,3} = \{D2\ \}$
E	2	$B_{E,1} = \{B1, C1\ \}$	$B_{E,2} = \{E1, C2, B2\ \}$	E3 does not exist
F	3	$B_{F,1} = \{B1, C1\ \}$	$B_{F,2} = \{F1, C2, B2\}$	$B_{F,3} = \{F2, B3, C3\}$
G	1	$B_{G,1} = \{D1, E1\}$	G2 does not exist	G3 does not exist
H	3	$B_{H,1} = \{D1, E1\}$	$B_{H,2} = \{H1, F2, C2\}$	$B_{H,3} = \{H2, C3, F3\}$

$$p_i(t) = \begin{cases} \text{“On Air”}, & \text{if aircraft is in flight} \\ \text{Airport } p \in P, & \text{if aircraft is on ground} \\ \text{Airport } p = 0, & \text{if } t = 1 + \Delta t((L \text{ div } \Delta t) - 1) \end{cases}$$

Knowledge of all the following parameters fully represents all input data required to start solving the LMSP problem.

Parameters:

–D_i: deadline of task i
–C_i: due time of task i
–Q_i: release time of task i
–T_i: processing time or duration of task i
–I_i: interval of task i after it is converted to a common unit of time
–r_i^k: amount of resource $k \in K$ consumed by job i
–c_i: cost of leaving non-mandatory task $i \in N$ unscheduled (i.e., cost of assigning non-mandatory task i to the maintenance opportunity at the fictitious airport $p = 0$)
–a_i: penalty cost per unit of earliness for a job i
–b_i: penalty cost per unit of delay for a job i
–l_m: starting time of maintenance opportunity m
–u_m: finishing time of maintenance opportunity m
–L: duration of the planning horizon in minutes
–ε: Tolerance value used in strict inequalities
–Max_i: the maximum number of executions possible for task i within the planning horizon
–$R_k^{(t,p)}$: the amount of resource type $k \in K$ available at airport p at time t before any of the tasks in set J is scheduled (i.e., before the scheduling process)
–Δt: the temporal resolution of planning horizon discretization
–$\text{YUL}_{i,e}$: maximum possible earliness of the e^{th} execution of task i
–OT: sum of the maximum number of executions possible for all mandatory tasks combined (i.e., $OT = \sum_{i \in O} \text{Max}_i$). This represents the actual number of mandatory tasks the model is scheduled, including several occurrences of each task

The following are the variables used in the LMSP of this work. The model contains only one decision variable, which is the binary variable $S_{i,e}^{(t,p_i(t))}$ and the rest of the variables are used to derive the value of the objective function and to obtain a feasible optimal solution.

Variables:

–$S_{i,e}^{(t,p_i(t))}$: binary indicating if time t is the optimal starting time of the e^{th} execution of task i
–$R_{i,e}$: binary that indicates if the execution number e of task i will be scheduled within the planning horizon. Only executions that have their deadline before the planning horizon will be scheduled (i.e., if $D_{i,e} > L$ then $R_{i,e} = 0$ else $R_{i,e} = 1$)
–$x_i^{(t,p_i(t))}$: binary indicating if at time t the e^{th} execution of task i will be under execution
–$Y_{i,e}$: the actual earliness of the e^{th} execution of task i
–$Z_{i,e}$: the tardiness of the e^{th} execution of task i
–fvalue: the value of the objective function

3.1.2 **Time-indexed MILP formulation of the LMSP**

The proposed time-indexed MILP formulation of the LMSP is as follows:

$$\min \sum_{\substack{i\in O\\ e\in E_i}} (a_i \cdot Y_{i,e} + b_i \cdot Z_{i,e}) + \sum_{i\in N} c_i \cdot S_i^{[(L \text{ div } \Delta t)+\Delta t],0} \quad (1)$$

Objective function (1) comprises the penalty costs of the total earliness and tardiness of scheduled mandatory tasks. Expression 1 also includes the sum of the penalty cost c_i of assigning $t = 1 + \Delta t((L \text{ div } \Delta t) - 1)$ as the starting time of the non-mandatory task i executed at dummy airport $p = 0$. This represents the cost of hiring additional staff or outsourcing the completion of the task to a third-party company. Since the objective function is to be minimized, the function necessitates that a scheduled mandatory task is either early or late (i.e., for each mandatory task either $Z_{i,e}$ or $Y_{i,e}$ is equal to zero, and the other has a positive value). The following constraints are applied to the LMSP:

$$\sum_{t\in Ps_{i,e}} S_{i,e}^{t,p_i(t)} = R_{i,e} \forall\ i \in J,\ e \in E_i \quad (2)$$

$$R_{i,1} = 1 \forall\ i \in J \quad (3)$$

$$[(T_i \text{ div } \Delta t) + 1]\, S_{i,e}^{t,p_i(t)} \le \sum_{x=0}^{(T_i \text{ div } \Delta t)} X_{i,e}^{t+x\Delta t,p_i(t)} \forall\ i \in J,\ e \in E_i,\ t \in Ps_{i,e} \quad (4)$$

$$\left(\sum_{t\in Ps_{i,e-1}} t \cdot S_{i,e-1}^{t,p_i(t)} + I_i + T_i\right) - L \cdot R_{i,e-1} \ge -L \cdot R_{i,e} + \varepsilon(1 - R_{i,e}) \forall\ i \in O,\ e \in E_i' \quad (5)$$

