Physician Scheduling in Emergency Rooms

Michel Gendreau[1,2], Jacques Ferland[1,2] Bernard Gendron[1,2], Noureddine Hail[1],
Brigitte Jaumard[1,3], Sophie Lapierre[1,4], Gilles Pesant[1,4], and Patrick Soriano[1,5]

[1] Interuniversity Centre for Research on Enterprise Networks,
Logistics and Transportation (CIRRELT)
Université de Montréal, C.P. 6128, succ. Centre-ville, Montreal, Canada, H3C 3J7
`michelg@crt.umontreal.ca`
[2] Département d'informatique et de recherche opérationnelle, Université de Montréal
[3] Concordia Institute for Information Systems Engineering, Concordia University
[4] Département de mathématiques et génie industriel
École Polytechnique de Montréal
[5] Service d'enseignement des méthodes quantitatives de gestion, HEC Montréal

Abstract. We discuss the problem of constructing physician schedules
in emergency rooms. Starting from practical instances encountered in
six different hospitals of the Montreal (Canada) area, we first we pro-
pose generic forms for the constraints encountered in this context. We
then review several possible solution techniques that can be applied to
physician scheduling problems, namely tabu search, column generation,
mathematical programming and constraint programming, and examine
their suitability for application depending on the specifics of the situa-
tion at hand. We conclude by discussing the problems encountered when
trying to perform computational comparisons of solution techniques on
the basis of implementations in different practical settings.

1   Introduction

Constructing schedules (rosters) is not an easy task to accomplish in settings
where work must be performed 24 hours per day and 7 days a week, such as
in police and fire departements, or in emergency rooms of hospitals. The prob-
lem that one is faced with is to generate "good schedules" that satisfy many
complicated rules, including ergonomic rules as defined by Knaunth [20, 19]. As
mentioned by Carter and Lapierre [11], ergonomic constraints are very impor-
tant in order to manage the circadian rhythm of the staff and it is critical to
take them into account when building schedules.

   In this paper, we focus on the problem of the scheduling of physicians in
emergency rooms (ER) in health care institutions where work is continuous. It
is known that ER are a very stressful place for physicians, but it is also great
challenge for them to work in such a place. According to Lloyd *et al.* [23], 24.5% of
physicians in Canadian ER are not satisfied with their jobs. Consequently, mak-
ing a "good" schedule for physicians in ER is very important. A good schedule
for a physician is a schedule that satisfies a large number of the requests he or

she may have regarding different issues: total amount of work to be performed, specific timing of shifts, sequencing of shifts, etc.

As already mentioned, building such schedules is quite difficult and it may take up to several weeks for a human expert to generate an acceptable solution [3]. In order to reduce time and efforts, an automated approach is therefore imperative.

Besides the biological and psychological effects involved in the scheduling of physicians, one must also pay careful attention to the fairness of the schedules among physicians. This important aspect is unfortunately very difficult to address because there are usually many individual requests and several of them turn out to be conflicting.

In this paper, we give an overview of the typical constraints that may be encountered in physician scheduling by building on the lessons learned from five practical cases encountered in hospitals of the Montreal (Canada) area: Jewish General Hospital (JGH), Charles-Lemoyne Hospital (CLH), Santa-Cabrini Hospital (SCH), Sacré-Coeur Hospital (SaCH), and Côte-Des-Neiges Hospital (CNH). An important purpose of the paper is to formalize the specific constraints of these five settings into "generic constraints" that could be used to describe problems in other practical contexts. We also review major approaches for solving the problem: mathematical programming, tabu search, constraint programming and column generation.

The remainder of this paper is organized as follows. In Section 2, we define more precisely the problem of scheduling physicians in ER and review the relevant literature. In Section 3, we propose the generic constraints that capture the essence of the various constraints encountered in the five physician scheduling case studies. Section 4 is devoted to solution approaches. Finally, we conclude in Section 5.

