Decentralized Ground Staff Scheduling

Martin Dedenroth Sørensen
Carmen Consulting AB
Fruebjergvej 3, Boks 248
DK-2100 Copenhagen, Denmark
martin.sorensen@carmenconsulting.com

Jens Clausen
Informatics and Mathematical Modelling
Technical University of Denmark
DK-2800 Kongens Lyngby, Denmark
jc@imm.dtu.dk

11th October 2002

Abstract — Typically, ground staff scheduling is centrally planned for each terminal in an airport. The advantage of this is that the staff is efficiently utilized, but a disadvantage is that staff spends considerable time walking between stands. In this paper a decentralized approach for ground staff scheduling is investigated. The airport terminal is divided into zones, where each zone consists of a set of stands geographically next to each other. Staff is assigned to work in only one zone and the staff scheduling is planned decentralized for each zone. The advantage of this approach is that the staff work in a smaller area of the terminal and thus spends less time walking between stands. When planning decentralized the allocation of stands to flights influences the staff scheduling since the workload in a zone depends on which flights are allocated to stands in the zone. Hence solving the problem depends on the actual stand allocation but also on the number of zones and the layout of these. A mathematical model of the problem is proposed, which integrates the stand allocation and the staff scheduling. A heuristic solution method is developed and applied on a real case from British Airways, London Heathrow Airport. This study shows that decentralization generally increases the number of staff needed compared to centralized planning. The case study also shows that there is a trade-off between the extra staff needed and the quality of the stand allocation. Furthermore, the robustness of solutions with respect to disruptions deteriorates with the number of zones. In practice, the "cost" of introducing decentralization should therefore be weighted against the benefits gained.

Keywords: decentralization, ground staff scheduling, stand allocation, zoning, simulated annealing.

Introduction

For the airline industry, efficient scheduling is vital in order to improve productivity and profitability. Ground staff at airports comprises a large portion of an airline's workforce and hence the manpower planning process for these are important. The work carried out by the ground staff includes among many other tasks passenger check-in, boarding of passenger, loading and un-loading of baggage, etc. Some of the work, e.g. at the check-in counter, is usually carried out at one or few locations in the airport, whereas flight dependent work, e.g. associated with boarding passengers or loading and un-loading baggage, is carried out at the many gates/stands located around the terminals.

Ground staff scheduling for flight dependent work is typically planned centralized for each terminal in the airport. The advantage of this is that the staff can be efficiently utilized, but a disadvantage is that staff spends considerable time on transportation between the stands/gates. Another disadvantage is the organizatorial problems with managing a large number of staff working the in terminal.

In this paper, a decentralized ground staff planning method is investigated focusing on the flight dependent work. The airport terminal is divided into zones, where each zone consists of a set of stands/gates geographically next to each other. Staff is assigned to work in only one zone and the staff scheduling is planned decentralized in each zone. The advantage of this approach is that the staff work in a smaller area of the terminal and thus spends less time walking between gates/stands. Other benefits can be improved capability of monitoring performance, improved team morale and better service of customers. The disadvantages can be less efficient utilization of staff due to the decentralization.

When planning decentralized the allocation of stands to flights influences the staff scheduling since the workload in a zone depends on which flights are allocated to stands in the zone. Hence solving the decentralized ground staff scheduling problem depends on the actual stand allocation but also on the number of zones and the layout of these.

The contribution of this paper is twofold. A mathematical model for decentralized ground staff scheduling — *the Zoning Problem* — is presented. This model integrates the stand allocation and the staff scheduling problems. Secondly, a heuristic solution method is presented and applied on a real case from British Airways, London Heathrow Airport.

Section 1 is a survey of related work. In Section 2 the mathematical model for the Zoning Problem is described and in Section 3 solution methods are discussed. In Section 4 results obtained with the solution methods applied to real data from British Airways is presented. In section 5 is the conclusion.

1 Related Work

1.1 Stand Allocation

Previous research on the stand allocation includes Braaksma 1977, Babic 1984, Mangoubi and Mathaisel 1985, Bihr 1990 and Haghani and Chen 1998, who propose integer programming models with the objective of minimizing passenger walking time in the terminals. When minimizing passenger walking time, the stand allocation problem is a quadratic assignment problem. The problem is where to locate a facility (aircraft in this case) at a certain site (stand) and the cost of the allocation depends not only on the distance from other facilities, but also on the interaction with other facilities (passenger flow).

Other researchers also focus on disruptions in scheduled activities and hence take into consideration a dynamic aspect. Yan and Huo 2001 formulate a dual-objective zero-one integer programming model for the stand allocation. The first objective is minimization of passenger walking time and the second objective is to minimize passenger waiting time. They argue that, e.g. in peak hours, an aircraft might have to wait for an available gate/stand, and hence passengers have to wait on the aircraft until a gate is available. Bolat 1999 and Bolat 2000 is primarily considering the dynamic aspects (and not passenger walking time). He is focusing on the fact that flight schedules in reality often are interrupted by unanticipated events such as severe weather conditions, mechanical failures, late arrivals, etc. A delayed departure may delay the arrival of another aircraft scheduled into the same stand/gate, or require the flight to be reassigned to another stand/gate. He formulates a mixed integer programming model with the objective of minimizing the difference between the maximum and minimum idle times. The idea is to spread idle times evenly among stands to obtain solutions which are more insensitive to disruptions. In Bolat 2000, a quadratic objective function for minimizing the variance of idle times is proposed.

Cheng 1998 presents another model for the stand allocation model where the focus is on congestion problems, e.g. at apron taxiways or during push-back operations of aircraft. He proposes a network-based simulation model and apply this to real data.

Vanderstraeten and Bergeron 1988 consider off-pier allocations (allocation of flights to stands not directly placed at the terminal). From the passengers' point of view, off-pier allocations may be inconvenient and hence they should be avoided if possible.

1.2 Staff Scheduling

In many service organizations the utilization of labour has long been recognized as one of the key elements of successful management and a significant amount of research have been carried out on staff scheduling. Staff scheduling can be classified into (1) days-off, (2) shift, and (3) tour (see Bechtold and Brusco 1994). The objective of the days-off scheduling is to specify work and non-work days for employees. The focus of shift scheduling is to determine a set of work schedules for employees across a daily planning horizon. The work schedules specify daily start and finish times (shifts) for employees. Tour-scheduling typically integrates the days-off and shift scheduling for an organization with a weekly planning horizon. The tour-scheduling problem is the most complex, and most of the research lies within this area.

Dantzig 1954 was the first to formulate staff scheduling problems as linear programs with decision variables representing the number of employees assigned to specific schedules (determining shifts). Brusco and Jacobs 1998 describe a two-stage heuristic solution strategy for staff scheduling. The first stage is to determine the shift starting times where the total number of shift starting times is limited. The second stage is to determine the number of staff on each shift and also patterns of shifts to assign to individual staff. Research on tour-scheduling can be found e.g. in Brusco et al. 1995, Brusco and Johns 1995, Brusco and Johns 1996, Brusco 1998, Segal 1974, Berg and Panton 1994.

Brusco and Jacobs 2001 formulate an integer programming model for the tour scheduling problem where the number of shift starting times are restricted. They apply the model to different cases from different industries, and they find that limiting the number of shift starting times to four or five did not result in a substantial increase in the workforce size.

1.3 Zoning Problem

To best of the authors' knowledge no previous work have been published focusing on the *Zoning Problem* described in this paper — i.e. the integration of the stand allocation problem and the staff scheduling problem.

2 Problem Description and Formulation

We consider a planning horizon of one week since an airline company's flight timetable are roughly identical from week to week (within a season). Each day is divided into equal length time intervals (e.g. of 15 minutes). The number of these is denoted P. The total number of time intervals in a week is denoted T, so $T = 7P$.

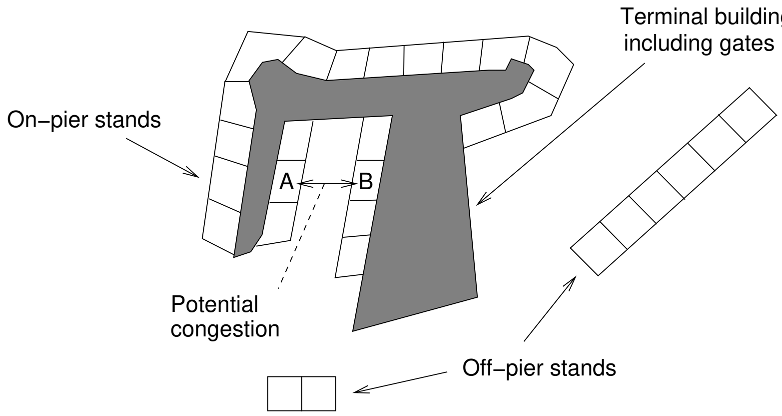

Figure 1: Example of a terminal.