$$Z_{i,1} - Y_{i,1} = \sum_{t\in Ps_{i,1}} t \cdot S_{i,1}^{t,p_i(t)} + T_i - C_i \forall\ i \in O \quad (6)$$

$$Z_{i,e} - y_{i,e} = \sum_{t\in Ps_{i,e}} t \cdot S_{i,e}^{t,p_i(t)} - \sum_{t\in Ps_{i,e-1}} t \cdot S_{i,e-1}^{t,p_i(t)} - C_i + Q_i \forall\ i \in O,\ e \in E_i' \quad (7)$$

$$Y_{i,e} \le R_{i,e} \cdot \text{YUL}_i \forall\ i \in O,\ e \in E_i' \quad (8)$$

$$Y_{i,e} \le y_{i,e} \forall\ i \in O,\ e \in E_i' \quad (9)$$

$$Y_{i,e} \ge y_{i,e} - \text{YUL}_i \cdot (1 - R_{i,e}) \forall\ i \in O,\ e \in E_i' \quad (10)$$

$$\sum_{i\in J_{t,p}} X_{i,e}^{t,p} \cdot r_i^k \le R_k^{t,p} \forall\ p \in P,\ t \in DT,\ k \in K \quad (11)$$

$$X_{i,e}^{t,p_i(t)} + X_{u,g}^{t,p_u(t)} \le 1 \forall\ i \in O,\ e \in E_i, g \in E_u, u \in U_i,\ t \in (Px_{i,e} \cap Px_{u,g}) \quad (12)$$

$$\sum_{t\in Ps_{i,e}} t \cdot S_{i,e}^{t,p_i(t)} \ge T_b + \sum_{t\in Ps_{b,e}} t \cdot S_{b,e}^{t,p_b(t)} - M(1 - R_{i,e}) \forall\ i \in O,\ e \in E_i,\ b \in B_{i,e} \quad (13)$$

$$S_{i,e}^{t,p_i(t)}, X_{i,e}^{t,p_i(t)}, R_{i,e} \in \{0,1\} \forall\ i \in J,\ e \in E_i, t \in Ps_{i,e} \quad (14)$$

$$Z_{i,e}, Y_{i,e}, y_{i,e} \ge 0 \forall\ i \in J,\ e \quad (15)$$

Constraint (2) defines $R_{i,e}$ to have a value of 1 if the e^{th} execution of task i was assigned to one of its possible starting times $Ps_{i,e}$ within the planning horizon L (i.e., if $\sum_{t\in Ps_{i,e}} S_{i,e}^{t,p_i(t)} = 1$). The function $p_i(t)$ is to indicate at which airport the aircraft associated with task i will be at time t. Constraint (3), along with constraint (2) forces the 1^{st} execution of all mandatory tasks and non-mandatory tasks to be assigned to one of their possible starting times $Ps_{i,e}$. Note that all non-mandatory tasks have only one occurrence within the planning duration.

Constraint (4) assigns a value of 1 for all the binary decision variable $X_{i,e}^{t,p_i(t)}$ until $X_{i,e}^{t+(T_i \text{ div } \Delta t)\Delta t,p_i(t)}$ in increments of Δt if the optimal starting time of the e^{th} execution of task i is at time t. $[(T_i \text{ div } \Delta t) + 1]$ represents one plus the quotient of $T_i/\Delta t$ (i.e., the number of time points in the discretized planning horizon that task i requires to be executed). Since some repetitions of a mandatory task can be left unscheduled if they would have a deadline that exceeds L, this means that $S_{i,e}^{t,p_i(t)}$ can have a value of 0 if $R_{i,e} = 0$ as indicated by constraint (1). Consequently, constraint (4) is of the inequality type to accommodate zero value for $S_{i,e}^{t,p_i(t)}$ and $X_{i,e}^{t,p_i(t)}$. $S_{i,e}^{t,p_i(t)} = 0$ in this model means that the e^{th} execution of task i is not scheduled within the planning horizon.

When one of the occurrences of a mandatory task has a deadline within the planning horizon L, constraint (5) schedules this occurrence within the planning horizon by forcing $R_{i,e}$ to take a value of 1. The deadline D_i of the 1^{st} occurrence of any task is fixed and given in the problem,

however for the repeated executions, the deadline depends on the starting time of its immediately preceding occurrence $\sum_{t \in Ps_{i,e-1}} t \cdot S_{i,e-1}^{t,p_i(t)}$, the task duration T_i and the task interval I_i. Furthermore, constraint (5) prevents scheduling a mandatory task with an immediately preceding unscheduled occurrence. For mandatory tasks, expressions (6) and (7) define the earliness/lateness of tasks performed for the 1st time and subsequent executions, respectively. In this research, we consider earliness/lateness to be measured between the due time and the actual finish time of a task. For the 1st execution of tasks, the due time is given in the problem as C_i. However, for repeated executions, the due time depends on the starting time of its immediately preceding occurrence and how far is the due date away from the release date $C_i - Q_i$ (i.e., the safety time buffer).

$Z_{i,e}$ in constraint (7) could have a false positive value when an execution of a task is scheduled even though its immediately preceding occurrence has a starting time of 0 (i.e., it will be left unscheduled). To prevent this, precedence constraint (13) is formulated to avoid violations in terms of the precedence relationship between occurrences of tasks and between different tasks as shown in Figure 1. Similarly, $y_{i,e}$ will have a misleading positive value if a task is decided to be left unscheduled (i.e $R_{i,e} = 0$ and $S_{i,e}^{t,p_i(t)} = 0$) while its immediately preceding occurrence is scheduled. Constraints (8) to (10) tackle this by defining a decision variable $Y_{i,e}$ that represents the actual true earliness of mandatory tasks. True earliness $Y_{i,e}$ will have a value of zero if $R_{i,e} = 0$ and a $y_{i,e}$ value if $R_{i,e} = 1$.

