A Rolling Planning Horizon Heuristic for Scheduling Agents with Different Qualifications

Raik Stolletz & Emilio Zamorano[†]

Chair of Production Management, University of Mannheim, Business School, Mannheim, Germany

[†]ezamoran@mail.uni-mannheim.de

February 25, 2014

Abstract

This paper presents an optimization model for the tour scheduling problem for agents with multiple skills and flexible contracts in check-in counters at airports. The objective is to minimize the total assignment costs subject to demand fulfillment and labor regulations. In order to solve this problem we develop a rolling planning horizon-based heuristic. Our heuristic is robust and provides near-optimal schedules within reasonable computation time for real-world cases, although the parameter selection is important to its performance. In addition, we discuss the impact of the skill distribution on the scheduling costs for several instances.

1 Introduction

Third-party ground-handling agencies provide passenger handling services, such as check-in of passengers at airport terminals, to airlines. After the release in 1996 of Directive 96/67/EC on the liberalization of the ground-handling market at European airports, the number of third-party ground handlers increased by 90% in five years (SH&E, 2002), leading to higher competition and price reductions (Airport Research Center, 2009). Because staff costs represent 66% to 75% of ground handlers' operating costs, optimized workforce schedules significantly increase their profitability (Steer Davies Gleave, 2010).

A typical workforce planning process undergone by ground-handling agencies is divided into four sub-stages, each with different planning horizons, objectives, and constraints: (i) head count planning, (ii) tour scheduling, (iii) task assignment, and (iv) replanning (Stolletz, 2010). Our work focuses on the second phase of this process: monthly tour scheduling. This decision problem consists of assigning individual employees to tours (*e.g.*, daily shifts and days off) to satisfy the time-dependent employee requirements for check-in of individual flights. The requirements are driven by the flight schedule and depend on the contract with the airline. Airlines work with different computer systems for their check-in processes, and only qualified agents are allowed to operate them. Agents can specialize on the operation of a single system or can be cross-trained to operate multiple systems, although no downgrading is allowed (agents with higher skill levels cannot be assigned to cover lower skill requirements *i.e.*, agents have non hierarchical skill sets). In addition, agents are allowed to switch between different skills during a working shift. Agent assignment also must satisfy rules based on the contracts and qualifications of the employees. The contracts, however, enforce flexible schedules by a set of tour building rules: variable shift length and starting times, variable tour length, variable number of working hours, or non-consecutive days-off, for example. This leads to a huge amount of possible tours per employee. Such a combination of multiple non-hierarchical skills and highly flexible contracts has not been previously addressed in tour scheduling literature. The objective of this planning task is to minimize operative workforce costs: overall scheduled hours, overtime and outsourcing. The resulting schedules need to respect the given pool of agents and their individual contracts while ensuring that enough qualified employees are assigned to cover the demand over the course of each day.

This paper presents a Mixed Integer Programming (MIP) model for tour scheduling with multiple skills. The consideration of multiple skills prevents a solution for the MIP in short CPU times with standard software. Therefore, this study presents a new heuristic based on a rolling planning horizon approach. The main idea is to decompose the main tour scheduling problem for multiple weeks into smaller but well-connected subproblems that cover only part of the entire planning horizon. That is contrary to the common the decomposition into daily shift scheduling and a line of work construction (Ernst (2004); Alfares (2004)). Such a decomposition approach could lead to infeasibilities in the second step, especially for workforce with heterogeneous skills. Instead of looking at a planning horizon of a single day only, our heuristic extend it to several days. For such a short horizon, the resulting subproblem (which is the tour scheduling problem for a significantly shorter horizon) could be solved exactly.

To the best of our knowledge, no previous efforts have been directed toward applying a rolling planning horizon to the problem at hand.

To summarize, the main contributions of this paper are the following:

- The development of Mixed Integer Programming (MIP) model for the tour scheduling problem of a multiskilled workforce with flexible contracts, and
- the development of a new heuristic based on a rolling planning horizon approach for this problem.

This paper is organized as follows: Section 2 gives an overview of related work. Section 3 presents a description of the problem and the formulation as an MIP. The Rolling Planning Horizon heuristic (RPH) is described in Section 4. The numerical studies described in Section 5 demonstrate the reliability of the solution of the RPH compared to that of the MIP, obtained for data from different ground-handling agencies. A sensitivity analysis shows the impact of the skill distribution on the overall scheduling costs. Concluding remarks and managerial insights are presented in Section 6.

2 Previous Research

Reviews of decision models and solution approaches in workforce scheduling and, in particular, tour scheduling are given in Alfares (2004), Ernst (2004), Ernst et al. (2004), and Van den Bergh et al. (2013). The types of scheduling flexibility, as defined by Jacobs and Bechtold (1993) and Stolletz (2010), are: (i) shift length flexibility, (ii) tour length flexibility, (iii) meal-break flexibility, (iv) shift-start flexibility, (v) start-time float flexibility, (vi) days-off flexibility, and (vii) overall-hour flexibility. The more of these flexibility types are incorporated into a model, the higher the degree of scheduling flexibility the model has. In contrast to the tour scheduling problem which we consider in this paper, shift scheduling determines shifts and allocate them to agents for a single day. Avramidis et al. (2010) and Helber and Henken (2010) address the shift scheduling problem for multiskilled agents in call centers. They consider, similar to our approach, non-hierarchical skill sets and high degree of shift length and shift-start flexibility. However, this review concentrates on tour scheduling approaches. First, we review tour scheduling approaches considering a high degree flexibility although a homogeneous workforce is assumed. Second, we comment on approaches considering hierarchical skills and discuss on the degree of flexibility they assume. Third, we discuss more general models assuming non-hierarchical skill sets and different degrees of flexibility.

1.- Homogeneous workforce

Stolletz (2010) and Brunner and Stolletz (2014) consider the tour scheduling problem for check-in counters at airports with flexible contracts. Stolletz (2010) includes all aforementioned types of scheduling flexibility, except meal-break flexibility, and proposes a MIP model for this problem. Brunner and Stolletz (2014) extend this by including meal-break flexibility and develop a branch-and-price algorithm to solve this problem. In contrast to our approach, they both consider a homogeneous workforce.

2.- Heterogeneous workforce with hierarchical skills

Models for tour scheduling with multiple qualified agents often assume hierarchical skill sets. In this case, employees with higher skill levels are allowed to be assigned to jobs that require lower skill levels, *i.e.*, downgrading is allowed. The general non-hierarchical model can solve this as a special case. Billionnet (1999), Bard (2004), and Rong and Grunow (2009) assume hierarchical skills. In addition, they incorporate some scheduling flexibility types into their assumptions. Billionnet (1999) proposes a Integer Programming (IP) model to assign agents with hierarchical skills to work days and flexible days-off on a weekly basis. This models allows downgrading agents and ensures that each agent receives n days-off during a week. Bard (2004) addresses a scheduling problem for postal service personnel. In this model, full- and part-time agents are assigned to daily shifts of and days-off in a weekly basis. Shifts can have variable lengths and meal-breaks are allocated flexibly. Rong and Grunow (2009) consider full- and part-time employees in an air cargo terminal scheduling problem. They propose a MIP to assign full- and part-time workers to weekly schedules where the shift length and shift start times are flexible. These approaches differ to ours on their lower degree of scheduling flexibility and on the hierarchical skills assumption.

3.- Heterogeneous workforce with non-hierarchical skills

Existing models for tour scheduling that assume a workforce with non-hierarchical skills consider a very limited scheduling flexibility, as they consider few different shift types or do not consider tours of variable length (*e.g.*, Cai and Li (2000), Eitzen et al. (2004) and Detienne et al. (2009)).

Cai and Li (2000) considers the problem of scheduling staff with mixed non-hierarchical skills. They propose a multi-criteria optimization model and a genetic algorithm to solve it. Their approach differs from ours as they only consider two shift types and few possible shift start times.

Eitzen et al. (2004) provide a generalized set-covering formulation for the fortnightly scheduling of multiskilled full- and part-time employees. They propose three different solution techniques (column expansion, column subset, and branch and price) and conduct numerical tests for problem instances with different numbers of skill levels, workforce sizes and demand patterns. Compared to our model, their shift and tour building rules are less flexible (*i.e.*, a smaller number of tour combinations are possible), and skill switching during a shift is forbidden.