Before the Zoning Problem is described, the two underlaying problems — the stand allocation problem and the staff scheduling problem — are described.

2.1 Stand Allocation Problem

The *Stand Allocation Problem* (SAP) consists of allocating stands to the flights arriving and departing from the terminal. The stands in the terminal are indexed $\{1, \ldots, S\}$ and are characterized by the size and whether they are on-pier or off-pier:

σ_s	size of stand s (non-negative integer)
π_s	1 if stand s is off-pier 0 otherwise

The difference between on-pier and off-pier stands is that on-pier stands are located next to the terminal building whereas off-pier stands are some distance from the terminal building as illustrated in Figure 1. The flights to be assigned to the stands are indexed $\{1, \ldots, F\}$ and modelled as:

α_f	size of flight f (non-negative integer)
G_{ft}	1 if flight f is at a stand in time interval t, 0 otherwise
$ArrT_f$	arrival time interval of flight f
$DepT_f$	departure time interval of flight f

The coefficients G_{ft} defines the ground time for the flights. G_{ft} is equal to one if $ArrT_f \leq t \leq DepT_f$ and zero otherwise. Without loss of generality it is assumed that the flights are ordered chronologically such that $f_1 < f_2 \rightarrow ArrT_{f_1} \leq ArrT_{f_2}$. The allocation of a flight to a stand is modelled with the binary decision variable x_{fs}, which is equal to one if flight f is allocated to stand s, otherwise zero.

The constraints for the stand allocation are:

$$\sum_{s=1}^{S} x_{fs} = 1, \ \forall f \tag{1}$$

$$\sum_{f=1}^{F} x_{fs} G_{ft} \leq 1, \ \forall s, t \tag{2}$$

$$x_{fs}(\sigma_s - \alpha_f) \geq 0, \ \forall s, f \tag{3}$$

Equation (1) ensures that all flights are allocated to exactly one stand. Equation (2) guarantees that no more than one flight is allocated to the same stand at the same time. Equation (3) ensures that the capacity (size) of the stand is feasible.

The objective for the stand allocation is:

$$\min \ \omega_1 \text{CONG} + \omega_2 \text{STANDPREF} + \omega_3(-\text{IDLETIME}) + \omega_4 \text{OFFPIER} \tag{4}$$

Each of the terms is multiplied with a weight $(\omega_1, \omega_2, \omega_3, \omega_4)$. The first term, CONG, is a measure of congestion problems for the stand allocation. Congestion can occur if two flights, which are departing at the same time, are allocated to stands A and B as illustrated in Figure 1, since there is not enough space for taxing both flights to the runways at the same time (push-back congestion). Consequently, one of the flights will be delayed. Preferably, congestions should be avoided. The term is calculated as:

$$\text{CONG} = \sum_{q=1}^{Q} \sum_{s_1=1}^{S} \sum_{s_2=1}^{S} \sum_{f_1=1}^{F} \sum_{f_2=1}^{F} \text{SR}^q_{s_1 s_2} \text{FR}^q_{f_1 f_2} x_{f_1 s_1} x_{f_2 s_2} \tag{5}$$

The coefficients $\text{SR}^q_{s_1 s_2}$ and $\text{FR}^q_{f_1 f_2}$ defines the congestion conflicts. For example, the congestion conflict sketched in Figure 1 can be defined with the coefficient SR^q_{AB} equal to one (or larger) and $\text{FR}^q_{f_1 f_2}$ equal to one for pairs of flights departing at the same time (or e.g. within 10 minutes). Different types of congestion conflicts can be presents and are indexed with q.

The second term, STANDPREF, measures how well preferences for the stand allocation are fulfilled — certain flights may have preferred stands and the reasons for these preferences can be business considerations, e.g. to offer good service to "valuable" customers. The term is calculated as:

$$\text{STANDPREF} = \sum_{s=1}^{S} \sum_{f=1}^{F} \text{SP}_{fs} x_{fs} \tag{6}$$

The coefficients SP_{fs} defines the stand preferences.

The third term, IDLETIME, concerns robustness of the stand allocation. Idle time is the time interval from the (scheduled) time of departure to the (scheduled) time of arrival of the subsequent flight at a stand. Obviously, a delayed departure from a stand will not disrupt the subsequent allocation to this stand, if the idle time to the next arrival is longer than the delay. In other words, the minimum idle times should be maximized as to hopefully absorb any disruptions. Since the total scheduled ground time of flights and the total available time of stands are constant, this goal can be achieved if the idle times are distributed evenly among the stands. But instead of minimizing the variance of idle times the surrogate of maximizing the minimum idle time is used. The minimum idle time is found as:

$$\text{IDLETIME} = \min_{s, f_1, f_2 > f_1} \{ArrT_{f_2} - DepT_{f_1} - 1 + (1 - x_{f_1 s})M + (1 - x_{f_2 s})M\} \tag{7}$$

were M is a large number.

The last term, OFFPIER, concerns off-pier allocations. Off-pier allocations can be perceived as giving poor customer service, and in general off-pier allocations should be avoided (passengers would have to walk or be coached to flights at off-pier stands). If off-pier allocations are necessary, no destination, origin or flight number should be consistently allocated to off-pier stands. Thereby the off-pier allocations are distributed "fairly". The term is calculated as:

$$\textsc{OffPier} = \max\left[\max_j\{\mathrm{OPA}_j\}, \max_j\{\mathrm{OPD}_j\}, \max_l\{\mathrm{OPN}_l\}\right] \tag{8}$$

$$\mathrm{OPA}_j = \frac{1}{|\mathcal{A}_j|}\sum_{s=1}^{S}\sum_{f\in\mathcal{A}_j} x_{fs}\pi_s \tag{9}$$

$$\mathrm{OPD}_j = \frac{1}{|\mathcal{D}_j|}\sum_{s=1}^{S}\sum_{f\in\mathcal{D}_j} x_{fs}\pi_s \tag{10}$$

$$\mathrm{OPN}_l = \frac{1}{|\mathcal{N}_l|}\sum_{s=1}^{S}\sum_{f\in\mathcal{N}_l} x_{fs}\pi_s \tag{11}$$

OPA_j, OPD_j and OPN_l are the ratios of off-pier allocations and total allocations of flights arriving from origin $j \in \Omega$, departing to destination $j \in \Omega$, with flight number $l \in \Phi$, respectively. $\mathcal{A}_j$, $\mathcal{D}_j$ and $\mathcal{N}_l$ are index sets of flights arriving from origin j, departing to destination j, with flight number l, respectively. OffPier is the value of the largest of these ratios and a value of one implies that all flights from a specific (or several) origin/destination or with a specific flight number are allocated to off-pier stands.

2.2 Staff Scheduling Problem

The staff scheduling problem (SSP) considered in this paper is limited to calculating the workload and determining anonymous shifts to cover the workload. That is, rostering is not considered (the process of constructing lines of work/shifts for individual employees). Only fulltime employees are considered all working the same shift length (e.g. 8 hour shifts), but specific work qualifications, meal break, leave, etc. are not considered. Furthermore, only flight dependent work is considered like e.g. employees working at the gates during boarding or employees loading or un-loading baggage. The staff scheduling can be made independently for different categories of staff, e.g. passenger service staff and baggage handlers.

The workload W_t^c in time interval t is calculated based on the airline's engagement standards. For each flight the required number of staff needed is defined in the engagement standards (e.g. 3 staff members needed in 30 minutes from time of arrival). With centralized planning, the workload is constant and calculated by adding the staff requirement for each flight:

$$\mathrm{W}_t^c = \sum_{f=1}^{F} \mathrm{R}_{ft}^c, \forall t, c \tag{12}$$

R_{ft}^c is the number of staff from staff category c required for flight f in time interval t.

When planning decentralized, the workload is calculated separately for each zone by adding the workload requirement for each flight allocated to stands in the given zone (which will be described later).

The shift starting times must be the same each day of the week and a maximum of θ different shifts on one day is allowed. The reason for limiting the number of different shifts is mainly due to the administrative burden associated with managing a large number of shift starting times, but can also be conducive to a team work environment. The shifts can be overlapping, but must be feasible so that they cover the workload. The shift length (number of time intervals) is denoted λ.