Constraint (11) is the resource constraint ensuring that at any point in the discretized planning horizon, the resource available at an airport $R_k^{t,p}$ (including fictitious airport $p = 0$) is equal to or greater than the total resource demand of all tasks being executed at the same moment and at the same airport. For non-mandatory tasks assigned to be conducted beyond the planning horizon L at fictitious airport $p = 0$, an unlimited amount of resource availability at any moment beyond L will be assigned to the airport $p = 0$, so the resource constraint is always satisfied if $p = 0$. Constraints (12) ensures that occurrences of mandatory task $i \in O$ are never done simultaneously with any other task in their non-simultaneity set $U_{i,e}$. Finally, constraints (14) and (15) define the binary and the positive mixed integer nature of the problem decision variables.

4 Solution Methodology

The solution to the LMSP is split into two sequential stages: a pre-processing stage and an exact branch-and-bound-based algorithm stage. The pre-processing stage defines Max_i, $Ps_{i,e}$, $Pe_{i,e}$, $J_{t,p}$ and YUL_i. In this stage, airport restrictions, release time, and deadline constraints are considered as well as making sure that tasks start and finish within a turnaround time without interruption. The results of the first stage are then used in the second stage to finalize the solution of the LMSP by applying and optimality solving the discussed MILP model using commercial and academic MILP solvers.

4.1 Pre-processing stage

Given the first release time of a task, along with its interval, and due date, it is possible to compute Max_i, which is the maximum number of possible executions of this task within the planning duration. Following the work of Max Witteman et al. in [25] the following steps can be used for each mandatory task:

- **Step 1:** set the maximum possible occurrences of task i to be 1 (i.e., $\text{Max}_i = 1$) then determine the set $M_{i,1}$ containing all valid maintenance opportunities for the 1st execution of task i. A valid maintenance opportunity is one that is at an allowed airport based on the set P_i and the maintenance opportunity has an upper bound u_m greater than task i release time Q_i by at least the tasks duration T_i (i.e $u_m \geq Q_i + T_i$) and that has a lower bound l_m smaller than task i deadline D_i by at least the tasks duration T_i (i.e $l_m \leq D_i - T_i$).
- **Step 2:** If there is only one maintenance opportunity in $M_{i,1}$ then the final maximum possible occurrences of task i is $\text{Max}_i = 1$. Otherwise, set the release date of the second execution of task i (i.e $Q_{i,2}$) to be equal to the upper bound of the earliest maintenance opportunity in $M_{i,1}$ (i.e., $Q_{i,2} = \min_m(u_m) \ \forall \ m \in M_{i,1}$) or $Q_{i,2}$ to be equal to $D_{i,1}$ if $D_{i,1} < \min_m(u_m)$. Accordingly, set the deadline of the second execution of task i as $D_{i,2} = Q_{i,2} + T_i + I_i$.
- **Step 3:** If the deadline for the 2nd execution of the task exceeds the planning horizon L (i.e $D_{i,2} > L$), then the final maximum possible occurrences of task i is Max_i. Otherwise, increase the maximum possible occurrences of task i by 1 (i.e., $\text{Max}_i = \text{Max}_i + 1$)
- **Step 4:** Determine the set containing all valid maintenance opportunities for the 2nd execution of task i (i.e., $M_{i,2}$) based on $Q_{i,2}$ and $D_{i,2}$ then go to step 2 to proceed with the following occurrences.

These steps will result in the earliest possible release time and the latest possible deadline for every possible execution of each mandatory task. Each possible execution of a task is treated as a single task to be scheduled in this problem, therefore the parameter Max_i is the most influential parameter in terms of the size of the problem. Values of Max_i are usually high and unrealistic because it is calculated assuming that task i will be executed in every maintenance opportunity after the release time of the first execution $Q_{i,1}$ as long as the maintenance slot is at one of the allowed airports and has enough duration. A constant maximum upper limit for

Max_i could be safely used to put a limit to the size of the problem since the model always results in very few scheduled occurrences for each task within the planning compared to the value of Max_i as discussed in the solution section. The feasibility of the solution could be easily checked, and the upper limit of Max_i could be incremented if it resulted in an infeasible solution until there is no significant difference in the value of the objective function. The next step in the pre-processing stage is to define $Ps_{i,e}$, and $Pe_{i,e}$. $Ps_{i,e}$ is the set that contains all possible starting times in T_0 for the e^{th} execution of task i including $t = 1 + \Delta t((L \text{ div } \Delta t) - 1)$ if $i \in N$. At this stage, a possible starting point in $Ps_{i,e}$ apart from $t = 1 + \Delta t((L \text{ div } \Delta t) - 1)$ is the one that satisfies the following three conditions:

1. It lies between the release date of the task and its deadline minus the task duration since we consider the deadline to be the last time the task is allowed to be finished and closed at (i.e $Q_{i,e} \leq t \leq D_{i,e} - T_i$)
2. It lies within the lower bound and the upper bound minus the task's duration of any of the suitable maintenance opportunities $m \in M_{i,e}$ of task i (i.e $l_m \leq t \leq u_m - T_i$). This is the only point in the solution that ensures the task is conducted when the aircraft is on the ground. M_i contains all turnaround times or overnight stays that fully or partially fall within the release date and deadline of occurrence number e of task i.
3. A time when the aircraft will be at an airport in the allowed airport set P_i of the task.