Detienne et al. (2009) propose two IP formulations for a timetabling problem for multiskilled employees. They present a Lagrangian lower bound and two cut generation based approaches to solve this problem. Their approach differs from ours in that they do not consider tour building rules but rather assign employees to a limited list of predefined work patterns (*i.e.*, tours).

Few examples of tour scheduling models assume agents with non-hierarchical skill sets and several types of scheduling flexibility. Love, Jr. and Hoey (1990), Loucks and Jacobs (1991) and Hojati and Patil (2011) address tour scheduling problems in the fast food industry. They consider a workforce with limited availability (*i.e.*, agents are eligible to work only during specific time windows), shifts and tours can vary in length, and days-off may not be consecutive. As with our application, agents are cross-trained and allowed to change the tasks they are assigned to during a shift. Their approaches differ from ours in that the shift-start flexibility is more limited, as shifts are allowed to start in fewer periods.

Love, Jr. and Hoey (1990)'s work differs from ours in that the surplus of working hours, skills, availability of employees and number of working days are modeled as coefficients of the shifts to be assigned in the objective function. In addition, their solution technique differs from ours in that it consists of decomposing the tour scheduling problem into two network flow subproblems: construction of tours and allocation of tours to employees with the objective of minimizing the surplus of manpower.

Loucks and Jacobs (1991), in contrast to our application, propose a multi-criteria optimization model to minimize the total man-hours of overstaffing and minimize the deviation from target working hours. Their solution approach mainly differs from ours in that shifts are determined by concatenating segments (each segment corresponding to a different task), while the shift and tour building rules (minimum shift length, maximum shift length, maximum number of working days, etc.) are modeled implicitly as constraints. Their approach also differs from ours in that they propose a two-phase procedure: (i) a construction phase in which rules are used to assign task-segments to employees while relaxing the maximum number of working days constraint and (ii) an improvement phase in which violations to this constraint are eliminated and other measures (overstaffing, deviation from target hours, etc.) are improved.

Hojati and Patil (2011) revisit the model proposed by Loucks and Jacobs (1991) and propose another solution approach. Their approach differs from ours in that they decompose the main problem into a two-phase algorithm: (i) determining good shifts via a linear program that maximizes the total number of eligible and available employees and (ii) assigning shifts to employees through the use of an integer linear programming-based heuristic that determines all the shifts to be assigned to an employee, one employee at a time.

To address the complexity of tour scheduling problems, the aforementioned papers propose different decomposition approaches. As Bartholdi (1981) shows, standard tour scheduling problems are already NP-complete. With the combination of flexible schedules and non-hierarchical skills, additional complexity is expected. The basis of our heuristic is the rolling planning horizon approach, an idea commonly used in production scheduling to provide partial production schedules (e.g., on a weekly basis) for longer planning periods; see Modigliani (1955), Baker (1977) and Baker and Peterson (1979). This approach can also be used to decrease the size of a planning problem by solving a series of multi-period subproblems that, in the end, cover the entire planning horizon of the original problem.

Aside from production scheduling, few efforts have been directed at applying a rolling planning horizon approach to workforce scheduling. Examples of general workforce scheduling problems using a rolling horizon approach are the allocation of flexible employees to workstations in a production line (Gronalt and Hartl, 2003) and reactive nurse scheduling (Bard and Purnomo, 2005). In the tour scheduling literature we find that Day and Ryan (1997) address a fortnightly tour scheduling problem for flight attendants for short-haul flights. In their solution method, they divide the main problem into a days-off allocation and lines of work construction subproblems. For the latter, they apply a two-phase procedure by which they (i) construct lines of work based on a days-off template and (ii) improve the lines of work using a branch-and-price algorithm. They make use of a rolling horizon based-procedure to link these two steps recursively.

3 Problem Description and Model Formulation

This study considers the tour scheduling problem for check-in counters with discontinuous schedules, *i.e.*, with no overlap between shifts on different days. The planning horizon spans D days ($d = 1,...,D$), and each day is divided into T periods ($t = 1,...,T$) of equal length. Based on the flight schedules and the contracts between the ground handler and the airlines, the agent requirements are known. These requirements r_{qdt} specify the number of employees with a specific skill q needed in check-in counters on day d in period t. The goal is to assign each employee e to a shift of type j on day d such that the agent requirements are covered and the overall scheduling costs are minimized.

With respect to multiple qualifications, each employee e can be differently qualified and cross-trained. The set of skills of each employee is reflected in the parameter g_{eq}, with a value of 1 if employee e has skill q and 0 otherwise.

The working conditions regarding schedules are established by the ground-handling agency in the employee contracts. The following regulations are considered:

- Shifts can vary in length between 3 and 10 hours.
- Shifts can start at any period t on day d and at different times on different days.
- A minimum resting time of R hours between shifts needs to be respected.
- A maximum of w consecutive workdays without a day off is allowed. Days off can be non-consecutive.
- There is no limit on the total working hours per week. The overall number of workdays must remain greater than d_e^{min} and less than d_e^{max} days during the planning horizon of D days.
- At least h_e^{min} and at most h_e^{max} working hours for a planning horizon of D days are allowed per employee. If h_e^{max} is exceeded, the respective overtime periods ot_e have an additional cost of c^{ot} per period. Similar to that constraint, one can formulate bounds $\bar{h}_e^{min}$ and $\bar{h}_e^{max}$ for the working hours in subhorizons $\bar{D}$, *e.g.*, weeks.

The operational costs consist of the following elements:

- Paid hours, *i.e.*, total working hours.
- Overtime, *i.e.*, total overtime hours.
- Outsourcing, *i.e.*, in case the requirements are not met in a particular period within a day, outsourcing additional resources is allowed. This ensures feasibility independent of the problem instance. This is, however, paired with an outsourcing cost c_q^{out}, which could be interpreted as the penalty for unmet requirements.

Meal-breaks are not explicitly considered. However, they could be included as non-working periods of shift type j.

A commonly used formulation for tour scheduling problems is the set-covering formulation proposed by Dantzig (1954); see also Ernst (2004). This MIP formulation uses a matrix with all possible combinations of tours (daily shifts and days off) as input that are to be assigned to employees to meet the requirements per period. However, in our case, the flexibility of the schedules and the heterogeneity of the workforce produces a tour matrix with a large amount of data, thus making the solution intractable. Stolletz (2010) proposes a Reduced Set-Covering (RSC) formulation for the tour scheduling problem with single skills. This approach requires a matrix of all possible daily shifts only. The tour-building rules (maximum and minimum overall working days, maximum consecutive working days, etc.) are then modeled implicitly. This study

Table 1: Notation

Indices:	
$t = 1, \ldots, T$	periods to be scheduled over a day
$d = 1, \ldots, D$	days of the planning horizon
$j = 1, \ldots, J$	shift types for a day
$e = 1, \ldots, E$	employees to be assigned to shifts
$q = 1, \ldots, Q$	skills
Parameters:	
r_{qtd}	employee requirements of skill q for period t of day d
a_{tj}	1, if period t of shift j is a working period, 0 otherwise
s_j	first working period of shift j
f_j	last working period of shift j
d_e^{min}	minimum number of workdays for employee e
d_e^{max}	maximum number of workdays for employee e
R	minimum rest periods between two shifts
w	maximum number of consecutive workdays
h_e^{max}	maximum number of working hours for employee e
h_e^{min}	minimum number of working hours for employee e
g_{eq}	1, if employee e has skill q, 0 otherwise
c_{ej}	cost of shift type j assigned to worker e
c_q^{out}	cost of outsourcing skill q for one period
c^{ot}	cost of an overtime period
α	wage differentiation factor
l	number of periods in an hour

extends the RSC formulation to a Multiskilled Workforce Scheduling (MWS) model. Table 1 summarizes the notation used.

Four main decision variables are used. First, the binary variable p_{ejd} assigns employees to shifts on a day:

$$p_{ejd} = \begin{cases} 1, & \text{if employee } e \text{ is assigned to shift } j \text{ on day } d \text{ and} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Second, the skill that an employee applies to work in each period t on day d is selected through binary variable z_{eqtd}:

$$z_{eqtd} = \begin{cases} 1, & \text{if employee } e \text{ uses skill } q \text{ in period } t \text{ of day } d \text{ and} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Third, we use an auxiliary variable y_{ed} to indicate whether an employee works on a particular day:

$$y_{ed} = \begin{cases} 1, & \text{if employee } e \text{ is assigned to work on day } d \text{ and} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Last, the number of exceeding periods over the monthly hour limit h_e^{max} per employee are determined by the variable ot_e.

