Computers & Operations Research ■■■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Computers & Operations Research

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

A multi-objective approach for robust airline scheduling

Edmund K. Burke [a], Patrick De Causmaecker [b], Geert De Maere [a,d,*], Jeroen Mulder [c], Marc Paelinck [c], Greet Vanden Berghe [d]

[a] School of Computer Science, Jubilee Campus, 1 Wollaton Road, NG8 1BB Nottingham, UK
[b] Faculty of Sciences, Department of Informatics, Katholieke Universiteit Leuven Campus Kortrijk, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
[c] KLM Decision Support, Corporate Information Office, KLM Royal Dutch Airlines, KLM Headquaters, Amsterdamseweg 55, 1182 GP Amstelveen, The Netherlands
[d] Vakgroep Informatie Technologie, KaHo Sint Lieven, Gebroeders Desmetstraat 1, 9000 Ghent, Belgium

A R T I C L E I N F O

Keywords:
Multi-objective improvement
Robust airline scheduling
Multi-meme memetic algorithms
Large scale simulation

A B S T R A C T

We present a memetic approach for multi-objective improvement of robustness influencing features (called *robustness objectives*) in airline schedules. Improvement of the objectives is obtained by simultaneous flight retiming and aircraft rerouting, subject to a fixed fleet assignment. Approximations of the Pareto optimal front are obtained by applying a *multi-meme memetic algorithm*. We investigate *biased meme selection* to encourage exploration of the boundaries of the search space and compare it with *random meme selection*. An external population of high quality solutions is maintained using the adaptive grid archiving algorithm.

The presented approach is applied to investigate simultaneous improvement of *reliability* and *flexibility* in real world schedules from KLM Royal Dutch Airlines. Experimental results show that the approach enables us to obtain schedules with significant improvements for the considered objectives. A large scale simulation study was undertaken to quantify the influence of the robustness objectives on the operational performance of the schedules. Rigorous sensitivity analysis of the results shows that the influence of the schedule reliability is dominant and that increased schedule flexibility could improve the operational performance.

1. Introduction

Delay analysis carried out by Eurocontrol, shows that the European air transport industry is suffering from an indisputable increase in delays [1]. In 2005, 42% of the flights were delayed and as much as 20% of the flights were delayed by 15 min or more. In the same year, the average delay per movement was approximately 11 min, which is an increase of 9% compared to 2004. Investigating the causes of delays reveals that 50% are due to airline related issues, whereas 19% are due to airport operations [1]. In addition, the predicted growth in air transport over the next few years (approx. 26% by 2013 for Europe [2]) will cause increased congestion of airports and airspace, which is likely to result in a further increase in the number of delays [3]. This makes the construction of more *robust schedules*—which are less likely to be delayed—an absolute necessity for airlines.

For most of the airlines, different departments are involved in the scheduling process: the network department, the operational plan management department and operations control. The *network department* is responsible for the construction of an initial feasible schedule in the early planning stages. The main goals of the network department are maximisation of the market share, maximisation of passenger revenue and minimisation of the operating cost. At KLM, schedule construction takes the network department approximately 2 months, after which it is passed on to the operational plan management department. The *operational plan management department* is responsible for making minor changes to the schedule to improve its operational performance and to anticipate changes in the market. Approximately two weeks before the day of operation, the schedule is passed on to *the operations department*, which is responsible for making last minute adaptations and managing the schedule on the day of operation. This includes implementing recovery strategies to mitigate the effects of a disruption. The degree of complexity required to implement recovery strategies is influenced by the scheduling practices of the network and operational plan management department. Ideally, these departments would include

* Corresponding author at: School of Computer Science, Jubilee Campus, 1 Wollaton Road, NG8 1BB Nottingham, UK.
E-mail addresses: ekb@cs.nott.ac.uk (E.K. Burke), Patrick.DeCausmaecker@kuleuven-kortrijk.be (P. De Causmaecker), gdm@cs.nott.ac.uk (G. De Maere), Jeroen.Mulder@klm.com (J. Mulder), Marc.Paelinck@klm.com (M. Paelinck), Greet.VandenBerghe@KaHoSL.be (G. Vanden Berghe).

0305-0548/$ - see front matter © 2009 Published by Elsevier Ltd.
doi:10.1016/j.cor.2009.03.026

robustness improving patterns in the schedule, which could be exploited by the operations department. However, well-considered inclusion of sophisticated recovery patterns is a difficult task and requires a deep understanding of their individual and combined influence on the schedule's operational performance.

Unfortunately, most of the current decision support tools for airline scheduling assume deterministic operating conditions and discard the influence of potential disruptions on the day of operation [3,4]. In addition, current models have a strong focus on minimising the operating cost, improving the utilisation of resources and decomposing the scheduling problem into several independent sub-problems. We distinguish between flight scheduling, fleet assignment, aircraft rotation-maintenance scheduling and crew scheduling. The *flight scheduling* problem considers the construction of a feasible flight network by selecting a set of origin–destination pairs, assigning the flight times and determining the operating frequencies. The resulting flight network is used as input to the *fleet assignment problem*, which assigns a fleet type to each individual flight leg. The main goals of the fleet assignment are the maximisation of the captured and recaptured passengers, maximisation of revenue and through revenue (additional revenue stemming from the premium passengers pay for the comfort of staying on the same plane) and minimisation of the operating cost [5]. Feasibility constraints for the fleet assignment process include flow constraints, balance constraints, availability constraints and flight coverage constraints [5]. *Aircraft rotations* are built by assigning a sequence of flights to each individual aircraft of a certain fleet type and assigning the maintenance slots according to the maintenance regulations of the Federal Aviation Authorities (FAA). *Crew rotations* or *crew pairings*—which are not necessarily the same as the aircraft rotations—are constructed during *crew scheduling*. Crew scheduling involves assigning pilots and cabin crew to each individual flight, subject to FAA regulations, airline specific rules and labour agreements. We refer to Etschmaier and Mathaisel [6] for a survey on early work in airline scheduling and to Barnhart et al. [7] and Barnhart and Cohn [8] for more recent literature surveys. A rigorous literature review of the state-of-the-art crew scheduling models can be found in Gopalakrishnan and Johnson [9].

Applying deterministic and decomposed approaches is known to result in sub-optimal schedules with many tightly interconnected resources, leaving less flexibility in the schedule to recover from delays. Recent trends in airline scheduling focus on integrated scheduling models and new approaches to build more robust schedules. *Integrated scheduling* considers the modelling of downstream effects on the subsequent scheduling steps and constructing fully integrated scheduling models by simultaneously considering two or more individual sub-problems. The potential benefits of integrated scheduling were estimated in [10]. Other work on integrated scheduling includes [11–14]. Research on *robust scheduling* investigates new approaches to build schedules that have an improved performance in operation. The *robustness of a schedule* is influenced by its *sensitivity* to stochastic events, the flexibility within the schedule and its stability. The *flexibility* is related to the number of recovery options available to mitigate the effects of a disruption, whereas the *stability* of the schedule is a measure for the probability of a delay to propagate through the schedule and the availability of local recovery strategies with a limited impact on the rest of the schedule. Previously considered approaches for robust airline scheduling include [15–18]. An analytical approach to evaluate the robustness of airline schedules based on the presence of robustness influencing features in the schedules—called *robustness objectives*—was presented in [19]. Approaches that focus on improving a schedule's robustness by manipulating individual robustness objectives were presented in [3,4,20–26]. An important shortcoming of the latter work is the fact that robustness objectives are considered in isolation. However, multiple robustness objectives are likely to mutually interact and simultaneously influence the schedule's operational performance [4,15]. The construction of a robust airline schedule should therefore be considered as a multi-objective optimisation problem that generates schedules with a good trade-off between the individual robustness objectives, resulting in better operational performance.

In this paper, we present an approach for *multi-objective optimisation of robustness objectives* in airline schedules. Multi-objective optimisation of robustness objectives is a promising area of research that has, to the best of the authors' knowledge, not been addressed before. The presented approach enables us to investigate *the mutual interaction between robustness objectives* and quantify *their simultaneous influence on the schedule's operational performance* (through large scale simulation). The approach enables us to identify *dominant influences* and *good trade-offs* between robustness objectives that result in a better operational performance. Multi-objective improvement is obtained by simultaneously *retiming flights* and *rerouting aircraft* in existing schedules, subject to a fixed fleet assignment. Diverse approximations of the Pareto optimal front are obtained by applying a multi-meme memetic algorithm. Experimental results for real world data from KLM Royal Dutch Airlines are presented. The approach can be applied by schedule operators in the network and operational plan management department.

The outline of our paper is as follows. Section 2 provides a detailed description of the underlying network model and the robustness objectives that we investigate. Section 3 describes the algorithmic approach and is followed by a discussion of the results that were obtained for real world data from KLM in Section 4. We conclude and focus on our future work in the final section of this paper.