$Pe_{i,e}$ represents a larger set containing all possible execution times of task i where it can be considered in the process of being executed. A time t is regarded as an element from $Pe_{i,e}$ if it satisfies the same three conditions as $Ps_{i,e}$ but the upper limit of the inequalities of conditions 1 and 2 will not include the subtraction of T_i. So, t is an element of $Pe_{i,e}$ if $Q_{i,e} \leq t \leq D_{i,e}$ (condition 1), $l_m \leq t \leq u_m$ (condition 2), and the third condition is that the task should be conducted at one of its allowed airports. In addition to these elements, for non-mandatory tasks $i \in N$, $Pe_{i,e}$ will further contain all time points starting from $t = 1 + \Delta t((L \text{ div } \Delta t) - 1)$ in increments of Δt until the non-mandatory task duration T_i is reached or exceeded. After defining $Pe_{i,e}$, set T_0 can be redefined in a more compact set $T_0 = \bigcup_{i \in J} Pe_{i,e}$ to include only relevant time points, which contributes to fewer resource constraints.

$\text{YUL}_{i,e}$ is the maximum possible earliness of the e^{th} execution of task i, which can be calculated as $Q_{i,e} + (C_{i,1} - Q_{i,1}) - (\min(Ps_{i,e}) + T_i)$ which assumes that the task starts at the earliest possible starting time in set $Ps_{i,e}$. The last set to be computed in the pre-processing stage is $J_{t,p}$. $J_{t,p}$ represents the list of all tasks, including non-mandatory tasks that can be processed at airport p at time t. A task i with any of its occurrences is an element of $J_{t,p}$ if $t \in Pe_{i,e}$ and $p_i(t) = p$.

4.2 Exact solution algorithm for the time-indexed MILP formulation of the LMSP

The output of the pre-processing stage can be directly utilized in defining the discussed time-indexed MILP formulation, which will be solved optimally. This formulation makes branch-and-bound-based MILP commercial and academic solvers such as CPLEX and Gurobi suitable for solving this model. Therefore, a commercial solver will be used in this work to solve the LMSP. One main advantage of using the exact MILP solution to the LMSP is that all constraints are considered simultaneously in an integrated way in each step of the solver contributing to the optimal objective function value.

5 Results and discussions

5.1 Test Environment

To validate the model and capture the performance of the solution approach under different inputs, a set of instances was created, shown in Table 2. The cases increase in complexity and size mainly based on the total number of tasks to be scheduled $|J|$, the planning horizon L, and the fleet size A. The smallest instance is the 1st one in Table 2 which shows a fleet of 5 aircraft with a total of 51 tasks with a mean task interval $\bar{I}$ of 135 hours, all to be scheduled for a planning duration of 10 days. On the other hand, the largest instance is the last one in Table 2, which shows a fleet of 11 aircraft with a total of 248 mandatory tasks and 83 non-mandatory tasks to be allocated to 1312 available maintenance opportunities in 4 different airports.

Table 2: Test instances characteristics

A/L/J	|N|	|O|	|P|	|M|	$\bar{I}$ (hours)
5/10/51	13	38	4	215	135
5/20/100	25	75	5	418	261
6/10/60	15	45	5	227	117
6/20/120	30	90	6	480	260
7/20/140	35	105	4	555	265
7/30/211	53	158	4	880	395
8/20/160	40	120	6	625	269
8/30/240	60	180	5	949	395
9/20/180	45	135	4	736	263
9/30/271	68	203	4	1049	395
10/20/200	50	150	4	777	260
10/30/300	75	225	6	1185	401
11/20/220	55	165	5	863	266
11/30/331	83	248	4	1312	400

The required input parameters for each instance are randomly generated within realistic bounds based on Table 3. The planning horizon L, the number of available maintenance stations $|P|$, and the fleet size A were the main independent parameters that change the size of the problem to produce Table 2. Ideally, the temporal resolution Δt of the planning horizon discretization would be 1 minute to get a very accurate schedule, however since this will dramatically increase the size of the LMSP, especially for a long planning horizon, Δt was set as 5 minutes for planning horizons up to 20 days, and 10 minutes if the planning horizon is 30 days. This is acceptable since, practically, it is very likely that a maintenance team will not be able to start each scheduled task with an accuracy of 1 minute. By increasing Δt and choosing small correct value for the upper limit of Max_i, the problem can be solved requiring less RAM memory.

A flight schedule for the whole planning horizon is generated for each instance which represents the departure and arrival time of each aircraft in the fleet and the airport it is flying to. The schedule is constructed in a way that makes it cyclic, meaning that at the end of the planning horizon, each aircraft will be at the same airport it started from at the beginning of the planning horizon. For all instances, it was assumed that the manpower availability for the three types of resources (technicians, avionics Engineers, and Airframe and Engine Engineers) at night hours is reduced by 25% and that at least one airport will have the highest number of resources to represent the central maintenance hub of the airline.

