*MEDINFO 98*
*B. Cesnik et al. (Eds)*
*Amsterdam: IOS Press*


# The Nurse Scheduling Problem: a Combinatorial Problem, Solved by the Combination of Constraint Programming and Real Users Heuristics

**Georges Weil[a], Kamel Heus[a], Peter Chan[b], Patrice François[c],**

[a] *TIMC Laboratory, Grenoble University Hospital, France*
[b] *COSYTEC, Parc Club Orsay-Université, Orsay, France*
[c] *Quality Assurance Department, Grenoble University Hospital, France*

**Abstract**

*Nurse scheduling consists of assigning shifts and rest to nurses for each day, on a time schedule, taking in account legal and collective constraints, and individual wishes. The Nurse Scheduling Problem (NSP) is a highly difficult and complicated problem that has already be explored using several operational research methods. Nevertheless, those methods are not flexible enough to match the individual requests from nursing staff and they don't allow for effective management of unforeseen absences. This paper explores the use of a Constraint Programming (CP) to build Gymnaste, which is since June 1997 a available package, after five years of a slow research and development process. We describe the way the problem is modeled, the constraints typology, and the interface. In fact, the NSP is also an ill defined problem, making the man computer interface critical to let the user chose the best heuristics. Gymnaste is currently beta-tested on several pilot sites. Schedules have been generated very quickly, typically in less than a minute for 20-30 persons over 4 weeks. An evaluation process is running, which preliminary results also show some interesting sociological and organizational aspects.*

***Keywords***

Constraint Programming; Constraint Satisfaction Problem; Nurse Scheduling; Staff Scheduling; Software Evaluation; Nursing Informatics.

## Introduction

The basic Nurse Scheduling Problem (NSP) is defined as assigning working shifts (such as Morning, Evening and Night) to each member of the nursing staff over the scheduling period of say 4 weeks, to meet total minimal skill requirements specified per day. This task is complicated by numerous laws and regulations that have emerged over the years, making it time-consuming and frustrating for the head nurse. Moreover, changes in work time organization such as partial time, chosen time, and a general trend for flexibility (i.e. adapting the number of nurses to the team activity day by day) makes it not only a complex and time consuming task, but also a milestone for hospital human resources optimization.

The NSP has important sociological aspects, since it manages 2 of the most sensitive concepts in the world of work:

- time, since time is life and money,
- human relationships in a professional team.

Introducing such a tool may lead to changes in the social game inside the team. This has three main consequences:

- the general design of the application, especially the man/computer interface, the way functionalities are ordered in the menu tree, the data and constraint model, the users rights management ;
- change management when implementing the tool in the real world.
- a negotiation inside the hospital to decide the way benefits of a human resources allocation system will be distributed towards the hospital, the teams, and the employees.

A major difficulty is to clearly identify in this problem what has to deal with combinatorial computing, and what belongs to the negotiation, under the responsibility of the head nurse. This distinction is not obvious, and it is not sure that it is stable, through the space (teams, hospitals, and countries), and time. Moreover, transparency of management rules, individual wishes and optimization could meet some psycho-sociological resistance.

From the sociological point of view, it is long suspected that there is a link between employee satisfaction (relative to their schedule) and sociological factors such as productivity, quality of work, and absenteeism [1, 2, 3].

Nevertheless, in this paper we will concentrate on the NSP as a combinatorial problem, and describe how it can be solved by the conjunction of CP and a man-computer interface designed for an ill-defined problem.

## Materials and Methods

### The Nurse Scheduling problem, an old problem

The NSP has attracted the attention of computer scientist since 1960s. It arises from the operational research domain belonging to the resources allocation, administration time-table, and planning fields. This problem has been studied using several approaches including operational research techniques and artificial intelligence [4, 5, 6, 7, 8, 9, 10, 11, 12].

Like any multi-factorial problem, it is often difficult to find a perfect mathematical model for solutions, especially when efficient use of human resources is concerned.