2. Research approach

Due to the multi-objective nature of the problem, we propose Pareto optimisation [27–29] for the simultaneous improvement of multiple robustness objectives in existing airline schedules. In Pareto optimisation, each of the solutions x in the decision space χ has a vector $z(x)=\{z_1(x), z_2(x), \ldots, z_k(x)\}$ of objective values that represents the trade-off between the objectives. The *Pareto optimal front* is the set of solutions that contains all solutions that are not dominated by any other solution in the entire feasible search space. A solution x_1 dominates x_2 if none of the components in x_1 is worse than the corresponding value in x_2 and at least one of the components in x_1 is strictly better than its corresponding value in x_2 [29]. In the context of our work, the Pareto optimal front represents the set of schedules with an optimal trade-off between the individual robustness objectives. Details on the robustness objectives that we investigate and the underlying network model in our approach are provided below.

2.1. Model

Our model, of which a graphic representation is shown in Fig. 1, is based on a *weekly time-space multi-commodity flow network*. Similar models were presented in [12,30–32]. Each commodity represents an aircraft that circulates through the network. Each airport in the network is represented by a timeline that models its activities (e.g. C_1 in Fig. 1). Timelines are connected through *flight arcs* that represent non-stop flights between them (e.g. f_1 and f_2 in Fig. 1). A *reserve arc* connects two points at the same timeline and defines an interval during which an aircraft is allocated as a spare at the given location (e.g. r_7 in Fig. 1). *Ground arcs* are used to connect two flight arcs or a flight arc with a reserve arc (e.g. between f_1 and f_2 or between f_6 and r_7 in Fig. 1). *Wrap around arcs*—not shown in Fig. 1—connect the last arc of a rotation to the first arc of the rotation and thereby enforce periodic schedules. The exact aircraft that is allocated to an arc—and thus the fleet type that is assigned—is determined by the

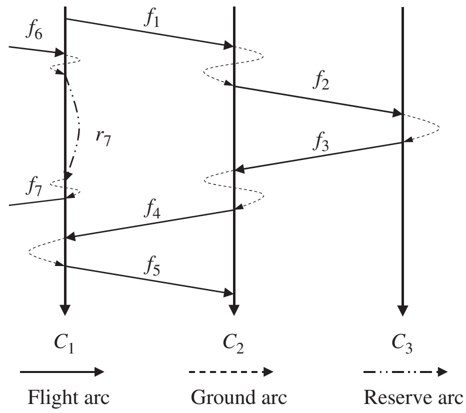

Fig. 1. Time-expanded multi-commodity flow network.

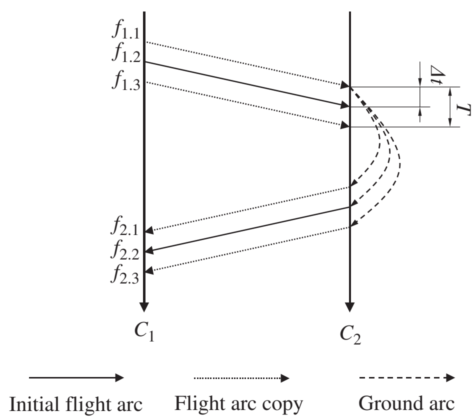

Fig. 2. Time-windows and arc copies, $\mathscr{K}=3$, $\Delta t=5$ min, $\mathscr{T}=\pm 5$ min.

aircraft that is used to carry out the first arc of the corresponding rotation.

Flexibility in the flight times is modelled by making $\mathscr{K}-1$ arc copies of each individual flight arc, as shown in Fig. 2. Arc copies are spaced at regular intervals Δt within a pre-specified time window of size $\mathscr{T}$. The time windows are centred around the initial flight arc. Keeping the size of $\mathscr{T}$ small enables us to discard the impact of retimings on the passenger demand, the passenger connections and the profitability of the schedule. In the case of larger values for $\mathscr{T}$, changes to passenger demand, the passenger connections and the market value of the itineraries (as a consequence of the retimings) must be taken into account. Multi-objective improvement is obtained by making a well-considered selection of arc copies and the connections between them. A feasible solution is one that does not violate any network and schedule feasibility constraints. The *network feasibility constraints* include flow balance and aircraft availability, whereas the *schedule feasibility constraints* include minimum ground time restrictions, maintenance feasibility and coverage constraints. The latter two require that each plane undergoes maintenance at least every 60 flight hours and that at least one of the arc copies for each flight in the initial schedule is selected.

2.2. Objectives

2.2.1. Schedule reliability

The reliability objective $\mathscr{R}(\mathscr{S})$ for a schedule $\mathscr{S}$ is a measure of the schedule's ability to absorb the effects of minor stochastic influences in its operating environment. $\mathscr{R}(\mathscr{S})$ is defined in terms of the probability p_i that the next flight (f_i+1) of an aircraft can leave on time, given the time that is allocated to its previous flight (f_i) and its turn around operations (arrival and departure handling). The value of p_i is estimated based on stochastic distributions that are derived from large sets of historic data. The stochastic nature of flight times and departure handling is modelled by γ-distributions, whereas arrival handling is modelled by deterministic distributions. Systematically changing operating conditions—such as congestion patterns and weather influences—are modelled by defining different distributions depending on the time of the day, the time of the year, the origin/destination of the flight and the time that is scheduled to carry out the activities. The latter enables the modelling of a behavioural response. In addition, different distributions are defined for different types of aircraft. The stochastic nature of a sequence of activities, s_i, is estimated by calculating the approximate convolution $\mathscr{C}$ of the individual distributions [33]. s_i is thereby defined as the flight f_i—which connects to f_i+1—followed by its arrival and departure handling. The corresponding value of the cumulative distribution function for $\mathscr{C}$ for the time t_i that is allocated to s_i, $\mathscr{C}(t_i)$, provides an estimation of p_i.

The value of the reliability objective $\mathscr{R}_i(\mathscr{S})$ for s_i is equal to the scaled value of p_i, as defined by Eqs. (1) and (2). The non-linear scaling functions encourage improvement of the weakest links in a schedule by penalising low values of p_i more severely. An additional penalty is incurred in case p_i violates the minimum reliability p_{min}. The scaling parameter y in Eqs. (1) and (2) enables us to manipulate the quadratic shape of the scaling function and influences the ratio of the objective value for less reliable connections versus more reliable connections. The parameter P in Eq. (2) controls the penalty that is incurred in case p_i violates p_{min}. Low values for y and P penalise larger violations of p_i less severely and encourage exploration of the search space by facilitating crossing fitness barriers and migration to new, possibly better, regions of the search space. The overall schedule reliability $\mathscr{R}(\mathscr{S})$ is calculated as the sum of the individual values of $\mathscr{R}_i(\mathscr{S})$ for all s_i in $\mathscr{S}$, as defined by Eq. (3). Notice that $|f|$ represents the total number of flights in $\mathscr{S}$

$$p_i > p_{min} : \mathscr{R}_i(\mathscr{S}) = f(p_i) = (1-p_i)^y \tag{1}$$

$$p_i \leqslant p_{min} : \mathscr{R}_i(\mathscr{S}) = f(p_i) = (1-p_i)^y + P(p_{min}-p_i) \tag{2}$$

$$\mathscr{R}_i(\mathscr{S}) = \sum_{i=0}^{|f|} \mathscr{R}_i(\mathscr{S}) \tag{3}$$

2.2.2. Schedule flexibility

The flexibility objective $\mathscr{F}(\mathscr{S})$ for a schedule $\mathscr{S}$ is defined in terms of the number of single point swap opportunities in $\mathscr{S}$. A *single point swap* is defined as a sufficiently long overlap between the ground time of two sequences, s_i and s_j, that enables a feasible exchange of their aircraft. Swap opportunities are typically used on the day of operation to reduce the impact of a disruption by redistributing slack between the aircraft rotations. Implementing a single point swap between s_i and s_j implies that the aircraft that was allocated to f_i will carry out f_{j+1} and the one that was allocated to f_j will carry out f_{i+1}. Three different types of single point swap opportunities are illustrated in Fig. 3. Notice that g_i, g_j, $g_{i\to j+1}$ and $g_{j\to i+1}$, respectively, denote the scheduled ground time between f_i and f_{i+1}, f_j and f_{j+1}, f_i and f_{j+1}, and f_j and f_{i+1}. As can be seen from Fig. 3, implementing a *type* 1 and *type* 2 swap enables the creation of additional ground time, Δg, respectively, for f_i and f_j that could prevent knock-on delays in case f_i, respectively, f_j, is delayed. *Type* 3 swaps do not allow the creation of additional ground time for any of the flights involved, yet could be used to exchange aircraft on the day of operation to meet maintenance requirements, to swap the aircraft back into their original position, or to prevent a delay of the most important flight of f_{i+1} and f_{j+1} in case one of the inbound flights is delayed.

