

# A mathematical programming approach for scheduling physicians in the emergency room

Huguette Beaulieu<sup>a</sup>, Jacques A. Ferland<sup>b</sup>, Bernard Gendron<sup>b,\*</sup> and Philippe Michelon<sup>c</sup>

<sup>a</sup> Imperial Oil Limited, 111 St. Clair Ave. W., Toronto, Ontario, Canada M5W 1K3

<sup>b</sup> Département d'informatique et de recherche opérationnelle, Université de Montréal, P.O. Box 6128, Succursale Centre-Ville, Montréal, Québec, Canada H3C 3J7

<sup>c</sup> Laboratoire d'informatique d'Avignon, Université d'Avignon et des Pays de Vaucluse, B.P. 1228, F-84911 Avignon, France

Received 27 July 1999; accepted 4 February 2000

Preparing a schedule for physicians in the emergency room is a complex task, which requires taking into account a large number of (often conflicting) rules, related to various aspects: limits on the number of consecutive shifts or weekly hours, special rules for night shifts and weekends, seniority rules, vacation periods, individual preferences, . . . In this paper, we present a mathematical programming approach to facilitate this task. The approach models the situation in a major hospital of the Montréal region (approximately 20 physicians are members of the working staff). We show that the approach can significantly reduce the time and the effort required to construct a six-month schedule. A human expert, member of the working staff, typically requires a whole dedicated week to perform this task, with the help of a spreadsheet. With our approach, a schedule can be completed in less than one day. Our approach also generates better schedules than those produced by the expert, because it can take into account simultaneously more rules than any human expert can do.

**Keywords:** health administration, emergency physician scheduling, mathematical programming

## 1. Introduction

Preparing a schedule for physicians in the emergency room is a complex task, which requires taking into account a large number of (often conflicting) rules, related to various aspects: limits on the number of consecutive shifts or weekly hours, special rules for night shifts and weekends, seniority rules, vacation periods, individual preferences, . . . In this paper, we present a mathematical programming approach to facilitate this task. The approach models the situation in a major hospital of the Montréal region (Sacré-Coeur Hospital, where approximately 20 physicians are members of the working staff at the emergency room).

The paper is organized as follows. First, our contributions are highlighted in section 2, which provides a review of the literature on methods for scheduling health care personnel. Then, section 3 gives an overview of the problem faced by the planner in our case study. The mathematical programming model is presented in section 4. In section 5, we describe the solution method based on the model, we give a brief outline of its implementation (the interested reader is referred to [2] for further details), and we compare the schedules it produces with those proposed by a human expert (a member of the working staff who has been in charge of the scheduling task for many years). In the conclusion, we summarize our work and propose extensions.

\* Corresponding author. E-mail: gendron@iro.umontreal.ca.

## 2. Literature review

Methods for generating workforce schedules are typically divided into cyclic and non-cyclic approaches. Cyclic techniques proceed by defining fixed sequences of shifts, which are then assigned equally (or almost) among workers. Such techniques are well-adapted to situations where the same schedules can accommodate all workers. Even then, they must account for vacations and days-off, and in the presence of many seniority rules and individual preferences, as in our case (see the next section), they are usually of no help. Non-cyclic methods must then be considered, which generally fall into two categories: (1) those requiring human expertise and the use of a spreadsheet (which helps the planner to balance the schedules among categories of workers); (2) optimization approaches. These latter methods have two main advantages over the former ones: first, they require very little human intervention and can therefore be (almost) fully computerized; second, when properly designed, they can handle many more rules simultaneously than any human expert can do, even with the help of a spreadsheet. However, mastering them is by no means a trivial task, and their application to a specific case might require several years of development.

Much of the research on scheduling health care personnel has been devoted to the case of hospital nurses [18]. A recent account of the nurse scheduling literature is given in [9]. An example of a non-cyclic method based on human expertise and the use of a spreadsheet is given in [20], while classical optimization approaches are described in [14,23]. Progress in computer technology and software tools has

recently provided a number of successful applications of optimization methods to nurse scheduling problems (see in particular [4,5,8,10,16,22,24] and the references therein).

