Journal of Air Transport Management 129 (2025) 102855

Contents lists available at ScienceDirect

Journal of Air Transport Management

journal homepage: www.elsevier.com/locate/jairtraman

Multi-agent task allocation and path planning for autonomous ground support equipment

Manouk van der Zwan[1], Gülçin Ermiş *, Alexei Sharpanskykh

Delft University of Technology, Faculty of Aerospace Engineering, Kluyverweg 1, Delft, 2629HS, The Netherlands

ARTICLE INFO

Keywords:
Automated aircraft ground handling
Task allocation
Multi-agent path planning
Vehicle routing with pick-ups and deliveries

ABSTRACT

We aim to contribute to the automation of ground handling tasks using autonomous ground support equipment (GSE) at airports. Automation of airside operations has recently become critical for the airports to achieve higher levels of safety and efficiency under growing traffic demand and requires solving a complex scheduling and path planning problem. To address this problem, we present a multi-agent task allocation and path planning model for handling airside operations on the apron. In the problem, the ground handling tasks are to be allocated to the equipment, the trips of vehicles should be scheduled within specific time windows considering the flight schedules, and the collisions of vehicles on the apron and service roads should be avoided. We present a centralized multi-agent task allocation and routing model which aims to optimize the allocation and routing of various types of ground handling tasks over a heterogeneous set of GSE vehicles. We convert the allocation and routing problem into vehicle routing problem with time windows, pick-ups, deliveries and solve the problem using a warm start mixed integer linear programming (MILP) model. We also introduce a nonlinear objective function which converts the MILP model into a mixed integer nonlinear programming (MINLP) model, to minimize the time service locations at the stands are occupied. Then, we solve the corresponding path finding problem to find collision free paths for the GSE, by the multi-agent path finding model. The proposed model outperforms the decentralized approach in previous research regarding the allocation rate of assigning tasks to vehicles and the performance indicators of finding conflict free paths, and in CPU time. The mean deviations from shortest paths were considerably small in path planning which means that the solution quality was high. Furthermore, the CPU time of allocating tasks has been reduced by 48% compared to the CPU time of decentralized allocation.

1. Introduction

Automation of airside operations is among the recent strategies in aviation to achieve higher levels of safety and efficiency. Royal Schiphol Group aims to make all vehicles and associated processes on the airside sustainable and autonomous at Amsterdam Schiphol Airport by 2050 Royal Schiphol Group (2020). Automation of ground support equipment plays a critical role in achieving fully autonomous airside operations. The ground support equipment will be replaced by autonomous emission-free vehicles in the near future, due to the autonomous airside operations program initiated by Royal Schiphol Group in 2020 Schiphol Innovation (2021). A similar trend is also observed at other large airports.

Aircraft ground handling operations play a central role in airside operations. They include various services carried out on the ground to ensure safe, efficient, and timely arrival and departure of the aircraft, such as aircraft refueling and passenger (de)boarding. A limited number of vehicles is used to complete ground operations at aircraft stands. The equipment is shared to serve the aircraft that are to be handled in different stands of aircraft. For automation, the allocation and scheduling, as well as the collision-free routing of ground support equipment are crucial.

Since many of the procedures for aircraft ground handling operations are standardized, the aircraft ground handling is a promising field for automation. Examples of tasks are refueling, catering or baggage handling, and the service locations around the aircraft for handling these tasks are usually similar Tabares and Mora-Camino (2017a). At Amsterdam Schiphol Airport, an autonomous baggage handling tractor is being tested for automation of baggage handling Schiphol News Room (2021).

* Corresponding author.
E-mail addresses: m.vanderzwan-1@student.tudelft.nl (M. van der Zwan), g.ermis@tudelft.nl (G. Ermiş), o.a.sharpanskykh@tudelft.nl (A. Sharpanskykh).
[1] Present e-mail and address: manouk.zwan@gmail.com, Gibbs Analytics Consulting, Coolsingel 104, Rotterdam, 3011 AG, The Netherlands.

https://doi.org/10.1016/j.jairtraman.2025.102855
Received 11 December 2024; Received in revised form 30 May 2025; Accepted 2 July 2025
Available online 18 July 2025

Existing research on task allocation and path planning focus on warehouse environments Li et al. (2020), Chen et al. (2021), Ma and Koenig (2016), Liu et al. (2019), where autonomous vehicles collaborate to execute tasks. Recently, Chen et al. (2023) introduced a new problem referred to as *automated aircraft ground handling* where autonomous ground support equipment collaborate to complete the tasks around the aircraft on aircraft stands. They designed a grid-based map environment that represent the area of the airport where ground handling takes place. This area is a part of a pier including several aircraft stands, a service road and exists to pier. The areas occupied by the standing aircraft and passenger boarding bridges were modeled as static obstacles on the grid-map. A group of operations including the refueling, catering, baggage handling, water and lavatory service tasks were aimed to be served around the aircraft at several stands by the autonomous refueling, catering, baggage handling, water and lavatory service trucks. Traveling on defined map required avoiding collisions with static and dynamic obstacles.

The defined aircraft ground handling problem is a more complex version of task allocation and path planning which includes also the time windows, strict traveling directions, precedence relations for pick-up and delivery tasks, and vehicle capacities as constraints. We focus on solving the defined aircraft ground handling problem with small variations in constraints and problem setting, using a centralized optimization approach for allocation and routing.

To solve the problem, Chen et al. (2023) presented a decentralized task allocation and multi-agent path planning model. Tasks were allocated using an optimization based auction, where a single vehicle pick-up and delivery optimization model was solved for each vehicle to bid on the announced task, at each round of auction. The goal of using a decentralized approach was to balance the interests of multiple agents, however the local optimization approach did not always guarantee to find a feasible allocation for all tasks even though it exists. In our research, we overcome this drawback using a centralized optimization which ensures finding feasible solutions for all vehicles or proving infeasibility otherwise while searching for a global optimum.

Furthermore, the existing route of each agent is iteratively re-optimized to find an optimal position to insert the candidate task in Chen et al. (2023), which contributed to the increase in CPU time. Rather than solving a single vehicle pick-up and delivery optimization problem for each candidate task and vehicle iteratively, we solve a multi-agent pick-up and delivery optimization model to allocate all tasks to vehicles at once, and achieve a significant improvement in CPU time and task *allocation rate*.

We consider the infrastructure and resources at Amsterdam Schiphol Airport to generate problem instances for aircraft ground handling. We convert the task allocation problem into multi-vehicle pick-up and delivery problem with time windows and multiple objectives and introduce warm start MILP and MINLP models to solve the problem. Then, we obtain conflict-free routes using path planning, considering the MINLP solutions as input, since MINLP minimizes the occupation of service locations at the stands.

Besides the research in Chen et al. (2023), our solution approach, which combines the global multi-vehicle task allocation and scheduling optimization with multi-agent path finding, is a novel method for coordination of autonomous ground handling vehicles using scheduling and conflict-free routing. The majority of the literature focuses on task scheduling only and does not incorporate the conflict-free path planning of ground support equipment into this solution. Avoiding collisions is strictly required for safety in autonomous systems. The proposed MILP outperforms the solution approach of Chen et al. (2023) by 48% reduction in CPU time in the higher frequency scenario, and 100% success in allocating tasks. Also, we improve the performance indicators of path planning, compared to Chen et al. (2023), due to the introduction of parking locations at the stands and waiting times at the depots in the proposed MINLP and path planning models.

We summarize our main contributions and some differences in modeling compared to the research in Chen et al. (2023) as follows:

(i) We use a solution approach, $insertion\_heuristic + MILP\ (MINLP) + P.SIPP$, where the $MILP$ or $MINLP$ solver takes the output of the $insertion\_heuristic$ as initial feasible solution, and initiates the branch and bound search from this point, to solve a multi-vehicle pick-up and delivery problem for each GSE type, and then the prioritized safe interval path planning algorithm ($P.SIPP$) is applied to resolve conflicts, regarding the target locations, which were defined by solving the MILP or MINLP models. For each GSE type, there are several vehicles over which we allocate tasks. Solving the $insertion\_heuristic + MILP\ (MINLP)$, tasks are allocated to multiple vehicles of the same type. However, Chen et al. (2023) uses a decentralized solution approach, $TESSI + MILP + P.SIPP$, where $MILP$ is a single vehicle pick up and delivery optimization solver that is iteratively used as a sub-procedure of a temporal sequential single-item ($TESSI$) auction, where all agents reschedule their partial schedule using $MILP$, based on the prospect of including the announced task, to generate a bid.

(ii) Using the proposed MILP and MINLP models, we are able to allocate all tasks to GSE vehicles, unlike the decentralized approach in Chen et al. (2023) where some of the tasks remain unallocated after the auction process ends.

(iii) We introduce waiting times at the depots and parking locations at the stands for some vehicles, in the proposed models. This reduces the time GSE vehicles park around service locations while they are in the queue and decreases the obstacles and conflicts to deal with on the aircraft stands.

(iv) We improve deviations of conflict-free path length and duration from shortest ones, delays of tasks and GSE vehicles related to detours or waiting for conflict resolution, and relevant indicators in path planning, compared to Chen et al. (2023), due to the introduction of parking locations and waiting times at depots.

(v) We use warm start MILP and MINLP models to solve the multi-vehicle pick up and delivery representation of the ground handling problem to accelerate the search in solution space, while Chen et al. (2023) solve the MILP models of single vehicle pick up and delivery models without introducing any cuts on the solution space.

(vi) We solve different versions of MILP and MINLP models for different types of GSE vehicles, since we represent the corresponding problems with different types of multi-vehicle pick-up and delivery models based on the characteristics such as being paired, unpaired, capacitated, uncapacitated.

(vii) Unlike the state-of-the-art MILP model for the multi-vehicle pick-up and delivery problem with time windows, we solve a multi-objective version of the MILP model. Chen et al. (2023) also experiment with an additional objective but test this separately in a single objective form.

(viii) While the two single objectives in Chen et al. (2023) are *minimizing makespan* and *minimizing total travel time*, the objectives we test in multi-objective forms are *minimizing makespan*, *minimizing start times of tasks on the stands*, and *maximizing a time buffer that represents additional parking times at depot locations at the terminal after finishing tasks there*, which allow us to improve the quality of the integrated path planning solutions, which is the critical goal for safety.

We present the existing research in Section 2. A general overview of aircraft ground handling process and problem definition are given in Section 3 and Section 4, respectively. The methodology is presented in Section 5. The proposed MILP and MINLP models and the multi-vehicle pick-up and delivery model regarding the aircraft ground handling problem are explained in Section 5.1. Section 5.2 describes the prioritized safe interval path planning for automated aircraft ground handling. Section 6, Section 7, Section 8 include respectively the results, discussion, and conclusion.

2. Related studies

The aircraft ground handling includes the airside activities at airports around the parked aircraft, regarding passengers, cargo, facilities, and supplies. Most of the operations are performed by different service providers, using specialized vehicles and equipment known as ground support equipment (GSE) Bye and Schaathun (2014).

IATA has reported an increase over 7% in global air traffic between 2014 and 2019 International Air Transport Association (IATA) (2019) and expected a growth in 2024 and afterwards International Air Transport Association (IATA) (2022). Automation of ground handling processes plays an important role in improving efficiency, safety, and reducing carbon emissions. Automation of tasks requires strategic allocation and scheduling of tasks given a limited size of heterogeneous GSE fleet, as well as determining conflict-free routes for GSE vehicles traveling on aircraft stands.

Delays on ground handling tasks lead to aircraft delays and congestion. Sheibani (2020) and Schmidt (2017) focus on estimation of turnaround time and required fleet size for each GSE type in different traffic scenarios. Appointment systems and optimization models are other approaches to deal with delays. A booking system that allows airlines to book GSE is offered by Saggar et al. (2021) and mathematical optimization models for scheduling GSE are presented in Padrón et al. (2016) and Guimarans and Padrón (2022). The new concepts of multi-processing GSE are introduced to improve congestion and unavailability of equipment on aircraft stands Schmidt et al. (2015). Recent studies focus on the transition to autonomous ground support equipment Morris et al. (2016). Tabares et al. (2021) propose heuristic methods to assign fully automated or semi-automated vehicles to ground handling tasks and illustrate the impact of automated operations at an airport.

Wang et al. (2021) propose solutions for accurate tracking, collision detection, and optimal scheduling of GSE which includes vehicles with one carriage, such as tractors and shutters, as well as the baggage transit trains that contain one tug plus multiple dollies. For scheduling, they present a mixed-integer linear programming model that aims to minimize the total rental cost and travel time of GSE and solve the problem using a heuristic. The problem constraints include flight timetables, speed limits, size of GSE fleet, maximum number of dollies that can be attached to baggage transit trains.

Assignment of GSE fleet to a heterogeneous set of ground handling tasks and generating task sequences that minimize both the turnaround time between consecutive flights and the makespan for all vehicles becomes a challenging problem when the available GSE fleet is limited, and the flights are frequent.

In a recent work Chen et al. (2023), a framework that combines task allocation and path planning for automation of ground handling operations is presented. Using a multi-agent perspective, the task allocation problem is handled by an integrated solver that combines an auction algorithm with a mixed integer programming model, which is used to generate bids at each round of the auction. For each candidate task, possible assignment to a potential position of the existing schedule of each GSE vehicle is evaluated by re-optimizing the schedule of the vehicle including the candidate. The candidate is assigned to a vehicle if only that vehicle is the winner of the auction in that round. In this way, partial schedules are created by taking the objectives of different agents into account at each decision phase until all tasks are allocated. To develop the mixed integer programming model, the task scheduling problem for a single vehicle is converted into a single vehicle pick-up and delivery problem with time windows, which considers also the movements of GSE vehicles on the paths in addition to processing times of tasks. The considered GSE fleet includes refueling, catering, baggage handling, water and lavatory service vehicles. A path planning algorithm for aircraft ground handling is presented to avoid collisions on the aircraft stands. Volt et al. (2022) develop a mathematical model for determining the number of airport equipment dedicated for baggage

Fig. 1. Monitoring aircraft stand, Smart VOP 1.0 Schiphol Innovation (2020).

loading and unloading. The demand for the carts and loaders is predicted and based on the prediction, the optimal number of equipment that can handle all flights is obtained. Sheibani (2020) design a model for scheduling ground handling operations with uncertain durations which might be due to breakdowns, weather conditions, cargo loading and unloading incidents. Critical Path Analysis and Monte Carlo Simulation are used to improve the aircraft ground handling operations during the turnaround. Bao et al. (2023) study the GSE scheduling problem with mixed fleet of fuel vehicles and electric vehicles with time windows and the objective of minimizing the sum of time, energy, and emission costs and propose an optimal fleet configuration model. However, the considered GSE fleet consists of aircraft towing vehicles rather than the GSE used for the ground handling operations at aircraft stands. Norin et al. (2012) focus on scheduling de-icing vehicles. The objective is to minimize the delay of flights due to de-icing, and the travel distance of the de-icing vehicles. They propose a greedy randomized adaptive search algorithm. A case study of real-life data from Stockholm Arlanda Airport shows that proposed method performs considerably better compared to simple scheduling strategies.

The technology required for autonomous vehicles is widely available and affordable Tabares and Mora-Camino (2017b). Amsterdam Schiphol Airport has recently initiated the automation process by testing automated baggage handling vehicles. In addition to this, a data-driven monitoring system, Smart VOP 1.0 Fig. 1, is used to measure the performance on aircraft stand Schiphol Innovation (2020). Tabares and Mora-Camino (2017b) outline the adaptations required to achieve full automation of GSE and ground handling processes, and the challenges of transformation. Adaptations include designing targets to guide autonomous vehicles in finding the docking locations, robotizing aircraft systems such as opening of doors and panels, using automatic connection systems and automated cargo loading systems, adaptation of traffic rules and processes for autonomous vehicles. The challenges include strict rules for certification.

Optimal allocation of tasks to agents is required to improve system performance and minimize resource utilization and operational costs. The problem exists in various fields such as robotics, distributed computing, transportation logistics, and workforce management Gombolay et al. (2018), Contini and Farinelli (2021). Allocation of ground handling tasks to GSE vehicles requires scheduling a set of tasks on a heterogeneous fleet of vehicles while respecting specific time windows to prevent delays.

Gerkey and Matarić (2004) classifies the multi-robot task allocation problems based on robot type, task type and allocation type, which are respectively the problems with single- and multi-task robots,

single- and multi-robot tasks, and instantaneous and time-extended assignments. Single-task robots perform one task at a time while multi-task robots perform multiple tasks simultaneously. Multi-robot tasks require multiple robots to be processed, while single-robot tasks can be completed by a single robot. The information regarding current and future tasks is known in advance in time-extended assignments, unlike instantaneous assignments. Korsah et al. (2013) provides a more complex classification which includes precedence relations and inter-dependencies. Khamis et al. (2015) classify the solution methods for solving task allocation problems based on the level of *control*. In a centralized approach, a central entity allocates tasks to the agents, considering the constraints and requirements for all tasks and agents. In a decentralized solution method, decisions are made by agents based on local information. The centralized approach is useful when there are complex dependencies between the tasks. A decentralized approach can be more flexible and adaptable, but may suffer from the lack of global information and coordination, leading to suboptimal task allocations.

Centralized task allocation approaches include mainly the optimization based solution methods such as Hungarian method Kuhn (2010), branch-and-bound algorithm, integer linear programming, and constraint programming Ponda (2012). Optimization based solution approaches are computationally expensive. The metaheuristics that used for solving multi-agent task allocation include genetic algorithms, simulated annealing, and particle swarm optimization Mirjalili (2019), Zhang et al. (2022). In our model, the agents need to move to the locations of allocated tasks, and we define the problem of allocating ground handling tasks to multiple agents as vehicle routing problem with time windows, pick-ups, deliveries, and multiple objectives. We also model paired, unpaired, capacitated, and uncapacitated variations, based on task and GSE features.

Zhang et al. (2025) present a comprehensive study of the use of autonomous electric dollies in baggage transport and service, and formulate the problem as split-demand multi-trip electric vehicle routing problem, including the constraints such as time windows, baggage release time, and battery capacity. Zhang et al. (2025) solve the problem using an *improved large neighborhood search* and use the performance of CPLEX for solving the proposed mathematical model as benchmark. The CPLEX is able to prove optimality on the instance sets up to seven gates, however the *improved large neighborhood search* outperforms CPLEX on the instance set with larger number of gates. In our problem instances, each gate handles around four or five aircraft in a time window of four hours due to the probability distribution of turnaround times, while in Zhang et al. (2025) each gate handles four aircraft in a time window of 12 h. Also, in our problem set-up, we do not include battery charging or more specific details such as baggage release times. However, we consider multiple types of vehicles, unlike Zhang et al. (2025).

In a recent work, Zhou et al. (2023) model the ground handling problem as vehicle routing problem with heterogeneous fleet and capacity, time windows, and precedence constraints. Our model differs from Zhou et al. (2023) by the inclusion of pick-ups, deliveries, related variations and multiple objectives. To solve the problem, Zhou et al. (2023) use a metaheuristic, *large neighborhood search*, where they incorporate a neural network based learning procedure into *destroy* procedure in neighborhood search to destroy a part of the *anchor solution* and use the MILP for the defined heterogeneous vehicle routing problem as an iterative subprocedure for reoptimizing the destroyed part of the solutions to find neighbor solutions. The proposed learning based method in Zhou et al. (2023) outperforms the state of the art CPLEX solvers in medium and large scale instances, while for the small scale instance set, CPLEX is able to find solution with a smaller optimality gap in 5 min.

Wu et al. (2023) present another neural network based solution approach for ground handling. They split the vehicle routing representation of the ground handling problem into subproblems and solve the subproblems using learning based construction heuristics where they use deep reinforcement learning to learn construction policies. They compare their method with metaheuristics and construction heuristics. Likewise Zhou et al. (2023), Wu et al. (2023) test the instance sets with 20, 50, 100, 200 flights. Wu et al. (2023) also extend the experiments to 300 flights for generalization and show that the learning based method performs well in large scale instances.

In both Zhou et al. (2023) and Wu et al. (2023), mainly, the arrivals for 20, 50, 100 flights are sampled over time, and based on the arrival events, the aircraft are randomly assigned over 91 gates at three terminals. However, in our case, we generate the aircraft arrivals in a four-hours time window, all of which are handled in a more restricted area that covers three gates, which increases the task load of the limited number of GSE vehicles in a short time and GSE traffic on the stands. This adds a safety layer to our problem. Therefore, in addition to task scheduling using a vehicle routing model, we also consider the collisions of GSE vehicles in a restricted area when task load is high in a short time window and solve an integrated path planning problem.