To include precedence, non-simultaneity, and allowed airport constraints, the sets $B_{i,e}$, $U_{i,e}$ and P_i were randomly generated for a few random mandatory tasks. A high-cost penalty $(c_i = 50)$ of leaving any non-mandatory task unscheduled is set to allow scheduling as many non-mandatory tasks as possible within the planning horizon. The cost penalty for each unit of earliness and tardiness of a task is set to 1, although it could be changed to represent the priority of scheduling each task. For each instance, there are nearly three times more mandatory tasks than non-mandatory tasks since an airline maintenance team conducts most of their maintenance tasks

5.2 Computational results

In this section, the results of each instance in Table 2 is presented using the exact method. Conclusions were drawn regarding the accuracy and efficiency of the model and the solution method. All cases were implemented in Matlab R2023 and solved using Gurobi Optimizer (v.10.0.1) on a computer equipped with an AMD Ryzen 5 6600H 3.30 GHz processor and 16 GB of RAM.

Table 4 summarizes the results of the instances discussed. Column 1 represents the instance in the form of A/L/J where fleet size is A, the planning horizon in days is L, and the total number of tasks is J. Column 2 represents the number of mandatory tasks that the model schedules (i.e $\sum_{i \in O} \text{Maxe}_i$) including all possible occurrences of each task without putting an upper limit on the size of the problem. Column 4 is the same as column 2, but after imposing an upper limit (values in column 3) to the maximum possible executions of each task (i.e Max_i). Values in column 4 are the number of mandatory tasks the model allocates starting times to; therefore, it is the most expressing parameter of the problem size. Columns 2, 3, and 4 are outputs of the pre-processing stage, while columns 5, 6, 7 and 8 are outputs of the exact optimizer stage of the solution.

Column 5 represents the actual number of scheduled mandatory tasks, including several occurrences of each task. Occurrences of mandatory tasks that were left unexecuted (i.e $R_{i,e} = 0$) are included in OT of column 2 but not in values of column 5; therefore, the subtraction of these two columns represents the number of unexecuted occurrences of mandatory tasks. Columns 6 and 7, respectively, represent the value of the objective function and the execution time of each instance, including the pre-processing stage and optimizer duration combined. Finally, the last column represents in percentage the ratio of the executed subsequent mandatory task to the total (executed plus unexecuted) number of subsequent mandatory tasks of the LMSP instance. It is calculated as $100 \times ((\text{Number of scheduled mandatory tasks} - |O|)/(OT - |O|))$. The number of non-mandatory tasks left unscheduled is not included in Table 4 since the model was able to schedule all non-mandatory tasks within the planning horizon for all instances. This is because there are much fewer constraints related to non-mandatory tasks.

It can be seen that very few subsequent occurrences are executed in the optimal solution for each instance. This indicates that the average percentage of performed subsequent mandatory tasks is only 0.5%. The highest percentage, 3.85%, was in instance 6/10/60, which had no limit for Max_i. In this instance, among 565 mandatory tasks, including several occurrences of each task, only 65 were executed, where 45 out of these 65 tasks are the 1st occurrences that must be executed based on constraint (3), leaving only 20 executed subsequent mandatory tasks out of 520 subsequent possible occurrences, i.e., 3.85%. This is further illustrated for each task for the same instance in Figure 2, showing how many occurrences of each task were executed and how many were to be scheduled. Figure 2 also illustrates the maximum possible occurrences assuming an upper limit of 6. This small percentage can be explained as each occurrence has its own contribution (earliness/lateness) to the objective function, which is to be minimized, so the model tries to find the optimal balance between reducing the occurrences in favor of increased earliness/lateness of some executions. This balance must be considered in a whole integrated way to the problem, which makes the usage of heuristic methods challenging.

Table 3: Parameters and bounds used to create the test instances and the problem inputs.

Parameter	**Range**
The temporal resolution of planning horizon discretization	$\{5,10\}$: $\Delta t = 5$ if $L \leq 20$ days and $\Delta t = 10$ if $L = 30$ days
Maximum daily aircraft utilization in flight hours for each aircraft	12 hours (i.e., the sum of all flight time of an aircraft in a day should not exceed 12 hours)
Minimum daily aircraft utilization in flight hours for each aircraft	8 hours
Maximum allowed duration of a flight	10 hours
Minimum allowed duration of a flight	1 hour
Minimum slot duration	3 hours
Maximum slot duration	12 hours (overnight stay)
Maximum number of flights per day per aircraft	6
Minimum number of flights per day per aircraft	2
Number of technicians available at an airport or maintenance station (R_1)	$[40,70]$
Number of airframes and engine engineers available at an airport or a maintenance station (R_2)	$[20,50]$
Number of avionics engineers available at an airport or a maintenance station (R_3)	$[10,40]$
Reduction percentage in manpower availability at night hours (6 pm to 6 am)	25%
Task processing duration T_i	$[100,310]$ minutes
Task period interval I_i	$[1,19]$ days
Number of technicians required by a task (r_1)	$\{3,4,5,6\}$
Number of airframes and engine engineers required by a task (r_2)	$\{2,3\}$
Number of avionics engineers required by a task (r_3)	$\{1,2\}$
Cost penalty for leaving any non-mandatory task unscheduled	50
Cost penalty for each minute of earliness or tardiness for a mandatory task (a_i, b_i)	1
Due time Safety Buffer bf_i	15 hours
Due time C_i	$C_i = D_i - 15$ hours
Fleet size A	$\{5,6,7,8,9,10,11\}$
Types of resources K	3
Number of airports P	$\{4,5,6\}$
Planning horizon L	$\{10,20,30\}$ Days

If Max_i is set to be 1, meaning the LMSP relaxed the constraints regarding the possibility of executing a task several times within the planning horizon; the values of the objective function will be smaller than the values in column 6, which is desirable; however, the solution will not be realistic since some tasks will have to be executed more than once within the planning horizon based on their period. It is possible to eliminate the need to execute a task several times by adding constraints forcing the duration between the end of the planning horizon L and the finish time of the 1st occurrence of each task to be greater than the task interval, thus making sure that each task will appear only once within the planning horizon. Such a schedule is already considered in the LMSP of this work, yet it was not the case in any of the results. This is because tasks will be more likely to be scheduled closer to their deadline to meet the added constraints, which will result in much higher delay and a higher value of the objective function, especially when the penalty of tardiness is more than the penalty of earliness which is usually the case for an airline.

