



Q ] ! [ çã \* Á [ ! \ { ! & ^ Á & @ ã ~ | ã \* Á ~  
æ ã & ! æ ã Á ^ Á æ ã ç } æ } & ^ Á æ ã ù æ ã ^ } æ ã ^ & @ æ •  
R ! [ ^ } Á ^ | ã } Ë Ó ! ^ & @ Á Õ æ ã [ ^ } Á æ ã ã Ò ã Á Õ ^ { ^ ~ | ^ { ^ ^ • ç !

DEPARTMENT OF DECISION SCIENCES AND INFORMATION MANAGEMENT (KBI)

# Improving Workforce Scheduling of Aircraft Line Maintenance at Sabena Technics

Jeroen Beliën<sup>†\*</sup>, Brecht Cardoen<sup>‡\*</sup>, Erik Demeulemeester<sup>§</sup>

**Abstract.** This paper presents our application of a visualization tool and optimization model based on mixed-integer linear programming to solve a workforce staffing and scheduling problem at Sabena Technics, a major aircraft maintenance company in Belgium. We used the software to generate many alternative, cost-efficient schedules and to analyze multiple scenarios. In several management meetings, stakeholders evaluated the schedules and raised concerns. We subsequently changed the model to successfully address their concerns. The model has resulted in considerable savings and a more efficient use of human resources.

**Keywords:** workforce staffing; scheduling; optimization; visualization; aircraft maintenance.

---

## Introduction

Labor is one of the most significant costs for companies in highly developed countries. Although manufacturers can move their production units en masse to low-wage countries, service companies are often constrained to their current locations. In developed countries, service personnel are usually paid high wages, making it expensive to let them be idle. Therefore, modern staffing and rostering software can be valuable for these service companies.

This paper presents our application of an enumeration approach based on mixed-integer linear programming (MILP) to solve a workforce staffing and scheduling problem at Sabena Technics. Sabena Technics offers a complete range of aircraft maintenance services across six sites in Europe. These services include (1) heavy maintenance including activities such as

---

<sup>†</sup>Hogeschool-Universiteit Brussel, Faculty of Economics and Management, Center for Informatics, Modelling and Simulation, Stormstraat 2, B-1000 Brussels, jeroen.belien@hubrussel.be Belgium

<sup>\*</sup>Affiliated researcher Katholieke Universiteit Leuven, Research center for Operations Management

<sup>‡</sup>Vlerick Leuven Gent Management School, Operations and Technology Management Center, Reep 1, B-9000 Gent, Belgium, brecht.cardoen@vlerick.be

<sup>§</sup>Katholieke Universiteit Leuven, Faculty of Business and Economics, Department of Decision Sciences and Information Management, Research Center for Operations Management, Naamsestraat 69, B-3000 Leuven, Belgium, erik.demeulemeester@econ.kuleuven.be

C-checks (whole aircraft is inspected, takes 1-2 weeks, once a year), D-checks (most demanding and comprehensive check, takes 3 weeks - 2 months, once every four years), stripping and painting, and cabin refurbishment, (2) light maintenance including A-checks (check certain service items requiring special tools, once every 65 flight hours), cabin maintenance, and engine changes, and (3) line maintenance including preflight inspections, transit checks, daily checks (visual inspection, fluid levels, general security and cleanliness of the flight deck, emergency equipment), weekly checks, and on call assistance (Sabena Technics 2010). This paper addresses the line maintenance at Brussels Airport. Line maintenance, which depends on the number of flight hours and take-off and landing cycles, is preventive (as opposed to unplanned or emergency maintenance). At Brussels Airport, Sabena Technics oversees the line maintenance for over 300 flights, arriving and departing according to a fixed weekly schedule. Aircraft maintenance must be done by highly qualified employees and cannot simply be moved to an airport in a low-wage country. The work is organized in teams and the rosters follow a cyclical pattern.

Staffing and rostering decisions typically result in complex combinatorial optimization problems that reinforce the need for advances in the operations management domain (Le-saint et al. 2000); for example, the application of a dynamic scheduler based on a combination of heuristic search and constraint-based reasoning for assigning field engineers saves British Telecommunications plc USD 150 million a year in operations costs. Burke et al. (2004) discuss a study in which they review new methodologies and algorithms to solve the nurse rostering problem. Aviation is also highly labor intensive and suitable for operational optimization (Barnhart and Cohn 2004). Even small improvements in staffing or scheduling can result in considerable savings. For example, Yu et al. (2003) discuss how Continental Airlines can quickly recover from unexpected events that could disrupt schedules and crew assignments. This approach results in yearly savings of about USD 40 million. Butchers et al. (2001), Desrosiers et al. (2000), and Alfares (1999) describe other successful staffing and scheduling examples in the aviation industry. We refer to Ernst et al. (2004) for a general overview of staffing and rostering in several domains.

Two important characteristics distinguish our work from previous applications described in the literature. First, the demand, which we express in man-hours, is not a fixed value. Each aircraft maintenance activity must occur within a given time window bounded by the

scheduled time of arrival (STA) and the scheduled time of departure (STD); however, Sabena Technics has some flexibility with respect to the exact timing of the maintenance and the number of people working on a particular aircraft. Hence, decisions on the timing of the maintenance are an additional element in the model. Workforce capacity is also a part of the scheduling decision process. Second, although the workforce number, which is determined by a higher-level staffing decision, is often assumed to be a fixed input for the scheduling process, our solution integrates the scheduling decision with the staffing decision. Upon building the new rosters for the line maintenance department, employees can be transferred to (or from) other departments of Sabena Technics; therefore, staffing is an integral part of the decision process.

Aircraft maintenance is subject to uncertainty. Although the model we present in this paper is deterministic in nature, we generate robust schedules by using the concept of buffers, and protect the schedules from unexpected events such as flight delays and illness. We distinguish between three major sources of uncertainty: (1) flight arrivals, (2) workforce presence or absence (e.g., for illness) and (3) the time required to perform line maintenance of a particular flight. We handle the first source by adding time buffer constraints to our deterministic model; the second and third sources require the introduction of capacity buffer constraints (see the *Results* section). Alfares (2007) presents a simulation-based alternative approach to address uncertainty.

