

Drexl, Andreas; Nikulin, Yuri

Working Paper — Digitized Version

Multicriteria time window-constrained project scheduling with applications to airport gate assignment. Part I: Methodology

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 595

Provided in Cooperation with:

Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Drexl, Andreas; Nikulin, Yuri (2005) : Multicriteria time window-constrained project scheduling with applications to airport gate assignment. Part I: Methodology, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 595, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/147653>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Manuskripte
aus den
Instituten für Betriebswirtschaftslehre
der Universität Kiel

No. 595

**Multicriteria time window-constrained project scheduling with applications to
airport gate assignment. Part I: Methodology¹**

Andreas Drexl and Yury Nikulin²

Working Paper, July 2005

Christian-Albrechts-Universität zu Kiel,
Institut für Betriebswirtschaftslehre,
Olshausenstr 40, 24118 Kiel, Germany

¹This work has been supported by the German Science Foundation (DFG) through the grant "Planung der Bodenabfertigung an Flughäfen" (Dr 170/9-1, 9-2).

²Corresponding author, nkln@lycos.com

Abstract

This paper addresses the airport flight gate assignment problem with multiple objectives. The objectives are to maximize the total flight gate preferences, to minimize the number of towing activities and to minimize the absolute deviation of the new gate assignment from a so-called reference schedule. The problem examined is a multicriteria multi-mode resource-constrained project scheduling problem with generalized precedence constraints or time windows. While in previous approaches the problem has been simplified to a single objective counterpart, we tackle it directly by a multicriteria metaheuristic, namely Pareto Simulated Annealing, in order to get a representative approximation of the Pareto front.

Keywords: airport gate assignment, multiple criteria optimization, project scheduling, time windows, Pareto simulated annealing.

1 Introduction

Scheduling problems arising in practice almost always have a multiple criteria nature, a fact which was indicated in [40] for the first time. Nevertheless until recently, research in scheduling continued to deal mainly with single objective problems and, hence, scheduling theory and multicriteria optimization so far have developed separately. One of the first attempts to improve the situation and to incorporate multicriteria approaches into scheduling theory was done in [48]. In this research it has been shown that a solution which is optimal with respect to one single objective might be arbitrarily bad with respect to other criteria and thus will be unacceptable for the decision maker. As a consequence it can be generally concluded that any problem arising in practice can hardly be properly solved without multicriteria optimization techniques. Nowadays this is commonly accepted, a fact which will stipulate scheduling research within the multiple criteria framework.

Due to the growth of air transport traffic (it has roughly doubled since the early 1980s) techniques for managing and allocating airport and airline resources in a dynamic operational environment effectively and efficiently have gained an ever-increasing interest. Strong competition between airlines and the demand of passengers for more comfort have lead to complex planning problems that require new models and methods. The scheduling problems nowadays faced by airport and airline managers are even more complicated than most other traditional scheduling problems. Classical flight gate scheduling deals with assigning aircraft to their terminal stand positions, which are known as 'gates'. We refer the reader to [17] where a comprehensive survey of existed models and solution techniques is presented.

The most recent models represent the flight gate scheduling problem (FGSP) as a quadratic assignment problem (QAP) with multiple objectives (see, e.g., [13, 14, 15, 49, 50]) or as a resource- and time window-constrained project scheduling problem (RCPSP; see, for instance, [16]). It should be emphasized, that these two distinct models generally highlight two different strategies adopted in the major airports of the United States and the European Union, respectively. While these models have been developed for different application domains both adopt multiple criteria. Moreover, both are very hard to solve even in the single objective case. Therefore, for practical purposes, it seems to be promising to combine different well-known metaheuristic approaches (simulated annealing, tabu search, genetic algorithms) providing a good approximation of optimal solutions with acceptable running times with multicriteria optimization concepts. In [21], for example, such an approach has been developed for FGSP modelled as QAP with multiple objectives. In this paper we elaborate such an approach for FGSP modelled as RCPSP with multiple objectives.

The paper is organized as follows. In section 2 the problem is formulated mathematically. In section 3 a survey and synthesis of algorithms developed for RCPSP type models is given. Section 4 presents a detailed description of a particular multicriteria procedure. A short summary and some conclusions can be found in section 5. Results of computational experiments will be presented in part II of the paper, which is issued separately.

2 Mathematical model

2.1 General description

Traditionally, a well-constructed flight gate assignment must satisfy two restrictions:

- no two aircraft may be assigned to the same gate simultaneously, i.e. one gate can process only one aircraft at the same time,
- every flight must be assigned to exactly one gate, that is, an aircraft cannot be moved or reassigned to another position once it has been located at a terminal gate.

Note that the models with such strict restrictions represent the strategy adopted for United States airports. This strategy considers arrival, departure and intermediate parking stages as a single non-split entity to be assigned to the same position.

In the model presented in this paper we relax the assumption that a flight has to be assigned as a whole to one and only one gate. The model uses a fairly large number of apron stands for passenger embarking and disembarking reflecting scarce terminal space. As indicated above such a model covers the situation encountered frequently at European airports.

Sometimes, special constraints are introduced mainly because of a particular airport configuration. The most commonly used such constraint is that any two large-type aircraft cannot be assigned to neighboring gates. Below we will refer to such constraints as 'shadow' restrictions. The mathematical formulation of the problem we consider in this paper was originally proposed in [16]. The purpose of the model is to assign three possible aircraft activities (arrival; optional intermediate parking, the length of which depends on the ground time; departure) to the available airport flight gates and to schedule start and completion times of the activities at the positions.

The model has several new ingredients:

- First, the activities are modeled separately and, hence, can potentially be assigned to different positions. The aircraft can be moved to another position using tow tractors, a procedure which is called towing.
- Second, in contrast to the standard objective function commonly used (which minimizes passenger walking distance) a multiple objectives formulation is introduced. As indicated above three objectives were considered to be most important:
 - maximization of total flight-gate preferences,
 - minimization of the number of towing activities,
 - minimization of the absolute deviation of the new gate assignment from a so-called reference schedule

The first and second objective are oriented towards convenience for airport services, whereas the third objective takes into account passenger comfort. The overall goal is to optimize all the objectives simultaneously. In fact, the multiple criteria nature of the problem makes it very unlikely that a so-called ideal optimal solution which simultaneously optimizes all objectives does exist – and can be found and verified in reasonable time. Therefore, one has to determine a solution that provides an appropriate compromise between all the different objectives while assuring a set of hard constraints.

In the following subsection a mathematical formalization of the problem is presented.

2.2 Model formulation

Let n be the number of aircraft (flights) and m be the number of terminal stand positions (gates). Each aircraft activity (arrival, departure or parking) i can be described by its start time S_i and by its completion time C_i . The start time for an arrival activity t_i^a and the completion time of a departure activity t_i^d are fixed and given a priori according to some time-table. All other start and completion times are decision variables of the model. Thus, we have $3n$ aircraft activities, and $4n$ decision variables related to time. Additionally, we have n implicit variables related to gate assignment. In total the number of variables is equal to $5n$.

Let V denote the set of all activities as a unification of the sets of arrival, parking and departure activities, respectively, that is, $V = V^a \cup V^p \cup V^d$. Each activity i has a minimum required processing time p_i^{\min} . Activity i can be assigned to different flight gates (that is, can be processed in different modes) M_i from the associated mode set \mathcal{M}_i which is a subset of the set of all possible modes \mathcal{M} . To cope with the situation where the constraints do not allow to assign all aircraft to real gates, a fictitious gate 0 with unlimited capacity is introduced. Every set \mathcal{M}_i contains this dummy gate, and an assignment to the dummy gate will be penalized in the objective functions.

If two interrelated (linked) activities are assigned to different flight gates, then they require a towing procedure in order to be moved from one position to another. Two activities are linked if they are subsequently served by the same aircraft (e.g. arrival/parking or parking/departure). Towing activity i assigned to gate M_i to activity j assigned to gate M_j takes some fixed processing time $d_{iM_ijM_j}^{\text{tow}}$. Let ε^{tow} represent the set of linked activities. It follows that the completion and start times of two linked activities i and j should satisfy the equality $C_i + d_{iM_ijM_j}^{\text{tow}} = S_j$ to provide continuous processing. In fact, this means that among all $4n$ decision variables only $2n$ variables are independent. The values $d_{iM_ijM_j}^{\text{tow}}$ are usually defined by a function $d_{iM_ijM_j}^{\text{tow}} := f^{\text{tow}}(i, M_i, j, M_j)$.

Gates are disjunctive resources that can only process one aircraft at a time (the only exception is, of course, dummy gate 0). Between the processing of two activities i and j at the same gate a fixed setup time $d_{iM_ijM_j}^{\text{setup}} \in \mathbb{N}_0$ must pass. The setup time can reflect the time required to push the first aircraft back from the gate and for moving the second aircraft to the gate as well as the time required for setting up gate equipment. Observe also that, if i and j are two activities served by the same aircraft, i.e. $(i, j) \in \varepsilon^{\text{tow}}$, then $d_{iM_ijM_j}^{\text{setup}} = 0$. In analogy to the time needed for towing, the setup time is usually calculated by some function $d_{iM_ijM_j}^{\text{setup}} := f^{\text{setup}}(i, M_i, j, M_j)$.

