Journal of the Operational Research Society

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjor20

A multi-fidelity modelling approach for airline disruption management using simulation

L. A. Rhodes-Leader, B. L. Nelson, B. S. Onggo & D. J. Worthington

To cite this article: L. A. Rhodes-Leader, B. L. Nelson, B. S. Onggo & D. J. Worthington (2021): A multi-fidelity modelling approach for airline disruption management using simulation, Journal of the Operational Research Society, DOI: 10.1080/01605682.2021.1971574

To link to this article: https://doi.org/10.1080/01605682.2021.1971574

Published online: 14 Sep 2021.

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY
https://doi.org/10.1080/01605682.2021.1971574

THE OPERATIONAL RESEARCH SOCIETY

Taylor & Francis
Taylor & Francis Group

ORIGINAL ARTICLE

OPEN ACCESS

Check for updates

A multi-fidelity modelling approach for airline disruption management using simulation

L. A. Rhodes-Leader[a], B. L. Nelson[a], B. S. Onggo[b] and D. J. Worthington[a]

[a]STOR-i CDT, Lancaster University, Lancaster, UK; [b]Southampton University, Southampton, UK

ABSTRACT
Disruption is a serious and common problem for the airline industry. High utilisation of aircraft and airport resources mean that disruptive events can have large knock-on effects for the rest of the schedule. The airline must rearrange their schedule to reduce the impact. The focus in this paper is on the Aircraft Recovery Problem. The complexity and uncertainty involved in the industry makes this a difficult problem to solve. Many deterministic modelling approaches have been proposed, but these struggle to handle the inherent variability in the problem. This paper proposes a multi-fidelity modelling framework, enabling uncertain elements of the environment to be included within the decision making process. We combine a deterministic integer program to find initial solutions and a novel simulation optimisation procedure to improve these solutions. This allows the solutions to be evaluated whilst accounting for the uncertainty of the problem. The empirical evaluation suggests that the combination consistently finds good rescheduling options.

ARTICLE HISTORY
Received 30 June 2020
Accepted 16 August 2021

KEYWORDS
Simulation; Optimisation; Integer Programming; Multi-objective; Transport

1. Introduction

Disruption to airline schedules creates major issues within the aviation industry. During planning stages, airlines use optimisation algorithms to produce schedules maximising profit with high utilisation of their resources. Developments in this field are enabling schedules with higher utilisation of resources whilst including some slack to recover from minor disruptions. However, due to complexity and high levels of uncertainty in the industry, it is rare that flight programmes operate as initially planned. The Federal Aviation Administration estimated that the total cost of delays in the U.S.A. were $33 billion in 2019 (Federal Aviation Administration, 2020). In the same year, only 77.6% of flights in Europe arrived within 15 min of the schedule, with 4.3% of flights experiencing departure delays exceeding 1 h (EUROCONTROL, 2020). Disruptive events have various roots which Bratu and Barnhart (2006) classify as either shortages in airline resources (such as aircraft requiring unscheduled maintenance) or shortages in airport resources (such as reductions in aircraft movements due to weather conditions). In addition, an airline's policy of waiting for connecting passengers can also cause delays. The impacts of such events can propagate through the system causing further delays and cancellations, particularly when the airline has high aircraft utilisation and airport resources operate at or near full capacity.

When a disruption occurs, the Operations Control Centre (OCC) of the airline must propose alterations to its schedule to Air Traffic Control (ATC) for their approval, bearing in mind the repercussions on finances, reputation and passengers. ATC has the right to reject proposals, in which case OCC needs to be able to suggest "good but different" solutions. The OCC wants to identify good actions and evaluate them from the perspective of the airline and the passengers. Kohl et al. (2007) describe the practice of the OCC during disrupted operations, stating that the problem is often split into three: aircraft, crew and passengers. This paper focusses on the Aircraft Recovery Problem (ARP). The available options for the ARP include delaying or cancelling flights, or exchanging aircraft.

The financial costs of a disruption can be significant. However, there are other concerns for the airline too. The airline will not want to incur high levels of delay, as this can lead to discontented passengers and affect reputation. Neither will it want to make too many changes to its schedule. Rerouting aircraft can interfere with crew and future fleet maintenance schedules and gate assignments (Thengvall et al., 2000). As the ARP solution must be used in conjunction with the crew recovery solution, minimising schedule alterations helps the integration. Whilst both considerations do have financial

CONTACT L. A. Rhodes-Leader l.rhodes-leader@lancaster.ac.uk

Supplemental data for this article is available online at http://doi:10.1080/01605682.2021.1971574.

consequences, these can be difficult to quantify, making this a multi-objective problem.

The complexity of the industry can make it difficult to determine the consequences of schedule alterations. Operational research techniques can be of use here, helping to search for good solutions (Kohl et al., 2007). Some of the complexity can be represented using deterministic models, a number of which have been proposed (Clausen et al., 2010). However, these models cannot fully account for the various uncertain elements of the environment, such as runway queueing times, which can affect the performance and viability of solutions. This raises the question of how the inherent variability of airline operations can be included within the search for the best recovery action, which is the focus of this paper.

A paradigm for modelling the uncertainty is stochastic simulation. In a simulation model, both complexity and uncertainty are explicitly represented. However, simulation does not provide a natural strategy for searching through the possible revised schedules. This search must cope with combinatorial constraints, a difficulty for simulation optimisation. One potential approach is to consider multi-fidelity modelling, combining multiple modelling paradigms of varying accuracy, each used to gain an insight into the problem.

This paper presents a multi-fidelity modelling framework for the ARP, which is designed to address key features of the real problem. Overall it recognises and takes advantage of the multi-objective nature of the problem to generate a range of solutions. Each of these solutions is found by combining the abilities of:

1. Integer Programming (IP) to efficiently find good solutions to a highly constrained deterministic problem, allowing the discrete reassignment aircraft allocation problem to be solved within appropriate time scales;
2. simulation to evaluate the performance of complex stochastic systems; and
3. a simulation optimisation algorithm designed to improve the solutions from the IP.

In our context, we take the 'low-fidelity' model to mean the IP, as it ignores uncertainty and works in discrete time. The simulation will be referred to as a 'high-fidelity model', as it aims to represent much more of the real system than the IP does.

The primary contribution of this paper is this framework, allowing stochastic information from the simulation to be used within the optimisation. To demonstrate the feasibility and potential benefits of the multi-fidelity modelling framework, the proposed approach is applied to three realistic (but not real) test cases which are designed to reflect the main features of this problem, including realistic flight schedules and disruptions, as well as the multi-criteria nature of the decision making process. Data used in these test cases are given in the Supplementary Material, including sources where available in case other researchers wish to apply other modelling frameworks. Early proof-of-concept work was presented in Rhodes-Leader et al. (2018).

The remainder of this paper is structured as follows. Section 2 discusses related work in airline disruption management and multi-fidelity modelling. The formal problem statement and method overview are in Section 3. Section 4 describes the low-fidelity IP model and how it is used to find "good but different" solutions to the deterministic problem. Section 5 describes the high-fidelity simulation model, and how simulation optimisation is used to improve upon the IP generated solutions. This includes how we extended the STRONG algorithm for simulation optimisation to deal with the constraints of the ARP. Computational experiments and results are discussed in Section 6, followed by conclusions in Section 7.

2. Related work

Airline disruption management has received much attention within the literature, with many deterministic models and solution methods proposed. Clausen et al. (2010) review the area. Rosenberger et al. (2003) use a set-packing IP formulation along with a problem reduction heuristic for the ARP. Recently, advances in solvers have allowed more sophisticated exact solution methods, such as column and row generation. These have been applied in both the ARP (Liang et al., 2018) and the combined aircraft and crew problem (Maher, 2016). Heuristic approaches include Steepest Ascent Local Search (Løve et al., 2005), Genetic Algorithms (Jeng, 2012) and Large Neighbourhood Searches (Sinclair et al., 2014).

One class of models reported in the ARP literature is time-space networks. These consider schedules as networks where flights are arcs between nodes representing airports at a given time. Examples of the use of these models include Thengvall et al. (2000) for the ARP, Bratu and Barnhart (2006) (aircraft and passenger) and Zhang et al. (2015) (aircraft and crew).

As with many real problems, the ARP has a number of performance measures of interest including costs, passenger delays and resuming normal operations quickly (Clausen et al., 2010). Thengvall et al. (2000) note the trade-off between maximum revenue and minimal schedule alterations, though not explicitly including these in the model. Hu et al.

(2017) propose a full multi-objective formulation, trying to identify Pareto optimal solutions, with objectives considering disruption cost, number of aircraft rerouted and maximum delay. The authors use a local neighbourhood search heuristic combined with an ϵ-constraint method.