Two sets of decision variables are used. The binary variable v_p^c is equal to one if the shift starting in time interval p is used. The integer variable y_t^c is equal to the number of staff assigned to *start* work in time interval t.

The workload for any time interval must be covered:

$$\sum_{m=t-\lambda+1}^{t} y_m^c \geq \mathrm{W}_t^c,\ \forall c, t \geq \lambda \tag{13}$$

$$\sum_{m=1}^{t} y_m^c + \sum_{m=T-\lambda+t+1}^{T} y_m^c \geq \mathrm{W}_t^c,\ \forall c, t < \lambda \tag{14}$$

Equation (13) specify that staff assigned to shifts which overlap a given time interval contribute to cover the workload in that time interval. Equation (14) specify that workload in the beginning of the planning horizon is also covered by staff starting work in the end of the planning horizon because of the cyclic scheduling.

The number of shifts allowed is constrained to θ:

$$\sum_{p=1}^{P} v_p^c \leq \theta,\ \forall c \tag{15}$$

Staff is only allowed to start working in a time interval if a shift starting in this time interval is chosen:

$$\mathrm{M} v_p^c - y_t^c \geq 0,\ \forall t, c,\ p = (t \bmod P) + 1 \tag{16}$$

where M is a large number.

The objective function for the staff scheduling problem is to minimize the total number of shifts that needs to be worked (for all categories of staff) while covering the workload:

$$\min \sum_{c=1}^{C} \sum_{t=1}^{T} y_t^c \tag{17}$$

The different categories of staff are indexed $\{1, \ldots, C\}$.

2.3 Zoning Problem

After having described the stand allocation problem and the staff scheduling problem, the model of the zoning problem can be formulated.

Zones are indexed $\{1, \ldots, Z\}$ and a zone Z_i is defined as a set of stands, $Z_i \subseteq \{1, \ldots, S\}$. The decision variable z_{si} is equal to one if stand s is assigned to zone i, otherwise zero. Each stand must belong to exactly one zone:

$$\sum_{i=1}^{Z} z_{si} = 1,\ \forall s \tag{18}$$

Thus a zoning is a partitioning of the stands. The minimum size of a zone is given by $Z_{\min}$ and is the minimum number of stands in a zone:

$$\sum_{s=1}^{S} z_{si} \geq Z_{\min},\ \forall i \tag{19}$$

The reason to specify a minimum zone size is that a zone consisting of only a few stands is not practically feasible.

Zones must be "connected", that is, stands in a zone must be geographically next to each other. Whether two stands are next to each other, is defined by a stand neighbour graph $G(\{1, \ldots, S\}, E, w)$ — see Figure 2. Each stand

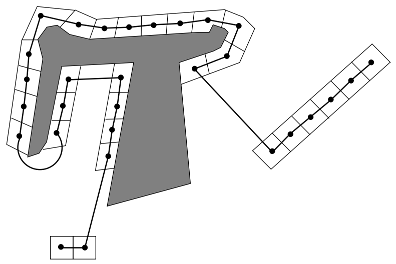

Figure 2: Example of a stand neighbour graph.

corresponds to a vertex in the graph and two vertices are connected with an edge if and only if the corresponding stands are considered to be close to each other. Each edge in the graph has a weight of one and the graph is assumed to be a connected graph.

The graph $GZ_i(Z_i, E_i, w)$ is defined as the subgraph of the stand neighbour graph $G(\{1, \dots, S\}, E, w)$ generated by Z_i:

$$GZ_i(Z_i, E_i, w) = G[Z_i] \tag{20}$$

and the cut, $\delta(A_i)$, generated by A_i on GZ_i is defined as:

$$\delta(A_i) = \{(s_1, s_2) \in E_i | s_1 \in A_i, s_2 \in Z_i \backslash A_i\} \tag{21}$$

where

$$A_i \subseteq Z_i, \emptyset \neq A_i \neq Z_i \tag{22}$$

The weight of the cut generated by A_i is defined as:

$$w(\delta(A_i)) = \sum_{(s_1,s_2) \in E_i, s_1 \in A_i, s_2 \in Z_i \backslash A_i} w_{s_1 s_2} \tag{23}$$

Ensuring that all zones are connected is obtained by:

$$w(\delta(A_i)) \geq 1, \ \forall A_i \subseteq Z_i, \emptyset \neq A_i \neq Z_i \tag{24}$$

This specify that the weight of any cut $\delta(A_i)$ must be at least one, i.e., for any partition of a zone in two parts there must be an edge with weight one connecting the two parts (the subgraphs GZ_i must be connected).

When introducing zones, the ground staff scheduling is planned separately for each zone and the workload in a zone Z_i is calculated as:

$$\mathrm{W}^c_{t,i} = \sum_{s=1}^{S} \sum_{f=1}^{F} R^c_{ft} x_{fs} z_{si}, \ \forall t, c, i \tag{25}$$

The objective of the zoning problem is to obtain a "good" stand allocation and minimize the total number of staff needed (all zones) and is modelled as:

$$\begin{aligned} \min \quad & \omega_1 \text{ Cong} + \omega_2 \text{ StandPref} + \omega_3(-\text{IdleTime}) + \\ & \omega_4 \text{OffPier} + \omega_5 \text{ Workforce} \end{aligned} \tag{26}$$

The first four terms are the terms from the stand allocation model and the last term is the minimization of the staff needed:

$$\text{Workforce} = \sum_{i=1}^{Z} \left[\sum_{c=1}^{C} \sum_{t=1}^{T} y^c_{t,i} \right] \tag{27}$$

2.3.1 Zoning Model

The complete zoning model consists of the stand allocation model and a staff scheduling model for each zone:

$$\begin{aligned} \min \quad & \omega_1 \textsc{Cong} + \omega_2 \textsc{StandPref} + \omega_3(-\textsc{IdleTime}) + \\ & \omega_4 \textsc{OffPier} + \omega_5 \textsc{Workforce} \end{aligned} \tag{28}$$

subject to:

$$\textsc{Cong} = \sum_{q=1}^{Q} \sum_{s_1=1}^{S} \sum_{s_2=1}^{S} \sum_{f_1=1}^{F} \sum_{f_2=1}^{F} \mathrm{SR}^q_{s_1 s_2} \mathrm{FR}^q_{f_1 f_2} x_{f_1 s_1} x_{f_2 s_2} \tag{29}$$

$$\textsc{StandPref} = \sum_{s=1}^{S} \sum_{f=1}^{F} \mathrm{SP}_{fs} x_{fs} \tag{30}$$

$$\textsc{IdleTime} = \min_{s, f_1, f_2 > f_1} \{ArrT_{f_2} - DepT_{f_1} - 1 + (1 - x_{f_1 s})M + (1 - x_{f_2 s})M\} \tag{31}$$

$$\textsc{OffPier} = \max \left[\max_j \{\mathrm{OPA}_j\}, \max_j \{\mathrm{OPD}_j\}, \max_l \{\mathrm{OPN}_l\} \right] \tag{32}$$

$$\textsc{Workforce} = \sum_{i=1}^{Z} \left[\sum_{c=1}^{C} \sum_{t=1}^{T} y^c_{t,i} \right] \tag{33}$$

$$\sum_{s=1}^{S} x_{fs} = 1, \ \forall f \tag{34}$$

$$\sum_{f=1}^{F} x_{fs} G_{ft} \leq 1, \ \forall s, t \tag{35}$$

$$x_{fs}(\sigma_s - \alpha_f) \geq 0, \ \forall s, f \tag{36}$$

$$\mathrm{OPA}_j = \frac{1}{|\mathcal{A}_j|} \sum_{s} \sum_{f \in \mathcal{A}_j} x_{fs} \pi_s, \ \forall j \tag{37}$$

$$\mathrm{OPD}_j = \frac{1}{|\mathcal{D}_j|} \sum_{s} \sum_{f \in \mathcal{D}_j} x_{fs} \pi_s, \ \forall j \tag{38}$$

$$\mathrm{OPN}_l = \frac{1}{|\mathcal{N}_l|} \sum_{s} \sum_{f \in \mathcal{N}_l} x_{fs} \pi_s, \ \forall l \tag{39}$$

$$\sum_{m=1}^{t} y^c_{m,i} + \sum_{m=T-\lambda+t+1}^{T} y^c_{m,i} \geq \mathrm{W}^c_{t,i}, \ \forall i, c, t < \lambda \tag{40}$$

$$\sum_{m=t-\lambda+1}^{t} y^c_{m,i} \geq \mathrm{W}^c_{t,i}, \ \forall i, c, t \geq \lambda \tag{41}$$