Surprisingly, the problem of scheduling physicians in the emergency room has not attracted much attention. In a recent survey on the topic, which covers the situation in six hospitals located in the Montréal area, Lapierre and Carter [13] highlight the basic difference between physicians and nurses: the former are not employed by the hospitals, contrary to the latter. Therefore, in the case of nurse scheduling, both maximizing personnel satisfaction and minimizing salary cost are often considered as two objectives to achieve simultaneously. In the case of emergency physician scheduling, maximizing satisfaction only matters, as physician retention is the most critical issue faced by hospital administrations (according to Lapierre and Carter). In addition, nurse schedules must adhere to collective union agreements, while emergency physician schedules are more driven by personal preferences. In general, planning the schedules for emergency physicians requires satisfying a very large number of (often conflicting) rules. Examples of such rules are given in the next section, which describes our case study (in light of Lapierre and Carter's survey, it appears very representative of the situation in other major hospitals).

We are aware of only two applications of operations research methods to emergency physician scheduling problems: [21], which presents a methodology based on stochastic models, and [11,12], where a cyclic method is improved by the use of modern heuristic techniques. To the best of our knowledge, our work is the first to present a mathematical programming approach for scheduling physicians in the emergency room. Note that some commercial software packages for emergency physician scheduling are available [1,6,15]. After a careful examination of the documentation available from their Web sites, it seemed to us that these software packages cannot handle all the rules of our problem. We should point out, however, that a fair evaluation of the capabilities of these software packages to deal with our problem would require extensive experiments.

### 3. Problem overview

At the Sacré-Coeur Hospital of Montréal, schedules for emergency physicians are established once every six months, with a special schedule being also planned for the two-week Christmas period. A human expert, member of the working staff, is in charge of the task (for many years, the same person has been responsible for generating the schedules). The staff is composed of 20 physicians, including 15 working full-time. Among them, there can be up to five "young" physicians (with less than three years of experience). These figures are only indicative, as the situation is changing every year.

Approximately two months before the beginning of the next planning horizon, each physician submits a list of indi-

vidual preferences for the next period; these include vacations and days-off, number of weekly hours, desired shifts or sequences of shifts (for example, some physicians prefer to work three consecutive nights, whenever they have to work at least one, while others prefer to work only one night at a time), . . . These personal preferences are inserted into a spreadsheet file representing a typical schedule: the days are along the horizontal axis and the physicians along the vertical one. The planner works directly on this sheet, starting by manually fixing the weekends, then the nights, and finally the other shifts. When assigning shifts to physicians, the planner attempts to respect a large set of rules, including individual preferences, but also ergonomic rules (such as after 2 or 3 nights worked, a physician should have the benefit of 2 or 3 days of rest), seniority rules (most senior physicians work fewer weekends and nights than others), . . . The use of the spreadsheet is essential in this context: it facilitates the task of balancing the number of shifts of each type (for example, day, evening or night) among physicians (according to their seniority). This is done by compiling statistics for each type of shift, statistics which include not only the days of the current planning horizon, but also those of the previous one (to account for possible unbalances in the previous schedule).

The process of generating the schedule by successive manual assignments and corrections guided by the statistics requires a whole dedicated week. This time is considered excessive by the planner. Moreover, because of the high risk of errors inherent to this type of manual approach, the planning time might be further increased (for example, mistakes can be discovered after the distribution of the "final" schedule, which can force the planner to go back to his working table).

However, the most annoying problem faced by the planner is the following: when attempting to assign a particular shift, it frequently happens that none of the available physicians can be assigned without violating important rules. The planner usually reacts by "backtracking" over his previous assignments. This process can be very long, and after a few trials, the planner will often be satisfied with an assignment that still violates some rules.

The challenge that we were facing when we started this project was therefore to design a solution approach that respects more rules than the human expert usually does and that requires significantly less time. Our approach is based on a mathematical programming formulation of the problem, which we describe in the next section.

### 4. The model

The model is an abstract representation of the rules of the problem, written in mathematical language. Rules are translated into constraints, which are linear inequalities built around variables. Before providing a detailed description of the model, we give an overview of the basic rules applied at the Sacré-Coeur Hospital of Montréal.

We first distinguish whether the rules are *compulsory* (e.g., rules that must absolutely be enforced) or *flexible* (e.g., rules that can occasionally be violated, at the cost of losing some “quality”). Demand rules are the most basic in the first category. They define how many physicians should work at different periods of a day and which responsibilities are attached to particular shifts. Each day is divided into three periods of eight hours: day, evening and night. Three physicians (two on weekends or holidays) work during day and evening shifts, including one exclusively in charge of traumas (“heavy” emergencies). “Trauma” shifts are considered heavier than “regular” shifts (which mostly involve the treatment of “light” cases and patients in stabilized condition). At night, there is only one night shift, the physician assuming the responsibilities of trauma and regular shifts. Three days per week, one physician works a four-hour shift, the “follow-up” shift, when he receives by appointment patients that have recently been treated at the emergency room. Other compulsory rules include: vacations, days-off, or particular shifts requested by the physicians, and the basic ergonomic rule: “there must be at least 16 hours between the end of one shift and the beginning of another one”.