2   Problem Definition and Literature Review

In the health care area, there are two important types of scheduling problems that involve medical staff: nurse scheduling problems and physician scheduling problems. In the first category of problems, nurses work under collective agreement while in the second category, there are no such rules for physicians. Moreover, in the nurse staff problem, one has to maximize their individual satisfaction and minimize the cost of salaries, whereas in the physician staff problem, one only cares about the maximization of their individual satisfaction. Despite these differences between nurse and physician problems, their mathematical formulation are not quite different. Indeed, according to Gendreau *et al.* [26], a pure mathematical approach given by Berrada *et al.* in [5, 4, 34] for the nurse scheduling problem can successfully be applied to the physician scheduling problem.

The physician scheduling problem can be described as the preparation of a rostering for physicians for a given planning period, such that every shift of every day must be assigned to exactly one physician. To achieve this goal, we have to deal with some rules that are divided into two categories : compulsory (or

hard) rules and flexible (or soft) ones. These rules are often in conflict with one another, therefore some of them have to be violated in order to have a complete schedule for all physicians. Carter and Lapierre [11] note in their investigation that some flexible rules in some hospitals might be compulsory in others and vice versa. This classification depends in general on the preferences of the hospital and on the physicians' flexibility.

The set of shifts that must be covered is specified for each day of the week. In many situations, the weekend shifts are quite different from week days shifts. In general, we have three kinds of shifts: days , evenings, and nights. A week usually begins on Monday, by the first day shift and ends Sunday with the last night shift. The planning period can be quite long (up to 6 month) or fairly short (between 2 and 4 weeks). The physicians who work in emergency rooms are divided into two categories: full-time doctors and part-time doctors. A full-time doctor works an average of 28 hours per week, part-time physician works on average between 8 and 16 hours.

The physician scheduling problem can be summarized as follows: given a set of doctors, a set of shifts and a planning period, one seeks to find fair schedules for all physicians in order to maximize their individual satisfaction.

As we have mentioned above, this problem has not received very much attention. There are, however, some software packages that have been used successfully in this context [11]:

- *Tangier Emergency Physician Scheduling Software*, by Peake Software laboratories [30];
- *Epsked*, by ByteBloc Medical Software [9];
- *Docs for Windows*, by Acme Express [1];
- *Physician Scheduler 4.0*, by Sana-Med.

These software packages have been sold to emergency departements in thousands of copies, but the research community did not benefit from the fundamental work that led to these products. The only academic works that we are aware of are some works on cyclic rostering [8, 21] and some on acyclic rostering [2, 3, 8, 10, 11, 14, 26, 31]. The solution methods developed in these references will be examined more closely in Section 4.

3   Physician Scheduling Problem Constraints

In this section, we propose generic forms for the constraints encountered in the five case studies mentioned in the introduction. As we have already mentioned, in the physician scheduling problem, we have to find a roster for every physician such that a large number of constraints are satisfied. Some constraints are applied for every physician and others only for some physicians. There are two types of constraints: hard and soft. A constraint is called *hard* if it must be satisfied; it is called *soft* if it can be violated. In this study, we have classified the constraints of the physician scheduling problem into four categories:

1. Supply and Demand Constraints

2. Workload Constraints
3. Fairness Constraints
4. Ergonomic Constraints

The first category of constraints deals with the availabilities of the physicians and the requirements of the emergency rooms that must be opened every day and 24 hours a day. The second category deals with the workload (number of hours or number of shifts) that is assigned to physicians during a week, a given period or the whole planning period. The third category controls the distribution of different kinds of shifts during the whole planning period. The fourth category of constraints covers various rules ensuring a certain level of quality for the schedules produced.

3.1   Supply and Demand Constraints

Two kinds of constraints are encountered in all physician scheduling problems. First, a sufficient number and variety of shifts must be staffed throughout the scheduling horizon in order to guarantee minimum coverage. Second, a given physician, according to his seniority, full/part time status, outside responsibilities, and planned vacations, is not available at all times.