We refer to our problem as the line maintenance workforce scheduling problem. In the remainder of this paper, we present a detailed description of this problem and our solution approach. In summary, we apply an enumeration approach in which we solve a mixed-integer linear problem for every promising combination of team size and weeks in the roster cycle. A promising combination is a combination for which the total number of workers is not too large (thus, not too costly) or not too small (to enable finding a feasible solution), and the number of weeks is likely to allow us to satisfy the weekend constraints, which we describe in the *Constraints* subsection. We present the insights gained from visualization and describe how the new rosters outperform the old ones with respect to both cost and robustness. We also discuss several suggestions of the management team to tailor the eventual workforce scheduling and staffing decisions during the optimization process.

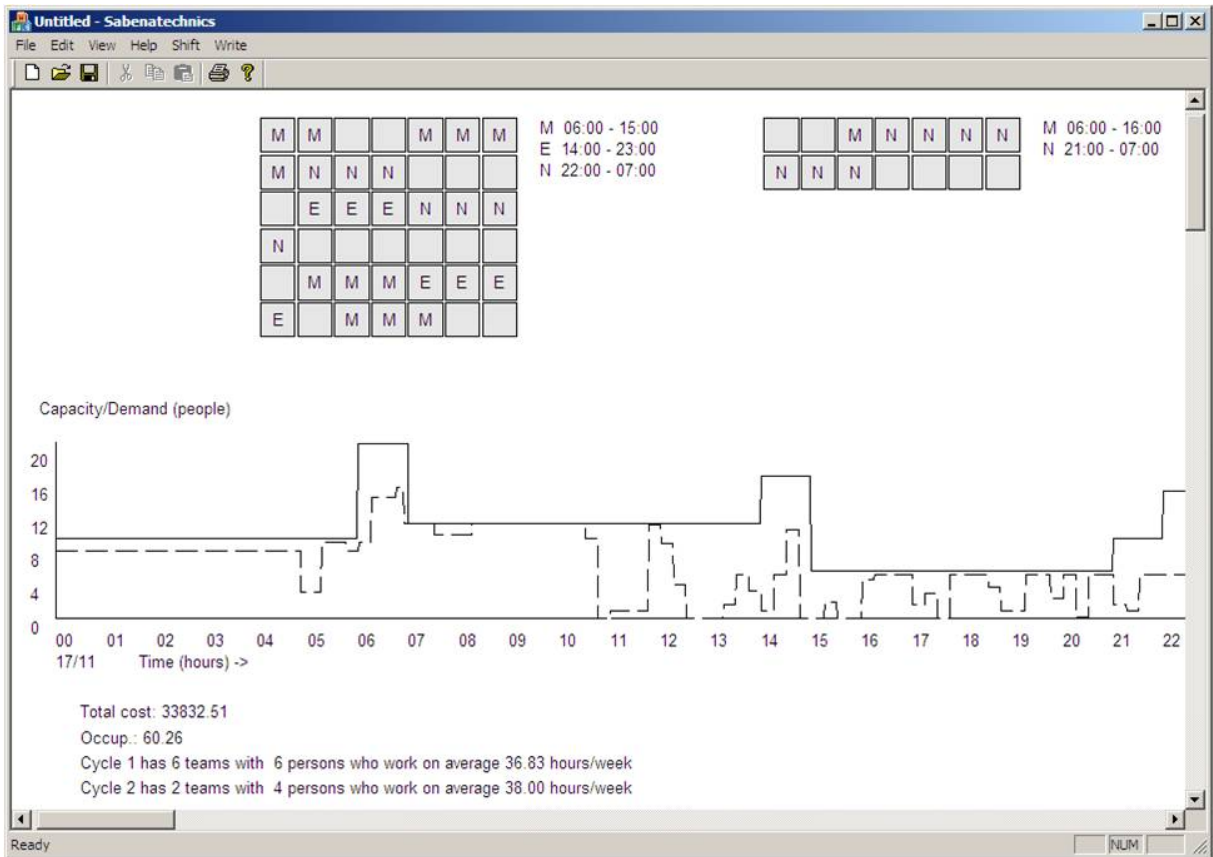
# Problem Description

## Objectives

Sabena Technics distinguishes between a summer season (focused on charter flights) and a winter season (focused on business flights). This paper describes the process of building new rosters for line maintenance for the 2008 winter season. Within a season, customer flights follow a cyclical weekly pattern (i.e., each week, the same set of flights is maintained). To match the workforce supply to this cyclical weekly demand for service, the rosters are organized in weekly cycles. The software generates the new rosters and is used in the negotiation process for getting these new rosters accepted (see Figure 1).

The line maintenance workforce schedule is organized into cycles. The workforce schedule in Figure 1 contains two cycles, which are represented by the two tables at the top of the figure. The first cycle (shown by table on the left side) consists of six weeks; the second cycle (shown by the table on the right side) consists of two weeks. In each table, the rows represent weeks and the columns represent the days of the week. The first column represents Monday, the second represents Tuesday, and the last represents Sunday. The number of teams that works in a particular cycle equals the number of weeks (rows) in that cycle. Hence, six teams work in the first cycle and two teams work in the second cycle. These teams work a cyclical pattern of shifts in which the cycle time equals the number of weeks in the cycle. Sabena Technics distinguishes between four shift types: morning (M), day (D), evening (E), and night (N). The starting and ending hours of these shifts are indicated at the right of each cycle. Note that a cycle does not necessarily include every shift type; the first cycle does not include a D shift and the second cycle does not include D and E shifts. Note also that the starting and ending hours of shift types can differ among cycles. For example, the M shift ends at 15:00 in the first cycle and at 16:00 in the second cycle. The N shift starts at 22:00 in the first cycle and at 21:00 in the second. The cyclical pattern is as follows. In the first cycle, the first team starts working the shifts in the first row. Thus, this team works M shifts on Monday and Tuesday, has two days off, and works M shifts on Friday, Saturday, and Sunday. Simultaneously, the second team works the shifts of the second row, the third team the third row, the fourth team the fourth row, the fifth team the fifth row, and the sixth team the sixth row. After one week, all teams shift one row down and the team that worked the last row (team 6) moves up to the first row. Using this scheduling approach, the

Figure 1: Example of a roster containing two cycles, shown above the workload and capacity graphs.



shifts in the table are worked consecutively—from top to bottom and, within one row, from left to right. After six weeks, each team again works the same shifts it worked six weeks earlier.