To model the autonomous GSE vehicle task allocation as vehicle routing problem with time windows and pick-ups and deliveries, we define time windows Sol and Savelsbergh (1995), Dohn et al. (2011), precedence Bredström and Rönnqvist (2008), and capacity Kumar and Panneerselvam (2012) constraints. We also define multiple objectives, which converts the problem into multi-objective VRP with time windows and pick-ups and deliveries. A multi-objective model was proposed by Wang et al. (2016) for solving the VRP with simultaneous delivery and pick-up and time windows, where five weighted objectives were defined as number of vehicles, total travel distance, makespan, total waiting time and, total delay. Capacitated multi-vehicle pick-up and delivery problem with precedence and time windows constraints is NP-hard. Heuristic Bianchessi and Righini (2007) or meta-heuristic Liu et al. (2013), Bianchessi and Righini (2007), Mirjalili (2019) methods are used to solve complex vehicle routing problems efficiently. For scheduling ground handling tasks, Padrón et al. (2016), Padrón and Guimarans (2019) find the initial solution for the vehicle routing problem with time windows using an insertion heuristic and apply a local search to improve the solution. In the MILP and MINLP solvers that we use to solve the automated aircraft ground handling problem, we use insertion heuristics to generate cuts on the solution space to accelerate the search.

We use a time limit of 30 s for the MILP solver and 60 s for the MINLP solver, for each group of GSE set. The MILP solver proves the optimality for refueling and catering instances in 0.038 and 10.723 s (*medians*), respectively, and provides the best solutions obtained within the time limit for the other GSE types. Similarly, the MINLP solver reaches the time limit for all GSE types. We use the solutions obtained by the MILP and MINLP solvers as input for path planning.

3. Aircraft ground handling process

We present a general overview of aircraft ground handling activities, process flow, required equipment. Ground handling services are grouped as cabin and ramp activities. Cabin activities are the services that are performed inside the aircraft and the services which occur outside such as refueling and cargo handling belong to the ramp activities. The ground handling process starts when the aircraft approaches the aircraft stand Aviation Learnings Team (2020). The aircraft is marshaled into the docking location by airport personnel or using a digital docking system. The aircraft is positioned along the indicated markings to maintain separation from other aircraft. When the aircraft is parked, engines are shut off, beacon lights are turned off and the parking brakes are installed. After this, the ramp and cabin handling operations start. The ground power unit, the pre-conditioned air unit and the aircraft access such as a passenger boarding bridge or mobile stairs are connected, and passenger deplaning and cargo unloading are performed. These operations are followed by cabin and ramp activities such as cleaning, refueling, cargo loading, passenger boarding which prepares the aircraft to depart again. Aircraft air start unit starts one

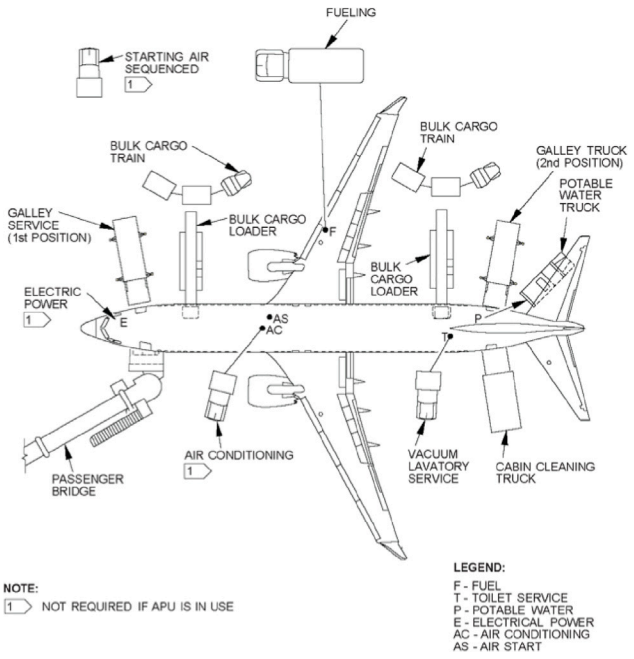

Fig. 2. Service Locations of GSE for Boeing 737-800 Boeing Commercial Airplanes (2022).

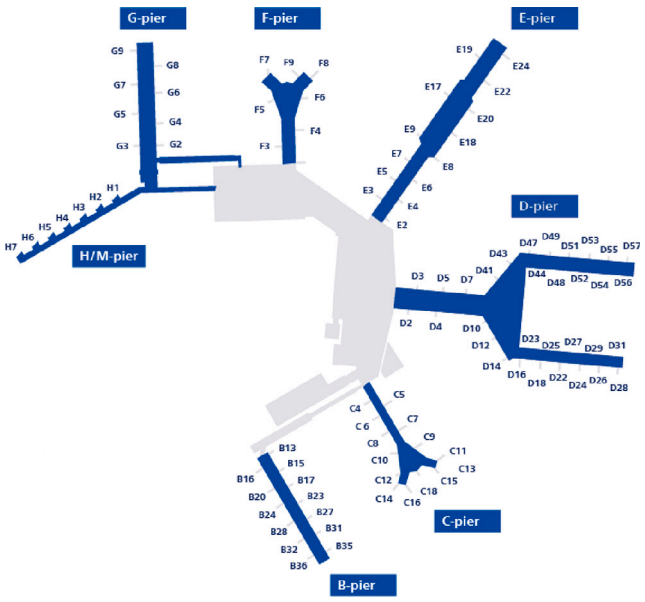

Fig. 3. Pier layout with gate numbers Schiphol Newsroom (2020).

engine of the aircraft and the pilot gives permission to disconnect the ground power unit, pre-conditioned air unit and parking brakes, and then it is possible to initiate push-back. Some tasks are not performed in some cases. Refueling is not always required for short-haul flights. The choice of emptying the waste tank also depends on the available infrastructure, the flight schedule, and the status of the tanks in the aircraft. The ground handling activities are performed by the ground support equipment (GSE). Each GSE type serves at a specific location around the aircraft. Fig. 2 depicts the service layout of Boeing 737-800 with nine types of GSE. When the auxiliary power unit of the aircraft is used, the ground power unit, pre-conditioned air unit and air start unit are not required. The galley truck in Fig. 2 collects the food trolleys from the aircraft and replaces them with new food trolleys. A bulk cargo train and a bulk cargo loader are used for baggage handling. The bulk cargo train (the baggage handling truck) consists of a tow tractor and baggage carts. When the bulk cargo train arrives at the service location on the ramp, the bulk cargo loader, which is a mobile conveyor belt, moves the cargo from the train to the cargo hold of the aircraft. The baggage is placed and secured by personnel in the cargo hold. For the long range aircraft, unit load devices are used. These devices are loaded with cargo, driven to the aircraft, and placed into the cargo hold. For refueling, the fuel hydrant dispenser or a refueling truck is used, depending on the infrastructure. The fuel hydrant dispenser, which is a small truck that connects the fuel line to the fuel tanks of the aircraft, is used if an underground network of fuel lines exists at the airport. Otherwise, the refueling truck is used. The portable water truck contains a water tank and connects to a valve outside the aircraft to refill the water tank of the aircraft. Cabin cleaning truck delivers cleaning supplies into the cabin. The lavatory service truck carrying an empty tank is connected to the aircraft to empty the waste tank.

4. Problem definition

The problem is to solve the multi-robot task scheduling problem for allocating ground handling operations over GSE vehicles (robots) and finding the collision-free trajectories for the robots. We consider the optimal allocation and reallocation of different types of ground handling operations over a heterogeneous set of GSE vehicles (robots) and conflict-free path planning for the GSE vehicles at Amsterdam Schiphol Airport, while respecting the time windows constraints regarding the flight schedules. In the problem, ground handling tasks are to be handled on several aircraft stands while the aircraft are parked at these stands between different flights. GSE fleet can operate at different stands. That is, vehicles travel between different task locations not only on a single stand but also on various stands. Automation of GSE requires finding conflict-free paths to prevent collisions on these stands. The aim is to improve the turnaround times of ground handling operations at different stands and to find the conflict-free routes of automated vehicles to travel between different task locations. We take one pier that includes several aircraft stands in the layout of Amsterdam Schiphol Airport as reference. The ground handling problem is restricted to the north side of pier B at Amsterdam Schiphol Airport, since we analyze the case where GSE vehicles are shared and reused in a condensed area. In this area, autonomous GSE vehicles serve to several aircraft and collisions might occur due to the high density of static and dynamic obstacles. Studying this area was recommended by airport domain experts.

In the following subsections, we present the problem details for the layout, operations, equipment, and other critical information related to simulation environment and problem design and constraints.

4.1. Layout

We solve the ground handling problem for the north side of pier B at Amsterdam Schiphol Airport. Amsterdam Schiphol Airport has 7 piers with 91 connected or semi-connected gates. The layout for 7 piers with gate numbers is shown in Fig. 3. Pier B is a linear pier with six or seven aircraft stands on each side, for short-haul flights. Fig. 4 presents the aerial view of Pier B. The aircraft stands at the gates are accessed by GSE vehicles using the perimeter road around the pier.

Each stand on the north side has a passenger boarding bridge (PBB) through which all passengers board the aircraft. Each of these stands is equipped with an underground fueling system. Therefore, only hydrant vehicles are necessary to refuel an aircraft. Pier B consists of the stands that are connected to a single perimeter road. GSE vehicles use this road to move between the stands and to or from the terminal. Each stand has an entrance point to the stand and exit point to the perimeter road.

Fig. 5 shows the infrastructure of an aircraft stand at pier B north side. The area 8 in Fig. 5 includes the entrance and exit locations of GSE

Fig. 4. Aerial view of pier B Chui (2020).

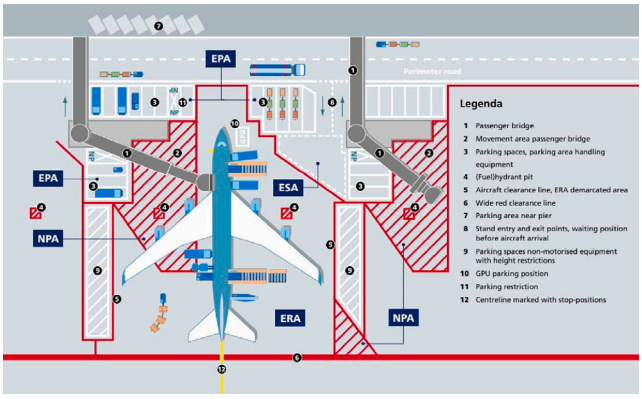

Fig. 5. Layout of an aircraft stand at pier B at north side HSE Office (2022b).

vehicles. The waiting, parking, maneuvering areas for the GSE vehicles are respectively the equipment staging area (ESA), equipment parking area (EPA) and equipment restraint area (ERA) in Fig. 5. Passengers walk to the mobile stairs Allaart and Kaijen (2022) or board the aircraft using the passenger boarding bridge (PBB) which is depicted as area 1 in Fig. 5).

GSE vehicles are parked at the EPA (areas 3, 7, 9 in Fig. 5) when they are not used at the platform. Speed limits which are 30 km/h on the perimeter road and 15 km/h for the platform and baggage area HSE Office (2022a) are applied for safety.

4.2. Ground handling operations

Ground handling operations are tasks that must be completed on the apron, within a turnaround process between the arrival and departure of an aircraft. Ground handling process can be different for short- and long- haul flights. The short-haul flights have tighter schedules to complete ground handling operations. We consider the ground handling operations of Boeing 737-800 for short-haul flights. Ground handling operations are also classified as over the wing and under the wing operations. We consider under the wing operations which include *refueling* the aircraft, *water service* by filling the water tank, *lavatory service* by emptying the waste tank, *baggage handling* by loading and unloading the baggage, *catering* by swapping the catering galleys. We also further split the tasks into the tasks that are handled at the front and aft of the aircraft or define sub-tasks such as pick-up and delivery tasks that are required to be completed to handle the main task. Furthermore, we deal with baggage handling tasks by defining distinct types of baggage handling tasks such as loading and unloading baggage. The explicit definitions of ground handling tasks we consider to solve the ground handling problem and the pick-up and delivery sub-tasks that we define to convert the problem into a multi-vehicle pick-up and delivery optimization model are presented in Section 5.1. We exclude the seasonal activities such as de-icing. We also exclude the cargo loading and unloading, assuming that short-haul flights are not used for freight transportation. Thus, in baggage handling, we consider the loading and unloading of passenger luggage, in which case the baggage tractor with carts can handle the task. We consider the ground handling tasks outside the aircraft cabin except for connecting and disconnecting the passenger boarding bridge. The passenger boarding bridges (PBB) have reserved areas at each aircraft stand, thus we do not consider the task allocation and path planning of PBB agents. The tasks inside the cabin include safety and security checks, cleaning, passenger boarding and unboarding. The tasks outside the cabin are handled mostly at standardized locations by the ground handling vehicles around the aircraft, which allows using autonomous vehicles for these operations and automating these processes. We focus on the safety and efficiency of autonomous vehicles around the aircraft.

4.3. Ground handling equipment

Each type of ground handling operation requires using a specific ground service equipment. We consider mainly five types of GSE vehicles, *refueling*, *catering*, *baggage handling*, *water service*, *lavatory service* vehicles, based on the ground handling operation types we aim to plan. We also define further sub-classes such as *catering front*, *catering aft*, or *baggage loading*, *baggage unloading*. Further details about all defined types of GSE, their priority and capacities are presented in Section 5.1. Vehicles have limited capacities except for the *refueling* vehicle which is assumed to have an unlimited capacity since an underground fueling system is used. Priorities are defined based on how critical handling of a task is i.e. fueling the aircraft is one of the most critical tasks. These priorities are also used as inputs to resolve conflicts of GSE vehicles in the path planning solution method we suggest. The capacities of the GSE vehicles are defined based on discussions with experts and considering also the literature. For baggage handling, two tractors are required per aircraft, one for loading and one for unloading the baggage. Each tractor can link up to five carts fully loaded with baggage, which is sufficient for carrying all passengers' baggage from or to an aircraft. Two catering trucks are able to serve three aircraft, therefore, the capacity of one catering vehicle is limited by the amount of galleys that can meet the requirements of 1.5 aircraft. In the problem we define, all GSE types are shared between the stands of the same pier, however each GSE type can only serve the specific types of tasks that are compatible with the GSE type. The available number of GSE vehicles of the same type is allowed to be larger than one.

4.4. Turnaround time

The time required to complete the scheduled ground handling activities between two flights is referred to as *turnaround time* (TAT). It is equal to the time the aircraft remains on the aircraft stand while the ground operations are being handled by the GSE. The time to complete each type of ground handling task and the turnaround time are estimated by the aircraft manufacturers. Boeing Commercial Airplanes (2022) presents the estimated TAT schedule for Boeing 737-800. In this schedule, the ground handling services which are on the critical path define the *turnaround time*. However, the *turnaround time* is also affected by the airline, airport, layout and available resources. Each airline has its own strategy and airlines contract different ground handling task providers for different type of tasks at each airport in their network. Because, ground handling providers are usually not specialized on all task types.

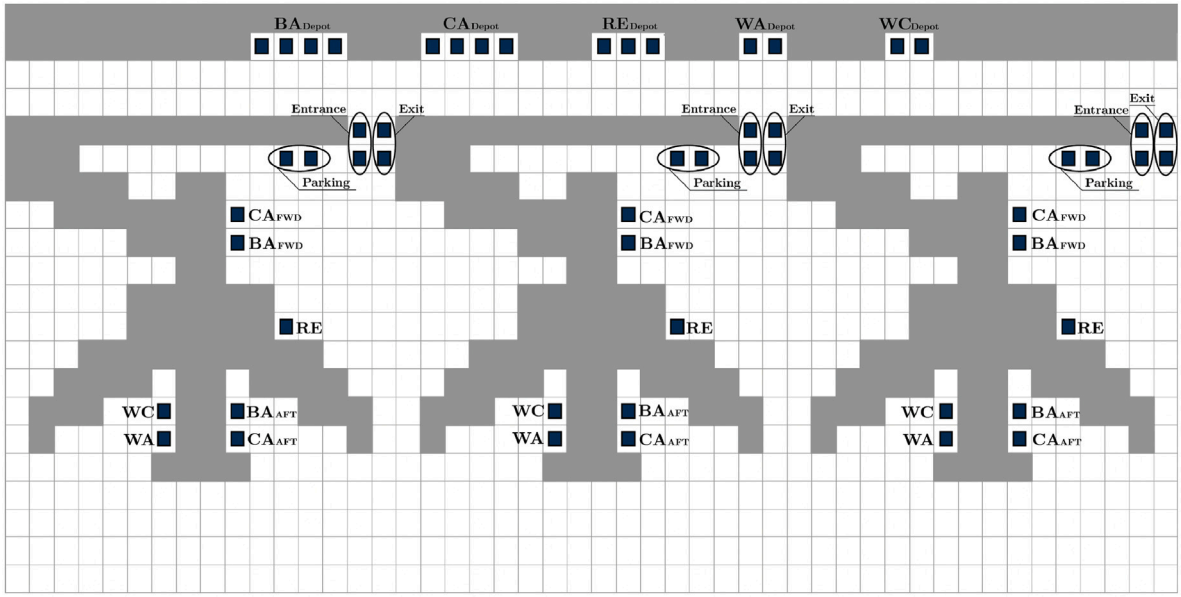

Fig. 6. Simulation environment.

4.5. Time windows

Considering the *turnaround time* schedule and lower and higher flight frequency scenarios in a four-hour planning window, we define time windows constraints for ground handling tasks. At each aircraft stand, the time interval between two flights is an *upper bound* for finishing all tasks for an aircraft, which depends on the flight schedules within the four hours planning windows. To set the duration and time windows for different types of tasks, we refer to the turnaround time chart of aircraft B737-800 provided in technical documentation of Boeing Boeing Commercial Airplanes (2022). The execution duration of ground handling tasks should lie in the specified time windows. The time chart is a guideline for various turnaround operations, however the real execution time of the ground handling tasks is affected by different factors. Further details on flight schedules and time windows constraints for ground handling tasks are given in Section 6.2.

4.6. Planning window and flight schedules

We design a model where the planning for ground handling operations is done for the flights occurring within four hours and involves the flights scheduled to use the three aircraft stands in pier B. In this way, the plans can be updated to respond to peak, normal, or non-peak hours of the day. We use uniform distribution to set the time intervals between flights in accordance with busy and non-busy traffic hours of the day and generate the scenarios for frequent and less frequent flights in the flight schedules for four hours. The details on generating higher and lower frequency flight scenarios are presented in Section 6.2.

4.7. Simulation environment

We take the pier B at Amsterdam Schiphol Airport as reference, to set the configuration, the number of aircraft stands, infrastructure and operating flights. The simulation environment is a grid-based modeling of the terminal parking area, perimeter road, and three aircraft stands of the pier. Three aircraft stands are tiled to each other. Each aircraft stand is represented by the 16×16 grid-map. Each cell of the grid-map has the size of 4 m $\times$ 4 m. There exist static obstacles such as the passenger boarding bridge and the parked aircraft, and they constitute 15%–20% of the environment. The service locations occupy single cells around the aircraft. The complete environment including the terminal, the service road, and three aircraft stands on the pier is shown in Figure Fig. 6.

4.8. Depot and service locations of GSE vehicles

The source locations (depots) of GSE vehicles at the terminal are randomly selected cells connected to the service road on the grid-maps. The service locations of GSE vehicles at aircraft stands are specific standard positions around the aircraft. Each GSE vehicle type has to serve the aircraft at a certain location, which is dissimilar to the service locations of other types of GSE vehicles. We have defined these locations considering the service locations for the aircraft type, Boeing 737.

4.9. Movements of vehicles

Examples of movements of GSE vehicles on simulation environment from depot locations at the terminal to service locations at the aircraft stands are given in Fig. 7. The model adopts a simplified movement scheme for the vehicles, where they navigate exclusively in four cardinal directions: north, south, east, and west. In this approach, rotational movements, acceleration, and deceleration of vehicles are not considered. Directional constraints exist for the service road to ensure that GSE vehicles are driving on the right side of the road. Each GSE vehicle occupies single cell. The speed limit of vehicles on an aircraft stand is 15 km/h, therefore the speed of the vehicles is assumed to be one cell per second which is equivalent to 14 km/h.