In case the requirements are not met in a particular period within a day, outsourcing additional resources is allowed and decided upon using integer variable o_{qtd}. This ensures feasibility independent of the problem instance. This variable is also paired with an outsourcing cost c_q^{out}, which could be interpreted as the penalty for unmet requirements.

The staffing costs c_{ej} for a certain shift j depend on the length, time of the day of shift j, and the number of skills of employee e. The number of working periods is obtained from the difference between the last working period (f_j) and the first working period (s_j) of shift j. In case the hourly wage per period depends on an agent's number of qualifications, the cost depends on the number of qualifications per agent multiplied by a differentiation factor α:

$$c_{ej} = \overbrace{(f_j - s_j + 1)}^{\text{working periods}} * \overbrace{\left(1 + \alpha \sum_q g_{eq}\right)}^{\text{skill costs}} \tag{4}$$

In this way, the more skills an agent has, the higher the hourly wage will be. Also, we can set $\alpha = 0$ when all agents are paid the same per hour regardless of the qualifications they have.

The objective function (5) minimizes the overall costs for all assigned shifts, for outsourced resources and for overtime:

$$\text{Minimize } F = \sum_{e=1}^{E} \sum_{j=1}^{J} \sum_{d=1}^{D} c_{ej} p_{ejd} + \sum_{q=1}^{Q} \sum_{d=1}^{D} \sum_{t=1}^{T} c_q^{out} o_{qtd} + \sum_{e=1}^{E} c^{ot} ot_e \tag{5}$$

Subject to the following constraints:

$$\sum_{j=1}^{J} p_{ejd} = y_{ed} \qquad \forall e;\ \forall d \tag{6}$$

$$\sum_{e=1}^{E} z_{eqtd} + o_{qtd} \geq r_{qtd} \qquad \forall q;\ \forall t;\ \forall d \tag{7}$$

$$z_{eqtd} \leq g_{eq} \qquad \forall e;\ \forall q;\ \forall t;\ \forall d \tag{8}$$

$$\sum_{q=1}^{Q} z_{eqtd} \leq \sum_{j=1}^{J} a_{tj} p_{ejd} \qquad \forall e;\ \forall t;\ \forall d \tag{9}$$

$$\sum_{j=1}^{J} \sum_{d=1}^{D} (f_j - s_j + 1) p_{ejd} \leq l \cdot h_e^{max} + ot_e \qquad \forall e \tag{10}$$

$$\sum_{j=1}^{J} \sum_{d=1}^{D} (f_j - s_j + 1) p_{ejd} \geq l \cdot h_e^{min} \qquad \forall e \tag{11}$$

$$\sum_{d=1}^{D} y_{ed} \geq d_e^{min} \qquad \forall e \tag{12}$$

$$\sum_{d=1}^{D} y_{ed} \leq d_e^{max} \qquad \forall e \tag{13}$$

$$\sum_{\bar{d}=d}^{d+w} y_{e\bar{d}} \leq w \qquad \forall e;\ \forall d \leq D - w \tag{14}$$

$$T - \sum_{j=1}^{J} f_j p_{ejd} + \sum_{j=1}^{J} s_j p_{ejd+1} - y_{ed+1} \geq y_{ed+1} R \qquad \forall e;\ \forall d \leq D - 1 \tag{15}$$

$$p_{ejd};\ y_{ed};\ z_{eqtd} \in \{0, 1\} \qquad \forall e;\ \forall j;\ \forall q;\ \forall t;\ \forall d \tag{16}$$

$$ot_e \geq 0 \qquad \forall e \tag{17}$$

Constraints (6) and (12) to (16) are equivalent to those in the formulation proposed by Stolletz (2010), while the objective function (5) and constraints (7) to (10) and (17) extend the formulation to a multiskilled workforce. Constraint (6) ensures that employee e is assigned to at most one shift per day and sets the variable y_{ed}. Equation (7) ensures that agent requirements are satisfied for each skill q, either by assigning agents or by outsourcing. Equation (8) ensures that employees are assigned to tasks for which they are qualified. An agent can be assigned to at most one task if the respective period is a working period, as stated in Equation (9). Equation (10) ensures that employees are assigned to at most h_e^{max} overall working hours, otherwise incurring an overtime of ot_e periods. Equation (11) represents the lower bound for the overall working hours. Constraints (12) and (13) represent the lower and upper bounds for the overall working days within a tour, while Equation (14) allows employees to work a maximum of w consecutive days. Equation (15) ensures that the sum of the off periods after the end of a shift on day d plus those before starting a shift on day $d+1$ is at least R periods long.

4 Rolling Planning Horizon Heuristic

This section describes our proposed solution procedure based on a rolling horizon approach. The main idea of the Rolling Planning Horizon heuristic (RPH) is to solve a tour scheduling problem for a planning horizon of D days by iteratively solving tour scheduling subproblems for shorter planning horizons of r days and freezing the value of the decision variables for the first s days in each iteration. As illustrated in Fig. 1, in iteration $k=1$ the RPH starts with solving the subproblem for the first r days of the planning horizon. Once a solution is obtained, the decision variables for the first s days are fixed to their solution values. These variables are kept fixed in iteration $k=2$, where the next subproblem for $s+r$ days is solved. Once a solution is obtained for this subproblem, the next block of s days is fixed (days $s+1$ to $2s$). For a certain iteration k, the subproblem covers $(k-1)s+r$ days while respecting the already fixed schedule for the previous $(k-1)s$ days. After solving the subproblem, the decision variables for days $(k-1)s+1$ to ks are then fixed to their solution values. If in an iteration $(k-1)s+r>D$, it is ensured that the remainder of the original planning horizon is covered. This process is repeated until the original planning horizon of D days is covered after $K=\lceil\frac{D}{s}\rceil$ iterations.

Figure 1 illustrates an example of the RPH for $[r=7, s=4]$ and $D=30$ days. This figure shows, for each iteration, the subproblem's planning horizon (white boxes), including the already fixed days (dark grey boxes). The pending days (light gray boxes) are not considered in the solution of the subproblem.

To avoid infeasibility, the bounds on the overall working days and overall working hours have to be changed. Specifically, the bounds set in constraints (10),(11), (12), and (13) need to be replaced, because they consider the full length of the planning horizon. The idea is to set different bounds in each iteration $k<K$ that are related to the proportion $\frac{(k-1)s+r}{D}$ of the considered planning horizon:

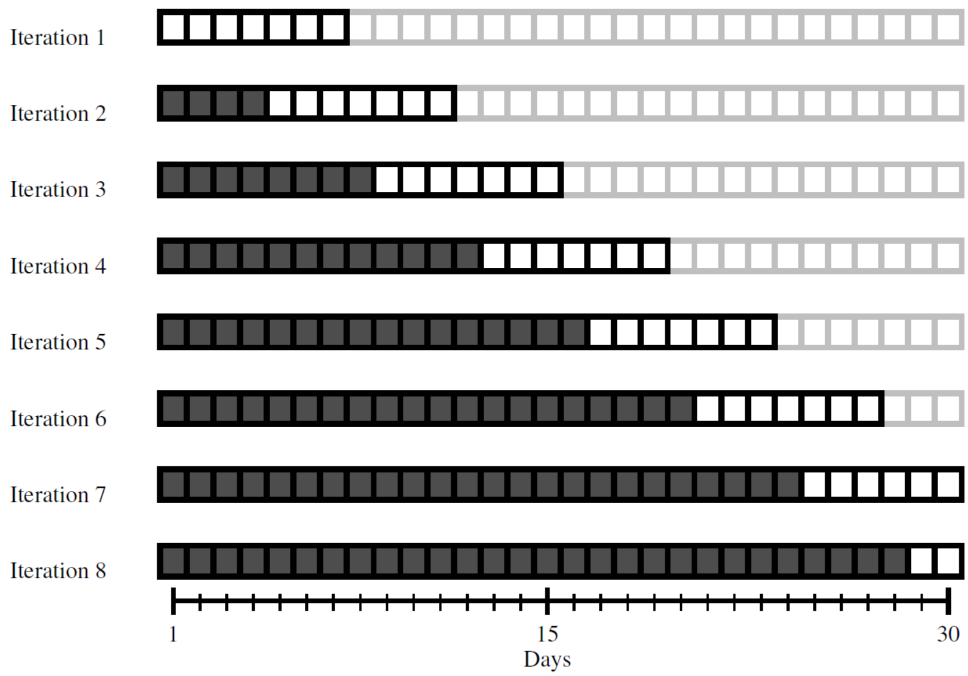
$$\sum_{j=1}^{J}\sum_{d=1}^{(k-1)s+r}(f_j-s_j+1)p_{ejd} \leq \left\lceil\frac{[(k-1)s+r]\cdot l\cdot h_e^{max}}{D}\right\rceil + ot_e \qquad \forall e, \tag{18}$$