$$\mathrm{M} v^c_{p,i} - y^c_{t,i} \geq 0, \ \forall i, t, c, \ p = (t \bmod P) + 1 \tag{42}$$

$$\sum_{p=1}^{P} v^c_{p,i} \leq \theta, \ \forall i, c \tag{43}$$

$$\mathrm{W}_{t,i}^c = \sum_{s=1}^{S}\sum_{f=1}^{F} R_{ft}^c x_{fs} z_{si}, \ \forall i,t,c \tag{44}$$

$$\sum_{i=1}^{Z} z_{si} = 1, \ \forall s \tag{45}$$

$$\sum_{s=1}^{S} z_{si} \geq Z_{\min}, \ \forall i \tag{46}$$

$$Z_i = \{1,\ldots,S\} \cap \{s|z_{si} = 1\} \tag{47}$$

$$GZ_i(Z_i, E_i, w) = G\left[Z_i\right] \tag{48}$$

$$w(\delta(A_i)) \geq 1, \ \forall A_i \subseteq Z_i, \emptyset \neq A_i \neq Z_i \tag{49}$$

$$w(\delta(A_i)) = \sum_{(s_1,s_2)\in E_i, s_1\in A_i, s_2\in Z_i\setminus A_i} w_{s_1 s_2}, \ \forall A_i \tag{50}$$

$$s, s_1, s_2 \in \{1,\ldots,S\}, \ f, f_1, f_2 \in \{1,\ldots,F\}, \ t \in \{1,\ldots,T\}$$

$$p \in \{1,\ldots,P\}, \ c \in \{1,\ldots,C\}, \ q \in \{1,\ldots,Q\}, \ i \in \{1,\ldots,Z\}$$

$$\mathcal{A}_j \subset \{1,\ldots,F\}; \ \mathcal{D}_j \subset \{1,\ldots,F\}; \ \mathcal{N}_l \subset \{1,\ldots,F\}, \ j \in \Omega, l \in \Phi$$

$$x_{fs} \in \{0,1\}, \ z_{si} \in \{0,1\}, \ v_{p,i}^c \in \{0,1\}, \ y_{t,i}^c \geq 0 \text{ and integer}$$

The model is a non-linear mixed integer programming model. The non-linearity stems from the quadratic terms modelling the congestion conflicts and from the equations specifying the workload in each zone. The model needs to be refined if it is to be solved with a standard solver like e.g. CPLEX or MINOS. The constraints ensuring the connectivity of the zones defined by (49) constitute one obvious problem. These correspond to connectivity constraints in the Travelling Salesman Problem and the number grows exponentially with the number of stands in a zone.

3 Solution Approach

Due to the complexity of the zoning model, implying that a realistic case has a very large number of variables and constraints, it is not in practice feasible to solve the problem with exact methods. Instead, a heuristic solution approach is considered which is sketched in Figure 3.

The input to the algorithm is the number of zones, the minimum zone size, the number of shift starting times (θ), a flight timetable, the weights of the terms in the objective function and the stopping criteria. An initial zoning and stand allocation is constructed which then is improved. Optimization of a zoning is done with SAP Optimizer (Stand Allocation Problem), which uses the SSP Algorithm (Staff Scheduling Problem). When a given zoning is optimized it is changed with the Zone Generator (one stand is moved from one zone to another) and the new zoning is optimized. The overall stopping criteria for the algorithm is a time-out, which means that the algorithm terminates after a specified time interval.

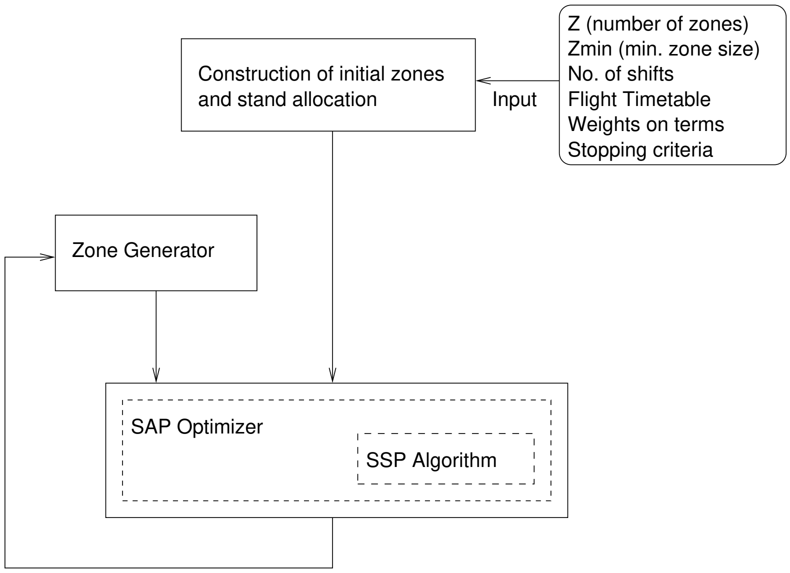

Figure 3: Algorithm for solving the zoning problem.

3.1 Construction of initial zoning and stand allocation

The initial zoning is found by assigning all the stands to one zone. If the number of zones (Z) is greater than one and the minimum zone size (Z_{min}) is greater than zero, then this zoning is infeasible since there are one or more zones with zero stands. In this case the initial zoning is repaired so that it becomes feasible. This is done with an algorithm based on Simulated Annealing where a neighbourhood solution is found by moving a random stand from one zone to another zone while ensuring that the zones stays "connected" and that stands are not moved from a zone with less than $Z_{min} + 1$ stands. The objective function is the number of stands that should be added to each zone so that the zones fulfill the constraint for the minimum zone size.

When the initial zoning is found, an initial stand allocation is constructed. First all the flights are allocated to stands by iterating through the flights (in a random fashion) and ensuring the constraints are fulfilled. If an available stand cannot be found for a flight, the time constraints are relaxed. When all flights are allocated to stands, the current solution is repaired so that no time constraints are violated. This is done by applying a Simulated Annealing to the non-feasible solution, where a time-infeasibility term is added to the objective function. The algorithm is stopped when the solution is feasible. The time-infeasibility term measures to which extent the time constraints are violated. The neighbourhood used is the same as in SAP Optimizer and is described in the following. In general there is no guarantee that a feasible solution exists at all, but for the data sets which have been considered, obtaining feasibility has not been a problem.

3.2 SAP Optimizer

The SAP Optimizer optimizes a given zoning with respect to the weights on the terms in the objective function. SAP Optimizer is an algorithm based on Simulated Annealing and a neighbourhood solution is found by moving one flight from one stand to another or swapping two flights between two stands. If flights are moved between two zones, the workloads in the two zones change and evaluating the neighbourhood solution requires resolving the staff scheduling problems for both zones. The staff scheduling problems are solved with the SSP

Algorithm, which is described later.

The simulated annealing is implemented without plateau ($nrep = 1$) and the temperature reduction function α is chosen as the geometric reduction function: $\alpha(t_i) = a \cdot \alpha(t_{i-1}), a < 1$. The actual values of the initial temperature and the constant a are chosen as $t = 1.0$ and $a = 0.999$ after pilot experiments. The stopping criteria is a sliding window where the algorithm is terminated if the improvement in the objective function for the last WINDOWSIZE iterations is less than CUTOFF percent.

The neighbourhood structure used consists of moving a flight from one stand to another stand. Typically it is not possible to move the flight to a new stand since another flight is already occupying that stand (violation of the time constraints). Hence it is inspected whether it is possible to swap the flight with the flight already at the stand. The selection of the initial flight to move is done in two ways: (randomly) with a probability of 50% the initial flight is randomly chosen and (guided) with a probability of 50% it is chosen from a random zone, but among flights within the zone which contribute to peaks in the workload for that zone. In the latter case the flight is moved to a stand in another zone, otherwise it moved to a random stand. By selecting flights in this way, it is attempted to balance the workload profiles for the zones, which can improve the solutions to the staff scheduling.

3.3 SSP Algorithm

The staff scheduling model considered in this paper is relative simple and one possibility, which have been considered, is to use a standard solver, e.g. CPLEX, to solve the models for each zone. The problem is however, that the staff scheduling model needs to be solved a very large number of times since the workload changes in the zones when flights are moved between zones and hence the computation time for solving the problem must be very small. A SSP Algorithm (Staff Scheduling Algorithm) have been developed and it has turned out to be more efficient than using e.g. CPLEX.