Rosenberger et al. (2003) conclude that a robust solution accounting for uncertainty may perform better than deterministically optimal solutions. Zhu et al. (2015) incorporate some uncertainty, assuming the duration of unplanned maintenance is unknown. They propose a two-stage recourse stochastic program, the first allocating aircraft to rotations, then retiming flights. The scenarios are the ready times of aircraft requiring unplanned maintenance. However, including scenarios with other uncertain elements would greatly increase the problem size, making it computationally intractable when decisions must be made quickly.

Rosenberger et al. (2000) develop a stochastic discrete event simulation model of airline operations known as SimAir to simulate the schedule over long periods (weeks or months). This is extended by Rosenberger et al. (2002) and Lee et al. (2003). The purpose of SimAir lies in the schedule planning stage, providing a method to evaluate recovery policies on a long term schedule and monitor its robustness. There is also scope to use simulation at the operational level. A simulation to perform "what if" analysis could be valuable to the OCC (Kohl et al., 2007). Abdelghany et al. (2008) propose a network simulation to predict which flights will be disrupted by considering future aircraft and crew connections. Only these flights are included in an IP model for the aircraft and crew recovery problem. Hutchison and Hill (2001) propose a Simultaneous Perturbation Stochastic Approximation simulation optimisation process to change the delays on flights of Ground Delay Programmes under poor weather conditions. This did not involve reallocating aircraft.

Stochastic turn times and flight durations within the ARP are considered by Arias et al. (2013), combining a deterministic constraint programming approach with a Monte Carlo simulation to try to minimise delay. The simulation evaluates the constraint program solution under several scenarios by adding normal noise to the turn times and flight durations. If the solution is not robust across these scenarios, it is rejected and the process begins again. Guimarans et al. (2015) expand this approach by combining the constraint program with a Large Neighbourhood Search to propose new solutions and use the simulation at each proposed solution to evaluate the acceptance criteria. However, the simulation remains rudimentary, and using a higher fidelity model would harm the performance of the algorithm if the simulation is used at every iteration. Of the current literature, our proposal most closely resembles the approaches of Arias et al. (2013) and Guimarans et al. (2015), with the aim to incorporate a higher-fidelity simulation model. Wang et al. (2019) develop a discrete event dynamic system simulation model for the ARP. Rather than using optimisation, which flights to delay or cancel is based on a set of rule-based methods. Reallocating aircraft is not considered. The simulation is used to identify the best proposed solution.

Multi-fidelity modelling could improve the efficiency of simulation optimisation by reducing computation times. The simpler, or 'low-fidelity', models can quickly identify potentially good solutions to be evaluated using the highest fidelity model for an improved understanding of their performance. These are prevalent in simheuristics (Juan et al., 2015), which often use deterministic models and heuristic searches to identify solutions to test in the simulation. Examples include combining a linear program with a simulation for production planning in a manufacturing context (Bang & Kim, 2010), a deterministic model with simulation for the inventory routing problem with stochastic demands (Onggo et al., 2019) and infinite-server queueing models and simulation for staffing A&E departments (Izady & Worthington, 2012). Xu et al. (2016) argue that low-fidelity models may have inconsistent inaccuracies, changing the optimal solution. This motivates utilising the simulation information within the search process itself. For example, one could model the high-fidelity performance as the low-fidelity performance plus an error function, which could be a quadratic response surface (Osorio & Chong, 2015) or a Gaussian Process (Huang et al., 2006; Inanlouganji et al., 2018). This allows both models to be used within the search. Our algorithm is more similar to that of Jian and Henderson (2015) who use simulation optimisation to improve upon the solutions from low-fidelity fluid and Markov chain models for bike allocation within a bike-sharing system.

3. Problem statement and method overview

Suppose an incident has occurred that disrupts an airline's intended schedule. The airline wishes to take action to minimise the disruption's impact, returning to normal operations within a certain time period known as the *recovery window*. The recovery window could extend to the end of the operating day (for short-haul carriers), as in Løve et al. (2005) and Zhang et al. (2015). Maher (2016) and Hu et al. (2017) use shorter recovery windows, whilst Sinclair et al. (2014) works with periods of multiple days. Note that when the recovery window

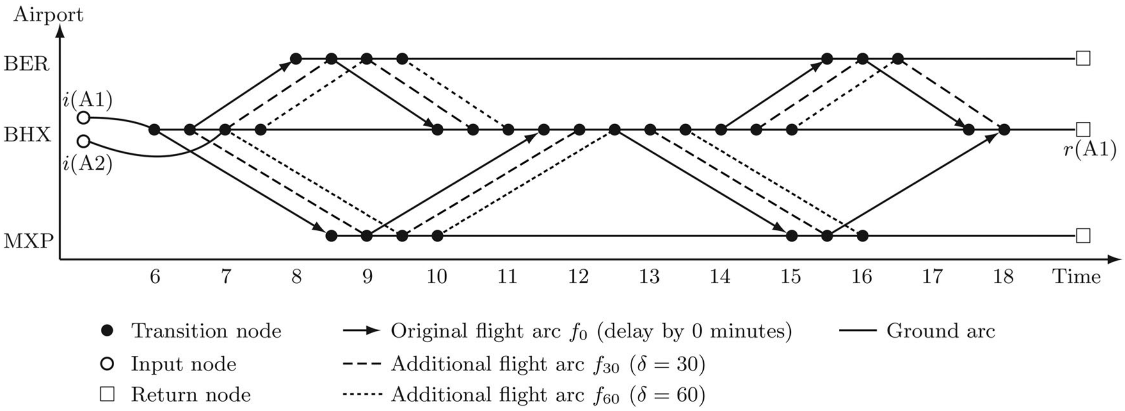
Airport
BER
BHX
MXP
i(A1)
i(A2)
r(A1)
6
7
8
9
10
11
12
13
14
15
16
17
18
Time
● Transition node
○ Input node
□ Return node
Original flight arc f_0 (delay by 0 minutes)
Additional flight arc f_{30} ($\delta = 30$)
Additional flight arc f_{60} ($\delta = 60$)
Ground arc

Figure 1. An example of the time-space network used by the IP under a disruption, $m = 30$ min and $M = 1$ h.

does not coincide with a break in operations, such as the day's end, one must consider the desired state of the fleet at the end of the recovery window. Maher (2016) points out that longer recovery windows can make the problems more difficult to solve.

Let A be the set of aircraft in play for resolving the disruption, wherever they are located. Here we assume all aircraft are of the same type. Let F be the set of flights originally covered by A within the recovery window, with $n = |F|$ being the number of flights. Each flight, $f \in F$, will require an aircraft, $a \in A$, and a planned delay time, d^f, or a cancellation which must be submitted to Air Traffic Control (ATC) before the departure time. Let $\mathbf{x} = \{x_a^f \in \{0,1\} : f \in F, a \in A\}$ denote the aircraft allocation (where $x_a^f \in \{0,1\}$ and $x_a^f = 1$ if and only if aircraft a is assigned to flight f) and $\mathbf{d} = (d^f : f \in F)$ be the vector of planned delays. The OCC has the multi-objective aim of rescheduling to minimise alterations to the original schedule, delays and costs by setting the decision variables $\mathbf{x}$ and $\mathbf{d}$. The costs arise directly from delays, passenger compensation and cancellation charges. Furthermore, this plan should be achievable, avoiding over-promising which could lead to flight f being delayed by more than d^f, damaging the airline's reputation and resulting in operational costs as a new plan with further schedule adjustments must be submitted to ATC. These additional costs are accounted for by a penalty if the actual delay exceeds d^f.

This paper proposes a multi-fidelity modelling approach to the ARP with two stages. Stage one uses a low-fidelity deterministic time-space network IP with discrete time. This IP identifies an aircraft allocation to flights and gives an initial value for the planned delay of each flight, considering all objectives. Solving the IP in a multi-objective manner produces a set of rescheduling options with different priorities for these objectives. This allows the combinatorial aircraft allocation aspect of the problem to be solved in a deterministic manner, separating it from the simulation optimisation.

The second stage uses each solution from the IP as a starting point for a local search for improvement in expected cost using simulation optimisation. This search enables more detailed information from the simulation to direct the optimisation process. We fix the aircraft allocation for each solution, treating only the planned delays as decision variables, removing the combinatorial constraints. The delays are treated as continuous variables, allowing the use of continuous simulation optimisation methods and gradient information.

4. Low-fidelity model

In this section we describe our low-fidelity model.

4.1. Time-space network model

The low-fidelity model is based on a time-space network IP (e.g. Thengvall et al., 2000). The flight schedule covering the recovery window is represented by a sequence of flight arcs between airports and ground arcs connecting a landing to a subsequent take-off. The departure and arrival of each flight is a transition node, $v \in V_T$, representing an airport at a particular time. Under disruption, the network is augmented with additional flight arcs for each flight starting from a later time. Let f_δ be the flight arc representing the flight f delayed by δ minutes. The potential delays are discrete time steps of size m up to a maximum M, i.e. $\delta \in \{0, m, 2m, \ldots, M\}$. This introduces new nodes and ground arcs into the network. Let L denote the set of all flight arcs, while the set of flight arcs associated with flight f is L^f. G is the set of all ground arcs in the network.

