JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. xx. No xx (200x), pp. xx-xx

Constraint satisfaction problem based modelling and value ordering based heuristic for nurse scheduling

Lakhdar DJEFFAL[*], Gilles GONCALVES, Malika BABES*****

[*]Université Hadj-lakhdar
Batna - Faculté des sciences
05000 – Algérie
lakdar_djeffal@yahoo.fr

*** Université d'Artois*
LGI2A-EA 3926
Technoparc-Futura 62400 Béthune
France
goncalves.gilles@univ-artois.fr

**** Université Badji mokhtar*
Faculté des Sciences ANNABA 23000 Algérie
malikababes@yahoo.fr

Abstract. *Nurse Scheduling Problem (NSP) consists of assigning shift-types (morning, afternoon and night) to qualified personnel over a certain planning period. It is a difficult and time-consuming task. In this paper we present a formulation of the hospital Nurse Scheduling Problem just like Constraint Satisfaction Problem ″CSP″ based constraint programming in order to find a solution, which minimizes the violation of Nurses' preferences. We would suggest a flexibility tool for helping to decide and for making negotiation easier. Our originality lies in the modelling of the problem, by defining the global constraints and in the algorithm of resolution to solve it, by proposing a new value ordering based heuristic for assignment of the decision variables taking into account the set of Nurses preferences. Our heuristic is based on the structure of the CSP and on the properties of constraints. It allows the reduction of the search space for solution and returns a solution within few second.*

Keyboards*: Constraint Programming, Global constraints, NP-Complete, Scheduling*

1. Introduction

Every hospital needs to repeatedly produce schedules for nursing staff. Because of, a large number of hospital constraints hard and soft [1] and a large number of possible Nurse assignments, Scheduling is one of the most difficult and complicated problems encountered by every Health care organization [2]. The Nurse Scheduling Problem (NSP) has attracted much research attention since the seventies. The main reason lies in that hospitals need to be staffed 24 hours a day over finite period P.

Two main approaches have been studied in the past: optimization problem type approach and decision problem type approach. In situation where there are a large number of constraints to be dealt with, the problem can be classified as a decision-type problem. Many NSPs are over-constrained, so that to find assignments to decision variable without violating any constraints is usually impossible. Consequently, the problem specification has to provide for the relative importance of constraints so that a solution to such a problem is allowed to violate a few constraints according to a priority order of constraints. Based upon the nurses' preferences for the various shift patterns, a penalty cost is associated to each assignment nurse-shift pattern. Naturally, NSP can be modelled as a Constraint Satisfaction Problem (CSP). CSPs form a simple formal frame to represent and solve certain problems in artificial intelligence (AI). They involve assigning values to decision variables subject to hard and soft constraints on which combinations are acceptable.

The problem of the existence of solutions to the CSP is NP-Complete. Therefore, many works have been developed to simplify the CSP before or during the search for solution. They are based more on the improved backtracking with importance on only the consistency rather than this one with the value ordering. It is well known that the efficiency of assigning values to decision variables, can be greatly affected by the choices made during solution searching.

For the purpose of this paper, we will study Nurse-Scheduling Problem (NSP) using a CSP in order to find a solution, which minimizes the violation of Nurses' preferences and we would suggest a flexibility tool for helping to decide and for making negotiation easier.

We present in section 2 a brief overview of background and in section 3 we give some definitions and notations. In section 4 we propose the modelling and one defines a tree based research algorithm including a new value ordering heuristic with consistency for solution of NSP. Experimental results are then reported in section 5 in order to value the efficiency of the algorithm. Finally, we conclude and we give some future prospects to our work in section 6.

2. Background

The organization of Nurses Scheduling is a hard problem, which aims to distribute in time the resources respecting a certain number of constraints, such as the law of work, regulation, workload, wishes or preferences of the staff. There are two basic scheduling types that are used for solving the NSP: cyclic and non-cyclic scheduling. In cyclic scheduling, each nurse works in a pattern, which is repeated in consecutive scheduling period, whereas, in non-cyclic scheduling, a new schedule is generated for each scheduling period. In our works, we focus on the second type.

Many algorithms and heuristics have been developed to solve NSP [3]. They are manifested in the different models. The optimization approach is usually based on Mathematical Programming (MP) techniques, while the decision approach usually employs heuristics and other artificial intelligence (AI) tools. In general, optimization-using MP can be classified in three categories: single-objective MP, multi-objective MP, and MP-based near-optimal approaches. For combinatorial problems, exact optimization usually requires large computational times to produce optimal solutions. In contrast, heuristic approaches can produce satisfactory results in reasonably short times. In the recent years, meta-heuristics, including taboo search, genetic algorithm have been proved to be very efficient in obtaining near-optimal solutions for problem including the Nurse-Scheduling problems [1], [4], [5].

AI techniques have been used to solve Nurse-Scheduling problems modelled as a CSP based on Constraint programming [6], [7], [8], [9], [10]. Commonly it employs algorithms of depth-first search in order to instantiate variables and to make modifications and a backtracking mechanism when dead ends occur [11]. Generally, CSP-based algorithms of resolution for the various Scheduling problems (specific or not to the hospital area) depend on variable/value ordering heuristics. ABDENNADHER and SCHLENKER [8] [12] adopt a partial CSP model for the problem. INTERDIP, which is their prototype system, supports semi-automatic creation of duty rosters and imitates certain aspects of manual planning to improve on the theoretical complexity of the problem, using a constraint package based on CHIP. The package includes linear equation, constraints over finite domains and Boolean constraints. MEYER AUF'M HOFE [13] modelled the problem as a special class of partial CSP, well known Hierarchical Constraint Satisfaction problem (HCSP), where legal regulation are hard constraints and nurses' preferences are usually lower level soft constraints. MEYER AUF'M HOFE [14] reported a commercial system ORBIS which models the problem as a HCSP with fuzzy constraints and inferred control strategies. ORBIS uses a Branch and Bound (B&B) algorithm with constraint propagation and variable/value ordering techniques to solve problems with on few minutes. In the context of job-shop scheduling problems, N.SADEH AND M.S.FOX [15] used a tree-like relaxation of the remaining problem. S. MINTON proposed a simple heuristic approach 'Min-conflict' [16]. It can be stated as: choose a value, which has the minimum number of conflicts with the assigned values for the other variables. DETCHER AND D. FROST introduced look a-head value ordering "LVO"[17]. "LVO" counts the number of times each value of the current variable conflicts is chosen first. Constraint solvers such as ILOG Solver and ECLIPSE by default use a search strategy similar to the Maintaining Arc Consistency algorithm [18]. WONG and CHUN use meta-level reasoning and probability- based ordering "MRPO" to solve Nurse Rostering Problem. Probability based ordering heuristic uses scoring functions based only on the properties of the constraints [19].

3. Preliminary

In this section, we introduce some definitions and notations used here after.

3.1. Definition$_1$

A Constraint Satisfaction Problem CSP is defined by (X, S, C) where:
$X = (x_1, x_2 ... x_N)$ is a finite set of N-variables such that each variable x_i has an associated domain S_i, i =1… N.
$S = (S_1, S_2, S_3 ... S_N)$ is a finite set of N-domains. S_i is the finite set of possible values for x_i, i =1...N.
$C = (C_1, C_2, C_3 ... C_m)$ is a finite set of constraints such that each constraint C_j (j=1,2 ... m) has an associated relation $T(C_j)$ denoting the set of tuples allowed for the variables $X(C_j)$ involved in the constraint. $X(C_j)$ is a finite subset of variables $(x_q \dots x_w) \subset X$, belonging to the constraint C_j.