Since it is always the case that very few percentages (0.5% on average) of the possible subsequent occurrences of mandatory tasks are actually executed, as shown in Figure 2, it is suitable if needed for large problems with long planning horizon to put an upper limit for the maximum possible occurrences of each task Max_i. This decreases the size of the problem without significantly affecting the solution. An upper limit for Max_i could be chosen by initially making it 3 and then incrementing it until there is no significant effect on the objective function value. Applying this approach to the test instance decreased the number of mandatory tasks to be scheduled

Table 4: Computational results for each instance

A/L/J	OT	Upper limit of Max_i	OT after putting an upper limit for Max_i	Number of scheduled mandatory tasks	fval	Solution time (sec)	Percentage of scheduled occurrences of mandatory tasks %
5/10/51	396	Inf	396	40	12586	104	0.56
5/20/100	1626	3	217	78	28778	430	0.19
6/10/60	565	Inf	565	65	9873	163	3.85
6/20/120	1941	5	355	95	15432	527	0.27
7/20/140	1933	4	209	107	13533	183	0.10
7/30/211	5012	4	471	161	54875	640	0.06
8/20/160	2447	5	239	124	13131	1882	0.17
8/30/240	5340	6	889	185	31949	1214	0.10
9/20/180	2948	7	669	148	37714	1489	0.46
9/30/271	6222	5	1009	214	23916	818	0.18
10/20/200	3127	6	597	159	13104	706	0.30
10/30/300	6689	4	896	236	63009	6336	0.17
11/20/220	3269	5	493	171	36853	5990	0.19
11/30/331	8118	6	738	268	40126	7468	0.25
Average	**3545**	**–**	**553**	**146.5**	**28207**	**2013**	**0.5**

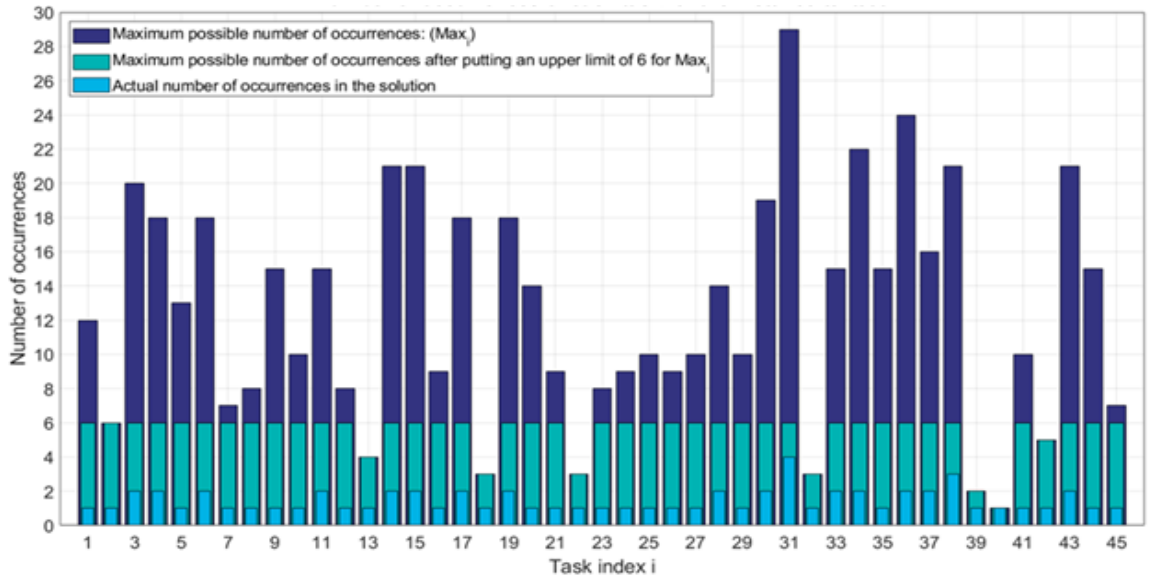

Fig. 2: Number of occurrences of each task for the instance 6/10/60

on average from 3545 tasks to 553 tasks, around an 84.4% reduction in size. Moreover, on average, the solution time was 2013 seconds (33 min 33 sec) which makes it applicable for everyday use as well and for trying different values for the upper limit of Max_i.