The graph below the cycles in Figure 1 depicts the resulting capacity and demand over the week. The horizontal axis represents time and contains all hours of the week (only the first 22 hours of Monday are shown). The vertical axis represents the capacity and demand in terms of number of workers. The solid line indicates the capacity; the dashed line indicates the demand for maintenance. To calculate the capacity at each time instance (the solid line), we sum the numbers of workers in all teams scheduled at that time. The rosters and shift definitions stated at the top of Figure 1 provide the timing information; the team sizes are stated below the graph. All six teams of the first cycle have six people. The two teams in the second cycle have four people. Note that because the rosters are cyclical, the capacity also follows a cyclical pattern, which is repeated each week. The average number of hours worked per week is shown below the graph. Note that this number is the same for all workers in a cycle because they work the same sequence of shifts.

Unlike many other workforce scheduling applications, the demand for services is not fixed for each period, but can be shifted to and (or) distributed over different periods. Although the flights that must be maintained are given and the corresponding workload is stipulated in service level agreements, Sabena Technics has some flexibility in determining the timing of the maintenance of each flight. The service level agreements only stipulate the time window within which particular aircraft must be maintained. These time windows are bounded by the STA and the STD of the client flights at Brussels Airport. For each client flight, Sabena Technics has an demand estimate in man-hours. Hence, the demand input for our optimization problem is a list of flights, the STA and STD, and the estimated demand for each flight (see Table 1).

In terms of modeling, the values of our coverage constraints on the right side (i.e., the constraints that ensure that the scheduled capacity exceeds the demand for services in each period) are not fixed. Instead, they consist of a sum of decision variables. These variables are defined for each flight whose time interval between STA and STD includes the period of the coverage constraint. These maintenance timing decision variables represent the decision of assigning (part of) the flight’s workload to that specific period. The flight maintenance

Table 1: Demand and time windows of flights

Flight	Company	STA	STD	Demand (man-hours)
1	SN	Monday 22:05	Tuesday 07:15	4
...	...	...	...	...
111	OA	Thursday 07:30	Thursday 10:40	6
...	...	...	...	...
310	LY	Sunday 05:30	Sunday 08:45	4.25

can be allocated to one or more periods. Thus, the company can manipulate the demand (see the dashed line in Figure 1) by selecting an appropriate number of workers and time intervals (between STA and STD) for maintaining each flight. In doing this, we must ensure that the solid line (capacity), which is determined by the generated rosters, is always at least equal to the dashed line (demand).

In addition to providing a better match between capacity and demand, Sabena Technics' main objective is to minimize costs. The workforce cost is determined by the number of workers, the lengths of the shifts, the shift types, and the timing of the shifts. Day shifts are the least expensive. A shift premium must be paid for the other shift types. The lowest premium is paid for a morning shift, followed by an evening and a night shift, respectively. Weekend shifts require a premium in addition to the shift-type premium.

To summarize, the problem of building new workforce schedules at Sabena Technics is broader than a pure scheduling problem. In addition to determining the traditional scheduling aspects (i.e., which and how many shifts on each day), we also must determine the shift definitions (starting hours, finishing hours, breaks), the organization of the rosters (number of cycles, number of teams [weeks] per cycle), and the number of workers per team. The objectives are to minimize the labor costs and satisfy a number of constraints, which we discuss in *Constraints* below.



## Constraints

A feasible roster satisfies the following restrictions. First, the roster must satisfy all collective agreement requirements. These restrictions include the definitions of feasible shift types (e.g., earliest starting hour, maximum duration, minimum break duration per shift), a limit on the average number of work hours per week, a limit on the number of weekends worked, and shift-succession constraints ensuring a minimal rest period of 12 hours between two successive shifts. For a detailed overview of these constraints, see Beliën et al. (2010). The weekend constraint, which states that at least half of the weekends must be completely free of any shifts, is one of the most restrictive conditions in this application. Weekend constraints must be very restrictive because demand is typically highest on weekends, as we discuss in the *Visualization* subsection. In this paper, we refer to the restrictiveness of the weekend constraints as the "weekend bottleneck."

Second, our coverage constraints ensure that at any time the scheduled capacity suffices to cover the clients' demand for between-flights maintenance service. Hence, the coverage constraints depend on client flight schedules.

Third, to simplify the organization of the work for all parties involved, Sabena Technics prefers to use a limited number of shift definitions. It was agreed that all shifts of a given shift type (M, D, E, or N) must have the same starting and ending hours within a cycle. Consequently, because each worker is assigned to a single cycle, no worker will be confronted with different shift definitions. This constraint also simplifies the job of the operational scheduler, who assigns specific flights to (teams of) workers. To avoid expensive overtime hours, the operational scheduler must be aware of the starting and ending hours of the teams working each day. The larger the number of distinct shift definitions, the more complex assigning maintenance jobs to the teams is. The constraint that allows at most one shift definition per shift type per cycle ensures that at any time the maximal number of different shift types equals the number of cycles. Hence, when assigning a job to a team of employees of the same cycle working a particular shift, none will quit earlier because of a different shift ending time.

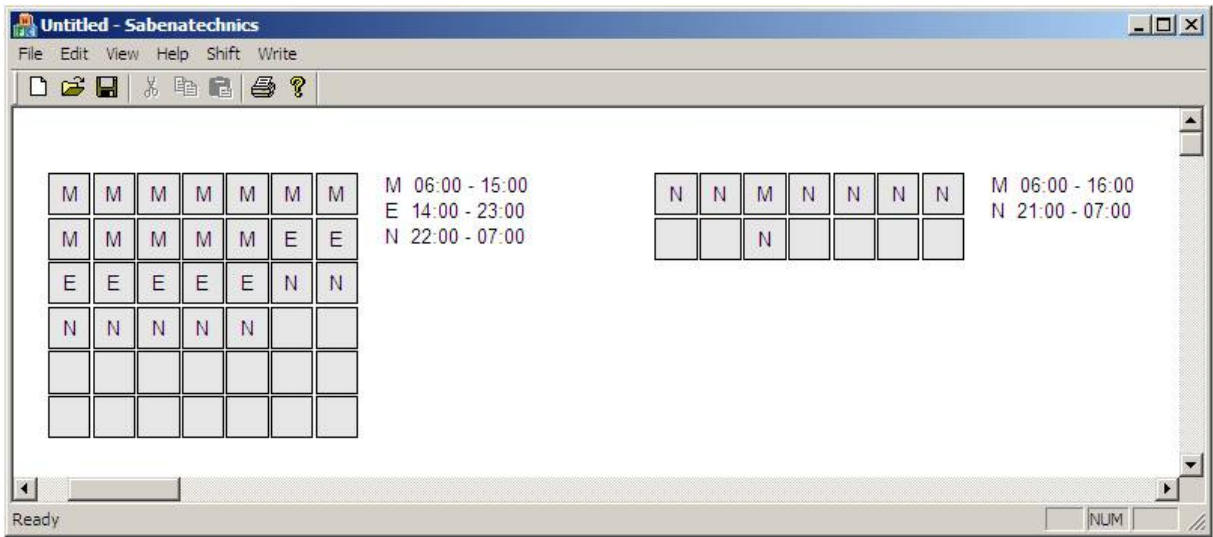
In addition to the constraints mentioned above, soft constraints also exist. Although violating soft constraints is not strictly forbidden, they should be satisfied as often as possible.