With C a constraint: $|X(C)|$ denotes the arity of constraint C and X(C, i) represent the i th position of decision variables in X(C). Moreover, we note by #(v, t), composed of a tuple $t \in T(C)$ and of a value v from the domain S, the number of occurrences of the value v in the tuple t.

3.2. Definition$_2$

Let (X, S, C) be a CSP problem. An instantiation of a subset $Y \subseteq X$ constitutes a consistent instantiation if and only if all the constraints set C upon the variables Y are satisfied.

3.3. Definition$_3$ [18]

Let (X, S, C) be a CSP problem, τ_k is a shift-type (morning (k=1), afternoon (k=2) or night (k=3)) assigned for $x_i \in X$ in one finite period P. Subsequence "$\tau_i \, \tau_{(i+1) \bmod (P)} \dots \tau_j$" is called '**stretch**' when $\tau_i = \tau_{(i+1) \bmod (P)} = \dots = \tau_j$ but $\tau_{(i-1) \bmod (P)} \neq \tau_i$ and $\tau_{(j+1) \bmod (P)} \neq \tau_j$. For each possible solution t of X(C), one calls size or span of the '**stretch**' block including τ_k (with $i \le k \le j$) its length defined by :

Span (τ_k, t) = (1+ (j-i)) mod (P). This definition enables us to facilitate assignment of identical or same values in S for each variable $x \in X$.

4. Modelisation

The problem is that of creating schedules over a two weeks period P, containing up to N Nurses at hospital while taking into account the set of Nurses' preferences. These schedules must respect the set of legal and organisational constraints and have to satisfy the demand in terms of number of nurses required for each day/night shift. Because of, a large number of hard and soft hospital constraints, a large number of possible Nurse assignments and the existing of Nurses' preferences for the various shift-patterns, we model the NSP in the form of a **C**onstraint **S**atisfaction **P**roblem (CSP). We define in the next, the decision variables, nurses' preferences domains, definition domains and various constraints intervening for the generation of planning.

4.1. Decision Variables and Domains

For each nurse i ($1 \le i \le N$) working the day j ($1 \le j \le P$), a decision variable x_{ij} is associated. Let X= $\{x_{11} \dots x_{1P;}\ x_{21}, x_{22} \dots x_{2P;} \dots\ x_{N1}, x_{N2} \dots x_{NP}\}$ be the set of (N*P) decision variables, with N is the number of nurses and P is the finite period of the work planning. To ensure the service, Nurses (x_{ij}) must be assigned by shift-types over one finite period P. Let $\tau = \{\tau_1, \tau_2, \tau_0, \tau_3\}$ be the finite set of the 4 possible shift-type values, which can be assigned to the x_{ij} decision variables. The various shift-type values respectively correspond to morning, afternoon, rest and night. Note that the shift-type "rest" stands for all types of leave: annual paid holiday, maternity leave, sick leave, exceptional leave, etc.

	τ_1 then Nurse i is considered working on the morning shift
$\forall j \in 1..P$; $\forall i \in 1..N$;	τ_2 then Nurse i is considered working on the afternoon shift
if x_{ij} is assigned by:	τ_3 then Nurse i is considered working on the night shift
	τ_0 then Nurse i takes a rest

Consequently the definition domain of each variable xij is restricted to $\tau = \{\tau 1, \tau 2, \tau 0, \tau 3\}$ and so over one finite work period P, the number of possible assignments of N nurses is T = 4(P*N).

To avoid the absenteeism problem and to increase the productivity and the quality of medical cares, it is necessary to satisfy as much as possible the nurse wishes for assignment of shift-types. Consequently we introduced another domain of values that can be considered as preferred shift-types by a nurse for her assignment. Let $Dxij \subseteq \tau$ be the finite set of possible values chosen by a nurse i which represent her shift–type preferences for the day j of the period P

For each nurse i and for each day j the domain Dxij $\subseteq \tau$ is supposed to be given before working out the NSP. Finally, decision variables and domains are:
$X = \{x_{11} \dots x_{1P;}\ x_{21}, x_{22} \dots x_{2P;} \dots x_{N1}, x_{N2} \dots x_{NP}\}$ is a finite set of (N*P) decision variables, with N the number of nurses and P is the finite period of the work planning. Each variable x_{ij} ($1 \le j \le P$; $1 \le i \le N$) has an associated preference domain $D_{xij} \subseteq \tau$.
From now, one note by x any variable x_{ij} in X, and by D_x any preference domains D_{xij} ($\forall$j,i $1 \le j \le P$; $1 \le i \le N$). This domain D_x is set by nurse x and it corresponds to its preferences for the future shift assignment

4.2. Constraints

There are two categories of constraints: hard constraints that must be always satisfied and soft constraints that may be sometimes violated, but should be satisfied as far as possible. We use the soft constraints because in practice it is impossible to completely satisfy all constraints.

4.2.1. Global Constraints of management and organization (Soft constraints)

4.2.1.1. Nurse workload per day

The first constraint is implicitly expressed by the fact that one nurse works during a time period of 8 hours for one day shift and during 10 hours for one night shift.

$$\text{If } x = \left|\begin{array}{l} \tau_1 \mid \tau_2 \text{ the time of workload is 8 hours long} \\ \tau_3 \text{ the time of workload is 10 hours long} \\ \tau_0 \text{ the time of workload is 0 hours long} \end{array}\right.$$

4.2.1.2. Nurse work duration per period

The duration constraint is set over period P. We consider a two weeks period (P=14 days). It limits to 80 hours the nurse work in case of a day shift and to 70 hours in case of a night shift. The writing of these constraints depends on the following cases:

4.2.1.2.1. Work night period

The night shift Constraint C is a **G**lobal **C**ardinality **C**onstraint **N**ight, defined on a set of variables $X(C) = \{x_{iq} \dots x_{iw}\} \subset X$ by a set of tuple denoted T(C). It constrains the number of times that a value $\tau_3 \in (D_x \subseteq \tau)$ is assigned to each variable in X(C) over the period P to be in an interval] 0, 7].

Syntax: GCCN (P, τ , $\forall x \in X$, $D_x \subseteq \tau$, 0, 7)
T(C) = {t, t one tuple of X(C), 0< #(τ_3, t) $\le$ 7}.