Constraint 1 (Demand) *During the overvall planning period, every shift must be performed by exactly one physician.*

Whereas in other contexts such as nurse scheduling, the number of staff members covering a shift must lie in a certain interval, for physician scheduling this number is almost always exactly one. This constraint is considered a hard constraint and it is encountered in all the hospitals listed in the Introduction. Carter and Lapierre[11] identify three variants of this situation, but we restrict our attention here to the two main ones.

1. *Uniform case*: the required number of physicians is the same for every day in a week, i.e., we have the same number of shifts for every weekday, even for Saturday and Sunday.
2. *Non-uniform case*: the required number of physicians is the same for every weekday expect for Saturday and Sunday. In this case, the number of physicians required on Saturday is the same as on Sunday.

Constraint 2 (Availability) *During the planning period, all the requests of every physician should be satisfied. There are four types of requests:*

1. *Preassignments,*
2. *Forbidden assignments,*
3. *Vacations,*
4. *Preferences or aversions.*

Each one of these types of requests is considered a hard constraint except for the last one, which is a soft version of the first two. That last type occurs for example in the context of religious practices at JGH: some physicians want to be off for the evening and the night shifts on Friday [8].

3.2   Workload Constraints

This category of constraints deals with the workload (number of hours or number of shifts) that is assigned to physicians during a week, a month or the whole planning period.

Constraint 3 (Limits on workload) *During a given period, a physician should be assigned an amount of work that lies within a specified interval.*

Example 1. In the SaCH case study, a physician who is supposed to work 28 hours a week could accept to work up to 32 hours.

Example 2. At JGH, at most four shifts are assigned to a physician on any given week.

This constraint is common to all the hospitals we considered. It is often specified over disjoint subsets of the planning period, either because of the terms of a contract or to encourage a uniform workload. Sometimes a target workload with the interval may be given: it can be viewed as a soft constraint. Another constraint encouraging uniform workloads is the following.

Constraint 4 (Limits on the number of shifts of the same type) *During a given period (e.g., a month), the number of shifts of the same type that are assigned to a physician cannot exceed a certain value.*

Example 3. At SacH, no physician should work more than three night shifts in a four-week period.

3.3   Fairness Constraints

This category of constraints ensures the fair distribution of different types of shifts among physicians with the same experience.

Constraint 5 (Distribution of Types of Shifts) *During the planning period, shifts of the same type (e.g., evening, night, weekend) should be distributed fairly among physicians with the same level of experience.*

Example 4. At SaCH, all physicians with more than four years of experience have to work the same number of night shifts during the planning period of six months.

Example 5. Again at SaCH, physicians should not work more than five weekend shifts in a four-week period. In this hospital, a working weekend can include up to three shifts.

3.4   Ergonomic Constraints

This is the largest and the most heterogeneous category of constraints. Various rules ensure a certain level of quality for the schedules produced and may be specified either globally for the staff or only for certain individuals. In his work on ergonomics, Knauth [20, 19] has shown the impact of work schedules on the circadian rhythm of workers. He proposed several rules, which we summarize below:

- minimizing permanent night shifts;
- reducing the number of successive night shifts to a maximum of two or three;
- avoiding short intervals of time off (less than 11 hours) between two consecutive shifts;
- shift systems including work on weekends should provide some free weekends with at least two consecutive days off;
- long work sequences followed by four to seven days of mini-vacations should be avoided;
- forward rotations (day shifts followed by evening shifts followed by night shifts) are preferred;
- individual schedules with few changes over time are preferred;
- shift lengths should be adjusted according to task intensity;
- shorter night shifts should be considered;
- a very early start time for the morning shift should be avoided;
- preference should be given to flexible working time arrangements among workers.

The constraints below address some of these ergonomic concerns.

Constraint 6 (Length of work sequences) *The number of identical shifts (or of shifts of the same type) in a sequence of consecutive days must lie within a given interval.*

Example 6. In the work of Carter and Lapierre [11], there must be at least two and at most four consecutive identical shifts.

Example 7. At SaCH, the interval is [1, 4] for shifts in general.