Flexible rules can be divided into two categories: *ergonomic rules* which aim at improving the “quality” of the schedule of each physician, and *distribution rules* which aim at distributing the assignment of particular types of shifts among physicians, sometimes according to their seniority. Flexible rules are frequently conflicting with each other, so they cannot always be satisfied simultaneously. Hence, some of these are regarded as “goals” to be reached, allowing for small deviations. Our approach exploits this feature. Indeed, the objective of the model is to minimize all deviations, which amounts to finding an *efficient solution* (e.g., such that we cannot find another solution with smaller or equal deviations and with at least one strictly smaller deviation). Hence, in the operations research jargon, the model is a special case of *multi-objective integer programming* (e.g., it includes several objectives, specified by the deviations, and all variables must take on integer values) [17,19].

#### 4.1. Notation and variables

The following notation is required to formulate the model:

1. Set of *physicians*  $I$ .
2. Set of *days of the planning period*  $J = \{1, 2, \dots, n\}$ . We assume day 1 is a Monday, day  $n$  is a Sunday and  $n \geq 28$  (otherwise, some constraints related to weekend shifts cannot be modeled). In addition, we also consider the set  $J_P = \{-m+1, \dots, -1, 0\}$ , which includes  $m$  days of the previous planning period (in our application, we used  $m = 5$ ). The assignments during these days are required to guarantee the continuity of the planning.

3. Set of *shifts*  $K$ . In the model description, we consider several subsets of this set, namely:  $K_D$ , the day shifts,  $K_E$ , the evening shifts,  $K_N$ , the night shifts,  $K_R$ , the regular shifts, and  $K_T$ , the trauma shifts.

Three different types of variables are used to formulate the model. The *assignment variables* are decision variables to indicate whether or not physician  $i$  is assigned to shift  $k$  on day  $j$ :

$$x_{ijk} = \begin{cases} 1, & \text{physician } i \text{ is assigned to shift } k \text{ on day } j, \\ 0, & \text{otherwise.} \end{cases}$$

A second type of variables is required to formulate constraints associated with rules involving sequences of consecutive shifts to be followed by days-off. These variables are called *succession variables*. They are specified shortly when we introduce this type of constraints.

Finally, *deviation variables* are used to capture positive and negative deviations from the targets in the constraints associated with the so-called “goal” rules. As mentioned above, the objective of the model is specified in terms of these deviation variables.

#### 4.2. Constraints

The constraints of the model are partitioned into four categories according to the types of rules to which they correspond: *compulsory constraints*, *ergonomic constraints*, *distribution constraints*, and *goal constraints*. It is worth noting that most of the constraints we introduce can be easily modified and that additional constraints can be included within the same framework to account for any specific application.

##### 4.2.1. Compulsory constraints

1. One physician must be assigned to each shift of the period:

$$\sum_{i \in I} x_{ijk} = 1, \quad j \in J, k \in K(j),$$

where  $K(j)$  is the set of shifts to be completed during day  $j$ . In order to simplify the notation, in the remainder we assume that  $K(j) = K$ ,  $j \in J$ . Note that, in order to satisfy these constraints, we might add to set  $I$  a “dummy” physician, who completes shifts that cannot be assigned to the regular members of the working staff.

2. A physician cannot be assigned to more than one shift per day:

$$\sum_{k \in K} x_{ijk} \leq 1, \quad i \in I, j \in J.$$

3. A physician assigned to an evening shift cannot be assigned to a day shift of the day after:

$$\sum_{k \in K_E} x_{i(j-1)k} + \sum_{k \in K_D} x_{ijk} \leq 1, \quad i \in I, j \in J.$$

4. A physician assigned to a night shift must not be assigned to a shift of another type on the next day:

$$\sum_{k \in K_N} x_{i(j-1)k} + \sum_{k \notin K_N} x_{ijk} \leq 1, \quad i \in I, j \in J.$$

Note that a physician can be assigned to the night shift of two consecutive days. Note also that the last three constraints imply the satisfaction of the basic ergonomic rule mentioned above: “there must be at least 16 hours between the end of one shift and the beginning of another one”.