All soft constraints relate to the "block" concept, which is defined as a sequence of shifts without a day off. First, the shifts within one block should have the same shift type (e.g, all morning shifts or all evening shifts). Second, the number of shifts in a block may not be too short or too long. Workers prefer to work a minimum of five and a maximum of eight shifts in a row. The schedule in Figure 1 exhibits some examples of soft constraint violations; for example, in cycle 1, week 3 switches between E and N shifts and week 6 includes a block of only three shifts.

For workforce scheduling problems of realistic dimensions, metaheuristic approaches often outperform exact approaches (e.g, MILP). Burke et al. (2004) reach this conclusion after reviewing the literature on nurse rostering problems, which are comparable in size and mathematical structure to many other employee scheduling problems. Although exact approaches succeed in finding the optimal solution for problems with small problem dimensions and a limited set of constraints, these approaches may not generate a feasible solution for problems of realistic dimensions and (or) a realistic set of constraints. The main advantage of applying an MILP approach over a metaheuristic approach is that the lower bounds provided by the MILP optimizer allow us to measure the quality of each solution. We decided to build the Sabena Technics rosters using a two-phase approach. An MILP approach is viable for the problem solved in the first phase. The hard shift-succession constraints and the soft constraints on the block lengths mentioned above dramatically increase the problem complexity (and the required solution times), because these constraints must be specified for each possible combination of two succeeding shifts. Satisfaction of these constraints can be postponed to a second (manual) phase. Hence, in the first (automated) phase, we ignore the hard shift-succession constraints and the soft constraints. Starting from the solution(s) found in this first phase, the (human) scheduler constructs rosters that satisfy the ignored hard (soft) constraints as much as possible. These ignored constraints are (1) the hard shift-succession restrictions, (2) the soft constraints on the block lengths (between five and eight shifts), and (3) those regarding one shift type per block. Figure 2 visualizes the output of our model, after which the cycle in Figure 1 could be constructed, by manually distributing the shifts over the different weeks for each day.

Thus, applying our model results in a number of cost-efficient schedules that satisfy the soft constraints and all but the hard shift-succession constraints. In a first phase, the

Figure 2: Example of the output of applying the MILP-based procedure



management team selects a number of these schedules, typically about 15, based on their cost and visualization, as Figure 2 shows. The (human) scheduler then tries to finish the selected rosters, incorporating the hard shift-succession constraints and the soft constraints described above. Although this second phase could also be automated, we have not included this automation in our model; hence, we must do the second phase manually by distributing the shifts over the weeks in each cycle using the graphical user interface (GUI). In the *Algorithmic Approach* section, we describe the enumerative MILP approach.

## Algorithmic Approach

We solved the Phase 1 problem using an enumerative MILP approach. The algorithm explores a restricted set of combinations of numbers of cycles, staff sizes, and team sizes for each cycle. This set of combinations excludes those for which the total staff size (number of weeks multiplied by the team size in each cycle) is too large (i.e., too costly) or too small (i.e., infeasible). The target value for the total staff size is updated during the search by recording the total staff size of solutions found earlier. Each combination corresponds to an MILP in which the numbers of rosters, staff sizes, and number of weeks in each roster are fixed. The appendix shows the MILP model. By providing the value of the best solution found as an upper bound in each MILP, many of the MILPs can be safely ended when their lower bound becomes higher than the provided upper bound. More details on the MILP

model and the enumeration procedure can be found in Beliën et al. (2010).

To analyze the scenarios discussed in the *Results* section, we added one element to accelerate the procedure. To save computation time and because of the "weekend bottleneck," the algorithm only explores those combinations for which the product of the number of weeks and the weekend fraction is an integer within each cycle. For example, consider a problem with a weekend fraction equal to 50 percent and a schedule that contains a cycle of three weeks; therefore, only one weekend in three is available to assign shifts (because  $2/3$  exceeds 50 percent). In this case, a schedule in which each cycle contains an even number of weeks is much more likely to result in a lower-cost solution.

Finally, in practice, the "optimal roster" does not exist. Therefore, we allowed for a certain optimality gap in the enumeration search. That is, the algorithm discards only those combinations for which the lower bound exceeds the objective value of the best solution found by a specified gap (e.g., 5 percent). As a result, the algorithm returned several feasible rosters, all having an objective value close to the best solution found.

## Results

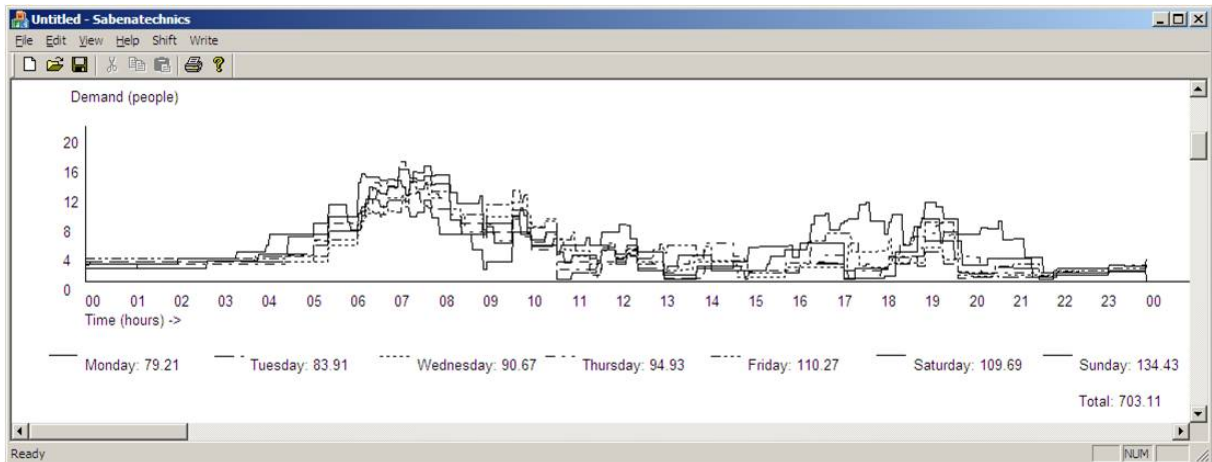
### Visualization

When analyzing business problems, visualization is often useful to help the analyst better understand the problem and suggest improvements. Our situation is a good example of how visualization can help.