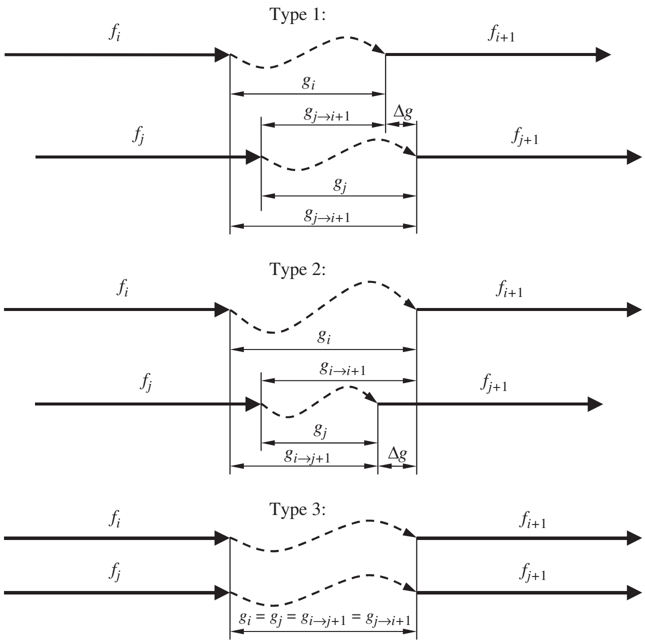

Fig. 3. Swap types.

The objective value $\mathscr{F}_{i,j}(\mathscr{S})$ for a single point swap between sequences s_i and s_j is defined by Eqs. (4)–(6). In case of a *type* 1 or *type* 2 swap, $\mathscr{F}_{i,j}(\mathscr{S})$ is, respectively, equal to $p_{j\to i+1}$ or $p_{i\to j+1}$. In the case of a *type* 3 swap, $\mathscr{F}_{i,j}(\mathscr{S})$ is equal to the minimum of $p_{i\to j+1}$ and $p_{j\to i+1}$. Notice that $p_{i\to j+1}$ and $p_{j\to i+1}$ represent the probability of an on-time departure of f_{j+1} and f_{i+1} after implementing the swap. The value of the flexibility objective for the whole schedule, $\mathscr{F}(\mathscr{S})$, is equal to the sum of the individual values for $\mathscr{F}_{i,j}(\mathscr{S})$, as defined by Eq. (7)

$$type\ 1 : g_{j\to i+1} < g_j : \mathscr{F}_{i,j}(\mathscr{S}) = p_{j\to i+1} \quad (4)$$

$$type\ 2 : g_{i\to j+1} < g_i : \mathscr{F}_{i,j}(\mathscr{S}) = p_{i\to j+1} \quad (5)$$

$$type\ 3 : g_i = g_j = g_{i\to j+1} = g_{j\to i+1} : \mathscr{F}_{i,j}(\mathscr{S}) = min(p_{i\to j+1}, p_{j\to i+1}) \quad (6)$$

$$\mathscr{F}(\mathscr{S}) = \sum_{i=0}^{|f|} \sum_{j=i+1}^{|f|} \mathscr{F}_{i,j}(\mathscr{S}) \quad (7)$$

3. Algorithmic approach

Our search methodology is based on a hybridisation of genetic algorithms with local search—also referred to as *genetic local search* [34,35], *hybrid genetic algorithms* [36] and *memetic algorithms* [37] in the literature. In comparison to many of the classical approaches for multi-objective optimisation (e.g. based on mathematical programming [38,39]), evolutionary approaches are less susceptible to concave or discontinuous Pareto fronts and enable us to evolve a set of solutions in one single run of the algorithm. In addition, genetic algorithms are highly valued for their ability to quickly locate promising and diverse regions of a search space [40], yet they can have difficulties in obtaining local improvements [37]. Hybridising genetic algorithms with local search facilitates local improvement of genetic material and has been effective for scheduling problems [41]. Yet, care must be taken to maintain the delicate balance between the exploration and the exploitation of the search space. Careless hybridisation of genetic algorithms could result in premature convergence of the search and might result in poor solutions [42,43]. Examples of hybridisations of genetic algorithms with multiple local search operators—often called *multi-meme memetic algorithms* in the literature—can be found in [44,45]. Hybridisations with local search operators that focus on the improvement of individual objectives were presented in [46,47]. Multi-meme hybridisations facilitate escaping from local optima and improve the robustness of the algorithm [37]. Details of our multi-meme memetic algorithm, including a detailed description of our genetic encoding, the genetic operators, the local search and the adaptive grid archiving algorithm are provided below.

Our genetic encoding is based on the adjacency representation that was first applied to the travelling salesman problem in [48]. The genetic encoding for a single aircraft rotation is presented in Fig. 4. Each gene in the chromosome represents a connection between two non-ground arcs in the network. The inbound arc of the connection is determined by the position of the gene within the chromosome and is equal to the arc that can be found on the corresponding position in the arc list (see Fig. 4). The *arc list* contains all flight and reserve arcs of the network and has a fixed order. Each gene has an *arc index* that determines the position of its outbound arc in the arc list and an *arc copy index* that determines the exact arc copy that is selected for its inbound arc. The aircraft that is allocated to an arc is determined by the aircraft that is assigned to the first arc of its rotation, as shown in Fig. 4.

The population $\mathscr{P}$ *is initialised by selecting random arc copy indices and making random changes to the aircraft rotations* (*by making swap moves—which are defined below*) *in an initial feasible schedule provided by schedule operators*. Each chromosome in $\mathscr{P}$ contains a random selection of arc indices and arc copy indices that meet the network feasibility and the coverage constraints. Each chromosome in $\mathscr{P}$ corresponds to a schedule with different departure times, arrival times and aircraft rotations. The fleet assignment of the chromosomes is fixed and equal to the fleet assignment of the initial schedule.

A *mating pool* $\mathscr{M}_n$ of size $|\mathscr{P}|$ is generated at every iteration n of the algorithm using binary tournament selection. Two chromosomes, x_1 and x_2, are randomly selected from $\mathscr{P}_{n-1}$ and compete against each other. Competition is based on domination. If x_1 dominates x_2, x_1 is added to $\mathscr{M}_n$, and the other way around. If x_1 and x_2 are non-dominated with respect to each other, one of them is selected at random and added to $\mathscr{M}_n$. The resulting mating pool $\mathscr{M}_n$ is likely to have an above average fitness in comparison to the population $\mathscr{P}_{n-1}$.

Our recombination operator randomly selects two chromosomes, x_1 and x_2, from $\mathscr{M}_n$ and generates two offspring, o_1 and o_2, using single point crossover. The resulting offspring contain a sequence of genetic material (that corresponds to a partial schedule) from both their parents. A *cyclic repair heuristic* is applied if an offspring violates the network feasibility constraints. This occurs, for instance, when multiple inbound arcs connect to the same outbound arc or when an arc has no inbound or outbound connection. Infeasibility is likely to occur for connections where the position of the inbound and the outbound arcs in the arc list is located on different sides of the crossover point. In the case of the example shown in Fig. 5, this occurs for the connection between f_{i-3} and f_{i+2}, represented by the gene at position $i-3$ in chromosome x_1. Single point crossover between x_1 and x_2, taking the first genetic sequence from x_1 and the second one from x_2, results in an infeasible offspring ($o_{1,infeasible}$): both f_{i-3} and f_{i-1} connect to f_{i+2} (illustrated in Fig. 5). The repair cycle is initiated by randomly selecting one of the conflict positions in one of the parents, e.g. position $i-1$ in x_2 (denoted by $x_{2_{i-1}}$). The repair heuristic iteratively searches for the value at the currently selected position ($=i+2$) in the other parent (x_1) and then selects the value at the corresponding position in the selected parent (x_2). This continues until the initial position ($x_{2_{i-1}}$) is reached again. In the case of Fig. 5, this results in the cycle $x_{2_{i-1}} \to x_{1_{i-3}} \to x_{2_{i-3}} \to x_{1_{i-2}} \to x_{2_{i-2}} \to x_{1_{i-1}} \to x_{2_{i-1}}$. Replacing the values of $o_{1,infeasible}$ by the values that are found at the corresponding positions of the selected parent (x_2) results in a feasible offspring, $o_{1,feasible}$. The repair procedure is

Please cite this article as: Burke EK, et al. A multi-objective approach for robust airline scheduling, Computers and Operations Research (2009), doi:10.1016/j.cor.2009.03.026

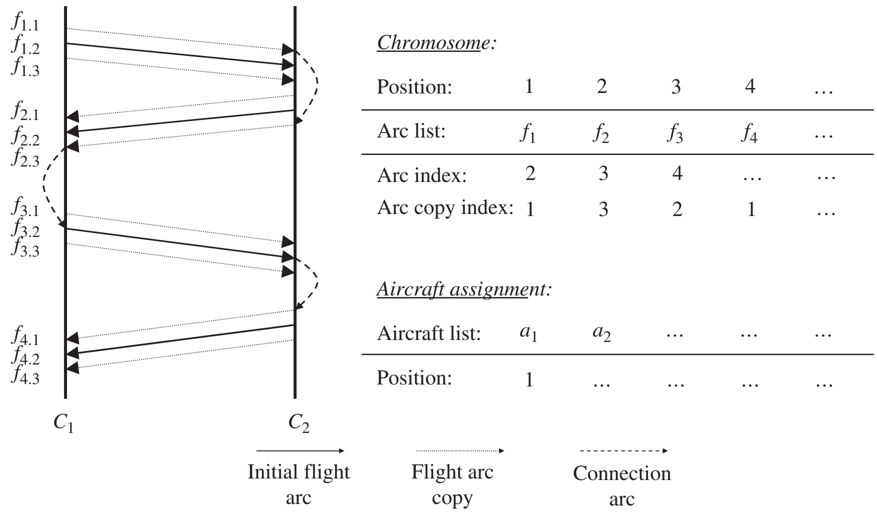

Fig. 4. Genetic representation.