In addition to the transition nodes, each aircraft $a \in A$ has an input node, $i(a) \in V_I$, representing its location at the beginning of the recovery window. This applies to all aircraft in the fleet, regardless of the airport at which they are currently located. The input node is connected to the first node in V_T at which a is available for flight by a ground arc. No other arcs leave $i(a)$. All aircraft will end the recovery window at a return node, $r \in V_R$. Some aircraft may have a specified return node, denoted $r(a)$. For example, an aircraft may be required at an airport for its scheduled maintenance. In this case, a constraint is imposed to ensure the new schedule respects the maintenance plan. Let A_R denote the set of aircraft required to be at specific return nodes. The complete set of all nodes is $V = V_T \cup V_I \cup V_R$.

Figure 1 shows an example network for two aircraft, A1 and A2. The original schedule is represented by the original flight arcs and the ground arcs. At the end of the day, A1 is required at BHX for maintenance, so the return node for BHX is labelled $r(\text{A1})$ and $A_R = \{\text{A1}\}$. Suppose A2 will not be ready to fly until 7:00, hence $i(A2)$ is connected to node (BHX,7) rather than (BHX,6). If $M = 60$ min and $m = 30$ min, two additional flight arcs are added for each flight representing a delay of 30 min (dashed arcs) and 60 min (dotted arcs). The recovery window ends at 18:00, so no additional arcs are added that arrive beyond this time.

To ensure continuous aircraft movement through the network, flow constraints are imposed. Let L^{ν}_{in} and G^{ν}_{in} be the sets of flight arcs and ground arcs incident on node $\nu \in V$, and L^{ν}_{out} and G^{ν}_{out} be the sets of flight arcs and ground arcs exiting $\nu \in V$.

Between flights, aircraft must be prepared for the next flight. The industry term for this is the 'turn time'. The schedule must leave a minimum turn time between flights, $t_{\min}$. For node ν representing an airport at time t_ν, let $V^{\nu}_{t_{\min}}$ be the set of all nodes, $\nu' \in V_T$, representing the same airport at time $t_{\nu'}$ less than $t_{\min}$ later than time t_ν, i.e. $t_{\nu'} \in [t_\nu, t_\nu + t_{\min})$. If an aircraft ends a flight at node ν, it cannot start a new flight from any node in $V^{\nu}_{t_{\min}}$. The choice of $t_{\min}$ affects the recovery solution. Since turn time is actually variable, larger values of $t_{\min}$ correspond to higher quantiles of the turn time distribution, creating more robust solutions. However, if $t_{\min}$ is too large, it may delay flights that are not disrupted (if there is little buffer between flights).

A recovery plan must also ensure that the schedule beyond the recovery window is feasible with little or no disruption. For this to occur, the airline must have sufficient aircraft at each airport to operate the schedule after the recovery window. This is known as aircraft balance. Let r_A be the number of aircraft required by return node r. This constraint is used so that normal operations, i.e. no delays or cancellations, are resumed by the end of the recovery window. However, the aircraft allocation may differ from the original schedule.

An airline may only be allowed to use an airport runway during particular time slots due to the airport's runway capacity constraints. A "slot" s is a time period at a particular airport, W_s, with a restriction on the number of aircraft movements. The set S lists all slots. Let K_s be the maximum number of aircraft movements allowed by the airline in slot s at the relevant airport W_s and L_s be all flight arcs impacting slot s. This mechanism also deals with curfew times at airports, for example, preventing aircraft taking off during the night.

4.2. Integer program formulation

The model formulation is adapted from the aircraft recovery model in Zhang et al. (2015), but also draws on the models of Thengvall et al. (2000) and Jeng (2012), modified to allocate specific aircraft to flights, rather than fleet assignment.

We assume the cost of delaying flight f by δ is given by $c^{f_\delta} = c_d\delta + P^f(\delta)$, where c_d is the cost of delay per minute and P^f is the appropriate passenger compensation cost function, with the cost per passenger often being a step function of the delay. For example, for short-haul flights arriving or departing Europe, Civil Aviation Authority (2015) state that for a delay of 2 h, the airline must cover food and drink of each passenger, whilst a delay of over 3 h leads to a compensation of €250 per passenger. However, more general delay costs could be accommodated within this framework. Let C^f be the cost of cancelling flight f. Let $o^f_a \in \{0, 1\}$ indicate whether aircraft a was assigned to flight f before the disruption occurred.

The decision variables for the IP are all binary variables. Let $x^{f_\delta}_a$ indicate that aircraft a is assigned to flight arc f_δ, y^f indicate that flight f is cancelled, and z^γ_a indicate that aircraft a uses ground arc γ. The complete formulation is as follows.

$$\min \quad \sum_{a\in A}\sum_{f_\delta\in L} c^{f_\delta} x^{f_\delta}_a + \sum_{f\in F} C^f y^f \tag{1a}$$

$$\min \quad \sum_{a\in A}\sum_{f_\delta\in L}(1 - o^f_a)x^{f_\delta}_a \tag{1b}$$

$$\min \quad \sum_{a\in A}\sum_{f_\delta\in L} \delta x^{f_\delta}_a \tag{1c}$$

subject to

$$\sum_{a\in A}\sum_{f_\delta\in L^f} x^{f_\delta}_a + y^f = 1 \qquad \forall f \in F \tag{1d}$$

$$\sum_{a\in A}\sum_{f_\delta\in L_s} x^{f_\delta}_a \le K_s \qquad \forall s \in S \tag{1e}$$

$$\sum_{f_\delta\in L^{\nu}_{\text{in}}} x^{f_\delta}_a + \sum_{f_\delta\in L^{\nu}_{\text{out}}} x^{f_\delta}_a + \sum_{\nu'\in V^{\nu}_{t_{\min}}}\sum_{f_\delta\in L^{\nu'}_{\text{out}}} x^{f_\delta}_a \le 1 \qquad \forall a \in A, \forall \nu \in V_T \tag{1f}$$

$$\sum_{f_\delta\in L^{\nu}_{\text{in}}} x^{f_\delta}_a + \sum_{\gamma\in G^{\nu}_{\text{in}}} z^\gamma_a - \sum_{f_\delta\in L^{\nu}_{\text{out}}} x^{f_\delta}_a - \sum_{\gamma\in G^{\nu}_{\text{out}}} z^\gamma_a = 0 \qquad \forall a \in A, \forall \nu \in V_T \tag{1g}$$

$$\sum_{\gamma \in G^{i}_{\text{out}}} z^{\gamma}_{a} = 1_{\{i=i(a)\}} \qquad \forall a \in A, \forall i \in V_I \tag{1h}$$

$$\sum_{f_\delta \in L^{r(a)}_{\text{in}}} x^{f_\delta}_{a} + \sum_{\gamma \in G^{r(a)}_{\text{in}}} z^{\gamma}_{a} = 1 \qquad \forall a \in A_R \tag{1i}$$

$$\sum_{a \in A} \sum_{f_\delta \in L^{r}_{\text{in}}} x^{f_\delta}_{a} + \sum_{a \in A} \sum_{\gamma \in G^{r}_{\text{in}}} z^{\gamma}_{a} \geq r_A \qquad \forall r \in V_R \tag{1j}$$

$$x^{f_\delta}_{a}, z^{\gamma}_{a}, y^{f} \in \{0,1\} \qquad \forall a \in A, f \in F, f_\delta \in L, \gamma \in G \tag{1k}$$

Objective (1a) relates to the cost of the recovery action, objective (1b) to the number of changes made to aircraft allocation and objective (1c) to the total planned delay. Constraint (1d) ensures that each flight is either flown once or cancelled. Constraint (1e) is the slot constraint. Constraint (1f) prevents a turn time of less than the minimum allowable turn time; if aircraft a uses a flight arc incident on node ν, it cannot then use a flight arc exiting ν or any other node ν' within $t_{\min}$ of ν. Constraint (1g) is a flow constraint for each $\nu \in V_T$, whilst constraints (1h) and (1i) are analogous for the input and return nodes (where applicable), respectively. Constraint (1j) ensures aircraft balance at the end of the recovery window.