5. Methodology

We explain our solution approach for task allocation and routing in Section 5.1 and path planning in Section 5.2. We also present a replanning method in Section 6.5.

5.1. Task allocation

We consider the *catering*, *refueling*, *water service*, *lavatory service*, and *baggage handling* operations and the GSE vehicles, which are namely *catering vehicle* (CA), *refueling vehicle* (RE), *water service vehicle* (WA), *lavatory service vehicle* (WA), and *baggage handling vehicle* (BA) for solving the automated ground handling problem.

Ground handling operations consist of multiple sub-tasks. For instance, to complete the *catering* operation, catering galleys are required

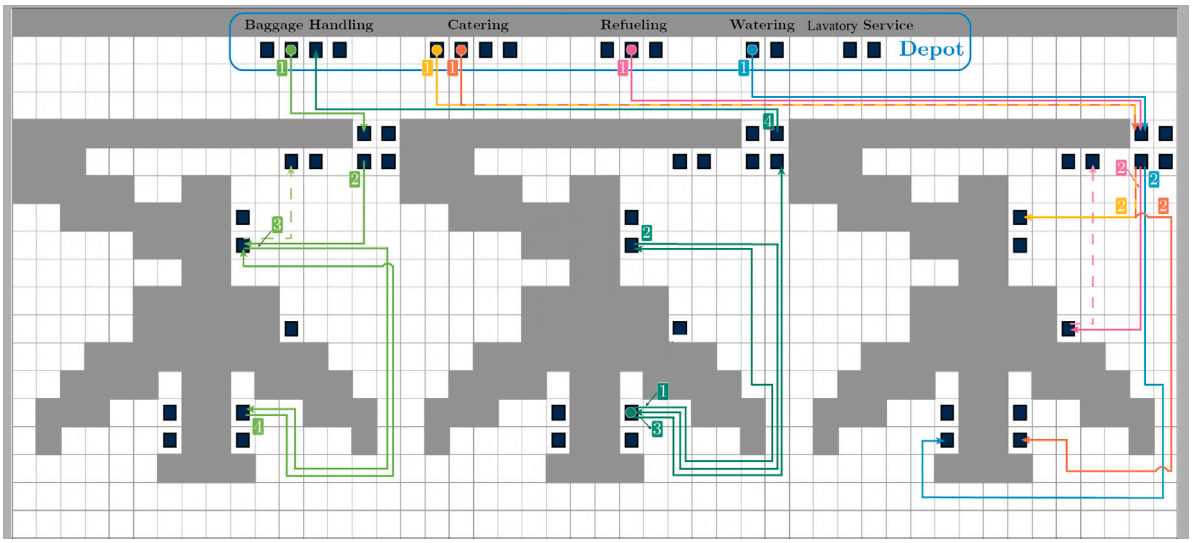

Fig. 7. Movements of GSE vehicles on simulation environment, from depot locations at the terminal to service locations at the stands.

Table 1
Pick-up sub-tasks for each type of ground handling task.

Task	Sub-task (pick-up)	Location
baggage loading	*load* baggage to GSE	*terminal*
baggage unloading	*unload* baggage from aircraft (*load* baggage to GSE)	*bay*
catering	*load* catering galleys to GSE	*terminal*
water service	Fill up water tank of GSE	*terminal*
lavatory service	Empty the waste tank of the aircraft (*load* the tank of GSE)	*bay*
refueling	N.A.	–

Table 2
Delivery sub-tasks for each type of ground handling task.

Task	Sub-task (delivery)	Location
baggage loading	*load* baggage to aircraft (*unload* baggage from GSE)	*bay*
baggage unloading	*unload* baggage from GSE	*terminal*
catering	*load* catering galleys to aircraft (*unload* galleys from GSE)	*bay*
water service	*load* water to aircraft tank (*unload* water from GSE tank)	*bay*
lavatory service	*unload* waste from GSE tank	*terminal*
refueling	*fuel* the aircraft	bay

to be loaded to the catering truck, transported to the aircraft stands and loaded to the aircraft.

We split the ground handling tasks into pick-up and delivery tasks with various pick-up and delivery locations on the pier, and present an overview of how we split each task type into sub-tasks in Tables 1 and 2.

In addition to splitting tasks into pick-up and delivery sub-tasks in Tables 1 and 2, the baggage handling task is also divided into loading and unloading tasks. Thus, four sub-tasks (two pick-up and two delivery) are defined for baggage handling task, while two sub-tasks (one pick-up and one delivery) are created for the other task types. The pick-up sub-task of *refueling* operation does not exist in Table 1 since the fueling is done using an underground fueling system at Amsterdam Schiphol Airport, thus, there is no pick-up sub-task at the terminal. However, the delivery sub-task is defined as fueling the aircraft at the bay in Table 2.

Table 3
Capacities of GSE vehicles.

GSE type	GSE code	Capacity
Baggage loading truck	*BA load*	1
Baggage unloading truck	*BA unload*	1
Catering truck	*CA*	1.5
Water service truck	*WA*	20
Lavatory service truck	*WC*	25
Refueling truck	*RE*	∞

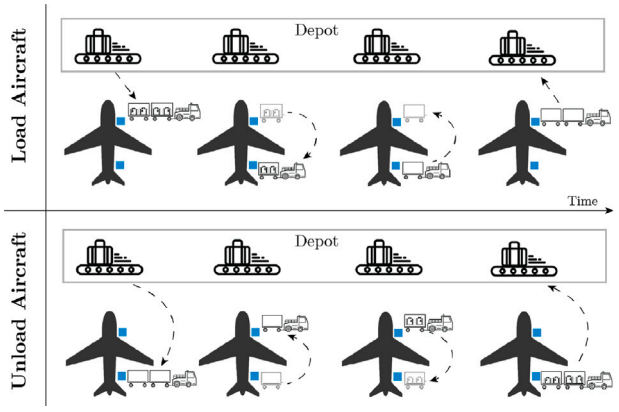

Fig. 8. Baggage loading and unloading operations.

Vehicle types, codes and capacities of the GSE vehicles are given in Table 3. All vehicles, except for the *refueling truck*, have limited capacities. Refueling is handled using an underground fueling system at Amsterdam Schiphol Airport, therefore we set the capacity of *refueling truck* to infinity.

Baggage tractors deliver carts to both the front and aft service locations. Fig. 8 depicts the baggage tractor moving between front and aft service locations of the aircraft during loading and unloading baggage.

Splitting ground handling into pick-up and delivery sub-tasks, the problem converts into a scheduling problem with pick-ups and deliveries. Furthermore, there is a heterogeneous group of tasks, where each homogeneous set within this group can only be served by a dedicated type of vehicle. A limited size of heterogeneous GSE fleet is available to serve the tasks at various service locations at several bays, while

also traveling among the bays and the terminal to visit the pick-up and delivery locations by respecting the precedence relations of the defined sub-tasks. Capacities of the GSE vehicles are restricted by the amount of tasks they can perform with a full load, and they have to revisit their pick-up locations at the terminal to reload required items. Also, each task has a time windows constraint. Thus, the problem becomes a complex version of vehicle routing problem with time windows, pick-ups and deliveries, and heterogeneous fleet and tasks, where the distances between various pick-up and delivery locations at the pier are defined by Manhattan distances.

In Section 5.1.1, we present the mathematical models we use to solve the complex vehicle routing problem for centralized allocation of tasks to GSE vehicles in automated ground handling problem.

5.1.1. Mathematical model

We model the multi-robot ground handling task scheduling problem as multi-vehicle paired pick-up and delivery problem with time windows (PDPTW) Parragh et al. (2008). The problem is modeled on a complete graph, $G = (V, A)$, where V is the set of all vertices and A the set of all arcs. $G = (V, A)$ also includes virtual vertices representing start and end nodes.

Parameters. The parameters of the model are as follows:

n: number of pick-up (delivery) vertices

P: set of pick-up vertices $P = \{1, \dots, n\}$

D: set of delivery vertices $D = \{n+1, \dots, n+n\}$

V: set of vertices, $V = \{0, 2n+1\} \cup P \cup D\} = \{0, \dots, 2n+1\}$, where the vertices 0 and $2n+1$ are the virtual start and end nodes, respectively.

A: set of all arcs $A = \{(i, j) : i, j \in V, i \neq 2n+1, j \neq 0, i \neq j\}$

V_{depot}: set of vertices corresponding to depots

V_{stand}: set of vertices representing aircraft stands

K: set of vehicles

q_i: demand/supply at vertex i, which is a positive (negative) value if it is a pick-up (delivery) vertex i.e. $q_i > 0$, $\forall i \in P$ and $q_i < 0$, $\forall i \in D$, or 0 otherwise, i.e. $q_i = 0\ \forall i \in V \setminus (P \cup D)$.

e_i: earliest time to begin service at vertex i.

l_i latest time to begin service at vertex i

d_i service duration at vertex i

t_{ij}: travel time from vertex i to vertex j

C^k: capacity of vehicle $k \in K$

GSE vehicles enter and exit aircraft stands via designated locations. We build the graph $G(V, E)$ on a grid-map. The travel distance from vertex i to vertex j is computed using the Manhattan distance on the grid-map, considering also the entrance or exit locations when they are required to be visited between these vertices. Travel speed is assumed to be unit and the same for all vehicles. Due to this assumption, the travel time is equivalent to travel distance. However, the travel times of the GSE vehicles in real life might vary due to the uncertain environmental conditions. For this reason, we use a distance buffer coefficient as proposed in Chen et al. (2023) to consider the uncertainties regarding travel times.

Decision variables. The decision variables of the model are presented as follows:

x_{ij}^k: binary decision variable which takes 1 if vehicle $k \in K$ travels from vertex $i \in V$ to vertex $j \in V$, or 0 otherwise

Q_i^k: the load of vehicle $k \in K$ after leaving vertex $i \in V$

B_i^k: the time vehicle $k \in K$ starts serving at vertex $i \in V$

Objective function. We use multiple objectives combined in different forms. We define three objective functions, z_1, z_2, and z_3. The objective function, z_1, which is shown in Eq. (1), is the total of makespans of all GSE vehicles. We define the makespan of each vehicle as the time difference between the start time at the virtual end node, $i = 2n+1$, and the start time at the virtual origin, $i = 0$. The aim is to minimize the sum of makespans of all GSE vehicles. The objective function, z_2, is shown in Eq. (2), and defined as the sum of start times of all tasks at aircraft stands. The aim is to minimize z_2. The motivation is to initiate the service tasks of the GSE vehicles as soon as possible so that the tasks at service locations are completed earlier. The objective function, z_3, is defined to reduce the time service locations are occupied by GSE vehicles at the stands.

$$z_1 = \sum_{k \in K} B_{2n+1}^k - B_0^k \tag{1}$$

$$z_2 = \sum_{k \in K} \sum_{i \in V_{stand}} B_i^k \tag{2}$$

The objective function, z_3, that is denoted in Eq. (3), is a nonlinear objective function and the goal is to maximize z_3 which encourages waiting at depot (terminal) after the pick up task is finished, instead of visiting the service locations immediately after completing the pick-up task. This waiting time is equal to the time difference between *the start time of task at the stand minus traveling time from depot to stand* ($B_j^k - t_{ij}$) and *the start time of task at the depot plus the duration of task at depot* ($B_i^k + d_i$), if $x_{ij}^k = 1$, $i \in V_{depot}$, and $j \in V_{stand}$. That is, in Eq. (3), we maximize a time buffer $\omega = (B_j^k - t_{ij}) - (B_i^k + d_i)$ for the links, $x_{ij}^k = 1$, $i \in V_{depot}$, $j \in V_{stand}$. Maximization of the sum of all ω buffers in z_3 has impact on the behavior of the constraint in Eq. (8). While the objective z_2 encourages all start time values at the stands take the minimum values given the constraint in Eq. (8), the objective z_3 leads to the increases in the buffer ω when a stand node is visited immediately after a depot. Thus, rather than initiating the task at the depot (a pick-up location at the terminal) as late as possible to minimize the waiting time at the stand, the solver tends to select the depot–stand links where it is more likely to have a buffer after finishing task at the depot before starting traveling.

$$z_3 = \sum_{k \in K} \sum_{i \in V_{depot}} \left(\sum_{j \in V_{stand}} (B_j^k - t_{ij} - d_i) \cdot x_{ij}^k - \sum_{v \in V} B_i^k \cdot x_{vi}^k \right) \tag{3}$$

Note that, by the term *waiting time* we refer to the time buffer, ω, which we add between the completion time of a task at the depot and the start time of its traveling, rather than the classical definition of a waiting time which is usually used to describe the time a vehicle waits before starting service after it arrives at the service location in scheduling literature. That is, we are holding a GSE vehicle at its origin. This is also similar to the gate holding procedures defined by the Federal Aviation Administration in the US Federal Aviation Administration (2025).

In classical network models, if arrival time to the next destination node is not explicitly defined as a decision variable, the *arrival time* of a vehicle to the destination is assumed to be *completion time at the origin plus traveling time*, which would be defined as $B_i^k + d_i + t_{ij}$ if $x_{ij}^k = 1$, $\forall\ k \in K$, $\forall\ i \in V$, $\forall\ j \in V$. However, in our model, we interpret *arrival time* as *completion time at the origin plus the time ω plus traveling time*.

This can be denoted as $B_i^k + d_i + \omega + t_{ij}$ if $x_{ij}^k = 1$, $k \in K$, $i \in V_{depot}$, $j \in V_{stand}$, in which case waiting time at the stand is actually equal to B_j^k - ($B_i^k + d_i + \omega + t_{ij}$) rather than B_j^k - ($B_i^k + d_i + t_{ij}$). Since the *arrival time* is not strictly defined as a variable in the mathematical model, we set the time a vehicle leaves its depot as $B_i^k + d_i + \omega$, if $x_{ij}^k = 1$, $k \in K$, $i \in V_{depot}$, $j \in V_{stand}$ while setting GSE schedules as input for the integrated path planning solver and while reporting the relevant performance indicators in the results.

In experiments, we use a combination of z_1 and z_2 as a baseline and only include the third objective z_3 in a single experiment to measure improvement in compliance with ground handling regulations. Therefore, the mathematical formulations of the two versions of the objective function are $z = z_1 + z_2$ and $z = z_1 + z_2 - z_3$, respectively. The aim is to minimize z in both versions.

The objective, z_3, is defined to improve compliance with regulations on ground handling. Regulations state that vehicles should minimize their time on an aircraft stand. Vehicles are therefore rewarded for waiting at a parking location instead of the service location on a stand. This is achieved by negating the maximization objective, z_3, when added to other objectives in the minimization function, z.

The non-linear objective function, z_3, converts the existing MILP model into a Mixed-Integer Non-Linear Programming (MINLP) model. The linearization of the non-linear objective function is automatically handled by the solver, GUROBI. Even though GUROBI is not a non-linear programming solver, it can deal with simple non-linear constraints or objectives using its features for quadratic programming.

Constraints. The problem is subject to the following constraints:

$$\sum_{k\in K}\sum_{j:(i,j)\in A} x_{ij}^k = 1, \quad \forall i \in P \cup D \tag{4}$$

$$\sum_{j:(0,j)\in A, j\in V\setminus\{2n+1\}} x_{0j}^k = 1, \quad \forall k \in K \tag{5}$$

$$\sum_{i:(i,2n+1)\in A, i\in V\setminus\{0\}} x_{i,2n+1}^k = 1, \quad \forall k \in K \tag{6}$$

$$\sum_{i:(i,j)\in A} x_{ij}^k - \sum_{i:(j,i)\in A} x_{ji}^k = 0, \quad \forall j \in P \cup D, k \in K \tag{7}$$

$$x_{ij}^k = 1 \rightarrow B_j^k \geq B_i^k + d_i + t_{ij}, \quad \forall (i,j) \in A, k \in K \tag{8}$$

$$x_{ij}^k = 1 \rightarrow Q_j^k = Q_i^k + q_i, \quad \forall (i,j) \in A, k \in K \tag{9}$$

$$\max\{0, q_i\} \leq Q_i^k \leq \min\{C^k, C^k + q_i\}, \quad \forall i \in V, k \in K \tag{10}$$

$$\sum_{j:(i,j)\in A} x_{ij}^k - \sum_{j:(n+i,j)\in A} x_{n+i,j}^k = 0, \quad \forall i \in P, k \in K \tag{11}$$

$$B_i^k \leq B_{n+i}^k, \quad \forall i \in P, k \in K \tag{12}$$

$$e_i \leq B_i^k \leq l_i, \quad \forall i \in V, k \in K \tag{13}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A, k \in K \tag{14}$$

The constraints (4) state that every vertex must be served exactly once. The constraints (5) and (6) ensure that each vehicle starts at the origin and ends at the sink node, $2n+1$. The route from the origin 0 to the sink node $2n+1$ is forbidden in the constraints (5) and (6), to ensure that each GSE vehicle is used. The original versions of these constraints which were presented in Parragh et al. (2008) allow using the route between the nodes 0 and 2n+1 which means that the number of used vehicles is allowed to be smaller than the number of available vehicles. The constraints (7) ensure the continuity of the flow between vertices. The constraints (8) are start time constraints and allow omitting the sub-tour elimination constraints, provided that $t_{ij} + d_i > 0$ for all $(i,j) \in A$. We linearize the constraints (8) as $(1-x_{ij}^k)\cdot M + B_j^k \geq B_i^k + d_i + t_{ij}^k$ using the big M method. The constraints (9) are the load constraints. Parragh et al. (2008) states that the constraints (9) can be relaxed as $Q_j^k \geq Q_i^k + q_i$ if the problem is a paired pick-up and delivery problem. Since we model a paired pick-up and delivery problem, we relax and linearize these constraints as follows: $(1 - x_{ij}^k)\cdot M + Q_j^k \geq Q_i^k + q_i$. The constraints (10) ensure that the load of any vehicle does not exceed its capacity. The constraints (11) ensure that the paired pick-up and delivery tasks are served by the same vehicle, and the constraints (12) require that the start time of a delivery task is larger than or equal to the start time of the pick-up task when they are paired. The constraints (13) are the time windows constraints. The domains of the decision variables $x_{i,j}$ are set in (14).

Table 4
Adaptations of the mathematical model for each GSE type.

GSE type	Time windows	Capacitated	Paired
BA (bag. handl.)	✓	✓	✓
CA (catering)	✓	✓	✓
WA (water ser.)	✓	✓	
WC (lavatory s.)	✓	✓	
RE (refueling)	✓		

There exists a heterogeneous fleet of GSE vehicles in autonomous ground handling problem, and there are several vehicles for each type of vehicle. We split the heterogeneous fleet into homogeneous sets of vehicles and solve the multi vehicle pick-up and delivery problem for each homogeneous fleet of vehicles. Each homogeneous fleet is used to handle a specific type of ground handling task at various stands, which is different from the task types other homogeneous vehicle sets can handle.

We solve a modified version of the multi-vehicle pick-up and delivery model, depending on the requirements of each fleet type. Since the catering galleys and baggage of passengers have to be delivered to the correct aircraft, the pick-up and delivery tasks are paired. Thus, for catering and baggage handling vehicles, we solve the paired pick-up and delivery problem using the constraints (4),…,(14). Water and lavatory service trucks include large tanks that are used to fill or empty the aircraft tanks. Although there is a capacity limit, there is no restriction on which aircraft to serve. Since the tasks are unpaired, we solve the unpaired pick-up and delivery problem for these vehicle types. We remove the constraints (11) and (12) and replace the vertex set V as $V = \{0, \dots, n+\tilde{n}+1\}$, where $P = \{1, \dots, n\}$, $D = \{n+1, \dots, n+\tilde{n}\}$ and the end node is represented by the node $n+\tilde{n}+1$. The end node $n+\tilde{n}+1$ also replaces the node $2n+1$ in the constraints (5) and (6). Amsterdam Schiphol Airport features an underground fueling system, thus fuel hydrant vehicles without capacity constraints are used for refueling. For solving the problem for refueling vehicles, we also remove the capacity and load constraints, (9) and (10). Table 4 shows an overview of the adaptations for each type of GSE vehicles, which are the baggage tractor (BA), catering truck (CA), portable water truck (WA), vacuum lavatory truck (WC), hydrant vehicle (refueling truck) (RE).