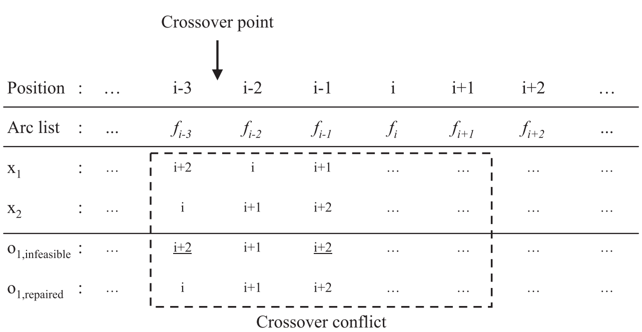

Fig. 5. Repair heuristic for solving a crossover conflict (no arc copy indices shown).

followed by a mutation phase in which each of the genes in the offspring is subject to mutation with a probability p_m. Our mutation operator is responsible for making random *arc copy* and *swap moves* (defined below).

After mutation, the offspring are subject to local search with a probability pl_{sc}. Once a chromosome is selected for local search, each gene in the chromosome has a probability pl_{sg} to be selected. A selected gene is used to initiate the neighbourhood of the local search operator by selecting $\mathcal{N}-1$ connections centred around it. The start time of the inbound arc of the left-most connection, t_{left}, and the finish time of the right-most connection, t_{right}, place a time frame $[t_{left}, t_{right}]$ on the chromosome/schedule that marks the neighbourhood/partial schedule that is subject to local search.

Our local search operators make random arc copy moves and swap moves. An *arc copy move* randomly increments or decrements the arc copy index of a gene. An arc copy move at position i thereby influences the ground time of the current connection, g_i, and its previous connection, g_{i-1}. This influences the value of $\mathcal{R}_i(\mathcal{S})$ and $\mathcal{R}_{i-1}(\mathcal{S})$, and can influence the value of $\mathcal{F}_{i,j}(\mathcal{S})$ and $\mathcal{F}_{i-1,j}(\mathcal{S})\forall j \in [0, |f|]$. A *swap move* involves a single point swap between two sequences s_i and s_j—carried out by aircraft of the same fleet type—and influences the value the ground times in the case of *type* 1 and *type* 2 swaps (see Fig. 3). A swap move can therefore influence the value of $\mathcal{R}_i(\mathcal{S})$, $\mathcal{R}_j(\mathcal{S})$, $\mathcal{F}_{i,k}(\mathcal{S})$ and $\mathcal{F}_{j,k}(\mathcal{S})$ for $\forall k \in [0, |f|]$.

The diversity of the population is encouraged by hybridising the underlying genetic algorithm with three different local search operators. A multi-objective local search operator, denoted by $LS_{\mathcal{R},\mathcal{F}}$, considers simultaneous improvement of $\mathcal{R}(\mathcal{S})$ and $\mathcal{F}(\mathcal{S})$ subject to the network feasibility and the schedule feasibility constraints. Two single objective local search operators, denoted by $LS_{\mathcal{R}}$ and $LS_{\mathcal{F}}$, consider the improvement of $\mathcal{R}(\mathcal{S})$ and $\mathcal{F}(\mathcal{S})$, respectively, subject to the feasibility constraints discussed above. In all cases, the local search is greedy and continues until the local optimum for the partial schedule, with respect to the considered objectives, is found. The acceptance of new solutions is based on domination.

Biased selection of the local search operators is applied to encourage exploration of the boundaries of the search space. The bias towards a certain local searcher is based on the relative position of the selected chromosome x (representing the schedule $\mathcal{S}$) with respect to boundaries of the objective space as they are known in the current stage of the search, denoted by $\mathcal{R}_{min}$, $\mathcal{R}_{max}$, $\mathcal{F}_{min}$ and $\mathcal{F}_{max}$. The relative position of x is defined by Eq. (8). Notice that $d_{\mathcal{R}(\mathcal{S})}$ and $d_{\mathcal{F}(\mathcal{S})}$ represent the normalised distance between $\mathcal{R}_{max}$ and $\mathcal{R}(\mathcal{S})$, respectively, $\mathcal{F}_{max}$ and $\mathcal{F}(\mathcal{S})$. The values of $d_{\mathcal{R}(\mathcal{S})}$ and $d_{\mathcal{F}(\mathcal{S})}$ are used to calculate the reliability and flexibility shares, denoted by $s_{\mathcal{R}(\mathcal{S})}$ and $s_{\mathcal{F}(\mathcal{S})}$, as defined by Eq. (9). The shares $s_{\mathcal{R}(\mathcal{S})}$ and $s_{\mathcal{F}(\mathcal{S})}$ are equally divided between the local search operators that consider $\mathcal{R}(\mathcal{S})$, respectively, $\mathcal{F}(\mathcal{S})$, as one of the objectives (see Eq. (10)). The values $s_{LS_{\mathcal{R},\mathcal{F}}}$, $s_{LS_{\mathcal{R}}}$ and $s_{LS_{\mathcal{F}}}$ determine the shares of the local search operators on the roulette wheel that is used to select a local searcher in our biased selection scheme

$$d_{\mathcal{R}(\mathcal{S})} = 1 - \frac{\mathcal{R}_{max} - \mathcal{R}(\mathcal{S})}{\mathcal{R}_{max} - \mathcal{R}_{min}}, \quad d_{\mathcal{F}(\mathcal{S})} = 1 - \frac{\mathcal{F}_{max} - \mathcal{F}(\mathcal{S})}{\mathcal{F}_{max} - \mathcal{F}_{min}} \tag{8}$$

$$s_{\mathcal{R}(\mathcal{S})} = \frac{d_{\mathcal{R}(\mathcal{S})}}{d_{\mathcal{R}(\mathcal{S})} + d_{\mathcal{F}(\mathcal{S})}}, \quad s_{\mathcal{F}(\mathcal{S})} = \frac{d_{\mathcal{F}(\mathcal{S})}}{d_{\mathcal{R}(\mathcal{S})} + d_{\mathcal{F}(\mathcal{S})}} \tag{9}$$

$$s_{LS_{\mathcal{R},\mathcal{F}}} = \frac{s_{\mathcal{R}(\mathcal{S})}}{2} + \frac{s_{\mathcal{F}(\mathcal{S})}}{2}, \quad s_{LS_{\mathcal{R}}} = \frac{s_{\mathcal{R}(\mathcal{S})}}{2}, \quad s_{LS_{\mathcal{F}}} = \frac{s_{\mathcal{F}(\mathcal{S})}}{2} \tag{10}$$

Finally, an archive, $\mathcal{A}$, is maintained as an external population of well-distributed and non-dominated solutions using the adaptive grid archiving algorithm [49]. This is necessary to prevent good solutions from getting lost due to the stochastic nature of genetic selection or non-improving moves with respect to one objective while a single objective local searcher improves the other objective. The k-dimensional objective space is divided into a fixed number of hypercubes—called grid regions—from which the bounds change over time depending on the distribution of the population in the objective space [49]. The grid region allocated to a chromosome depends on the value of its individual objectives. A crowding procedure is applied to improve the distribution of the different non-dominated grid regions. The pseudo-code that illustrates the implementation of our methodology is presented in Algorithm 1.

Algorithm 1. Pseudo-code

```
n = 1
initialise P_0;
initialise A_0;
while (n < numberOfGenerations) do
  while (|M_n| < |P|) do
    x_1 = selectRandomChromosome(P_{n-1})
    x_2 = selectRandomChromosome(P_{n-1})
    add(M_n, tournamentSelection(x_1, x_2));
  end while
  while (|P_n| < |P|) do
    x_1 = selectRandomChromosome(M_n);
    x_2 = selectRandomChromosome(M_n);
    offsprings[] = crossover(x_1, x_2);
    for (l = 0; l < |offsprings[]|; l = l + 1) do
      if (!feasible(offspring[l])) then
        repair(offspring[l]);
      end if
      if (random() < p_m) then
        mutate(offspring[l]);
      end if
      if (random() < pl_sc) then
        for (m = 0; m < |offsprings[l]|; m = m + 1) do
          if (random() < pl_sg) then
            LS = selectLocalSearcher(LS_{R,F}, LS_R, LS_F);
            apply(LS, offsprings[l], M);
          end if
        end for
      end if
      add(P_n, offsprings[l]);
    end for
  end while
  updateArchive(A_n, P_n);
  n = n + 1;
end while
```

4. Experimental results

4.1. Description of the datasets

The presented approach was applied to real world data from KLM Royal Dutch airlines. KLM provided us with their schedules for the second part of summer 2006 ($S06_2$) and the first part of winter 2006 ($W06_1$). The schedules were operated between the end of June 2006 and half of January 2007. The schedules are repeated on a weekly basis and have a dominant hub and spoke structure. The fleet used to carry out the schedule consists of four crew compatible subtypes of the same fleet type. Based on the initial schedules, eight new datasets were defined by selecting different subsets of aircraft rotations (carried out by aircraft of the same sub-type) from the initial schedules. An overview of the different datasets and their characteristics can be found in Table 1.