Figure 3 provides a clear view of the peaks in the demand for services during the week. To build the demand curves for each day of the week, we divide the estimated demand for each flight by the length of the maintenance time window (the time between STA and STD). Hence, in Figure 3, we assume that the flights' demands are evenly distributed over the STA–STD time windows.

This visualization is useful because it highlights some important characteristics of the demand, leading to two main observations. First, each day has two intervals of peak demand:

Figure 3: Demand distribution per day



a large peak between 06:00 and 09:00 and a smaller peak between 17:00 and 20:00. Second, on Friday, Saturday, and Sunday, the demand is 20 percent to 50 percent higher than on the other days of the week.

These observations gave rise to suggestions for improvement before any optimization occurred. The first observation suggests that shorter shift lengths (e.g., shifts of four hours instead of eight hours) can better cover the intervals of peak demand and might result in lower costs. However, because the management team believed that the employees would not accept shorter shift lengths, we did not examine this option further. The second observation is relevant to the weekend constraint discussed in the *Constraints* subsection—that workers can only work at most one of two weekends. Because weekend demand is higher than weekday demand, relaxing this constraint could have a substantial impact on costs. We analyzed a number of scenarios (e.g., working two of three weekends, working three of five weekends, adding an extra weekend cycle) in which this constraint was relaxed (see the *Weekend Constraint* subsection).

## Optimizing Scenarios

The model generated the new rosters for the 2008 winter season. The algorithm returned over 50 feasible solutions, all of them within 5 percent of the theoretical optimum (current lower bound). The company analyzed the least-cost solutions, which formed the subject of an intensive negotiation process. Our first model runs returned rosters that resulted in workforce cost savings of no less than 40 percent (from approximately 32,000 euros to

20,000 euros per week). Unfortunately, the management team did not accept these first rosters because of several shortcomings that stakeholders raised. Therefore, we adapted the initial model several times; each time we incorporated changes to address stakeholder concerns raised in our presentations of the rosters. The different analyses included the introduction of new and (or) more restrictive constraints, the examination of alternative scenarios, sensitivity analysis, and what-if questions. Generating new rosters after each negotiation process required several hours (days) of computation time. Below, we discuss nine major concerns put forward during these seven meetings and how we adapted the model to address them. This case study illustrates how optimization can help in analyzing different scenarios quickly and quantifying the trade-offs between cost and quality. After seven negotiation rounds, the schedule that was agreed upon for implementation resulted in a cost savings of 6 percent. All suggestions that are listed below, except for the relaxation of the weekend constraints and the introduction of a weekend cycle, were accepted and implemented.

### **Capacity Buffer.**

The first rosters were not robust; for example, a small difference in the workload to maintain a particular flight could cause huge problems because of the lack of a capacity buffer. In addition, there was the risk of reduced capacity because of factors such as employee illness. After consultation with the operations manager, we agreed that a 15 percent capacity buffer would suffice. We adapted the model by changing the coverage constraints to ensure that scheduled capacity exceeded the scheduled maintenance demands by at least 15 percent for each 15-minute interval during the day.

### **Time buffer.**

The operations manager's next concern related to the robustness of the resulting rosters. In the previously generated rosters, flight maintenance was often planned immediately after a flight's STA. Hence, in the event of a flight delay, maintenance could not be performed as planned. To avoid potential problems, we removed the maintenance timing variables that correspond to workload assignments immediately after each flight's STA, thus ensuring that maintenance can only start at a fixed time after the STA. The operations manager first

proposed to start the maintenance 30 minutes after the flight arrival. However, this led to an infeasible problem because some flights take off one hour after arrival. Hence, for flights with short time windows, maintenance was allowed to take place 15 minutes after flight arrival. A second successful attempt to make the rosters more robust was the addition of an extra coverage constraint to ensure that for all model periods, including those during which no client flights require service, at least one team will be on standby to serve flights with disrupted schedules. All stakeholders agreed to incorporating this extra coverage constraint.

### **Hidden maintenance.**

Maintenance must be avoided immediately before the flight's STD. Passengers do not like to see aircraft maintenance while they are waiting at the gate. As above, we adapted the model to handle this concern by narrowing the feasible time window for maintenance. The latest maintenance time was set to 15 or 30 minutes before departure, depending on the duration of the available time window. After including the time buffer and the constraints to ensure hidden maintenance, the demand becomes more strongly peaked (see Figure 4) compared to the demand in Figure 3.

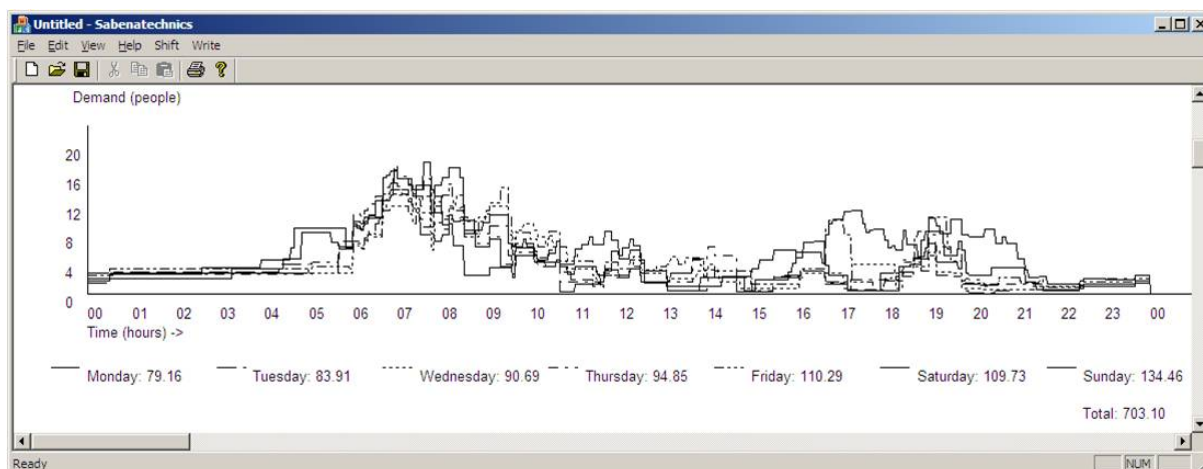
### **Shift transition and work transfer.**