$$\sum_{j=1}^{J}\sum_{d=1}^{(k-1)s+r}(f_j-s_j+1)p_{ejd} \geq \left\lceil\frac{[(k-1)s+r]\cdot l\cdot h_e^{min}}{D}\right\rceil \qquad \forall e, \tag{19}$$

$$\sum_{d=1}^{(k-1)s+r} y_{ed} \geq \left\lfloor\frac{[(k-1)s+r]d_e^{min}}{D}\right\rfloor \qquad \forall e, \tag{20}$$

$$\sum_{d=1}^{(k-1)s+r} y_{ed} \leq \left\lceil\frac{[(k-1)s+r]d_e^{max}}{D}\right\rceil \qquad \forall e, \tag{21}$$

Figure 1: RPH example



Constraints (18)-(21) bind the sum of the overall assigned working hours and days up to $(k-1)s+r$ to the ratio of the minimum and maximum hours and days, respectively. In this way, the bounds are updated in every iteration. Nevertheless, these new constraints still constitute hard constraints that strictly bind the length of each tour to a fraction of the working-days limit defined in the problem setting. The last iteration K runs with the original bounds and applies the constraints (18)-(21) in place of (10)-(13), which cover the entire planning horizon. Algorithm 1 shows the pseudocode for the RPH procedure.

5 Numerical Experiments

For our numerical tests, we analyze cases of two German ground-handling agencies. In the first case, we use the demand data from one ground handler to obtain good parameters r and s for the RPH. To focus on this goal, we assume fully qualified agents in Section 5.1. In Section 5.2, the same demand data are used to analyze the impact of different skill distributions on the quality of the schedules. In the second case, we test our proposed heuristic with data from a second ground-handling agency, which contain more information regarding the workforce configuration (Section 5.3). The MWS model and the subproblems of the RPH heuristic were solved using the Gurobi 4.6.1 solver on a 2.5-GHz Intel Core i7 machine with 8 GB of RAM.

Algorithm 1 RPH Pseudocode

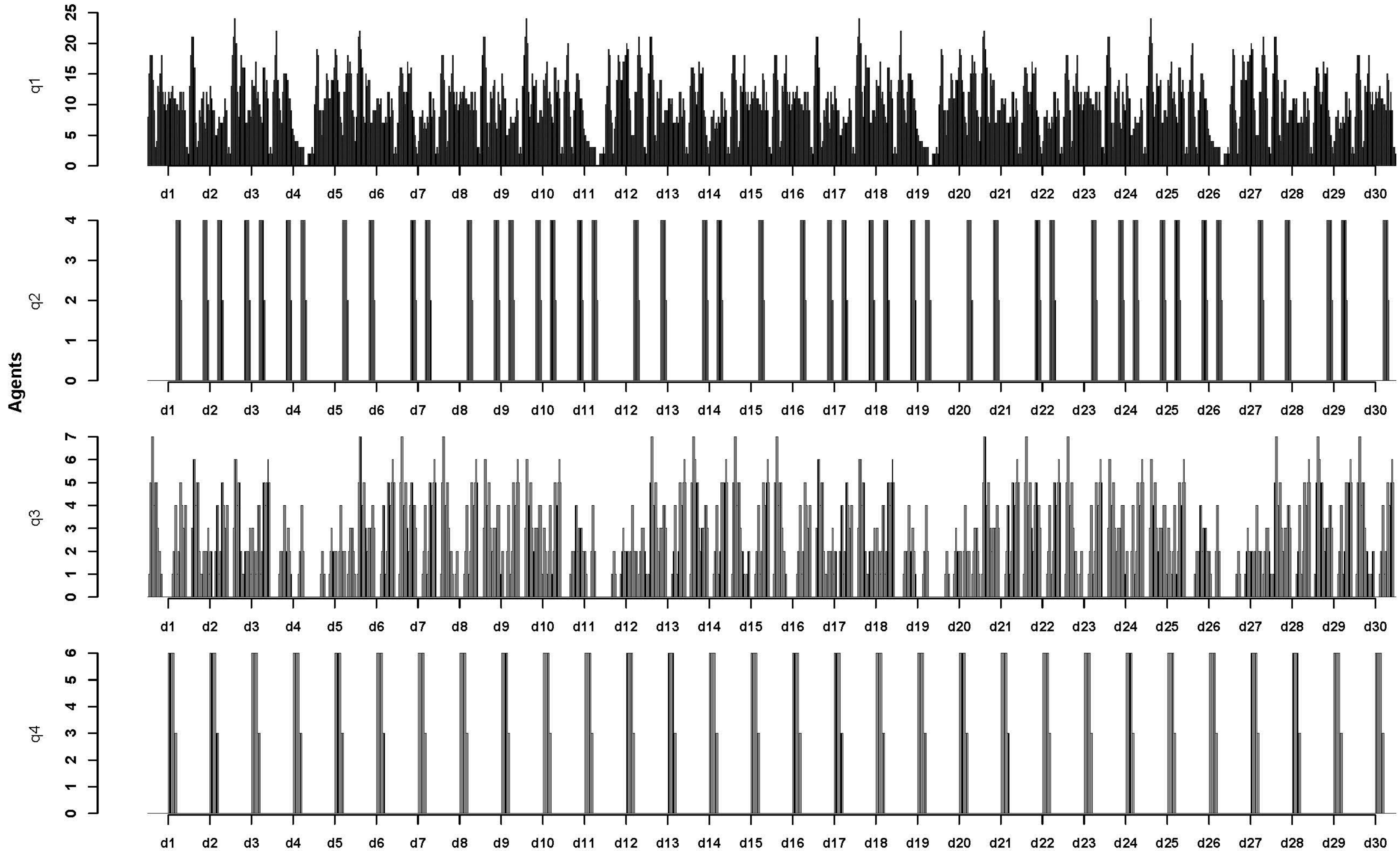
for $k = 1$ **to** $\left\lceil \frac{D}{s} \right\rceil - 1$ **do**
  if $(k-1)s + r < D$ **then**
    Solve MWS subject to (6)-(9), (14)-(17) and (18)-(21) for $d \leq (k-1)s + r$ with fixed decision variables up to $d = (k-1)s$
  else
    Solve MWS subject to (6)-(9), (14)-(17) and (18)-(21) for $d \leq D$ with fixed decision variables up to $d = (k-1)s$
  end if
  for $d = (k-1)s + 1$ **to** ks **do**
    Fix y_{ed}, o_{qdt}, z_{eqtd} and p_{ejd}
  end for
end for
Solve MWS subject to (6)-(17) for $d \leq D$ with fixed decision variables up to $d = (K-1)s$

5.1 Sensitivity of the RPH Parameters

For the first numerical tests, we study the impact of the choice of the parameters r and s on the performance of the RPH. The data from the first ground-handling agency contain the aggregated agent skill requirements for 30-minute intervals during the opening hours ($T = 34$) for different qualifications ($Q = 4$) within a 30-day planning horizon ($D = 30$) (see Fig. 2). The workforce consists of $E = 65$ employees. Because no real information on the skill distribution was given, we consider agents with all skills (*i.e.*, generalists) and no wage differentiation ($\alpha = 0$). The employee contracts establish a maximum of $w = 6$ consecutive working days without a day off and a minimum of $d_e^{min} = 19$ and maximum of $d_e^{max} = 23$ overall working days per employee. Shifts can be from 3 to 10 hours in length, with a minimum of $R = 10$ resting hours between consecutive working days. We assume there are no limits on the maximum or minimum overall working hours h_e^{max} and h_e^{min}. The costs consist of scheduled hours and outsourced hours (with $c_q^{out} = 1 \times 10^7$), and no overtime costs ($c_{ot} = 0$) are considered. This leads to a total of 914,795 decision variables and 337,778 constraints. We test different values of the RPH's parameters $r = 3, 4, 5, \ldots, 15$ and $s = 2, 3, 4$.