Fig. 9 depicts the representation of example tours for each GSE vehicle type, based on the structure of the problem, defined in Table 4.

Lavatory service truck is omitted in Fig. 9 since both water and lavatory service trucks traverse unpaired pick-up and delivery tours. The tour instances for baggage handling and catering vehicles are paired pick-up and delivery tours. For baggage handling and catering, the tasks in Fig. 9 are classified as *baggage loading*, *baggage unloading*, *catering aft* and *catering forward*. All vehicle types, except for the refueling truck are subject to the capacity restrictions and precedence constraints depending on existence of paired or unpaired pick-up and delivery tasks. Thus, the tours of these vehicles in Fig. 9 that they visit the depot several times before completing their tours. The tour of a refueling truck is not restricted by capacity or precedence constraints, therefore, no depot revisit exists in Fig. 9 for refueling truck.

For all GSE types the tour generation is also subject to time windows constraints. Multiple tours for each type of GSE vehicle exist in the

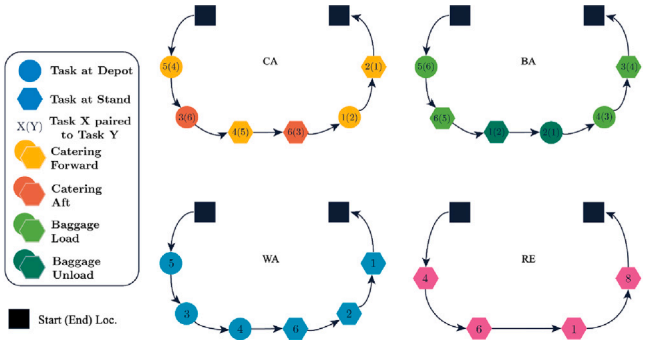

Fig. 9. Example tours for each type of GSE.

solution since several vehicles might be available for each type of vehicle. The complete solution is a combination of tours of heterogeneous vehicles, where each homogeneous vehicle subset in this heterogeneous group follow the tours obtained by solving a different type of vehicle routing problem than the one used for another homogeneous subset.

5.1.2. Warm start solution

We incorporate an insertion heuristic, which generates the initial feasible solution and cuts on the solution space, into the mixed-integer programming (MIP) model defined in Section 5.1.1. MIP solver initiates the search starting from the warm start solution provided by the insertion heuristic. We adapt the Solomon's insertion heuristic Solomon (1987) for finding the initial feasible solution for the pick-up and delivery model of the ground handling problem. Solomon (1987) propose insertion heuristics for solving the vehicle routing problems with time windows constraints, where the benefit of adding a set of tasks to an existing route compared to another route, the distance, and the urgency of completing the tasks are evaluated in various approaches. Solomon (1987) present the insertion cost of any task u that is to be placed between task i and j in Eq. (15),

$$c(i,u,j) = \alpha_1 \cdot c_1(i,u,j) + \alpha_2 \cdot c_2(i,u,j) + \alpha_3 \cdot c_3(i,u,j) \tag{15}$$

where $c_1(i,u,j) = d_{iu} + d_{uj} - d_{ij}$, $c_2(i,u,j) = b_{j_u} - b_j$, and $c_3(i,u,j) = l_u - b_u$. c_1 evaluates the change in distance when u is inserted between i and j, c_2 evaluates the change in start times, where b_{j_u} is the new start time of the task j in the event that u is on the route and b_j is the start time of the task j when u is not on the route, and c_3 evaluates the impact on urgency of the services. α_1, α_2, and α_3 are the weights assigned to the criteria, c_1, c_2, and c_3, where $\alpha_1 + \alpha_2 + \alpha_3 = 1$, $\alpha_1 \geq 0$, $\alpha_2 \geq 0$, and $\alpha_3 \geq 0$.

We modify the Solomon's heuristic as follows: Solving the ground handling problem using the temporal criteria is more important, thus, we increase the weights of c_2 and c_3, compared to c_1. We evaluate the insertion of pick-up and delivery tasks into existing routes, while Solomon's heuristic evaluates the insertion of independent tasks. Solomon's heuristic solves the problem where the number of existing vehicles is unlimited, and the vehicle fleet is homogeneous. We run the insertion heuristic separately for each type of GSE vehicle, and consider that there exists a finite set of homogeneous vehicles for each type of GSE vehicle. The modified insertion heuristic is presented in Algorithm 1.

Algorithm 1 takes the information related to GSE fleet characteristics, flight schedules and the turnaround time schedule (TAT) as input and evaluates the cost of inserting pick-up and delivery tasks, iteratively. The routes of GSE vehicles as well as the schedules of tasks that will be completed while traveling on these routes are produced as output.

Input : The set of GSE vehicles, flight schedules, turnaround time schedule of tasks
Output: Schedules for GSE vehicles and tasks

```
solution = ∅
insertion_cost = ∞
best_insertion_cost = ∞
while unassigned_tasks do
    P, D ← pick-up and delivery task with earliest deadline
    for k ∈ K do
        foreach pos_P in route of vehicle k do
            P.insertion_cost(pos_P) =c(i, P, j)
            foreach pos_D following pos_P in route vehicle k do
                D.Insertion_cost(pos_D) =c(i, D, j)
                PD.insertion_cost =P.insertion_cost + D.insertion_cost
                if PD.insertion_cost < insertion_cost then
                    insertion_cost = PD.insertion_cost
                end
            end
        end
        if Insertion_cost < Best_insertion_cost then
            best_insertion_cost = Insertion_cost
            best_vehicle = k
        end
    end
    if best_insertion_cost ≠ ∞ then
        unassigned tasks.remove(P, D)
        solution[best_vehicle].insert(P, D)
        insertion_cost = ∞
        best_insertion_cost = ∞
    end
end
```

Algorithm 1: Insertion heuristic for ground handling

5.2. Path planning

For automation of ground handling operations, we also solve the problem of collision-free path planning of GSE vehicles considering the allocation of tasks to GSE vehicles generated by solving the vehicle routing models proposed in Section 5.1. The routing based allocation models in Section 5.1 find the routes of GSE vehicles without considering the potential collisions of GSE vehicles. Taking the task allocation and routing plan obtained by the models in Section 5.1 as input, we solve a path planning problem to find the conflict-free paths of GSE vehicles on the grid-based map environment of the pier that we focus on to solve the ground handling problem. We use the path planning algorithm that is presented by Chen et al. (2023) for planning the paths of GSE vehicles on the apron area that is designed as a grid-based environment. This algorithm is based on prioritized safe interval path planning, where prioritized planning considering the priorities of GSE vehicles is used as the high-level solver and safe interval path planning (SIPP) is used as the low-level solver.

SIPP is a single agent path finding algorithm Phillips and Likhachev (2011) that considers other agents in the environment as dynamic obstacles. Dynamic obstacles in the environment change their position from one time step to the next, creating many constraints based on discrete time steps. To reduce the complexity, instead of discrete timesteps, safe time intervals are used. A safe interval is a continuous period of time during which it is safe for an agent to occupy the grid. In prioritized SIPP, the paths of higher priority vehicles are stored and the safe intervals for lower priority agents are modified accordingly. A reservation table for storing the information related to higher priority vehicles is proposed by Ma et al. (2019).

For solving the path planning problem, we split the grid-map of the complete environment into two segments: (i) the environment for the terminal plus the service road and (ii) the environment for the aircraft stands. The GSE vehicles have to visit the entrance and exit locations between the service road and the aircraft stands while

entering into and leaving the aircraft stands. The entrance and exit locations are the intermediate stops and we treat them as goals or origins in split segments and solve the path planning problem for each segment, separately. We present the path planning algorithm we use for automation of ground handling operations in Algorithm 2.

Input : The set of GSE vehicles and their schedules, maps of service road and aircraft stands, static obstacles, safe intervals
Output: The path, P_a, for each agent $a \in A$

foreach $a \in A$ **do**
  foreach *segment* $\in$ *schedule of agent* a **do**
    if *segment* $\in map_{road}$ **then**
      $path$ =SIPP($segment_{start}$, $segment_{goal}$, map_{road}, $safe\_interval_{road}$)
      P_a.append(path)
      safe_interval$_{road}$ = update_safe_interval($path$, map_{road}, $safe\_interval_{road}$)
    end
    else if *segment* $\in map_{stand}$ **then**
      idx = number of aircraft stand
      $path$ =SIPP($segment_{start}$, $segment_{goal}$, map_{stand}, $safe\_interval_{stand}[idx]$)
      P_a.append(path)
      safe_interval$_{stand}[idx]$ = update_safe_interval($path$, map_{stand}, $safe\_interval_{stand}[idx]$)
    end
  end
end

Algorithm 2: Prioritized SIPP for ground handling

In Algorithm 2, we take the set of agents, $A = a_1, a_2, \ldots, a_n$, the schedules generated by the proposed task allocation model in Section 5.1, the grid-based maps for the service road and the aircraft stands, static obstacles and the safe intervals as input. The order in which the paths of the agents are planned is based on the priorities of the GSE vehicles. Once the paths are planned, safe intervals are updated to ensure the generation of conflict-free paths. The output is the set of conflict-free paths of agents. In this algorithm, the solution procedure for prioritized SIPP is modified for planning the paths of GSE vehicles on the apron area. The potential moves and successor states are restricted by the directional constraints in our environment, which is not the case in the original SIPP. We refer the reader to Chen et al. (2023) for further details of prioritized safe interval path planning of GSE vehicles on grid-based pier maps.

5.3. Replanning

The ground handling process is prone to delays due to the variety of stakeholders, including passengers, airlines, and GSE vehicle providers. The turnaround time schedule is tightly planned. The delay of one task on the GSE schedule potentially cause delays on other tasks. Thus, the ability to reassign tasks to different vehicles is crucial to prevent or reduce the total delay. To deal with delays, we simulate disruptions that trigger the replanning process.

We apply replanning in two conditions: (i) several tasks are delayed due to a disruption, (ii) the total deviation from the plan is substantial. When a disruption occurs and these conditions are met, all subsequent tasks are replanned. An adapted version of the insertion heuristic presented in Section 5.1.2 is used for replanning. In this adaptation, the procedure to find insertion cost is modified. The pick-up sub-tasks that have already been completed before the disruption remain as planned. A relaxation is applied on the time windows that allows to find feasible solutions for reallocation of remaining tasks, and the delayed tasks are penalized in the adapted insertion cost, which is presented in Eq. (16),

$$c^*(i,u,j) = c(i,u,j) + t_{delay} * \rho \tag{16}$$

where $c(i,u,j)$ is the insertion cost of a task after disruption.

An overview of the replanning process is given in Fig. 10.

Furthermore, path planning algorithm is solved, using the replanned schedule as input, to obtain the updated paths after disruption. The performance of replanning is tested in Section 6.5.

6. Results

Experiments were performed on an 8-core Apple M1 Pro chip with 16 GB RAM. For solving the mathematical model of the PDPTW related to the task allocation model, we used GUROBI 10.0.1 as the MILP solver. For path planning algorithms we used Python.

First, to test the scalability and solution quality of the proposed MILP model, we applied preliminary tests on large scale benchmark instances of the PDPTW for which best known solutions were published. We present the preliminary tests in Section 6.1 regarding the large scale instances and verification of the quality of KPIs on small instance sets for MILP and path planning models. Second, we generate real life instances considering the airport infrastructure, flight schedules and ground handling procedures at Pier B of Amsterdam Schiphol Airport and perform experiments on these instances. We explain the features of real life instances and design of experiments in Section 6.2. We present the results of experiments for the proposed MILP and MINLP models on real life instances in Section 6.3. Later, we provide the results for path planning and replanning experiments in Section 6.4 and Section 6.5 and sensitivity analysis in Section 6.6.

6.1. Preliminary tests

We present the large-scale benchmark instances in Section 6.1.1, preliminary tests on large scale benchmark instances using the proposed warm start MILP model in Section 6.1.2, and verification of the quality of KPIs on small instances of MILP and path planning models in Section 6.1.3.

6.1.1. Large-scale benchmark instances

Li and Lim (2001) present the best known heuristic solutions and CPU times for large-scale instances of the pick-up and delivery problem with time windows (PDPTW) and related instance sets. To test the performance of the MILP models we use to solve the ground handling problem on large instances, we use the instance sets of the Li and Lim (2001) as benchmark. To avoid high computational time in experiments, we stop MILP solvers before reaching optimality and compare the solution quality with the best known solutions of the corresponding instances in benchmark sets. We present further details on verification and validation using these benchmark instances in Section 6.1.2.

6.1.2. Solving the proposed MILP on large scale instances

The benchmark instances of Li and Lim (2001) include the best known solutions for the instance sizes of 100, 200, 400, 600, 800, and 1000 tasks, approximately, for vehicle routing problem with time windows, pick-ups and deliveries. In these instances, each task is either a pick-up or a delivery and has a paired task which is respectively a delivery or a pick-up task. The objective is minimizing the total distance as well as minimizing the number of vehicles.

We perform tests on the *instance sets*[2] of Li and Lim (2001) for the problem sizes of 100 tasks for verification of the MILP models we use to solve the pick-up delivery model of the ground handling problem. In the problem instances of Li and Lim (2001), the number of used vehicles is also a variable to be minimized, although it is a fixed parameter in our models. To be able to compare the solution quality of objective functions, we make an adjustment on our objectives. By modifying the constraints (5) and (6), we allow using the arc that emanates from the origin and goes directly to the sink node, which means that some of the available vehicles are allowed to remain unused.

[2] https://www.sintef.no/projectweb/top/pdptw/100-customers/

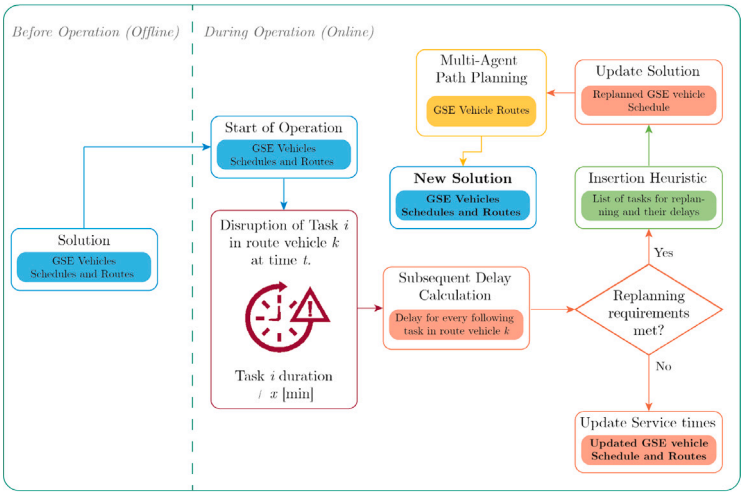

Fig. 10. Flowchart of replanning process.

The new constraints are given in Eqs. (17) and (18).

$$\sum_{j:(0,j)\in A, j\in V} x^k_{0j} = 1, \quad \forall k \in K \tag{17}$$

$$\sum_{i:(i,2n+1)\in A, i\in V} x^k_{i,2n+1} = 1, \quad \forall k \in K \tag{18}$$

Due to Eqs. (17) and (18), if a vehicle $k \in K$ is assigned to the arc, $(0, 2n + 1)$, in a paired pick-up and delivery model, it shows that the vehicle does not actually leave the depot.

We perform the verification experiments over a set of 100 randomly selected benchmark instance sets using the proposed warm start MILP model. Compared to the benchmark solutions, we obtain identical *number of used vehicles* and the deviation from the values of *total distance* is negligible (< 0.1). This shows that even though we limit the solution time of the MILP solver by short time bound of 30 s, the solution quality of the best known results of the state-of-the art metaheuristics is maintained.

6.1.3. Verification of KPIs on small instance sizes

In addition to the general objectives that are verified over the large-scale benchmark instance sets, we also verify the quality of other KPIs that we define for ground handling operations in Section 5.1.1, using a small input data set. For verification of path planning algorithm, we compare the routing results of the proposed MILP models which do not attempt to avoid collisions but aim only to find shorter routes, with the collision-free solutions of the path planning algorithm, and verify the quality of the results on small datasets.

6.2. Experiments on real life instances for Pier B

The real life instance sizes for ground handling operations are smaller than the large-scale benchmark instances that were used for verification in preliminary tests in Section 6.1. We performed the experiments taking the pier infrastructure, flight schedules, time windows of tasks based on the *turnaround time* (TAT) schedule, and the characteristics of the GSE vehicles as input. We explain the features of real life instances and the experimental design in Sections 6.2.1 and 6.2.2

6.2.1. Instance features

Flight schedules are generated for a planning window of four hours. The higher and lower frequency of flights are simulated using respectively the uniform distributions $U \sim (45, 55)$ and $U \sim (50, 60)$ of the time intervals between flights, in terms of minutes, while the planning time window remains constant. The TAT schedule is built based on the *turnaround time* schedule of Boeing Boeing Commercial Airplanes (2022) and the information received from industrial experts. The duration and time windows of the tasks are identical for all flights. Table 5 shows an example for time windows in seconds, for the start times of different task types processed at the forward or aft of the aircraft. The time windows are set with respect to the arrival of the aircraft at the stand. That is, the bounds for the time windows are determined by the times that pass after the aircraft parks at the stand.

Table 5
Time windows for start times of tasks (front-aft) after aircraft arrivals.

Task type	Location	Time windows (s)
Refueling	–	[300, 420]
Catering	(fwd, aft)	[360, 480], [360, 480]
Bag. loading	(fwd, aft)	[1260, 1380], [1260, 1380]
Bag. unloading	(fwd, aft)	[240, 260], [60, 180]
Water service	–	[60, 2100]
Lavatory service	–	[600, 1320]

Table 6
Priority order of GSE vehicles.

GSE type	Priority order
Refueling	1
Catering	2
Baggage handling	3
Water service	4
Lavatory service	5

The number of available GSE vehicles for each GSE type to serve three aircraft stands in a four-hour planning window is defined based on the discussions with industry experts. Accordingly, the number of refueling, catering, baggage handling, water service and lavatory service trucks are defined respectively as 3, 4, 4, 2, 2, and the capacities of these trucks in terms of the carriage units they hold are set as ∞, 1.5, 1, 20, 25, respectively. The priority order of each GSE type is given in Table 6. The lower the priority value in Table 6, the higher the priority of the vehicle is.

The simulation environment is the 16×16-grid map of three aircraft stands, service road and the exits to terminal at pier B. As different than the 16×16-grid map of pier B, which was presented in Chen et al. (2023), we also modeled parking locations at the stands for refueling and baggage handling vehicles.

6.2.2. Experimental design

To test higher and lower traffic demands, we generate the scenarios, *higher_freq* and *lower_freq* for frequent and less frequent flights in

Table 7
Indicators for allocation and routing and statistical significance tests.