5. Vacations, days-off, particular shifts requested by physicians:

$$x_{ijk} = 1 \quad (0), \quad i \in I, j \in \tilde{J}, k \in \tilde{K},$$

where  $\tilde{J} \subseteq J$  and  $\tilde{K} \subseteq K$  are used to represent either working days (or vacations and days-off) requested by a physician, or shifts a physician requested (not) being assigned to.

#### 4.2.2. Ergonomic constraints

1. Upper limits on the number of weekly (or monthly) hours of certain types of shifts:

$$\sum_{j \in \tilde{J}} \sum_{k \in \tilde{K}} h_k x_{ijk} \leq U(\tilde{J}, \tilde{K}), \quad i \in I,$$

where  $h_k$  represents the number of hours corresponding to shift  $k$ ;  $\tilde{J} \subseteq J$  and  $\tilde{K} \subseteq K$  capture weeks or months, and the types of shifts (for example, night or follow-up shifts, or even all shifts), respectively; and  $U(\tilde{J}, \tilde{K})$  is the upper limit on the number of hours corresponding to days  $\tilde{J}$  and shifts  $\tilde{K}$ .

2. Limited number of successive working days:

$$\sum_{l=j-d}^j \sum_{k \in K} x_{ilk} \leq d, \quad i \in I, j \in J,$$

where  $d \leq m$  denotes the admissible maximum number of successive working days (typically,  $d = 4$  is used, but a limit of five days can be used during vacation times, when four or more physicians are away at the same time).

3. Consecutive weekend periods (evening, day or night) are assigned together to the same physician, trauma and regular shifts alternating between physicians (note that weekends include Friday evenings and Friday nights):

$$\begin{aligned} \sum_{k \in K_N} x_{ijk} &= \sum_{k \in K_N} x_{i(j+1)k} = \sum_{k \in K_N} x_{i(j+2)k}, \\ i &\in I, j \in \{5, 12, \dots, n-2\}, \\ \sum_{k \in K_E \cap K_T} x_{ijk} &= \sum_{k \in K_E \cap K_R} x_{i(j+1)k} \\ &= \sum_{k \in K_E \cap K_T} x_{i(j+2)k}, \end{aligned}$$

$$\begin{aligned} i &\in I, j \in \{5, 12, \dots, n-2\}, \\ \sum_{k \in K_E \cap K_R} x_{ijk} &= \sum_{k \in K_E \cap K_T} x_{i(j+1)k} \\ &= \sum_{k \in K_E \cap K_R} x_{i(j+2)k}, \\ i &\in I, j \in \{5, 12, \dots, n-2\}, \\ \sum_{k \in K_D \cap K_T} x_{ijk} &= \sum_{k \in K_D \cap K_R} x_{i(j+1)k}, \\ i &\in I, j \in \{6, 13, \dots, n-1\}, \\ \sum_{k \in K_D \cap K_R} x_{ijk} &= \sum_{k \in K_D \cap K_T} x_{i(j+1)k}, \\ i &\in I, j \in \{6, 13, \dots, n-1\}. \end{aligned}$$

4. After working a weekend, a physician should not work on the next Monday:

$$\begin{aligned} \sum_{k \in K} \{x_{i(j-1)k} + x_{ijk}\} &\leq 1, \\ i &\in I, j \in \{1, 8, \dots, n-6\}. \end{aligned}$$

5. Whenever he works a night shift, a physician requests to work three consecutive night shifts:

$$\begin{aligned} \sum_{k \in K_N} \{x_{ijk} - x_{i(j+1)k}\} &= 0, \\ i &\in \tilde{I}, j \in \{2, 9, \dots, n-5\}, \\ \sum_{k \in K_N} \{x_{ijk} - x_{i(j-1)k} - x_{i(j+2)k}\} &= 0, \\ i &\in \tilde{I}, j \in \{2, 9, \dots, n-5\}, \end{aligned}$$

where  $\tilde{I} \subseteq I$  is the set of physicians who require to work three consecutive nights whenever they work one (some physicians prefer to work one or two nights successively, while others do not express any preference in this respect). Note that the constraints need only be defined for Tuesdays, as corresponding constraints for weekend days are already taken into account by the constraints on weekends, defined above.