Solving the MWS MIP model with this setting to optimality yields a total of 7,941.5 scheduled hours with an overcapacity of 543.5 hours and no outsourcing. This optimal solution was found in 16,742 seconds. Table 2 shows the overall scheduled hours, outsourced hours, the overcapacity (in hours), and the computation time (in seconds) obtained for each configuration of the RPH. The absolute gap to the optimal scheduled hours of the RPH, compared to the optimal solution, is shown in the last column. The absolute gap is only reported for instances without outsourcing (50% of the cases) because the optimal solution does not include any outsourced periods. For the instance considered, there is no combination of r and s that clearly outperforms the others. Nevertheless, among the cases with no outsourcing, three configurations ([$r = 6, s = 3$], [$r = 6, s = 4$], and [$r = 7, s = 4$]) could be solved in less than one hour. They showed an accept-

Figure 2: Monthly agent requirements per qualification



able optimality gap of below 0.29%.

Table 2: Performance tests results for different RPH parameters

r	s	K	**Scheduled (h)**	**Outsourced (h)**	**Overcapacity (h)**	**CPU (s)**	**Opt. hours gap (%)**
3	2	15	7836.0	71.50	509.50	613	-
	3	10	7940.5	10.50	553.00	381	-
4	2	15	7944.5	9.00	555.50	13576	-
	3	10	7916.5	24.00	542.50	2407	-
5	2	15	7936.5	10.50	549.00	12185	-
	3	10	7930.0	14.00	546.00	28689	-
	4	8	7810.5	90.50	503.00	861	-
6	2	15	7962.0	1.00	565.00	97054	-
	3	10	7964.5	0.00	566.50	3243	0.290%
	4	8	7955.5	0.00	557.50	2304	0.176%
7	2	15	7955.0	13.50	570.50	39808	-
	3	10	7962.0	0.50	564.50	10017	-
	4	8	7957.5	0.00	559.50	1729	0.201%
8	2	15	7957.0	0.00	559.00	4302	0.195%
	3	10	7952.0	2.50	556.50	42260	-
	4	8	7941.0	5.00	548.00	38550	-
9	2	15	7960.0	0.00	562.00	13367	0.233%
	3	10	7959.0	0.00	561.00	5121	0.220%
	4	8	7930.0	6.00	538.00	6246	-
10	2	15	7952.5	0.00	554.50	14673	0.201%
	3	10	7959.0	0.00	561.00	17982	0.252%
	4	8	7959.5	0.00	561.50	18448	0.220%
11	2	15	7957.5	0.00	559.50	18740	0.176%
	3	10	7960.5	0.00	562.50	20722	0.120%
	4	8	7954.0	0.00	556.00	21035	0.120%
12	2	15	7957.5	0.00	559.50	18740	0.000%
	3	10	7961.5	0.00	563.50	7682	0.000%
	4	8	7959.0	0.00	561.00	17982	0.000%
13	2	15	7956.0	0.00	558.00	31461	0.000%
	3	10	7944.5	0.00	546.50	33921	0.000%
	4	8	7952.0	0.00	554.00	35577	0.000%
14	2	15	7941.0	0.00	548.00	38550	0.000%
	3	10	7955.0	0.00	570.50	39808	0.000%
	4	8	7952.0	0.00	556.50	42260	0.000%
15	2	15	7955.5	0.00	557.50	44787	0.000%
	3	10	7951.0	0.00	553.00	11687	0.000%
	4	8	7951.0	0.00	553.00	47946	0.000%

To show that these configurations result in good solutions for different problem instances, we scale the workforce and the requirements. For this purpose, we multiply the demand and the workforce size by a scaling factor σ to simulate scenarios with lower demand and a smaller workforce ($\sigma < 1$) or higher demand and a larger workforce ($\sigma > 1$) than the original setting. For several values of the scaling factor ($\sigma = 0.2, 0.4, \ldots, 1.4$) we solve the MIP formulation. In the case of $\sigma = 0.2$ an exact solution could not be found even after 33 hours of computation. We also solve the RPH with $[r = 6, s = 3]$, $[r = 6, s = 4]$, and $[r = 7, s = 4]$ for each scaling factor $\sigma = 0.2, 0.4, ..., 1.4$. In order to control the computation time we set a time limit of 3600 seconds forthe solution of each subproblem in the RPH algorithm. Table 3 shows, for each RPH configuration, σ, and workforce size: the overall requirements (in hours), the overall scheduled hours, the sum of the outsourced hours, the overcapacity (in hours) and the computation time (in seconds). The last column shows the ratio of outsourced hours (as a percentage of the required hours) for each problem instance. Although for most of the instances outsourcing is required, the ratio of outsourced hours does not exceed 1.45% for all cases with $\sigma \leq 1.2$. For the case with $\sigma = 1.4$, all configurations result in a higher ratio of outsourced hours because the time limit to solve the subproblems was reached in several iterations. Yet again, no single configuration clearly outperforms the others, *e.g.*, while $[r = 7, s = 4]$ shows the lowest computation time in five out of seven instances, $[r = 6, s = 4]$ has the lowest outsourcing ratio and/or optimality gap in six out of seven instances. In conclusion, these tests confirm that the proposed heuristic is robust enough to be applied to different problem sizes, although testing different configurations of the RPH parameters for different problem instances is recommended.

Table 3: Performance tests for different problem sizes

	σ	E	Required (h)	Scheduled (h)	Outsourced (h)	Overcapacity (h)	CPU (s)	Outsourced ratio
MIP	0.2	13	1479.6	2099.0	15.5	660.4	33 h	1.05%
	0.4	26	2959.2	3711.0	0.0	751.8	5671.7	0.00%
	0.6	39	4438.8	5324.0	0.0	885.2	8889.5	0.00%
	0.8	52	5918.4	6904.0	0.0	985.6	8001.9	0.00%
	1	65	7398.0	7941.5	0.0	543.5	16741.7	0.00%
	1.2	78	8877.6	10065.0	0.0	1187.4	66875.9	0.00%
	1.4	91	10357.2	12126.0	0.0	1768.8	28236.4	0.00%
RPH [r=6, s=3]	0.2	13	1479.6	2060.0	21.5	601.9	516.3	1.45%
	0.4	26	2959.2	3662.5	12.0	715.3	1733.5	0.41%
	0.6	39	4438.8	5279.5	8.5	849.2	7389.2	0.19%
	0.8	52	5918.4	6879.0	6.0	966.6	7085.7	0.10%
	1	65	7398.0	7964.5	0.0	566.5	3243.0	0.00%
	1.2	78	8877.6	10054.5	5.0	1181.5	3172.0	0.06%
	1.4	91	10357.2	11961.5	775.0	2379.3	16743.7	7.48%
RPH [r=6, s=4]	0.2	13	1479.6	2065.0	15.5	600.9	289.2	1.05%
	0.4	26	2959.2	3690.5	4.0	735.0	890.6	0.14%
	0.6	39	4438.8	5310.0	2.5	873.7	1864.5	0.06%
	0.8	52	5918.4	6905.0	0.5	987.1	2843.1	0.01%
	1	65	7398.0	7955.5	0.0	557.5	2304.4	0.00%
	1.2	78	8877.6	10074.5	0.0	1196.9	3790.1	0.00%
	1.4	91	10357.2	12125.5	998.0	2766.3	12488.7	9.64%
RPH [r=7, s=4]	0.2	13	1479.6	2064.5	17.0	601.9	474.5	1.15%
	0.4	26	2959.2	3685.0	6.0	731.8	396.1	0.20%
	0.6	39	4438.8	5301.5	4.5	867.2	1175.0	0.10%
	0.8	52	5918.4	6900.5	2.0	984.1	3507.5	0.03%
	1	65	7398.0	7957.5	0.0	559.5	1728.9	0.00%
	1.2	78	8877.6	10060.0	3.5	1185.9	2836.4	0.04%
	1.4	91	10357.2	11949.0	983.0	2574.8	9357.5	9.49%

5.2 Multiple Skills Analysis

In the following tests, we address the impact that different skill distributions have on the solutions. For this purpose, we use the same requirement data as in Section 5.1, although with different skill distributions and workforce sizes. We consider a combination of $\beta \cdot E$ generalist agents and $(1-\beta) \cdot E$ agents with a single skill profile (specialists). The number of specialists of a specific skill is calculated by multiplying the proportion each skill is present in the requirements with $(1-\beta) \cdot E$ (rounding up for the two most required skills and rounding down for the two least required skills).