In the literature two types of scheduling are presented. See [13, 14] for review.

- Cyclical scheduling : each nurse work pattern is repeated over a cycle of n weeks[15]
- Non-cyclical : for each scheduling period a new schedule is generated [16]

Resolution techniques for the NSP are of two types:

- Optimizing solution techniques: they include linear programming, mathematical programming, and goal programming. These techniques are used to generate a schedule that optimizes a certain programming function (nursing satisfaction, nursing disappointments, etc.)
- Heuristic techniques : the goal of these techniques is to find a good schedule, which is not guaranteed to be the most satisfactory one [17]

There are also some methods using artificial intelligence techniques to solve the nurse scheduling problem

The two major disadvantages of the methods used to solve the NSP are the rigidity of the models in relation to the individual requests of the nursing staff (often a nurse has to choose a pattern schedule among a limited number of proposed patterns), and the difficulty of managing unforeseen absences.

**Constraint Programming Approach**

CP has only recently become available and is situated at the intersection of artificial intelligence and operational research. CP has allowed successful solution of some complex, combined problems such as: scheduling, planning, resources allocation, [18].

X = {x1,....., xn} a set of n variables of the problem,

D = {d1,....., dn} a set of n domains; each variable xi takes its value in di,

C = {c1,....., cm} a set of m constraints where ci is a constraint between (xi1,.....,xip).

Solving a CSP means finding one or all instance(s) V for each variable of X such that:

$$\forall x_i \in X,\ V(x_i) \in d_i$$

$$\forall c_i \in C,\ c_i(V(x_{i1}), \ldots\ldots\ldots, V(x_{ip}))$$ is satisfied.

CP enables dissociation of problem modeling from the algorithms used for solution.

The nurse scheduling problem can be readily be formulated as such [19]. This approach incrementally constructs solutions by assigning possible values to variables, in turn. After each assignment, the constraint propagation paradigm combines both local value propagation with demon-driven computation.

CP tools where variables are of discrete (or integral) domain, are most appropriate to modeling the NSP. A variable is created for each person-day pair, initially with possible values equal to all working affectations of the category of the person and including rest. CP tools are most situated for building interactive decision-making systems. Users can preaffect some variables, followed by running the CP solver, halting it if and when necessary, undoing or changing some CP - found values, and reiterating. This characteristic is most useful when the NSP necessarily results from negotiations between the nurses and the head nurse. CP is very convenient in modeling compulsory (or binding) constraints. It allows handling of user preferences.

***Constraint Programming with CHIP V5***

One of the earliest CP tools have been created at European Computer-Industry Research Center. CHIP (Constraint Handling In Prolog) extends the declarative aspects of Logic Programming with the efficiency of Constraint Programming and has already been applied to several real world problems with success. This research work during the period 85-90, has led to several commercial products, including Charme from Bull, SNI-Prolog from SIEMENS, DecisionPower from ICL, and CHIP V5 from COSYTEC, which is founded by the original CHIP team in 90.

**Development methods and process**

The development method look like rapid application development approach, except it was really slow, up to five years, from the initial idea to a package. This means that there was an early and permanent users commitment, reacting in front of multiple prototypes versions. It allows findings about the man-computer interface, the users heuristics, and the users criteria about what is important and what is not, what should be calculated and what should be negotiated. In the same time, there was a research to qualify CP tools for such problem, and to define a data and constraint model of the problem. This lead firstly to find a constraint typology that could be understood by the users and managed by the system, and secondly to adapt the man-computer interface in order to let the user free to use his own heuristics.

## Results

**Constraints typology**

The constraints can be classified in two ways:

- a hierarchical way

  Legal constraints are mainly work laws and regulations. They should be mandatory, but in practice they are frequently violated.

  Community constraints are not legal but they are usually applied to every one, since they have to do with equity, in example having almost the same numbers of week-end on rest

  Individual wishes are frequently negative; people better know what they dont want.