6. If a physician is assigned to a night shift after completing any shift type on the day before, then he must have at least two days off before being assigned to a shift of any other type than a night shift. We have to introduce succession variables  $s_{ij}^1$  to formulate constraints associated with this rule. These variables are specified through the following constraints:

$$\begin{aligned} s_{ij}^1 - \sum_{k \in K} x_{i(j-2)k} &\leq 0, \quad i \in I, j \in J, \\ s_{ij}^1 - \sum_{k \in K_N} x_{i(j-1)k} &\leq 0, \quad i \in I, j \in J, \\ s_{ij}^1 - \sum_{k \in K} x_{i(j-2)k} - \sum_{k \in K_N} x_{i(j-1)k} + \sum_{k \in K_N} x_{ijk} \\ &\geq -1, \quad i \in I, j \in J, \\ s_{ij}^1 + \sum_{k \in K_N} x_{ijk} &\leq 1, \quad i \in I, j \in J. \end{aligned}$$



Now, it is easy to verify that

$$s_{ij}^1 = \begin{cases} 1, & \text{if physician } i \text{ is working on day } j-2, \\ & \text{if he is completing a night shift on} \\ & \text{day } j-1, \text{ and if he is not assigned} \\ & \text{to a night shift on day } j, \\ 0, & \text{otherwise.} \end{cases}$$

Then, the rule is verified by adding the constraints:

$$s_{i(j-1)}^1 + \sum_{k \in K} x_{i(j-3)k} + \sum_{k \in K_N} x_{i(j-2)k} \\ + \sum_{k \notin K_N} x_{i(j-1)k} + \sum_{k \in K} x_{ijk} \leq 3, \quad i \in I, j \in J.$$

7. If a physician completes a night shift on three consecutive days, then he must be off for the next three consecutive days. We must introduce a second type of succession variables  $s_{ij}^2$  specified through the following constraints:

$$s_{ij}^2 - \sum_{k \in K_N} x_{i(j-3)k} \leq 0, \quad i \in I, j \in J, \\ s_{ij}^2 - \sum_{k \in K_N} x_{i(j-2)k} \leq 0, \quad i \in I, j \in J, \\ s_{ij}^2 - \sum_{k \in K_N} x_{i(j-1)k} \leq 0, \quad i \in I, j \in J, \\ s_{ij}^2 - \sum_{k \in K_N} \{x_{i(j-3)k} + x_{i(j-2)k} + x_{i(j-1)k}\} \\ + \sum_{k \in K_N} x_{ijk} \geq -2, \quad i \in I, j \in J, \\ s_{ij}^2 + \sum_{k \in K_N} x_{ijk} \leq 1, \quad i \in I, j \in J.$$

It is easy to verify that

$$s_{ij}^2 = \begin{cases} 1, & \text{if physician } i \text{ is assigned to a night shift} \\ & \text{on days } j-3, j-2, j-1, \text{ but not on} \\ & \text{day } j, \\ 0, & \text{otherwise.} \end{cases}$$

Then, the rule is verified by adding the constraints:

$$s_{i(j-2)}^1 + s_{i(j-2)}^2 \\ + \sum_{k \in K_N} \{x_{i(j-5)k} + x_{i(j-4)k} + x_{i(j-3)k}\} \\ + \sum_{k \in K} \{x_{i(j-2)k} + x_{i(j-1)k} + x_{ijk}\} \leq 5, \\ i \in I, j \in J.$$

Note that the last two sets of constraints, in conjunction with compulsory constraints 4, imply the rule: “after  $n \leq 3$  consecutive night shifts, any physician should have at least  $n$  days off”.

We have modeled a number of other ergonomic constraints, using similar arguments (see [2] for further details). These constraints include: “after  $n \leq 3$  consecutive night shifts, any physician should have at least 14 days

without night shifts”; “after coming back from vacation, any physician should have at least two days without night or trauma shifts”; “in a sequence of four consecutive days, there should be no more than three consecutive evenings”.

#### 4.2.3. Distribution constraints

All distribution constraints take the form:

$$\sum_{j \in \tilde{J}} \sum_{k \in \tilde{K}} x_{ijk} \geq (\leq) F_i(\tilde{J}, \tilde{K}), \quad i \in \tilde{I},$$

where  $F_i(\tilde{J}, \tilde{K})$  is a minimal (maximal) frequency for any subset of shift types  $\tilde{K} \subseteq K$  and for any subset of days  $\tilde{J} \subseteq J$ , that can be specified by any physician  $i \in \tilde{I} \subseteq I$ . Some of these distribution constraints take seniority into account, as is the case for weekend shifts: “young” physicians (with less than three years of experience) should work during two weekends every month (if possible, these weekends should not be consecutive), while other physicians should work during at least one weekend per month (the remaining weekend shifts are assigned evenly among “old” physicians). Night shifts also obey seniority rules: physicians with more than four years of experience are assigned approximately 10 night shifts for the next six months, those with less than four but more than three years of experience are assigned 12 night shifts, etc. . . . , up to a maximum of 18 night shifts assigned to physicians with less than one year of experience (with the exception of newcomers, who are not assigned any night shift).