Different workforce sizes $E \in \{65, 75, 85\}$ and skill distributions ($\beta = 0.0, 0.2, ..., 1.0$) are considered, with the original agent requirements held unchanged for all cases. Furthermore, we look at different cost differentiation factors ($\alpha = 0, 1, 2$), such that for every additional skill an employee has, the hourly cost increases by α units. A total of 54 different problem instances were tested. Each instance is solved exactly with the MIP formulation in less than 55.60 h with an average of 8.86 h of computation times. In addition, each instance was also solved with three different RPH configurations ([$r = 6, s = 3$], [$r = 6, s = 4$], and [$r = 7, s = 4$]). In order to control the computation time we set a time limit of 3600 seconds for each iteration of the algorithm.

Figures 3 and 4 show a comparison of the results obtained with the RPH heuristic for these tests. Figure 3 presents a comparison of the computation time (in seconds) as a function of β for each combination of α and E. The computation time for the RPH is less than 11900 seconds in all cases with an average of 2585 seconds. Figure 4 shows the absolute gap (in percentage) of the solutions of the RPH with respect to the optimal scheduled hours. Only the cases without outsourcing are reported. For these cases, the gap remains below 0.45% from the optimum. To summarize, the RPH solves all the considered instances very quickly with an acceptably small optimality gap. Because of the short CPU times, it is possible to evaluate the value of flexibility for different skill-mix scenarios.

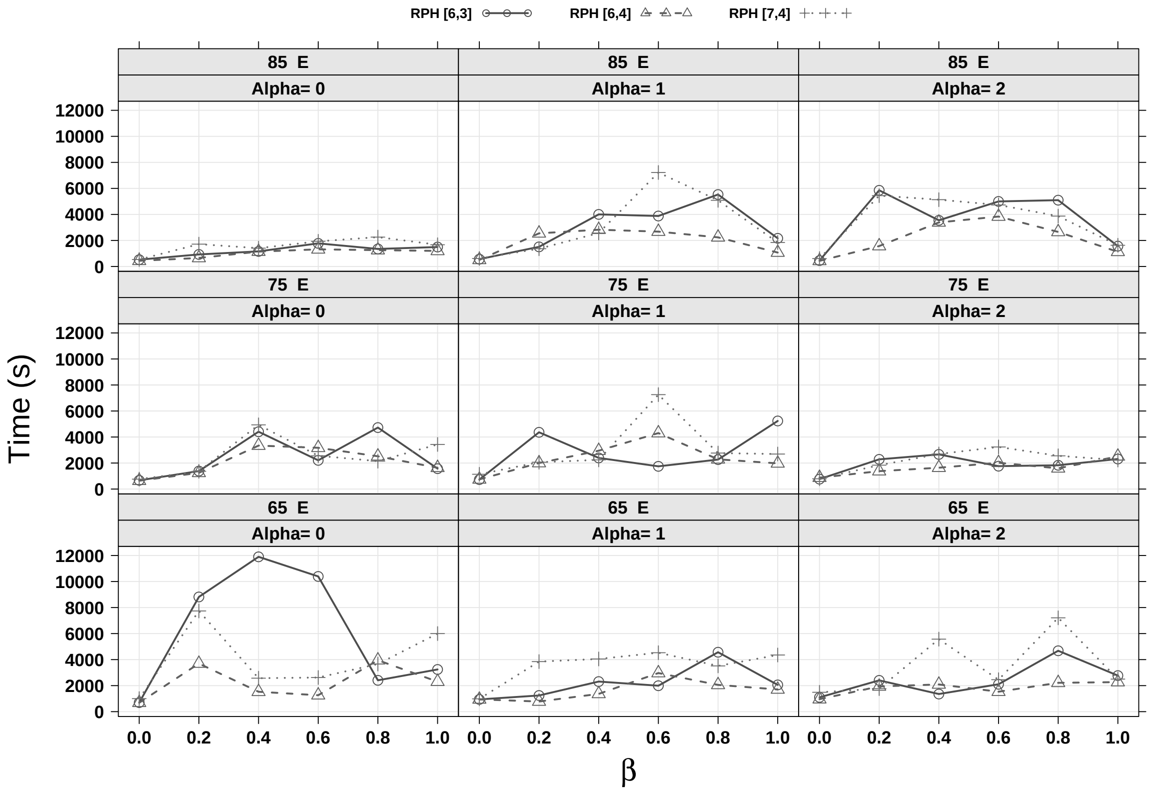
Figure 5 shows a comparison of the overcapacity hours obtained for the solutions for all RPH configurations and all combinations of β, α, and E. Additional flexibility is acquired with larger workforce sizes, and thus, overcapacity hours are reduced in most cases. This is because more agents can be assigned to shorter shifts and fewer idle periods are covered. On the other hand, we observe different behavior in the overcapacity level when varying the values of β and α. If all agents are paid the same ($\alpha = 0$), the full benefits of adding additional fully qualified agents (generalists) have less impact when 20% or more generalists are present in the workforce. For $\alpha = 1$ and $\alpha = 2$, the overcapacity hours are not strictly decreasing with increasing values of β, because more fully qualified agents provide more flexibility, albeit at with a higher cost.

5.3 Real-world Workforce Case

The tests described in the previous sections were conducted using real-world demand data with artificial workforce configurations from a ground-handling agency . In this section, in contrast, we use real-world information on both the demand and the workforce from a second ground-handling agency to test the RPH heuristic under a more realistic setting.

In this new case, the agent requirements r_{qtd} are known and given in 5-minute periods ($T =$

Figure 3: Computation time comparison

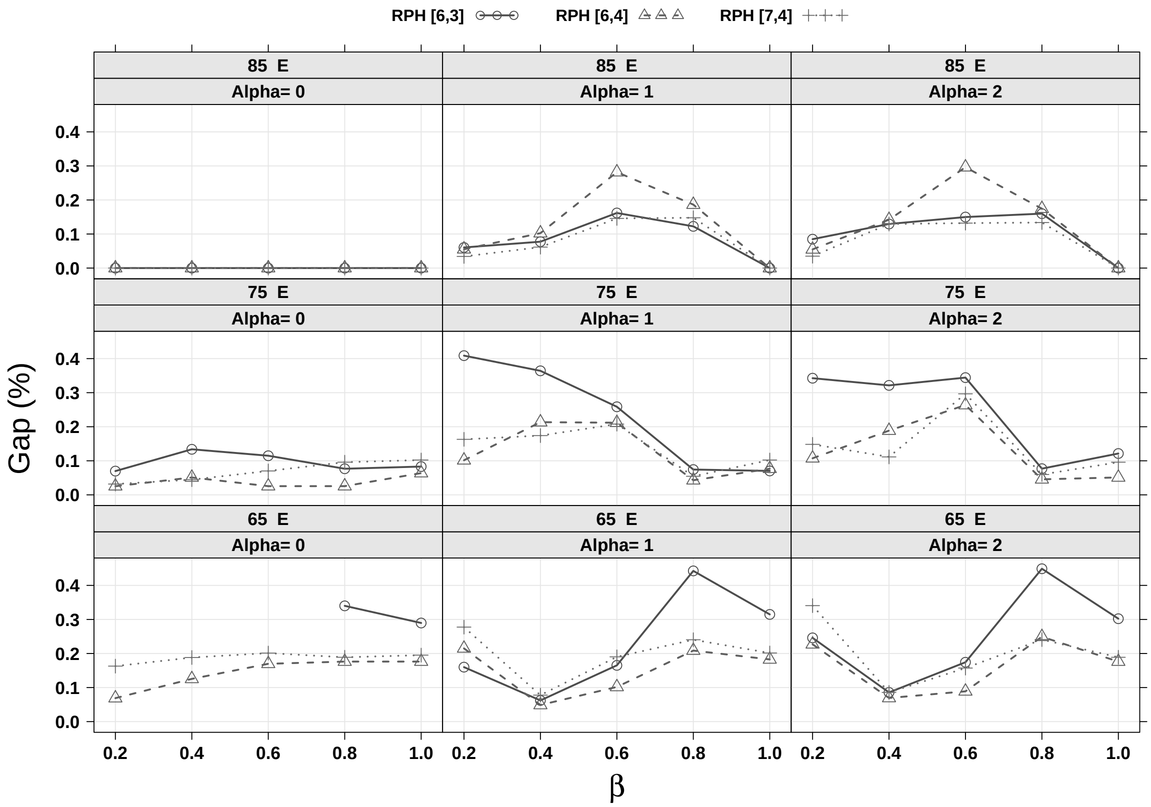


254) for a planning horizon of $D = 30$ days. There are 12 different check-in systems, *i.e.*, $Q = 12$ skills. The requirements for each skill are highly time-dependent and are not correlated to the requirements of other skills.

