Expert Systems with Applications 39 (2012) 385–394

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Using Constraint Satisfaction Problem approach to solve human resource allocation problems in cooperative health services

Cicero Ferreira Fernandes Costa Filho [a,*], Dayse Aparecida Rivera Rocha [b], Marly Guimarães Fernandes Costa [a], Wagner Coelho de Albuquerque Pereira [c]

[a] *Centro de Tecnologia Eletrônica e da Informação, Universidade Federal do Amazonas, Av. General Rodrigo Otávio Jordão Ramos, 3000, Aleixo, Campus Universitário – Setor Norte, Pavilhão Ceteli, CEP: 69077-000, Manaus, AM, Brazil*

[b] *Fundação Centro de Análise, Pesquisa e Inovação Tecnológica – FUCAPI, Av. Danilo Areosa, 381, Distrito Industrial, Manaus, AM, Brazil*

[c] *Programa de Engenharia Biomédica, COPPE/UFRJ, Caixa Postal 68510, CEP 21941-972 Rio de Janeiro, RJ, Brazil*

ARTICLE INFO

Keywords:
Human resource allocation
Cooperative services
Heuristics
Constraint Satisfaction Problem
Backtracking search algorithm

ABSTRACT

In developing countries, the increasing utilization of health services, due to a great life expectancy, is followed by a reduction in incomes from the public health system and from private insurance companies, to the payment of medical procedures. Beyond this scenery, it is mandatory an effective hospital cost control though the utilization of planning tools.

This work is intended to contribute to the reduction of hospital costs, proposing a new tool for planning human resources utilization in hospital plants. Specifically, it is proposed a new tool for human resources allocation in health units. The solution to the allocation problem uses the CSP technique (Constraint Satisfaction Problem) associated with the backtracking search algorithm. With the objective of enhancing the backtracking search algorithm performance a new heuristics is proposed. Through some simulations the performance of the proposed heuristics is compared to the other heuristics previously published in literature: remaining minimum values, forward checking and grade heuristics.

Another important contribution of this work is the mathematical modeling of the constraints, that could be unary, multiple, numeric and implicit constraints. In the results it is presented a case study of a human resource allocation in a cooperative health service.

Based on the results, it is proposed that for a real allocation problems solution, the best approach is to combine the remaining minimum values heuristics with the grade heuristics, to select the best unit allocation to be filled, and then use the proposed heuristic to select the best physician to the chosen unit allocation. This association shows a satisfactory result for the human resource allocation problem of the case study, with an algorithm convergence time of 46.7 min with no backtracks. The same problem when manually resolved took about more than 50 h.

1. Introduction

The infrastructure involved in providing medical services is complex and expensive, and encompasses both human resources and equipments; therefore, it needs an adequate resource management to attain profitable results. As stated by Spyropoulos (2000), a hospital infrastructure is composed of:

(a) Human resources: physicians, nurses, administrative personal, technicians for equipment maintenance, etc.
(b) Intensive therapy units, with an expensive infrastructure.
(c) Surgery room, with dedicated equipments for several procedures.
(d) Specialized laboratories: X-rays, ultrasound, tomography, magnetic resonance, etc.
(e) Auxiliary infrastructure: ambulance for emergency transfer, patient's rooms, pharmacy, restaurant, etc.

In recent decades many tools with the aim of providing efficient management of this infrastructure have been proposed.

Oddi and Cesta (2000) considered that managers of medico-hospital facilities are facing two general problems when allocating resources to activities: (1) to find an agreement between several and contrasting requirements; (2) to manage dynamic and uncertain situations when constraints suddenly change over time due to medical needs. This paper describes the results of a research aimed at applying constraint-based scheduling techniques to the

* Corresponding author.
E-mail address: cicero.costa@uol.com.br (C.F.F. Costa Filho).

0957-4174/$ - see front matter
doi:10.1016/j.eswa.2011.07.027

management of medical resources. A mixed-initiative problem-solving approach is adopted in which a user and a decision support system interact to incrementally achieve a satisfactory solution to the problem of allocating resources to medical activities. The authors claim two main contributions of the paper. The first one concerns the domain modeling. The medical problem is represented as a Constraint Satisfaction Problem (CSP) (Tsang, 1996), hence described as a set of variables and a set of constraints on the values of the variables. A solution to the problem is a variable assignment which is compatible with all the constraints. Two main objects are represented in this schema: medical protocols and resources. The constraints are classified as relaxable or non-relaxable. The solution represents an agreement between different and contrasting goals by reducing the total amount of violations of non-relaxable constraints. A second contribution is the introduction of a new solution algorithm, in which two types of algorithm are integrated: a greedy procedure to create an initial solution and a local search method to improve the initial solution with respect to the amount of violations contained in it. The local method used is tabu-search.