The algorithm performs a greedy search through combinations of shift starting times to find the combination with the lowest number of staff needed to cover the workload.

3.3.1 Shift Starting Times

The workload pattern for many airports/terminals is discontinuous. That is, staff is only needed in a time interval shorter than 24 hours on a day, for example from 4:15 to 22:15 on a day. In this case the optimal shift starting times for the first (p_{s_1}) and last shift (p_{s_θ}) can easily be determined; The first shift starts at the earliest time interval on a day where staff is needed and similar the last shift ends in the latest time interval on a day where staff is needed (illustrated in Figure 4).

The remaining shift starting times are then found in the interval $]p_{s_1}; p_{s_\theta}[$ (indicated in the figure with the dotted box). The number of possible combinations of these depends on the actual values of p_{s_1}, p_{s_θ} and the number of shifts starting times (θ). It is sufficient to consider combinations where $p_{s_2} < p_{s_3} < p_{s_4} \cdots$, that is shift 2 starts before shift 3, and shift 3 before shift 4 etc. In Figure 5 the number of combinations are shown depending on the number of shifts (θ) with $p_{s_1} = 18$ (4:15) and $p_{s_\theta} = 55$ (13:45).

If the total number of shifts is 4 or 5, or even 6, the number of combinations is relatively small and it is thus feasible to enumerate all the combinations.

3.3.2 Assigning Staff to Shifts

The other part of the staff scheduling problem is to determine the number of staff to assign to each shift. If the given shift starting times are denoted

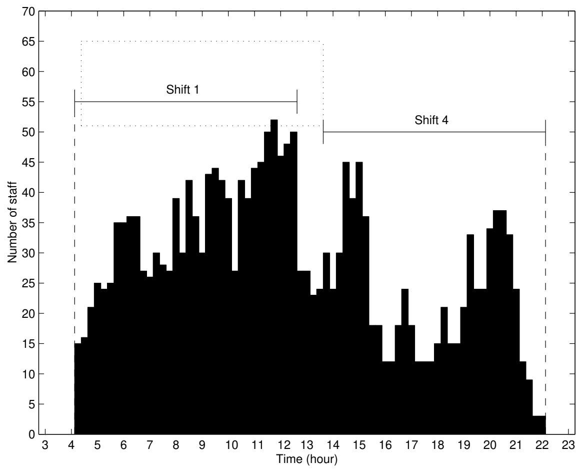

Figure 4: When the shift starting times for first and last shift are found ($\theta = 4$), the shift starting times for the remaining shifts are to be found within the dotted box.

θ	**Combinations**
3	36
4	630
5	7140
6	58905
7	376992

Figure 5: Combinations of shift starting time versus the number of shift starting times. Based on the first shift 4:15–12:45 and last shift 13:45–22:15.

$p_{s_1}, p_{s_2}, \ldots, p_\theta$ then the minimum number of staff needed on day one on shift 1 is:

$$\hat{y}_{p_{s_1}} = \max_t \{W_t\}, t \in [p_{s_1}; p_{s_2}[\tag{51}$$

It cannot be lower since this part of workload is only covered by shift 1. On the other hand, in an optimal solution, it does not need to be higher. If $y_{p_{s_1}}$ is increased, a surplus of staff is introduced in the time interval $[p_{s_1}; p_{s_2}[$. The extra staff can cover workload in the time interval $[p_{s_1}; p_{s_1} + \lambda - 1]$. But the extra staff could just as well be added to shift 2 and then the extra staff can also cover workload in the time interval $[p_{s_2}; p_{s_2} + \lambda - 1]$. So an optimal assignment of staff to shift 1 on day one is found by Equation (51). Similarly, the optimal number of staff assigned to shift 1 on the other days (d) is:

$$\hat{y}_{p_{s_1}+(d-1)P} = \max_t \{W_t\}, t \in [(d-1)P + p_{s_1}; (d-1)P + p_{s_2}[, d \in \{1, \ldots 7\} \tag{52}$$

When the optimal assignment of staff to shift 1 on each day is found, the remaining uncovered workload $\hat{W}_t$ is calculated as:

$$\hat{W}_t = \begin{cases} \max\left\{0, W_t - \hat{y}_{p_{s_1}+(d-1)P}\right\}, t \in [p_{s_1} + (d\text{-}1)P;\ p_{s_1} + \lambda\text{-}1 + (d\text{-}1)P] \\ W_t, \text{otherwise} \end{cases} \tag{53}$$

Figure 6 shows the remaining uncovered workload when staff is assigned to shift 1 (one day).

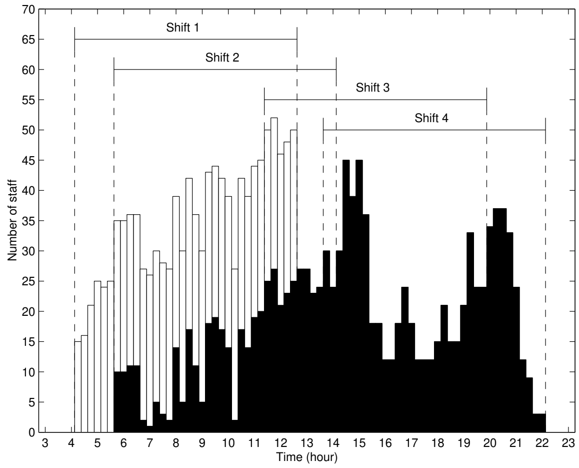

Figure 6: The remaining uncovered workload when staff is assigned to shift 1.

When staff is assigned to shift 1, the remaining problem is similar to the initial problem, but now there are only $\theta - 1$ shifts (each day) to which to assign staff. Hence the optimal assignment of staff to shift 2 is found in the same way as for shift 1, but now using the remaining uncovered workload $\hat{W}_t$ instead of the initial workload W_t:

$$\hat{y}_{p_{s_2}+(d-1)P} = \max_t\{\hat{W}_t\}, t \in [(d-1)P + p_{s_2}; (d-1)P + p_{s_3}[, d \in \{1, \dots 7\} \quad (54)$$

This procedure is repeated for shift 3, ..., $\theta - 1$, while using the remaining uncovered workload $\hat{W}_t$, which is updated after each assignment to a shift. The optimal assignment of staff to the last shift θ is found as:

$$\hat{y}_{p_{s_\theta}+(d-1)P} = \max_t\{\hat{W}_t\}, t \in [(d-1)P + p_{s_\theta}; (d-1)P + p_{s_\theta} + \lambda - 1], d \in \{1, \dots 7\} \quad (55)$$

That is, the last shift covers whatever workload that is left (illustrated in Figure 7).

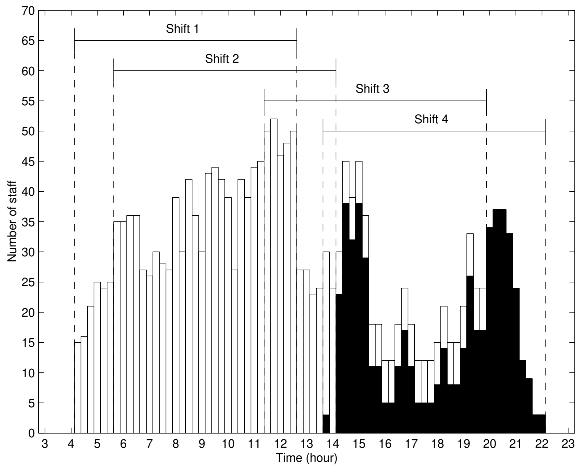

Figure 7: The remaining uncovered workload when staff is assigned to shift 1, 2 and 3.

Summarizing, the Staff Scheduling Algorithm works by enumerating all combinations of shift starting times, where each combination is evaluated, i.e. the staff assignment is determined. The shift starting times with the lowest total number of staff needed is the optimal solution and this is returned.

3.4 Zone Generator

The process of optimizing the zoning is iterative. The starting point is a zoning feasible with respect to connectivity and minimum zone size. Each iteration consists of randomly selecting a stand and moving this to another zone while ensuring that the zones remain connected and the minimum zone size constraints are fulfilled. Connectivity of a zone i is inspected by making a depth first search of the subgraph GZ_i of the stand neighbour graph (G). When a stand is moved from one zone to another, any flights assigned to the stand remain at the stand, so the flights are also moved to the new zone. This new zoning is then optimized with the SAP Optimizer and the SSP Algorithm.

To avoid optimizing the same zoning several times a "zone memory" is used. The zone memory is a circular buffer where each element in the memory is a string representation of a zoning.