4.3. *Solving the integer program*

To generate multiple Pareto optimal solutions, the IP is solved using the ϵ-constraint method (Haimes et al., 1971). The objectives (1b) and (1c) are added to the problem as constraints:

$$\sum_{a \in A} \sum_{f_\delta \in L} (1 - o^{f}_{a}) x^{f_\delta}_{a} \in [l_E, u_E]$$

$$\sum_{a \in A} \sum_{f_\delta \in L} \delta x^{f_\delta}_{a} \in [l_D, u_D].$$

The program is solved several times with different constraint limits, l_D, u_D, l_E and u_E, minimising objective (1a). These parameters are modified efficiently using the method proposed by Laumanns et al. (2006) to explore various regions of the Pareto frontier. For further information about this method, see Section 1 of the accompanying supplementary material. A time limit is imposed for this application. Each iteration, defined by a set of constraint limits, is solved to optimality, except the final iteration which may run out of time.

The result is a set of solutions, $\mathcal{X}$, each with an aircraft allocation (including which flights are cancelled), $\mathbf{x}$, and a delay value for all flights in the programme, $\mathbf{d}$. The time limit means Pareto optimal solutions cannot be guaranteed. These solutions become starting points for the simulation optimisation, seeking local improvement.

5. High-fidelity model

The input recovery schedule for the simulation is given by an aircraft allocation $\mathbf{x}$ and a set of planned delays, $\mathbf{d}$. The elements of $\mathbf{x}$ and $\mathbf{d}$ link to the IP variables as follows:

$$x^{f}_{a} = \sum_{f_\delta \in L^f} x^{f_\delta}_{a},$$

$$d^{f} = \sum_{a \in A} \sum_{f_\delta \in L^f} \delta x^{f_\delta}_{a},$$

whilst a cancellation of flight f can be inferred when $\sum_{a \in A} x^{f}_{a} = 0$.

Let D^f be the actual delay of flight f. This is a random variable with a distribution dependent on $(\mathbf{x}, \mathbf{d})$. The objective function is the disruption's expected cost:

$$g(\mathbf{x}, \mathbf{d}) = \mathbb{E}\left\{ \sum_{f \in F} \left(c_d D^f + c_p (D^f - d^f)^{+} + P^f(D^f) \right) \right\} + C(\mathbf{x}). \tag{2}$$

Here, c_p is the penalty per minute for the actual delay D^f exceeding the planned delay d^f, incurred in addition to the delay cost $c_d D^f$. $P^f(D^f)$ represents the compensation associated with passenger delays of flight f and $C(\mathbf{x})$ is the cost of cancellations in the allocation $\mathbf{x}$. This objective function corresponds to (1a) in the IP. However, the framework could be used to handle any cost function used by an airline.

5.1. *Simulation model*

The simulation model is built within AnyLogic 8.2.3 (The AnyLogic Company, 2017). It simulates a homogeneous sub-fleet of a short-haul airline operating the recovery action $(\mathbf{x}, \mathbf{d})$ over a set of airports during the defined recovery window. Each aircraft follows its assigned schedule, subject to stochastic flight durations, turn times, queueing times and maintenance. The general framework is similar to SimAir, as discussed in Lee et al. (2003).

Where possible, distribution parameters used Maximum Likelihood Estimators based on the available data (Flightradar24 AB, 2017), obtained using the pyflightdata Python package (Allamraju, 2017). The data set contains information on 2.8 million flights, starting or ending at a set of 163 European airports. The information consists of origin, destination, aircraft type and identification, airline, departure and arrival times, and approximately half of the entries also have scheduled departure and arrival times. Where the data source does not track this information, assumptions on distributions and parameters were made. For a more in depth reporting of the simulation model used here, see Sections 2 and 3 of the supplementary material.

Whilst the simulation does not model all the details of airline operations, it does contain the key features necessary to evaluate our multi-fidelity approach. An airline has access to their own historical data for flight durations and turn times at all airports of interest, and a detailed knowledge of aircraft time-to-failure distributions and the times to repair aircraft, based on their facilities across the network. In addition, they will know the operating procedures at the individual airports at which they operate, accurate weather forecasts, and use of general data sources such as Flightradar24 (2017) to estimate how other airline's operations may impact on runway operations. All of this information will allow an airline to improve upon the simulation model used here, making it more specialised and relevant to their circumstances.

5.2. Simulation optimisation

Solutions found using the low-fidelity model, $(\mathbf{x}, \mathbf{d}_0) \in \mathcal{X}$, can be evaluated with more fidelity using the simulation model. Furthermore, to account for uncertainty not present in the low-fidelity model, an improvement is sought. Fixing $\mathbf{x}$ for each solution removes the combinatorial aspects, leaving a continuous optimisation problem over the planned delays, $\mathbf{d}$. To reduce the problem further, only the n^+ flights to which the IP allocates non-zero delays, $f \in F^+ = \{f \in F : d_0^f > 0\}$, are varied in the optimisation. The simulation optimisation problem is therefore

$$\min_{\mathbf{d}} \quad g(\mathbf{x}, \mathbf{d}) \tag{3a}$$

$$\text{subject to} \quad d^f \in [0, u^f] \quad \forall f \in F^+ \tag{3b}$$

$$d^f = 0 \quad \forall f \notin F^+ \tag{3c}$$

where g is defined in equation (2), and the constraint (3 b) states that a flight cannot leave early and must not exceed a maximum delay u^f. Thus, the feasible region, $\mathcal{D}$, is a hyper-box in $\mathbb{R}^{n^+}$.

We created an extended version of the STRONG simulation optimisation algorithm proposed by Chang et al. (2013) to enable it to cope with constraints. The main components of the extension are described below.

STRONG combines trust-region optimisation (see Conn et al., 2000) with Response Surface Modelling (see Kleijnen, 2014). At the j^{th} iteration, a linear or quadratic meta-model, $\hat{q}_j(\mathbf{d})$, is built to approximate the objective function around the current solution, $\mathbf{d}_j$. A new solution, $\mathbf{d}_j^*$, is proposed using a sub-problem of (approximately) minimising $\hat{q}_j(\mathbf{d})$ over an area around $\mathbf{d}_j$, $\mathcal{B}_j$, known as the trust region and determined by the trust-region radius, Δ_j. The proposed step to $\mathbf{d}_j^*$ is accepted or rejected based on two tests. The first compares the observed decrease in g with the predicted decrease in $\hat{q}_j$, whilst the second is a hypothesis test to see if $g(\mathbf{x}, \mathbf{d}_j^*)$ is significantly lower than $g(\mathbf{x}, \mathbf{d}_j)$ in the statistical sense. The trust region shrinks or expands depending on whether the meta-model produces a good solution. The primary difficulty for simulation optimisation is that $g(\mathbf{x}, \mathbf{d})$ and $\hat{q}_j(\mathbf{d})$ can only be estimated by taking multiple replications of the simulation at the solution $(\mathbf{x}, \mathbf{d})$.

As STRONG is for unconstrained optimisation, some significant adaptations are required as the optimum may lie on the boundary of $\mathcal{D}$. We briefly describe these in the remainder of this section. For further details on STRONG and the implementation of these adaptations, see Chang et al. (2013) and Section 4 of the Supplementary Materials, respectively.

5.2.1. Building the meta-model

When near a boundary, the classical factorial designs proposed by Chang et al. (2013) for fitting $\hat{q}_j(\mathbf{d})$ are not feasible as some design points may lie outside $\mathcal{D}$. A good design matrix, $\mathbf{D}_j$, is required to estimate $\hat{q}_j(\mathbf{d})$. One criterion for designs is D-optimality, which seeks to minimise the generalised variance of the design, $\det[(\mathbf{D}_j^T \mathbf{D}_j)^{-1}]$ (see Montgomery, 2009). Finding such a design is a complex optimisation problem. We implement the Coordinate-Exchange Algorithm heuristic proposed by Meyer and Nachtsheim (1995) to build the design in each iteration. This reduces the task to a sequence of one-dimensional problems by iteratively moving the design points around the trust region in a coordinate-wise manner to increase $\det[\mathbf{D}_j^T \mathbf{D}_j]$. Whilst not guaranteeing optimality, it produces good designs quickly, an important feature as it is run at each iteration in STRONG.