- a functional way: it is the way they are in practice expressed. In fact, there is no systematic relation between the hierarchy of the constraints and their priorities, when looking at real schedules. (And this could be a surprise with consequences for general managers of human resources in Hospitals). Hereunder we briefly describe those constraints in Backus-Naur like formalism.

*Needs constraints:*

In columns: numbers of employees per shift and per skill;

In line: number of shifts to be done per employee

***Distribution constraints***

In example, Michael should not work more than 4 consecutive nights shifts, but when working at night, he wants to have at least three consecutive nights.

[employee-1/list of employee/all] /at least [n], at max [m] [shift type] consecutive per [w] weeks from [date-begin] to [date-end].

***Disposability***

In example, Baptistine does not want to work the Wednesday afternoon, every 2 weeks. (because her children dont go to school Wednesday).

For [one or several employees of the list/ all/one category][shift type] [day type/list of day type] {always, if possible, if possible not, never} per [w] weeks from [date-begin] to [date-end].

***Cumulative constraints***

In example, Barry should at least do 5 night shifts and max 7 nightly shifts every 2 weeks.

[employee-1/list of employee/all] /at least [n], at max [m] [shift type], per [w] weeks from [date-begin] to [date-end].

***Team composition constraints:***

In example « Brunehilde should be at least 3 time with Ultrogothe per 2 weeks », or « John should never be with Childebert »

[Employee-1]{always, if possible, if possible not, never} with [employee-2] option/at least [n] times per [m] weeks from [date-begin] to [date-end].

***Shifts transition constraints:***

In example a forbidden transition like « no morning shift after a night shift », or a legal transition such as « at least two consecutive day every 2 weeks »

For [one or several employees of the list/ all/one category][shift type] {always, if possible, if possible not, never} followed by [shift code], option/at least [n] times from [date-begin] to [date-end].

**Resolution Heuristics**

***Computerized heuristics***

GYMNASTE handles the NSP in one single pass for one category at a time, i.e. without predetermination of day-on, day-off or mixing people from different categories. For a given day, the enumeration process assigns exactly what is required of each shift. Remaining variables are left free so that no useless commitment is made. Once the minimum work load is respected by an assignment, the reminding variables are filled in to fit the other constraints.

***Users heuristics***

Users will not usually handle the NSP in one single pass. There is a fist level of heuristics which is the users one. Fundamentally, the NSP is an ill defined problem [20]. Gymnaste is designed to allow users to handle the NSP in the best way, at each planification situation, which can vary in time and space. The users usual heuristic is this one:

1. Post long term constraints,
2. Post short term constraints,
3. Post punctual constraints,
4. Pre-affect some employees, especially rest shifts
5. Start the solving process,
6. Looking at the result,
7. Changing manually part of the solution, with a specialized editor, (bloc by bloc, two by two permutations, partial solutions searching, etc.)
8. Optionally, changing again some constraints and repeating the process until 7.

It is of interest to notice that this users heuristic is the result of the theoretical impossibility to define a cost function in order to select the « best » solution amongst the set of solutions. The reasons why is that:

- there are still some constraints which remain implicit, and all the solutions are not equivalent,
- the « users cost function » may vary in time.
- last but not least, it is difficult to express it mathematically. The head nurse is not able to order the coefficient and the parameters of this cost function.

**Role of the interface**

***Looking like usual paper schedules***

It mandatory to design the interface minimizing the differences with a paper schedule. But this is not obvious, because of the number of functionalities, and because the paper data model is theoretically fuzzy specified. In example, on a paper schedule, one may find remarks about someone being late one day, or a special wish, or secondary permutation, and so on. However, the general aspect of the schedule is a Gantt diagram, with time in column, and employees in line. Three main view of the same Gantt showing in an underneath window either the needs per shift, and per category, either the content of the cells, either blackboard information by employees, such as numbers of Saturday or Sunday, cumulative working time over the planed period, or since a specified date. We will not develop in this paper those ergonomical aspects of the work, but it partially explain the duration of the development process and has to deal with cognitive ergonomy.