T(C) is the set of the tuples t (possible solutions); each one contains no more than 7 times the $\boldsymbol{\tau_3}$ value.
For example: for one nurse $x \in X$, the domain of definition is $\tau = \{\tau_1, \tau_2, \tau_0, \tau_3\}$ and the domain of preference is $D_x = \{\boldsymbol{\tau_0}, \boldsymbol{\tau_3}\}$. For one period P of 14 days, the set of possible solutions T(C) that satisfied the above constraint are given in Tab.1

Jours / Tuples	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14
Tuple t1	τ_0	τ_0	τ_0	τ_3	τ_3	τ_0	τ_3	τ_0	τ_3	τ_0	τ_0	τ_3	τ_3	τ_0
Tuple t2	τ_3	τ_3	τ_0	τ_0	τ_3	τ_3	τ_3	τ_0	τ_0	τ_3	τ_0	τ_0	τ_3	τ_0

Tab.1 set of possible solutions

For tuple t1, $X(C) = \{x_{i4}, x_{i5}, x_{i7}, x_{i9}\ x_{i12}, x_{i13}\}$
For tuple t2, $X(C) = \{x_{i1}, x_{i2}, x_{i5}, x_{i6}\ x_{i17}, x_{i10}, x_{i13}\}$

4.2.1.2.2. Work morning period

The morning shift Constraint C is a **G**lobal **C**ardinality **C**onstraint **M**orning, defined on a set of variables $X(C) = \{x_{iq} \ldots x_{iw}\} \subset X$ by a set of tuple denoted T(C). It constrains the number of times that the morning shift corresponding to value $\tau_1 \in (D_x \subseteq \tau)$ is assigned to variables x on one period P to be in an interval]0, 10]

Syntax: GCCM (P, τ , $\forall x \in X, D_x \subseteq \tau$, 0, 10)
T(C) = {t, t one tuple of X(C), $0 < \#(\tau_1, t) \leq 10$}.
T(C) is the set of the tuples t (possible solutions) each one contains no more than 10 times the $\boldsymbol{\tau_1}$ value. In other words, the number of shift-type $\boldsymbol{\tau_1}$ occurs in a tuple t is more than 0 times and less or equal 10 times.

4.2.1.2.3. Work afternoon period

The afternoon shift Constraints C is a **G**lobal **C**ardinality **C**onstraint **A**fternoon, defined on a set of variables $X(C) = \{x_{iq} \ldots x_{iw}\} \subset X$ by a set of tuple denoted T(C). It constrains the number of times that the shift-type value τ_2 occurs in tuple t to be in the following interval] 0, 10].

Syntax: GCCA (P, τ, $\forall x \in X, D_x \subseteq \tau$, 0, 10)
T(C) = {t, t a tuple of X(C), $0 < \#(\tau_2, t) \leq 10$}.

T(C) is the set of the tuples t; each one contains no more than 10 times the $\boldsymbol{\tau_2}$ value of the domain.

4.2.1.2.4. Work mixed Period

The Work mixed Period Constraints C is a **G**lobal **C**ardinality **C**onstraint **Mix**ed, defined on a set of variables $X(C) = \{x_{iq} \ldots x_{iw}\} \subset X$ by a set of tuple denoted T(C) (i.e. a subset of the Cartesian product D_{xiq} * … * D_{xiw} such as $(1 \leq q \leq w \leq P)$). It is specified in terms of a set of decision variables x, which take their various values $\tau_{k,\, k=1,2,3}$ of ($D_x \subseteq \tau$) on period P. It constrains the number of times that various shift-type $\tau_{k,\, k=1,2,3}$ assigned to variables in X(C), to be in an interval]0,8].

<u>Syntax:</u> GCCMx(P,τ, $\forall x \in X$, $D_x \subseteq \tau$, 0, 8)
T(C) = t, t a tuple of X(C), $\forall\{\tau_{k,\, k=1,2,3}\} \in D_x$, $0 < \#(\tau_{k,\, k=1.2.3}, t) \leq 8$}.
T(C) is the set of tuples t, each one contains at least the 8 times various value $\tau_{k,\, k=0,1,2,3}$ of the domain $D_x \subseteq \tau$. For example: for one nurse $x \in X$, the domain of definition is $\tau = \{\tau_1, \tau_2, \tau_0, \tau_3\}\rangle$ and the domain of preference is $D_x = \{\boldsymbol{\tau_1}, \boldsymbol{\tau_0}, \boldsymbol{\tau_3}\}$. On one finite period P=14 days, the set of possible solutions T(C) that satisfied the above constraint are given in Tab.2.

Tuples \ Jours	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14
Tuple t1	τ_0	τ_1	τ_0	τ_3	τ_3	τ_0	τ_3	τ_1	τ_1	τ_0	τ_0	τ_3	τ_2	τ_0
Tuple t2	τ_3	τ_1	τ_0	τ_1	τ_3	τ_1	τ_3	τ_0	τ_0	τ_3	τ_0	τ_1	τ_3	τ_3

Tab.2 set of possible solutions

For tuple t1, X(C) = $\{x_{i2}, x_{i4}, x_{i5}, x_{i7}, x_{i8}, x_{i9}\ x_{i12}, x_{i13}\}$. In term of work time, the sum of the hours carried out by the nurse x is 72 hours (see the first constraint).

4.2.1.3. Constraints of consecutive works

It is very significant to ensure stability in nurse planning, i.e. to be able to specify a minimum (λ_{min}) and a maximum (λ_{max}) number of identical assignments in the shift sequence assigned to the nurse. The constraint ″Stretch″ [20], [21], was recently proposed to limit the consecutive work assignment in shift-type sequences and to improve the quality of the generated timetables.

Let λ_{min} and λ_{max} be the vectors of m > 0 length represented respectively the minimal and maximal size of the '**stretch**' block for each τ_i value. One then poses the constraint with following semantic:

-$\forall i$, $0 \leq i \leq (P-1)$, $\tau_i \in (D_x \subseteq \tau)$,
-$\forall i$, $0 \leq i \leq (P-1)$, $\lambda_{kmin} \leq span(\tau_i = \tau_k, t) \leq \lambda_{kmax}$.
-$\forall \tau_{k(\, k=1,2,3)} \in D_x$, $\lambda_{kmin} \leq \lambda_{kmax}$.

4.2.1.3.1. Case of day's-continuation

This constraint is set for each block of 9 days. Nurse cannot exceed 07 days continuation of work and must be in rest at least for 2 days. Constraint day's-succession of works C is specified in terms of a set of decision variables x that take their same τk, k=1,2 values. It constrains the number of times that a given shift-type τk assigned to variables x, to be in an interval]λmin, λmax].

<u>Syntax</u>: Stretch(X, P, $D_x \subset \tau$, τ, λ_{min}, $\lambda_{max.}$)
T(C) = {t, t a tuple of X(C), $\forall \tau_{k\ (k=1,2)} \in D_x$, $\lambda_{min} \leq span(\tau_k, t) \leq \lambda_{max.}$} Such as λ_{min}=1; $\lambda_{max.}$ =7; $D_x = \{\tau_1, \tau_2\} \subset \tau$.

T(C) is the set of the tuples t (possible solutions) of X(C), each one contains more than λ_{min} times and at least λ_{max} times the same $\tau_{k,\ k=1,2}$ values of the domain $D_x \subset \tau$.
X(C) = $\{x_{iq} \ldots x_{iw}\} \subset X$ such as i is fixed and ($1 \leq q \leq w \leq P$);

4.2.1.3.2. Case of night-continuation

This constraint is set on each block of 07 days. Nurse cannot exceed 04 nights continuation of work and must be in rest at least for 02 days. Constraint night-succession of works C is specified in terms of a set of decision variables x that take on one finite period P the value τ3. It constrains the number of times that night shift assigned to variables x to be in an interval]λmin, λmax] (see example in tab-3).

<u>Syntaxe</u>: Stretch ($x_{i1}, x_{i2}, x_{i3}, \ldots x_{i(N*P)}$, P, $D_x \subset \tau$, τ , λ_{min}, $\lambda_{max.}$)
T(C) = {t, t a tuple of X(C), $\tau_3 \in D_x$, $\lambda_{min} \leq Span(\tau_3, t) \leq \lambda_{max}$} such as λ_{min} =1; λ_{max} =4; $D_x = \{\tau_3\} \subset \tau$ $\tau = \{\tau_1, \tau_2, \tau_0, \tau_3\}$.