Initially, we assumed that workers are productive during their entire shifts. However, when teams finish their shifts, they must transfer maintenance jobs to the incoming teams. Therefore, the first and the last quarter hour of each shift cannot be spent to maintain flights. To model this, we simply excluded the first and last quarter hour of each shift from the left side of the coverage constraints, turning these quarters into 'unproductive' periods. In addition, the unproductive first quarter-hour forms an extra time buffer to protect against late arrivals of workers.

### **Service contracts.**

We estimated the flight maintenance demand based on the large number of data records from the previous year. During the meetings, however, a sales manager questioned the estimated demand for a number of important clients. After further investigation, the data on which the demand estimates were based proved to be correct. However, these clients had negotiated

Figure 4: The changed demand pattern after the inclusion of the time buffer and the hidden maintenance constraints



service contracts in which the presence of at least two employees was required during the entire time interval between arrival and departure. Hence, even if the real maintenance demand was, for example, only one man-hour, the scheduled capacity for these flights must equal two times the number of hours between the STA and STD. We adapted the model by ensuring sufficient capacity for the maintenance of the flights subject to these service level agreements; we fixed the maintenance timing variables at the level of the required amounts during certain periods as stipulated in the special service contracts. In addition, we removed the constraints that specify that these flights must be maintained; fixing the maintenance timing variables automatically satisfied these constraints. The resulting rosters had a much higher cost. This motivated a sensitivity analysis in which the impact of these contracts (and alternatives) on workforce costs was analyzed. As a result, Sabena Technics is planning to renegotiate many of these contracts when they expire.

### **Work pressure and shift successions.**

The management team did not accept a number of our rosters (generally, the least-expensive rosters) because it expected problems with the labor unions. The least-expensive rosters resulted in an occupation rate (total demand divided by total scheduled capacity) of 80 percent or more; the existing occupation rate was about 50 percent. Therefore, employee work pressure would suddenly increase dramatically, which would certainly lead to employee dissatisfaction and possibly strikes. Some of the generated rosters also exhibited less-desirable



shift successions (e.g., a number of day shifts immediately followed by a morning shift instead of a day off between different shift types). Moreover, a lower employee satisfaction rate entails a lower quality of work and hence a potential safety problem. Because safety has the highest priority for Sabena Technics, it did not retain these unfavorable rosters. To overcome these problems, we adapted the model by adding an extra capacity buffer of 5 percent and a constraint to ensure that, within a roster, the number of shifts of each shift type was constant on each day of the week. Therefore, in the second phase (after optimization), one could easily construct a roster in which each block (period of days between days off) includes only one shift type. We refer to this as the "desirable shift-successions" scenario. However, the rosters that this scenario generated were too expensive; hence, we removed this last constraint from the model. The occupation rate of the final schedule, 59 percent, is higher than that of the original schedule, which might suggest that the cost savings have come at the expense of robustness. However, the original schedule included many periods with no (or even a negative) excess capacity; in the new schedule, the excess capacity is at least 20 percent at any given time. Thus, much of the benefit comes from a more even distribution of buffer capacity, leading to a less-expensive and significantly more robust solution.

### **Lower bound on the average number of work hours.**

Some generated rosters included a cycle in which the employees were scheduled for fewer than 20 hours per week on average. We did not include a minimum bound on the average number of work hours in the initial model, because the remaining hours could be "sold internally" to another department. For example, one could transfer an employee working in a cycle of 20 hours for two shifts of 9 hours to another department to obtain a 38-hour working week. However, the number of employees working in such a cycle was sometimes higher than the demand for workers in the other departments. Moreover, some rosters had a cycle for which the remaining hours, for example, 12 hours, could not easily be divided into 8-hour or 9-hour shifts. When confronted with these rosters, the management team decided to set a minimum of 36 hours for the average number of working hours. In the event of a 36-hour work week (on average), the 2 remaining hours could be used for in-service training.

### **Night cycle.**

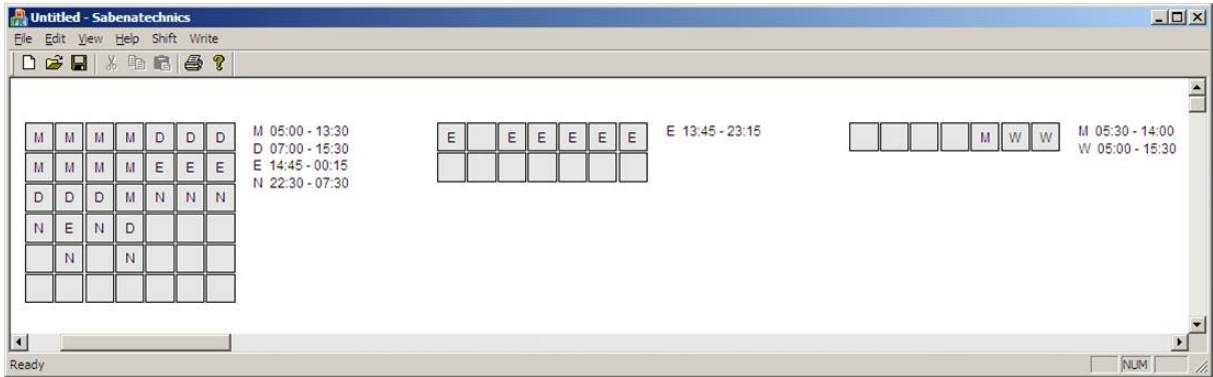
In the original roster, one cycle included night shifts only (except for one morning shift every two weeks). In practical terms, this meant that the employees working in this cycle were used to working night shifts. Management expected that these employees would not be satisfied with a cycle in which they would alternate between blocks of day and night shifts. Therefore, we extended the model with an extra constraint stating that in the second cycle, only night shifts were allowed. The resulting rosters gave insight into the trade-off between the roster cost and the roster quality as experienced by the employees. For an efficient implementation of this constraint, we simply removed the shift variables (i.e., the variables that assign shifts to days) for the M, D, and E shifts in the second cycle.