Instead of using this random search through all possible zoning combinations, it has been considered to swap between two optimization methods: 1) fixing the zones and optimizing the stand allocation and staff scheduling and 2) fixing the stand allocation and optimizing the zones and staff scheduling. This approach did not work very well. The reason for this is that moving one stand (including the flights at the stand) from one zone to another zone changes the workload in the zones considerably.

3.5 Implementation Details

In the implementation, the number of occurrences of OFFPIER and IDLETIME are also calculated and included in the terms. The reason is that a solution with the same value of e.g. IDLETIME may have different number of occurrences of this value. Clearly, a solution with a lower number of occurences is better. Furthermore, each term in the objective function is scaled between zero and one such that selecting the weights is more easy.

4 Results

The zoning model has been applied to a real case from British Airways, London Heathrow Airport. The terminal considered contains 31 stands (of this 8 off-pier). Three different data sets (F1, F2, F3) have been used with about 1000 flights each. Two different stand neighbour structures have been used, see Figure 8. Stand neighbour structure A require the off-pier stands to be assigned to a few zones, whereas the off-pier stands will be more evenly distributed between the zones in stand neighbour structure B. Note that the figure only shows the main idea of the two structures, but not the actual layout of the terminal.

The combinations of number of zones and minimum zone size which have been investigated are listed in Table 1. Furthermore, two different weight sets have been used, see Table 2. The number of shifts is $\theta = 4$ (for each zone and category of staff) and two categories of staff are considered, passenger service staff and baggage handler staff.

The computations have been carried on a SUN Fire 3800 with 8 750MHz Sparc III CPUs and 8GB RAM running Solaris 8. However, the implementation does only use one CPU at the time. The stopping criteria is a time-out after 45 minutes and a window size on 500 with a cut-off on 0.001.

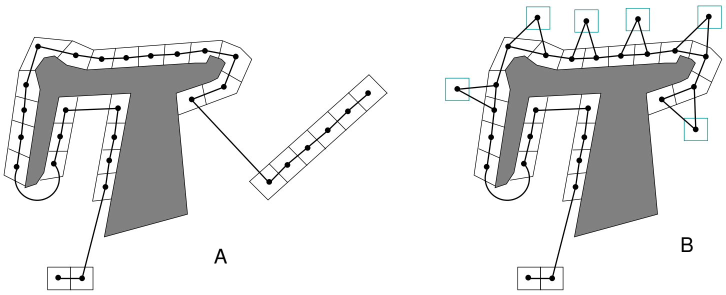

Figure 8: Stand neighbour structure A and B. Note that the figure only shows the main idea of the two structures, but not the actual layout of the terminal.

Minimum Zone Size	**Number of zones**				
	1	2	3	4	5
6	•	•	•	•	•
7		•	•	•	
8		•	•		
10		•	•		
15		•			

Table 1: Combinations of the number of zones and the minimum zone size which are investigated.

Term	**Weight Set**	
	1	2
CONG	10	0
STANDPREF	20	0
OFFPIER	22	0
IDLETIME	4	0
WORKFORCE	40	100

Table 2: The two weight sets which are used.

4.1 Excess Workforce

The box-plot in Figure 9 shows the excess workforce versus the number of zones. The excess workforce is calculated as the percentage of extra staff needed compared to the number of staff needed with one zone.

When introducing zoning, i.e. decentralized staff scheduling, one might expect the excess workforce to increase with the number of zones since zoning imposes restrictions on the staff scheduling. Generally, this is also the case as the figure shows. With 2 zones the excess workforce is about 1% on average and with 5 zones it is about 9%.

However, with 2 zones there are a few solutions where the excess workforce is below zero. That is, with 2 zones it is possible to decrease the number of staff needed, but the reduction is only marginal (about -0.3%). The reason is that with two zones the restrictions imposed by the zones are not binding and at the same time, additional shift starting times become available compared to solutions with one zone.

The figure also shows that for a given number of zones the variance of the excess workforce is large. This can be explained by the different weight sets and the flight data sets as Figure 10 shows. This figure shows the average excess workforce versus the number of zones for weight set 1 and 2 and flight data set

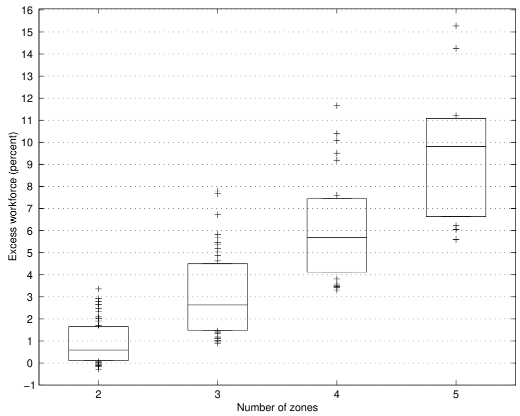

Figure 9: Box-plot showing the excess workforce versus the number of zones. Each box has lines at the lower quartile, median, and upper quartile values. Outlier values are plotted with '+'.

F1 and F2.

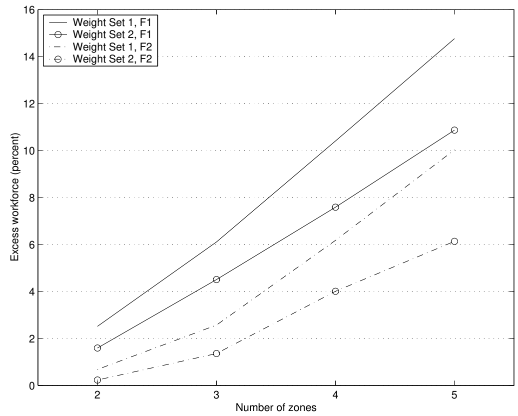

Figure 10: Average excess workforce versus the number of zones for weight set 1 and 2 and flight data set F1 and F2. Data set F3 is similar to F2.

The excess workforce is generally higher for flight data set F1 than F2, which can be explained by the fact that they are from different seasons. Furthermore, the excess workforce is lower with weight set 2 than with weight set 1 for all number of zones and for both flight data sets. Weight set 2 is only optimizing the workforce term, whereas weight set 1 is also optimizing the other terms. This indicates that there is a trade-off between the quality of the stand allocation and the excess workforce.

4.2 Layout of Zones

Figure 11 shows a box-plot where the excess workforce is plotted versus the number of zones and the minimum zone size (measured as the number of stands).

The figure shows that for a fixed number of zones, the average excess workforce is roughly the same for different minimum zone sizes. Either the minimum zone size is not binding or it is optimal to have zones of roughly the same size. So, in general the choice of the minimum zone size has no influence on the excess workforce.

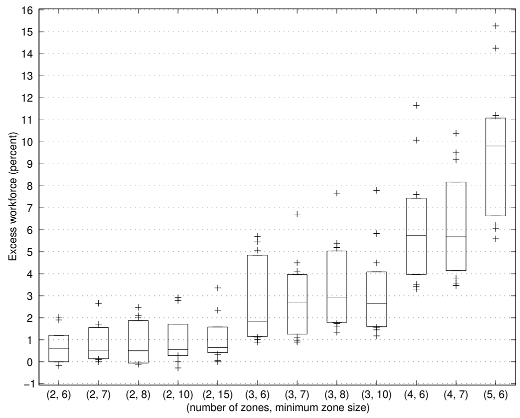

Figure 11: Box-plot showing excess workforce versus the number of zones and the minimum zone size.

Figure 12 shows a box-plot where the excess workforce is plotted versus the number of zones and stand neighbour structure A and B. The average excess

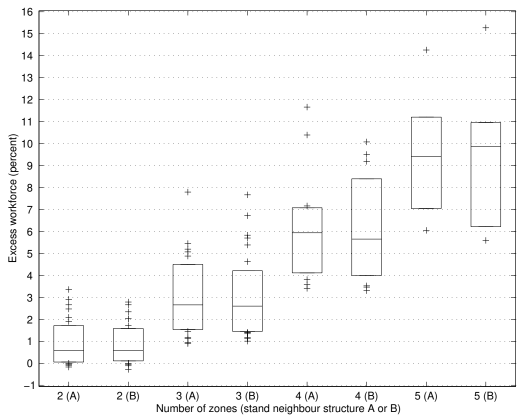

Figure 12: Box-plot showing excess workforce versus the number of zones and stand neighbour structure A and B.

workforce for a fixed number of zones is approximately the same for both stand neighbour structures. So using stand neighbour structure A or B does not influence the excess workforce.