6 Conclusion

Nowadays, due to the recent development in aviation technology, many aircraft maintenance duties can be performed during the turnaround time (TAT) between two flights and during overnight layovers for airports with night-flying restrictions. This paper aims to utilize this flexibility by scheduling as many line maintenance tasks as possible on turnaround time (TAT) and overnight layovers in an approach that makes the most out of the lifetime of the aircraft components. This is known as the Line Maintenance Scheduling Problem (LMSP), which has the objective of minimizing the sum of the deviation of tasks from their due times, providing better usage of aircraft operation, ground time, manpower, and other resources in the line maintenance environment which can help improve the efficiency of aircraft maintenance and

help balance the workload at maintenance bases throughout the peak periods. In the LMSP variant studied in this research, we provide a new compact time-indexed MILP optimization model capturing several realistic conditions, including a) precedence relationship between tasks, b) simultaneous job inclusions/exclusions, c) cyclical nature of tasks, d) maintenance airport restrictions, e) release time and deadline constraints, and finally f) time-varying resource capacities at airports.

Several virtual airline models and instances were generated and tested using a pre-processing stage that defines the time-indexed formulation, which is then optimally solved using a commercial MILP solver. It was clear from the results that reducing the frequency of tasks within the planning horizon, even if it led to greater deviation from due time for some tasks, was more beneficial regarding the overall objective function. This makes it possible to significantly decrease the size of the problem without affecting the solution by setting an upper limit for the maximum assumed possible occurrences of each task within the planning horizon. In addition, the results showed an average execution time of 33 minutes, making it a practical tool under realistic conditions.

Some potential directions for future research in this field include developing heuristic and metaheuristic techniques that would provide near-optimal solutions for the variants of the LMSP with larger fleets and longer planning durations under reasonable solution time. Apart from considering time-dependent resource consumption of tasks, more realistic resource constraints could be incorporated by allowing optional and mandatory overtime for workers at a cost penalty to the objective function. The problem could also be used to investigate improvements in developing flight schedules and resource capacity allocation. Finally, more robust solutions could be studied using dynamic optimization and buffer management to tackle the uncertainty and stochasticity associated with unplanned corrective maintenance and unexpected flight delays causing variations in turnaround times, task processing times, workforce, and resource availability.

7 Acknowledgment

The authors extend their appreciation to JADARA University in Jordan to support this work.

References

[1] Kožović, D.V., et al., *Air Traffic Modernization and Control: ADS-B System Implementation Update 2022–a Review.* Fme Transactions, 2023. **51**(1).
[2] Cook, G.N. and B.G. Billig, *Airline operations and management: a management textbook*. 2023: Routledge.
[3] Kinnison, H.A. and T. Siddiqui, *Aviation maintenance management.* 2013.
[4] Aubin, B.R., *Aircraft Maintenance: The art and science of keeping aircraft safe*. 2004: SAE International.
[5] Stolzer, A.J., R.L. Sumwalt, and J.J. Goglia, *Safety management systems in aviation*. 2023: CRC Press.
[6] Tisdall, L., *Directions in Australian general aviation: contemporary issues in planning and policy*. 2021, University of Southern Queensland.
[7] Jennions, I.K., *Integrated vehicle health management: the technology*. 2013: SAE International.
[8] Gupta, P., M. Bazargan, and R. McGrath. *Simulation model for aircraft line maintenance planning*. in *Annual Reliability and Maintainability Symposium, 2003.* 2003. IEEE.
[9] Stone, R. and J.K. Ball, *Automotive engineering fundamentals*. 2004: SAE International.
[10] Aubin, B.R., *Aircraft Maintenance*. 2004, SAE Technical Paper.
[11] Manukaji John, U., *Categorisation Of Clay Deposits In The Federal Capital Territory Of Abuja.*
[12] Senturk, C. and I. Ozkol, *The effects of the use of single task-oriented maintenance concept and more accurate letter check alternatives on the reduction of scheduled maintenance downtime of aircraft.* International Journal of Mechanical Engineering and Robotics Research, 2018. **7**(2): p. 189–196.
[13] Shaukat, S., et al., *Aircraft line maintenance scheduling and optimisation.* Journal of Air Transport Management, 2020. **89**: p. 101914.
[14] Shaukat, S., *Aircraft line maintenance scheduling optimisation-a heuristic approach*. 2015, UNSW Sydney.
[15] Morales-España, G., J.M. Latorre, and A. Ramos, *Tight and compact MILP formulation for the thermal unit commitment problem.* IEEE Transactions on Power Systems, 2013. **28**(4): p. 4897–4908.
[16] Van den Bergh, J., et al., *Aircraft maintenance operations: state of the art.* HUB Research Paper 2013/09, 2013.
[17] Hu, Y., et al., *Reinforcement learning-driven maintenance strategy: A novel solution for long-term aircraft maintenance decision optimization.* Computers & industrial engineering, 2021. **153**: p. 107056.
[18] Eltoukhy, A.E., F.T. Chan, and S.H. Chung, *Airline schedule planning: a review and future directions.* Industrial Management & Data Systems, 2017. **117**(6): p. 1201–1243.
[19] Fuentes, M., et al., *A novel approach to the tail assignment problem in airline planning.* Transportation Research Procedia, 2021. **58**: p. 53–60.
[20] Khaled, O., et al., *A compact optimization model for the tail assignment problem.* European Journal of Operational Research, 2018. **264**(2): p. 548–557.
[21] Fuentes, M., et al., *The tail assignment problem: A case study at vueling airlines.* Transportation Research Procedia, 2021. **52**: p. 445–452.
[22] Eltoukhy, A.E., et al., *Heuristic approaches for operational aircraft maintenance routing problem with maximum flying hours and man-power availability considerations.* Industrial Management & Data Systems, 2017. **117**(10): p. 2142–2170.
[23] Pazhooh, S.F. and H.S. Shemirani, *An Efficient Continuous-Time MILP for Integrated Aircraft Hangar Scheduling and Layout.* arXiv preprint arXiv:2508.02640, 2025.
[24] Giacotto, A., H.C. Marques, and A. Martinetti, *Prescriptive maintenance: a comprehensive review of current research*