KPI	Description, $\bar{M}_i$, $\bar{s}_i$ - *minutes*, $\bar{t}_i$ - *seconds*
$\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$	*mean makespan* for RE, CA, BA, WA, WC vehicles, respectively.
$\bar{M}^1_{tot}$	*mean* over all GSE types (*mean* of $\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$ for single instance)
$\bar{M}^2_{tot}$	*mean* of *medians* of $\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$ in large scale experiments
$\bar{s}_{RE}$, $\bar{s}_{CA}$, $\bar{s}_{BA}$ $\bar{s}_{WA}$, $\bar{s}_{WC}$	*mean start time* for RE, CA, BA, CA, WC services at bays, respectively.
$\bar{s}^1_{tot}$	*mean start time* over all service types (*mean* of $\bar{s}_{RE}$, $\bar{s}_{CA}$, $\bar{s}_{BA}$, $\bar{s}_{WA}$, $\bar{s}_{WC}$)
$\bar{s}^2_{tot}$	*mean* of *medians* of $\bar{s}_{RE}$, $\bar{s}_{CA}$, $\bar{s}_{BA}$, $\bar{s}_{WA}$, $\bar{s}_{WC}$ in large scale exp.
t_{RE}, t_{CA}, t_{BA} t_{WA}, t_{WC}	*CPU time* for routing RE, CA, BA, WA, WC vehicles, respectively.
$\bar{t}^1_{tot}$	*mean CPU time* over all GSE types (*mean* of t_{RE}, t_{CA}, t_{BA}, t_{WA}, t_{WC})
$\bar{t}^2_{tot}$	*mean* of *medians* of t_{RE}, t_{CA}, t_{BA}, t_{WA}, t_{WC} in large scale exp.
$W_{\bar{M}_{RE}}$, $W_{\bar{M}_{CA}}$, $W_{\bar{M}_{BA}}$, $W_{\bar{M}_{WA}}$, $W_{\bar{M}_{WC}}$	Wilcoxon signed rank test p-value for $\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$, respectively.
$A_{\bar{M}_{RE}}$, $A_{\bar{M}_{CA}}$, $A_{\bar{M}_{BA}}$, $A_{\bar{M}_{WA}}$, $A_{\bar{M}_{WC}}$	Vargha–Delaney A-value for $\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$, respectively.
A_{tot}	Vargha–Delaney A-value

four-hours planning window, respectively. In the *higher_freq* scenarios, the turnaround times for the aircraft are shorter due to the higher frequency of flights and unchanged value of the planning window. The turnaround time variable is uniformly distributed between 45 and 55 min, $U \sim (45, 55)$, in the *higher_freq* scenario, while in the *lower_freq* scenario, the turnaround time is a uniform distribution between 50 and 60 min ($U \sim (50, 60)$). Thus, in *higher_freq* and *lower_freq* scenarios, around 14 and 12 flights are handled at three aircraft stands in the pier, in a four-hours planning window.

For task allocation experiments using the proposed warm start MILP and MINLP models, and path planning experiments, we report the key performance indicators for *higher_freq* and *lower_freq* scenarios. For statistical analysis, we perform *statistical significance* tests.

For each experiment, we repeat the simulations over a set of 100 instances. For each multi-vehicle pick-up and delivery optimization test, we set the solution time limit for the MILP solvers as 30 s.

We present replanning experiments and sensitivity analysis for specific cases. For sensitivity analysis, we test the impact of changing the number of available GSE vehicles.

In Table 7, we present the key performance indicators (KPIs) that we use to measure the quality and performance of the task allocation and vehicle routing model which we solve using the proposed warm start MILP models of PDPTW for GSE routing.

The metrics $\bar{M}_{tot}$, $\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$ in Table 7 are related to makespan of the vehicles, overall and separately, while the metrics, $\bar{s}_{tot}$, $\bar{s}_{RE}$, $\bar{s}_{CA}$, $\bar{s}_{BA}$, $\bar{s}_{WA}$, $\bar{s}_{WC}$ are related to start times of services at aircraft stands that are to be performed by the GSE vehicles after they reach to these destinations. These indicators are also related to the objectives in Eqs. (1) and (2). The indicators t_{tot}, t_{RE}, t_{CA}, t_{BA}, t_{WA}, t_{WC} are related to CPU times for allocating tasks and routing GSE vehicles. Furthermore, the metrics W_{tot}, $W_{\bar{M}_{RE}}$, $W_{\bar{M}_{CA}}$, $W_{\bar{M}_{BA}}$, $W_{\bar{M}_{WA}}$, $W_{\bar{M}_{WC}}$, A_{tot}, $A_{\bar{M}_{RE}}$, $A_{\bar{M}_{CA}}$, $A_{\bar{M}_{BA}}$, $A_{\bar{M}_{WA}}$, $A_{\bar{M}_{WC}}$ are related to statistical significance tests.

We test the impact of the proposed MINLP model, which includes a nonlinear objective, by comparing the indicators in Table 8 for the proposed MINLP and MILP modes. In Table 8, the indicators $\bar{M}_{tot}$, $\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$ are related to *makespan* of the vehicles, $\bar{w}_{d\,tot}$, $\bar{w}_{d\,RE}$, $\bar{w}_{d\,CA}$, $\bar{w}_{d\,BA}$, $\bar{w}_{d\,WA}$, $\bar{w}_{d\,WC}$ are related to the waiting times at depots, and $\bar{w}_{S\,tot}$, $\bar{w}_{S\,RE}$, $\bar{w}_{S\,CA}$, $\bar{w}_{S\,BA}$, $\bar{w}_{S\,WA}$, $\bar{w}_{S\,WC}$ are related to the waiting times at the stands, in Table 8. In addition to the metrics in Table 8, we also evaluate the total $CPU\ time$, t_{tot}, and the $CPU\ time$ for each GSE type, t_{GSE}. We set the time limit for the MINLP solver as 60 s.

Table 8
Indicators for allocation and routing with nonlinear objective.

KPI	Description $\bar{M}_i$, $\bar{w}_{d_i}$, $\bar{w}_{S_i}$ – *minutes*
$\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$	*mean makespan* for RE, CA, BA, WA, WC vehicles, respectively.
$\bar{M}^1_{tot}$	*mean* of $\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$
$\bar{M}_{tot}$	*mean* of *medians* of $\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$
$\bar{w}_{d\,RE}$, $\bar{w}_{d\,CA}$ $\bar{w}_{d\,BA}$, $\bar{w}_{d\,WA}$ $\bar{w}_{d\,WC}$	*mean waiting time* at depot for RE, CA, BA, WA, WC vehicles, respectively.
$\bar{w}_{d\,tot}^1$	*mean waiting time* at depot over all GSE types (*mean* of $\bar{w}_{d\,RE}$, $\bar{w}_{d\,CA}$, $\bar{w}_{d\,BA}$, $\bar{w}_{d\,WA}$, $\bar{w}_{d\,WC}$)
$\bar{w}_{d\,tot}^2$	*mean* of *medians* of $\bar{w}_{d\,RE}$, $\bar{w}_{d\,CA}$, $\bar{w}_{d\,BA}$, $\bar{w}_{d\,WA}$, $\bar{w}_{d\,WC}$
$\bar{w}_{S\,RE}$, $\bar{w}_{S\,CA}$, $\bar{w}_{S\,BA}$, $\bar{w}_{S\,WA}$, $\bar{w}_{S\,WC}$	*mean waiting time* at bays for RE, CA, BA, WA, WC vehicles, respectively.
$\bar{w}_{S\,tot}^1$	*mean waiting time* at bays over all GSE types (*mean* of $\bar{w}_{S\,RE}$, $\bar{w}_{S\,CA}$, $\bar{w}_{S\,BA}$, $\bar{w}_{S\,WA}$, $\bar{w}_{S\,WC}$)
$\bar{w}_{S\,tot}^2$	*mean* of *medians* of $\bar{w}_{S\,RE}$, $\bar{w}_{S\,CA}$, $\bar{w}_{S\,BA}$, $\bar{w}_{S\,WA}$, $\bar{w}_{S\,WC}$

Table 9
Indicators for integrated path planning experiments.

KPI	Description $\bar{l}_{path}$ – meters, $\bar{d}_{path}$, $\bar{a}_{delay}$, $\bar{T}_{delay}$, t_{tot} – seconds
$\bar{l}_{path}$	*mean path length* over all vehicles
$\bar{d}_{path}$	*mean path duration* over all vehicles
$\bar{a}_{delay}$	*mean delay* per agent
$\bar{T}_{delay}$	*mean delay* per task
T_{ontime}	*task ontime rate*
a_{succes}	*agent success rate*
t_{tot}	*total CPU time* for all vehicles

For path planning experiments, we report the values of indicators defined in Table 9, which are related to the *length* and *duration* of the paths, *delays* for tasks and vehicles, the *rates* of being *ontime*, *success rates* and $CPU\ time$.

The performance indicator *success rate* which measures the success rate of finding conflict-free paths on various environments is used widely in existing literature for multi-agent path planning since it is not always possible to find conflict-free paths for all agents, thus the *success rate* is defined to quantify the performance in finding conflict-free paths.

For refueling truck the results are reported as the medians of the $\bar{M}_{RE}$ outputs over 100 simulation instances. Similarly, we report the medians of $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$ values. $\bar{M}_{tot}$ denotes the mean of the medians over all GSE types. For the $\bar{s}_{RE}$, $\bar{s}_{CA}$, $\bar{s}_{BA}$, $\bar{s}_{WA}$, $\bar{s}_{WC}$, $\bar{s}_{tot}$, $\bar{s}_{RE}$, $\bar{t}_{CA}$, $\bar{t}_{BA}$, $\bar{t}_{WA}$, $\bar{t}_{WC}$, $\bar{t}_{tot}$, $\bar{w}_{d\,RE}$, $\bar{w}_{d\,CA}$, $\bar{w}_{d\,BA}$, $\bar{w}_{d\,WA}$, $\bar{w}_{d\,WC}$, $\bar{w}_{d\,tot}$, $\bar{w}_{S\,RE}$, $\bar{w}_{S\,CA}$, $\bar{w}_{S\,BA}$, $\bar{w}_{S\,WA}$, $\bar{w}_{S\,WC}$, $\bar{w}_{S\,tot}$ in Tables 7 and 8 we use the same approach. For the items in Table 9, we report the means over 100 instance outputs.

6.3. Task allocation experiments

The results for the proposed warm start MILP model are presented in Table 10. The results show the medians of the simulation outputs over 100 simulations, i.e. the median of average makespans over 100 simulations for each GSE and the mean of medians over all GSE.

Table 10
The *medians* of $\tilde{M}_i$ ($\tilde{M}_{RE}$, $\tilde{M}_{CA}$, $\tilde{M}_{BA}$, $\tilde{M}_{WA}$, $\tilde{M}_{WC}$), t_i (t_{RE}, t_{CA}, t_{BA}, t_{WA}, t_{WC}), and $\bar{s}_i$ ($\bar{s}_{RE}$, $\bar{s}_{CA}$, $\bar{s}_{BA}$, $\bar{s}_{WA}$, $\bar{s}_{WC}$) values over 100 instances, and the *means* of *medians* of $\tilde{M}_i$, t_i, and $\bar{s}_i$, for the proposed warm start MILP for task allocation and routing, in *lower_freq* and *higher_freq* scenarios.

KPI	lower_freq	higher_freq	KPI	lower_freq	higher_freq	KPI	lower_freq	higher_freq	
$\tilde{M}_{RE}$	235.15	215.61	t_{RE}	0.038	0.041	$\bar{s}_{RE}$	129.89	133.00	*medians of* $\tilde{M}_{RE}$, t_{RE}, $\bar{s}_{RE}$
$\tilde{M}_{CA}$	246.03	253.03	t_{CA}	10.723	30.370	$\bar{s}_{CA}$	130.83	134.10	*medians of* $\tilde{M}_{CA}$, t_{CA}, $\bar{s}_{CA}$
$\tilde{M}_{BA}$	251.16	270.30	t_{BA}	30.282	30.381	$\bar{s}_{BA}$	135.89	139.08	*medians of* $\tilde{M}_{BA}$, t_{BA}, $\bar{s}_{BA}$
$\tilde{M}_{WA}$	134.20	136.28	t_{WA}	30.041	30.071	$\bar{s}_{WA}$	126.63	129.09	*medians of* $\tilde{M}_{WA}$, t_{WA}, $\bar{s}_{WA}$
$\tilde{M}_{WC}$	201.62	136.03	t_{WC}	30.051	30.057	$\bar{s}_{WC}$	125.83	129.00	*medians of* $\tilde{M}_{WC}$, t_{WC}, $\bar{s}_{WC}$
$\tilde{M}_{tot}$	213.62	202.65	$\bar{t}_{tot}$	20.225	24.185	$\bar{s}_{tot}$	129.82	132.85	*means over medians*

For the *mean makespan* indicators, $\tilde{M}_{RE}$, $\tilde{M}_{CA}$, $\tilde{M}_{BA}$, $\tilde{M}_{WA}$, $\tilde{M}_{WC}$, the results reported in Table 10 show the medians of the outputs over 100 simulation instances for each GSE type. For the *overall makespan* indicator, $\tilde{M}_{tot}$, the results denote the *means* of the *medians* of the simulation outputs for $\tilde{M}_{RE}$, $\tilde{M}_{CA}$, $\tilde{M}_{BA}$, $\tilde{M}_{WA}$, $\tilde{M}_{WC}$. We compare the results for the high and low frequency scenarios. Except for the *medians* of *mean makespans* ($\tilde{M}_{WC}$ and $\tilde{M}_{RE}$ values) for the lavatory service and refueling trucks, the *median* values of *mean makespans* ($\tilde{M}_{CA}$, $\tilde{M}_{BA}$, $\tilde{M}_{WA}$) have slightly increased for the GSE types in the *higher_freq* scenario compared to *lower_freq*. For the lavatory service truck, the *median* of the *mean makespans* ($\tilde{M}_{WC}$ values), has decreased by 32% and there was a small decrease in the *median* of the $\tilde{M}_{RE}$ values. As a result, the mean over medians (the value for $\tilde{M}_{tot}$) has decreased by 5%.

The solution times in Table 10 show that the *medians* of the *CPU times* over 100 simulations have slightly increased in the *higher_freq* scenario, for the *CPU time* indicators, t_{RE}, t_{CA}, t_{BA}, t_{WA}, t_{WC}, of GSE types. Consequently, the *mean* over the *medians* has increased for the *overall CPU time* indicator, $\bar{t}_{tot}$. Due to the solution time limit of 30 s, the increase in *CPU time* indicators is insignificant in *higher_freq* scenarios. For the catering vehicles, the *median* of the *CPU times* (t_{CA}) over 100 simulations show that optimal solution is obtained before the solution time limit is reached in *lower_freq* scenarios. In both *lower_freq* and *higher_freq* scenarios, the *median* of the t_{RE} values (*CPU times*) are also within the time limits, which means that optimality was proven before the *solution time limit* is reached and considerably smaller than other trucks. The nonexistence of capacity constraints for the refueling vehicles has significantly improved the computational time.

The *median* values of *CPU times* in Table 10 not only include the search time of the MILP solver but also the solution time of the insertion heuristic which was used as a warm start solution. The average *CPU times* to generate the initial solutions for all GSE vehicles, using the insertion heuristics are respectively 0.08 and 0.09 s for *lower_freq* and *higher_freq* scenarios.

The *medians* of the *mean start times* of the tasks at the bays ($\bar{s}_{RE}$, $\bar{s}_{BA}$, $\bar{s}_{CA}$, $\bar{s}_{WA}$, $\bar{s}_{WC}$) are higher in the *higher_freq* scenario, for all GSE types in Table 10. As a result of this, the *mean* over the *medians* of the *mean start times*, which is the output for $\bar{s}_{tot}$ is also higher in *higher_freq* scenario, compared to the *lower_freq* scenario.

For statistical analysis, we used statistical significance tests. To assess the impact of increasing the number of flights, we used the non-parametric Wilcoxon signed-rank test. In these tests, statistical significance results are reported as *p*-values and the effect sizes of all pairs of experiments were reported as Vargha–Delaney *A*-values. The *A*-value is the probability that the experimental values in one dataset is larger than the values in the compared dataset. The *p* value with $p < 0.05$ means that the test outcome is *statistically significant*. The *A*-value with $A > 0.71$ or $A < 0.29$ shows that the effect size is large. In Table 11 we present the obtained *p*- and the *A*-values for evaluating the significance on *CPU time* and *makespan* indicators considering the *higher_freq* and *lower_freq* scenarios.

The $W_{\tilde{M}_{RE}}$, $W_{\tilde{M}_{CA}}$, $W_{\tilde{M}_{BA}}$, $W_{\tilde{M}_{WA}}$, $W_{\tilde{M}_{WC}}$ values in Table 11 indicate the *p*-values for the GSE types, and $W_{\tilde{M}_{tot}}$ denotes the *p*-value over all vehicles. $W_{\bar{t}_{tot}}$ is the *p*-value related to the *CPU time* indicator over all vehicles. Similarly, the $A_{\tilde{M}_{RE}}$, $A_{\tilde{M}_{CA}}$, $A_{\tilde{M}_{BA}}$, $A_{\tilde{M}_{WA}}$, $A_{\tilde{M}_{WC}}$ values are the Vargha–Delaney *A*-values for different GSE types, $A_{\tilde{M}_{tot}}$ is the Vargha–Delaney *A*-value over all vehicles, and $A_{\bar{t}_{tot}}$ is the Vargha–Delaney *A*-value for the *CPU time* over all vehicles.

Table 11
Statistical significance test results for the proposed warm start MILP.

KPI	lower_freq vs. higher_freq	KPI	lower_freq vs. higher_freq
$W_{\tilde{M}_{RE}}$	0.75	$A_{\tilde{M}_{RE}}$	0.56
$W_{\tilde{M}_{CA}}$	0.63	$A_{\tilde{M}_{CA}}$	**0.75**
$W_{\tilde{M}_{BA}}$	0.38	$A_{\tilde{M}_{BA}}$	0.69
$W_{\tilde{M}_{WA}}$	1.00	$A_{\tilde{M}_{WA}}$	0.50
$W_{\tilde{M}_{WC}}$	1.00	$A_{\tilde{M}_{WC}}$	0.50
$W_{\tilde{M}_{tot}}$	0.74	$A_{\tilde{M}_{tot}}$	0.60
$W_{\bar{t}_{tot}}$	0.13	$A_{\bar{t}_{tot}}$	**0.72**

Since the *p*-value of $\tilde{M}_{tot}$ is higher than 0.05 and the *A*-value remains between 0.29 and 0.71 in Table 11, no statistically significant difference exists between lower and higher frequency datasets regarding the results for the *overall average makespan*, $\tilde{M}_{tot}$. For the *CPU time*, although the *p*-value shows no statistical significance the Vargha–Delaney *A*-value is equal to 0.72 which satisfies the criterion, $A > 0.71$, and shows that the effect size is large. This indicates that the statistical difference in *CPU time* in lower and higher frequency scenarios is relevant to the number of flights.

6.3.1. Non-linear objective function

We present the results of experiments related to the third objective, *max* z_3 of the MINLP model in Eq. (3). The objective function *max* z_3 or *min* $-z_3$ is a nonlinear objective function that aims to minimize the waiting times at the stands by encouraging the vehicles to wait at the depots rather than traveling to stand immediately after finishing the pick-up task at the depot, when necessary. In these experiments, we use the *lower_freq* scenario to compare various indicators for the main objective *min* $z = z_1 + z_2$ of the MILP model, and the extended objective *min* $z = z_1 + z_2 - z_3$ of the MINLP model.

The introduction of the third objective has also required defining new locations where vehicles can wait longer time periods. We have set the depot locations of vehicles at the terminal also as waiting locations. The vehicles visit the depot locations to load or unload the units of materials, except for the uncapacitated *refueling* vehicles. Therefore, we have defined virtual depot/waiting locations for *refueling* vehicles at the terminal.

Solving the proposed MILP and MINLP models, we computed the actual waiting times at the depots and stands for each type of GSE vehicle separately. In Table 12, the results regarding the *makespan* and *waiting times* at depots and stands are compared for the objective functions *min* $z = z_1 + z_2$ and *min* $z = z_1 + z_2 - z_3$ in which cases the proposed mathematical model takes the forms of MILP and MINLP, respectively. In Table 13 the *medians* of *CPU times* over 100 simulations are presented for each vehicle type and the *means* of the *medians* concerning all vehicle types are presented for the MILP and MINLP models.

The results in Table 12 indicate that using the MINLP model the *medians* of the *mean waiting times* at depots have significantly increased

Table 12
The *medians* of $\bar{M}_i$ ($\bar{M}_{RE}$, $\bar{M}_{CA}$, $\bar{M}_{BA}$, $\bar{M}_{WA}$, $\bar{M}_{WC}$), $u\bar{t}_{d\,i}$ ($u\bar{t}_{d\,RE}$, $u\bar{t}_{d\,CA}$, $u\bar{t}_{d\,BA}$, $u\bar{t}_{d\,WA}$, $u\bar{t}_{d\,WC}$), and $u\bar{t}_{S_i}$ ($u\bar{t}_{S\,RE}$, $u\bar{t}_{S\,CA}$, $u\bar{t}_{S\,BA}$, $u\bar{t}_{S\,WA}$, $u\bar{t}_{S\,WC}$) values over 100 instances, and the *means* of *medians* of $\bar{M}_i$, $u\bar{t}_{d\,i}$, and $u\bar{t}_{S_i}$, for the proposed warm start MINLP for task allocation and routing, compared to the proposed MILP.