The zonings found with the two stand neighbour structures are different, hence the actual layout of the zones is not important for the excess workforce (but of course the number of zones is). This is maybe not so remarkable since most of the stands are identical and able to service all flights.

4.3 Quality of Stand Allocation

4.3.1 Off-pier

Figure 13 shows the average value of the OffPier term versus the number of zones for weight set 1 and 2. For comparison, the best (optimal) off-pier ratio found in the experiments is plotted.

As the figure shows the off-pier ratio for weight set 1 is constantly about 0.6 for 2–5 zones. This corresponds approximately to maximum 4 out of 7 flights compared to the optimum of 2 out of 7 flights. So by introducing zones,

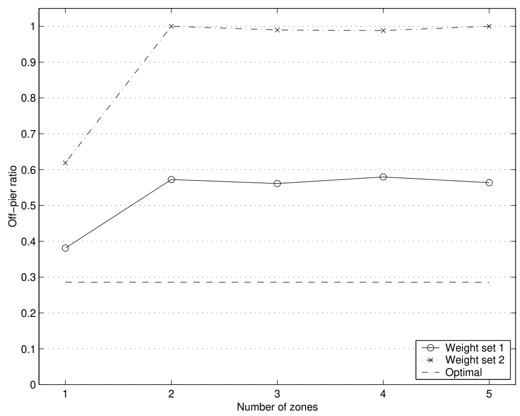

Figure 13: Average value of the OffPier term versus the number of zones for weight set 1 and 2.

the off-pier term increases compared to the optimal value, but consistent off-pier allocations for a destination/origin/flight number are still avoided. However, for weight set 2 the average off-pier term is 1 for $2-5$ zones, and hence some flights are consistently allocated to off-pier stands. (An off-pier value of 1 indicates that e.g. all flights from some origin are allocated to off-pier stands.)

4.3.2 Idle Time

In Figure 14 the quality of the IdleTime term is illustrated. All solutions have a minimum idle time of zero. So instead of plotting the minimum idle time, the average number of occurrences of the minimum idle time versus the number of zones are plotted.

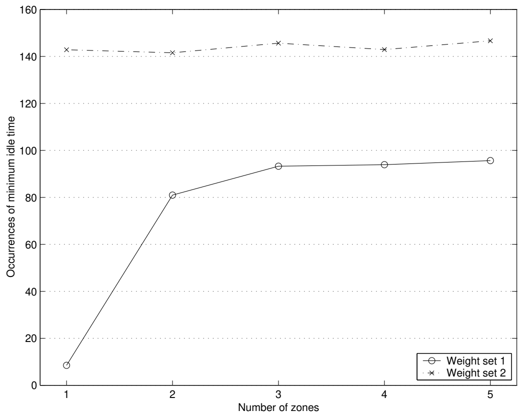

Figure 14: Average number of occurrences of the minimum idle time versus the number of zones. All solutions have a minimum idle time of zero.

The figure shows that introducing zones increases the number of occurrences of the minimum idle time. With weight set 1 the number of occurrences increases from about 10 to almost 100. This is a rather large increase, but the weight on the idle time term for weight set 1 is also only 4. However, if not optimizing the term at all (weight set 2), the number of occurrences is more than 140 for 1–5 zones.

4.3.3 Congestion

Figure 15 shows the average value of the CONG term versus the number of zones for weight set 1 and 2. The CONG term is the total number of congestion conflicts.

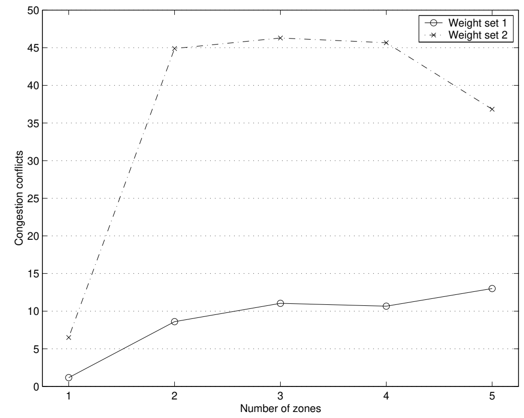

Figure 15: Average value of the CONG term versus the number of zones for weight set 1 and 2.

The figure shows that for weight set 1 the number of congestions conflicts generally increases with the number of zones, but only slowly. The optimal value of the term is zero and with 5 zones the number of congestion conflicts is about 10–15. If not optimizing the term at all (weight set 2), the number of congestions conflicts is about 35–45.

4.3.4 Stand Preferences

Figure 16 shows the average value of the STANDPREF term versus the number of zones for weight set 1 and 2. The STANDPREF term is the number of flights allocated to non-preferred stands.

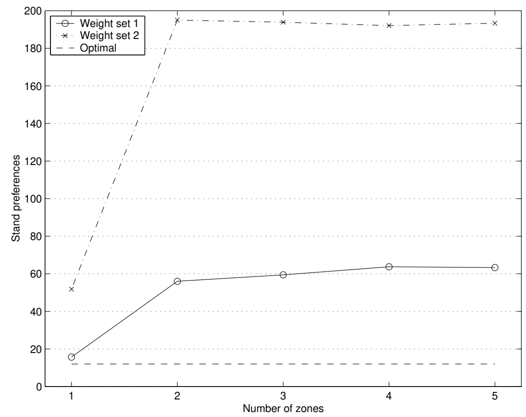

Figure 16: Average value of the STANDPREF term versus the number of zones for weight set 1 and 2.

The figure shows a similar trend as for the CONG term; The number of non-preferred allocations increases from about 55 to 65 for 2 to 5 zones for weight set 1. But if not optimizing the term at all (weight set 2), it is almost 200. Note that the total number of stand preferences is about 300.

Summarizing, when introducing zones the quality of the stand allocation generally deteriorates when also optimizing the workforce term. For weight set 2, the stand allocation is really poor (not optimized at all), but for weight set 1 it is better. On the other hand, the excess workforce with weight set 2 is lower than with weight set 1. So the selection of weights on the stand allocation terms and the workforce term is a trade-off between the quality of the stand allocation and the level of excess workforce.

4.4 Robustness

Robustness of solutions are important since disruptions inevitable occur in reality, i.e. flights arrive and depart earlier and later than scheduled. The zoning model does not include any specific terms for robustness except from the minimum idle time term which could give some robustness in the stand allocation.

In this section it is analyzed how robust zonal solutions are with respect to the stand allocation and the staff scheduling. For the stand allocation it is analyzed how well a zonal plan can absorb early and late arrivals and departures. For example, if a flight (at a given stand) is departing later than scheduled, it is inspected whether that will disrupt the next arriving flight allocated to the given stand. For the staff scheduling it is analyzed whether the planned workforce for a zonal plan can handle disruptions. For example, if a flight is arriving later than scheduled, the requirement of staff for the flight is shifted forward in time (to the actual arrival time) and then the planned level of staff may not be sufficient to cover the actual workload.

The analysis is made by applying historical data of actual arrival and departure times to the flight data sets. For each of the flight data sets, a representative week of historical data is chosen. By representative is understood that the late and early arrivals and departures for this week are "typical" (i.e. no extreme delays or special circumstances, etc.).

4.4.1 Stand Allocation

For a given zonal plan, the robustness of the planned stand allocation is evaluated by considering the actual arrival and departure times (historical data) of flights instead of the scheduled times. When using the actual arrival and departure times, the planned stand allocation might not be feasible with respect to the time constraints (Equation (2) in the model) and hence not robust. Robustness is measured as:

$$SAPRobust = \sum_{s=1}^{S}\sum_{t=1}^{T}\left[\max\left(\sum_{f} x_{fs}\hat{G}_{ft} - 1, 0\right)\right] \tag{56}$$

where $\hat{G}_{ft}$ is the ground time matrix for flights based on the actual (historical) arrival and departure times and the x_{fs}'s are the planned stand allocations based on the scheduled arrival and departure times (G_{ft}). If $SAPRobust$ is zero then the planned stand allocation is robust enough to absorb any of the disruptions. Otherwise, higher values of $SAPRobust$ mean less robustness in the stand allocation. For example, if $SAPRobust$ is equal to 40 it corresponds to 40 time intervals with two flights allocated to the same stand at the same time.

In Figure 17 the average value of the robustness term versus the number of zones is plotted for weight set 1 and 2.