5.2.2. Proposing a new solution

After the meta-model is estimated, the sub-problem needs to be solved. Constraints on the feasible region mean that a solution proposed by minimising $\hat{q}_j(\mathbf{d})$ in the negative gradient direction within the trust region (as in STRONG) may not be feasible. Thus we adapt a procedure to guarantee a feasible solution. For bound constraints, such as (3b), Conn et al. (1988) suggested using a hyper-box trust region, defined by the ℓ_∞ norm, as this aligns itself with the boundaries of $\mathcal{D}$. Thus, our sub-problem becomes:

$$\min_{\mathbf{s}} \quad \hat{q}_j(\mathbf{d}_j + \mathbf{s}) \tag{4a}$$

$$\text{subject to} \quad \mathbf{d}_j + \mathbf{s} \in \mathcal{D} \tag{4b}$$

$$||\mathbf{s}||_\infty \leq \Delta_j. \tag{4c}$$

The objective is to find a step $\mathbf{s}$ that minimises the meta-model, $\hat{q}_j(\mathbf{d})$, whilst retaining feasibility (constraint (4b)) and remaining within the trust region (constraint (4c)). Rather than solving this constrained problem exactly, Conn et al. (1988) propose finding the Generalised Cauchy Step, $\mathbf{s}_j^*$, the step that minimises $\hat{q}_j(\mathbf{d})$ along the negative gradient direction projected onto the feasible trust region. This path is known as the projected gradient path, and consists of a series of straight lines. This simplified problem can be solved exactly. For the linear model, the step is to a corner of the hyper-box $\mathcal{D} \cap \mathcal{B}_j$. For the quadratic models, we use Algorithm 17.3.1 of Conn et al. (2000, p. 791), which finds local minima in each line segment of the projected gradient path, and selects the smallest. The proposed solution, $\mathbf{d}_j^* = \mathbf{d}_j + \mathbf{s}_j^*$, is then simulated to estimate $g(\mathbf{x}, \mathbf{d}_j^*)$.

5.2.3. Final stages

The algorithm ends when a replication budget is exhausted or a tolerance on optimality has been met. Due to discretisation of the IP, some unnecessary delays may have been inserted. Therefore, the algorithm tests the final solution $(\mathbf{x}, \mathbf{d}^*)$ against $(\mathbf{x}, \mathbf{0})$, to check that these delays are necessary. A one-sided Welch's t-test (Welch, 1938) is used to compare:

$$H_{\text{null}} : g(\mathbf{x}, \mathbf{0}) \geq g(\mathbf{x}, \mathbf{d}^*); \qquad H_{\text{alt}} : g(\mathbf{x}, \mathbf{0}) < g(\mathbf{x}, \mathbf{d}^*).$$

If H_{null} is rejected, the final solution returned is $(\mathbf{x}, \mathbf{0})$.

The whole procedure is performed for each $(\mathbf{x}, \mathbf{d}_0) \in \mathcal{X}$, producing a set of improved solutions that have been thoroughly evaluated using the simulation. The airline can then consider each solution alongside other practicalities, before a solution is proposed to ATC.

6. Empirical evaluation

In practice, the IP would be solved as in Section 4.3 to produce a set of solutions and the simulation optimisation procedure of Section 5.2 would then be used once on each solution. However, when performing a computational evaluation of simulation optimisation algorithms on real-world problems, the reporting must consider two levels of uncertainty. The first is the uncertainty in the performance of an individual solution. As the exact performance is unknown, performance must be estimated using many replications of the simulation. As the rescheduling is a one-time-only decision, understanding the full distribution of outcomes is important, rather than just the expected value. Thus, the solution returned by the simulation optimisation algorithm is simulated many times for a thorough understanding of its performance. Using Common Random Numbers (CRN) improves the comparison by ensuring, as much as possible, that the solutions face the same situations.

The second layer is in the algorithm itself. The algorithm described in Section 5 relies on the random output of simulation replications. Therefore, the solution returned is dependent on the starting seed of the random number stream; a single algorithm run cannot characterise the algorithm's performance. Thus, for the purposes of evaluating this method, for each problem, we perform macro-replications of the whole algorithm (excluding the IP, which is deterministic). Each macro-replication produces a possibly unique solution, which is simulated using 1000 CRN replications.

Another difficulty is that the optimal expected cost is unknown. Thus, one cannot simply measure the optimality gap. Instead, we either evaluate the simulation optimisation performance in terms of the improvement over the IP solutions or the gap to the best solution found across the macro-replications.

6.1. Problem descriptions

Three realistic problems were created based on short-haul carrier schedules extracted from the data source (Flightradar24 AB, 2017). Problem 1 involves a fleet of 8 aircraft, one of which has a technical fault requiring a few hours to repair. Problem 2 has a similar setting, but with a fleet of 48 aircraft. In both cases, the proposed method is used to reschedule the remainder of the day's flights.

In Problem 3 poor weather at a hub airport causes high congestion at the runways, affecting 11 immediate flights and their subsequent schedules. The fleet size is 102 aircraft. Here, the airline cannot make changes to flights from the affected airport, which are excluded from the IP, but can rearrange future flights. The original schedule under the disruption is simulated to estimate when the affected aircraft become available after completing their weather disrupted flights. These times are used for the respective input node times within the IP.

The cost per minute of delay is set to $c_d = €50$ with a penalty cost of $c_p = €20$ per minute. Further details of the problems, algorithm parameter settings and results can be found in the supplementary material.

6.2. Integer program results

For each problem, the IP was solved using the Gurobi Optimizer 7.0.2 (Gurobi Optimization LLC, 2017) on a Cluster of 4 nodes each with 2 x Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.6 GHz and 512 Gb RAM, running Ubuntu Linux. For Problem 1, 8

Table 1. IP solutions for Problems 1, 2, and 3.

Problem	Max. Delay *M*	Time Step *m*	Solution	Cost (€1000)	Aircraft Exchanges	Total Delay	Max. delay in Solution	Cancelled Flights	Solution Time
1	180	5	1	15.00	30	300	95	0	4.23
			2	15.00	15	300	95	0	15.67
			3	15.75	12	315	95	0	12.61
			4	16.50	8	330	95	0	25.66
			5	25.25	4	495	125	0	18.71
2	600	15	8	6.00	19	120	60	0	441.51[a]
			9	6.25	17	135	60	0	65.06
			10	9.75	15	195	60	0	90.28
			11	9.75	14	195	90	0	62.00
			12	11.06	13	195	120	0	76.96
			13	12.75	12	255	90	0	104.75
3	60	20	12	27.00	17	540	60	0	2131.90[a]
			13	29.00	13	580	60	0	67.41
			14	31.00	9	620	60	0	72.30
			15	33.00	8	660	60	0	61.63
			16	35.00	4	700	60	0	447.72

Note: Delays are measured in minutes, and the solution time is measured in seconds.
[a] Cumulative time of multiple iterations of the multi-objective algorithm producing cost optimal solutions but with more aircraft exchanges than the Pareto optimal solutions.

processors were used, and 20 processors for Problems 2 and 3. The results are shown in Table 1.

Each of the problems show a clear trade-off between cost and the number of aircraft exchanges. This is predictable, since allowing more aircraft to be exchanged gives more flexibility to make schedule changes, and thus lower cost solutions become available. In each problem, the algorithm prevents cancellations and largely avoids long delays. Therefore, compensation costs are generally avoided, as these only apply to cancelled flights and delays exceeding 2 h. Thus, cost and delay are highly correlated.

Many of the solutions are achieved within two minutes. This speed would be appropriate for the application. However, for the larger problems, some of the solution times are too long, especially as the maximum allowable delay for Problem 3 is only $M = 60$ min. Whilst the time available to solve a problem depends on the nature of the disruption, the time limits used in the literature vary from 3 min (e.g. Løve et al., 2005) to 20 min for a multi-objective approach (Hu et al., 2017). As the IP is only the first stage of our proposed framework, this suggests more specialised problem reduction and solution approaches are required.

One important observation is the time to find the first Pareto optimal solution in the larger problems: it took 8 and 12 iterations of the multi-objective algorithm for Problems 2 and 3, respectively. This suggests that the pure ϵ-constraint method could be improved upon, such as using a hybrid multi-objective to penalise aircraft exchanges that are not Pareto optimal.

Additional experimentation (not included in Table 1) indicated a further trade-off in the choice of m. A smaller value of m allows a greater choice of flight arcs; thus a more flexible solution space. This can improve solutions significantly as new aircraft exchanges become available. However, the IP grows quickly as m decreases (the number of variables is $O(1/m)$). Moreover, Thengvall et al. (2000) found it increased the number of non-integer solutions to the linear relaxation, increasing the solution time. This is particularly clear in Problem 3. Using $m = 10$ min instead gives an optimal cost of €15,000, but the first iteration takes 864.70 s to solve. The improvement in solution quality suggests that developing problem reduction techniques that enable a smaller step size would be advantageous for real implementation.

6.3. Simulation optimisation results

Each of the solutions in Table 1 provide a starting point for the simulation optimisation algorithm, with a budget of 2000 simulation replications, and 50 replications for the comparison with $\mathbf{d} = \mathbf{0}$. (An iteration in progress when the budget runs out is allowed to complete.) The maximum number of samples at each design point to build the metamodel $\hat{q}_j(\mathbf{d})$ is 50. Up to 100 replications are used for the acceptance tests. The full set of algorithm parameters used is listed in Section 5 of the supplementary material.

Each experiment is repeated multiple times (100 for Problem 1, and 50 for Problems 2 and 3). This produces either 100 or 50 (possibly unique) solutions for each starting point. To evaluate these solutions, each is simulated 1000 times using CRN to estimate its cost distribution.

