Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

A SIMHEURISTIC APPROACH FOR ROBUST SCHEDULING OF AIRPORT TURNAROUND TEAMS

Yagmur S. Gök
Maurizio Tomasella

University of Edinburgh Business School
29 Buccleuch Place
Edinburgh, EH8 9JS, UK

Daniel Guimarans

Amazon
22 Rue Edward Steichen
L-2540 Luxembourg, LUXEMBOURG

Cemalettin Ozturk

Raytheon Technologies, United Technologies Research Center Ireland
2nd Floor, Penrose Business Centre, Penrose Wharf
Cork, T23 XN53, IRELAND

ABSTRACT

The problem of developing robust daily schedules for the teams turning around aircraft at airports has recently been approached through an efficient combination of project scheduling and vehicle routing models, and solved jointly by constraint programming and mixed integer programming solvers, organized in a matheuristic approach based on large neighborhood search. Therein, robustness is achieved through optimally allocating time windows to tasks, as well as allocating slack times to the routes to be followed by each team throughout their working shift. We enhance that approach by integrating discrete event simulation within a simheuristic scheme, whereby results from simulation provide constructive feedback to improve the overall robustness of the plan. This is achieved as a trade-off between the interests of each separate turnaround service provider and that of the airport as a whole. Numerical experiments show the applicability of the developed approach as a decision support mechanism at any airport.

1 MOTIVATION

Aircraft turnaround services consist of sequences of operations through which aircraft, while parked in areas of an airport known as *aprons*, are set ready for the following take-off. Several operations may be needed: passenger disembarking and boarding, baggage unloading and loading, refueling, cabin cleaning, catering, toilet and potable water servicing, and aircraft push-back. Figure 1 shows an example of a typical turnaround. A number of related important aspects of apron operations can be highlighted:

- Ideally, each turnaround should start as soon as the aircraft arrives at its parking stand, preferably at the time for which it was scheduled to arrive there – or *Start of In-Block Time* $(SIBT)$ – and should be completed by the time it was scheduled to be pushed-back into the taxiway on its way to the runway – which we call its *Start of Off-Block Time* $(SOBT)$.
- Conjunctive directed arcs within each turnaround represent precedence relations between services, e.g., refueling often cannot start before passengers have disembarked due to safety regulations.
- There exist specific time windows through which each activity should be performed. Each time window shortens or stretches depending on what happens to other related operations. Setting

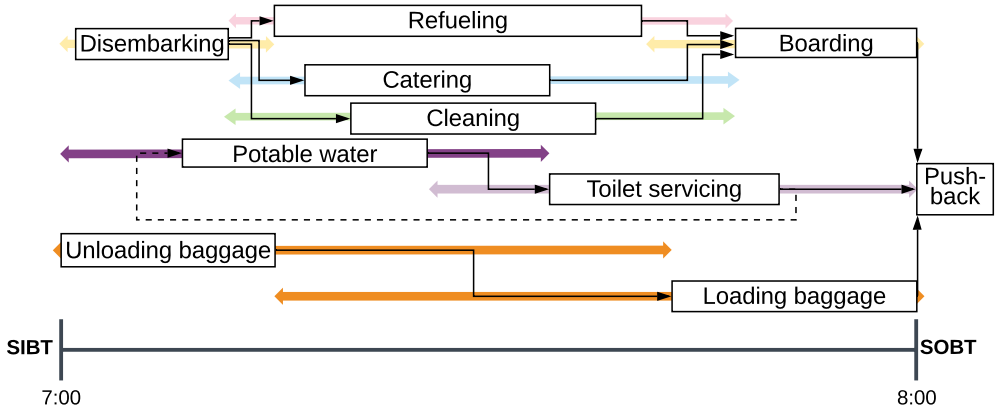

Figure 1: An example of aircraft turnaround operations and precedence relationships between them.

appropriate time windows for all activities may provide for basic albeit essential opportunities to enable viable operations.

Effective scheduling of resources (staff and equipment) in apron operations is crucial. With no loss of generality, in the remainder we focus on teams of staff and assume that each team carries with them the related equipment in between consecutive tasks they are assigned to. Such a scheduling normally happens ahead of the day of operation and covers the whole working shifts of ground service teams. With many factors out of the control of airport decision makers – weather, technical faults or delays of aircraft, late passengers on boarding, etc. – plans that are in some form *robust* should be sought.

Apron operations are often handled by different service providers (SPs), each of which provides a mix of services to a subset of airlines and related flights. An SP consists of teams of different service types. Resource allocation decisions are interconnected and, therefore, the potential for flight delays due to knock-on effects is rather high. Cross-turnaround delay propagation may happen if teams from SPs are scheduled back-to-back between subsequent aircraft. Still busy with a turnaround, a team will also occupy the stand for longer, affecting the next turnaround assigned to the same stand. Delay propagation across airport aprons is inevitable.

In Gök et al. (2020), a robust scheduling approach to the above problem is presented. Therein, authors assume that the airport operator (AO) first sets time windows for all tasks of all SPs, by targeting minimal tardiness across all tasks and flights. Then, each SP routes all teams of all turnaround services they provide in a way that slack times are optimally distributed among operations, so that foreseeable disruptions can be absorbed, at least in most cases. Gök et al. (2020) checked the robustness of the plan using simulation as a post-optimization process. Different than Gök et al. (2020), our contributions in the present work are as follows:

- We add a feedback mechanism from simulation back to optimization, using simulation results to guide the search by learning and adding new constraints to the optimization model. This integration helps guiding the search towards more robust solutions, and extends the use of simulation beyond an evaluation tool, e.g., simply to accept or reject intermediate solutions.
- We define robustness criteria for both the AO and SPs, which are then used to determine if additional simheuristic iterations are needed.

The developed framework falls within the simheuristics family of approaches (Juan et al. 2015) and is described in detail in Section 4.