### **Weekend constraint.**

The model adaptations discussed above increased the cost of the rosters obtained. Based on the visualization and the sensitivity analysis of the weekend constraints, we analyzed alternative weekend-constraint scenarios to reduce the total cost. Recall that the weekend constraint implies that one of two weekends must be free of any shift. We reran the model with weekend constraints of 2/3, 3/5, and combinations of these constraints (and of the original constraint of 1/2) for the various cycles. However, the savings we obtained were not sufficient to warrant a proposal to the labor unions.

A second suggestion to better address the high weekend demand was the introduction of an extra weekend cycle. Workers in this so-called weekend cycle would complete shifts on both Saturday and Sunday in each week (i.e., a weekend constraint of 1/1) (see Figure 5). To find volunteers prepared to work in this cycle, we made it attractive by allowing a decrease in the number of work hours per week (to 28 instead of 38) and by ensuring that the workers had a block of four days off each week (either Tuesday to Friday or Monday to Thursday). To this end, we introduced a weekend shift of 10 hours and restricted the remaining 8-hour shift to be scheduled on either Friday or Monday. The resulting rosters were less expensive than the rosters without a weekend cycle. However, they were rejected after negotiations with the labor unions because of a lack of employee interest.

Figure 5: Example of a weekend cycle, shown on the upper right (note that in these cycles the shifts still have to be divided on a weekly basis)



## Conclusion

In this paper, we discussed the use of an MILP-based enumeration method for workforce staffing and scheduling. The solution integrates the staffing and scheduling phases and is characterized by a special coverage constraint in which the timing of the maintenance represents an extra decision. The exact enumeration algorithm was used in a heuristic setting and returned all the feasible schedules whose cost does not exceed the theoretical optimum by more than 5 percent. These schedules were evaluated in several management meetings in which stakeholders expressed concerns and made suggestions. Based on their feedback, we added new constraints and analyzed alternative scenarios to provide lower costs and less excess capacity. Without the optimization model, evaluating these scenarios within the limited time available between meetings would have been impossible. In the meetings, visualization was also crucial to help attendees detect the binding constraints in the model and to suggest opportunities for improvement. The model's output consists of several proposals of (nearly) optimal rosters, which were used in the negotiations with the labor unions to find a new generally accepted schedule. The final schedule saved 6 percent over the existing schedule.

## Acknowledgements

The authors are grateful to Willy Buysse and Thierry Van Damme of Sabena Technics for their enthusiasm for our work and their feedback on it.

## Appendix: The MILP Model

This appendix presents the MILP model that we solved for each combination in the enumeration procedure.

### *Indices and sets*

- $i \in I$  : feasible shifts.
- $t \in T$  : shift types  $\{M, D, E, N.\}$
- $i \in I_t$  : feasible shifts of shift type  $t$ .
- $c \in C$  : cycles in the schedule.
- $d \in \{1, 2, \dots, 7\}$  : days in the week.
- $p \in P$  : periods in one week (e.g., the 672 15-minute periods).
- $f \in F$  : flights to be serviced.
- $f \in F_p$  : flights that are available to be serviced during period  $p$ .
- $p \in P_f$  : periods during which flight  $f$  can be serviced.

### *Data parameters*

- $L_f$  : the demand (in man-hours) of flight  $f$ .
- $M_c$  : the team size in cycle  $c$ .
- $T_c$  : the number of teams (= weeks) in cycle  $c$ .
- $K_{id}$  : the total cost (for one worker) of shift  $i$  on day  $d$ .
- $D_i$  : the duration of shift  $i$  (in hours).
- $B$  : the required capacity buffer (%).
- $a_{idp}$  : = 1 if period  $p$  is included in shift  $i$  on day  $d$ ; 0 otherwise.
- $b_i$  : fraction of duration of shift  $i$  available to work.  
= 1 - (duration of lunch break / duration of shift  $i$ ).

### *Decision variables*

- $x_{idc}$  = the number of shifts  $i$  scheduled during day  $d$  in cycle  $c$ .
- $y_{fp}$  = the number of workers assigned to maintain flight  $f$  during period  $p$  (referred to as the maintenance timing variables throughout this paper).
- $z_{ic} = 1$ , if shift  $i$  is used in cycle  $c$ ; 0 otherwise.

The  $y_{fp}$  variables are defined for the periods  $p$  that fall within the time window (i.e., between the STA and STD) of each flight  $f$ . To model the time buffer and hidden maintenance constraints, some of these  $y_{fp}$  variables (i.e., those for which  $p \leq (\text{STA} + \text{time buffer})$  or

$p \geq$  (STD - number of periods stipulated in the hidden maintenance requirement) for each flight) are removed from the model.

The model can be stated as follows:

Minimize

$$\sum_{c \in C} \sum_{i \in I} \sum_{d=1}^7 K_{id} M_c x_{idc} \quad (1)$$

subject to

$$\sum_{i \in I} x_{idc} \leq T_c, \quad \forall d = 1, \dots, 5 \quad \forall c \in C \quad (2)$$

$$\sum_{i \in I} x_{idc} \leq 0.5T_c, \quad \forall d = 6, 7 \quad \forall c \in C \quad (3)$$

$$36T_c \leq \sum_{i \in I} \sum_{d=1}^7 D_i x_{idc} \leq 38T_c, \quad \forall c \in C \quad (4)$$

$$x_{idc} \leq T_c z_{ic}, \quad \forall i \in I \quad \forall d = 1, \dots, 7 \quad \forall c \in C \quad (5)$$

$$\sum_{i \in I_t} z_{ic} \leq 1, \quad \forall t \in T \quad \forall c \in C \quad (6)$$

$$\sum_{i \in I} \sum_{d=1}^7 \sum_{c \in C} a_{idp} x_{idc} \geq 1, \quad \forall p \in P \quad (7)$$

$$\sum_{i \in I} \sum_{d=1}^7 \sum_{c \in C} a_{idp} b_i M_c x_{idc} \geq (1 + B) \sum_{f \in F_p} y_{fp}, \quad \forall p \in P \quad (8)$$