For comparison, each starting point and a "No Action" response are also simulated using the same CRN replications. This solution involves no changes to the schedule; each disrupted aircraft operates its original schedule once it is available to do so, generally leading to large unplanned delays. Whilst unrealistic, the "No Action" response of making no

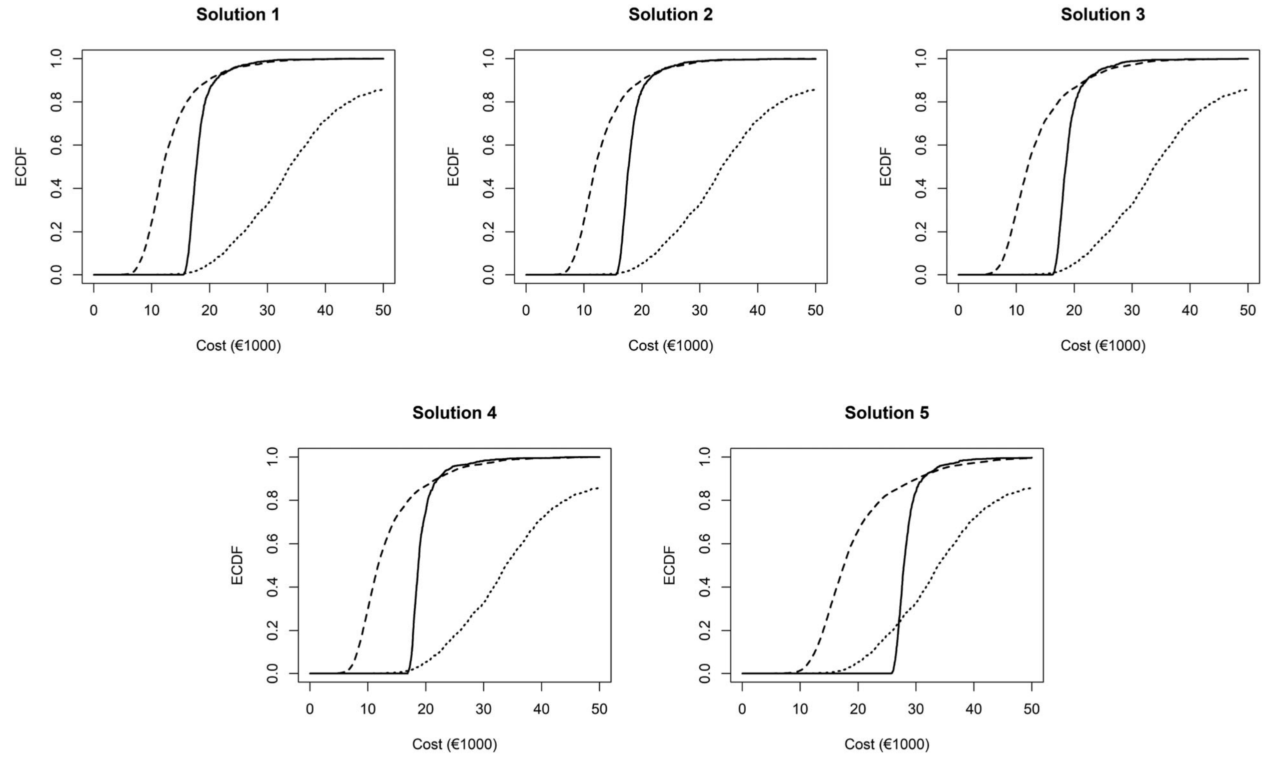

Figure 2. ECDFs for each starting solution in Problem 1. Solid line is the IP solution, dashed line is the combined simulation optimisation solutions, dotted line is 'No Action'.

changes to the schedule gives a benchmark for potential improvement.

Figures 2, 3, and 4 show the Empirical Cumulative Distribution Functions (ECDFs) of the simulated cost of the IP solution, the "No Action" solution and the simulation optimisation solutions (combined into one distribution). Combining the simulation optimisation solutions into one ECDF helps to account for the variance due to the starting seed, though it can hide poor performing solutions.

The ideal outcome from these plots would be the simulation optimisation ECDF never going below the IP solution ECDF. In general, whilst the mean decreases in all scenarios, the variance increases as the simulation optimisation capitalises on the scheduled buffer to reduce delays where possible. The worst case is seen in Problem 1 (Figure 2) where the upper tail of the distributions get longer. The majority of the macro-replication solutions see an increase in the 95th percentile. However, the upper quantiles are similar to the IP solutions, giving a sense of robustness.

For Problem 2, see Figure 3, the improvement is smaller, as much of the gains have been found using the IP. For each of the starting solutions the 95th percentile is reduced in at least 78% of macro-replications.

In the third problem, Figure 4, the gain made by the IP is fairly small. These poor initial results could be due to conservatively estimated ready times of the aircraft directly affected by the poor weather. However, the change in aircraft allocation allows the simulation optimisation to reduce the costs. The combination leads to a much improved result. In all macro-replications, the simulation reduces both the mean and the 95th percentile cost.

In all cases the IP and simulation optimisation solutions are more robust than the "No Action" option.

The final analysis considers the consistency of the simulation optimisation. As the aim is to minimise the expected cost, we consider the gap between the best mean cost from the macro-replications. The estimated "relative optimality gap" of the solution of the k^{th} macro-replication of Plan p, $(\mathbf{x}_p, \mathbf{d}_{kp})$, is

$$\rho_{\text{gap}}(\mathbf{x}_p, \mathbf{d}_{kp}) = \frac{\hat{g}(\mathbf{x}_p, \mathbf{d}_{kp}) - \min_k\{\hat{g}(\mathbf{x}_p, \mathbf{d}_{kp})\}}{\min_k\{\hat{g}(\mathbf{x}_p, \mathbf{d}_{kp})\}} \times 100\%.$$

Note that as the true optimal is unknown, this is an estimate of the optimality gap relative to the best solution discovered. Figure 5 shows the ECDFs of $\rho_{\text{gap}}(\mathbf{x}_p, \mathbf{d}_{kp})$ for each problem (all starting points combined into one plot). It shows that the algorithm is least consistent for Problem 1, where it is only within 5% of the best solution in 60% of macro-replications. The consistency is much higher for Problems 2 and 3, where in 90% of macro-replications it achieves within 5% of the best solution found.

7. Conclusions and further work

This paper has proposed a multi-fidelity modelling approach for the ARP. It combines integer

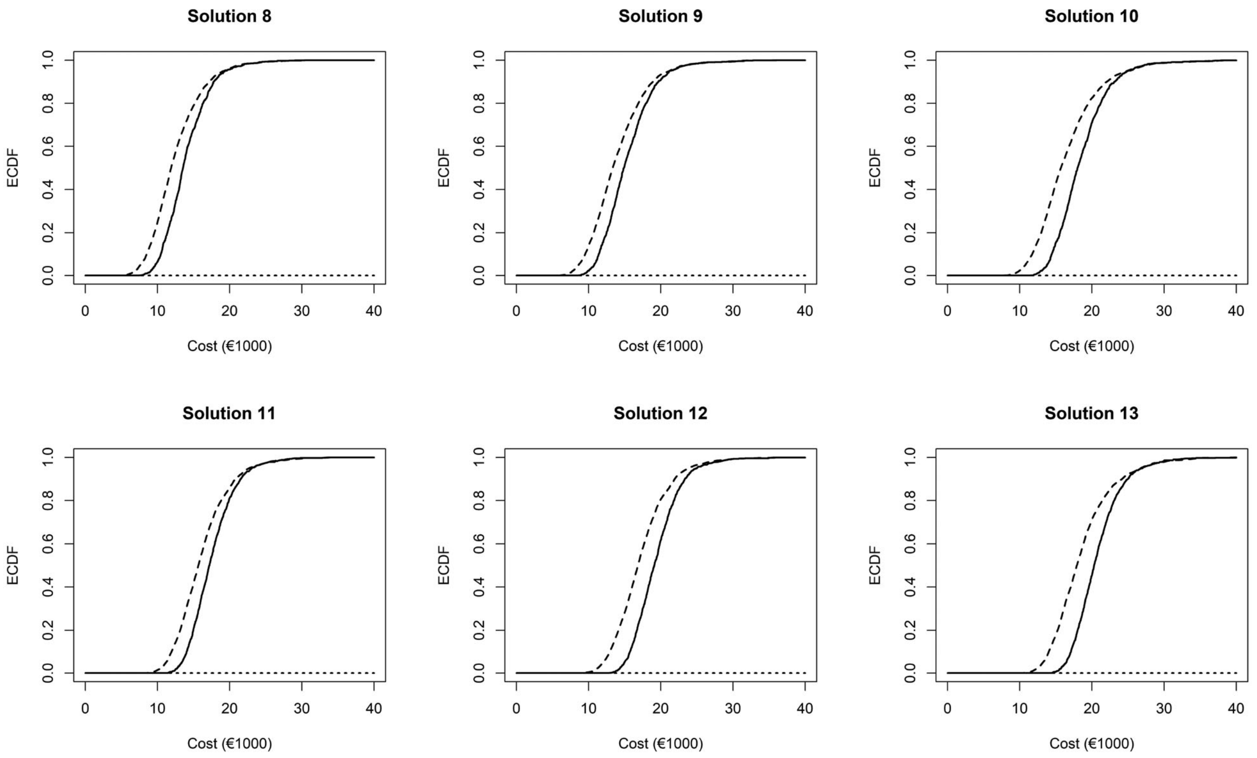

Figure 3. ECDFs for each starting solution in Problem 2. Solid line is the IP solution, dashed line is the combined simulation optimisation solutions, dotted line is 'No Action'.