#### 4.2.4. Goal constraints

1. A physician should work a specified number of hours per week:

$$\sum_{j \in J(l)} \sum_{k \in K} h_k x_{ijk} + uw_{il} - vw_{il} = TW_i, \\ i \in I, l \in \{1, 2, \dots, n/7\},$$

where  $TW_i$  is the target number of weekly hours specified for physician  $i$ ,  $J(l)$  is the subset of days in week  $l$ ,  $uw_{il}$  and  $vw_{il}$  are the deviation variables.

2. Certain types of shifts (night, evening and follow-up shifts) must be fairly distributed among physicians:

$$\sum_{j \in \tilde{J}} x_{ijk} + uk_{ik} - vk_{ik} = TK_{ik}, \quad i \in I, k \in \tilde{K},$$

where  $TK_{ik}$  is the target number of shifts of type  $k \in \tilde{K} \subseteq K$  required by physician  $i$ ,  $uk_{ik}$  and  $vk_{ik}$  are the deviation variables. Note that  $TK_{ik}$  can be established by taking into account the assignments of type  $k$  shifts during the previous planning periods (for example, in our application, we use a planning period of 28 days, but take into account the last five periods). In this way, we can allow for some degree of unfairness for a specific physician during a specific planning period, but still distribute the shifts fairly among physicians over an horizon including several planning periods.

3. Two “antagonist” types of shifts (days versus evenings, regular versus trauma) must be fairly distributed among physicians:

$$\begin{aligned} N_2 \sum_{j \in J} \sum_{k \in K_1} x_{ijk} - N_1 \sum_{j \in J} \sum_{k \in K_2} x_{ijk} + ur_i - vr_i \\ = N_1 P_{2i} - N_2 P_{1i}, \quad i \in I, \end{aligned}$$

where  $K_1$  and  $K_2$  are subsets of  $K$  representing “antagonist” types of shifts,  $N_1$  and  $N_2$  are the number of shifts of types  $K_1$  and  $K_2$  to be completed over each planning period,  $P_{1i}$  and  $P_{2i}$  are the numbers of shifts of types  $K_1$  and  $K_2$  completed by physician  $i$  during the previous planning periods,  $ur_i$  and  $vr_i$  are the deviation variables.

## 5. Implementation and results

The deviations can be ordered according to their relative importance (for example, any physician would complain if too many night shifts are assigned to him, but probably not if he works on too many follow-up shifts). In this case, theoretical results (which are summarized in [2]) allow us to reformulate the model as a single-objective optimization model, which seeks to minimize a weighted sum of all deviations. Indeed, if weights are properly adjusted to reflect the relative importance of the deviations, an optimal solution to the single-objective model is guaranteed to be an efficient solution to the multi-objective one. The resulting model can in theory be solved using a *branch-and-bound method*, a general technique implemented in commercial software packages dealing with linear integer optimization models. However, given the current limitations of this method and the dimension of our model (we comment on this issue below), finding an optimal solution is impractical. The branch-and-bound method nevertheless allows us to quickly identify effective feasible solutions (the effectiveness of a solution, i.e., its “distance” to an optimal solution, can be estimated by the method).

Because of its dimension (approximately 40,000 variables and 75,000 constraints), the model formulated over six months becomes intractable. Therefore, we rather solve six four-week horizon models associated with the six consecutive four-week periods. A planning horizon of four weeks was selected because using smaller time horizons, it would be difficult to take into account the constraints limiting the number of hours or shifts of certain types worked every month. Note that we do not solve four-week models independently of each other; we rather take into account the schedules already established in the last five months to construct a schedule for the current period. This is reflected in the determination of the targets in the goal constraints.

In our first attempts to construct schedules with our methodology, the models included all possible rules. However, it was quickly realized that no feasible solution can usually be found using the branch-and-bound method. This

is not surprising given the conflicting nature of most ergonomic rules, as already experienced by the planner. To deal with some of these feasibility issues, we used heuristic procedures that modify the right-hand side of some selected constraints to make them more flexible. But even then, a large set of constraints remains questionable. After extensive experiments, we have identified the constraints that are most often in conflict with each other. First, we simply eliminated them from the model. Unfortunately, the schedule then identified by the branch-and-bound method is of poor quality. However, it can be useful as a starting schedule for an *iterative approach* which proceeds as follows: (1) identify the rules that are violated in the current schedule; (2) add the corresponding constraints to the model; (3) use the branch-and-bound method to identify a new schedule, which hopefully improves over the previous one (e.g., satisfies more rules). This relaxation process is repeated until the branch-and-bound method cannot identify any feasible schedule. We have adopted this approach, which is quite effective, since it gradually reduces the number of violated rules. It is also efficient, since usually only two or three iterations (or calls to the branch-and-bound method) are required.