KPI	MILP	MINLP	KPI	MILP	MINLP	KPI	MILP	MINLP	
$\bar{M}_{RE}$	235.15	235.24	$u\bar{t}_{d\,RE}$	31.43	114.92	$u\bar{t}_{S\,RE}$	85.62	0.00	*medians of* $\bar{M}_{RE}$, $u\bar{t}_{d\,RE}$, $u\bar{t}_{S\,RE}$
$\bar{M}_{CA}$	246.03	246.04	$u\bar{t}_{d\,CA}$	105.45	130.09	$u\bar{t}_{S\,CA}$	24.17	0.00	*medians of* $\bar{M}_{CA}$, $u\bar{t}_{d\,CA}$, $u\bar{t}_{S\,CA}$
$\bar{M}_{BA}$	251.16	252.12	$u\bar{t}_{d\,BA}$	52.68	90.575	$u\bar{t}_{S\,BA}$	43.03	4.93	*medians of* $\bar{M}_{BA}$, $u\bar{t}_{d\,BA}$, $u\bar{t}_{S\,BA}$
$\bar{M}_{WA}$	134.20	134.22	$u\bar{t}_{d\,WA}$	55.32	83.78	$u\bar{t}_{S\,WA}$	27.59	0.00	*medians of* $\bar{M}_{WA}$, $u\bar{t}_{d\,WA}$, $u\bar{t}_{S\,WA}$
$\bar{M}_{WC}$	201.62	201.65	$u\bar{t}_{d\,WC}$	98.97	151.98	$u\bar{t}_{S\,WC}$	51.52	0.00	*medians of* $\bar{M}_{WC}$, $u\bar{t}_{d\,WC}$, $u\bar{t}_{S\,WC}$
$\bar{M}_{tot}$	213.62	213.85	$u\bar{t}_{d\,tot}$	68.77	114.27	$u\bar{t}_{S\,tot}$	46.39	0.99	*means* over *medians*

Table 13
The *medians* of t_i (t_{RE}, t_{CA}, t_{BA}, t_{WA}, t_{WC} - *CPU times* in *seconds*, multi-vehicle), and the *mean* over *medians*, for the proposed MINLP, compared to the proposed MILP, in *lower_freq* scenario.

KPI	MILP	MINLP	
t_{RE}	0.038	60.048	*medians* of t_{RE} values
t_{CA}	10.723	60.325	*medians* of t_{CA} values
t_{BA}	30.282	60.307	*medians* of t_{BA} values
t_{WA}	30.041	60.049	*medians* of t_{WA} values
t_{WC}	30.051	60.048	*medians* of t_{WC} values
t_{tot}	20.225	60.155	*means* over *medians*

Table 14
The *medians* of t_i^1 (t_{RE}^1, t_{CA}^1, t_{BA}^1, t_{WA}^1, t_{WC}^1 - *CPU times* in *seconds*, single vehicle), and the *mean* over *medians*, for the proposed MILP and MINLP, compared to the *baseline*, in *lower_freq* and *higher_freq* scenarios.

KPI	*lower_freq*			*higher_freq*	
	Chen et al. (2023)	MILP	MINLP	Chen et al. (2023)	MILP
t_{RE^1}	18.38	0.013	20.016	20.29	0.014
t_{CA^1}	2.78	2.681	15.081	15.08	7.593
t_{BA^1}	0.90	7.571	15.077	1.32	7.595
t_{WA^1}	7.73	15.021	30.025	32.2	15.036
t_{WC^1}	9.10	15.026	30.024	16.7	15.029
t_{tot^1}	7.78	8.062	22.045	17.118	9.053

Table 15
The *medians* of the *allocation rates*, $a_{r_{(RE)}}$, $a_{r_{(CA)}}$, $a_{r_{(BA)}}$, $a_{r_{(WA)}}$, $a_{r_{(WC)}}$, and the *mean* over the *medians*, for the proposed MILP and MINLP, compared to the *baseline*, in *lower_freq* and *higher_freq* scenarios.

KPI	*lower_freq*			*higher_freq*	
	Chen et al. (2023)	MILP	MINLP	Chen et al. (2023)	MILP
$a_{r_{(RE)}}$	0.923	1	1	0.875	1
$a_{r_{(CA)}}$	1	1	1	1	1
$a_{r_{(BA)}}$	1	1	1	1	1
$a_{r_{(WA)}}$	1	1	1	0.875	1
$a_{r_{(WC)}}$	1	1	1	0.938	1
$a_{r_{(mean)}}$	0.985	1	1	0.938	1

for the GSE types. Consequently, the *mean* of the *medians* of *mean waiting times* at depots, over all GSE types, has also increased significantly (65%). Similarly, the *medians* of *mean waiting times* at service locations on the stands have considerably decreased for all GSE types and the *mean* over the *medians* of all GSE types has also decreased considerably (98%).

Due to the results, the MINLP model has improved the results of the MILP by encouraging the vehicles to wait at the depot locations rather than the stands which prevents occupying the service locations at the stands for too long. Using this strategy, it is also aimed that the path planning solver will perform better, since there will be less conflicts to resolve.

The reduction of waiting times at the stands and the increase in waiting times at the depots have not had a negative impact on the *makespan* related indicators, since using the MINLP model, the values of the *makespan* indicators have remained almost the same for all vehicle types and over all vehicles, compared to the MILP model. Table 12 shows that there is a slight increase, from 213.62 to 213.85, in the *mean* of the *medians* of *mean makespans* using MINLP model. However, the size of deviation is insignificant, therefore it is negligible.

The *means* of the *medians* of *CPU times* of GSE types are respectively 20.22 and 60.16 s for the MILP and MINLP models in Table 13 where the solution time limit for the MINLP model was set to 60 s. Thus, the MINLP model has reduced the GSE traffic at the stands compared to the MILP model, in reasonable time.

The *means* of the *medians* of *CPU times* of GSE types of the proposed MILP and MINLP models are compared to the *means* of the *medians* of *CPU times* of the *baseline* solution Chen et al. (2023) for the *lower_freq* scenario in Table 14. For the *higher_freq* scenario, comparisons are presented for the proposed MILP and the *baseline* Chen et al. (2023) in Table 14. The *baseline* algorithm is a hybrid negotiation and optimization approach that we refer to as *TESSI + MILP*, where the MILP model of single vehicle pick-up and delivery model is iteratively solved as a subprocedure of *TESSI* auction to reoptimize the schedules of the agents to generate bids.

To be able to make a direct comparison in Table 14, we divided the *CPU times* presented in Table 13 for the proposed MILP and MINLP models by the number of GSE vehicles for each GSE type, since in the proposed MILP and MINLP models the *CPU times* correspond to solving multi-vehicle pick-up and delivery problem for each type. Therefore, in Table 14, we denote the *CPU times* of each single vehicle of type *refueling*, *catering*, *baggage handling*, *water* and *lavatory* service as t_{RE^1}, t_{CA^1}, t_{BA^1}, t_{WA^1}, and t_{WC^1}, instead of t_{RE}, t_{CA}, t_{BA}, t_{WA}, t_{WC}, which represent the *CPU times* of scheduling multiple vehicles for each type.

The results in Table 14 show that the *means* of the *medians* of the *CPU times* t_{RE^1}, t_{CA^1}, t_{BA^1}, t_{WA^1}, t_{WC^1}, which are respectively 7.78 and 8.062 s for the *baseline* and *proposed MILP*, are similar in the *lower_freq* scenario. In the *higher_freq* scenario, the *mean* over *medians* increases from 7.78 to 17.118 s for *baseline* while it maintains its stability for the proposed MILP, increasing from 8.062 to 9.053 s. The proposed MINLP improves the quality of input for the path planning in 22.045 s per vehicle, on average.

In Table 15, we compare the *medians* of *allocation rates*, $a_{r(RE)}$, $a_{r(CA)}$, $a_{r(BA)}$, $a_{r(WA)}$, $a_{r(WC)}$ in the proposed MILP and MINLP models and the *baseline* Chen et al. (2023) solution, in the *lower_freq* and *higher_freq* scenarios. Even though we do not present the allocation rate as a performance indicator for the proposed MILP and MINLP, since all tasks were able to be allocated, we compare these with the allocation rates in the *baseline* solution. This is a significant improvement compared to the allocation rates in the *baseline* solution.

The comparison of the *medians* of *makespan* indicators in Table 16 for the *baseline* and the proposed models does not have much meaning on its own, since the datasets are randomly generated, however, when interpreted in combination with the statistical significance comparison in Table 17, it is seen that the statistical significance and the effect size are high in the *baseline* solutions between the *lower_freq* and *higher_freq* instance sets due to the statistical tests that result in $p < 0.05$ and $A < 0.29$ values, for all GSE types, while for the proposed MILP, p and A values show that there are no significant difference in *lower_freq* and *higher_freq* scenario instances. This can also be observed in the deviations in the *makespan* indicators in Table 16 in *lower_freq* and *higher_freq* scenarios.

Table 16
Comparison of makespan indicators.

KPI	*lower_freq*			*higher_freq*	
	Chen et al. (2023)	MILP	MINLP	Chen et al. (2023)	MILP
$\bar{M}_{RE}$	142.53	235.15	235.24	202.62	215.61
$\bar{M}_{CA}$	138.65	246.03	246.04	220.47	253.03
$\bar{M}_{BA}$	203.48	251.16	252.12	221.05	270.30
$\bar{M}_{WA}$	140.12	134.20	134.22	199.97	136.28
$\bar{M}_{WC}$	144.25	201.62	201.65	222.88	136.03
$\bar{M}_{tot}$	153.81	213.62	213.85	213.40	202.65

Table 17
Statistical significance test results compared to *baseline*.

	lower_freq vs. *higher_freq*			*lower_freq* *higher_freq*	
	9	MILP		9	MILP
$W_{\bar{M}_{RE}}$	3.88e−18	0.75	$A_{\bar{M}_{RE}}$	0.00	0.56
$W_{\bar{M}_{CA}}$	3.88e−18	0.63	$A_{\bar{M}_{CA}}$	0.00	**0.75**
$W_{\bar{M}_{BA}}$	3.88e−18	0.38	$A_{\bar{M}_{BA}}$	0.00	0.69
$W_{\bar{M}_{WA}}$	3.88e−18	1.00	$A_{\bar{M}_{WA}}$	0.00	0.50
$W_{\bar{M}_{WC}}$	3.85e−18	1.00	$A_{\bar{M}_{WC}}$	0.00	0.50
$W_{\bar{M}_{tot}}$	3.87e−18	0.74	$A_{\bar{M}_{tot}}$	0.00	0.60
$W_{t_{tot}}$	–	0.13	$A_{t_{tot}}$	0.00	**0.72**

6.3.2. Further remarks

In a recent work Zhou et al. (2023), the ground handling problem is modeled as a vehicle routing model, and a learning-based large neighborhood search that uses the MILP solver as a subprocedure to repair the destroyed part of the anchor solution during neighbor search is applied to solve the model. In the problem instances in Zhou et al. (2023), random aircraft arrivals are sampled from the interval $[5, 25]$ per hour, turnaround times are set within the time interval [30,60] in minutes, and the stands for aircraft arrivals are assigned to arriving aircraft randomly from the set of 90 aircraft stands at three terminals. For the instances with 20 aircraft arrivals over time, the learning-based solver Zhou et al. (2023) finds a solution with 12.5% gap and the MILP solver CPLEX achieves a 1.65% gap in terms of solution quality, in 1 and 5 min of *CPU time*, respectively. In contrast to this, for the total number of 50 and 100 aircraft arrivals, the learning-based method in Zhou et al. (2023) obtains better gaps (14.24%, 20.86%) than the CPLEX solver in 5 and 30 min. In our experiments, the main difference is that all flights generated in *lower_freq* and *higher_freq* scenarios are handled in a four-hours time window over three stands at one terminal, which defines rather a limited capacity with busy schedule, and the risk for collisions of vehicles. Our MILP solver obtains solutions with optimality gap of 0% for refueling instances in a *CPU time* around 0.038 s (the *median* over all instances), and achieves an optimality gap of 0% for catering vehicles approximately in 10.723 s (the *median* over all). That is, optimality is proven within seconds for catering and refueling vehicles. For the other vehicles, the MILP solver is run 30 s per vehicle and in total *CPU time* is around 100 s, including all vehicles. For the MINLP solver, the *CPU time* is around 300 s for all vehicles, due to the solution time bound of 60 s for each, and within this time bound, the waiting time on the stands are improved 100% for refueling, catering, water and lavatory service vehicles, and 98% for all vehicles. Also, due to our comparisons with the best known solutions of the *benchmark* instances of pick-up and delivery problem in preliminary experiments, we observed a 1% deviation from best known solutions in a *CPU time* limit of 30 s, for the instance sets with 100 tasks, using the warm start MILP model, after some adjustments on the model to make an exact comparison.

In Zhang et al. (2025), where automated electric baggage handling process is modeled as split-delivery vehicle routing model, the instances include 1 to 48 gates each of which handles 4 flights in a time window of 12 h. On these instances, CPLEX provides optimal solutions in 0.60 to 1.54 s for 1 to 3 gates. For 4 to 6 gates, CPLEX finds the optimal solution in a considerably longer time compared to *improved large neighborhood search*. For 7 to 12 gates, CPLEX obtains feasible solutions in two hours, while *improved large neighborhood search* finds the same or better feasible solutions in 27 to 123 s. In our instance sets, depending on the turnaround time variables generated by the probability distribution we define, each gate handles around four or five aircraft in a time window of four hours, unlike the case of four aircraft per gate in a time window of 12 h in Zhang et al. (2025). For baggage handling, the solutions we present are the ones we obtain within the time limits of 30 (60) seconds, using the MILP (MINLP) solver GUROBI.

We define four hours of planning windows to reflect different time periods of the day with busy and less busy scenarios and solve our model for different frequency levels in a four hour window, however, due to the reasonable computational time, the solver is able to find new solutions at every four-hours during the day considering changing frequency levels of flights. Thus, higher number of flights is able to be handled during an entire day at three stands, compared to the four-hours planning window, similar to Zhou et al. (2023) where they generate random aircraft arrivals per hour, but solve the model for longer periods that include several hours and more aircraft arrivals.

6.4. Integrated path planning experiments

To evaluate the performance of the path planning model we used the key performance indicators in Table 9. The examples of commonly used environments in multi-agent path finding literature are Dragon Age Origins (DAO), Open $N \times N$ grid-maps where the values of N are usually 8, 16, 32, $N \times N$ grid-maps with random obstacles, grid-maps resembling the real-world autonomous warehouses.

We perform the experiments on a 16×16 grid-map environment of a pier which is different than the classical environment structures. This environment was first presented by Chen et al. (2023) for finding conflict-free paths of GSE vehicles to solve the problem of completing the ground handling operations on aircraft stands of a specific pier at Amsterdam Schiphol Airport. The 16×16 grid-map of pier B includes three aircraft stands that are tiled next to each other as well as the service road, the depot area of the terminal, entrance and exit locations to aircraft stands. The areas occupied by static obstacles in various shapes such as the aircraft and passenger boarding bridges are also defined on this map.

To design the 16×16 grid-map environment for automated aircraft ground handling, the ramp layouts of the aircraft type B737 of Boeing and the aircraft type A320-200 of Airbus were used as reference layouts. The shapes of two aircraft types are similar and typical GSE layout on an aircraft stand is presented as a grid-structured environment by Boeing and Airbus. Airbus provides a more sophisticated layout. The layout of A320-200 in Fig. 11 includes 32 cells (unit-grids) vertically and 27 cells horizontally, each cell having an area of $2\ \text{m} \times 2\ \text{m}$, which is shown in the scale in Fig. 11. The GSE types we model usually occupy around 4 cells in Airbus layout. In the path planning model we use, each agent is allowed to occupy one cell at a timestep. Thus, in our 16×16 grid-map environment, the size of the agents (GSE vehicles) are defined as a 1×1 grid-map (1 cell) of $16\ \text{m}^2$ which is equivalent to the 2×2 grid-map (4 cells) of $16\ \text{m}^2$ in the layout of Airbus. For this reason, each cell in the 16×16 grid-map is defined as a 1×1 grid-map with $4\ \text{m} \times 4\ \text{m}$ area, where agents move 4 m per second. 16×16 grid-maps are also among the commonly used environments in multi-agent path finding literature.

The indicators in Table 9 are computed for the lower and higher flight frequency scenarios. The values of the indicators are compared to the shortest paths that are obtained without considering the collisions while generating paths. The results are presented in Table 18. The deviation of the values of the indicators that are associated with the *length* or *duration* of paths from shortest paths indicates the level of

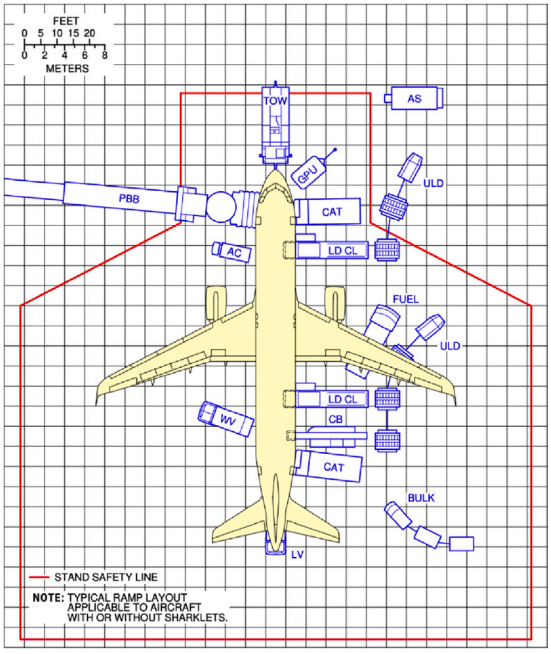

Fig. 11. Ramp layout of A320-200, Airbus .

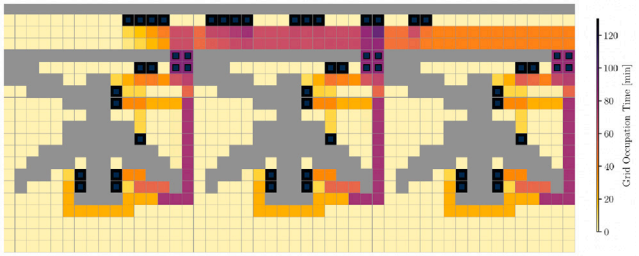

Fig. 12. Heatmap of the planned paths in the proposed solution. The squares represent the depots, service locations, and entrances and exits of bays.

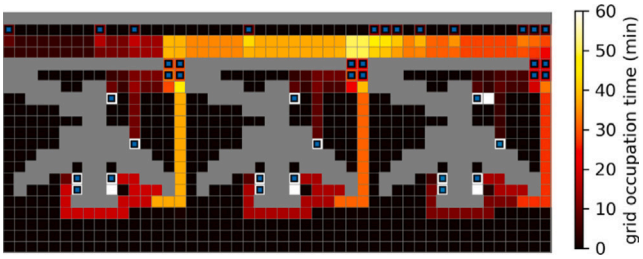

Fig. 13. Heatmap of the planned paths in the *baseline* solution Chen et al. (2023).

Table 18
Results for path planning experiments. S.P. = Shortest Path, P.SIPP = Prioritized Safe Interval Path Planning.

KPI	lower_freq		higher_freq		Deviation	
	S.P.	*P. SIPP*	*S.P.*	*P. SIPP*	*lower_freq*	*higher_freq*
$\bar{l}_{path}$	1794.93	1794.93	2230.67	2232.26	0.00%	+0.07%
$\bar{d}_{path}$	423.8	423.93	527.07	634.20	+0.03%	+20.3%
$\bar{a}_{delay}$	0.00	0.00	0.00	4.53	–	–
$\bar{T}_{delay}$	0.00	0.00	0.00	0.37	–	–
T_{ontime}	1.00	1.00	1.00	0.98	0.00%	−2.00%
a_{succes}	1.00	1.00	1.00	1.00	0.00%	0.00%
t_{tot}	3.113	3.120	4.018	4.023	+0.21%	+0.12%

achievement on finding paths and resolving conflicts of GSE vehicles in the ground handling area.

Using the proposed path planning method, the *mean path length* over 100 simulation outputs was the same in the *lower_freq* scenario compared to shortest paths, while there was a slight increase (0.03%) in *mean path duration* over 100 simulation outputs. This means that there was no detours to avoid collisions but a few waiting actions were performed to prevent collisions.