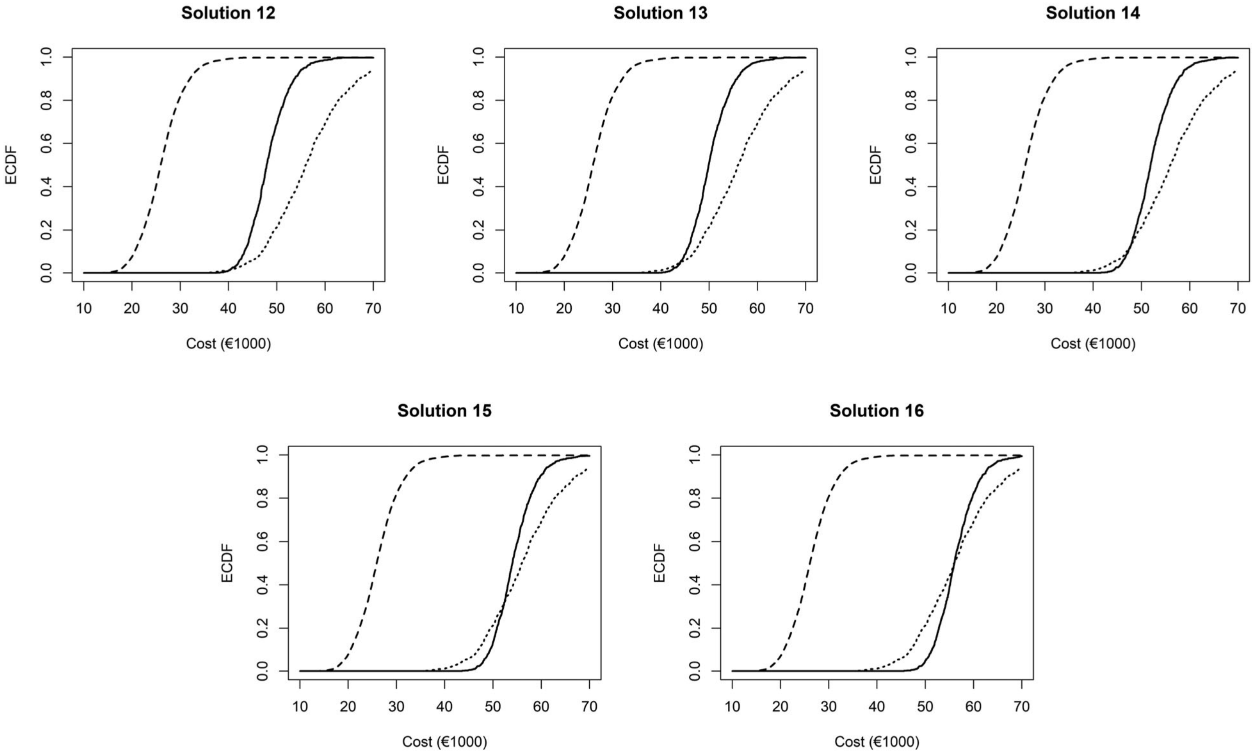

Figure 4. ECDFs for each starting solution in Problem 3. Solid line is the IP solution, dashed line is the combined simulation optimisation solutions, dotted line is 'No Action'.

programming with simulation to effectively search through the solution space. Using the simulation as part of the optimisation allows some correction for simplifications in the IP. The nature of the real problem is that OCC needs to be able to integrate the ARP solution with other parts of the recovery plan and provide it to ATC, who can reject a proposed plan. This increases the value of producing a range of good but different solutions within a limited time. The results show that this combination is

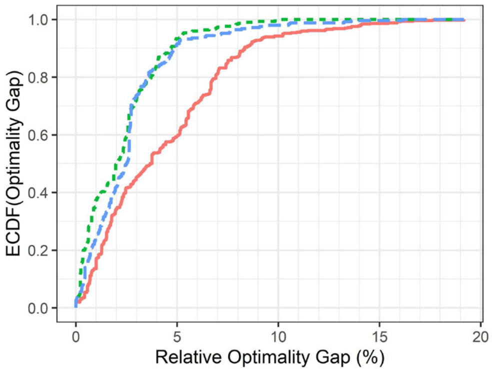

Figure 5. ECDFs of the Relative Optimality Gap for each Problem.

consistently able to find good solutions to the ARP, with the simulation optimisation providing improvements over the initial solutions. We believe that the combination of IP to find Pareto-optimal aircraft allocations to a deterministic version of the problem, followed by a local-search based simulation optimisation to fine tune the delays is well-suited to the ARP context.

As part of the work it was necessary to adapt the simulation optimisation algorithm STRONG to cope with bound constraints, and as such could be applicable beyond the ARP studied here.

Whilst we have demonstrated our framework across three realistic problems, further investigation of the framework across a larger variety of problem instances would improve our understanding of the performance and potentially identify if there are particular problem structures or sizes that cause a difficulty for this framework.

To move to a real implementation, the major challenge is a reduction in the computation time. The pure ϵ-constraint method for the IP could be compared with a hybrid multi-objective approach, investigating which finds the Pareto frontier sooner. Furthermore, more problem-specific solution methods could improve the solution times for the IP. The simulation optimisation procedure could be naturally parallelised by simulating each design point on a different processor. This would decrease the computation time significantly, as this is the most expensive part of the process. One may also wish to improve what the simulation explicitly models, such as passenger connections and non-linear costs. This would improve the performance estimation and increase the value of the method.

Acknowledgements

We thank Richard Standing, Nigel Jackson, Stewart Preston and Mike Chester at Rolls-Royce (R^2 Data Labs) for the original research idea and contextual information.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the EPSRC under Grant EP/L015692/1 STOR-i Centre for Doctoral Training; NSF under DMS-1854562; and Rolls-Royce Limited.

ORCID

L. A. Rhodes-Leader http://orcid.org/0000-0002-9140-6887
B. L. Nelson http://orcid.org/0000-0002-1325-2624
B. S. Onggo http://orcid.org/0000-0001-5899-304X
D. J. Worthington http://orcid.org/0000-0003-1400-0194

References

Abdelghany, K. F., Abdelghany, A. F., & Ekollu, G. (2008). An integrated decision support tool for airlines schedule recovery during irregular operations. *European Journal of Operational Research*, *185*(2), 825–848. https://doi.org/10.1016/j.ejor.2006.12.045

Allamraju, H. (2017). pyflightdata. Retrieved March 11, 2017. https://github.com/supercoderz/pyflightdata.

Arias, P., Mujica Mota, M., Guimarans, D., & Boosten, G. (2013). A methodology combining optimization and simulation for real applications of the stochastic aircraft recovery problem. In *Proceedings of the 8th EUROSIM Congress on Modelling and Simulation, EUROSIM 2013* (pp. 265–270). IEEE Computer Society.

Bang, J.-Y., & Kim, Y.-D. (2010). Hierarchical production planning for semiconductor wafer fabrication based on linear programming and discrete-event simulation. *IEEE Transactions on Automation Science and Engineering*, *7*(2), 326–336.

Bratu, S., & Barnhart, C. (2006). Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, *9*(3), 279–298. https://doi.org/10.1007/s10951-006-6781-0

Chang, K.-H., Hong, L. J., & Wan, H. (2013). Stochastic trust-region response-surface method (STRONG) - a new response-surface framework for simulation optimization. *INFORMS Journal on Computing*, *25*(2), 230–243. https://doi.org/10.1287/ijoc.1120.0498

Civil Aviation Authority. (2015). Your rights when you fly. Retrieved March 20, 2020, from https://www.caa.co.uk/Passengers/Resolving-travel-problems/Delays-cancellations/Your-rights/Your-rights-when-you-fly/.