Example 8. In each of the hospitals studied, the number of consecutive night shifts lies between one and three.

Example 9. AT SaCH, a physician requires at least 14 days between two night shifts belonging to different work sequences. This can be recast as a constraint on the length of sequences of non-night shifts.

Constraint 7 (Patterns of Shifts) *Over a given number of consecutive days, a set of patterns of shifts describes what a physician is allowed to do or not to do.*

Example 10. There must be a minimum number of hours of rest between two consecutive shifts. Consequently, certain patterns of shifts over two consecutive days are forbidden.

Example 11. At SaCH, a set of restrictive patterns govern weekend work. For instance, a physician working the 8 AM regular shift on Saturday must also cover the 10 AM trauma shift on Sunday; working the 4 PM regular shift on Friday requires working the 4 PM trauma shift on Saturday and the 4 PM regular shift on Sunday as well.

Example 12. A physician should work at most one night shift in every sequence of three consecutive work shifts.

Example 13. A physician should not work a non-homogeneous sequence of four consecutive work shifts.

Constraint 8 (Patterns of Sequences of Shifts) *This is similar to the previous constraint, except that patterns are expressed not over a fixed number of consecutive days, but rather over a fixed number of sequences of consecutive work shifts.*

Example 14. At JGH, every two consecutive sequences of work shifts should satisfy the forward rotation principle.

Constraint 9 (Patterns of Sequences of a Given Length) *Patterns are expressed over both the type and the length of sequences.*

This has the flavour of the previous constraint and of the first ergonomic constraint.

Example 15. After coming back from a vacation, no physician should work a night shift for the first two days.

Example 16. At SaCH, there must at least three days off after a sequence of three night shifts.

Table 1 presents a summary of these generic constraints.

4   Four Optimization Techniques for the Physician Scheduling Problem

In this section, we present general descriptions of four solution techniques for the physician scheduling problem. These methods are completely different from one another, as we shall see later:

1. Mathematical programming
2. Column generation
3. Tabu search
4. Constraint programming

Table 1. Generic constraints in the five hospitals studied

Constraints	CNH	CLH	JGH	SaCH	SCH
Demand	X	X	X	X	X
Availability	X	X	X	X	X
Limits on workload	X	X	X	X	X
Limits on shifts of the same type		X	X	X	X
Distribution of types of shifts	X	X	X	X	X
Length of work sequences	X	X	X	X	X
Pattern of shifts	X	X	X	X	X
Pattern of sequences of shifts			X	X	
Pattern of sequences of given length				X	

4.1   Mathematical Programming

Beaulieu *et al.* [3] have proposed a mixed 0-1 programming formulation of the physician scheduling problem where the objective function is the sum of penalties associated to some constraints, called deviation constraints. This formulation was also used by Forget[14] in the context of Santa-Cabrini Hospital (SCH). In these case studies, constraints are classified in three categories: ergonomic constraints, distribution constraints and deviation constraints. After obtaining the mathematical formulation of problem under study, Beaulieu *et al.* [3] first considered using branch-and-bound on this formulation to find a solution, but this approach had to be dropped, unfortunately, due to the huge dimension (large number of variables and constraints) of some instances. The solution technique that was applied is a heuristic approach based on a partial branch-and-bound, instead of a complete branch-and-bound, which requires more computational time. Moreover, branch-and-bound was not applied to the original formulation, but to a modified one. Indeed, as mentioned by Beaulieu *et al.*, it was quickly realized that there was no feasible solution to the original formulation. This was due to the presence of some ergonomic constraints that were conflicting and led to an infeasible problem. The solution technique proposed by the authors is to solve the model with a subset of constraints which contains all hard constraints and some soft constraints that are not in conflict with each other. Afterwards, they modified some of the soft constraints and introduced them one by one in an iterative process, which can be summarized as follows [3]:

- Identify the rules that are violated in the current schedule.
- Add the corresponding constraints to the model.
- Use the branch-and-bound method to identify a new schedule, which hopefully improves over the previous one(e.g., satisfies more rules).