4.2. Multi-objective improvement

The presented approach was applied to the datasets described above to investigate the mutual interaction and simultaneous improvement of $\mathcal{R}(\mathcal{S})$ and $\mathcal{F}(\mathcal{S})$. The experiments were carried out with $\mathcal{T} = 20$ and $\mathcal{K} = 9$, resulting in a time window of ±10 min and $\Delta t = 2.5$ min. Each population contained 200 chromosomes and

Table 1
Description of the datasets.

Id	Initial schedule	# Rotations	# Flights and reserves arcs
D_1	$S06_2$	5	172
D_2	$S06_2$	14	445
D_3	$S06_2$	13	493
D_4	$S06_2$	14	496
D_5	$W06_1$	5	152
D_6	$W06_1$	15	473
D_7	$W06_1$	12	415
D_8	$W06_1$	13	504

was evolved for 1000 generations. The number of generations was set based on empirical testing. This showed that only marginal improvements of the approximated Pareto front were obtained after 1000 generations. Crossover and mutation probabilities were fixed and, respectively, equal to 1 and 0.01. The value of p_{min} was set to 0.7 and g_{min} was set to 40 min. Both values are based on common scheduling practices within airlines. The penalty parameters y and P were, respectively, set to 1.5 and 0.5. These values ensure that in all cases, improvement of less reliable sequences is preferred above the improvement of more reliable sequences. The local search rates, pl_{sc} and pl_{sg}, were both set to 0.01. Empirical testing has shown that low levels of local search (of the order of 0.01) result in significantly better approximations of the Pareto front and come at a minor increase in the computational cost. Random selection was compared with biased selection of the local search operators to enable us to quantify the potential benefits of a more sophisticated selection scheme. The neighbourhood size $\mathcal{N}$ for the local search operators was set to 5. All tests were carried out on the University of Nottingham's high performance computing facility, for which the specifications can be found in [50]. Each of the experiments was repeated 20 times, using different seeds for the random generator.

Figs. 6 and 7 show a graphical representation of the non-dominated solutions in the final archives obtained using biased and random selection of the local search operators. The corresponding schedules meet the network and schedule feasibility constraints. The results for random selection are summarised in Table 2. Notice that the 2nd and 3rd columns of Table 2 contain the objective values for the initial schedules, $\mathcal{R}(\mathcal{S}_0)$ and $\mathcal{F}(\mathcal{S}_0)$. The 4th and 5th columns show the values for $\Delta\mathcal{R}(\mathcal{S}) = \mathcal{R}(\mathcal{S}_0) - \mathcal{R}(\mathcal{S})$ and $\Delta\mathcal{F}(\mathcal{S}) = \mathcal{F}(\mathcal{S}) - \mathcal{F}(\mathcal{S}_0)$, with $\mathcal{S}$ the schedule of the non-dominated set that has a minimal distance between $\mathcal{F}(\mathcal{S})$ and $\mathcal{F}(\mathcal{S}_0)$, respectively, $\mathcal{R}(\mathcal{S})$ and $\mathcal{R}(\mathcal{S}_0)$. Notice that the values between the parentheses, denoted by $\mathcal{R}_{\Delta_{min}}(\mathcal{S})$ and $\mathcal{F}_{\Delta_{min}}(\mathcal{S})$, represent the values of $\mathcal{R}(\mathcal{S})$ and $\mathcal{F}(\mathcal{S})$ for the corresponding schedule. The boundaries of the objective space for the non-dominated set are shown in columns 6–9. Column 10 shows the p-value for the hypothesis "*random selection is better than biased selection*", calculated using the hypervolume indicator presented by Zitzler and Thiele [51] and the Mann–Whitney test. The final column shows the computational cost in minutes.

As can be seen from Figs. 6 and 7, the presented approach enables us to generate diverse sets of schedules with significant improvements for $\mathcal{R}(\mathcal{S})$ and $\mathcal{F}(\mathcal{S})$, compared to $\mathcal{S}_0$. A clear trade-off between $\mathcal{R}(\mathcal{S})$ and $\mathcal{F}(\mathcal{S})$ exists. Notice that, given the definition of the objectives, $\mathcal{R}(\mathcal{S})$ needs to be minimised whereas $\mathcal{F}(\mathcal{S})$ requires maximisation. The values for $\Delta\mathcal{R}(\mathcal{S})$ and $\Delta\mathcal{F}(\mathcal{S})$ (4th and 5th columns in Table 2) show that significant improvements for $\mathcal{R}(\mathcal{S})$ and $\mathcal{F}(\mathcal{S})$ can be obtained for "similar" values for $\mathcal{F}(\mathcal{S})$, respectively, $\mathcal{R}(\mathcal{S})$ compared to $\mathcal{S}_0$. It is observed from Figs. 6, 7, and column 10 in Table 2 that random meme selection performs equally well compared to biased selection of the local search operators. This can be explained by the fact that the local searchers create high quality sequences of genetic material that are locally optimal

Please cite this article as: Burke EK, et al. A multi-objective approach for robust airline scheduling, Computers and Operations Research (2009), doi:10.1016/j.cor.2009.03.026

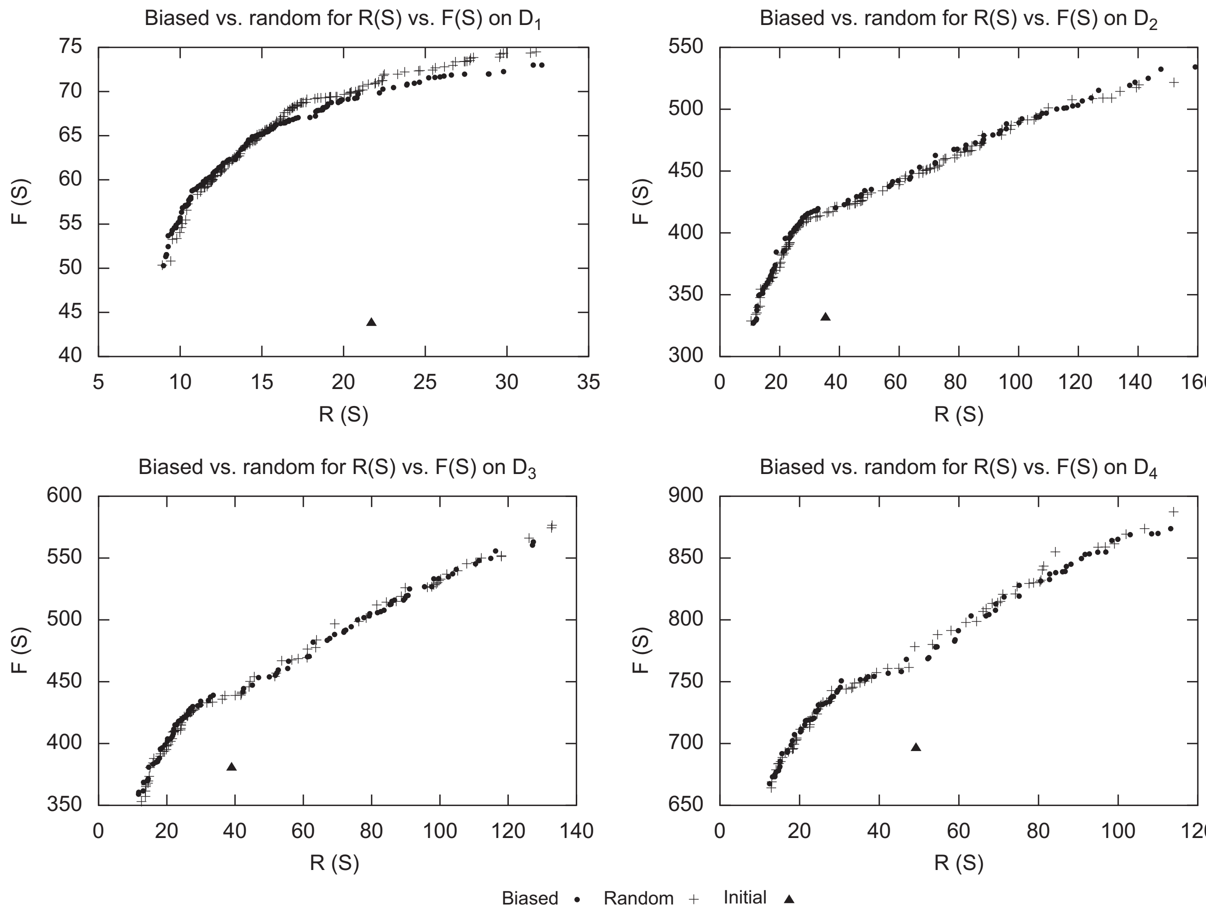

Fig. 6. Non-dominated solutions for D_1, D_2, D_3 and D_4.

and successfully migrate through the underlying genetic algorithm. The computation times shown in Table 2 provide an indication of the computational cost of our implementation which employed Java using a component based approach. This approach greatly enhances the flexibility of our code, yet slightly compromises its performance. Delta evaluation was implemented for $\mathscr{R}(\mathscr{S})$ and $\mathscr{F}(\mathscr{S})$ in combination with result caching for $\mathscr{R}(\mathscr{S})$. Significant improvements of the performance could be obtained by implementing result caching for $\mathscr{F}(\mathscr{S})$ and computing the different objectives in parallel. This is currently done sequentially by different components of the implementation.