Additionally to these basic disjunctive constraints we have to consider so-called ‘shadow’ restrictions (i, M_i, j, M_j) between gates M_i and M_j which can be interpreted as follows: if mode $M_i \in \mathcal{M}_i$ is assigned to activity i , then activity j must not be processed simultaneously in mode $M_j \in \mathcal{M}_j$. The set of shadow restrictions is denoted as $\varepsilon^{\text{shadow}}$.

Now, the model can be summarized as follows: Find a schedule (S, C, M) which assures con-

straints (1) to (5) and optimizes goals (6a) to (6c).

Minimal processing time

$$S_i + p_i^{min} \leq C_i \quad \forall i \in V \quad (1)$$

Continuous processing

$$C_i + d_{iM_i j M_j}^{tow} = S_j \quad \forall (i, j) \in \varepsilon^{tow} \quad (2)$$

Disjunctive activities and setup times

For all activities $i, j \in V$ such that either $M_i = M_j \neq \emptyset$ or $\exists (i, M_i, j, M_j) \in \varepsilon^{shadow}$ one of the following two conditions must be fulfilled:

$$C_i + d_{iM_i j M_j}^{setup} \leq S_j \quad (3a)$$

$$C_j + d_{jM_j i M_i}^{setup} \leq S_i \quad (3b)$$

Start and completion time

$$S_i = t_i^a \quad \forall i \in V^a \quad (4a)$$

$$C_i = t_i^d \quad \forall i \in V^d \quad (4b)$$

$$S_i, C_i \in \mathbb{N}_0 \quad \forall i \in V \quad (4c)$$

Mode selection

$$M_i \in \mathcal{M}_i \quad \forall i \in V \quad (5)$$

Objectives

The objective function is a vector containing the goals: (1) maximization of the total assignment preference score, (2) minimization of the number of required towing operations, and (3) minimization of the deviation from a given reference gate schedule. Hence, the objective function $z(M) := (z_1, z_2, z_3)$ is a vector with the three partial objectives where

$$z_1 := \sum_{i \in V} w_i u_{iM_i}, \quad (6a)$$

$$z_2 := |(i, j) \in \varepsilon^{tow} : M_i \neq M_j|, \text{ and} \quad (6b)$$

$$z_3 := \sum_{i \in V: M_i \neq M'_i} w_i. \quad (6c)$$

Typically $u_{i\mu} \in [0, 1]$ is a preference value associated with every activity-mode combination, $w_i \in [0, 1]$ is a priority weight associated with every activity and M'_i denotes the reference gate of activity i , respectively. Apparently, the choice of appropriate preference weights and priorities may have a substantial impact on the optimal gate schedule. Note that any maximization (minimization) objective can be converted into a minimization (maximization) objective by simply changing its sign.

3 Algorithms: survey and synthesis

Generally speaking project scheduling is concerned with single-item or small batch production where scarce resources have to be allocated to dependent activities over time. Accordingly,

applications can be found in diverse industries such as construction engineering, software development, etc. Due to the variety of potential applications there do exist two main algorithmic streams of research. One of them does not take into account any specifics of potential applications, that is, it is concerned with the development of general purpose algorithms (for a general project scheduling model type under consideration). On the contrary researchers develop algorithms for specific 'project scheduling' application domains (as we do here). Of course, both streams are interrelated and mutually stimulate each other. We review algorithms belonging to the first and to the second variety in subsections 3.1 and 3.2, respectively. While doing so the work done on simulated annealing is reviewed separately.

3.1 General purpose algorithms

It is well-known that even the single objective RCPSP is NP-hard [3], hence it cannot be solved in polynomial time unless $P = NP$ (see [23]). In the following we sketch exact and heuristic algorithms for the RCPSP and for the multi-mode RCPSP (MRCPSP) with a special emphasis on heuristics for the latter. Heuristics for solving the MRCPSP are of special importance because of two reasons: First, our model is close to the classical MRCPSP. Second, exact algorithms for solving the MRCPSP are not able to solve highly resource-constrained MRCPSP instances with more than 20 activities and 2 modes in acceptable running times. An in-depth survey of exact and heuristic algorithms for the RCPSP and the MRCPSP is given in [6]. For project scheduling techniques with special emphasis on time windows we refer the reader to [39].

Exact algorithms There exists a huge variety of exact solution techniques. Among them procedures based on the branch and bound principle are most popular. In the single objective case almost all authors consider makespan minimization as objective. In [41] for instance an algorithm based on branch and bound and guided by the so-called precedence tree is proposed. The algorithm developed in [12] uses the concept of minimal delaying alternatives originally proposed in [9]. The branch and bound algorithm presented in [7] uses the notion of schedule scheme and some ideas from [2], contrary to the concept of constructing partial schedules introduced in [47]. In [28] and [45] algorithmic enhancements based on dominance criteria are given. Finally, truncated branch and bound algorithms have been presented in [19, 20].

Heuristic algorithms A couple of heuristics for the RCPSP have been evaluated in [29]. Heuristic algorithms for solving the MRCPSP have for instance been provided in [34]. Recently, Hartmann [27] developed the most effective and efficient heuristic algorithm for solving the MRCPSP. Basically the algorithm works as follows. The genetic algorithm generates an initial population, i.e. the first generation, containing a certain number of individuals and then determines their fitness values. Then the population is randomly partitioned into pairs of individuals. To each pair of (parent) individuals, the crossover operator produces two new offsprings. Subsequently, the mutation operator is applied to the genotypes of the newly produced children. After computing the fitness of the offsprings, they are added to the current population, duplicating the population size temporarily. Then the selection operator is applied in order to reduce the population to its former size and to obtain the next generation to which the crossover operator is applied again. This process is repeated for a prespecified number of generations.

Simulated annealing A classical single objective SA for arbitrary optimization problem was originally proposed in [32] and [33]. In fact, SA originates from Monte Carlo methods developed in [37] to simulate physical annealing processes. To the best of our knowledge, one of the first attempts to adapt SA for solving RCPSP was done in [4] and it has been shown there that

the proposed procedure was able to outperform TS. In [36] an algorithm based on simulated annealing and priority scheduling methods has been proposed for the RCPSP. The priority scheduling method selects and schedules an activity with the highest priority value among those activities for which the predecessors have been completed and for which there is enough resource available at the current point of time. Performance of the algorithm is compared with existing heuristics from [41]. In [8] the SA from [36] is slightly modified in order to delay some activities, a fact which extends the search space and requires extra computation time. A new SA algorithm for RCPSP and MRCPSP was proposed in [5]. The conventional SA scheme is replaced by a new design, where the search is based on (1) alternated activities and a time incrementing process – in the case of RCPSP, (2) two embedded search loops alternating activities and mode neighborhood exploration – in the case of MRCPSP. Multicriteria versions of SA have been developed in [25, 26]. In [25] a two-stage approach is proposed: in the first stage a large representative sample of approximately non-dominated schedules is produced, whereas the second stage presents an interactive procedure searching for the best solutions in the sample. In [26] the approach developed in [25] is generalized for the MRCPSP with fuzzy data. A decision support system for a multicriteria version of the MRCPSP is described in [44].

A new variant of SA for MRCPSP was presented in [31], where two slightly different versions of SA were considered – SA with or without penalty functions. Contrary to the simple standard cooling scheme used in [4] and [5], both versions use a cooling scheme which has initially been described in [1]. It is based on the concept of Markov chains and their associated lengths. The most recent versions of local search-based probabilistic metaheuristics (SA and TS) for the MRCPSP with positive discounted cash flows and four different payment models have been presented in [38].

The multicriteria SA proposed in [10, 11] is a metaheuristic that uses the concepts sketched out above for simulated annealing and genetic algorithms. The output is a representative approximation of the Pareto set, the reason why it is known as Pareto simulated annealing (PSA). The fundamental milestones are as follows: (1) the concept of neighborhood, (2) use of aggregation function-based probabilities for acceptance of new neighborhood solutions, (3) the concept of generating solutions or agents, (4) management of the population of generating solutions or repulsion, (5) normalization of objectives, (6) the scheme of changing of the annealing temperature, and (7) updating the set of potentially Pareto optimal solutions. A general description of PSA is given in subsection 4.2. Observe that PSA defines only a general algorithmic scheme, which has to be customized for a particular combinatorial problem by defining the way new solutions are generated from a neighborhood for a given solution. For the FGSP the PSA is adopted in subsections 4.7 to 4.9.

3.2 Special purpose algorithms

Recall the major idea behind the FGSP presented above, that is, to model the problem as a special multi-mode resource-constrained project scheduling problem with generalized precedence constraints or time windows. The model can be interpreted in project scheduling terms as follows: (i) continuous processing restrictions (2) represent temporal constraints, (ii) disjunctive activities and setup time restrictions (3a) to (3b) generate unary resource constraints, (iii) start and completion time restrictions (4a) to (4c) define domain constraints. It should be noted that the objectives (6a) to (6c) depend only on the gate assignment and not on the start and completion times of processing activities at the assigned positions as in the classical RCPSP.

Algorithms specifically dedicated to the model (1) to (6) are very rare. To be honest, only two references, that is [16, 18], do exist. Some details are reviewed in what follows.