T(C) is the set of the tuples t (solutions) of X(C), each one contains more than λ_{min} times and at least λ_{max} times the τ_3 value.
X(C) = $\{x_{iq} \ldots x_{iw}\} \subset X$ such as i is fixed and ($1 \leq q \leq w \leq P$),

Period / tuples	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14
tuple t1	τ_3	τ_3	τ_3	τ_0	τ_0	τ_0	τ_0	τ_0	τ_0	τ_3	τ_3	τ_3	τ_0	τ_3
tuple t2	τ_0	τ_3	τ_0	τ_0	τ_3	τ_3	τ_3	τ_3	τ_0	τ_0	τ_0	τ_3	τ_3	τ_0

Tab.3 set of possible solutions

T(C) = {t1, t2} such as:
For tuple t1, X(C) = $\{x_{11}, x_{12}, x_{13}, x_{1\ 10}, x_{1\ 11}, x_{1\ 12}, x_{1\ 14}\}$
For tuple t2, X(C) = $\{x_{12}, x_{15}, x_{16}, x_{17}, x_{18}, x_{1\ 12}, x_{1\ 13}\}$

4.2.1.4. Case of unauthorized continuation (Hard constraints)

It is inconceivable to work during the night shift on day j then the morning or the afternoon next day on one finite period P. Constraints unauthorized continuation of works C is a <u>**G**</u>lobal <u>**C**</u>onstraint <u>**N**</u>ot <u>**A**</u>uthorized for succession of shift-types. It is specified in terms of a decision variables x in X, which take on one finite period P their succession various $\tau_{k,\ k=1,2,3}$ values of domain. It constrains the nature of shift-type successions assigned to a variables x, $\forall x \in X$. In other words:

$\forall j \in 1..P$; $\forall i \in 1..N$, If x_{ij} is assigned by the value τ_3 then the decision variables $x_{i(j+1)}$ does not assigned by the value τ_1 or τ_2.

Syntax: GCNA(P,$\forall x \in X$, $D_x \subseteq \tau$, τ, s);
T(C) = {t, t a tuple of X(C), $x \in X$, s:(x_{ij}, τ_3)$\nRightarrow$ s:($x_{i(j+1)}$, τ_3) or ($x_{i(j+1)}$, τ_0)}

T(C) is the set of the tuples t (solutions); each one contains a sequence **s** of τ_k values such as the nature continuation of shift-types **s** = (…$\tau_3\tau_1$...) or ($\tau_3\tau_2$...) are unauthorized. X(C) = {x_{iq} … x_{iw}} τ_3 X such as i is fixed and ($1 \le q \le w \le P$).

4.2.2. Global Constraint Cardinalities (Hard constraints)

4.2.2.1. Constraint of nurse requirements

The Constraints min/max of nurse requirement C is <u>G</u>lobal <u>C</u>onstraint <u>C</u>ardinalities, which specify nurse requirements for each day j and for each shift (morning, afternoon, night) of period P. It is specified in terms of a set of decision variables X that take their values in a subset (union of domains) $(\cup_{i \in X(C)} D_{X_i} \subseteq \tau), 1 \le i \le N$. It constrains the number of times that a given shift-type τ_k value assigned to variables of X, to be in an integer interval [l_{kj}, u_{kj}]. The $(\cup_{i \in X(C)} D_{X_i})$ corresponds to Cartesian product of preference domains for a subset variables x_{ij} of nurse requirement constraint C.

Syntax: GCC ($\forall x \in X$, $j \in P$, $\tau, \cup_{i \in X(C)} D_{X_i} \subseteq \tau)$, l_{kj} , u_{kj});

T(C) = {t, t a tuple of X(C), $\forall k$, $\tau_k \in \cup_{i \in X(C)} D_{X_i} \subseteq \tau)$, $l_{kj} \le \#(\tau_k, t) \le u_{kj}$}

T(C) is the set of the tuples t (solutions), each one contains with more the lkj times and at least ukj times the τk value. In other words, the number of occurs of any shift-type τk in a tuple t of X(C) is more than lkj times and less or equal than ukj times. The values l_{kj} and u_{kj} respectively represent the minimal and the maximal (min/max) number of nurse requirements for day j ($1 \le j \le P$) in morning shift, in afternoon shift or in night shift.

4.2.2.2. Cost constraints

Based upon the nurses' preferences for the various shift patterns, the recent history of shift-patterns worked, and the overall attractiveness of the shift-pattern, a penalty "Cost" is associated to each assignment. The constraint is:

<u>Syntax</u>: COST-Function ($j \in 1…P$, τ, $\forall x \in X$, $D_x \subseteq \tau$, l_{kj}, u_{kj}, *Cost, Mini*)

T(C) = {t, t tuple of X(C) and $\forall k$, $\tau_k \in D_x$: $l_{kj} < \#(\tau_k, t) \le u_{kj}$ and $\forall j \in 1\ldots P$,

$$\sum_{i=1}^{|X(C)|} Cost(X(C,i), t[i]) \le Mini\ \}$$

"COST-Function" is a conjunction of Global Cardinality Constraint of Nurse requirement and sum constraints of costs for the corresponding assignments. It is defined on a set of variables $X(C)=\{x_{1j} \ldots x_{nj}\}$ by a subset of the Cartesian product $D_{x1j} * \ldots * D_{xNj}$, associated with an index performance "Cost", an integer value *"Mini"* in which each $\tau_k \in D_x$ is associated with two positives integers l_{kj} and u_{kj}. "Cost" is a function that associates to each violate assignment an integer number noted "Cost (x_{ij}, τ_k)" on X(C).

5. Proposed Algorithm

The main problem is **how to select or choose good values τ_k** from the preference domain D_x in order to minimize the violation of the registered preferences of nurses?

One defines two types of heuristic:

1-A decision variable ranking based heuristic for selecting the next variable x to instantiate and its domain $D_x \subseteq \tau$ is used to obtain trees with fewer branches.

2-A value ordering based heuristic for choosing a value τ_k among a finite set of values of preference domains $D_x \subseteq \tau$ for the variable x (selected by the first heuristic) is used to satisfy a constraint of needs for nurses and to reduce search space by propagation.

DATA:
- P: period of 14days, **N**: number of nurses

- Decision variables to be assigned on one finite period P are:
X $=\{x_{11}, x_{12} \ldots x_{1P}; x_{21}, x_{22} \ldots x_{2P}; \ldots x_{N1}, x_{N2} \ldots x_{NP}\}$, /* x_{ij}: represent the nurse i in day j */.

- $\boldsymbol{\tau} = \{\tau_1, \tau_2, \tau_0, \tau_3\} = \{\langle 1\rangle, \langle 2\rangle, \langle 0\rangle, \langle 3\rangle\}$ the definition domains of each $x_{ij} \in X$ such as $\langle 1\rangle, \langle 2\rangle, \langle 0\rangle, \langle 3\rangle$ represent respectively the morning, afternoon, rest and night shift, that can be used when there are violations of nurses' preferences.

- Registration of Nurses by filling the form of identification and preferably of the shift-type $\tau_{k;\ k=0..3}$. This registration defines the nurses' preferences or domain of preference $\mathbf{D_x} \subseteq \tau$, $\forall (x=x_{ij}) \in X$.