2 RELATED LITERATURE

In the first subsection of this condensed literature review, we give an overview of methods being used for planning apron operations. Then, we conclude our review with a summary of the state of the art of simheuristic approaches to robust apron operations planning.

2.1 Tactical Robust Planning in Apron Operations

Recently, modeling and solving approaches for airport airside operational problems have been reviewed by Ng et al. (2018). Apron operations are just an example of one type of airside operation, others being taxiing, take-off and landing, air traffic control, and flow management. This review clearly shows that apron operations have not been targeted as often as other operational problems, such as the assignment of boarding gates to flights. The authors also show how problems in this wide category have been modeled after the following classes: Project Scheduling, Quadratic Assignment, Traveling Salesman, or Vehicle Routing Problems (PSP, QAP, TSP, and VRP, respectively). This, in turn, suggests the likely levels of complexity expected from airside operational problems and their combinatorial nature.

Gök et al. (2020) discuss a novel approach to the optimization of the tactical robust problem anticipated in Section 1, which is based on its decomposition into subsequent steps, modeled respectively as forms of PSP, TSP, and VRP. The approach itself will be summarized in Section 3, as it is a precursor to its simheuristic version presented in this work and discussed in Section 4. Padron et al. (2016) present the closest approach, which also breaks down the problem in a similar sequence of two main steps. First, centrally, the AO sets time windows for all apron services to be executed across all flights over the day of operation. Second, decentralized decisions are made by each SP, who by then have developed their staff rosters for the day of operation and can, thus, focus on optimally routing their ground teams through the related turnaround tasks. Other similarities between the two approaches include: *(i)* both papers consider TSP and VRP versions with time windows; *(ii)* both include elements of Constraint Programming (CP) in their approaches; *(iii)* both integrate Large Neighborhood Search (LNS) in their search scheme (Padron et al. 2016 also consider Variable Neighborhood Search); and *(iv)* both are validated against the same case study. Along the way, both Gök et al. (2020) and Padron and Guimarans (2018) – directly based on the development from Padron et al. (2016) – integrate simulation at the bottom-end to evaluate ex-post the robustness of the solutions generated from the heuristic search. Beyond methodological differences, our approach differs from that of Padron et al. (2016) in the following aspects: *(i)* our approach facilitates a more flexible scheduling of turnaround tasks, instead of using predefined configurations and task sequencing for each type of aircraft and turnaround and *(ii)* we fix task time windows at the centralized phase, when the task scheduling is solved, and do not allow for updating these windows at later stages. The former allows for a more efficient usage of resources, since tasks can be flexibly scheduled in the turnaround according to resource availability, while the latter allows for tackling the problem without the need for information sharing between SPs.

Based on a similar problem setting, van Leeuwen and Witteveen (2009) adopt a substantially different perspective, where SPs are considered as separate legal entities making fully independent decisions. Additional related works (Neiman et al. 1994; Kuster and Jannach 2006; Mao et al. 2006; Mao et al. 2008; Norin et al. 2009; Norin et al. 2012; Ip et al. 2013; Andreatta et al. 2014) have less overlap with the work presented in this paper.

2.2 The Role of Simheuristics in Apron Operations Planning

A simheuristic algorithm (Juan et al. 2015) is an approach to integrating simulation and optimization techniques to efficiently tackle a combinatorial optimization problem characterized by stochastic components in either its objective function, constraints, or both, and is particularly suitable if efficient metaheuristics are already available for the problem at hand, as it is the case for apron operations. A thorough review of alternative design schemes for simheuristics tackling a variety of problems is out of the scope of this paper.

In recent years, the Winter Simulation Conference has constantly seen streams dedicated to simheuristics, and the frequency of journal articles has increased as well. The design of the simheuristic approach proposed in this paper was inspired by Guimarans et al. (2015). We integrate simulation with LNS, in the context of the multi-stage approach to our optimization of ground team routes in apron operations. Section 4 provides the details of this integration within the proposed simheuristic approach.

A fair question is whether a different approach to stochastic optimization may be more advisable than simheuristics for the proposed problem. In particular, one of the scenario-based approaches from the literature (e.g., Stochastic Programming), as they allow significant freedom in modeling undesirable events and recourse actions, could be applicable. However, in a recent paper (Markov et al. 2018), where the authors propose a unified framework for routing problems with stochastic demands and uncertainty only in the objective function, they argue that scenario-based approaches are computationally very heavy for an already hard combinatorial problem like theirs. In the case of our apron operational problem, we even see it harder, if not impossible, to think of sufficiently meaningful scenarios in the first place, given the number of tasks involved, the continuous nature of many of the random variables, etc. However, the last five years of development in simheuristics have provided plenty of guidance in developing schemes to integrate certain metaheuristics with simulation. In these schemes, the uncertain elements of the problem are located in the objective function, in the constraints of the problem, or in both. As an example, Grasas et al. (2016) discuss SimILS, an integration of simulation with Iterated Local Search.

Finally, our paper targets two open research lines that were raised by Juan et al. (2018) in a recent review of applications of simheuristics, namely:

1. A higher level of integration between simulation and optimization, including increasing use of the feedback provided by the simulation to better guide the search for better solutions.
2. Use of more 'sophisticated' simulation approaches (e.g., discrete-event simulation or agent-based simulation), to cater for the complexity of real-world applications.

Regarding the former, many simheuristic approaches indeed exploit information from simulation runs mostly to apply an acceptance or rejection criterion that ultimately might flag up the solution just simulated as a 'promising' or an 'elite' solution. As for the latter, Monte Carlo simulation schemes are prevalent. But, the adoption of more-sophisticated stochastic dynamic simulation models might considerably add to the computational burden, ultimately risking to jeopardize the timely performance and, possibly, the viability of the overall simheuristic algorithm. In the remainder of this paper, we show how we successfully dealt with both aspects.