Exact algorithms The basic optimization algorithm for the FGSP is a truncated branch and bound procedure. The algorithm proceeds by assigning modes to the activities and by resolving resource conflicts that might appear. In comparison with a standard branch and bound procedure, it has several distinctive features. First of all, it uses two different types of branching: (i) branching over flight gates (modes) by assigning the best mode to some unscheduled activity according to some rule and accepting or forbidding this assignment afterwards, (ii) branching over disjunctive constraints by resolving resource conflicts and defining which activity from the set of already scheduled ones (an activity is considered to be scheduled if it has a mode assignment) must be the predecessor. The second feature of the proposed method is that it uses constraint propagation techniques. This means that at each node of the binary search tree induced by the branching scheme constraint propagation techniques are applied in order to reduce the search space until a fixed point has been computed.

Heuristic algorithms For dealing with large instances arising in practice (which have a huge number of aircraft activities and airport gates), the branch and bound procedure was upgraded with additional problem decomposition (variable partitioning) techniques. Additionally, large neighborhood search techniques have been implemented. Computational experiments with large real-life data as well as with manually constructed small examples demonstrate the effectiveness of the technique especially in comparison with the results of a modern rule based decision support system.

Nevertheless, it has to be emphasized that, in fact, the multicriteria core of the problem was not fully considered in [16, 18]. The authors propose to use a standard linearization technique in order to convert the multiple criteria problem into a single criterion one without giving spacial emphasis on selecting aggregation coefficients. Hence, the problem of finding the solutions which are efficient with respect to all objectives still seems to be open. Apparently, a simple aggregation of objectives has several disadvantages: (i) the interpretation of numeric values of an aggregated objective function is difficult due to the different meaning of the partial objectives, (ii) the meaning of the weights is not clear for the decision maker, and (iii) tuning the aggregation function by playing with a set of large weights may be too costly.

Summarizing we conclude that the search for methods which are better suited for FGSP under multiple objectives is an important and interesting topic of research.

4 Pareto simulated annealing

In this section we describe a Pareto simulated annealing algorithm (PSA) for solving the FGSP (1) to (6).³ The choice of simulated annealing is not occasional: As shown in [30], SA has computational advantages in a multiple criteria setting.

4.1 Optimality in the presence of multiple criteria

We start with necessary definitions of optimality in the multiple criteria case. Contrary to the single criterion case, where the structure associated with the set of feasible solutions is totally ordered (i.e. there is no incomparability between two solutions, and the definition of optimality is straightforward) in the multicriteria case this definition is no longer trivial, because partial

³In [21] PSA has been adapted to the gate assignment problem modelled as a multiple criteria QAP.

criteria, which have to be optimized simultaneously, are often in conflict with each other (that is, incomparable).

Let $\Pi \subset \mathbb{R}^\gamma$ be the set of feasible solutions. Moreover, let $\mathbb{R}^\delta \supset Z = \{z_p\}_{p=1}^\delta$ be the mapping of δ criteria in the space of Π . The concept of optimality is based on the dominance relation between two vectors in the criteria space. In case of minimization

$$z(\pi) = (z_1(\pi), z_2(\pi), \dots, z_\delta(\pi)) \rightarrow \min_{\pi \in \Pi}$$

we say that π' dominates π if $z_p(\pi') \leq z_p(\pi)$ for all $p = 1, \dots, \delta$, with at least one strict inequality, and π is non-dominated otherwise. A set of solutions is said to be Pareto optimal if, while moving from solution π belonging to the set of Pareto optimal solutions (or briefly, the Pareto set) $\mathcal{P}^\delta \subseteq \Pi$ to another solution $\pi' \in \Pi$ in the feasible set, an improvement of the value of one criterion would at least deteriorate the value of one other criterion.

In terms of vector dominance the definition of Pareto optimality can be reformulated as follows: $\pi \in \Pi$ is a Pareto optimal solution (Pareto optimum) if and only if there exists no solution π' that dominates π . Any π which is not Pareto optimal cannot represent an optimal solution, because $\exists \pi' \in \Pi$ such that $z_p(\pi') \leq z_p(\pi)$ for all $p = 1, \dots, \delta$ with a strict inequality in at least one case. Thus, roughly speaking, the Pareto set is the set of all non-dominated solutions.

The method commonly used for solving a multicriteria problem is called weighting method [46]. It takes each objective function z_p and multiplies it by the corresponding weighting coefficient λ_p . The modified functions are then added together to obtain a single cost function, which can easily be solved using any single objective method. Mathematically, the new function is written as

$$z_\lambda(\pi) = \sum_{p=1}^{\delta} \lambda_p z_p(\pi), \quad 0 \leq \lambda_p \leq 1, \quad \sum_{p=1}^{\delta} \lambda_p = 1, \quad \lambda = (\lambda_1, \dots, \lambda_\delta).$$

If the problem is convex, i.e. the set of feasible solutions and the objective functions are convex, then a complete set of proper Pareto optimal solutions can be found ('proper' in the sense of [24]; see, e.g., [22] and [48], too). However, if the problem is not convex, then there is no guarantee that this method will yield the entire Pareto set. Usually, the weighting coefficients are initialized somehow, and they are varied afterwards to generate the entire Pareto set.

For solving the multicriteria FGSP one might now suggest to proceed as follows: (1) aggregate all objectives into one goal function, and (2) use, e.g., some standard heuristics for resolving the new problem with the aggregated objective function. As outlined above such an approach has been chosen by [16, 18]. In the sequel, however, we will show that such an approach is inferior to a multicriteria heuristic.

4.2 Pareto simulated annealing: general description

Pareto simulated annealing (PSA) proposed in [10], [11] is a multiple criteria metaheuristic that uses some general building blocks developed for single objective simulated annealing (SA) as well as several specific concepts which take into account the multiple criteria nature of the problem (see [30]).

Standard concepts of SA

- *Neighborhood concept*

At each iteration q the neighborhood $N(\pi)$ of solution π is specified (usually problem-specific). $N(\pi)$ represents the subset of solutions π' which can be reached in one iteration emanating from π . Generally, it is not possible to store the neighborhood structure, because the set of feasible solutions has an exponential size. To overcome this difficulty a set AM of allowed modifications F or neighborhood search moves is introduced. In this case for a given solution π the neighborhood $N(\pi)$ can be defined by $N(\pi) = \{F(\pi) \mid F \in AM\}$.

- *Probabilistic acceptance of a new neighborhood solution*

In each iteration q solution π' is accepted with probability

$$P(\pi, \pi', T_q) = \min \left\{ 1, \exp \left(-\frac{z(\pi') - z(\pi)}{T_q} \right) \right\}.$$

This probability is decreased during the course of the algorithm. In other words, we can escape a local optimum, but the probability for doing so is low after a large number of iterations. It guarantees that there is a high probability to locate a global optimum while avoiding to be trapped in local optima.

- *Parameter (temperature) dependent acceptance probability*

The probability of making the transition to the new solution depends on a global time-varying parameter T called the annealing temperature: $(T_q)_{q=1}^{\infty}$ is a sequence of positive control parameters with $\lim_{q \rightarrow \infty} T_q = 0$.

- *Cooling schedule*

Generally, the sequence $(T_q)_{q=1}^{\infty}$ is created by a function g , i.e. $T_{q+1} = g(T_q)$ for all q . The initial annealing temperature T_0 has to be defined in advance.

- *Termination criterion*

One has the freedom to introduce different stopping criteria. Typically, SA is repeated until the system reaches a state which is "good enough", or until a given time limit has been reached. The annealing temperature decreases to (nearly) zero short before termination.

Specific concepts of PSA

- *Use of aggregation function-based probabilities for acceptance of new neighborhood solutions*

PSA uses the concept of multiple objective acceptance rules proposed by Serafini [42], [43]. In the multiple criteria case while moving from solution π to some randomly generated solution $\pi' \in N(\pi)$ one of the following three mutually exclusive situations may arise: (i) π' dominates or is equivalent to π , (ii) π' is dominated by π , (iii) π' is non-dominated with respect to π . In the first case the new solution may be considered to be not worse than the current one and, hence, accepted with probability one. In the second case the new solution may be considered to be worse than the current one (π is considered as potentially Pareto optimal) and accepted with probability less than one in order to

avoid getting trapped in local optima. In the third case π and π' are incomparable (and initially non-dominated). It follows that we have to introduce a multicriteria probabilistic acceptance rule which can be defined in various ways, interpreted as a local aggregation of all objectives with the weighted Tchebycheff scalarizing function or weighted linear scalarizing function with reference point $z(\pi)$. Thus, the probability of accepting the new solution π' is defined as:

$$P(\pi, \pi', \lambda, T_q) = \min \left\{ 1, \exp \left(- \max_{p=1, \dots, \delta} \frac{\lambda_p [z_p(\pi') - z_p(\pi)]}{T_q} \right) \right\}$$

- *The concept of generating solutions or agents*

Initially a set (sample) G of generating solutions with fixed cardinality $|G| = \phi$ is randomly produced to provide a sufficiently large search space for PSA. With each of the generating solutions a separate weight vector $\lambda^\pi = (\lambda_1^\pi, \dots, \lambda_\delta^\pi)$ is associated. For each $\pi \in G$ the weight vector can be generated easily by the following iterative procedure:

1. Let $I = \{1, \dots, \delta\}$ and $\Omega = 1$;
2. Randomly choose $p \in I$;
3. If $|I| \neq 1$, then $\lambda_p^\pi := \Omega \cdot \text{Random}(0, 1)$, $\Omega := \Omega - \lambda_p^\pi$,
else $\lambda_p^\pi := \Omega$ and stop;
4. Set $I := I \setminus \{p\}$ and go to step 2.