In the *higher_freq* scenario, both the *mean path length* and the *mean path duration* increased in the outputs for conflict-free paths compared to the shortest paths. The *mean path length* increased by 0.07% and *mean path duration* increased by 20.3%, which show that both detours and waiting motions were performed to avoid conflicts. In this case, the increase in the *mean path duration* could have been caused by both the increase in the *mean path length* and the waiting actions. Since the 0.07% increase in the *mean path length* is relatively small, the contribution to the 20.3% increase in the *mean path duration* was more likely due to the waiting to avoid collisions which means that waiting actions were performed more frequently than the detours. In the previous work Chen et al. (2023), the increase in average path length for a high frequency scenario (16 flights in a 4 h planning window) was 0.12% and the increase in average path duration was 28%.

The *mean* of *average agent delays* over 100 simulation outputs were both 0 for the shortest path solutions and conflict-free path solutions, in the *lower_freq* scenario. In the *higher_freq* scenario, although no delays were observed using the shortest paths, an average of 4.53 s delay occurred using the conflict-free path planning. The same pattern was observed for the *mean* of average task delays for which an average of 0.37 s delay occurred with conflict-free path planning in the *higher_freq* scenario.

The ratio of tasks that were scheduled *ontime* were equal to 1 using both the shortest and conflict free paths in the *lower_freq* scenario. This means that all tasks were *ontime*. This was also the case with shortest paths in the *higher_freq* scenario but there was a 2% decrease in task *ontime* rate when conflict-free paths are used.

The *agent success rate* was 1 in all cases in Table 18. For path planning algorithms, this shows that conflict-free paths were able to be generated for all agents which is promising. For shortest paths it only shows that there exists a path for each agent from its origin to destination, on the defined map environment.

When the *CPU time* is considered, a slight increase was observed in both scenarios for conflict-free path planning but this is not significant.

We generated a heat map of the path planning solutions, to identify the bottlenecks and analyze the paths of the GSE vehicles. The paths generated over a set of 100 simulation instances are plotted in the heatmap in Fig. 12 to visualize the frequency of the occupation of the cells of the grid.

We compare the heatmap in Fig. 12 with the heatmap of the *baseline* solution in Fig. 13.

In the heatmap in Fig. 12, a high occupancy is observed around the parking locations at the stands and terminal. While vehicles can park at the depots at the terminal, some GSE types such as *refueling* trucks do not need to visit the depot to pick up fuel, thus they wait at the parking locations at the stands between consecutive tasks. These parking locations are located at the top of the aircraft stands next to the entrance and exit locations. The goal of waiting at the parking location rather than the service location is not to occupy the service locations around the aircraft while waiting to be redirected to another task location or waiting until the service for the next task starts. Exit and entrance locations of the stands and the cells around are also among the frequently visited areas.

The *baseline* heatmap in Fig. 13 differs from the one in Fig. 12 by high occupation around the service locations rather than the parking locations or depots.This is due to the introduction of the new objective

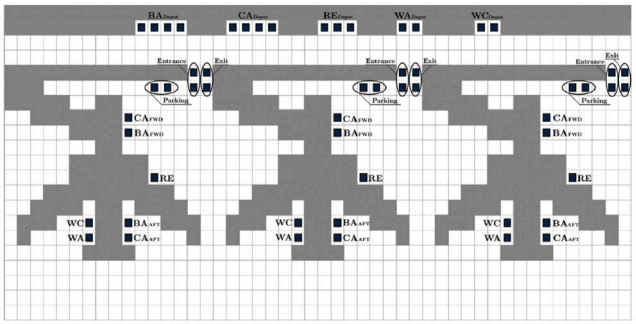

Fig. 14. The environment in the *proposed* model.

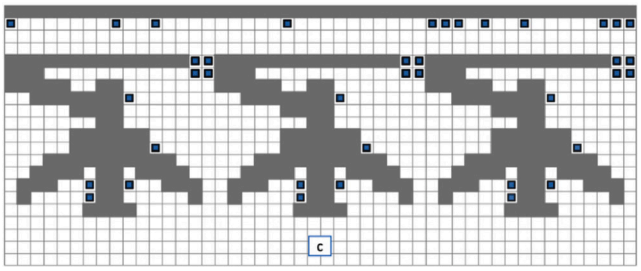

Fig. 15. The environment in the *baseline* model.

function which affects the solution in Fig. 12 by leading to the shifting of the waiting times around service locations on the stands to the depots at the terminal or to the parking locations on the stands. Apart from this difference, similar to the results in Chen et al. (2023), vehicles prefer waiting until a conflict is resolved rather than taking a detour, since the preferred paths do not highly deviate from the shortest paths.In the solution in Fig. 12, conflicts on the stands are reduced by the introduction of *holding* at the depots and *parking* at specific locations on the stands. We present the environments related to the heatmaps of the *proposed* and *baseline* solutions in Figs. 14 and 15 to make the comparison of heatmaps in Figs. 12 and 13 clear. In the environment of the *proposed* model in Fig. 14, there are seven service locations around each aircraft for *catering front*, *catering aft*, *refueling*, *baggage handling front*, *baggage handling aft*, *water service*, *lavatory service* tasks, and the *black squares* around the aircraft in the related heatmap in Fig. 12 represent the locations of these tasks. In the *baseline* environment in Fig. 15 there are five service locations for each aircraft, since the *catering* and *baggage handling* tasks replace the *catering front*, *catering aft*, *baggage handling front*, *baggage handling aft* tasks. The service locations of the *baseline* model are represented as *blue squares* in the heatmap of the *baseline* solution in Fig. 13. The lighter colors in the *baseline* heatmap in Fig. 13 show the higher occupation of cells, thus the *white* colors around the service locations mean high occupation. In the heatmap of the *proposed* solution in Fig. 12, darker colors mean higher occupancy and these can be seen at parking, depot, exit, entrance, service locations, and on the cells that connect service locations to entrance and exit locations and the cells that connect entrance and exit to other aircraft stands or to depots at the terminal. Parking locations do not exist in the *baseline* and depot locations at the terminal are different compared to the environment of the *proposed* model.

In Tables 19 and 20, we compare the performance indicators of the proposed path planning, which is performed taking the solutions of the proposed warm start mathematical models as input and also introducing parking locations at the stands for some vehicles in addition to holding at depot locations, with the results in Chen et al. (2023) where the performance indicators for the *normal* scenario are presented in detail. The *normal* scenario in Chen et al. (2023) is similar to *higher_freq*

Table 19
The deviations of conflict-free path length and duration in the *proposed solution* from the shortest paths with conflicts in *lower_freq* and *higher_freq* scenarios, compared to the *baseline solution*.

KPI	% deviation from shortest path		
	lower_freq	*higher_freq*	*baseline solution* Chen et al. (2023)
$\bar{l}_{path}$	0.00%	+0.07%	+0.12%
$\bar{d}_{path}$	+0.03%	+20.3%	+28%

Table 20
Agent and *task delays* ($\bar{a}_{delay}$ - $\bar{T}_{delay}$), *task ontime* and *agent success rates* (T_{ontime} - a_{succes}) and *CPU time* (t_{tot}) of the *proposed solution* in *lower_freq* and *higher_freq* scenarios, compared to the *baseline solution*.

KPI	*lower_freq*	*higher_freq*	Chen et al. (2023)	
$\bar{a}_{delay}$	0.00	4.53	22.28	*seconds*
$\bar{T}_{delay}$	0.00	0.37	15.49	*seconds*
T_{ontime}	1.00	0.98	0.96	*scale*: 0.00-1.00
a_{succes}	1.00	1.00	1.00	*scale*: 0.00-1.00
t_{tot}	3.120	4.023	5.11	*seconds*

in our instances, in terms of flight frequency. There are higher number of ground handling tasks per flight in our scenarios, since we split some tasks into front and aft tasks. Therefore, in terms of number of tasks, our *lower_freq* and *higher_freq* scenarios converge to the *normal* and *peak* scenarios in Chen et al. (2023), respectively. Although we take the flight frequency as the main reference to compare the results, we include the results for both the *lower_freq* and *higher_freq* scenarios, compared to the *normal* scenario in the *baseline*, due to its equivalency to *higher_freq* scenario in terms of flight frequency and to *lower_freq* scenario in terms of number of tasks.

Table 19 shows that the solutions for the *proposed* model outperform the *baseline* both in the *lower_freq* and *higher_freq* scenarios, which are equivalent to the scenario of the *baseline* in terms of number of tasks and flight frequency, respectively. The smaller the deviation of *length* ($\bar{l}_{path}$) and *duration* ($\bar{d}_{path}$) of the conflict-free paths from the shortest paths, the better the solution quality is. The 0.00% and +0.07% increases in $\bar{l}_{path}$ in the *lower_freq* and *higher_freq* scenarios, compared to +0.12% in the *baseline* indicate that the conflict free paths were able to be obtained without highly increasing the length of the shortest paths in the *proposed* model. This is due to the existence of less conflicts based on the proposed MILP and MINLP solutions, which define the *target locations* and *tour schedules* of the automated vehicles before planning paths and allocating parking locations on the stands. That is, the introduction of *holding* and *parking* options improves the solutions. The +0.03% and +20.3% increases in $\bar{d}_{path}$ are also lower than the +28% increase in the *baseline*, vehicles tend to wait rather than detour as in the *baseline* but this waiting time is reduced.

In Table 20, the values of the performance indicators, *agent delay* ($\bar{a}_{delay}$) and *task delay* ($\bar{T}_{delay}$) are significantly better both in the *lower_freq* and *higher_freq* scenarios, compared to the delays in the equivalent *baseline* scenario. The *task ontime rate* (T_{ontime}) is also better, taking higher ratios in *lower_freq* and *higher_freq* scenarios, compared to *baseline*. The *agent success rate* ($a_{success}$) remains the same, taking the value of 1.00 over a scale of 0.00 - 1.00, in all solutions. This means that all conflicts were resolved. The *CPU time* (t_{tot}) remains similar, 3.12 and 4.024 s in the *lower_freq* and *higher_freq* scenarios, compared to 5.11 s in the *baseline*.

6.5. Replanning experiments

In replanning experiments, we measure the ability to respond disruptions, which are in the form of task delays. We select the baggage handling tasks for replanning tests, since a number of external conditions impact baggage handling operations. The passengers not arriving at the gate, delays in transfer flights and luggage loading tasks are examples to disruptions of baggage handling operations. We model

Table 21
Results for replanning.

KPI	*replanned*	*original*
$\bar{M}_{BA}$	262.17	251.16
$\bar{s}^R_{BA}$	815.72	815.32

the disruption as delay on loading luggage. We define the delay as uniformly distributed variable, $U \sim (0.3 \cdot d_i, 0.5 \cdot d_i)$, bounded by $0.3 \cdot d_i$ and $0.5 \cdot d_i$, where d_i is the task duration. After introducing the delay in obtained routes, a quick verification procedure is called whether the subsequent tasks can still be completed on time or not. If several of subsequent tasks are delayed due to the disruption of one task and if the total delay is larger than 5 min, replanning is activated. All tasks scheduled after the disrupted task are replanned using the adapted insertion heuristic described in Section 5.3.

The indicators presented in Table 7 regarding the specific GSE types are evaluated for replanning experiments. The insertion heuristic which is used for replanning obtains feasible solutions. On the other hand, although a solution time limit is set which might stop the solver before reaching optimality, the proposed warm start MILP model which is used to solve the task allocation and routing problem produces results that are closer to optimality, compared to the results of insertion heuristic that is used for replanning.

Since no optimality guarantee exists for the heuristic solution and due to the delays in disrupted and subsequent tasks, we expected that the *median* of the *mean makespans* of baggage handling solutions would be higher in the replanned schedule, compared to the original solution. Furthermore, we also expected an increase on the *median* of the *mean start times* of replanned tasks, $\bar{s}^R_{BA}$.

The results are presented in Table 21. The $\bar{s}^R_{BA}$ in Table 21 is the *mean* of the *start times* of the replanned tasks, which excludes the start times of tasks that remain unchanged since they precede the disrupted task, and the related outputs for the *replanned* and *original* schedules show the *medians* of 100 simulation outputs regarding the *mean start times* for baggage handling. In Table 21, we compare the *median* of *mean makespans* ($\bar{M}_{BA}$) for baggage handling, and the *median* of *mean start times* for replanned tasks, $\bar{s}^R_{BA}$, for *original* and *replanned* schedules. There is also the *disrupted* schedule which is the schedule constructed by shifting tasks after the delayed task and no replanning algorithm is applied. The *replanned* schedule is produced by applying the insertion heuristic for replanning the disrupted schedule.

When the results were analyzed, no difference in the *median* values of *mean makespans* was observed between *disrupted* and *original* schedule. This could mean that the extended times that contribute to makespan could have been compensated by the waiting time buffers that were originally occupying a larger portion of the makespan. On the other hand the *median* of the *mean makespans* has increased by 4.38% in the *replanned* schedule. This might be due to the relaxed time windows that are set for replanning which might have led the replanning algorithm to insert more tasks to some vehicles or due to the change in allocation structure after reallocation of remaining tasks. The *median* of the *mean start times* of replanned tasks over 100 simulation outputs has increased by 0.05% compared to the *original* solution. The percentage of the increase was higher in *disrupted* schedule compared to the *replanned* schedule. Application of replanning algorithm on the *disrupted* schedule has improved the *median* of the *mean start times* by reducing the *start times*. Thus, even though the *median* of the *mean start times* is still higher in the *replanned* schedule compared to *original* schedule, it could have been higher if replanning was not applied, which was observed in the *disrupted* schedule.

6.6. Sensitivity analysis

We performed the sensitivity analysis, to evaluate the impact of changing certain input parameters. The GSE vehicles are the most critical resources. The limited number of available GSE vehicles increases the complexity of the task allocation and routing model. Therefore, we measured the sensitivity of the proposed task allocation and routing model by changing the number of available vehicles while keeping other parameters unchanged. We present the results for the *higher_freq* (*peak*) and the *lower_freq* (*normal*) scenarios.

Table 22
The minimum required, original, and increased number of available vehicles to serve three stands in a four-hour planning window.

GSE type	*minimum*	*baseline*	*maximum*
RE	3	3	4
CA	4	4	5
BA	4	4	5
WA	1	2	3
WC	1	2	3
total	13	15	20

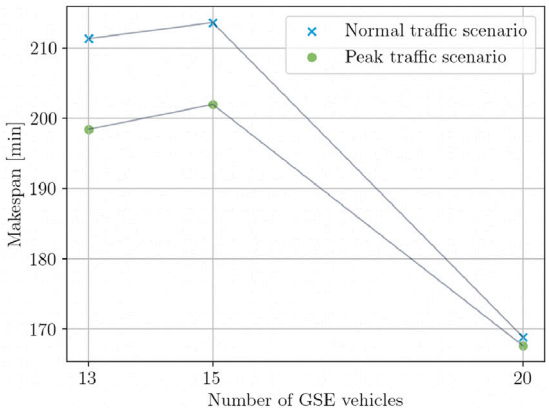

Fig. 16. Makespan indicator for different number of GSE vehicles.

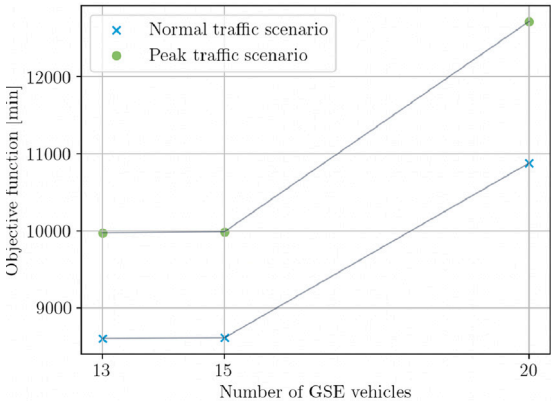

Fig. 17. Objective function value for different number of GSE vehicles.

The number of vehicles is increased by 1 for each GSE type. The minimum number of required GSE types (*minimum*), the number of available GSE types in real life (*baseline*) and the increased number of available GSE types (*maximum*) that are included in sensitivity analysis are presented in Table 22. We define the minimum number of vehicles, the *minimum* in Table 22, by solving the PDPTW, by adapting the proposed MILP to allow some of the vehicles remain unused and minimize also the number of used vehicles.

The results are plotted for the *makespan* and the MILP objective *min* $z = z_1 + z_2$ in Figs. 16 and 17. Both *lower_freq* (*normal*) and *higher_freq* (*peak*) scenarios are included in Figs. 16 and 17.

The results in Figs. 16 and 17 show similar trends when the number of GSE vehicles increases. When the number of vehicles is minimized, there is a slight decrease in the value of *makespan indicator* compared to

original number of vehicles, in *lower_freq* and *higher_freq* scenarios, and in both scenarios, the objective function values which also include the sum of start times remained almost unchanged. This might be due to the fact that there is not a big difference between the *minimized* and the *original* number of vehicles, which are respectively 13 and 15 vehicles. When the total number of available GSE vehicles is increased to 20, the value of the *makespan* indicator decreased significantly. The reason is most likely the case that the tasks are handled in parallel by higher number of vehicles. However, the value of the indicator for the objective function, which is the sum of two objectives where one of them is the total of start times of all tasks, increased when the number of vehicles was higher compared to the original number. The value of the sum of start times depends on the sequence of tasks in the schedule. While for minimizing the makespan only the maximum start or completion time over all tasks is crucial, to minimize the total start time starting or finishing times of all tasks needs to be considered and it is highly sensitive to the sequencing structure.

7. Discussion

We defined the automated ground handling problem consulting the experts in the field and referring to technical reports of Boeing and Airbus. Due to the definition of the problem, reducing the problem to multi-vehicle pick-up and delivery problem with time windows and its variations provides with a well defined structure to exploit the state-of-the-art mathematical models and solvers. We used the mathematical models of multi-vehicle pick-up and delivery problems for allocation and routing of the tasks in the corresponding ground handling problems.

Using the proposed MILP and MINLP models integrated with the proposed path planning algorithm, all tasks were able to be allocated and the success rate of finding paths was 1 (100%) for all agents. Furthermore, the time the cells around service locations are occupied at the stands were considerably reduced due to the proposed MINLP model and integrated path planning. We measured the performance of the model using several experiments that focus on the scalability, robustness, and solution quality.

To test the scalability of the proposed warm start MILP model, we performed experiments on large-scale PDPTW benchmark instances for which best known solutions are publicly available. In the existing literature, best known solutions are produced by a heuristic algorithm. We compared the solution quality of the results we obtain by solving the proposed warm start MILP model within a CPU time limit of 30 s with the solution quality of best known solutions. For comparison, we converted the proposed warm start MILP model into PDPTW form by small adaptations to allow some vehicles to remain unused and introducing also the objective of minimizing the number of used vehicles. For randomly selected instances with 100 tasks, we achieved a similar solution quality with a max of 1% deviation. Thus, the scalability of the proposed MILP model was high enough to solve the real time instances.

In real life instances, the number of tasks that need to be handled in a four-hour planning window is not higher than 100 even in frequent flights scenarios, which were defined based on the discussions with industry experts. For these instances, we performed statistical analysis to evaluate the results in lower and higher frequency flight scenarios (*lower_freq* and *higher_freq*), compared the proposed MILP and MINLP models, analyzed specific performance indicators and performed sensitivity analysis for changing the number of available vehicles, to investigate the ability of the model in dealing with potential cases in real life.

For the proposed MILP, we compared the makespan and CPU time for *lower_freq* and *higher_freq* scenarios. Despite expecting the opposite, increasing the number of tasks in a planning window did not lead to an increase in makespan. This was due to the time gaps between tasks in the *lower_freq* scenario which were reduced by 25% in *higher_freq* scenario. This means that the *lower_freq* scenario allowed inserting new tasks without increasing the average makespan in *higher_freq* scenario. Also, no significant increase was observed in CPU time. Thus, there was no deterioration in solution quality and CPU time in *higher_freq* scenario.