4.3. Simulation results

The operational performance of the schedules in the final archives—for random meme selection—was estimated using KLM's simulation model [52]. The simulation model randomly generates disruptions to the input schedule and calls a problem solver in case the disruption would result in a (knock-on) delay. The problem solver applies a heuristic approach that is based on the current practices at KLM's operations control department. The recovery strategies include swapping aircraft, cancelling flights and accepting delays. The crew schedule is not explicitly taken into account in the simulation model. This is realistic, given the fact that KLM has reserve crew available to resolve crew problems. The output of the simulation model is a detailed overview of the 0, 5 and 15 minute arrival and departure punctuality of the schedule per fleet type, per location, and per flight number. The results presented below are based on the average 15 minute on-time performance for the whole schedule, denoted by $\mathscr{OTP}_{15}(\mathscr{S})$. The value of $\mathscr{OTP}_{15}(\mathscr{S})$ is equal to the average number of aircraft that has left/arrived not later than 15 minutes after the scheduled time. $\mathscr{OTP}_{15}(\mathscr{S})$ is a commonly used measure in the airline industry and is considered to be a good surrogate measure for the robustness of a schedule.

A graphical representation of the simulation results for D_5 and D_6 is shown in Figs. 8 and 9.[1] Notice that, in contrast to Figs. 6 and 7, the graphs show all schedules in the final archives. A summary of the simulation results for all datasets is provided in Table 3. The 2nd and 3rd columns in Table 3 represent the number of schedules that were included in the simulation study and the on-time performance for the initial schedule ($\mathscr{OTP}_{15}(\mathscr{S}_0)$). The 4th and the 5th columns show the averaged value for $\mathscr{OTP}_{15}(\mathscr{S})$ for the 20 schedules in the archives that are most similar to $\mathscr{S}_0$ with respect to $\mathscr{R}(\mathscr{S})$, and $\mathscr{F}(\mathscr{S})$, respectively. The last column of Table 3 shows the averaged on-time performance for the 20 most reliable schedules in the final archives (with the lowest values for $\mathscr{R}(\mathscr{S})$). Notice that the values between the parentheses in columns 4–6 represent the difference between the averaged value for $\mathscr{OTP}_{15}(\mathscr{S})$ and $\mathscr{OTP}_{15}(\mathscr{S}_0)$.

It can be seen from Figs. 8 and 9 that, with respect to the tradeoff between $\mathscr{R}(\mathscr{S})$ and $\mathscr{F}(\mathscr{S})$, the influence of $\mathscr{R}(\mathscr{S})$ is dominant. More reliable schedules result in a better operational performance in terms of $\mathscr{OTP}_{15}(\mathscr{S})$. We also found that the correlation shown

[1] Additional graphs for the other datasets are available from the corresponding author upon request.

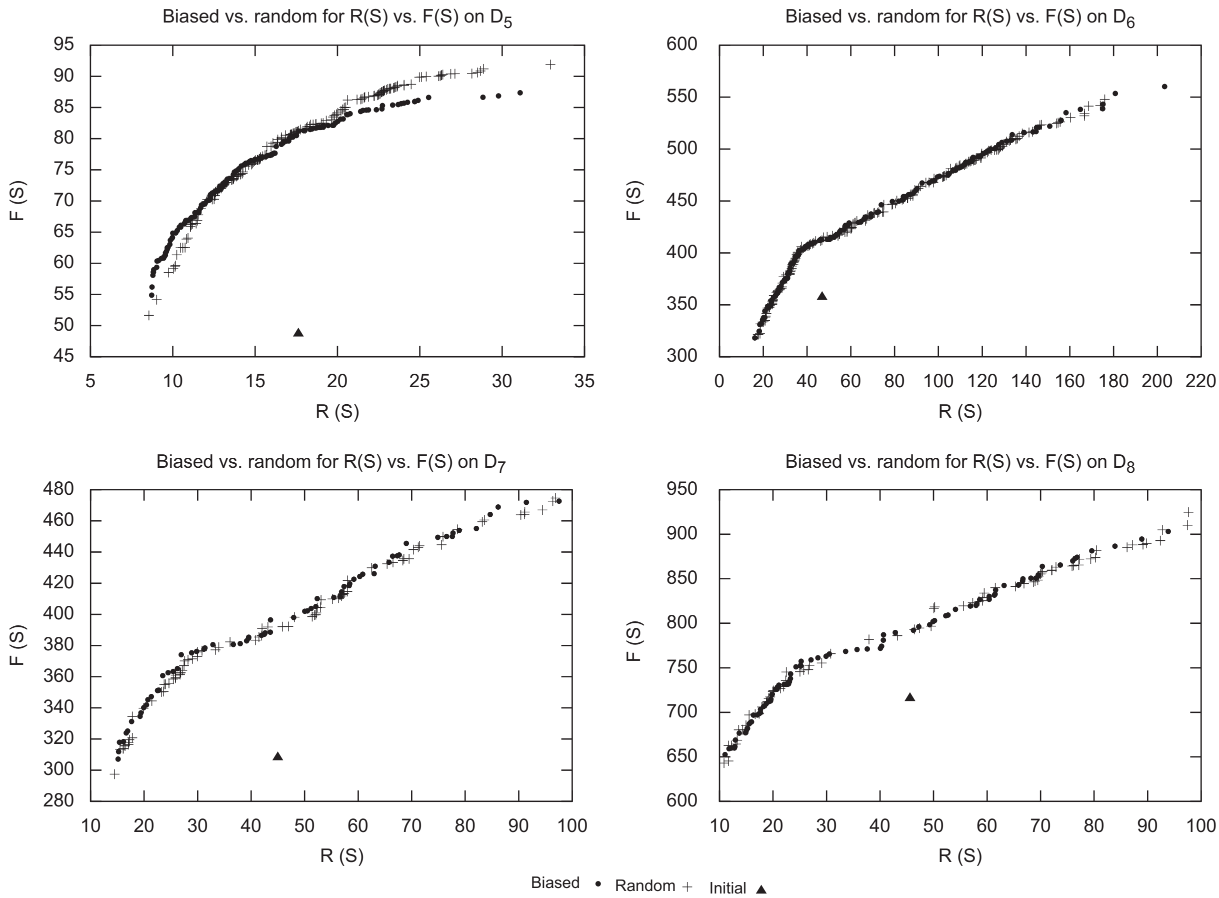

Fig. 7. Non-dominated solutions for D_5, D_6, D_7 and D_8.

Table 2
Summary of the improvement results for random selection.

Id	$\mathcal{R}(\mathcal{S}_0)$	$\mathcal{F}(\mathcal{S}_0)$	$\Delta\mathcal{R}(\mathcal{S})$ for $\mathcal{F}_{\Delta_{min}}(\mathcal{S})$	$\Delta\mathcal{F}(\mathcal{S})$ for $\mathcal{R}_{\Delta_{min}}(\mathcal{S})$	$\mathcal{R}_{min}$	$\mathcal{R}_{max}$	$\mathcal{F}_{min}$	$\mathcal{F}_{max}$	p-Value	t_{cpu}
D_1	21.7	43.8	12.8(50.3)	27.0(21.5)	8.9	31.8	50.3	74.5	0.971	7
D_2	35.4	331.2	24.9(328.7)	85.0(35.8)	10.4	152.0	328.7	521.7	0.667	60
D_3	39.0	380.3	23.9(380.5)	58.5(40.0)	12.6	132.9	353.0	576.5	0.165	72
D_4	49.2	696.0	30.8(695.9)	82.3(48.9)	12.9	113.9	664.1	887.4	0.304	96
D_5	17.6	48.7	9.1(51.6)	32.5(17.6)	8.5	32.9	51.6	91.9	0.821	7
D_6	46.8	357.4	22.0(359.0)	54.8(46.8)	17.5	176.0	321.3	547.9	0.543	60
D_7	45.0	308.1	29.5(313.4)	84.0(45.9)	14.5	96.9	297.4	474.6	0.554	72
D_8	45.6	715.7	26.4 (715.8)	77.4 (46.4)	10.9	97.7	643.1	924.7	0.935	98

for D_5 and D_6 was representative of all datasets that were addressed in the experiments. Apart for the smaller datasets (D_1 and D_5), the increased flexibility for schedules that are similar to $\mathcal{S}_0$ with respect to $\mathcal{R}(\mathcal{S})$ is shown to result in improved values for $\mathcal{OTP}_{15}(\mathcal{S})$ (Table 3—column 4, Figs. 8 and 9). In the case of D_1 and D_5, slightly worse values for $\mathcal{OTP}_{15}(\mathcal{S})$ are observed. This is likely to be caused by the increased variance in the simulation results that was observed for those datasets and the fact that the location of swap opportunities in smaller schedules is expected to play a more important role. Yet, the location is not taken into account in the definition of our flexibility objective. Finally, the increased reliability for schedules that are highly similar to $\mathcal{S}_0$ with respect to $\mathcal{F}(\mathcal{S})$ is shown to result in improved values for $\mathcal{OTP}_{15}(S)$ (5th column in Table 3). In all cases, apart from D_1, the better on-time performance was observed for the most reliable schedules in the final archives.