Note that Ω corresponds to the total sum of weights $\sum_{p=1}^{\delta} \lambda_p^\pi$. Manipulating these weights provides good dispersion of the outcome Pareto front and makes the solutions of PSA more representative. The generating solutions may be treated as “spy agents” working almost independently but exchanging information about their positions. It allows to control the distance between agents, predicting the generating solutions to be very close or too far from each other.

- *Management of the population of generating solutions or repulsion*

The degree of repulsion α is fixed and usually takes a very small positive value close to zero. The weight vector λ^π associated with a given generating solution π is modified in order to increase the probability of moving π away from its closest neighbor $\tilde{\pi}$ in the generating sample. This is obtained by increasing the weights of the objectives for which π is better than $\tilde{\pi}$ and decreasing the weights of the objectives for which π is worse than $\tilde{\pi}$:

$$\lambda_p^\pi := \begin{cases} \lambda_p^\pi + \alpha & \text{if } z_p(\pi) \leq z_p(\tilde{\pi}), \\ \lambda_p^\pi - \alpha & \text{otherwise,} \end{cases} \quad p = 1, \dots, \delta$$

The following metric distance between solutions in the space of normalized objectives is used to determine the closest neighbor $\tilde{\pi}$ of π :

$$\sum_{p=1}^{\delta} (z_p(\pi) - z_p(\tilde{\pi}))^2 \rightarrow \min_{\tilde{\pi} \in G \setminus \{\pi\}}$$

An additional requirement is that the closest neighbor $\tilde{\pi}$ has to be non-dominated with respect to π . If there is no generating solution that meets this requirement each weight is either increased or decreased with probability equal to $\frac{1}{2}$:

$$P(\lambda_p^\pi := \lambda_p^\pi + \alpha) = P(\lambda_p^\pi := \lambda_p^\pi - \alpha) = \frac{1}{2}$$

Note that this mechanism never repulses the generating solutions outside the non-dominated set. Moreover, each repulsion iteration also requires normalization to satisfy $\sum_{p=1}^{\delta} \lambda_p^{\pi} = 1$.

- *Normalization of objectives*

In real-life problems the ranges of objectives usually are quite different. The idea of PSA is to work with normalized weight vectors. So, the scalarization should be applied to normalized objective values. For each objective z_p , $p = 1, \dots, \delta$, we introduce the so-called range equalization factor [46]

$$\mu_p = \frac{1}{r_p}$$

where r_p is the (approximate) range of objective p . Objective function values multiplied by range equalization factors are called normalized objective function values. This method is working properly if the objective range is large enough. Sometimes when the exact calculation of the objective range is too difficult, one can set r_p equal to the expected approximate maximum value of the objective.

- *Updating the set of potentially Pareto optimal solutions*

The set of potentially Pareto optimal solutions $\mathcal{P}_{pot}^{\delta}$ is empty at the beginning of the method. It is updated after a new solution is generated. Updating the set of potentially Pareto optimal solutions with solution π consists of: (1) adding π to $\mathcal{P}_{pot}^{\delta}$ if no solution in $\mathcal{P}_{pot}^{\delta}$ dominates π and (2) removing from $\mathcal{P}_{pot}^{\delta}$ all solutions dominated by π . The process of updating $\mathcal{P}_{pot}^{\delta}$ may be very time consuming. Thus, several simple additional techniques could be used in order to reduce the time required for updating; for details see [30].

The characteristics of PSA may vary dependent on particular characteristics of the problem under consideration and, of course, substantially influence the effectiveness of PSA. Initial values of the following parameters have to be defined in advance: the number of generating solutions ϕ , the initial annealing temperature T_0 , the repulsion coefficient $a > 0$, the cooling schedule and the termination criterion. A flowchart of PSA summarizing all ingredients is presented in Figure 1.

4.3 Arrival, parking and departure activities

Let us now specifically address model (1) to (6). Figure 2 illustrates the most general case when arrival activity $i \in V^a$, parking activity $j \in V^p$ and departure activity $k \in V^d$ of a particular aircraft are assigned to different gates M_i , M_j , M_k , respectively, and connected with towing procedures. The black part of an activity identifies the predefined minimal processing time, whereas the length of the grey part of each activity depends on variables' values and, hence, may vary.

In the sequel for each activity $i \in V$ we will use the following notation: ES_i/LS_i – earliest/latest start time of activity i , EC_i/LC_i – earliest/latest completion time of activity i . These parameters depend on the activity type. Initially they can be calculated as follows:

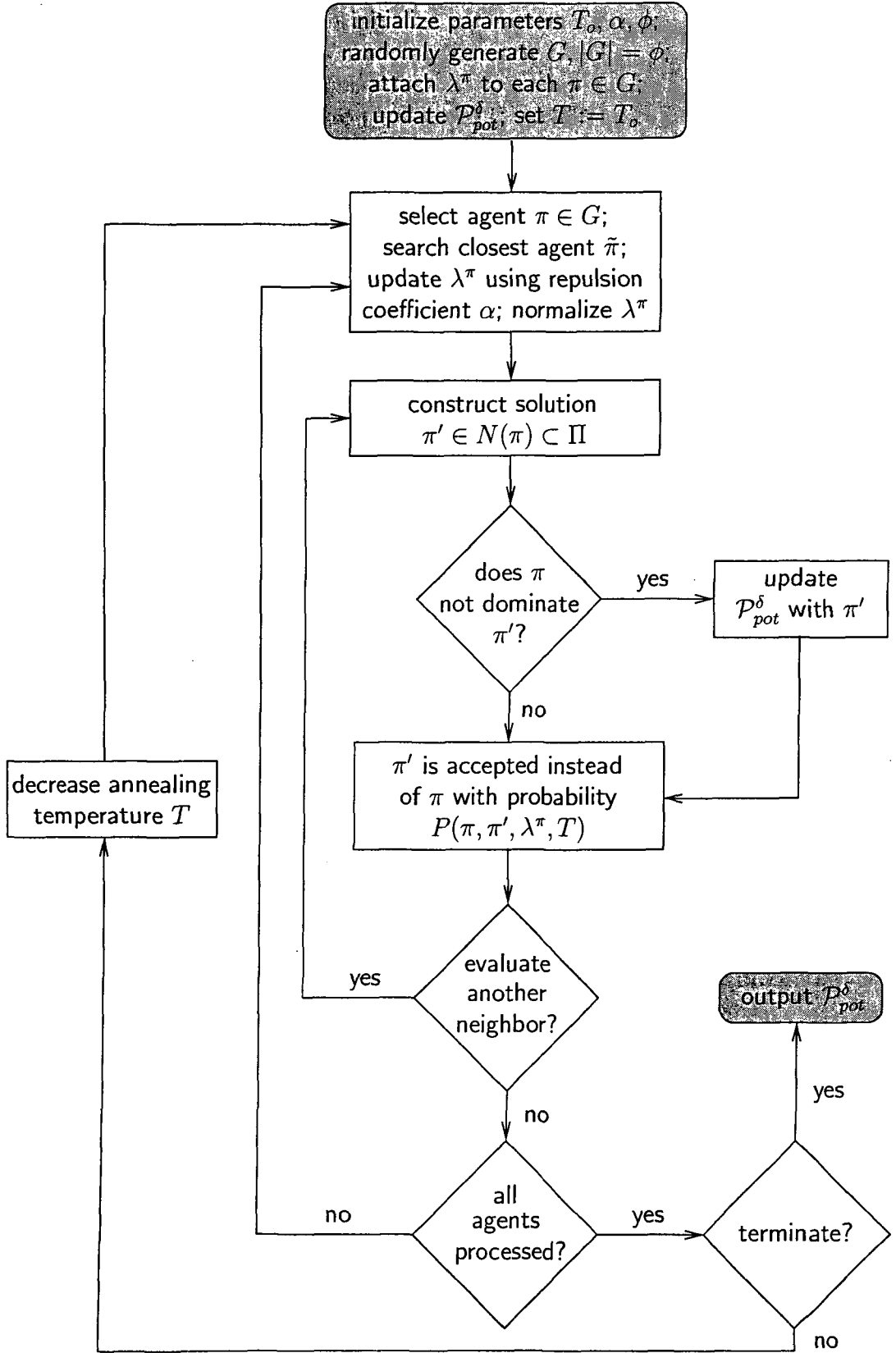


Figure 1: Flowchart of PSA

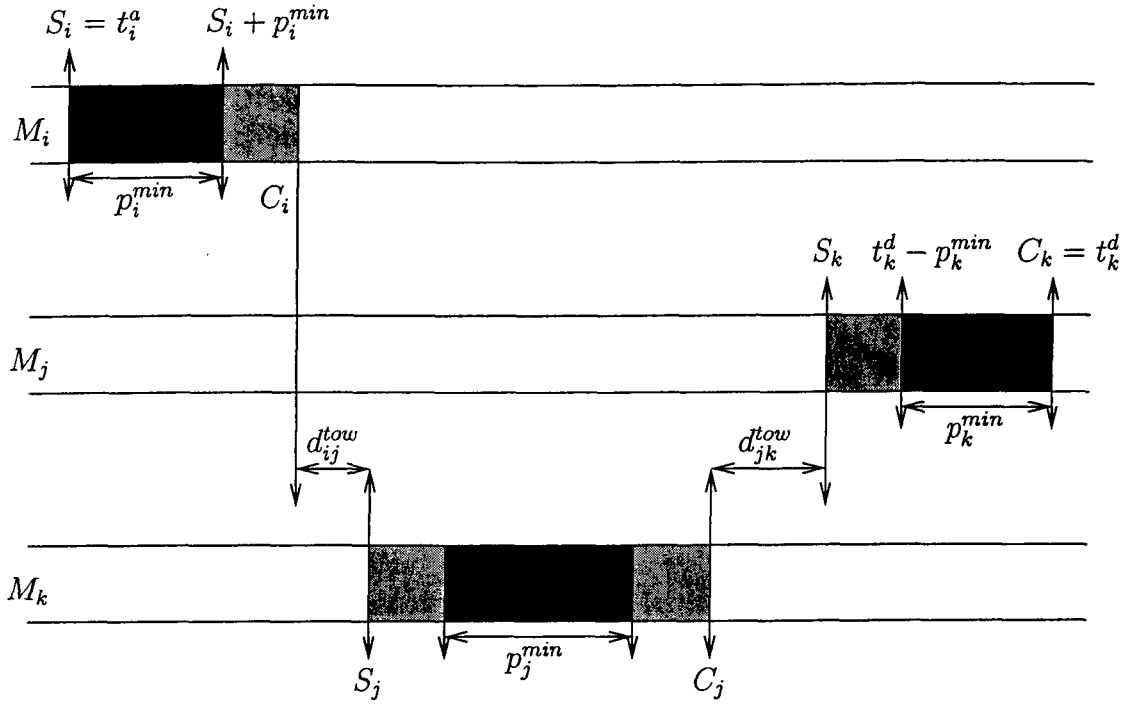


Figure 2: Arrival, Parking and Departure Activity

Arrival activity $i \in V^a$	$ES_i = LS_i = S_i = t^a$ $EC_i = t^a + p_i^{min}$ $LC_i = t^d - p_k^{min} - d_{jM_jkM_k}^{tow} - p_j^{min} - d_{iM_ijM_j}^{tow}$
Parking activity $j \in V^p$	$ES_j = t^a + p_i^{min} + d_{iM_ijM_j}^{tow}$ $LS_j = t^d - p_k^{min} - d_{jM_jkM_k}^{tow} - p_j^{min}$ $EC_j = t^a + p_i^{min} + d_{iM_ijM_j}^{tow} + p_j^{min}$ $LC_j = t^d - p_k^{min} - d_{jM_jkM_k}^{tow}$
Departure activity $k \in V^d$	$EC_k = LC_k = C_k = t^d$ $ES_k = t^a + p_i^{min} + d_{iM_ijM_j}^{tow} + p_j^{min} + d_{jM_jkM_k}^{tow}$ $LS_k = t^d - p_k^{min}$

If $M_i = M_j$, that is, arrival and parking activities are assigned to the same gate, then $d_{iM_ijM_j}^{tow} = 0$, that is, there is no need for towing. Likewise, if $M_j = M_k$, then $d_{jM_jkM_k}^{tow} = 0$. Otherwise, $d_{iM_ijM_j}^{tow} > 0$ and $d_{jM_jkM_k}^{tow} > 0$. Recall that in order to reduce the number of parameters which have to be defined explicitly, towing times are usually calculated by means of a function $d_{iM_ijM_j}^{tow} := f^{tow}(i, M_i, j, M_j)$.

Initial values of earliest and latest start and completion times define the admissible domain for each time variable as follows:

Activity	Variable	Interval
$i \in V^a$	S_i, C_i	$\{t_i^a\}, [EC_i, LC_i]$
$j \in V^p$	S_j, C_j	$[ES_j, LS_j], [EC_j, LC_j]$
$k \in V^d$	S_k, C_k	$[ES_k, LS_k], \{t_k^d\}$

Without loss of generality we assume that all variables take their values from corresponding time domains, the values of which are cardinal numbers.

Initially, for each activity $j \in V^p$ the following relations are obvious:

$$EC_j = ES_j + p_j^{min}, \quad LC_j = LS_j + p_j^{min}, \quad \text{that is, } LC_j - EC_j = LS_j - ES_j.$$

Observe that start and completion time domains depend on chosen modes (gates). In order to avoid this, we initialize the admissible interval by taking the minimal (usually zero because $f^{tow}(i, \mu, j, \mu) = 0$) value of $f^{tow}(i, M_i, j, M_j)$ instead of $d_{iM_ijM_j}^{tow}$ for each $(i, j) \in V \times V$.

4.4 Mode domains and their reduction

Recall that the objectives depend on the mode assignment only. Therefore, we will say that activity $i \in V$ is scheduled or assigned if some mode from domain Δ_{M_i} that initially coincides with \mathcal{M}_i is assigned to i , in other words the set of scheduled activities represents the set of activities V^S which are assigned to some particular gates. All other activities are considered as non-scheduled and belong to the set of free activities $V^f := V \setminus V^S$.

Let i and i' be two activities served by different aircraft, and assume the temporal constraint i before i' (denoted as $i \rightarrow i'$) to be valid. If no such constraint exists for a particular gate assignment $\mu \in \Delta_{M_{i'}}$, then μ can be removed from $\Delta_{M_{i'}}$. Formally, we have

$$i \rightarrow i', \mu \in \Delta_{M_{i'}} \wedge ES_i + \min_{\nu \in \Delta_{M_{i'}}} d_{i\mu i'\nu}^{setup} > LS_{i'} \Rightarrow \Delta_{M_{i'}} = \Delta_{M_{i'}} \setminus \{\mu\}. \quad (7)$$

In other words, before each new activity assignment we have to scan all activities with their mode domains and update the domains by excluding those modes which are not feasible at the moment. Such a procedure is known as 'mode shaving' (see [16]). We include the apron into mode domains in order to assure that the domain set always contains at least one element, given the fact that the apron has unlimited capacity and assignment is never forbidden.

4.5 Disjunctive activities

Gates are disjunctive resources. They can process one activity at a time. The only exception is dummy gate 0 that represents the apron. It has infinite capacity and can process an unlimited number of flights simultaneously.

The set of disjunctive activities, i.e. activities that may potentially overlap and for which constraints (3a) to (3b) must be explicitly defined, can be generally described as follows:

$$\begin{aligned} \mathcal{D} := \{ & (i, i') \in V \times V \mid LC_i + \max_{\mu \in \mathcal{M}_i, \nu \in \mathcal{M}_{i'}} d_{i\mu i'\nu}^{setup} > ES_{i'} \wedge \\ & LC_{i'} + \max_{\mu \in \mathcal{M}_i, \nu \in \mathcal{M}_{i'}} d_{i'\nu i\mu}^{setup} > ES_i \wedge \\ & [\mathcal{M}_i \cap \mathcal{M}_{i'} \neq \emptyset \vee \exists (i, \mu, i', \nu) \in \varepsilon^{shadow} : \mu \in \mathcal{M}_i, \nu \in \mathcal{M}_{i'}] \}. \end{aligned}$$

In fact, any disjunctive conflict depends on the gate assignment, so we delay its resolution until the corresponding activities are scheduled. In other words, if a disjunctive conflict appears, that is, if (i) the same mode is assigned to activities i and i' or (ii) shadow restrictions between i assigned to gate μ and i' assigned to gate ν arise, and, additionally, i may precede i' as well as i' may precede i , then we have to resolve this conflict by ordering the activities, i.e. defining whether activity i or i' should be processed first.

For illustrative purposes consider the following situation: Let i be an activity already assigned to gate μ and let i' be potentially assigned feasible gate ν . Suppose that a disjunctive conflict induced a shadow restriction $(i, \mu, i', \nu) \in \varepsilon^{shadow}$ between the two activities, i.e.

$$LC_i + d_{i\mu i'\nu}^{setup} > ES_{i'} \text{ and } LC_{i'} + d_{i'\nu i\mu}^{setup} > ES_i.$$

Observe that these two inequalities are valid only if either i or i' represents a parking activity (or both, of course); otherwise there is no disjunctive conflict at all. Then we choose one the two alternative precedence relations at random. This heuristic rule we will use for generating a new solution. Assume now that w.l.o.g. i must precede i' which implies that $C_i + d_{i\mu i'\nu}^{setup} \leq S_{i'}$ has to be assured. In the next section we describe exact time domain reduction rules that can be interpreted as implicit additional time constraints.