Example: Suppose that we have the 3 Nurses' registration in day j ($1 \leq j \leq P$):
x_{1j}::={⟨1⟩, ⟨0⟩}; x_{2j}::= {⟨1⟩, ⟨2⟩, ⟨0⟩}; x_{3j} ::= {⟨1⟩, ⟨2⟩, ⟨0⟩, ⟨3⟩}; Then the finite set of preference domains D_{xij} for each nurse in day j are : D_{1j} = {⟨1⟩,⟨0⟩}, D_{2j} = {⟨1⟩,⟨2⟩,⟨0⟩}, D_{3j} = {⟨1⟩,⟨2⟩,⟨0⟩, ⟨3⟩}

We suppose that the dimensioning of manpower or Nurse Requirement Constraints C_{rj} (r=1, 2, 3) on one finite period is known. That result in the nurses needed into a minimum number l_{rj} and a maximum number u_{rj} for each day j each shift (morning, afternoon and night). $C_{rj}= [l_{rj}, u_{rj}]$; r = 1, 2, and 3 and for $j \in 1... P$. In other words, the min/max of shift –type τk needed to be assigned for nurses in morning shift, afternoon shift, and night shift for each day j of period P.

Procedure
For ($j\in 1..P$, X = {x_{1j}, x_{2j} … x_{Nj}}: N decision variables to be instantiated by the values of $D_x\subseteq \tau$) **do**
Begin1
-. Let Code (C_{rj}) = ⟨r⟩;
/* 'Code' is a simple variable, which represents the code of load constraints C_{rj} (Morning r=1,
Afternoon r=2 and Night r=3; */
.. **For** (r=1...3) **do**
. $[l_{0j}, u_{0j}] = [N, N] - ([l_{rj}, u_{rj}])$
/* $[l_{0j}, u_{0j}]$ is the load constraint of rest shift on each day j of Period P*/
.. **If** ($[l_{0j}, u_{0j}] = [0, 0]$) **then**
For ($i\in 1..N$) **do** $D_{xij} = D_{xij}$ – {⟨0⟩};
****. Repeat**
Begin2
1- A decision variable ranking based heuristic: MinEq1
- Let $h \in 1... N$, such as [D_{hj} = **Min** ($|D_{xij}|$ such as $D_{xij}\neq\{\phi\}$, $\forall i\in 1…N$ and $x_{ij}\in X$)**];**
<u>MinEq1</u>: select the variable x_{ij} associated at D_{xij} which corresponds to the minimal not empty domain size or the minimal number of values belonging to the recent domain of preference $D_{xij}\subseteq \tau$, because D_{xij} can be reduced from one moment to another at time of search for solution. In case of equality a lexical order are used.
- So, a Decision variable x_{hj} that corresponds to the above D_{hj} is selected**.**

2- Value Ordering based Heuristic: MinEq2
Once the variable x_{ij} is selected by the previous heuristic, we have to choose the more appropriate value τ_k for its assignment. Which value τ_k

to choose is a complex factor of our problem? For that, we propose a value ordering based heuristic as follow:

<u>MinEq2</u>: The choice of τ_k value from the domain D_{hj} for the decision variable x_{hj} selected by heuristic <u>MinEq1</u> corresponds to minimum non-null recent values of load constraints C_{rj} relating to D_{hj}. The C_{rj} identification is such as $\langle r \rangle \in D_{hj}$. **MinEq2 = min** $([l_{rj}, u_{rj}] \neq 0, \forall \langle r \rangle \in D_{hj})$**;** In case of equality, a lexical order is again used.

- If (D_{hj} exists) **/*** D_{hj} is the domain selected by MinEq1 */ **then**
Begin3
. $\exists(\tau_k \in (D_{hj} \subseteq \tau))$ such as τ_k corresponds to CODE (MinEq2);
/*Example If MinEq2=C_3 then CODE (C_3) = $\langle 3 \rangle$ /*
. $\langle r \rangle \leftarrow \tau_k$
. $x_{hj} \leftarrow \tau_k$; /*creation of the tree node (Fig-1) /*
label_19:**.While** (Constraints **Stretch, GCNA, GCCMx, GCCN, GCCA & GCCM**) are not satisfied
Begin
If $((D_{hj} - \{\tau_k\}) \neq \{\phi\})$ **then**
- $\exists(\tau_{k'} \in D_{hj})$; $x_{hj} \leftarrow \tau_{k'}$; $\tau_k \leftarrow \tau_{k'}$; /* D_{hj} : domain of preference */
Else /* we violate the constraint of nurses' preferences */
- $\forall(l \neq k), \exists(\tau_l \in \tau)$; $\tau_k \leftarrow \tau_l$; /* τ: domain of definition */
$x_{hj} \leftarrow \tau_k$; /*creation of a node in search tree (Fig-1) /*
End
. **If** ($\tau_k = \langle 1 \rangle$ or $\langle 2 \rangle$) **then**
.. **If** ($x_{h(j-1)} = \langle 3 \rangle$) **then** $x_{hj} \leftarrow \langle 3 \rangle$ or $x_{hj} \leftarrow \langle 0 \rangle$; (with $j \neq 1$)
. $C_{rj} = C_{rj} - 1$; /* C_{rj} corresponds to the global constraint of cardinality **GCC** */
. **If** C_{rj} is satisfied **then**
For $x_{ij} \in X$, **do** $D_{xij} = D_{xij} - \{\langle r \rangle\}$; /* that explains the underlined value in (Fig-1) */
. $X = X - \{x_{hj}\}$; $N = N - 1$;
End3
- **Else**
Begin4

/* In this case it is necessary to use a procedure of backtracking in the search tree (Fig-1) [11] to change the value of the variable susceptible to be the cause of blockage /*

Let h' such as $x_{h'j}$ was instantiated before x_{hj} by $\tau_{h'}$ value, that considered a value of blockage. We memorize it for not to use in case of another blockage and we use Mineq2 heuristic to select another value.

. If (h'≠h) exists & ($D_{h'j}$-{$\tau_{h'}$}) ≠ {ϕ} **then**
Begin
/* h' is given such as $x_{h'j}$ was instantiated before x_{hj} */
.. Memo ← $\tau_{h'}$
.. ∃h" / $\tau_{h''} \in (D_{h'j} \subseteq \tau)$
.. $x_{h'j} \leftarrow \tau_{h''}$;
.. Actualize Variables X, D_x, C_{rj}, N;
.. k← h";
.. h ← h'
End
. Else
Begin
.. We must violate the nurse' preference and one uses definition domains τ instead of D_{hj} and we choose a value $\boldsymbol{\tau_k \in \tau}$ that corresponds to nurse requirement constraint C_{rj} not satisfied
.. $D_{hj} \leftarrow \tau$;
.. $\tau_k \leftarrow \tau_l$;
.. $x_{hj} \leftarrow \tau_l$;
End
- Go to line labeled by **label_19**
End4
End2
** **Until** (X ≠{ϕ})
End1.

The phenomenon of backtrack arises when the preference domain $Dx_{ij} \subseteq \tau$ of a variable becomes empty, that is to say for $x_{ij} \in X$, Dx_{ij} = {ϕ}. At this time backtrack is carried out on the search tree until reaching the susceptible variable to be the cause of blocking. We memorize it to prevent of taking it again in another blocking and we choose another value for this variable. If we cannot then we violate the preference of nurse and we take a value of definition domain τ which corresponds to the load constraint not satisfied.

Note that if load constraint C_{rj} is satisfied, the value chosen $\tau_k \in Dx_{ij}$ for variable $x_{ij} \in X$ must to be removed from all other domains of the decision variables not instantiated ($Dx_{ij} = Dx_{ij} - \{\tau_k\}$, ∀i∈1..N and $x_{ij} \in X$). This avoids the exploration of the sterile branches in the tree (Fig1) at the time of choice of values $\tau_k \in Dx_{ij}$.