5. Conclusions and future research directions

We have presented an approach for multi-objective optimisation of robustness objectives in airline schedules. Multi-objective improvement is obtained by making incremental changes to the flight schedule and the aircraft rotation/maintenance schedule. The solution methodology is based on a multi-meme memetic algorithm. The presented approach was applied to real world data from KLM Royal Dutch Airlines to investigate reliability and flexibility improvements of their schedules. The results show that diverse sets with significant improvements for the considered objectives can be obtained. The operational performance of the improved schedules was estimated in a large scale simulation study. This enables us to quantify the influence of the robustness objectives. Sensitivity analysis of the results shows that the influence of schedule reliability is dominant

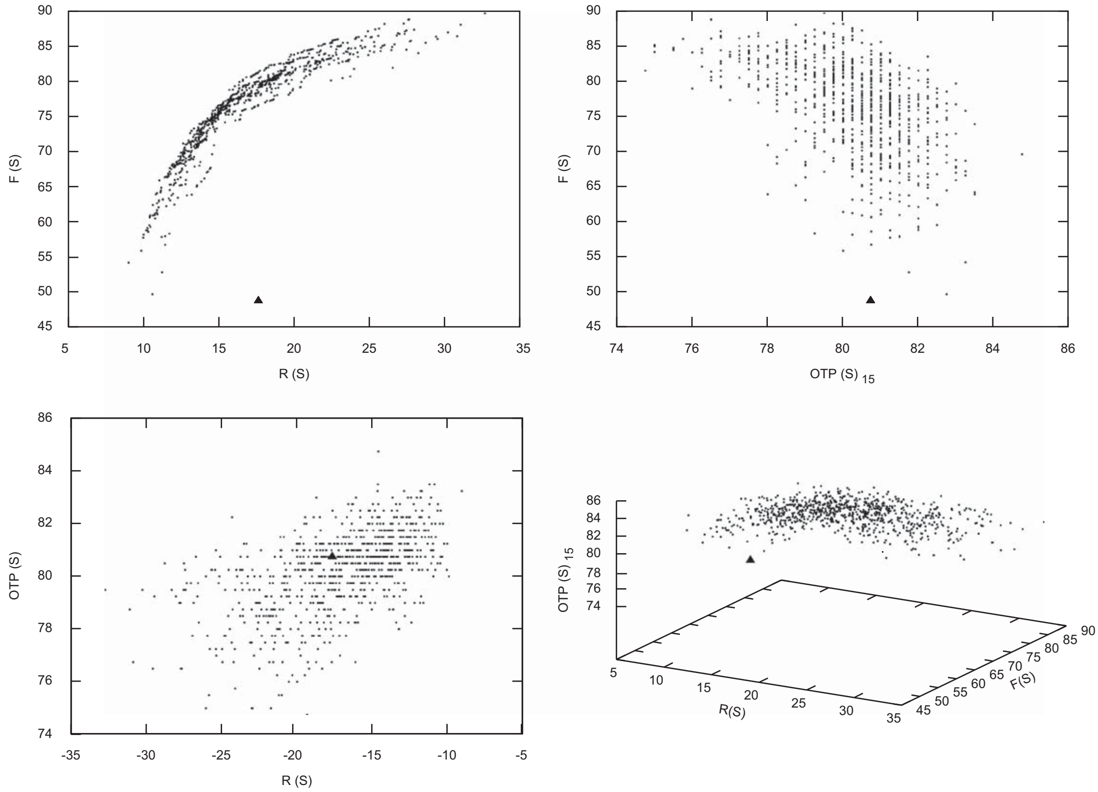

Fig. 8. Simulation results for D_5.

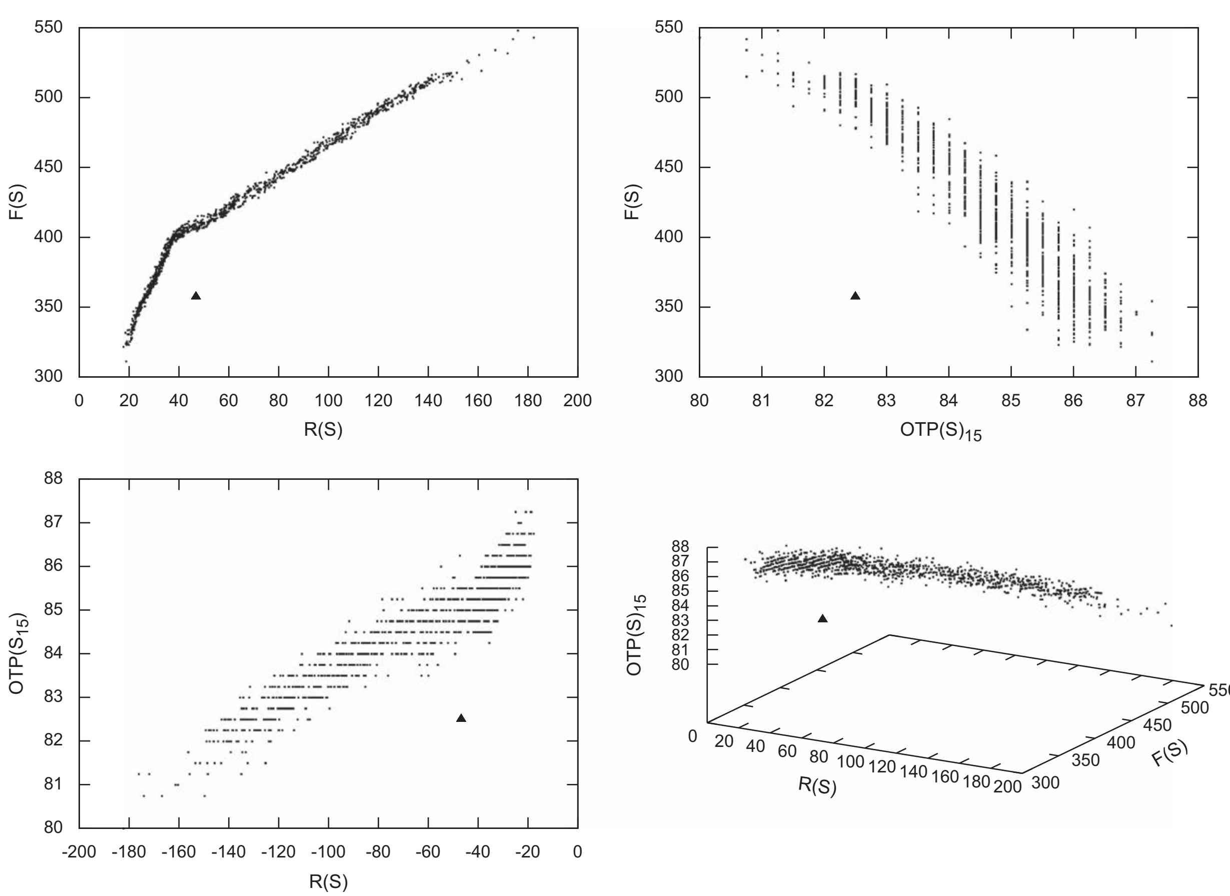

Fig. 9. Simulation results for D_6.

Table 3
Summary of the results using $\mathcal{OTP}_{15}(\mathcal{S})$.

Id	$\#\mathcal{S}$	$\mathcal{OTP}(\mathcal{S}_0)$	$\mathcal{OTP}_{15}(\mathcal{S})$ for $\mathcal{R}_{\Delta_{min}}(\mathcal{S})(\Delta)$	$\mathcal{OTP}_{15}(\mathcal{S})$ for $\mathcal{F}_{\Delta_{min}}(\mathcal{S})(\Delta)$	$\mathcal{OTP}_{15}(\mathcal{S})$ for $\mathcal{R}_{max}(\mathcal{S})(\Delta)$
D_1	1819	85.0	83.8(−1.2)	85.0(0.0)	84.8(−0.2)
D_2	899	85.2	85.4(0.1)	86.3(1.0)	86.3(1.1)
D_3	1224	86.2	86.8(0.6)	87.4(1.1)	87.6(1.3)
D_4	1664	88.5	89.1(0.6)	90.0(1.5)	90.1(1.6)
D_5	1750	80.8	79.7(−1.1)	81.3(0.6)	81.5(0.8)
D_6	1095	82.5	85.1(2.6)	86.1(3.6)	86.4(3.9)
D_7	2324	82.5	88.2(5.7)	89.6(7.2)	89.6(7.1)
D_8	1062	86.5	86.6(0.1)	87.5(1.0)	87.8(1.2)

and that increased flexibility could result in improved operational performance. We therefore recommend schedule operators to primarily focus on schedule reliability, and take schedule flexibility into account while building the schedule.