Wilcoxon signed-rank test resulted in non-significant *p*-value, which also suggests that there is no statistically significant difference in *makespan* between *lower_freq* and *higher_freq* scenarios. However, the A-value from the Vargha–Delaney test revealed a moderate effect size which means that there is a moderate impact of adding additional flights without changing the number of vehicles. This effect size was only observed on catering and baggage handling. This indicates that tasks were more tightly planned on these vehicles in *lower_freq* scenario and inserting new tasks was not possible without increasing the makespan in *lower_freq* scenario. In the statistical analysis for the CPU time, *p*-value did not show a significant difference. However Vargha–Delaney A-value showed a large effect size. This could be due to the significant increase in CPU time for the catering vehicle, since the CPU time did not highly deviate for other vehicles.

Compared to the results of Chen et al. (2023), we observed that the average makespan increased by 39% in one of the scenarios. However, the decentralized task allocation method proposed by Chen et al. (2023) did not always allocate all tasks since there was a negotiation based approach that requires solving a single vehicle pick-up and delivery optimization for each vehicle separately and applying an auction to select the winner at each iteration. Therefore, Chen et al. (2023) reported *task allocation rate* to show the performance of the model in allocation of tasks. In the centralized task allocation method we used, the multi-vehicle optimization method was able to allocate all tasks to vehicles in all instances. Thus, task allocation rate was always 100%. A failure in allocating tasks is only possible when there is no feasible solution, which was not observed for the instances we tested.

Apart from the difference in modeling, we introduced new tasks by splitting the baggage handling and catering tasks into front and aft tasks and set the duration of some tasks differently, which could be other reasons for difference in *makespan* indicator in Chen et al. (2023).

We compared the proposed MILP and MINLP model with respect to average makespan and average waiting time at stands and depots. In the proposed warm start MINLP model, we had introduced a new nonlinear objective function to the proposed warm start MILP model. The proposed MINLP achieved the aim of motivating vehicles to wait at the depots instead of service locations at the stands, without creating a significant increase in makespan. There was a 98% decrease in waiting time at the aircraft stands compared to the solution with the original objective and waiting times at the depots were increased. In the resulting schedules, waiting times were shifted from service locations to parking or depot locations.

In the MINLP model, we defined the depot locations also as parking locations to allow waiting at the depot instead of service locations at the stands. A further improvement could be adding parking locations for all vehicle types at the stands. These locations should be defined in such a way that they do not block GSE traffic.

We performed path planning experiments to evaluate the ability to find conflict-free solutions given the MINLP solutions. For the path planning model, for some vehicle types such as fuel hydrant and baggage handling trucks, we set parking locations at the stands unlike the case in MINLP model where depots are used as waiting locations in task allocation and routing solutions. Refueling vehicle does not need to visit the depot to pick up fuel since an underground fueling system exists and it is more convenient for baggage handling vehicles to wait at a parking location at the stand between front and aft tasks rather than waiting at service locations or traveling back to depot. Unlike this, for solving the mathematical model we created a virtual location at the depot for the refueling vehicle and used the depot location of the baggage handling location also as a parking location for simplicity. Introducing parking locations at the stands for some of the vehicles

makes our path planning model different from the model in Chen et al. (2023).

The obtained conflict-free paths were identical to shortest paths in the *lower_freq* scenario. In the heatmap generated over a number of simulations, usually the shortest paths were selected. Thus, *waiting motions* instead of *detours* were more likely to occur, as in Chen et al. (2023). Thus, rather than the *path length*, the *path duration* increased in *conflict-free paths* compared to the *shortest paths*. The increase in path duration as a result of waiting actions was higher in Chen et al. (2023) compared to our model, due to the introduction of parking locations at the stands and depot locations in our model. In the generated schedules, vehicles avoid occupying service locations, but stay longer at depots or parking locations. Therefore, the safe time intervals during which any vehicle can pass through a grid unit with no collision risk tend to be longer. In the *higher_freq* scenario, there was also a higher deviation in *path duration* compared to *path length*.

The success rate of the proposed path planning algorithm was 1(100%) when integrated with the proposed allocation model. The success rate is an indicator that is commonly used to measure the performance of path finding algorithms on various environments. In path planning, all tasks were completed on time with respect to the time windows constraints in the *lower_freq* scenario. For the *higher_freq* scenario, average delays per agent and task were respectively 4.53 and 0.37 s, which are acceptable in real world applications. Thus, solving the centralized task allocation and routing using the proposed MILP and MINLP models and finding the collision-free paths of GSE vehicles using the integrated solver resulted in high-quality solutions that help to achieve the goal of automating ground handling operations in real life.

We measured the ability of responding to disruptions by introducing stochastic delays for baggage handling tasks and compared the results for the routes with no disruption, delayed routes without replanning and replanned routes. The increases in start times of tasks in disrupted schedule of tasks were minimized by replanning, however could not be avoided. The increase in makespan and start times were both negligible in replanned schedules. Replanning was completed in a CPU time of 8.9E−4 s.

Sensitivity analysis showed the impact of increasing and reducing the number of available vehicles which can be used as a guideline for allocating the right GSE fleet size in different conditions.

In the real life instances of the automated ground handling problem we define, due to the limited number of gates and restricted time window, the number of aircraft that can be served is subject to capacity constraints. Thus, with respect to the number of tasks that are to be handled in the constrained problem, the instance sets are moderate in size. In this regard, the instances do not constitute very large scale vehicle routing problems but rather present complexity due to congestion. Using the mathematical models, we obtain good solutions in a short time for real life and larger instances. Optimality is proven in less than 30 s for some vehicles in real life instances and the objective values of the solutions are close to best known solutions for larger instances. Thus, we use mathematical programming solvers, rather than metaheuristics or learning-based methods. The results in a recent research Zhou et al. (2023) also show that even though learning based methods perform well on large scale vehicle routing instances of a ground handling problem, the related mathematical programming solver perform better in small instances (20 flights), regarding optimality gaps.

Further research could include shapes and kinematics of vehicles, different replanning scenarios, modeling parking locations for all GSE types at the parking stands. Also, we restricted the ground handling problem to the north side of pier B at Amsterdam Schiphol Airport, since we aimed to solve safety issues of autonomous vehicles due to the collision risks in a small area when the schedules of the vehicles are busy. A larger area can include several or all piers at the airport or both sides of the pier B. Although it is possible to extend the problem to a larger area considering also the safety aspect, a fleet allocation among the piers at the higher level of planning is required first, which converts the automated ground handling problem into a new complex form. That is, given the restricted GSE fleet size at the north side of the pier, it takes time for the GSE fleet to visit other piers or serve to the aircraft on the other side of the pier, and could lead to the violation of time constraints and increase in turnaround times. Thus, a dedicated fleet of GSE vehicles should be assigned to each pier including both sides, considering the flight schedules and other constraints, after which the automated ground handling problem is solved for each pier or pier side. This generalization of the problem, where GSE fleet allocation, task allocation and path planning problems are integrated to address the safety issues on a large area with several or more piers, remains as an interesting direction for further research.

8. Conclusion

We proposed warm start MILP and MINLP models for allocating and routing tasks on a 16 × 16 grid-based ground handling environment as well as solved the related path planning problem to prevent collisions of automated GSE vehicles. We showed that utilization of the proposed global optimization technique for the assignment of ground handling tasks improved the computational time of the multi-agent task allocation model of Chen et al. (2023) by 48% in normal *higher_freq*) scenario, and unlike the existence of unallocated tasks in Chen et al. (2023), we achieved 100% allocation rate for all instances. Also introducing a nonlinear objective function that rewards waiting at depot or parking locations rather than service locations at the stands has led to a reduction on GSE traffic and obstacles at the stands and facilitated finding conflict-free solutions in path planning problem.

CRediT authorship contribution statement

Manouk van der Zwan: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Gülçin Ermiş:** Writing – review & editing, Writing – original draft, Supervision, Software, Methodology, Formal analysis, Conceptualization. **Alexei Sharpanskykh:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by SESAR 3 Joint Undertaking under grant agreement No. 101114684, under the European Union's Horizon Europe research and innovation program.

References

Allaart, G., Kaijen, J., 2022. Aircraft Stand Table. Amsterdam Airport Schiphol, Schiphol, The Netherlands.

Aviation Learnings Team, 2020. A guide to airport ramp operations, ground handling & Ground Support Equipment(GSE). URL: https://aviationlearnings.com/ground-handling-ramp-operations-aircraft-ground-support-equipment-gse-machines-that-supplement-the-airplane/#Common_Terminologies.

Bao, D.-W., Zhou, J.-Y., Zhang, Z.-Q., Chen, Z., Kang, D., 2023. Mixed fleet scheduling method for airport ground service vehicles under the trend of electrification. J. Air Transp. Manag. 108, 102379. http://dx.doi.org/10.1016/j.jairtraman.2023.102379.

Bianchessi, N., Righini, G., 2007. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Comput. Oper. Res. 34 (2), 578–594. http://dx.doi.org/10.1016/j.cor.2005.03.014.

Boeing Commercial Airplanes, 2022. 737 MAX Airplane Characteristics for Airport Planning. Technical Report, Boeing, pp. 100–115, Document number: D638A004.

Bredström, D., Rönnqvist, M., 2008. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. European J. Oper. Res. 191 (1), http://dx.doi.org/10.1016/j.ejor.2007.07.033.

Bye, R., Schaathun, H., 2014. An improved receding horizon genetic algorithm for the tug fleet optimisation problem. In: ECMS 2014 Proceedings, vol. 2014-May, pp. 682–690. http://dx.doi.org/10.7148/2014-0682.

Chen, Z., Alonso-Mora, J., Bai, X., Harabor, D.D., Stuckey, P.J., 2021. Integrated task assignment and path planning for capacitated multi-agent pickup and delivery. IEEE Robot. Autom. Lett. 6 (3), http://dx.doi.org/10.1109/LRA.2021.3074883.

Chen, S.T., Ermis, G., Sharpanskykh, A., 2023. Multi-agent planning and coordination for automated aircraft ground handling. Robot. Auton. Syst. 167, 104480. http://dx.doi.org/10.1016/j.robot.2023.104480.

Chui, S., 2020. Amsterdam air traffic control visit. URL: https://samchui.com/2020/01/20/amsterdam-air-traffic-control-visit/#.Y2t9iS8w30p.

Contini, A., Farinelli, A., 2021. Coordination approaches for multi-item pickup and delivery in logistic scenarios[Formula presented]. Robot. Auton. Syst. 146, http://dx.doi.org/10.1016/j.robot.2021.103871.

Dohn, A., Rasmussen, M.S., Larsen, J., 2011. The vehicle routing problem with time windows and temporal dependencies. In: Networks, vol. 58 no. 4, http://dx.doi.org/10.1002/net.20472.

Federal Aviation Administration, 2025. URL: https://www.faa.gov/air_traffic/publications/atpubs/aim_html/chap4_section_3.html.

Gerkey, B.P., Matarić, M.J., 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. Int. J. Robot. Res. 23 (9), http://dx.doi.org/10.1177/0278364904045564.

Gombolay, M.C., Wilcox, R.J., Shah, J.A., 2018. Fast scheduling of robot teams performing tasks with temporospatial constraints. IEEE Trans. Robot. 34 (1), http://dx.doi.org/10.1109/TRO.2018.2795034.

Guimarans, D., Padrón, S., 2022. A stochastic approach for planning airport ground support resources. Int. Trans. Oper. Res. 29 (6), http://dx.doi.org/10.1111/itor.13104.

HSE Office, 2022a. Schipholregels, 30.2 ed. Amsterdam Airport Schiphol, Schiphol, The Netherlands.

HSE Office, 2022b. Safety and Security Pocketguide. Amsterdam Airport Schiphol, Schiphol, The Netherlands.

International Air Transport Association (IATA), 2019. Industry statistics fact sheet. URL: https://www.iata.org/en/iata-repository/publications/economic-reports/fact-sheet-industry-statistics-dec19.pdf.

International Air Transport Association (IATA), 2022. Air passenger numbers to recover in 2024. URL: https://www.iata.org/en/pressroom/2022-releases/2022-03-01-01/.

Khamis, A., Hussein, A., Elmogy, A., 2015. Multi-robot task allocation: A review of the state-of-the-art. Stud. Comput. Intell. 604, http://dx.doi.org/10.1007/978-3-319-18299-5{_}2.

Korsah, G.A., Stentz, A., Dias, M.B., 2013. A comprehensive taxonomy for multi-robot task allocation. Int. J. Robot. Res. 32 (12), http://dx.doi.org/10.1177/0278364913496484.

Kuhn, H.W., 2010. The Hungarian method for the assignment problem. In: 50 Years of Integer Programming 1958-2008: from the Early Years to the State-Of-The-Art. http://dx.doi.org/10.1007/978-3-540-68279-0{_}2.

Kumar, S.N., Panneerselvam, R., 2012. A survey on the vehicle routing problem and its variants. Intell. Inf. Manag. 04 (03), 66–74. http://dx.doi.org/10.4236/iim.2012.43010.

Li, H., Lim, A., 2001. A metaheuristic for the pickup and delivery problem with time windows. In: Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. pp. 160–167.

Li, J., Tinka, A., Kiesel, S., Durham, J.W., Satish Kumar, T.K., Koenig, S., 2020. Lifelong multi-agent path finding in large-scale warehouses. In: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, vol. 2020-May.

Liu, M., Ma, H., Li, J., Koenig, S., 2019. Task and path planning for multi-agent pickup and delivery. In: 18th International Conference on Autonomous Agents and Multiagent Systems. URL: www.ifaamas.org.

Liu, R., Xie, X., Augusto, V., Rodriguez, C., 2013. Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. European J. Oper. Res. 230 (3), 475–486. http://dx.doi.org/10.1016/j.ejor.2013.04.044.

Ma, H., Hönig, W., Satish Kumar, T.K., Ayanian, N., Koenig, S., 2019. Lifelong path planning with kinematic constraints for multi-agent pickup and delivery. In: 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019.

Ma, H., Koenig, S., 2016. Optimal target assignment and path finding for teams of agents. In: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS.

Mirjalili, S., 2019. Evolutionary Algorithms and Neural Networks Theory and Applications, vol. 780, Springer, Warsaw, URL: http://www.springer.com/series/7092.

Morris, R., Păsăreanu, C.S., Luckow, K., Malik, W., Ma, H., Satish Kumar, T.K., Koenig, S., 2016. Planning, scheduling and monitoring for airport surface operations. In: AAAI Workshop - Technical Report, vol. WS-16-01 - WS-16-15.

Norin, A., Yuan, D., Granberg, T., Värbrand, P., 2012. Scheduling de-icing vehicles within airport logistics: a heuristic algorithm and performance evaluation. J. Oper. Res. Soc. 63 (8), 1116–1125. http://dx.doi.org/10.1057/jors.2011.100.

Padrón, S., Guimarans, D., 2019. An improved method for scheduling aircraft ground handling operations from a global perspective. Asia-Pac. J. Oper. Res. 36 (4), http://dx.doi.org/10.1142/S0217595919500209.

Padrón, S., Guimarans, D., Ramos, J.J., Fitouri-Trabelsi, S., 2016. A bi-objective approach for scheduling ground-handling vehicles in airports. Comput. Oper. Res. 71, http://dx.doi.org/10.1016/j.cor.2015.12.010.

Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008. A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations. J. Fur Betriebswirtschaft 58 (2), http://dx.doi.org/10.1007/s11301-008-0036-4.

Phillips, M., Likhachev, M., 2011. SIPP: Safe interval path planning for dynamic environments. In: Proceedings - IEEE International Conference on Robotics and Automation. http://dx.doi.org/10.1109/ICRA.2011.5980306.

Ponda, S.S., 2012. Robust Distributed Planning Strategies for Autonomous Multi-Agent Teams (ProQuest Dissertations and Theses 0828990).

Royal Schiphol Group, 2020. Vision 2050; Storyline. Amsterdam Airport Schiphol, Schiphol, The Netherlands.

Saggar, S., Tomasella, M., Cattaneo, G., Matta, A., 2021. Enhanced operational management of airport ground support equipment for better aircraft turnaround performance. In: Proceedings - Winter Simulation Conference, vol. 2021-December, Institute of Electrical and Electronics Engineers Inc., http://dx.doi.org/10.1109/WSC52266.2021.9715320.

Schiphol Innovation, 2020. Real-time data for smarter working: Smart VOP 1.0. URL: https://www.schiphol.nl/en/innovation/blog/smart-vop/.

Schiphol Innovation, 2021. An autonomous airport in 2050. URL: https://www.schiphol.nl/en/innovation/blog/an-autonomous-airport-in-2050/.

Schiphol News Room, 2021. Schiphol tests self-driving baggage tractor. URL: https://news.schiphol.com/schiphol-tests-self-driving-baggage-tractor/?.

Schiphol Newsroom, 2020. 2 april 2020 - update: Schiphol zet volgende stap naar kern schiphol en sluit pieren. URL: https://nieuws.schiphol.nl/schiphol-blijft-open-met-fors-kleinere-operatie/?.

Schmidt, M., 2017. A review of aircraft turnaround operations and simulations. Prog. Aerosp. Sci. 92, http://dx.doi.org/10.1016/j.paerosci.2017.05.002.

Schmidt, M., Nguyen, P., Hornung, M., 2015. Novel aircraft ground operation concepts based on clustering of interfaces. In: SAE Technical Papers, vol. 2015-September, http://dx.doi.org/10.4271/2015-01-2401.

Sheibani, K., 2020. Scheduling aircraft ground handling operations under uncertainty using critical path analysis and Monte Carlo simulation: Survey and research directions. Int. J. Bus. Strat. Autom. 1 (1), 37–45. http://dx.doi.org/10.4018/IJBSA.2020010103.

Sol, M., Savelsbergh, M.W.P., 1995. The general pickup and delivery problem. Transp. Sci. 29 (1), 17–29.

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35 (2), 254–265. http://dx.doi.org/10.1287/opre.35.2.254.

Tabares, D.A., Mora-Camino, F., 2017a. Aircraft ground handling: Analysis for automation. In: 17th AIAA Aviation Technology, Integration, and Operations Conference, 2017. http://dx.doi.org/10.2514/6.2017-3425.

Tabares, D.A., Mora-Camino, F., 2017b. Aircraft ground handling: analysis for automation. In: 17th AIAA Aviation Technology, Integration, and Operations Conference. p. 3425.

Tabares, D., Mora-Camino, F., A., D., 2021. A multi-time scale management structure for airport ground handling automation. J. Air Transp. Manag. 90 (1), 101959. http://dx.doi.org/10.1016/j.jairtraman.2020.101959.

Volt, J., Stojić, S., Had, P., 2022. Optimization of the baggage loading and unloading equipment. Transp. Res. Procedia 65, 246–255. http://dx.doi.org/10.1016/j.trpro.2022.11.029.

Wang, S., Che, Y., Zhao, H., Lim, A., 2021. Accurate tracking, collision detection, and optimal scheduling of airport ground support equipment. IEEE Internet Things J. 8 (1), 572–584. http://dx.doi.org/10.1109/JIOT.2020.3004874.

Wang, J., Zhou, Y., Wang, Y., Zhang, J., Chen, C.L., Zheng, Z., 2016. Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms. IEEE Trans. Cybern. 46 (3), 582–594. http://dx.doi.org/10.1109/TCYB.2015.2409837.

Wu, Y., Zhou, J., Xia, Y., Zhang, X., Cao, Z., Zhang, J., 2023. Neural airport ground handling. IEEE Trans. Intell. Transp. Syst. 24 (12), 15652–15666. http://dx.doi.org/10.1109/TITS.2023.3253552.

Zhang, H., Ge, H., Yang, J., Tong, Y., 2022. Review of vehicle routing problems: Models, classification and solving algorithms. Arch. Comput. Methods Eng. 29 (1), 195–221. http://dx.doi.org/10.1007/s11831-021-09574-x.

Zhang, X., Wang, X., Dong, W., Xu, G., 2025. Adaptive large neighborhood search for autonomous electric vehicle scheduling in airport baggage transport service. Comput. Oper. Res. 182, 107086. http://dx.doi.org/10.1016/j.cor.2025.107086, URL: https://www.sciencedirect.com/science/article/pii/S0305054825001145.

Zhou, J., Wu, Y., Cao, Z., Song, W., Zhang, J., Chen, Z., 2023. Learning large neighborhood search for vehicle routing in airport ground handling . IEEE Trans. Knowl. & Data Eng. 35 (09), 9769–9782. http://dx.doi.org/10.1109/TKDE.2023.3249799, URL: https://doi.ieeecomputersociety.org/10.1109/TKDE.2023.3249799.