Clausen, J., Larsen, A., Larsen, J., & Rezanova, N. J. (2010). Disruption management in the airline industry – concepts, models and methods. *Computers & Operations Research*, *37*(5), 809–821. https://doi.org/10.1016/j.cor.2009.03.027

Conn, A. R., Gould, N. I. M., & Toint, P. L. (1988). Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, *25*(2), 433–460. https://doi.org/10.1137/0725029

Conn, A. R., Gould, N. I. M., & Toint, P. L. (2000). *Trust region methods*. Society for Industrial and Applied Mathematics.

EUROCONTROL. (2020). CODA Digest – Annual report for 2019: All-causes delay and cancellations to air transport in Europe.

Federal Aviation Administration. (2020). *Cost of delay estimates*: 2019. https://www.faa.gov/data_research/aviation_data_statistics/media/cost_delay_estimates.pdf.

Flightradar24 AB. (2017). *Flightradar24*. Retrieved March 7, from 2017. https://www.flightradar24.com.

Guimarans, D., Arias, P., & Mujica Mota, M. (2015). Large neighbourhood search and simulation for disruption management in the airline industry. In M. Mujica Mota, I. Flores De La Mota and D. Guimarans Serrano (Eds.), *Applied simulation and optimization: in logistics, industrial and aeronautical practice* (pp. 169–201). Springer International Publishing.

Gurobi Optimization, LLC. (2017). Gurobi Optimizer Reference Manual. Retrieved July 23, from 2018. http://www.gurobi.com.

Haimes, Y., Lasdon, L. S., & Wismer, D. A. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, *1*(3), 296–297.

Hu, Y., Liao, H., Zhang, S., & Song, Y. (2017). Multiple objective solution approaches for aircraft rerouting under the disruption of multi-aircraft. *Expert Systems with Applications*, *83*, 283–299. https://doi.org/10.1016/j.eswa.2017.04.031

Huang, D., Allen, T. T., Notz, W. I., & Miller, R. A. (2006). Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, *32*(5), 369–382. https://doi.org/10.1007/s00158-005-0587-0

Hutchison, D. W., & Hill, S. D. (2001). Simulation optimization of airline delay with constraints. In B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer (Eds.), *Proceedings of the 2001 Winter Simulation Conference* (pp. 1017–1022). IEEE.

Inanlouganji, A., Pedrielli, G., & Fainekos, G. (2018). Continuous simulation optimization with model mismatch using gaussian process regression. In M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson (Eds.), *Proceedings of the 2018 Winter Simulation Conference* (pp. 2131–2142). IEEE.

Izady, N., & Worthington, D. (2012). Setting staffing requirements for time dependent queueing networks: The case of accident and emergency departments. *European Journal of Operational Research*, *219*(3), 531–540. https://doi.org/10.1016/j.ejor.2011.10.040

Jeng, C.-R. (2012). Real-time decision support for airline schedule disruption management. *African Journal of Business Management*, *6*(27), 8071–8079.

Jian, N., & Henderson, S. G. (2015). An introduction to simulation optimization. In L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti (Eds.) (Ed.), *Proceedings of the 2015 Winter Simulation Conference* (pp. 1780–1794). IEEE.

Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, *2*, 62–72. https://doi.org/10.1016/j.orp.2015.03.001

Kleijnen, J. P. C. (2014). Response surface methodology. In M. C. Fu (Ed.), *Handbook of simulation optimization* (Vol. 216, pp. 81–104). Springer. Retrieved December 24, 2019, from https://ebookcentral.proquest.com

Kohl, N., Larsen, A., Larsen, J., Ross, A., & Tiourine, S. (2007). Airline disruption management - perspectives, experiences and outlook. *Journal of Air Transport Management*, *13*(3), 149–162. https://doi.org/10.1016/j.jairtraman.2007.01.001

Laumanns, M., Thiele, L., & Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, *169*(3), 932–942. https://doi.org/10.1016/j.ejor.2004.08.029

Lee, L. H., Huang, H. C., Lee, C., Chew, E. P., Wikrom, J. Yong, Y. Y., Liang, Z., Leong, C. H., Tan, Y. P., Namburi, K., Johnson, E., & Banks, J. (2003). Discrete Event Simulation Model for Airline Operations: SIMAIR. In S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice (Eds.), *Proceedings of the 2003 Winter Simulation Conference* (pp. 1656–1662). IEEE.

Liang, Z., Xiao, F., Qian, X., Zhou, L., Jin, X., Lu, X., & Karichery, S. (2018). A column generation-based heuristic for aircraft recovery problem with airport capacity constraints and maintenance flexibility. *Transportation Research Part B: Methodological*, *113*, 70–90. https://doi.org/10.1016/j.trb.2018.05.007

Løve, M., Sørensen, K. R., Larsen, J., & Clausen, J. (2005). Using heuristics to solve the dedicated aircraft recovery problem. *Central European Journal of Operations Research*, *13*(2), 189–207.

Maher, S. J. (2016). Solving the integrated airline recovery problem using column-and-row generation. *Transportation Science*, *50*(1), 216–239. https://doi.org/10.1287/trsc.2014.0552

Meyer, R. K., & Nachtsheim, C. J. (1995). The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics*, *37*(1), 60–69. https://doi.org/10.1080/00401706.1995.10485889

Montgomery, D. C. (2009). *Design and analysis of experiments* (7th ed.). John Wiley Pte Ltd.

Onggo, B. S., Panadero, J., Corlu, C. G., & Juan, A. A. (2019). Agri-food supply chains with stochastic demands: A multi-period inventory routing problem with perishable products. *Simulation Modelling Practice and Theory*, *97*, 101970. https://doi.org/10.1016/j.simpat.2019.101970

Osorio, C., & Chong, L. (2015). A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. *Transportation Science*, *49*(3), 623–636. https://doi.org/10.1287/trsc.2014.0550

Rhodes-Leader, L. A., Onggo, B. S., Worthington, D. J., & Nelson, B. L. (2018). Multi-fidelity simulation optimisation for airline disruption management. In M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson (Eds.), *Proceedings of the 2018 Winter Simulation Conference* (p. 2179–2190). IEEE.

Rosenberger, J. M., Johnson, E. L., & Nemhauser, G. L. (2003). Rerouting aircraft for airline recovery. *Transportation Science*, *37*(4), 408–421. https://doi.org/10.1287/trsc.37.4.408.23271

Rosenberger, J. M., Schaefer, A. J., Goldsman, D., Johnson, E. L., Kleywegt, A. J., & Nemhauser, G. L. (2000). SimAir: A stochastic model of airline operations. In J. A. Joines, R. R. Barton, K. Kang, and P. A.

Fishwick (Eds.) (Ed.), *Proceedings of the 2000 Winter Simulation Conference* (pp. 1118–1122). IEEE.

Rosenberger, J. M., Schaefer, A. J., Goldsman, D., Johnson, E. L., Kleywegt, A. J., & Nemhauser, G. L. (2002). A stochastic model of airline operations. *Transportation Science*, *36*(4), 357–377. https://doi.org/10.1287/trsc.36.4.357.551

Sinclair, K., Cordeau, J.-F., & Laporte, G. (2014). Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem. *European Journal of Operational Research*, *233*(1), 234–245. https://doi.org/10.1016/j.ejor.2013.08.034

The AnyLogic Company (2017). *AnyLogic 8.2.3*. Retrieved July 23, 2018, from http://www.anylogic.com.

Thengvall, B. G., Bard, J. F., & Yu, G. (2000). Balancing user preferences for aircraft schedule recovery during irregular operations. *IIE Transactions*, *32*(3), 181–193. https://doi.org/10.1080/07408170008963891

Wang, D., Wu, Y., Hu, J.-Q., Liu, M., Yu, P., Zhang, C., & Wu, Y. (2019). Flight schedule recovery: A simulation-based approach. *Asia-Pacific Journal of Operational Research*, *36*(06), 1940010. https://doi.org/10.1142/S0217595919400104

Welch, B. L. (1938). The significance of the difference between two means when the population variances are unequal. *Biometrika*, *29*(3/4), 350–362. https://doi.org/10.1093/biomet/29.3-4.350

Xu, J., Zhang, S., Huang, E., Chen, C-h., Lee, L. H., & Celik, N. (2016). MO^{2}TOS: Multi-fidelity optimization with ordinal transformation and optimal sampling. *Asia-Pacific Journal of Operational Research*, *33*(3), 1650017. https://doi.org/10.1142/S0217595916500172

Zhang, D., Henry Lau, H. Y. K., & Yu, C. (2015). A two stage heuristic algorithm for the integrated aircraft and crew schedule recovery problems. *Computers & Industrial Engineering*, *87*, 436–453. https://doi.org/10.1016/j.cie.2015.05.033

Zhu, B., Zhu, J-f., & Gao, Q. (2015). A stochastic programming approach on aircraft recovery problem. *Mathematical Problems in Engineering*, *2015*, 680609. https://doi.org/10.1155/2015/680609