3 PROBLEM DESCRIPTION AND MODELING APPROACH

In this section, we summarize the crucial elements of our apron resource optimization problem and recall how we modeled each of the sub-problems in our decomposition-based approach. Gök et al. (2020) provide details around notation, formulations, role of CP, and use of alternative solvers for all sub-problems.

Flight schedules are known to both the AO and the SPs approximately a season ahead. With this information at hand, scheduling of all turnaround operations for all days of the planning horizon may commence. From a centralized viewpoint, the AO would normally want to be in the best possible conditions for achieving minimal delays, measured on turnaround completion (SOBT). This long-term planning phase concludes with the AO setting time windows, where each turnaround service should take place. Modeling-wise, the AO solves a PSP with infinite resource capacity first, optimizing for minimal tardiness across all daily operations at the airport. Then, this solution is further improved, achieving minimal resource requirements across all apron operations and SPs. This second sub-problem takes the well-known form of a 'resource-constrained PSP' (RCPSP). As a by-product of this problem-solving step, the AO is able to communicate to the SPs the overall likely levels of tardiness, for both the airport as a whole and each of them separately, as well as the resource levels (number of teams per SP and service

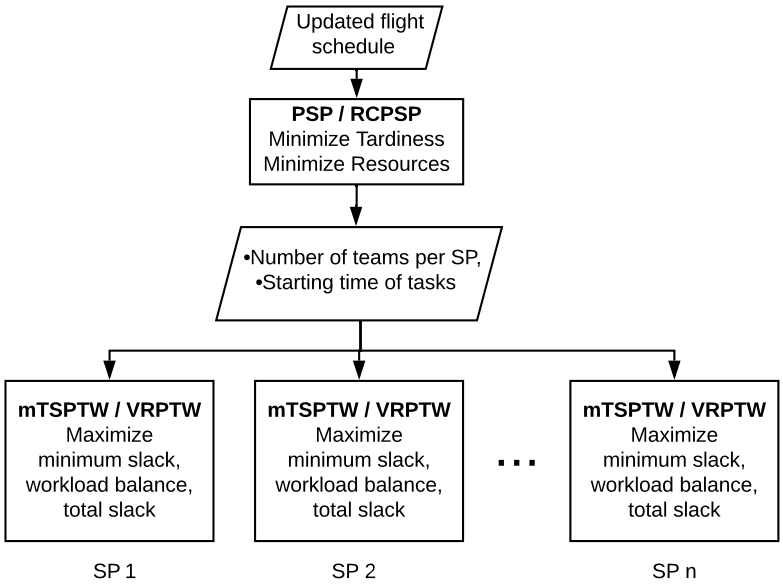

Figure 2: Proposed turnaround planning process (Gök et al. 2020).

type) required from them. This is precious information for the SPs to take forward to their own specific rostering optimization decision-making processes, which are out of our work's scope.

Approximately one week before the day of operation (top of Figure 2), the PSP/RCPSP sequence is repeated with the most up-to-date flight schedule as input. As a result, estimates of both time windows and resource requirements for each service and for each day are refined, and each SP can proceed with its normal short-term planning adjustments including any changes needed to its rosters.

At this stage, planning takes a more decentralized view (bottom of Figure 2), with each SP solving independent problems. Their own ground teams, separately for each type of provided service, are optimally routed throughout the related turnaround tasks and time windows. Unlike more typical optimal routing problems, and given the short distances between airport gates, minimizing distances traveled per team is not as important as optimally allocating slack time for teams in between consecutive tasks. The reason is that we ultimately aim to enforce a certain degree of robustness to the plan in spite of a bunch of uncertain aspects of the problem – with slack time enabling absorption of at least minor disruptions and related delays. To achieve this, we first maximize (separately for each SP and each related service) the minimum slack across all teams available for a shift. Then, we balance the workload across teams as much as possible, aiming to distribute the workload as fairly as possible. Finally, we maximize the total slack time. Certain turnaround services are provided by teams using vehicles and equipment of limited capacity (e.g., refueling trucks or catering trucks), while others do not share this characteristic (e.g., push-back and cleaning trucks). From a modeling standpoint, our routing problems in relation to the former type of resource take a form of vehicle routing problem with time windows (VRPTW), while the latter become a form of multi-traveling salesmen problem with time windows (mTSPTW) (Desrosiers et al. 1995).

4 SOLUTION APPROACH

We propose a simheuristic framework for solving the airport turnaround scheduling problem (Figure 3). First, we seek to maximize the minimum and total allocation of slack to the routes of the different teams (Z_1 and Z_3 in Figure 3, respectively) and balance the workload across all teams as much as possible (Z_2). These objectives are achieved in the deterministic version of the underlying combinatorial problem. Then, our implementation of LNS seeks to further improve the maximum total slack objective (Z_3). After reaching