4.6 Time domains and their reduction

We will denote start and completion time domains of activity $i \in V$ as Δ_{S_i} and Δ_{C_i} . The time domains will be updated iteratively during the course of the algorithm, and ES_i (EC_i) and LS_i (LC_i) will denote the current left and right border of the start and completion time interval, respectively. Initially the time domains are calculated as follows:

$$\left| \begin{array}{l} \text{Activity} \\ i \in V^a \\ j \in V^p \\ k \in V^d \end{array} \right| \left| \begin{array}{l} \text{Initial time domain} \\ \Delta_{C_i}^{init} = [EC_i^{init}, LC_i^{init}] \\ \Delta_{S_j}^{init} = [ES_j^{init}, LS_j^{init}], \Delta_{C_j}^{init} = [EC_j^{init}, LC_j^{init}] \\ \Delta_{S_k}^{init} = [ES_k^{init}, LS_k^{init}], \end{array} \right|$$

where

$$\begin{aligned} EC_i^{init} &= t^a + p_i^{min}, & LC_i^{init} &= t^d - p_k^{min} - p_j^{min}, \\ ES_j^{init} &= t^a + p_i^{min}, & LS_j^{init} &= t^d - p_k^{min} - p_j^{min}, \\ EC_j^{init} &= t^a + p_i^{min} + p_j^{min}, & LC_j^{init} &= t^d - p_k^{min}, \\ ES_k^{init} &= t^a + p_i^{min} + p_j^{min}, & LS_k^{init} &= t^d - p_k^{min} - t^d - p_k^{min}. \end{aligned}$$

Note that $\Delta_{S_i} = \Delta_{S_i}^{init} = \{t_i^a\} \forall i \in V^a$ and $\Delta_{C_k} = \Delta_{C_k}^{init} = \{t_k^d\} \forall k \in V^d$.

Time domain reduction for linked activities $(i, j, k) \in V^a \times V^p \times V^d$ is possible in case of towing procedures as follows:

$$\begin{aligned} \text{If } (i, j) \in V^s \times V^s, \text{ then } & LC_i^{new} = \min\{LC_i^{old}, LC_i^{init} - d_{iM_i j M_j}^{tow}\}, \\ & ES_j^{new} = \max\{ES_j^{old}, ES_j^{init} + d_{iM_i j M_j}^{tow}\}, \\ & EC_j^{new} = \max\{EC_j^{old}, EC_j^{init} + d_{iM_i j M_j}^{tow}\}, \\ & ES_k^{new} = \max\{ES_k^{old}, ES_k^{init} + d_{iM_i j M_j}^{tow}\}. \end{aligned}$$

$$\begin{aligned} \text{If } (j, k) \in V^s \times V^s, \text{ then } & LC_i^{new} = \min\{LC_i^{old}, LC_i^{init} - d_{jM_j k M_k}^{tow}\}, \\ & LS_j^{new} = \min\{LS_j^{old}, LS_j^{init} - d_{jM_j k M_k}^{tow}\}, \\ & LC_j^{new} = \min\{LC_j^{old}, LC_j^{init} - d_{jM_j k M_k}^{tow}\}, \\ & ES_k^{new} = \max\{ES_k^{old}, ES_k^{init} + d_{jM_j k M_k}^{tow}\}. \end{aligned}$$

$$\begin{aligned} \text{If } (i, j, k) \in V^s \times V^s \times V^s, \text{ then } & LC_i^{new} = \min\{LC_i^{old}, LC_i^{init} - d_{iM_i j M_j}^{tow} - d_{jM_j k M_k}^{tow}\}, \\ & ES_j^{new} = \max\{ES_j^{old}, ES_j^{init} + d_{iM_i j M_j}^{tow}\}, \\ & EC_j^{new} = \max\{EC_j^{old}, EC_j^{init} + d_{iM_i j M_j}^{tow}\}, \\ & LS_j^{new} = \min\{LS_j^{old}, LS_j^{init} - d_{jM_j k M_k}^{tow}\}, \\ & LC_j^{new} = \min\{LC_j^{old}, LC_j^{init} - d_{jM_j k M_k}^{tow}\}, \\ & ES_k^{new} = \max\{ES_k^{old}, ES_k^{init} + d_{iM_i j M_j}^{tow} + d_{jM_j k M_k}^{tow}\}. \end{aligned}$$

Here 'old' denotes the corresponding previous value while 'new' indicates the newly calculated one.

Let i and i' be two activities served by different aircraft which are assigned to gates μ and ν , respectively. If they are not in disjunction, there is nothing to reduce. Otherwise we can perform reduction only after resolving the disjunctive conflict. W.l.o.g. suppose that we have $i \rightarrow i'$. Then the following reduction of the time domain sets is possible:

$$\Delta_{C_i}^{new} = \Delta_{C_i}^{old} \setminus (LS_{i'}^{old} - d_{i\mu i'\mu}^{setup}, \infty) \quad (8)$$

$$\Delta_{S_{i'}}^{new} = \Delta_{S_{i'}}^{old} \setminus [0, EC_i^{old} + d_{i\mu i'\mu}^{setup}) \quad (9)$$

In other words rules (8) and (9) imply

$$LC_i^{new} = \min\{LC_i^{old}, LS_{i'}^{old} - d_{i\mu i'\mu}^{setup}\}, \text{ and}$$

$$ES_{i'}^{new} = \max\{ES_{i'}^{old}, EC_i^{old} + d_{i\mu i'\mu}^{setup}\}.$$

The reduction of the domain of an arrival, parking or departure activity may imply a reduction of the domains of all linked (connected with towing) activities served by the same aircraft. Such domain reduction does not depend on the fact whether the linked activity has already scheduled or not. In other words, if we perform domain reduction according to rules (8) and (9), then we should perform mode reduction for activities j and k linked to activity i as well as for activities j' and k' linked to activity i' .

Suppose that we apply time domain reduction according to rule (8). Two cases are possible:

- *Case 1.* Activity $i \in V^a$ represents an arrival activity. Let $j \in V^p$ and $k \in V^d$ be two activities linked to i . Then time domain reduction for j and k works as follows:

$$LS_j^{new} = \min\{LS_j^{old}, LS_j^{old} - (LC_i^{old} - (LS_{i'}^{old} - d_{i\mu i'\mu}^{setup}))\},$$

$$LC_j^{new} = LC_j^{old}, \quad LS_k^{new} = LS_k^{old}.$$

- *Case 2.* Activity $i \in V^p$ represents a parking activity. Let $j \in V^a$ and $k \in V^d$ be two activities linked to i . Then time domain reduction for j and k works as follows:

$$LS_i^{new} = \min\{LS_i^{old}, LS_i^{old} - (LC_i^{old} - (LS_{i'}^{old} - d_{i\mu i'\mu}^{setup}))\},$$

$$LC_j^{new} = \min\{LC_j^{old}, LC_j^{old} - (LC_i^{old} - (LS_{i'}^{old} - d_{i\mu i'\mu}^{setup}))\},$$

$$LS_k^{new} = \min\{LS_k^{old}, LS_k^{old} - (LC_i^{old} - (LS_{i'}^{old} - d_{i\mu i'\mu}^{setup}))\}.$$

It should be emphasized that if activity i represents a departure activity, then the completion time domain consists of one element, $LC_i = EC_i = t_i^d$, and domain reduction does not make any sense at all.

Now suppose that we apply time domain reduction according to rule (9). For activities j' and k' linked to activity i' it works as follows. Again two cases are possible:

- *Case 1.* Activity $i' \in V^d$ represents a departure activity. Let $j' \in V^p$ and $k' \in V^a$ be two activities linked to i' . Then time domain reduction for j' and k' works as follows:

$$EC_{j'}^{new} = \max\{EC_{j'}^{old}, EC_{j'}^{old} + ((EC_i^{old} + d_{i\mu i'\mu}^{setup}) - ES_{i'}^{old})\},$$

$$ES_{j'}^{new} = ES_{j'}^{old}, \quad EC_{k'}^{new} = EC_{k'}^{old}.$$

- *Case 2.* Activity $i' \in V^p$ represents a parking activity. Let $j' \in V^a$ and $k' \in V^d$ be two activities linked to i . Then time domain reduction for j and k works as follows:

$$\begin{aligned} EC_{i'}^{new} &= \max\{EC_{i'}^{old}, EC_{i'}^{old} + ((EC_i^{old} + d_{i\mu i'\mu}^{setup}) - ES_{i'}^{old})\}, \\ EC_{j'}^{new} &= \max\{EC_{j'}^{old}, EC_{j'}^{old} + ((EC_i^{old} + d_{i\mu i'\mu}^{setup}) - ES_{i'}^{old})\}, \\ ES_{k'}^{new} &= \max\{ES_{k'}^{old}, ES_{k'}^{old} + ((EC_i^{old} + d_{i\mu i'\mu}^{setup}) - ES_{i'}^{old})\}. \end{aligned}$$

It should be emphasized that if activity i' represents an arrival activity, then the start time domain consists of one element, $LS_{i'} = ES_{i'} = t_{i'}^a$, and domain reduction does not make any sense at all.

For the case where $i' \rightarrow i$, one can construct domain reduction rules in a similar way.

Note, that time domain reduction assures minimal processing time constraints (1), continuous processing constraints (2) and disjunctive precedence constraints (3) once we have resolved the disjunctive conflict.

Observe, that this modification of time domains can only lead to reduction of unresolved constraints without resolving time constraints itself. The remaining freedom has no influence on objective values and can be easily resolved (because no conflicts appear after time domain reduction) in a supplementary optimization subproblem, which is beyond the scope of this paper.