As the figure shows, for weight set 1 the $SAPRobustness$ for one zone is about 80 and increases with the number of zones. With 5 zones it is 130. However, with weight set 2 the values are above 140 for 1 to 5 zones. Only for weight set 1, the minimum idle time is optimized and when comparing this figure with Figure 14 there seems to be a correspondence between the minimum

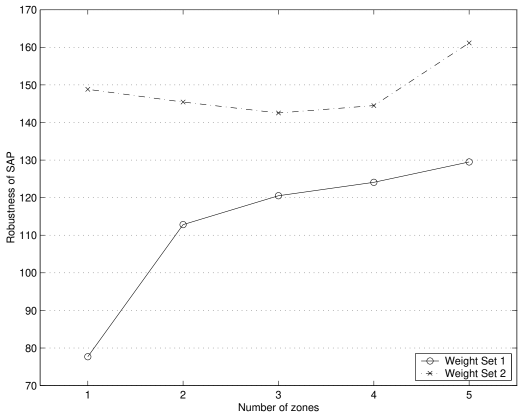

Figure 17: Robustness of the stand allocation versus the number of zones for weight set 1 and 2.

idle time (the number of occurrences) and the robustness. This suggests that optimizing the minimum idle time can provide some robustness for the stand allocation.

4.4.2 Staff Scheduling

The robustness of the staff scheduling is evaluated in a similar way. The *actual workload* based on the actual arrival and departure times is calculated. That is, if a flight arrives later than scheduled the requirement of staff for the flight is shifted forward in time to the actual arrival time. Similar for early arrivals and early and late departures. The planned workforce is compared to the actual workload and the robustness is measured as the total shortfall of staff for all zones and time intervals. For example, if the actual workload in a time interval is 50 and the planned workforce is 40, the shortfall is 10. Figure 18 shows an example of the planned workload, the actual workload and the planned workforce.

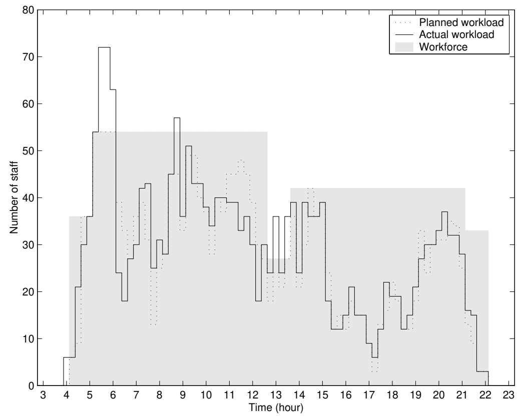

Figure 18: Illustration of planned workload, actual workload and planned workforce.

In Figure 19 the average shortfall of staff (in percent compared to the total workload) is plotted versus the number of zones for weight set 1 and 2.

The figure shows that the shortfall of staff increases with the number of zones. With one zone the average shortfall or uncovered workload is about

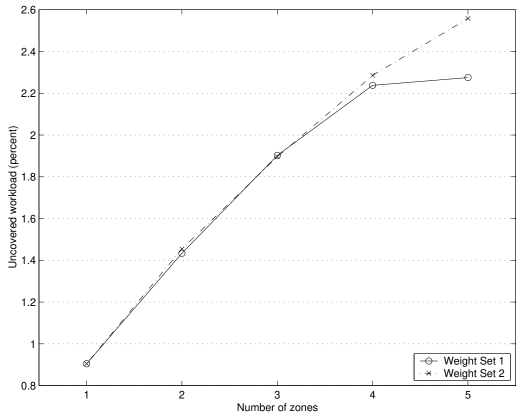

Figure 19: Average shortfall in staff (percent) versus the number of zones for weight set 1 and 2.

0.9% and with 4 zones is about 2.2% for both weight sets. For weight set 1 and 5 zones, the shortfall is a little bit lower than for weight set 2. This can be explained by the higher level of staff in solutions with weight set 1.

5 Conclusion

In this paper a problem regarding decentralized ground staff scheduling — the Zoning Problem — is defined and presented. A mathematical model is proposed and a heuristic solution method is developed. The model is basically a multi-criteria model with the different criteria aggregated into one objective function through weights.

The method has been implemented and applied to a real case from British Airways, London Heathrow Airport. Generally, analysis of the results shows that it is possible to quantify concepts like congestion, and by the adjustment of weights for the different objectives to control the structure of a solution. The analysis also shows that introducing decentralized ground staff scheduling or zoning, the number of staff required generally increases with the number of zones compared to centralized planning. The actual layout of zones does not seem to influence the excess workforce — for a fixed number of zones, the minimum zone size and stand neighbour structures is of minor importance to the excess workforce. As expected, there is a trade-off between the number of extra staff needed and the quality of the stand allocation, and the robustness of the solution decreases with the number of zones. Finally, evaluating whether implementing decentralized ground staff scheduling as defined with the zoning model is beneficial should not only depend on the cost in terms of excess workforce and the quality of the stand allocation, but also on the value/advantage of decentralization (as e.g. staff spending less time walking between the stands, improved capability of monitoring performance, and improved team morale).

Further research on the topic includes improving the model, especially concerning staff scheduling. Addressing robustness in the staff scheduling model is also relevant. The solution methods constitute a first cut solution and improvements are possible as e.g. a more sophisticated approach for searching through zonings. Other heuristics than Simulated Annealing should be tested, e.g. Genetic Algorithms, which has been extensively used in connection with multi-criteria optimization. Developing a more integrated model and using techniques as Dantzig-Wolfe Decomposition or Bender's Decomposition might also be feasible, although this approach is less obvious.

Acknowledgements

The authors would like to thank Alex Ross and Sarah Poynter from British Airways for providing data and being very helpful.

References

O. Babic, C. Teodorovic and V. Tosic, "Aircraft Stand Assignment to Minimize Walking", *Journal of Transportation Engineering* 110, 55–66 (1984).

S.E. Bechtold and M.J. Brusco, "Working set generation methods for labor tour scheduling", *European Journal of Operational Research* 74, 540–551 (1994).

Y. Berg and D.M. Panton, "Personnel Shift Assignment: Existence Conditions and Network Models", *Networks* 24, 385–394 (1994).

R.A. Bihr, "A conceptual solution to the aircraft gate assignment problem using 0,1 linear programming", *Computers & Industrial Engineering* 19, 280–284 (1990).

A. Bolat, "Assigning arriving flights at an airport to the available gates", *Journal of the Operational Research Society* 50, 23–34 (1999).

A. Bolat, "Procedures for providing robust gate assignments for arriving aircrafts", *European Journal of Operational Research* 120, 63–80 (2000).

J.P. Braaksma, "Reducing Walking Distances at Existing Airports", *Airport Forum* 4, 135–142 (1977).

M.J. Brusco, L.W. Jacobs, R.J. Bongiorno, D.V. Lyons and B. Tang, "Improving personnel scheduling at airline stations", *Operations Research* 43, 741–751 (1995).

M.J. Brusco and T.R. Johns, "Improving the dispersion of surplus labor in personnel scheduling solutions", *Computers and Industrial Engineering* 28, 745–754 (1995).

M.J. Brusco and T.R. Johns, "A sequential integer programming method for discontinuous labor tour scheduling", *European Journal of Operational Research* 95, 537–548 (1996).

M.J. Brusco and L.W. Jacobs, "Personnel tour scheduling when starting–time restrictions are present", *Management Science* 44, 534–547 (1998).

M.J. Brusco, "Solving personnel tour scheduling problems with the dual all–integer cutting plane", *IIE Transactions* 30, 835–844 (1998).

M.J. Brusco and L.W. Jacobs, "Starting–time decisions in labor tour scheduling: An experimental analysis and case study", *European Journal of Operational Research* 131, 459–475 (2001).

Y. Cheng, "Solving push-out conflicts in apron taxiways of airports by a network-based simulation", *Computers and Industrial Engineering* 34, 351–369 (1998).

G.B. Dantzig, "A comment on Edie's 'Traffic delays at toll booths'", *Journal of the Operations Research Society of America* 2, 339–341 (1954).

A. Haghani and M. Chen, "Optimizing gate assignments at airport terminals", *Transportation Research Part A* 32, 437–454 (1998).

R.S. Mangoubi and D.F.X Mathaisel, "Optimizing Gate Assignments at Airport Terminals", *Transportation Science* 19, 173–188 (1985).

M. Segal, "The operator–scheduling problem: a network–flow approach", *Operations Research* 22, 808–823 (1974).

G. Vanderstraeten and M. Bergeron, "Automatic assignment of aircraft to gates at a terminal", *Computers and Industrial Engineering* 14, 15–25 (1988).

S. Yan and C. Huo, "Optimization of multiple objective gate assignments", *Transportation Research Part A* 35, 413–432 (2001).