Consequently our heuristics reduce considerably the search space in the tree. This reduction is done by the procedures of local consistency.

LOCAL CONSISTENCY (Filtering)

The filtering for each constraint defined in the previous paragraph is based on the dynamic decision variables ordering and value based ordering of preference domains.

We try to reduce each load constraints C_{rj} to be zero. It is possible for severe reasons of needs, that the dimensioning of manpower for the rest load constraints (C_{0j}) is null at the beginning of the procedure. In this case, a filtering procedure eliminates the τ_0 value from the all domains D_{xij} of decision variables $x_{ij} \in X$.

Begin

$\tau = \{\tau_1, \tau_2, \tau_0, \tau_3\}$; $D_{xij} \subseteq \tau$, $\forall x_{ij} \in X$, $i \in 1…N$ and $j \in 1...P$.

If (($l_{0j} = 0$ & $u_{0j} = 0$) **then**

For each $x_{ij} \in X$ and $\tau_0 \in D_{xij}$, $D_{xij} = D_{xij} - \{\tau_0\}$;

End

If the dimensioning of manpower of a load constraint has a non-null lower limit l_{kj} and upper limit u_{kj}, one uses then a counter 'Copt' for each one, which consists in checking satisfaction and which is useful for actualization of parameters X, D_{xij} and load constraints C_{kj} if a conflict problem appears.

One increases the counter Copt [k] of load constraint C_{kj} and one decrease the C_{kj} value for each decision-making on the τk value choices. If C_{kj} value is null then one reduces the domains D_{xij} of each variable $x_{ij} \in X$ not instantiated (that explains the underlined value in the search tree (Fig-1). If not, one decreases it and so on until so that it becomes null.

1- Input [l_{kj}, u_{kj}] or C_{kj} and initialized Copt [k]
2- **For** (x_{ij} not assigned) do

/* (**If all variables x_{ij} are assigned one stops**)/*

3- **Call** heuristic of decision variables ordering 'Mineq1'
4- **Call** heuristic of value based ordering heuristic 'Mineq2'
5- Let k such as $\tau_k \in D_x$, Copt [k] = Copt [k] +1;
* **If** (x_{ij}, τ_k) **then** /* $\exists(\tau_k \in Dx_{ij})$ such as x_{ij} is assigned by this value*/
Begin
$C_{kj} = C_{kj} - 1$
If $C_{kj} = 0$ **then**

For i∈ 1…N do $D_{xij} = D_{xij} - \{\tau_k\}$;
Else go to step2
End
***Else**
Begin
*/$D_x = \{\phi\}$: all τ_k values of the selected domain are filtered and thus one blocking arises. */

- Call backtrack heuristic [11] that returns a likely value to be the cause of blocking.
- Update of the counter 'Copt' and actualize the parameters X, D_{xij} and each C_{kj}

End
6- **Go to** step 2.

<u>Filtering for the constraint "**Stretch**"</u>

The filtering of this constraint is strongly necessary. It is inconceivable, to program nursing in night shift followed by the morning or afternoon shift, as it is significant, to specify the minimal number and the maximal number of assignment of the same activity to the continuation.

$\forall j \in 1\ldots P$, $\forall i \in 1\ldots N$ and $x_{ij} \in X$, $\tau = \{\tau_1, \tau_2, \tau_0, \tau_3\}$

Let s the variable, which specifies the case days continuation (s=7) and the case nights continuation (s=4);

Let k / $\tau_k \in (D_{xij} \subseteq \tau)$; $x_{ij} \leftarrow \tau_k$
For ii= j+1, (j+2) mod (P) **do**
Begin
$x_{ii} \leftarrow \tau_k$
If ($\tau_k = \tau_{k'}$) **then**
Begin
Span (ii, j) = 1+(ii-j) mod (P)
If (Span (ii, j) ≤ s) **then**
{- Repeat the same assignment.}
Else
{Let kk/ $\tau_{kk} \in (D_x \subseteq \tau)$; $x_{ii} \leftarrow \tau_{kk}$/ ($\tau_{kk} \neq \tau_k$);}
End
End.

For more clarification and other algorithms for the filtering of 'Stretch' constraints, please see [18], [19]

6. Experimental methods and results

Suppose that, we have: N=8 nurses; $X = \{x_{11}, x_{12}, \dots x_{1P}; x_{21}, x_{22}, x_{23}, \dots x_{8P}, \dots x_{81}, x_{82}, x_{83}, \dots x_{8P})$ the set of 112 decision variables to be assigned on one finite period P=14 days long.

The size of search space or the possible number for assignments is $T = 4^{112}$. The definition domains $\tau = \{\tau_1, \tau_2, \tau_0, \tau_3\} = \{\langle 1\rangle, \langle 2\rangle, \langle 0\rangle, \langle 3\rangle\}$ and the registration of the 8 Nurses that we considered as preference domains $D_{xij} \subseteq \tau$, $\forall x_{ij} \in X$ $(1 \le i \le 8; 1 \le j \le 14)$ are the following:

D_{xij}	Nurse1	Nurse2	Nurse3	Nnurse4	Nurse5	Nurse6	Nurse7	Nurse8
Day 1	{⟨1⟩,⟨2⟩,⟨0⟩, ⟨3⟩}	{⟨1⟩, ⟨0⟩}	{⟨1⟩,⟨0⟩,⟨3⟩}	{⟨1⟩, ⟨2⟩, ⟨0⟩}	{⟨1⟩, ⟨0⟩}	{⟨2⟩,⟨0⟩, ⟨3⟩}	{⟨0⟩, ⟨3⟩}	{⟨2⟩, ⟨0⟩}
Day 2	{⟨1⟩,⟨2⟩,⟨0⟩, ⟨3⟩}	{⟨1⟩,⟨2⟩,⟨0⟩}	{⟨2⟩,⟨0⟩,⟨3⟩}	{⟨1⟩,⟨2⟩,⟨0⟩,⟨3⟩}	{⟨1⟩, ⟨0⟩}	{⟨1⟩,⟨2⟩, ⟨0⟩}	{⟨2⟩, ⟨0⟩}	{⟨1⟩, ⟨2⟩, ⟨0⟩}
Day 3	{⟨1⟩,⟨2⟩,⟨0⟩, ⟨3⟩}	{⟨1⟩,⟨2⟩,⟨0⟩}	{⟨2⟩,⟨0⟩,⟨3⟩}	{⟨1⟩,⟨2⟩,⟨0⟩,⟨3⟩}	{⟨1⟩,⟨0⟩}	{⟨1⟩, ⟨2⟩, ⟨0⟩}	{⟨2⟩, ⟨0⟩}	{⟨1⟩,⟨2⟩,⟨0⟩}
Day 4 …	{⟨1⟩, ⟨0⟩}	{⟨1⟩,⟨0⟩,⟨3⟩}	{⟨1⟩,⟨2⟩,⟨0⟩, ⟨3⟩}	{⟨0⟩, ⟨3⟩}	{⟨1⟩,⟨2⟩⟨0⟩}	{⟨0⟩,⟨3⟩}	{⟨1⟩,⟨2⟩,⟨0⟩, ⟨3⟩}	{⟨2⟩, ⟨0⟩}

Tab.4 nurses' preferences

The nurses' requirements constraints C_{rj} are: C_{1j} (morning), C_{2j} (afternoon) and C_{3j} (night) respectively given for each day j in tab.5.