The check-in personnel consist of $E_{full} = 4$ full-time agents, $E_{part} = 3$ part-time agents and $E_{flexi} = 47$ flexible agents. We assume an hourly wage of 20 euros. Each month, full-time agents are allowed to work at most $h^{max}_{full} = 165$ hours and part-time agents $h^{max}_{part} = 40$ hours. Flexible agents have no monthly h^{max}_{flexi} limit. Working hours that exceed these limits are considered overtime and are paid 25% more than normal working hours. No lower bound on minimum overall working hours h^{min}_{e} is considered. In case the agent requirements cannot be met with the available workforce, additional non-check-in personnel can be assigned to cover them, although this is not desirable because it removes them from their original job positions (*i.e.*, equivalent to outsourcing). We assume non-check in personnel perceive a wage of 40 euros, per hour worked in check-in counters. The different skill combinations and their distribution among the agents are shown in Table 4. The first column shows the number of skills found on each combination and the second column shows the number of agents that have this skill combination. The following columns show which skills are present on each skill combination (marked with an X)

Shifts can have a duration of 3 to 10 hours per day and can start in 15-minute intervals, with a minimum of $R = 10$ rest hours between the end and the start of the shifts of consecutive working days. A maximum of $w = 6$ consecutive working days is allowed without a day off, and days

Figure 4: Optimal scheduled hours gap (%)



off can be non-consecutive. With respect to the overall working days, the corresponding contract rules do not take into account vacation and sick days in a specific month. Therefore, to more faithfully represent the real setting, we fix the assigned working days to the actual number of working days per employee from the data set.

The full MIP model for this instance has 8'516,699 decision variables (from which 8'425,200 are binary) and 5'937,980 constraints. An exact solution could not be obtained by solving the MIP formulation because of out-of-memory errors. We solve the RPH for this problem instance with $[r = 6, s = 3]$, $[r = 6, s = 4]$,and $[r = 7, s = 4]$. In order to control the computation time we set a time limit of 5 hours per iteration of the algorithm. Table 5 shows the overall scheduled hours for both check-in and non-check in personnel, the overtime (in hours), the staffing costs (in euros), the computation time (in hours) and the utilization for each parameter configuration. These solutions, in average, incurred 40.73 overall overtime hours and 7.56 hours during which non-check-in personnel were required.The best solution is found with $[r = 7, s = 4]$ obtained within 5.81 hours. Figure 6 shows a daily comparison of the overall scheduled hours from the RPH solutionS with respect to the overall requirements.

This case study shows that the RPH is able to solve complex problem instances as they appear in real applications.