$$\sum_{p \in P_f} y_{fp} = L_f * \frac{|P|}{24 * 7}, \quad \forall f \in F \quad (9)$$

$$\sum_{i \in I_t} x_{idc} = \sum_{i \in I_t} x_{i(d+1)c} \quad \forall c \in C \quad \forall t \in T \quad \forall d = 1, \dots, 6 \quad (10)$$

$$x_{idc} \in \{0, 1, \dots, T_c\}, \quad \forall i \in I \quad \forall d = 1, \dots, 7 \quad \forall c \in C \quad (11)$$

$$y_{fp} \geq 0, \quad \forall f \in F \quad \forall p \in P \quad (12)$$

$$z_{ic} \in \{0, 1\}, \quad \forall i \in I \quad \forall c \in C. \quad (13)$$

The objective function (1) minimizes the total costs. The cost of a shift  $K_{id}$  grows linearly with the shift hours. Furthermore, the shift cost depends on the shift type (day shifts are the least expensive, morning shifts incur an extra premium of 7 percent, evening shifts

incur an extra premium of 9 percent, and night shifts incur an extra premium of 20 percent plus an additional fixed bonus of 45 euro per night shift) and on the day of the week (shifts on Saturday have an extra premium of 16.67 percent. Sunday shifts incur an extra premium of 95 percent).

Constraint set (2) guarantees that the number of shifts scheduled on a particular day during the cycle does not exceed the number of teams (= weeks) in the cycle.

Constraint set (3) ensures that for each employee, at least one of two weekends is free of any shifts. Hence, this constraint implements the weekend bottleneck. Recall that in addition to  $1/2$ , we also experimented with other weekend fractions:  $2/3$ ,  $3/5$ , etc.

Constraint sets (4) ensure that the average number of working hours per week per worker is between 36 and 38 hours. Recall that the lower bound of 36 hours was not present in the initial model, but was added after one of the revision rounds, as we describe in *Lower bound on the average number of work hours*.

Constraint sets (5) and (6) guarantee that all shifts (M, D, E, N) have the same hours within one cycle.

Constraint sets (7)–(9) model the coverage constraints. Constraint set (7) ensures that at least one team is always present to service flights with disrupted schedules. Constraint set (8) ensures that the capacity (i.e., workers present) always exceeds the demand (i.e., the planned maintenance, increased by the required capacity buffer). Constraint set (9) guarantees that each flight is maintained.

Constraint set (10) was only present in the "desirable shift-successions" scenario, as we describe in *Work pressure and shift successions*. This constraint set ensures that the number of shifts of a certain type is the same on all days within a particular cycle.

Finally, constraints (11)–(13) define the domains of the decision variables.

A typical MILP for our problem setting has the following dimensions: a cycle time of seven days and four shift types with 57 feasible shifts, which results in 798 ( $=7*57*2$ ) integer variables in the case of two cycles and 1,197 ( $=7*57*3$ ) integer variables in the case of three

cycles. Additionally, 310 flights have an average time window of 20 15-minute periods, which results in 6,200 continuous variables to model the flight maintenance timing decisions. The numbers of constraints are 2,476 and 2,887 for the two-cycle and three-cycle cases, respectively.

Solving such a typical MILP to optimality using IBM ILOG CPLEX 12.0 (standard settings) takes less than one minute for some combinations; however, others require more than three hours. To limit the overall computation time, we gave each MILP a tolerance gap and a time limit. When we had enough time between two meetings to evaluate a new scenario, we set the tolerance gap to 1 percent and the time limit to 10 minutes. However, when we had to evaluate a scenario in less than one hour, we specified a 5 percent optimality gap and a time limit of one minute. As a result, the total time required to run the complete enumeration procedure ranged from one (5 percent gap and one minute for each MILP) to 40 hours (1 percent gap and 10 minutes for each MILP), dependent on the specific problem instance and scenario constraints (see the *Optimization* section). Because new rosters are only generated every six months (see the *Problem Description* section), these computation times are acceptable.

We coded the algorithm in C++ and ran it on a 2.4 GHz Pentium 4 personal computer running the Windows XP operating system. The MILP models are solved by the IBM ILOG CPLEX 12.0 MILP optimizer through the callable library. A GUI developed using the Microsoft Foundation Classes (MFC) visualizes the results and allows for easy interaction with the human scheduler. The figures throughout this paper depict screenshots of this GUI.

## References

- Alfares, H.K. 1999. Aircraft maintenance workforce scheduling: A case study. *J. Quality Maintenance Engrg* **5(2)** 78–88.
- Alfares, H.K. 2007. A simulation approach for stochastic employee days-off scheduling. *Internat. J. Model. Simulation* **27(1)** 9–15.

- Barnhart, C., A. Cohn. 2004. Airline schedule planning: Accomplishments and opportunities. *Manufacturing Service Oper. Management* **6(1)** 3–22.
- Beliën, J., E. Demeuleester, B. Cardoen. 2010. Integrated staffing and scheduling for an aircraft line maintenance problem. Tech. rep., Hogeschool Universiteit Brussel, Research Paper 2010/23.
- Burke, E.K., P. De Causmaecker, G. Vanden Berghe, H. Van Landeghem. 2004. The state of the art of nurse rostering. *J. Scheduling* **7(6)** 441–499.
- Butchers, E., P. Day, A. Goldie, S. Miller, J. Meyer, D. Ryan, A. Scott, C. Wallace. 2001. Optimized crew scheduling at Air New Zealand. *Interfaces* **31(1)** 30–56.
- Desrosiers, J., A. Lasry, D. McInnis, M. Solomon, F. Soumis. 2000. Air Transat uses ALTITUDE to manage its aircraft routing, crew pairing, and work assignment. *Interfaces* **30(2)** 41–53.
- Ernst, A.T., H. Jiang, M. Krishnamoorthy, D. Sier. 2004. Scheduling and rostering: A review of applications, methods, and models. *Eur. J. Oper. Res.* **153(1)** 3–27.
- Lesaint, David, Christos Voudouris, Nader Azarmi. 2000. Dynamic workforce scheduling for british telecommunications plc. *Interfaces* **30(1)** 45–56.
- Sabena Technics, TAT Group. 2010. Airframe services. accessed october 18, 2010. <http://www.sabenatechnics.com>.
- Yu, G., M. Argüello, G. Song, S. McCowan, A. White. 2003. A new era for crew recovery at Continental Airlines. *Interfaces* **33(1)** 5–22.