and future directions. Journal of Quality in Maintenance Engineering, 2025. **31**(1): p. 129–173.
[25] Zhang, Q., et al., *A heuristic maintenance scheduling framework for a military aircraft fleet under limited maintenance capacities.* Reliability Engineering & System Safety, 2023. **235**: p. 109239.
[26] Gavranis, A. and G. Kozanidis, *An exact solution algorithm for maximizing the fleet availability of a unit of aircraft subject to flight and maintenance requirements.* European Journal of Operational Research, 2015. **242**(2): p. 631–643.
[27] Andrade, P., et al., *Aircraft maintenance check scheduling using reinforcement learning.* Aerospace, 2021. **8**(4): p. 113.
[28] Safaei, N. and A.K. Jardine, *Aircraft routing with generalized maintenance constraints.* Omega, 2018. **80**: p. 111–122.
[29] Li, H., et al., *Optimal combination of aircraft maintenance tasks by a novel simplex optimization method.* Mathematical Problems in Engineering, 2015. **2015**(1): p. 169310.
[30] Van Buskirk, C., et al., *Computer-aided aircraft maintenance scheduling.* Institute for Software-Integrated Systems, 2002.
[31] Deng, Q. and B.F. Santos, *Lookahead approximate dynamic programming for stochastic aircraft maintenance check scheduling optimization.* European Journal of Operational Research, 2022. **299**(3): p. 814–833.
[32] Lagos, C., F. Delgado, and M.A. Klapp, *Dynamic optimization for airline maintenance operations.* Transportation Science, 2020. **54**(4): p. 998–1015.
[33] Ozkol, I. and C. Senturk. *The effects of the use of single task-oriented maintenance concept and more accurate letter check alternatives on the reduction of scheduled maintenance downtime of aircraft*. in *2017 8th International Conference on Mechanical and Aerospace Engineering (ICMAE)*. 2017. IEEE.
[34] Senturk, C., M.S. Kavsaoglu, and M. Nikbay. *Optimization of aircraft utilization by reducing scheduled maintenance downtime*. in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*. 2010.
[35] Öhman, M., et al., *Frontlog scheduling in aircraft line maintenance: From explorative solution design to theoretical insight into buffer management.* Journal of Operations Management, 2021. **67**(2): p. 120–151.
[36] Xiong, H., et al., *A survey of job shop scheduling problem: The types and models.* Computers & Operations Research, 2022. **142**: p. 105731.
[37] Kleeman, M.P. and G.B. Lamont. *Solving the aircraft engine maintenance scheduling problem using a multi-objective evolutionary algorithm*. in *Proceedings of the 7th annual workshop on Genetic and evolutionary computation*. 2005.
[38] Witteman, M., Q. Deng, and B.F. Santos, *A bin packing approach to solve the aircraft maintenance task allocation problem.* European Journal of Operational Research, 2021. **294**(1): p. 365–376.
[39] Curran, O.R., *A maintenance packaging and scheduling optimization method for future aircraft.* Air Transport and Operations, 2012: p. 343.
[40] Lufthansa Technik, A., *A Maintenance Packaging and Scheduling Optimization Method for Future Aircraft.*
[41] Hölzel, N., et al. *System analysis of prognostics and health management systems for future transport aircraft*. in *28th International Congress of the Aeronautical Sciences (ICAS), Brisbane, Australia*. 2012.
[42] Hartman, S. and D. Briskorn, *A survey of variants and extensions of the resource-constrained project scheduling problem.* Operations Research Management Science, 2011. **51**(1): p. 67.
[43] Hartmann, S. and D. Briskorn, *An updated survey of variants and extensions of the resource-constrained project scheduling problem.* European Journal of operational research, 2022. **297**(1): p. 1–14.
[44] Vanhoucke, M., E. Demeulemeester, and W. Herroelen, *An exact procedure for the resource-constrained weighted earliness–tardiness project scheduling problem.* Annals of Operations Research, 2001. **102**(1): p. 179–196.
[45] Abdolshah, M., *A review of resource-constrained project scheduling problems (RCPSP) approaches and solutions.* International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies, 2014. **5**(4): p. 253–286.
[46] Bagherinejad, J. and Z.R. Majd, *Solving the MRCPSP/max with the objective of minimizing tardiness/earliness cost of activities with double genetic algorithms.* The International Journal of Advanced Manufacturing Technology, 2014. **70**(1): p. 573–582.
[47] Bartusch, M., R. Möhring, and F. Radermacher, *Quantitative Models, Data Structuring and Information Processing.* Annals of Operations Research, 1988. **16**: p. 201–240.
[48] Dehesa Rodríguez, J., *Aplicación de algoritmos genéticos multiobjetivo al problema multiproyecto descentralizado: SPEA2-DRCMPSP.* 2024.
[49] Schwindt, C. and N. Trautmann, *Batch scheduling in process industries: an application of resource–constrained project scheduling.* OR-Spektrum, 2000. **22**(4): p. 501–524.
[50] Hartmann, S., *Project scheduling with resource capacities and requests varying with time: a case study.* Flexible Services and Manufacturing Journal, 2013. **25**(1): p. 74–93.