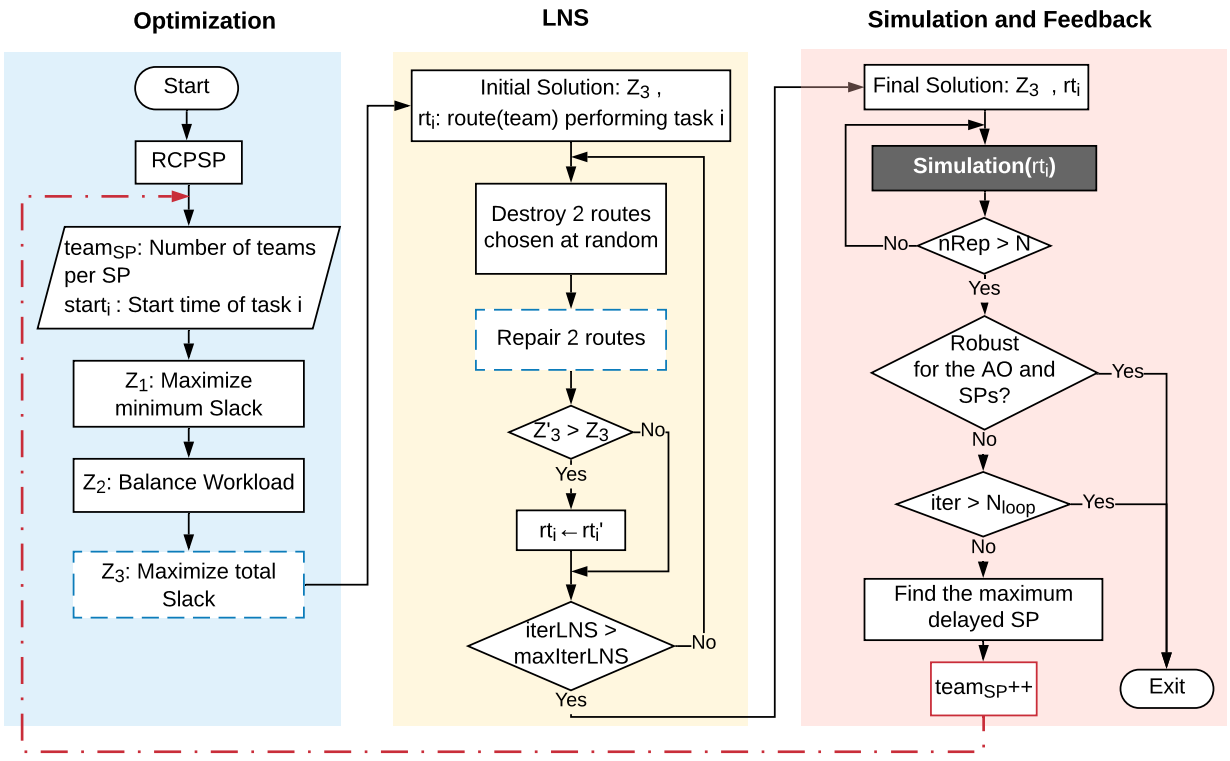

Figure 3: Simheuristic framework.

a maximum number of iterations (*maxIterLNS*), we use a discrete-event simulation model of the airport turnaround operations for evaluating the robustness of the deterministic solution in the stochastic setting. If the robustness criterion is not met, this inference is fed back to the optimization step to be considered in a new iteration of the whole procedure.

In the following subsections, we provide relevant details about the role of LNS in the solution of the optimization problem (Section 4.1) and further details about our simheuristic algorithm (Section 4.2).

4.1 Integration of Large Neighborhood Search

All our sub-problems (left-hand side of Figure 3) were formulated using CP and implemented in Minizinc, a solver-independent language to model both constraint satisfaction and optimization problems. We have tried different solvers, including CP solvers (Gecode, Chuffed) and Mixed-Integer Programming (MIP) solvers (Gurobi). Solving the majority of instances to optimality and in computationally reasonable times proved particularly challenging, especially in maximizing for total slack (Z_3 in Figure 3). Then, we chose to apply a time limit to the execution of the CP solver for this last step in the optimization routine, and take the best solution obtained up to that stage as a starting point for an LNS approach to improve further. LNS is a metaheuristic framework introduced by Shaw (1998) and widely used in combination with CP with promising results (Guimarans et al. 2015). We show the steps of our own version of LNS for our problem in the middle part of Figure 3. Essentially, we take the solution from objective Z_3 and apply a 'destroy' operator to it, removing a number of team-to-turnaround assignments. Then, we feed this 'incomplete' solution to the maximization of the total slack sub-problem, where the CP solver will 'repair' the solution by making sure that all tasks of all types are allocated to a related team. Solutions are repaired effectively due to dealing with a considerably smaller search space. On examining this new solution, it may turn out that it is the new 'best-so-far'. We repeat this destroy-repair-evaluate loop a number of times (*maxIterLNS*),

before we conclude the LNS phase. At this stage, we have an allocation of the available ground teams to turnaround tasks at our disposal, which we believe embeds a level of slack that should render it 'robust'. It should enable to absorb many of the foreseeable delays and disruptions that are likely to take place on the day of operation once the plan is enacted. To what extent that happens is usually not known until at least an *a posteriori* simulation of the enactment of the plan under a significant coverage of scenarios is run. This has been conducted by Gök et al. (2020), where we also provided details of all optimization sub-problems as well as of our LNS implementation.

An interesting aspect of our problem is the need for adopting appropriate robustness metrics to be computed by simulation into the optimization. This requirement is common for many planning and scheduling problems (see, e.g., Jamili 2016 for a review and discussion of robustness metrics in job-shop scheduling). The dual centralized-decentralized nature of our problem translates quite easily in radically different notions of robustness, depending on the perspective (AO or SP). In the next subsection, we discuss how we dealt with this dual nature in our simheuristic approach. Furthermore, the core novelty in this paper and the very focus of the next subsection relate to the specific way in which we added simulation to the decision framework (right-hand side of Figure 3). The idea is to use the inference from simulation results to give feedback to the optimization sub-problems to improve operational robustness.

4.2 The Simheuristic Algorithm: Simulation 'in the Loop'

In our problem, the arrival time of all aircraft, the processing time of all turnaround tasks, the time for teams to travel between consecutive turnarounds, and the time to replenish any finite capacity resource (e.g., the fuel truck), are all major examples of stochastic components that need to be considered for robustness of the plans. In our approach, we accounted for all these aspects by developing our own Petri-Net-based discrete-event simulator. Its major features are its generality, adaptability, easiness of integration, and computational lightness. Following one iteration of the deterministic Optimization and LNS steps (Figure 3), the algorithm automatically builds an enhanced Petri Net model of the system based on the obtained solution, which is then simulated exploiting recursivity. This solution permits a fine-grained tailoring of the simulation outputs for further analysis, as well as a reduction of the development time required to implement and maintain such a complex model in commercially available simulation tools. Even though some of these tools have automatic model generation capabilities (such as Anylogic), they are dependent on the use of professional licenses. Our approach, on the other hand, is independent of any specific tool. Furthermore, our simulator also enhances the approaches' modularity and flexibility, since the simulation model can be automatically built independently of the solving approach, and can accommodate new operational constraints and adapt operational rules without requiring any adjustment before or during the algorithm's execution.