Figure 5: Overcapacity

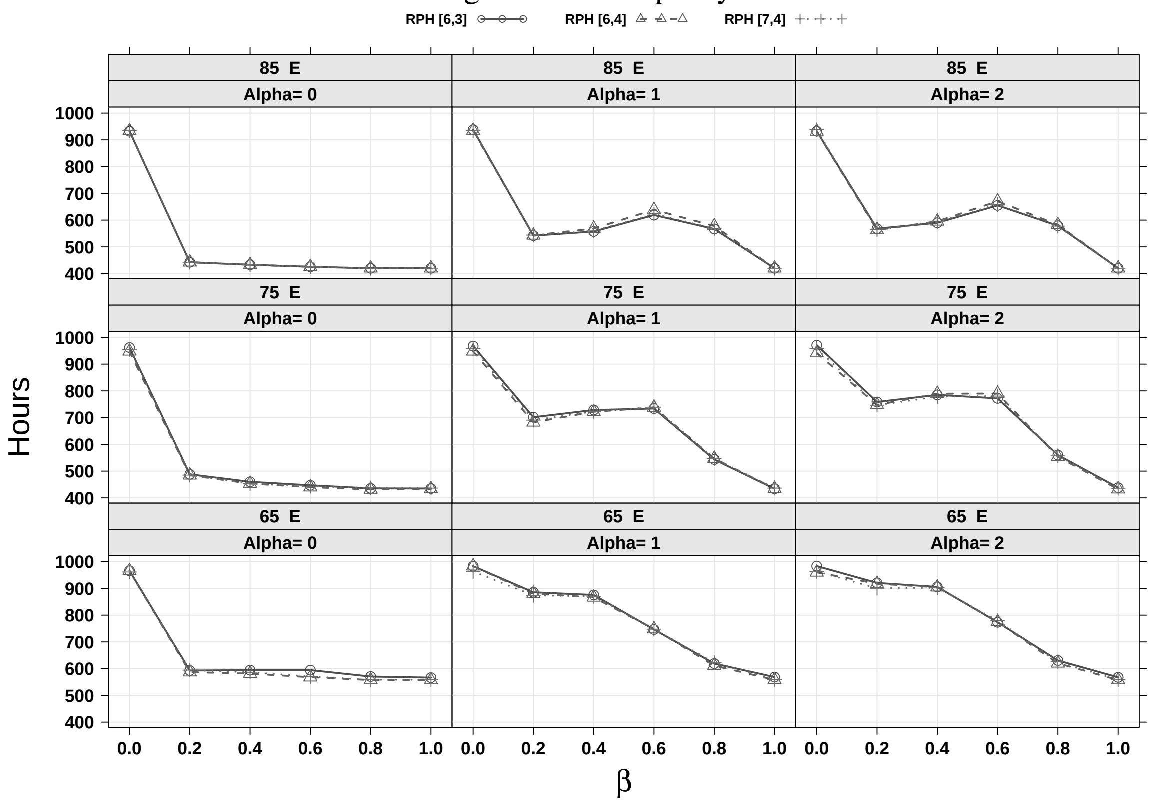


Figure 6: Comparison of overall scheduled and required hours

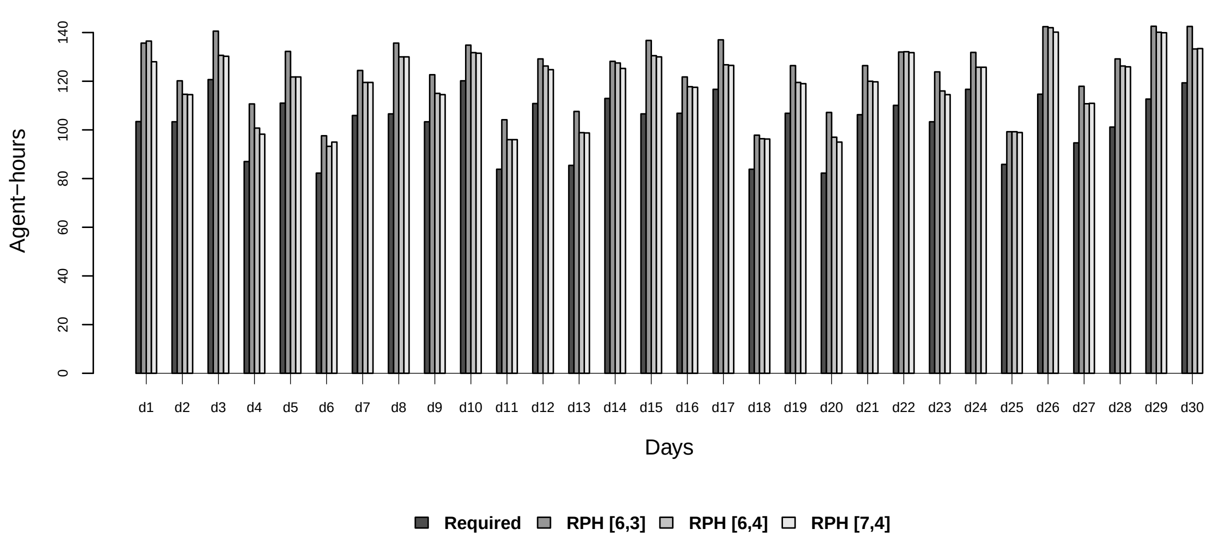


Table 4: Skill distribution of real-world workforce case

Skills	Agents	Skill combination q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12
11	9	X	X	X	X	X	X	X	X	X		X	X
10	6	X	X	X	X	X	X	X			X	X	X
9	5	X	X	X	X	X	X	X				X	X
8	3	X	X	X	X	X		X				X	X
8	3	X	X	X	X		X	X				X	X
8	3	X	X		X	X		X	X			X	X
7	2	X	X		X	X		X	X			X	
7	2	X	X		X	X		X				X	X
6	2	X	X		X			X				X	X
6	2	X	X				X	X				X	X
5	2	X	X		X	X		X					
5	2	X	X		X							X	X
5	2	X	X			X		X				X	
5	2	X	X					X				X	X
4	1	X	X		X			X					
4	1	X	X			X		X					
4	1	X	X					X				X	
3	1	X	X			X							
3	1	X	X					X					
3	1	X	X									X	
3	1	X			X			X					
3	1	X						X				X	
3	1		X					X				X	
2	1	X	X										
2	1	X						X					
2	1		X					X					
2	1		X									X	
1	1	X											

Table 5: Summary of results

	RPH [$r = 6, s = 3$]	[$r = 6, s = 4$]	[$r = 7, s = 4$]	Average
Scheduled (h)				
Check-in	3724.25	3568.25	3552.25	3614.92
Non-check-in	14.17	7.50	1.00	7.56
	3738.42	3575.75	3553.25	3622.47
Overtime	48.53	59.17	14.50	40.73
Costs (euros)				
Check-in	74485.01	71365.00	71045.00	72298.34
Non-check-in	566.64	300.00	40.00	302.21
Overtime	1213.25	1479.18	362.50	1018.31
	76264.9	73144.18	71447.5	73618.86
CPU (h)	7.35	10.58	5.81	7.91
Utilization	83.04%	86.82%	87.37%	86%

6 Conclusions and Further Research

This paper addresses the tour scheduling problem for a multiskilled workforce at check-in counters at airports. The main contributions to the literature in this area are the formulation of a tour scheduling model for agents with multiple non-hierarchical skills and flexible employee contracts, and the development of a Rolling Planning Horizon heuristic that can solve this problem to near-optimality within an acceptable amount of time.

The results of a numerical study conducted using real-world data from a ground-handling agency are presented. These results show that the proposed heuristic provides good solutions in an acceptable amount of time for different problem sizes and skill distributions. However, a proper selection of the parameters for the RPH is crucial to the performance of the heuristic, and additional testing is recommended for different problem settings. Furthermore, the results show that additional flexibility can be gained by increasing the proportion of generalist agents in the workforce. On the other hand, this additional flexibility can be acquired at a higher cost if salaries depend on the number of qualifications of the agents.

The results of additional tests conducted using data from a second ground-handling agency show that with our proposed heuristic, it is possible to solve realistic problems that are otherwise not solvable with an MIP formulation, and to obtain good-quality solutions.

Further research needs to be conducted to extend the proposed model to consider agent preferences and fairness measures in the planning process. Another research direction is to implement subsequent phases of the workforce planning process (*e.g.*, task assignment and replanning) with the present model to serve as an integrative robust planning tool for ground-handling agencies. With relation to the proposed heuristic, increasing the complexity of the problem increases the computation time for the subproblems of the algorithm. Therefore, further research is needed to apply heuristics to speed up the solution of these subproblems.

References

Airport Research Center (2009). *Study on the Impact of Directive 96/67/EC on Ground Handling Services 1996-2007*. Aachen.

Alfares, H. K. (2004). Survey, Categorization, and Comparison of Recent Tour Scheduling Literature. *Annals of Operations Research 127*(1-4), 145–175.

Avramidis, A. N., W. Chan, M. Gendreau, P. L Ecuyer, and O. Pisacane (2010, February). Optimizing daily agent scheduling in a multiskill call center. *European Journal of Operational Research 200*(3), 822–832.

Baker, K. (1977). An experimental study of the effectiveness of rolling schedules in production planning. *Decision Sciences 8*(1), 19–27.

Baker, K. and D. Peterson (1979). An analytic framework for evaluating rolling schedules. *Management Science 25*(4), 341–351.

Bard, J. F. (2004). Staff scheduling in high volume service facilities with downgrading. *IIE Transactions 36*(10), 985–997.

Bard, J. F. and H. W. Purnomo (2005). Hospital-wide reactive scheduling of nurses with preference considerations. *IIE Transactions 37*(7), 589–608.

Bartholdi, J. J. I. (1981). A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. *Operations Research 29*(3), 501–510.

Billionnet, A. (1999). Integer programming to schedule a hierarchical workforce with variable demands. *European Journal Of Operational Research 114*, 105–114.

Brunner, J. O. and R. Stolletz (2014). Stabilized branch and price with dynamic parameter updating for discontinuous tour scheduling. *Computers & Operations Research 44*, 137–145.

Cai, X. and K. Li (2000). A genetic algorithm for scheduling staff of mixed skills under multi-criteria. *European Journal of Operational Research 125*(2), 359–369.

Dantzig, G. (1954). A Comment on Edie's "Traffic Delays at Toll Booths". *Journal of the Operations Research Society of America 2*(3), 339–341.

Day, P. R. and D. M. Ryan (1997). Flight attendant rostering for short-haul airline operations. *Operations Research 45*(5), 649–661.

Detienne, B., L. Péridy, É. Pinson, and D. Rivreau (2009). Cut generation for an employee timetabling problem. *European Journal of Operational Research 197*(3), 1178–1184.

Eitzen, G., D. Panton, and G. Mills (2004). Multi-Skilled Workforce Optimisation. *Annals of Operations Research 127*(1-4), 359–372.

Ernst, A. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research 153*(1), 3–27.

Ernst, A., H. Jiang, and M. Krishnamoorthy (2004). An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research 127*(1-4), 21–144.

Gronalt, M. and R. Hartl (2003). Workforce planning and allocation for mid-volume truck manufacturing: a case study. *International Journal of Production Research 41*(3), 449–463.

Helber, S. and K. Henken (2010). Profit-oriented shift scheduling of inbound contact centers with skills-based routing, impatient customers, and retrials. *Or Spectrum 32*(1), 109–134.

Hojati, M. and A. S. Patil (2011). An integer linear programming-based heuristic for scheduling heterogeneous, part-time service employees. *European Journal of Operational Research 209*(1), 37–50.

Jacobs, L. W. and S. E. Bechtold (1993). Microcomputer-based Workforce scheduling. *International Journal of Service 4*(1), 36–48.

Loucks, J. S. and F. R. Jacobs (1991). Tour Scheduling and Task Assignment of a Heterogeneous Work Force: A Heuristic Approach. *Decision Sciences 22*(4), 719–738.

Love, Jr., R. and J. Hoey (1990). Management science improves fast-food operations. *Interfaces 20*(2), 21–29.

Modigliani, F. (1955). Production planning over time and the nature of the expectation and planning horizon. *Econometrica, Journal of the Econometric Society 23*(1), 46–66.

Rong, A. and M. Grunow (2009). Shift designs for freight handling personnel at air cargo terminals. *Transportation Research Part E: Logistics and Transportation Review 45*(5), 725–739.

S&HE International Air Transport Consultancy (2002). *Study on the quality and efficiency of ground handling services at EU airports as a result of the implementation of Council Directive 9667/EC*. London.

Steer Davies Gleave (2010). *Possible revision of Directive 96/67/EC on access to the groundhandling market at Community airports*. London.

Stolletz, R. (2010). Operational workforce planning for check-in counters at airports. *Transportation Research Part E: Logistics and Transportation Review 46*(3), 414–425.

Van den Bergh, J., J. Beliën, P. D. Bruecker, E. Demeulemeester, and L. D. Boeck (2013). Personnel scheduling: A literature review. *European Journal of Operational Research 226*(3), 367 – 385.