4.7 Initial solution

In order to start PSA we need to construct an initial solution π^0 , which is both time and resource feasible, i.e. satisfies constraints (1) to (5). Formally, the procedure of finding an initial solution can be divided into two stages.

Preprocessing stage. Sometimes the time span between arrival and departure, which is named as ground time of a particular flight, is too short. In this case the arrival activity i , parking activity j and departure activity k cannot be assigned to different gates, that is, must be assigned to the same gate. This fact can be easily detected during preprocessing. By defining minimal processing times of the activities $p_i^{min} := t_k^d - t_i^a$, $p_j^{min} := 0$ and $p_k^{min} := 0$, we can immediately guarantee $M_i = M_j = M_k$, that is, all activities will be assigned to the same gate. Observe that assigning all three activities (i, j, k) of the same flight to the same gate M means that $d_{iMjM}^{tow} = 0$ and $d_{jMkM}^{tow} = 0$. We can label such activities as 'non-split' and all other activities as 'split', respectively.

Main stage. The initial feasible solution can be constructed by assigning different modes to the activities according to the rules below. The procedure terminates when all activities have been scheduled and an initial feasible solution has been constructed.

Initialization

Initialize $V^s := \emptyset$ and $V^f := V$. Apply mode shaving (see subsection 4.4).

Iteration (proceed until all activities are scheduled)

- Randomly choose activity $i \in V^f$. Select $M_i \in \Delta_{M_i}$ at random, assign M_i to activity i , i.e. set $V^s := V^s \cup \{i\}$ and $V^f := V^f \setminus \{i\}$. Then exclude M_i from Δ_{M_i} .
- If there appears a disjunctive conflict between activities i and $j \in V^s$, then resolve it by choosing a precedence relation for i and j at random.
- Perform start and completion time reduction as described in subsection 4.6. Apply mode shaving.

Because the objective functions depend on mode assignment only, we search for a solution in which all modes have been assigned. In this case the start and completion time domains are generally reduced, but may still have more than one element, that is, temporal constraints may still not be resolved. The remaining degree of freedom can be easily exploited by a supplementary optimization problem that allows to schedule all required towing procedures.

4.8 Neighborhood search moves

Recall that PSA defines only a general scheme. In order to be applicable to a particular optimization problem PSA has to be customized according to the combinatorial specifics of the problem. In order to do so we need neighborhood search moves, i.e. rules which define how to move from one feasible solution π to another solution π' (called neighbor). These moves should fulfill two conditions: (i) ease of calculation of cost change $\Theta_p = z_p(\pi') - z_p(\pi)$ for each partial objective $p = 1, \dots, 3$; (ii) coverage of the entire search space or at least a major part of it.

The general idea behind the construction of search moves is as follows: In each iteration of PSA we will choose at random one 'split' activity i (suppose that it was assigned to gate μ) and relax the decision concerning i but keep all other decisions concerning activities $j \in V(i) := V \setminus \{i\}$. Then we reconstruct the partial solution for all activities from $V(i)$ by resetting mode and time domains to their initial values and then making all mode assignment and disjunction orientation decisions we have made in the previous stage during the construction of schedule π . Observe, that the application of time reduction rules during reconstruction may essentially change time domains for activities $j \in V(i) := V \setminus \{i\}$, but feasibility of chosen modes is preserved. Afterwards, we have to complete the new schedule π' by randomly choosing one of two strategies:

a) *Basic strategy*. We forbid an old mode assignment by excluding μ from Δ_{M_i} and choose another feasible gate. If no feasible gate is available (it is possible when set of feasible modes Δ_{M_i} consists of fictions gate $m+1$ only that represents the apron), then neighborhood search move fails.

b) *Disjunctive strategy*. If a chosen activity i is a disjunctive one, then we can alter the previous disjunctive decision (i.e. invert the precedence relation) and assign activity i to gate μ afterwards.

We will call a move according to both strategies Basic move or Disjunctive move, respectively. Obviously, the calculation of the costs change $\Theta_1, \Theta_2, \Theta_3$ is cheap in both cases.

4.9 Summary of PSA

Stepwise the algorithm is summarized as follows:

1. Construct an initial feasible solution π^0 by the strategy described above. Set the annealing temperature T as a linear function of the input size $T = T_0 \cdot n$, where T_0 is the starting temperature and n is the number of flights. Initialize the list of potential Pareto optimal solutions: $\mathcal{P}_{pot}^3 = \{\pi^0\}$. Define the number of agents ϕ . Initialize the repulsion coefficient α .
2. If $\phi = 1$, then solution π^0 represents a single agent, otherwise, if the number of agents is greater than one, generate set G , $|G| = \phi$, of agents using the same strategy which we used for construction of initial solution. Attach a normalized weight vector $\lambda^\pi = (\lambda_1^\pi, \lambda_2^\pi, \lambda_3^\pi)$ to each of the generating solutions $\pi \in G$. Update \mathcal{P}_{pot}^3 by checking dominance relations between agents.

3. For each agent $\pi \in G$ do:

(a) Select agent $\tilde{\pi} \in G$ which is nearest to π , that is, such that

$$\sum_{p=1}^3 (z_p(\pi) - z_p(\tilde{\pi}))^2 \rightarrow \min_{\tilde{\pi} \in G \setminus \{\pi\}}$$

and which is non-dominated with respect to π . If there is no such solution $\tilde{\pi}$, then each objective is increased or decreased with probability equal to $\frac{1}{2}$:

$$P(\lambda_p^\pi := \lambda_p^\pi + \alpha) = P(\lambda_p^\pi := \lambda_p^\pi - \alpha) = \frac{1}{2}, \quad p = 1, 2, 3;$$

Else for each objective $p = 1, 2, 3$:

$$\lambda_p^\pi := \begin{cases} \lambda_p^\pi + \alpha & \text{if } z_p(\pi) \leq z_p(\tilde{\pi}) \\ \lambda_p^\pi - \alpha & \text{otherwise.} \end{cases}$$

Normalize the weights such that $\lambda_1^\pi + \lambda_2^\pi + \lambda_3^\pi = 1$.

(b) Determine the type of neighborhood move used in the current iteration: Disjunctive Move or Basic Move (details are given below).

Randomly generate a neighborhood π' of this type and calculate the cost Θ if the generated neighborhood move is performed.

(c) If π' is non-dominated by π then update \mathcal{P}_{pot}^3 .

(d) Accept π' instead of π with probability

$$P(\pi, \pi', \lambda, T) = \min \left\{ 1, \exp \left(-a \cdot \max_{p=1,2,3} \frac{\lambda_p [z_p(\pi') - z_p(\pi)]}{b \cdot T} \right) \right\},$$

where a and b are constants that define the acceptance rate.

4. If the condition for changing the annealing temperature is fulfilled decrease T using predetermined cooling scheme.

5. If termination criterion is valid, then stop; otherwise go to 3.

The set of potentially Pareto optimal solutions \mathcal{P}_{pot}^3 is empty at the beginning of the method. It is updated after a new solution is generated. Updating the set of potentially Pareto optimal solutions with solution π consists of: (1) adding π to \mathcal{P}_{pot}^3 if no solution in \mathcal{P}_{pot}^3 dominates π and (2) removing from \mathcal{P}_{pot}^3 all solutions dominated by π . The process of updating \mathcal{P}_{pot}^3 may be very time consuming. Thus, several simple additional techniques could be used in order to reduce the time required for updating (for details see [30]).

Let q represent the number of iterations of PSA. For computational purposes we use the following parameters: initial annealing temperature $T_0 = 2$; cooling schedule $T_q = 0.998^q \cdot T_0 \cdot n$; termination criteria: $T_q \leq 0.001$ or $q \geq 10,000$ or $|\mathcal{P}_{pot}^3| \geq 20$; the constants a and b which define the acceptance probability are set to 2 and 0.5, respectively.

The detailed analysis of computational experiments is performed in part II [?] of the paper.

5 Conclusions

This paper deals with the FGSP under multiple criteria. The problem is modelled as a special multi-mode project scheduling problem with unary resources. We propose to attack the problem with a metaheuristic approach based on Pareto Simulated Annealing in order to solve the problem effectively and efficiently in reasonable time.

The development of other metaheuristic methods which are well suited for FGSP under multiple objectives is an important area of future research. Another interesting direction is to incorporate in the model possible earliness or tardiness of flights and construct a solution which is flexible to such type of uncertainty of parameters.

The set of potentially Pareto optimal solutions generated by PSA may contain a large number of elements. Therefore, there is need for a supplementary procedure that yields a unique best compromise solution according to the preferences of a decision maker. One might either employ sound mathematical techniques or some interactive procedures oriented on feedback from the decision maker. Techniques belonging to the first category are based on lexicographic ordering and predetermined, fixed reference points (see, e.g. [35]), whereas interactive procedures based on light beam search depend on psychology rather than mathematics (see, e.g., [25]). Such procedures take into account local preferences with respect to a given sample of solutions. The preferences' influence is restricted to one phase of calculations. The major advantages of interactive procedures are both simple format of preferences and visibility of the output for the decision maker thus increasing confidence in the final results.