One notices that: $C_{0j} = (N - (C_{1j} + C_{2j} + C_{3j}))$ the nurse requirement constraint in rest.

Solution:

For day-j=1 and day-j=2 one uses the same nurses' requirement constraints C_{rj}, r=1, 2, 3 and one varies the preference domains D_{xij}.

For day-j=2 and day-j=3 one preserves the same preference domains D_{xij} and one varies nurses' requirement constraints C_{rj}, r=1, 2, 3.

For day j=3 and day-j=4 one varies both.

Req. Const.	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8				J_{13}	J_{14}
C_1	4	4	2	3	3	..	..	..	..	..	..	..	.
C_2	2	2	3	1	2	..	..	..	..	..	..	..	.
C_3	2	2	1	3	3	..	..	..	..	..	..	..	.

Tab.5 nurses' requirements

The assignments are given as follow:

Day j = 1

x_{1j}	x_{2j}	x_{3j}	x_{4j}	x_{5j}	x_{6j}	X_{7j}	x_{8j} (nurse 8 is assigned to afternoon-shift on day j=1)
$\langle 1\rangle$	$\langle 1\rangle$	$\langle 3\rangle$	$\langle 1\rangle$	$\langle 1\rangle$	$\langle 2\rangle$	$\langle 3\rangle$	$\langle 2\rangle$

Day j = 2

x_{1j}	X_{2j}	x_{3j}	x_{4j}	x_{5j}	x_{6j}	x_{7j} (nurse 7 is assigned to night-shift on day j=2)	X_{8j}
$\langle 2\rangle$	$\langle 2\rangle$	$\langle 3\rangle$	$\langle 1\rangle$	$\langle 1\rangle$	$\langle 1\rangle$	$\langle 3\rangle$	$\langle 1\rangle$

Day j = 3

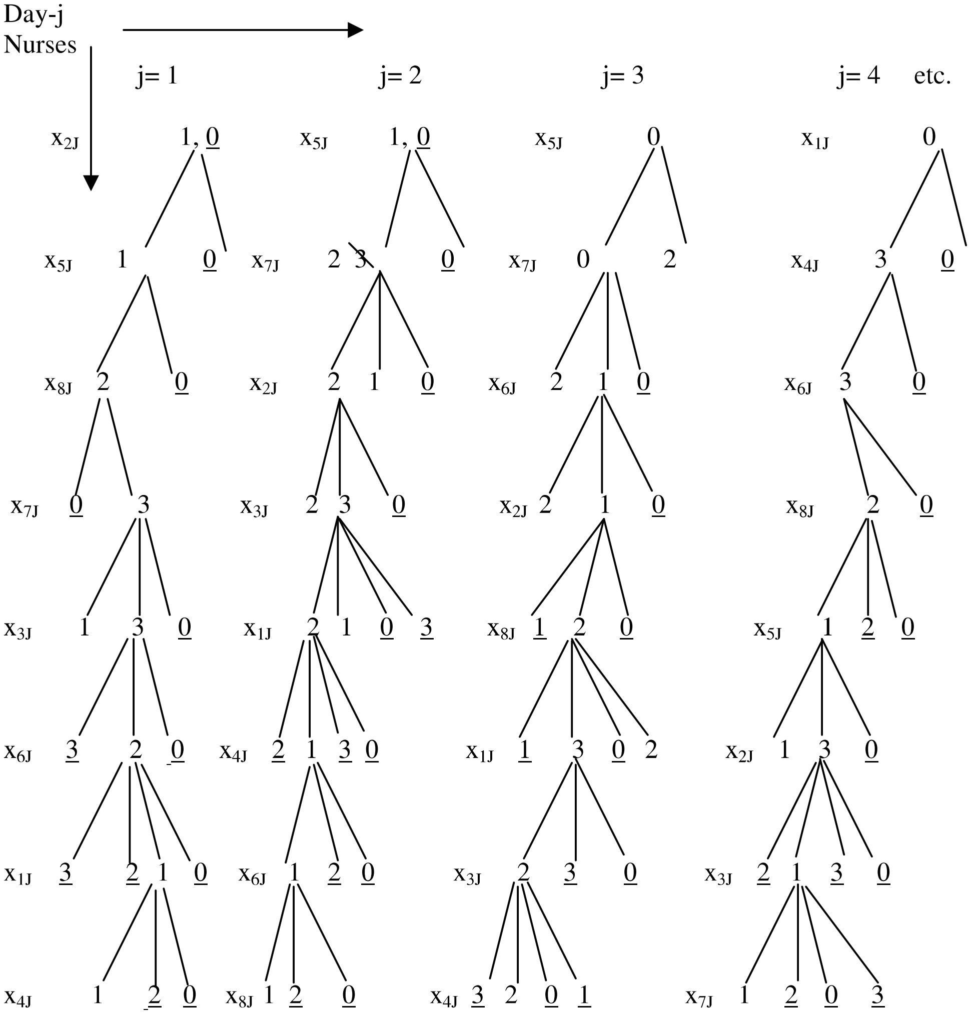
x_{1j}	X_{2j}	x_{3j}	x_{4j}	x_{5j} (nurse 5 is in rest on day j=3)	x_{6j}	x_{7j}	X_{8j}
$\langle 3\rangle$	$\langle 1\rangle$	$\langle 2\rangle$	$\langle 2\rangle$	$\langle 0\rangle$	$\langle 1\rangle$	$\langle 0\rangle$	$\langle 2\rangle$

Etc…

This solution is given by the search tree (Fig. 1). Underlined values are to be eliminated by filtering from the search space and the barred values correspond to preference violations.

For example, we explain in the following how the nurses assignment of the day j=1 is made. We have $X = \{x_{11}, x_{21}, x_{31}, x_{41}, x_{51}, x_{61}, x_{71}, x_{81}\}$. Each x_{i1} (Nurses) has an associated preference domain D_{xij}, ($1 \le j \le P$; $1 \le i \le N$).

At the beginning of the algorithm, load constraint of rest is set to null, then we eliminate the value $\tau_0=0$ from all the preference domains $D_x, \forall x\ (=x_{ij}) \in X$.



(**Fig-1** search tree of solution)

One applies the decision variables ordering heuristic to choose the next decision variable xij to be assigned. It corresponds to the domains Dx that has the minimal sizes not empty: (MinEq1(Dx , $\forall x \in X$ ($Dx \neq \{\phi\}$))). We have Dx11 =3, Dx21 =1, Dx31 =2, Dx41 =2, Dx51 =1, Dx61 =2, Dx71 =1, Dx81 =1 Then MinEq1(Dx $\neq \{\phi\}$, $\forall x \in X$) corresponds to decision variables x21.

Secondary, apply value ordering heuristic to the previous selected variable x_{21}. This heuristic is based on minimum or equal non-null of recent nurses' requirements (load) constraints values: MinEq2 ($C_{rj}\neq 0$, $\forall r\in D_{x21}$). In our case, it corresponds to C_{1j} (morning load constraint). So the value to be assigned for $x_{21}\in X$, is $\tau_k=\langle 1\rangle$ ('creation of node'). In other words, the nurse 'x_{21}' will be affected the morning-shift on day j=1. We decrease the load constraints C_{r1} selected and its satisfaction is checked. If it is, we eliminate or filtering from all $D_x,\forall x\in X$ the $\tau_k=\langle 1\rangle$ value assigned to decision variables selected $x_{21.}$ The same procedure is repeated until the finite set of decision variables X becomes empty. We notice that on day j=2, they are a violation of preference domain for nurse x_{72}, because he is worked on night shift yesterday (i.e. on day j=j-1).