Following one iteration of the simheuristic algorithm and the analysis of the robustness of operations of both the airport as a whole and of each SP, changes in the 'system configuration' (in our case, resources available to each SP and service type) are given back to the first sub-problem of the optimization stage (i.e.,, the maximization of minimum slack). After that, the optimization and LNS stages proceed as explained above, giving way to a new run of the simulation stage. Resource availability and all other settings for our simulator are automatically provided at the end of the LNS stage, so that a simulation of this configuration can be run for a suitable number of independent replications. This gives rise to new estimates for the robustness metrics, and the whole process repeats with a cap to the maximum number of iterations. Our simulator is then fully and automatically integrated within the now 'triple-stage' Optimization-LNS-Simulation loop.

At the conclusion of every iteration of the loop, we check two robustness metrics:

- Assuming the centralized view from the AO, we compute the percentage of departing flights for which the push-back starts later than a certain threshold (td) after the SOBT. At the planning stage, there is normally a maximum percentage d that the AO is willing to accept.
- Taking a more decentralized viewpoint, each SP is interested in meeting their Service Level Agreements (SLAs) with the airlines they service and the AO. SLAs normally include many

potentially alternative performance metrics, of which we selected one as an example. We assume SPs to be interested in keeping delays in starting the related turnaround tasks below a maximum acceptable threshold of tt minutes. For the sake of simplicity and with no loss of generality, in our computational experiments we assume this threshold to be equal for all SPs at the airport.

At each iteration of the simheuristic algorithm, N_i independent replications of the simulation model are run. The number of replications depends on the specific instance i, the related underlying variability, etc., and is computed with conventional methods (Law 2014). Immediately following simulation, both of the robustness metrics are checked. If both of the metrics are below the chosen threshold, the overall procedure stops. If either of them does not meet the required level, then the service type and SP with the highest total delay is selected and this information communicated back to the optimization stage, where resource capacity for that particular service is increased by one unit, and the triple-stage loop may restart. Note that the newly added team should in fact be 'off' on that day of operations. This normally comes at a costs for the SP, but would hopefully be counterbalanced by the avoidance of costs from the related delays that the SP may have to pay for otherwise. The whole procedure is repeated until either both required levels are met, or a set cap of N_{loop} iterations of the loop are performed. In the next section, we show how this feedback mechanism can be effectively used to enhance the 'planned' robustness of apron operations on the whole.

5 EXPERIMENTS

Our proposed simheuristic methodology was tested on four generated instances, which were built based on an original data set from service providers on one of the major airports in Spain, with different mixes of aircraft and frequencies of arrivals and departures for one day of operations. All tasks are assumed to belong to a single shift. There are seven SPs in total, each one responsible for different types of tasks: baggage loading and unloading, cleaning, catering, fueling, clean water service, toilet service, and push-back. If there is more than one shift a day, the solution approach is decomposed to multiple shifts and solved separately until the simulation phase. Simulation is run for the whole day of operations.

We ran our approach on a personal laptop (1.6 GHz Intel Core i5) running mac-OS High Sierra. The overall integration was achieved in Python 3.7, calling MiniZinc Python (MiniZinc version 2.3.2) to run the optimization and our own java-based simulator for the discrete-event simulation runs. The number of independent simulation replications is calculated with a 95 % confidence level, and an absolute error of seven minutes for low variability and 15 minutes for high variability.

At the optimization step, the RCPSP is proven to be optimal for minimizing tardiness, as well as for minimizing total resources and teams per SP for all instances. For the mTSPTW/VRPTW, we proved optimality for the minimum slack time for the given tardiness and resource levels. The optimization of the total slack is terminated with a given time limit, and the resulting solution is accepted as an initial solution for the LNS. The average optimality gap was 0.68 % across all instances, with a maximum observed gap of 2.72 %. Hence, the performance of our optimization approach is promising, especially considering the high number of tasks involved, ranging from 228 for the smallest instance to 966 for the largest one.

Table 1 shows the computation times (*CPU*) per instance and per level of variance. Even though computation times are high, especially for the largest instance, it is not our main concern, since there will be enough time to run this algorithm one week before the day of operations when the changes in the flight schedules are finalized. This time frame also provides sufficient time for SPs to arrange additional teams if necessary. We also observe that simulation constitutes a computationally expensive portion of the overall simheuristic algorithm (*Sim. CPU*).

For some instances, we require a high number of iterations to reduce delays, especially for scenarios with high variability. In our experiments, we stop after 20 iterations (N_{loop}) if the simheuristic approach is not able to reach robustness both for the AO and SP metrics. However, our results indicate that ten iterations are sufficient to reach solutions close enough to robustness for at least one of the stakeholders

Table 1: Results from our approach for the four generated instances and different levels of variability.

Instance	# Tasks	Time Horizon (min)	Variability	# Sim. replications	# Iterations	CPU (s)	Sim. CPU (%)
ta24_t120	228	120	low	31	3	184.77	42.53
			high	201	20	5981.61	48.22
ta48_t240	465	240	low	29	5	2180.08	37.43
			high	118	20	17487.71	69.36
ta72_t360	692	360	low	55	1	1740.11	50.55
			high	35	20	29121.24	40.83
ta96_t480	966	480	low	48	8	23507.93	63.62
			high	66	20	67157.31	75.78

as to facilitate the decision-making process (Figure 4). Hence, the computation times could be improved drastically by cutting the number of simheuristic iterations to half. We deliberately decided to keep a maximum of 20 iterations for this study for clarity of interpretation of the results.