***Innovative Features***

- Data Model Flexibility : handles any number of categories of employees, number of persons, type and segmentation of shifts, scheduling horizon,
- Preaffectations: the user can graphically preaffect any given nurse-day combination. The user may specify preferences for one or more shifts.
- Interactive resolution : manually affect some nurse-day combinations, activate the solver, stop it, validate some

solver decisions and delete others, and reiterate with manual affectations, until the final schedule is obtained.

- Constraints Flexibility : Before making a run, the user may express his scheduling constraints by a specialized constraint editor.

## Discussion

**General difficulties in specifying news tool, especially when part of the task can be computerized, and another remain manual.**

This is not exactly specific of this application and one can have the same kind of problems for electronic patient record. But in the case of the NSP, they seam to be more acute because of the importance of negotiation, and because a lack of experience in computerizing this function. This explains, partially the length of the development process.

**Showing the way to task coordination through time and space.**

The NSP belongs to the class of resources allocation problems, which are in healthcare represented by medical multiple appointments, healthcare services facilities location allocation problems, medical and nursing procedures implying task coordination through time, space, and healthcare organizations. There is here a big applied research field, with huge social, economical, and even ethical stakes.

**Optimizing human resources at the hospital level, with inter-teams interchanges.**

We modeled the NSP as a single team NSP. As a Hospital is constituted of numbers of teams, with possibility of employees interchange, it is interesting to model it in order to allow and optimize such interchange. In fact, we did it in previous prototypes, then remove it because psycho-sociological resistance and Unions opposition. Nevertheless, it should be try again after clear discussion about the distribution of the optimization benefits.

**The long march from an idea to a product : From RAD to SAD.**

Rapid Application Development (RAD), or fast prototyping is always describe as an interesting approach in those kind of case : new concepts and software, importance of users commitment, technology risks, market risks, etc. In fact, we did long term iterative RAD, arriving to a so called Slow Application Development (SAD). Nevertheless, the duration of this process looks usual to industrial research and development investors having a habit of those situations. And it is frequently the case in the field of medical informatics, when going up to the product. It is also an interesting lesson for searchers. We had some research results in term of publication after 2 years, but it takes at least 3 years more, and a lot of effort and investment to get a real product.

**Current status**

Gymnaste is currently beta-tested. An evaluation of the software and of resulting schedules is running through several dimensions such as completeness, optimality, equity, healthiness, and continuity [21]. It is not complete yet and will be published later. But, it already appears that there are big variations throughout the teams of the task of nurse scheduling. Every head nurse spends a lot of time on this task. But in some departments, it is done in one single pass plus adjustments while in other ones, it permanently done with short lasting tasks up to half a minute. In those cases, it will be hard to use a nurse scheduling software, in the current state of the technology.

## Conclusion

The NSP is a very interesting and multi-faceted problem, resuming a lot of aspects of medical informatics. It shows the way of optimization and coordination in Healthcare. The challenge is to do it in respect with medical traditional values.

## Acknowledgments

This work was supported by Medicine and Biology Engineering (GBM) of Rhône-Alpes France and by the FITT. Special thanks to Denis Leroux from the Industrial Relations Department of Grenoble I University.

## References

[1] Brya JR and Pierce FA. A Personal Allocation and Scheduling System for Patient Care Units. *The Forum on the Patient Unit - A Systems Approach*, sponsored by NSCS/HME, 1974.

[2] Bradley DJ and Martin FB. Continuous Personal Scheduling Algorithms: A Literature Review. *Journal of the Society for Health Systems*, Vol. 2, No.2, pg 8-23, 1990.

[3] Weil G, Heus K, François P, Jacques P, Poujade M. A Solver for the Nurse Scheduling Problem Using Constraint Programming : A technical and sociological challenge. In J. Brender, J.P. Christensen, J.R. Scherrer and P. McNair (Eds). *Medical Informatics Europe 96*. Amsterdam, the Netherlands: IOS Press, pp 1032-1036.