We had to code two programs to realize a computer implementation of our approach. The first one generates the model in a format accessible to the branch-and-bound software [7] by reading data from an *input file* that includes all information necessary for the creation of the model, such as the composition and seniority of the working staff or data extracted from previous schedules. The second program reads the solution identified by the branch-and-bound method from a file, translates it into a “readable” format for the planner by creating an *output file*, and it also identifies the violations of the ergonomic rules for their use in the iterative approach (the corresponding constraints are then generated by the first program).

The planner has made available to us the schedule that he has produced for the six-month period starting January 5, 1998 and ending June 21, 1998. We used it as a testbed for our method, which was run on a PC 486, operating at 66 MHz and equipped with a memory of 16 MB. Note that this is an obsolete technology, since today’s microcomputers (Pentiums) can operate at 500 MHz. Despite this handicap, we could generate a complete schedule in less than three days of computer time. With up-to-date technology it would require less than one day. Therefore, our approach significantly saves time and effort for the planner.

We evaluate the quality of the respective schedules according to two criteria. The first is the number of ergonomic rules violated over the six-month planning horizon, the second is the deviations from the targets.

The first criterion clearly favors our method. Indeed, over the whole planning horizon, the planner’s schedule exhibits 185 violations of the ergonomic rules (this is very good, as it represents less than 1% of the total number of rules). Our method generates a schedule with only 111 violations. The rules most frequently violated in the planner’s

schedule are related to successive night shifts; namely, “after  $n$  consecutive night shifts, any physician should have at least  $n$  days off” (violated 19 times) and “after  $n$  consecutive night shifts, any physician should have at least 14 days without night shifts”. These two rules are not violated in our schedule. Sometimes, the limit on monthly hours is exceeded in the planner’s schedule; this is the case in May and June, when many physicians are on vacation (and the remaining ones are then assigned more shifts than usual). Despite this lack of available physicians, our schedule does not violate these rules.

With respect to the second criterion, our approach performs as well as the planner, who accomplishes a wonderful job in distributing the shifts among physicians (thanks to the help of the spreadsheet). Indeed, with regard to the most important deviations, namely those related to night shifts, to weekly hours and to trauma shifts, in that order, we observe that our schedule is better than the planner’s schedule as far as weekly hours and trauma shifts are concerned (only slightly in the latter case). However, in our schedule, the average deviation (over each physician and each four-week period) for the night shifts is slightly worse, even though one physician less on average is affected by a deviation of this type. The following two observations might explain these results. First, recall that in our approach, six four-week models are sequentially solved where the targets for each period are adjusted according to the schedules for the previous periods. Now, in our test, the adjustments are made according to the planner’s schedule, which is not always well balanced. Hence this induces an impact on the result of our approach, especially during the first four-week period. Second, sometimes a violation of an ergonomic rule might help to achieve the targets. For example, when the rule “after  $n$  consecutive night shifts, any physician should have at least  $n$  days off” is violated, this might help to achieve the weekly hours target. Nevertheless, even if it violates fewer ergonomic rules, our schedule is as well balanced as the planner’s schedule.

## 6. Conclusion

We presented a mathematical programming approach for scheduling physicians in the emergency room. To the best of our knowledge, this is the first application of such an approach to this type of problem (although it has been extensively used to schedule other health care personnel, more specifically nurses). We studied the case of the Sacré-Coeur Hospital of Montréal, which is very representative of the situation in major hospitals in the Montréal area [13]. We described a model based on multi-objective integer programming theory, and a solution method which makes use of a commercial branch-and-bound solver. We compared the schedules that it produces with those generated by a human expert, who typically requires a whole dedicated week to construct a six-month schedule. Such a schedule can be generated in less than one day with our implementation (using today’s technology). Moreover, it is significantly better

than the one produced by the expert, because our approach can take into account simultaneously more rules than any human expert can do.

Because the particular case studied is representative of the situation in other major hospitals, our approach would require small adaptations to be implemented for their emergency rooms. In particular, one could embed the implementation of the solution method within a spreadsheet interface that would be easy to use and reliable [3]. Further developments made in this direction will soon be reported.