Our future work will focus on applying the presented approach with different robustness objectives. We will focus in particular on prioritising strategic flights, quantifying the influence of different flexibility patterns, and time/location based robustness. The latter recognises the different strategic value of the robustness during the course of the day and depending on the location. This will provide us with additional insights in the robustness of airline schedules and enable us to provide schedule operators at KLM with detailed feedback on how to further improve the robustness of their schedules and reduce the discomfort caused to their passengers.

References

[1] Delays to air transport in Europe: digest annual 2005. Technical Report, Eurocontrol, 2005.
[2] Eurocontrol medium-term forecast: IFR flight movements 2007–2013. Technical Report, Eurocontrol, 2006.
[3] Lan S, Clarke J, Barnhart C. Planning for robust airline operations: optimizing aircraft routings and flight departure times to minimize passenger disruptions. Transportation Science 2006;40:15–28.
[4] Rosenberger J, Johnson E, Nemhauser G. A robust fleet-assignment model with hub isolation and short cycles. Transportation Science 2004;38:357–68.
[5] Abara J. Applying integer linear programming to the fleet assignment problem. Interfaces 1989;19:20–8.
[6] Etschmaier M, Mathaisel D. Airline scheduling: an overview. Transportation Science 1985;19:127–38.
[7] Barnhart C, Belobaba P, Odoni A. Applications of operations research in air transport industry. Transportation Science 2003;37:368–91.
[8] Barnhart C, Cohn A. Airline schedule planning: accomplishments and opportunities. Manufacturing and Service Operations Management 2004;6: 3–22.
[9] Gopalakrishnan B, Johnson E. Airline crew scheduling: state-of-the-art. Annals of Operations Research 2005;140:305–37.
[10] Barnhart C, Lu F, Shenoi R. Integrated airline schedule planning. Dordrecht: Kluwer Academic Publishers; 1998 [chapter 9, p. 384–403].
[11] Klabjan D, Johnson E, Nemhauser G, Gelman E, Ramaswamy S. Airline crew scheduling with time window and plane-count constraints. Transportation Science 2002;36:337–48.
[12] Clarke L, Hane C, Johnson E, Nemhauser G. Maintenance and crew considerations in fleet assignment. Transportation Science 1996;30:249–60.
[13] Lohatepanont M, Barnhart C. Airline schedule planning: integrated models and algorithms for schedule design and fleet assignment. Transportation Science 2004;38:19–32.
[14] Mercier A, Soumis F. An integrated aircraft routing and crew scheduling model with time windows. Computers and Operations Research 2007;34:2251–65.
[15] Kang L. Degradable airline scheduling: an approach to improve operational robustness and differentiate service quality. PhD thesis, Sloan School of Management, Massachusetts Institute of Technology; 2004.
[16] Schaefer A, Johnson E, Kleywegt A, Nemhauser G. Airline crew scheduling under uncertainty. Transportation Science 2005;39:340–8.
[17] Yen J, Birge J. A stochastic programming approach to the airline crew scheduling problem. Transportation Science 2006;40:3–14.
[18] Lee L, Lee C, Tan Y. A multi-objective genetic algorithm for robust flight scheduling using simulation. European Journal of Operational Research 2007;117:1948–68.
[19] Bian F, Burke E, Jain S, Kendall G, Koole G, Landa-Silva J. et al. Making airline schedules more robust. Berlin: Springer; 2005 p. 381–92.
[20] Ageeva Y. Approaches to incorporate robustness into airline scheduling. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology; 2000.
[21] Rosenberger J. Topics in airline operations. PhD thesis, Georgia Institute of Technology, 2004.
[22] Smith B. Robust airline fleet assignment. PhD thesis, Georgia Institute of Technology, 2004.
[23] Smith B, Johnson E. Robust airline fleet assignment: imposing station purity using station decomposition. Transportation Science 2006;40:497–516.
[24] Wu C, Caves R. Towards the optimisation of the schedule reliability of aircraft rotations. Journal of Air Transport Management 2002;8:419–26.
[25] Ehrgott M, Ryan D. Constructing robust crew schedules with bicriteria optimization. Journal of Multicriteria Decision Analysis 2002;11:139–50.
[26] Shebalov S, Klabjan D. Robust airline crew pairing: move-up crew. Transportation Science 2006;40:300–12.
[27] Deb K. Multi-objective optimization using evolutionary algorithms. New York: Wiley; 2001.
[28] Landa Silva J, Burke E, Petrovic S. An introduction to multiobjective metaheuristics for scheduling and timetabling. Lecture notes in economics and mathematical systems, vol. 535. Berlin: Springer; 2004. [chapter 4, p. 91–129].
[29] Deb K. Multi-objective optimization. Berlin: Springer; 2005 [chapter 10, p. 273–316].
[30] Subramanian R, Scheff R, Quillinan J, Wiper S, Marsten R. Cold start fleet assignment at delta air lines. Interfaces 1994;24:104–20.
[31] Hane C, Johnson E, Marsten R, Nemhauser G, Sigismondi G. The fleet assignment problem: solving a large-scale integer program. Mathematical Programming 1995;70:211–32.
[32] Rushmeier R, Kontogiorgis S. Advances in the optimization of airline fleet assignment. Transportation Science 1997;31:159–69.
[33] Stewart T, Strijbosch L, Moors J, Batenburg PV. A simple approximation to the convolution of gamma distributions, center discussion paper, int. rep. 2006–27. Technical Report, University of Tilburg, 2006.
[34] Ishibuchi H, Murata T. A multi-objective genetic local search algorithm and its applications to flow shop scheduling. IEEE Transactions on Evolutionary Computation 1998;28:392–403.
[35] Jaszkiewicz A. Genetic local search for multi-objective combinatorial optimization. European Journal of Operational Research 2002;137:50–71.
[36] Sinha A, Chen Y, Goldberg D. Designing efficient genetic and evolutionary algorithm hybrids. Berlin: Springer; 2005 [chapter 12, p. 259–88].
[37] Krasnogor N, Smith J. A tutorial for competent memetic algorithms: model and taxonomy and design issues. IEEE Transactions on Evolutionary Computation 2005;9:474–88.
[38] Miettinen K. Nonlinear multiobjective optimisation. Boston, MA: Kluwer Academic Publishers; 1999.
[39] Ehrgott M. Multicriteria optimization. 2nd ed, Berlin: Springer; 2005.
[40] Sastry K, Goldberg D, Kendall G. Genetic algorithms. Berlin: Springer; 2005 [chapter 4, p. 97–126].
[41] Burke E, Landa-Silva J. The design of memetic algorithms for scheduling and timetabling problems. Berlin: Springer; 2004 [chapter 13, p. 289–312].
[42] Ishibuchi H, Yoshida T, Murata T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Transactions on Evolutionary Computation 2003;7:204–23.
[43] Ishibuchi H, Narukawa K. Performance evaluation of simple multiobjective genetic local search algorithms on multiobjective 0/1 knapsack problems. In: Proceedings of the 2004 congress on evolutionary computation (CEC 2004), IEEE Press, 2004. p. 441–48.
[44] Krasnogor N, Blackburne B, Burke E, Hirst J. Multimeme algorithms for protein structure prediction. In: Parallel problem solving from Nature—PPSN VII: seventh international conference. Berlin/Heidelberg: Springer; 2002. p. 769–88.
[45] Ong Y, Keane A. Meta-Lamarckian learning in memetic algorithms. IEEE Transactions on Evolutionary Computation 2004;8:99–110.
[46] Alkan A, Ozcan E. Memetic algorithms for timetabling. In: Proceedings of 2003 IEEE congress on evolutionary computation, 2003. p. 1796–1802.
[47] Ozcan E. An empirical investigation on memes, self-generation and nurse rostering. In: Proceedings of the 6th international conference on the practice and theory of automated timetabling, 2006. p. 246–63.

Please cite this article as: Burke EK, et al. A multi-objective approach for robust airline scheduling, Computers and Operations Research (2009), doi:10.1016/j.cor.2009.03.026

[48] Whitley D, Starkweather T, Fuquay D. Scheduling problems and the travelling salesman: the genetic edge recombination operator. In: Proceedings of the third international conference on genetic algorithms and their applications, 1989. p. 133–40.

[49] Knowles J, Corne D. Properties of an adaptive archiving algorithm for storing nondominated vectors. IEEE Transactions on Evolutionary Computation 2003;7:100–16.

[50] High performance computing, main cluster specifications, On-line resource, available at: ⟨http://www.nottingham.ac.uk/hpc/html/facility/specifications/index.php⟩.

[51] Zitzler E. Evolutionary algorithms for multiobjective optimization: methods and applications. PhD thesis, ETH Zurich, Switzerland; 1999.

[52] Jacobs P, Verbraeck A, Mulder J. Flight scheduling at KLM. In: Proceedings of the winter simulation conference, 2005. p. 299–306.

Please cite this article as: Burke EK, et al. A multi-objective approach for robust airline scheduling, Computers and Operations Research (2009), doi:10.1016/j.cor.2009.03.026