As described in Section 4.2, we determine the robustness of a solution based on threshold values defined for performance indicators for the AO (d, dependent on td) and the SPs (tt). Following real-world practices, we set td to 5 minutes of delay and 10 % for d. For the SPs, we set tt as 10 minutes of total delay. As discussed, for simplicity and without loss of generality, we assume the same tt is applicable to all SPs in our experiments. If SLAs are not met, then SPs might face penalties payable to the airlines, and the AO's reputation might suffer due to having a high percentage of delays, which in turn can have an impact as increased costs.

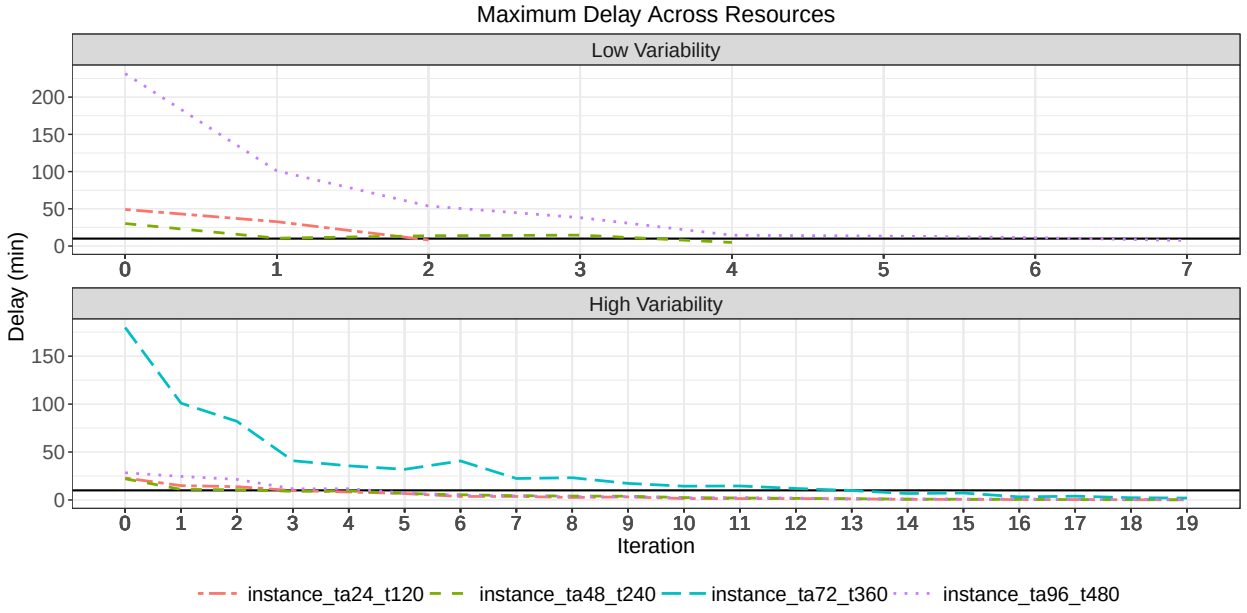

Figure 4: Evolution of the maximum delay across SPs per instance and for two levels of variability.

Figure 4 shows the evolution of the maximum delay across all resources for scenarios with low (top) and high (bottom) variability. The solid line indicates the threshold value (tt) under which robustness is achieved. i.e., where all SPs are able to schedule their resources with a maximum delay never exceeding tt. In each iteration, a new team is added for the operation showing the highest delays. For low variability cases, this is achieved quite early in the execution of our simheuristic approach. Except for ta72_t360, a fast convergence for this metric is also achieved for scenarios with high variability.

In some cases, we observe a little increase in the delay values after an iteration. This is due to dependencies among tasks, which could lead to the improvement of one SP actually worsening the performance of another one, and hence to an increase of the maximum delay in the system. This relation between different SPs is illustrated in Figure 5. In Case 1, we observe that both teams arrive ahead of time and their respective tasks start according to the schedule. In Case 2, although Team B arrives later than the scheduled time, it still needs to wait before starting Task *b* due to the late arrival of Team A and the corresponding delay of Task *a*, assuming there is a precedence relationship between both tasks. It is worth noting that the delay in Task *a* will be linked to the late arrival of Team A, but the actual delay of Task *b* is not caused by Team B. In this case, the algorithm will focus on improving the performance for Team A (Case 3), revealing the real issue caused by the late arrival of Team B. This will increase the total delay associated to Team B and, therefore, its performance might decline due to the improvement of Team A's schedule.

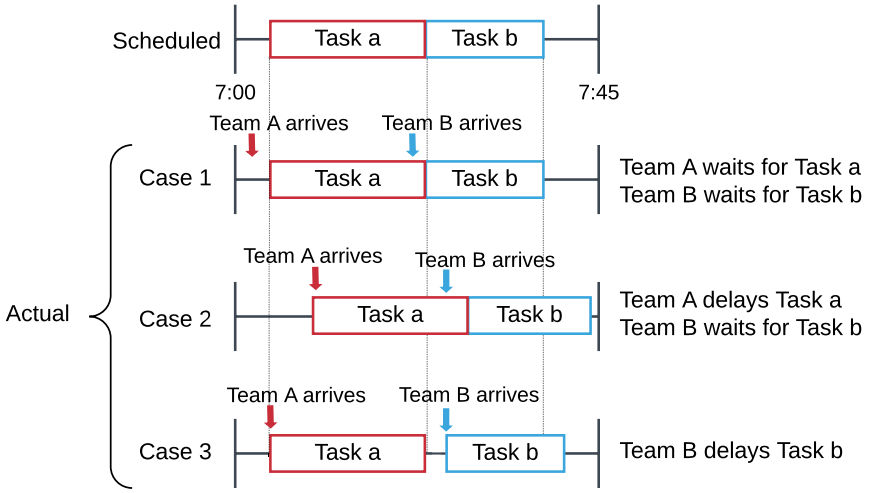

Figure 5: Example of propagation of delays between two teams performing dependent tasks.