## Acknowledgements

We are very grateful to Dr Martine Langlois, Dr Éric Leroux and Mrs Linda Bélanger, from the emergency room of the Sacré-Coeur Hospital of Montréal, for their support and the time they have so graciously shared with us. Financial support for this project was provided by NSERC (Canada), the Fonds FCAR (Québec) and FRSQ. This support is gratefully acknowledged.

## References

- [1] American College of Emergency Physicians, Directory of software in emergency medicine, Dallas (1998) (<http://www.acep.org>).
- [2] H. Beaulieu, Planification de l’horaire des médecins dans une salle d’urgence, M.Sc. Thesis, Département d’informatique et de recherche opérationnelle, Université de Montréal (1998).
- [3] H. Beaulieu, J.A. Ferland, B. Gendron and L. Lefebvre, A computer system based on optimization for scheduling physicians in the emergency room, Publication #1127, Département d’informatique et de recherche opérationnelle, Université de Montréal (1998).
- [4] I. Berrada, Planification d’horaires du personnel infirmier dans un établissement hospitalier, Ph.D. Dissertation, Département d’informatique et de recherche opérationnelle, Université de Montréal (1993).
- [5] I. Berrada, J.A. Ferland and P. Michelon, A multi-objective approach to nurse scheduling with both hard and soft constraints, Socio-Economic Planning Science 30 (1996) 183–193.
- [6] ByteBloc Software, *EPSKED 3.0*, ByteBloc Software, LongBeach (1995) (<http://www.bytebloc.com>).
- [7] CPLEX Optimization, *CPLEX 4.0*, CPLEX Optimization, Incline Village (1994).
- [8] É. Gagné, Application d’une méthode exacte pour la génération d’horaires en soins infirmiers, M.Sc. Thesis, Département d’informatique et de recherche opérationnelle, Université de Montréal (1996).
- [9] R. Hung, Hospital nurse scheduling, Journal of Nursing Administration 25 (1995) 21–23.
- [10] B. Jaumard, F. Semet and T. Vovor, A generalized linear programming model for nurse scheduling, Publication G-96-35, Groupe d’études et de recherche en analyse des décisions, École des Hautes Études Commerciales, Montréal (1997).
- [11] S. Labbé, La confection automatisée d’horaires pour les médecins en salle d’urgence, M.Sc. Thesis, École des Hautes Études Commerciales, Montréal (1998).
- [12] S. Labbé, M. Gendreau, S. Lapierre and P. Soriano, A tabu search for scheduling physicians, Presented at *INFORMS*, Montréal, April 26–29, 1998.
- [13] S.D. Lapierre and M.W. Carter, Scheduling emergency room physicians, Publication CRT-99-23, Centre for Research on Transportation, Université de Montréal (1999).

- [14] H.E. Miller, W.P. Pierskalla and G.J. Rath, Nurse scheduling using mathematical programming, *Operations Research* 24 (1976) 857–870.
- [15] MSI Software, *Physician Scheduler 3.2*, MSI Software, Fairfax (1998) (<http://www.mssoftware.com>).
- [16] I. Nabli, Horaires du personnel infirmier générés avec approches heuristiques, M.Sc. Thesis, Département d’informatique et de recherche opérationnelle, Université de Montréal (1995).
- [17] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization* (Wiley, New York, 1988).
- [18] W.P. Pierskalla and D.J. Brailer, Applications of operations research in health care delivery, in: *Handbooks in Operations Research and Management Science*, Vol. 6, eds. S.M. Pollock et al. (North-Holland, 1994).
- [19] R.L. Rardin, *Optimization in Operations Research* (Prentice-Hall, NJ, 1998).
- [20] L.D. Smith and A. Wiggins, A computer based scheduling system, *Computers and Operations Research* 4 (1977) 195–212.
- [21] G. Vassilacopoulos, Allocating doctors to shifts in an accident and emergency department, *Journal of the Operational Research Society* 36 (1985) 517–523.
- [22] S. Villeneuve, Confection d’horaires en soins infirmiers pour les infirmières des équipes volantes-secteur, M.Sc. Thesis, Département d’informatique et de recherche opérationnelle, Université de Montréal (1997).
- [23] D.M. Warner, Scheduling nursing personnel according to nursing preference: a mathematical programming approach, *Operations Research* 24 (1976) 842–856.
- [24] G. Weil, K. Heus, P. François and M. Poujade, Constraint programming for nurse scheduling, *Engineering in Medicine and Biology* 14 (1995) 417–422.