This process is repeated until the branch-and-bound cannot find any feasible schedule.

4.2   Column generation

The column generation technique [12, 25] is an exact method that relies on the decomposition principles of mathematical programming; it is usually used to solve large and complex problems, such as the cutting stock problem. This method was successfully applied to solve the nurse scheduling problem and a software called IRIS was produced [22]. In the column generation method, each new column is generated by solving an auxiliary problem (or subproblem). For instance, in the cutting stock problem, a knapsack problem is solved to find a new cutting pattern for rolls. In the nurse scheduling problem, a new column is obtained by solving a shortest path problem with ressource constraints on a directed graph [32]. The ressources correspond to the following constraints:

- The constraint dealing with the workload of every nurse for a given period (e.g., 2 weeks);
- The constraint that controls the vacation periods of every nurse;
- The constraint that deals with the succession of shifts of the same type;
- The constraint that is associated with the distribution of weekends.

The formulation of the master problem for the nurse scheduling problem includes the hard constraint that gives the required number of nurses for every shift of every day. Moreover, the objective function is given by the sum of penalty costs associated with the contraints not explicitly taken into account in either the auxiliary problem or the master problem.

This solution technique can be applied to the physician scheduling problem after some minor modifications. First, one can use the same auxiliary problem as for the nurse scheduling problem. Indeed, the constraints that define the ressources are also present in the physician scheduling problem. Second, the constraint dealing with the requirements (number of nurses per shift), which is used in the master problem for the nurse scheduling problem, is also present in the physician scheduling problem (one physician for every shift). One then simply has to modify the formulation of the objective function and define in it penalty costs for the remainder of the constraints that one wishes to consider.

4.3   Tabu Search

Tabu search is one of the most effective solution techniques for solving hard combinatorial problems. Originally proposed by Glover [18], it has been succesfully applied to a wide variety of application contexts, such as vehicle routing [16], machine scheduling [28], maximum clique problem [17], quadratic assignement problem [27, 29]. This method has also been applied to the nurse scheduling problem[7, 13], as well as the physician scheduling problem. In the case of physician staff, the solution technique was used to generate two kinds of schedules: cyclic schedules [21] and acyclic schedules [8].

Generally speaking, tabu search is a local search (LS) technique, i.e., an iterative search procedure that, starting from an initial feasible solution, progressively improves it by applying a series of local modifications. The key ingredient of any

LS technique is the set of modifications (or *moves*) that it considers: the richer this set, the better the solutions that one can expect to obtain, but also the slower the method. While classical LS methods stop when they encounter a local optimum w.r.t. to the modifications they allow, tabu search continues moving to the best non-improving solution it can find. Cycling is prevented through the use of short-term memory structures called *tabu lists* (see [15] for a comprehensive introduction to the topic).

Buzon's tabu search method for acyclic schedules [8] is in fact an extension and a generalization of previous work by Labbé [21]. In this approach, a solution S corresponds to a set of schedules: one for each physician. The solutions examined by the search have the property that they satisfy the demand constraints, i.e., all shifts are covered, but other constraints may be violated. The cost $c(S)$ of solution S is the sum of the costs of all schedules in S. If there are n physicians, then the cost of a solution S is $\sum_{p=1}^{n} cost(Schedule_p)$, where $cost(Schedule_p)$ is the cost of the schedule for physician p. The cost of a physician schedule is also the sum of all penalties that are associated with the unsatisfied constraints. There is exactly one penalty for each constraint. For example, suppose that physician p wants to work only 2 unbroken weekends. If the schedule associated with this physician in the current solution contains 3 unbroken weekends and 1 broken weekend, then the penalty associated with the weekend constraint would be $(3\text{-}2).P_{NBW} + 1.P_{BW}$, where P_{NBW} (respectively P_{BW}) is a certain value associated with one extra unbroken (respectively broken) weekend. Proper values for these penalty weights are not easy to determine; unfortunately, the quality of the solution that one can expect to find is quite sensitive to them [8].