[4] Arthur JL, Ravindran A. A multiple objective nurse scheduling model. *AIIE Transactions* 13:55-60, 1981.

[5] Darmoni S, Fajner A, Mahé N, Leforestier A, Vondracek M, et al. Horoplan : computer-assisted nurse scheduling using constrained-based programming. *Journal of the Society for Health Systems*, 1994.

[6] Miller HE, Pierskalla WP, Rath GJ. Nurse scheduling using mathematical programming. *Operation Research* 24 (5) :857-870, 1976.

[7] Rosenbloom ES, Goertzen NF. Cyclic nurse scheduling. *European Journal of Operational Research* 31 (1) :19-23, 1987.

[8] Musa AA, Saxena U. Scheduling nurses using goal-programming techniques. *AIIE Transactions* 16 :216-221, 1984.

[9] Okada M, Okada M. Prolog-based system for nursing staff

scheduling implemented on a personal computer. *Comput Biomed Res* 21 :53-63, 1988.

[10] Ozkarahan I. A flexible nurse scheduling support system. in *The 11th annual IEEE Symposium on Computer applications in Medical Care : 387-392, Waghington DC* 1987.

[11] Ozkarahan I. An integrated nurse scheduling model. *Journal of the Society for Health Systems* 3 (2) :79-101, 1991.

[12] Ozkarahan I, Bailey JE. Goal programming model subsystem of the flexible nurse scheduling support system. *IIE Transaction* 20 (3) :306-316, 1988.

[13] Sitompul D, U.Randhawa S. Nurse scheduling models: a state-of-the-art review. *Journal of the Society for Health Systems* 2 (1) :62-72, 1990

[14] R.C. Jelinek and J.A. Kavois. Nurse Staffing and Scheduling: Past Solutions and Future Directions. J*ournal of the Society for Health Systems* Vol.3, No.4, pg 75-82. 1992.

[15] Rosenbloom ES, Goertzen NF: Cyclic nurse scheduling. *European Journal of Operational Research* 31 (1) :19-23, 1987.

[16] Warner DM. Scheduling nursing personnel according to nursing preference : a mathematical programming approach. *Ops. Res.* 24 (5) :842-856, 1976.

[17] Smith LD, Wiggins. A Computer-Based nurse scheduling system. *Computer & Operations Research* 4 :195-212, 1977.

[18] Dincbas M, Van Hentenryck P, Simonis H, Aggoun A, Graf T and Berthier F. The Constraint Handling Language CHIP, *Proceedings of International Conference on Fifth Generation Computer Systems FGC88*, pg. 693-702, Tokyo, 1988.

[19] Weil G, Heus K, François P, and Poujade M. Solving the nurse scheduling problem using constraint programming", *IEEE Engineering in Medicine and Biology*, 14, 4, 1995 : pp. 417-422.

[20] Sissel Guttormsen Schär. The influence of the user interface on solving well- and ill-defined problems. *Int. J. Human-Computer Studies*, 44, 1996, pp. 1-18.

[21] Oldenkamp J.H., Lettenga. M.S. and Simmons J.L. Nursing schedule quality in theory and practice. In J. Brender, J.P. Christensen, J.-R. Scherrer and P. McNair (Eds). *Medical Informatics Europe 96*. Amsterdam, The Netherlands : IOS Press, pp. 863-866.

**Address for correspondence**

Georges Weil, MD, PhD
S.I.I.M.- Pavillon Pédiatrique
CHU de Grenoble
B.P. 217 - 38043 - Grenoble Cedex 9 - France
E-mail :Georges.Weil@imag.fr
URL : http://ujf-iab.ujf-grenoble.fr/IAB
Tél: +33 (0)4 76 76 50 47
Fax: +33 (0)4 76 76 88 44