Another remarks: during the execution of the algorithm, the case of empty preference domain ($D_x=\{\phi\}$) can be encountered. At this time one used the backtracking algorithm [11] in the search tree for solution to detect the value that caused blocking and to test another, else one is vis-à-vis the case of equality sizes of preference domains (D_x, $x\in X$) and thus one can exploit the choice in such way to avoid the impact. In the worst case, we violate the preference of nurses D_x and we make use the values $\{\tau_k\}$ of definition domains.

In the following, we compare our algorithm with "LVO" and "MRPO". All the test cases have the same sequence constraints but different or equal load constraints taking into account the set of Nurses' preferences. Tab.6 shows execution time in seconds of different test cases. Our programs were implemented in visual C# 2005 and tested on a Pentium IV based PC.

Time (sec)	Case1	Case2	Case3	Case4
LVO	0.940	0.921	1.215	1.00
MRPO	0.441	0.45	0.491	0.45
Mineq1+Mineq2	0.235	0.298	0.245	0.238

Tab.6 Comparison of different approaches

In the other hand, one notice that our value ordering heuristic "MinEq2" reduces considerably the size of search space given by (see Tab.7): $T=\prod_{i=1}^{N}(D_{X_{ij}}),1\leq j\leq P$.

Size T	Case 1	Case 2	Case 3	Case 4
Size T before	1728	5184	5148	2304
Size T after	2	16	64	4

Tab.7 Comparison of search space size

7. Conclusion

In this paper, we have presented the modelling of Nurse Scheduling Problem in the form of CSP and seen how the usual constraints found can be modelled using the global constraints of Cardinalities and Sequence. A new value ordering heuristic for instantiation of the decision variables when searching for the solution is also defined.

On each day j of work period P (two week) and for each Nurse x_{ij}, ($1\leq i \leq N$; $1\leq j \leq P$) the generated schedule should satisfy Nurse requirement (load) constraints for shift-types (Morning, Afternoon, Rest, Night) and sequence constraints taking into account the set of Nurses preferences D_x. Our test involves N = 8 Nurses on one finite period P=14 days with three work shifts τ_1, τ_2, τ_3 and a rest shift τ_0. The total number of Nurse Requirement (load) constraints and sequence constraints examined is respectively: P*z=56 and N*z=32. Based upon the Nurses' preferences for the shift-patterns our algorithm generates an important number of shift-pattern. It corresponds to (M! * z!) such as M represents the number of equal of preference domain in day j and z the number of shift type. The choice of one shift-pattern corresponds to the objective function.

Experiments show that our algorithm with its new value ordering based heuristic reduces considerably the size of search space during the search for solution and outperforms other common constraint programming technique such as Look-ahead Value Ordering for CSP "LVO" and Meta-level Reasoning and Probability-based Ordering "MRPO" algorithms.

In the forthcoming months, we plan to introduce other kinds of constraints like skill of nurse and we want to test our approach on a set of larger benchmarks.

References

[1] E. BURKE, P. De CAUSMAECKER, S. PETROVIC, G. VANDEN BERGHE, *Variable Neighbourhood Search for Nurse Rostering Problems*, In Mauricio G.C. RESENDE, Jorge PINOH de Sousa (editors), Meta-heuristics: Computer Decision-Making, Capter7, Kluwer, 2003, 153-172.

[2] J.H. OLDENKAMP, Quality in Fives: *On the analysis, Operationalization and application of Nursing Schedule Quality*, University of Groningen, 1996

[3] B. CHEAN. H. LI, A. LIM, B. RODRIGUES, *Nurse Rostering Problems-a bibliographic Survey*, European journal of operational research, www.elsevier.com/locate/dsw, 2003

[4] K.A. DOWSLAND, *Nurse scheduling with taboo search and strategic oscillation*, European Journal of Operational Research 106, 1998, pp 393-407.

[5] K. NONOBE, T. IBARAKI, *A taboo search approach to the constraint satisfaction problem as a general problem solver*, European Journal of Operational Research 106, 1998, pp 599-623.

[6] L. DJEFFAL, M. BABES, *Contribution à l'élaboration d'un algorithme de recherche arborescent par la programmation par contrainte et la programmation orientée objet, « Application Hospitalière »*, Proceeding of Conférence Internationale en Informatique Appliquée CIIA'05, Bourdj-BouArréridj, Algérie, November 2005
[7] S. KUSUMOTO, *Nurse Scheduling Using ILOG Solver*, Proceeding of the second Ilog solver and scheduler users Conference. Paris: ILOG, 1996
[8] S. ABDENNADHER, H. SCHLENKER, *Nurse scheduling using CLP*, AAAI/IAAI, 1999, pp. 838-843.
[9] A.H.W. CHUN, S.H.C CHAN, and al, *Nurse Rostering at the Hospital Authority of HONG–KONG*, AAAI/IAAI, 2000, pp 951-956.
[10] H. MEYER auf'm HOFE, *Nurse Rostering as Constraint Satisfaction with Fuzzy Constraints and inferred control stratégies*, DIMACS, 2001, pp67-99.
[11] C. LECOUTRE, F.BOUSSEMART, F.HEMERY, *Technique de retour arrière intelligent versus heuristiques dirigées par les conflits.* JNPC'04, pp 235-250, 2004.
[12] S. ABDENNADHER and H. SCHLENKER, *INTERDIP-An interactive constraint based nurse scheduler*, in PACLP-99. Available from http://www.pms.informatik.unimuenchen.de/~interdip, 1999
[13] H. MEYER auf'm HOFE, *Nurse rostering as constraint satisfaction with fuzzy constraints and inferred control strategies*, in E.C FREUDER, R.J. WALLACE, constraint programming and large scale discrete optimization, DIMACS Series in Discrete Mathematics and TCS, VOL 75, DIMACS, pp 67- 99, 2001
[14] H. MEYER auf'm HOFE, *ConPlan/SIEDA plan: Personal assignment as a problem of HCSP*, in proceedings of ICPA of Constraint technology, pp 257-272, 1997
[15] N. SADEH and M.S. FOX. *Variable and value ordering heuristics for Activity Based Job-Shop Scheduling*. In proceedings of the fourth International conference on Expert Systems in production and opération Management, pp 134-144, 1990
[16] S. MINTON and al, *Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems*.1994
[17] R. DETCHER and D. FROST. *Look a-head value ordering for constraint satisfaction Problem*. In proceeding of IJCAI'95; pages 572-578, Montréal Canada 1995
[18] D. SABIN and E.C. FREUDER, *Understanding and Improving the MAC Algorithm*. In G. SMOLKA, editor, principle and practice of constraint pprogramming – CP97, LNCS 1330, pp167–181, Springer 1997.
[19] G.Y.C. WONG, A.H.W. CHUN, *Constraint-based rostering using meta-level reasoning and probability-based ordering*. Engineering Applications of Artificial Intelligence, pp599-610, 2004.
[20] G. PESANT, *A Filtering Algorithm for the Stretch constraint*. In proceeding of the seventh ICPP of Constraint programming, pp183-195, 2001
[21] G. PESANT, *A Regular Language Membership Constraint for finite sequence of Variables*, In CP'04, pp482-495, 2004.