Figure 6 shows the evolution of the percentage of flights delayed over five minutes (d) for scenarios with low (top) and high (bottom) variability. In both cases, we set the robustness thresholds at 10 %. We observe that this is achieved after one iteration for all instances in scenarios with low variability. The remaining iterations are required to achieve robustness from the SPs' perspective (c.f. Figure 4 top). However, we observe that the desired robustness level is never achieved after 20 iterations for high variability scenarios, despite reaching robustness from the SPs' perspective (c.f. Figure 4 bottom). This is due to the high variability of the task duration, which might cause departure delays even when all teams are ready on time. Nonetheless, if we consider the total aircraft delay instead of d, we observe a clear decrease across iterations for scenarios with high variability, with the delay remaining between four and six minutes for most aircraft. These delays are unavoidable, unless the duration of tasks or the scheduled turnaround times are extended to absorb such variability. This is a strategic level adjustment which needs to be agreed with the airline in the first place. Hence, it is not in the scope of our study. However, some techniques proposed by Davenport et al. (2001) could be implemented in the future in order to avoid delays due to high processing time variability.

6 CONCLUSION

In this study, we propose a simheuristic approach for tackling the scheduling of airport turnaround services and teams. Different from the majority of simheuristic approaches, we more seamlessly integrate discrete-event simulation with the metaheuristic execution. Our use of simulation is two-fold: *(i)* to evaluate the robustness of solutions from the perspectives of different stakeholders (AO and SPs), and *(ii)* to provide

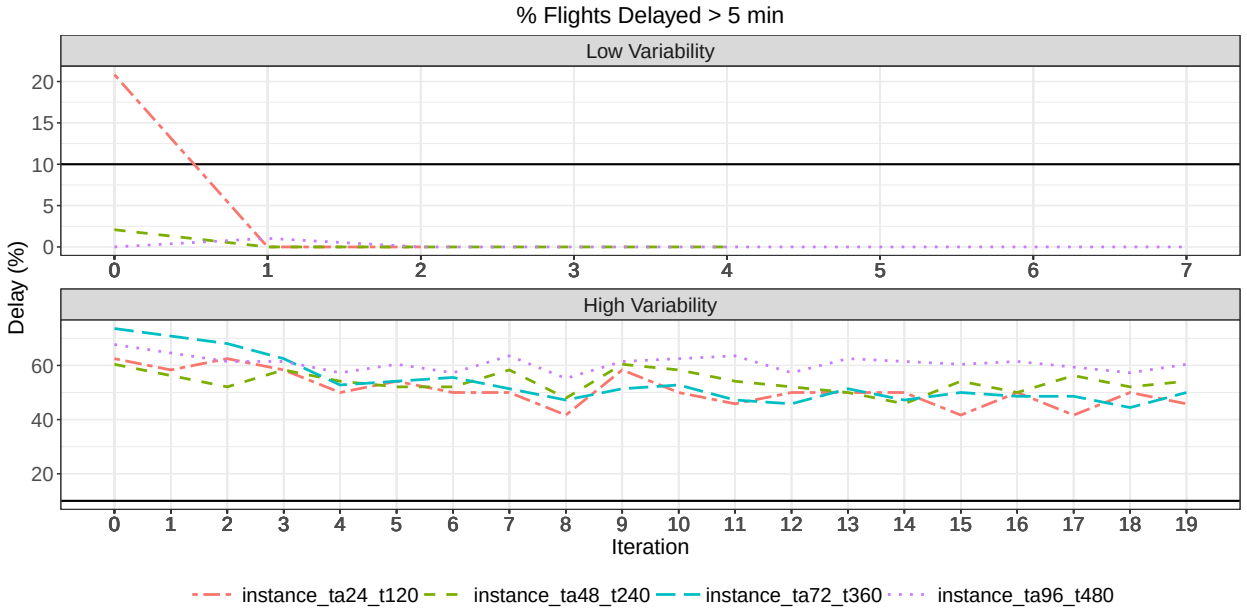

Figure 6: Evolution of the percentage of delayed aircraft for two levels of variability.

additional guidance to the optimization process by increasing the number of resources based on the automated analysis of simulation results. Our experiments prove that this approach provides a suitable decision-making support platform for multiple stakeholders. As a future line of research, this approach could be extended to include additional performance and robustness metrics to accommodate stakeholders' needs, as well as the mechanisms to provide diverse trade-off solutions to facilitate the decision-making process. Finally, the simheuristic framework could be enhanced by the generation of additional constraints derived from the analysis of simulation results.

REFERENCES

Andreatta, G., L. Capanna, L. De Giovanni, M. Monaci, and L. Righi. 2014. "Efficiency and Robustness in a Support Platform for Intelligent Airport Ground Handling". *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations* 18(1):121–130.

Davenport, A., C. Gefflot, and C. Beck. 2001. "Slack-based Techniques for Robust Schedules". In *Proceedings of the Sixth European Conference on Planning*, edited by A. Cesta and D. Borrajo. September 12th–14th, Toledo, Spain: The Association for the Advancement of Artificial Intelligence Press.

Desrosiers, J., Y. Dumas, M. M. Solomon, and F. Soumis. 1995. "Time Constrained Routing and Scheduling". In *Network Routing*, edited by M. Ball, T. Magnanti, C. Monma, and G. Nemhauser, Volume 8 of *Handbooks in Operations Research and Management Science*, Chapter 2, 35–139. Amsterdam: Elsevier.

Gök, Y. S., D. Guimarans, P. J. Stuckey, M. Tomasella, and C. Ozturk. 2020. "Robust Resource Planning for Aircraft Ground Operations". In *Proceedings of the 2020 International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, edited by E. Hebrard and N. Musliu. Berlin: Springer, forthcoming.

Grasas, A., A. A. Juan, and H. R. Lourenço. 2016. "SimILS: A Simulation-based Extension of the Iterated Local Search Metaheuristic for Stochastic Combinatorial Optimization". *Journal of Simulation* 10(1):69–77.