Buzon's method considers several different types of modifications to solutions (neighborhoods) of increasing complexity. The simplest one involves simply re-assigning a shift on one day to a physician currently off on that day. More complex neighborhoods involve swapping portions of schedules between two physicians. See [8] for further details.

4.4   Constraint programming

Constraint programming is a solution technique that is more and more applied to various optimization and combinatorial problems. Its application to complex problems like work schedules [24] is possible for each problem in which the set of values (*domain*) of every variable is finite. The domain of each variable is saved and updated during the progression of calculations by using the constraints that involve this variable and others whose domain has been modified. These constraints take part in the elimination of all the inconsistent values of a variable from its domain; this is done by using some techniques called *filtering algorithms*. This means that all infeasible solutions are removed and only feasible solutions are effectively considered.

This method was applied for the physician scheduling problem by Cangini [10], Rousseau *et al.* [26], Trilling [31] and Bourdais *et al.* [6]. The work of Rousseau *et al.* [26] is about using constraint programming to define a general algorithm

that takes into account two types of generic constraints: pattern and distribution constraints. We will not give more details about this general method, the interested reader is referred to [26].

This algorithm was successfully applied to two hospitals: SCH and CNH. The physician scheduling problem that is solved in [26] is formulated as follows:

Minimize $f(W)$

subject to $W_{ds} \in A_{ds}$
Distribution constraints
Pattern constraints

The set A_{ds} contains the physicians who can work shift s of day d. The variable W_{ds} represents the physician who will be on duty on shift s of day d. As for the methods presented earlier in this section, the formulation of objective function f is the most difficult part of the solution scheme. In this case, $f(W)$ represents the "cost" associated to the schedules that are generated for all physicians (one schedule for each physician). The cost of the schedule for a given physician p is the sum of the penalties associated with each constraint.

5   Conclusion

The physician scheduling problem is a challenging one. While we have proposed a series of generic constraints to describe it, it must be understood that the specific constraints that are in force in any given case study may vary wildly. This makes it difficult to come up with solution methods that can be used in a wide range of practical settings. It also greatly complicates the task of coming up with fair comparisons of different methods, since they may have been developed for settings that are quite different in nature. We have indeed attempted to compare the four approaches described in the previous section and found out that just creating a set of benchmark instances that would allow such a comparison was in itself a very challenging task. We hope to be able to report on this comparison at a later date.

References

1. Acme-Express: Medical   staff   and   physician   scheduling   software. http://www.docs2000.net/productdetailsy2k.asp, 2000
2. Beaulieu, H.: Planification de l'horaire des médecins dans une salle d'urgence. Master's thesis, Département d'informatique et de recherche opérationnelle, Université de Montréal, Canada, 1998
3. Beaulieu, H., Ferland, J.A., Gendron B., Michelon, P.: A mathematical programming approach for scheduling physicians in the emergency room. Health Care Management Science **3** (2000) 139–200
4. Berrada, I.: Planification d'horaires du personnel infirmier dans un établissement hospitalier. Ph.D. dissertation, Département d'informatique et de recherche opérationnelle, Université de Montréal, Canada, 1993

5. Berrada, I., Ferland, J.A., Michelon, P.: A multi-objective approach to nurse scheduling with both hard and soft constraints. Socio-Economic Planning Sciences **30** (1996) 183–193

6. Bourdais, S., Galinier, P., Pesant, G.: HIBISCUS: A Constraint Programming Application to Staff Scheduling in Health Care. Principles and Practice of Constraint Programming: Proceedings of the Ninth International Conference (CP'03), Kinsale (IE), September 2003, Springer-Verlag Lecture Notes in Computer Science **2833** (2003) 153–167

7. Burke, P., De Causmaecker, P., Vanden Bergue, G.: A hybrid tabu search algorithm for the nurse rostering problem. Lecture Notes in Computer Science **1585** (1999) 187–194