Acknowledgements

We would like to thank Ulrich Dorndorf and Erwin Pesch who provided us with valuable suggestions as well as constructive comments on a preliminary version of the paper. Furthermore, we are indebted to Harm Brandt and Jan Spitzmann for the professional help in coding the algorithms.

References

- [1] E.H.L. Aarts and J.H.M. Korst, *Simulated annealing and Boltzmann machines*, Wiley, Chichester, 1989.
- [2] M. Bartusch, R.H. Möhring and F.J. Radermacher, "Scheduling project networks with resource constraints and time windows". *Annals of Operations Research* 16, 201 – 240 (1988).
- [3] J. Błażewicz, J.K. Lenstra and A.H.G. Rinnoy Kan, "Scheduling projects subject to resource constraints: classification and complexity". *Discrete Applied Mathematics* 5, 11 – 24 (1983).
- [4] F. Bector, "Resource-constrained project scheduling by simulated annealing". *International Journal of Production Research* 34, 2335 – 2351 (1996).
- [5] K. Bouleimen and H. Lecocq, "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version". *European Journal of Operations Research* 149, 268 – 281 (2003).

- [6] P. Brucker, A. Drexl, R. Möhring, K. Neumann and E. Pesch, "Resource-constrained project scheduling: notation, classification, models and methods". *European Journal of Operational Research* 112, 3 – 41 (1999).
- [7] P. Brucker, S. Knust, A. Schoo and O. Thiele, "A branch-and-bound algorithm for resource-constrained project scheduling". *European Journal of Operational Research* 112, 262 – 273 (1998).
- [8] J.-H. Cho and Y.-D. Kim, "A simulated annealing algorithm for resource constrained project scheduling problems". *Journal of the Operational Research Society* 48, 736 – 744 (1997).
- [9] N. Christofides, R. Alvarez-Valdes, J.-M. Tamarit, "Project scheduling with resource constraints: a branch and bound approach". *European Journal of Operations Research* 29, 262 – 273 (1987).
- [10] P. Czyzak and A. Jaskiewicz, "Pareto simulated annealing". In *Multiple Criteria Decision Making*, G. Fandel, T. Gal (eds.), Proceedings of the XIIth International Conference, Hagen (Germany), Springer, Berlin–Heidelberg, 297 – 307 (1997).
- [11] P. Czyzak and A. Jaskiewicz, "Pareto simulated annealing – a metaheuristic technique for multiple-objective combinatorial optimization". *Journal of Multi-Criteria Decision Analysis* 7, 34 – 47 (1998).
- [12] E. Demeulemeester and W. Herroelen, "A branch-and-bound procedure for the multiple resource-constrained project scheduling problem". *Management Science* 38, 1803 – 1818 (1992).
- [13] H. Ding, A. Lim, B. Rodrigues and Y. Zhu, "New heuristics for the over-constrained airport gate assignment problem". *Journal of the Operational Research Society* 55, 760 – 768 (2004).
- [14] H. Ding, A. Lim, B. Rodrigues and Y. Zhu, "Aircraft and gate scheduling optimization at airports". In Proceedings of the 37th Annual Hawaii International Conference on System Sciences, IEEE, 74 – 81 (2004).
- [15] H. Ding, A. Lim, B. Rodrigues and Y. Zhu, "The over-constrained airport gate assignment problem". *Computers & Operations Research* 32, 1867 – 1868 (2004).
- [16] U. Dorndorf. *Project scheduling with time windows: from theory to applications*. Physica-Verlag, Heidelberg–New York 2002.
- [17] U. Dorndorf, A. Drexl, Y. Nikulin and E. Pesch, "Flight gate scheduling: state-of-the-art and recent developments". Working Paper No. 584, University of Kiel 2004.
- [18] U. Dorndorf and E. Pesch, "Stand Allocation at Airports", Working Paper, University of Siegen 2005. (in preparation)
- [19] U. Dorndorf, E. Pesch and T. Phan Huy, "A time-oriented branch-and-bound algorithm for the resource-constrained project scheduling problem with generalised precedence constraints". *Management Science* 46, 1365 – 1384 (2000).

- [20] U. Dorndorf, E. Pesch and T. Phan Huy, "A branch-and-bound algorithm for the resource-constrained project scheduling problem". *Mathematical Methods of Operations Research* 52, 413 – 439 (2000).
- [21] A. Drexel and Y. Nikulin, "Multicriteria airport gate assignment and Pareto simulated annealing". Working Paper No. 586, University of Kiel 2005.
- [22] M. Ehrgott, *Multicriteria optimization*. Springer, Berlin 2000.
- [23] M. Garey and D. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San-Francisco, CA, 1979.
- [24] A. Geoffrion, "Proper efficiency and the theory of vector maximization". *Journal of Mathematical Analysis and Applications* 22, 618 – 630 (1968).
- [25] M. Hapke, A. Jaskiewicz and R. Słowiński, "Interactive analysis of multiple-criteria project scheduling problems". *European Journal of Operational Research* 107, 315 – 324 (1998).
- [26] M. Hapke, A. Jaskiewicz and R. Słowiński, "Fuzzy multi-mode resource-constrained project scheduling with multiple objectives". In *Recent Advances in Project Scheduling*, J. Węglarz (ed.), Kluwer Academic Publishers, Boston, 355 – 382 (1998).
- [27] S. Hartmann, "Project scheduling with multiple modes: A genetic algorithm". *Annals of Operations Research* 102 111 – 135 (2001).
- [28] S. Hartmann and A. Drexel, "Project scheduling with multiple modes: a comparison of exact algorithms". *Networks* 32, 283 – 297 (1998).
- [29] S. Hartmann and R. Kolisch, "Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem". *European Journal of Operational Research* 127, 394 – 407 (2000).
- [30] A. Jaskiewicz, *Multiple objective metaheuristic algorithms for combinatorial optimization*. Habilitation thesis, Poznan University of Technology, 2001.
- [31] J. Józefowska, M. Mika, R. Różycki, G. Waligóra and J. Węglarz, "Simulated annealing for multi-mode resource-constrained project scheduling". *Annals of Operations Research* 102, 137 – 155 (2001).
- [32] S. Kirkpatrick, "Optimization by simulated annealing – quantitative studies". *Journal of Stat. Phys.* 34, 975 – 986 (1984).
- [33] S. Kirkpatrick, C. Gelatt and M. Vecchi, "Optimization by simulated annealing". *Science* 220, 671 – 680 (1983).
- [34] R. Kolisch and A. Drexel, "Local search for nonpreemptive multi-mode resource-constrained project scheduling". *IIE Transactions* 29, 987 – 999 (1997).
- [35] R. Landelma, K. Miettinen and P. Salminen, "Reference point approach for multiple decision makers". *European Journal of Operational Research* 162, 785 – 791 (2005).
- [36] J.-K. Lee and Y.-D. Kim, "Search heuristics for resource-constrained project scheduling". *Journal of the Operational Research Society* 47, 678 – 689 (1996).

- [37] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, "Equation of state calculations by fast computing machines". *Journal of Chemical Physics* 21, 1087 – 1092 (1953).
- [38] M. Mika, G. Waligóra and J. Węglarz, "Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models". *European Journal of Operational Research* 162, 639 – 668 (2005).
- [39] K. Neumann, C. Schwindt and J. Zimmermann, *Project scheduling with time windows and scarce resources*, 2nd Edition, Springer, Berlin 2003.
- [40] S. Panwalkar, R. Dudek and M. Smith, "Sequencing research and the industrial scheduling problem". In *Symposium on the theory of scheduling and its applications*, S.E. Elmaghraby (ed.), Springer, New-York 1973.
- [41] J.H. Patterson, R. Słowiński, F.B. Talbot and J. Węglarz, "Computational experience with a backtracking algorithm for solving a general class of precedence and resource constrained project scheduling problem". *European Journal of Operational Research* 49, 68 – 79 (1990).
- [42] P. Serafini, "Simulated annealing for multiple objective optimization problems", in: *Proceedings of the 10th International Conference on Multiple Criteria Decision Making*, Taipei, 87 – 96 (1992).
- [43] P. Serafini, "Simulated annealing for multiple objective optimization problems", in *Multiple Criteria Decision Making*. G.H. Tzeng, H.F. Wang, V.P. Wen, P.L. Yu (eds.), Springer, Berlin, 283 – 292 (1994).
- [44] R. Słowiński, B. Soniewicki and J. Węglarz, "DSS for multiobjective project scheduling". *European Journal of Operational Research* 79, 220 – 229 (1994).
- [45] A. Sprecher and A. Drexel, "Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm". *European Journal of Operational Research* 107, 431 – 450 (1998).
- [46] R. Steuer, *Multiple criteria optimization – theory, computation and applications*. Wiley, New York, 1986.
- [47] J.P. Stinson, E.W. Davis and B.M. Khumawala, "Multiple resource-constrained scheduling using branch and bound". *AIIE Transactions* 10, 252 – 259 (1978).
- [48] V. T'Kindt and J.-C. Billaut, *Multicriteria scheduling: theory, models and algorithms*. Springer, Berlin–New York (2002).
- [49] J. Xu and G. Bailey, "The airport gate assignment problem: mathematical model and a tabu search algorithm". In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, IEEE, 2001.
- [50] S. Yan and C. Huo, "Optimization of multiple objective gate assignments". *Transportation Research* 35A, 413 – 432 (2001).