Guimarans, D., P. Arias, and M. M. Mota. 2015. "Large Neighbourhood Search and Simulation for Disruption Management in the Airline Industry". In *Applied Simulation and Optimization*, edited by M. Mujica Mota, I. Flores De La Mota, and D. Guimarans Serrano, 169–201. Berlin: Springer.

Ip, W. H., D. Wang, and V. Cho. 2013. "Aircraft Ground Service Scheduling Problems and their Genetic Algorithm with Hybrid Assignment and Sequence Encoding Scheme". *IEEE Systems Journal* 7(4):649–657.

Jamili, A. 2016. "Robust Job Shop Scheduling Problem: Mathematical Models, Exact and Heuristic Algorithms". *Expert Systems with Applications* 55:341–350.

Juan, A. A., J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. 2015. "A Review of Simheuristics: Extending Metaheuristics to Deal with Stochastic Combinatorial Optimization Problems". *Operations Research Perspectives* 2:62–72.

Juan, A. A., W. D. Kelton, C. S. Currie, and J. Faulin. 2018. "Simheuristics Applications: Dealing with Uncertainty in Logistics, Transportation, and other Supply Chain Areas". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3048–3059. Piscataway, New Jersey: IEEE.

Kuster, J., and D. Jannach. 2006. "Handling Airport Ground Processes Based on Resource-Constrained Project Scheduling". In *Proceedings of the 2006 Advances in Applied Artificial Intelligence*, edited by M. Ali and D. Dapoigny, 166–176. Berlin: Springer.

Law, A. M. 2014. *Simulation Modeling and Analysis*. 5th ed. New York: McGraw-Hill Education.

Mao, X., N. Roos, and A. Salden. 2008. "Distribute the Selfish Ambitions". In *Proceedings of the 20th Belgian-Netherlands Conference on Artificial Intelligence*, edited by A. Nijholt, M. Pantic, M. Poel, and H. Hondorp. October 30th–31st, Enschede, Netherlands, Belgian-Dutch Association for Artificial Intelligence (BNVKI), 137–144.

Mao, X., A. Ter Mors, N. Roos, and C. Witteveen. 2006. "Agent-based Scheduling for Aircraft Deicing". In *Proceedings of the 18th Belgium-Netherlands Conference on Artificial Intelligence*, edited by P. Schobbens, W. Vanhoof, and G. Schwanen. October 6th–7st, Namur, Belgium, Belgian-Dutch Association for Artificial Intelligence (BNVKI), 229–236.

Markov, I., M. Bierlaire, J.-F. Cordeau, Y. Maknoon, and S. Varone. 2018. "A Unified Framework for Rich Routing Problems with Stochastic Demands". *Transportation Research Part B: Methodological* 114:213–240.

Neiman, D. E., D. W. Hildum, V. R. Lesser, and T. W. Sandholm. 1994. "Exploiting Meta-level Information in a Distributed Scheduling System". In *Proceedings of the 1994 National Conference on Artificial Intelligence*, Volume 1, 394–400. Palo Alto, CA: The Association for the Advancement of Artificial Intelligence Press.

Ng, K., C. Lee, F. T. Chan, and Y. Lv. 2018. "Review on Meta-heuristics Approaches for Airside Operation Research". *Applied Soft Computing* 66:104–133.

Norin, A., T. A. Granberg, P. Värbrand, and D. Yuan. 2009. "Integrating Optimization and Simulation to Gain More Efficient Airport Logistics". In *Proceedings of the Eighth USA/Europe Air Traffic Management Research and Development Seminar*, edited by S. Saunders-Hodge and V. Duong, 167–176. Brussels: The European Organisation for Safety of Air Navigation (Eurocontrol).

Norin, A., D. Yuan, T. A. Granberg, and P. Värbrand. 2012. "Scheduling De-icing Vehicles within Airport Logistics: A Heuristic Algorithm and Performance Evaluation". *Journal of the Operational Research Society* 63(8):1116–1125.

Padron, S., and D. Guimarans. 2018. "Using Simulation for Evaluating Ground Handling Solutions Reliability under Stochastic Conditions". In *Proceedings of the 2018 ROADEF*. February 21st–23rd, Lorient, France, French Society for Operational Research and Decision Support.

Padron, S., D. Guimarans, J. J. Ramos, and S. Fitouri-Trabelsi. 2016. "A Bi-objective Approach for Scheduling Ground-handling Vehicles in Airports". *Computers & Operations Research* 71:34–53.

Shaw, P. 1998. "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems". In *Proceedings of the 1998 Principles and Practice of Constraint Programming — CP98*, edited by M. Maher and J.-F. Puget, 417–431. Berlin: Springer.

van Leeuwen, P., and C. Witteveen. 2009. "Temporal Decoupling and Determining Resource Needs of Autonomous Agents in the Airport Turnaround Process". In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, edited by S. Marrara, Volume 2, 185–192. Piscataway, New Jersey: IEEE.

AUTHOR BIOGRAPHIES

YAGMUR SIMGE GÖK is a Doctoral Student at the University of Edinburgh Business School. Yagmur's work focuses on the development of hybrid simulation-optimization techniques applied to airport ground operations. Her email address is Yagmur.Gok@ed.ac.uk.

MAURIZIO TOMASELLA is a Lecturer in Management Science at the University of Edinburgh Business School. Previously at the University of Cambridge Engineering Department, his research interests include simulation, multi-criteria decision making, and their applications to airport operations as well as manufacturing. His e-mail address is maurizio.tomasella@ed.ac.uk.

DANIEL GUIMARANS is a Senior Research Scientist at Amazon Transportation Services. His main research interest lies on the hybridization of optimization methods with simulation techniques, with applications in areas such as logistics, manufacturing, and aviation. His email address is guimd@amazon.com.

CEMALETTIN OZTURK is Staff Research Scientist at Raytheon Technologies. Cemalettin is enthusiastic about applications of combinatorial optimisation and simulation in various domains. His email address is OzturkC@rtx.com.