8. Buzon, I.: La confection des horaires de travail des médecins dans une salle d'urgence résolue à l'aide de la méthode tabou. Master's thesis, École Polytechnique, Montréal, Canada, 2001

9. ByteBloc Software. Epsked 3.0 bytebloc software. *http://www.bytebloc.com*, 1995

10. Cangini, G.: A constraint programming local search algorithm for physician scheduling. Centre for Research on Transportation, Université de Montréal, Publication CRT-2000-26, 2000

11. Carter, M.W., Lapierre, S.D.: Scheduling emergency room physicians. Health Care Management Science **4** (2001) 347–360

12. Chvàtal, V.: Linear Programming. Freeman, 1983.

13. Downsland, K.A.: Nurse scheduling with tabu search and strategic oscillation. European Journal of Operation Research **106** (1998) 393–407

14. Forget, F.: Confection automatisée des horaires des médecins dans une salle d'urgence. Master's thesis, Département d'informatique et de recherche opérationnelle, Université de Montréal, Canada, 2003

15. Gendreau, M.: An Introduction to Tabu Search. In Handbook of Metaheuristics, F.W. Glover and G.A. Kochenberger (eds.), (2003) Kluwer Academic Publishers, Boston, MA, 37–54

16. Gendreau, M., Hertz, A., Laporte, G.: A tabu search algorithm for the vehicle routing problem. Management Science **40** (1994) 1276–1290

17. Gendreau, M., Soriano, P.,Salvail, L.: Solving the Maximum Clique Problem Using a Tabu Search Approach Annals of Operations Research **41** (1993) 385–403.

18. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers and Operations Research **13** (1986) 533–549

19. Knaunth, P.: Design better shift systems. Applied Ergonomics **27** (1996) 39–44

20. Knaunth, P.: The design of shift systems. Ergonomics **36** (1993) 15–28

21. Labbé, S.: La confection automatisée d'horaires pour les médecins en salles d'urgence. Master's thesis, École des Hautes Études Commerciales de Montréal, Canada, 1998

22. Labit, P.: Amélioration d'une méthode de génération de colonnes pour la confection d'horaire d'infirmières. Master's thesis, École Polytechnique, Montréal, Canada, 2000

23. Lloyd, S., Shannon, S., Steiner, D.: Burnout, depression, life and job satisfaction among Canadian emergency physicians. The Journal of Emergency Medicine **12** (1994) 559–565.

24. Marriott, K., Stuckey, P.J.: Programming with Constraints: An Introduction. MIT Press, 1998.

25. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley-Interscience, 1998

26. Rousseau, L., M., Pesant, G., Gendreau, M.: A Hybrid Algorithm to Solve a Physician Rostering Problem. In Second Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (2000), Paderborn, Germany.

27. Skorin-Kapov, J.: Tabu search applied to the quadratic assignment problem. ORSA Journal on Computing **2** (1990) 33–45

28. Taillard, É.: Some efficient heuristic methods for the flow shop sequencing problem European Journal of Operational Research **47** (1990) 65–74

29. Taillard, É.: Robust taboo search for the quadratic assignment problem Parallel Computing **17** (1991) 443–455.

30. Peake Software laboratories: Tangier emergency physician scheduling software. http://peakesoftware.com/peake/

31. Trilling, G.: Génération automatique d'horaires de médecins de garde pour l'hôpital Côte-des-Neiges de Montréal. Centre for Research on Transportation, Université de Montréal, Publication CRT-98-05, 1998.

32. Vovor, T.: Problème de chemins bicritère ou avec contraintes de ressources : algorithmes et applications. Ph.D. thesis, École Polytechnique, Montréal, 1997.

33. Weil, G., Heus, K., Poujade, P., Fran cois, M.: Constraint programming for nurse scheduling. Engineering in medecine and Biology **14** (1995) 417–422

34. Warner, D. M.: Scheduling Nursing Personnel According to Nursing Preference: A Mathematical Programming Approach. Operations Research **24** (1976) 842–856