Valouxis and Housos (2003) presented a detailed model and an efficient solution methodology for the monthly work shift and rest assignment of hospital nursing personnel. A model that satisfies the rules of a typical hospital environment based both on published research data and on local hospital requirements is designed. A hybrid methodology that utilizes the strengths of operational research and artificial intelligence was used for the solution of the problem. In particular, an approximate integer linear programming (ILP) model is firstly solved and its solution is further improved using local search techniques, as tabu-search strategy.

Other papers that address the problem of nurse allocation were published by Weil, Heus, François, and Poujane (1995), Oughalime, Ismail, and Yeun (2008), Tsai and Li (2009) and Dowsland (1998). The nurse area presents special characteristics that allow the use of automatic human resource allocation systems:

(a) There is a great number of actors that perform the same task in the hospital and, therefore, can be changed one by another without any impairment.
(b) There are some restrictions for allocation timing, due to profession regulation and hospital requirements.

Concerning physicians, nevertheless, the first of these conditions is not observed. The number of physicians in each specialty is significantly lower than the number of nurses. This fact implies that the effort needed to accomplish an automatic physician allocation is lower than the one needed to accomplish an automatic nurse allocation. In some Brazilian metropolis, otherwise, a special situation concerning physicians' services brings this condition into focus. The physician services are provided through medical cooperatives. Nowadays, in the city of Manaus (State of Amazonas, Brazil), for example, there are about two thousand cooperated physicians. These cooperatives (gathering pediatricians or anesthesiologists or obstetricians, etc), which comprehend between fifty and two hundred physicians, provide services to about twenty public hospitals, including emergency units, pediatric units, etc. Depending on the cooperative, the service is provided on time periods of 8, 12 or 24 h. In some units it is needed more than one physician/specialty in one time period. The cooperatives schedules are made in a monthly period. In this scenario, as with the nurse professionals, the first condition previously reported is satisfied. A manually solution to this allocation is a hard task, that demands a large time. In the present paper it is proposed an automatic solution to this allocation problem.

Frequently, two models are used to obtain an automatic solution to human resource allocation problem. In the first one the allocation is viewed as an optimization problem. A state concept is defined for the problem and is created a cost function that attributes a value for each state. Each of these states is a complete attribution to the problem: each vacancy of the schedule is filled with one cooperated physician. The cost function building considers the constraints established to the problem. Each constraint represents a term in the cost function, multiplied by a weight factor. So, the cost function is a linear combination of weighted constraints, as shown in Eq. (1).

$$c = k_1 r_1 + k_2 r_2 + \cdots k_n r_n \tag{1}$$

where r_i is the number of times a constraint is violated in a state of the problem; k_i the weight factor of the constraint. The more important is the constraint, the higher is the weight factor. The optimization method tries to find a state where Eq. (1) is a minimum value. The exploration of the state space can be done using one of the following methods: tabu-search, hill climbing, genetic algorithm, simulated annealing or any other local method. The search process consists of generating new states from old states, obeying defined rules in each one of these methods.

The second model uses an artificial intelligence technique, entitled Constraint Satisfaction Problem – CSP. Differently from the first method that begins with a complete attribution, CSP initiates with an empty attribution: no physicians addressed to any vacancy in the schedule. The attributions of physicians to vacancies are incremental, one each time. Each attribution is confirmed only if no constraint violation is verified. When this is not possible, the technique goes back to the last attribution and searches other possibilities to the new attribution. The use of the CSP technique is associated with problem modeling and choice of a search algorithm.

In the modeling stage are defined the variables, their domains and the problem constraints. One contribution of this paper is the mathematical modeling of the problem constraints that can be unary, multiple, numeric and implicit ones. Besides the theoretic work, it was developed a computational tool that enables the technique implementation. The main characteristic of this tool is the flexibility in constraint programming, allowing search solutions for different problems of similar nature. Moreover, it is possible the free insertion of the following information: duration of a period (8 12 or 24 h), duration of the schedule (one month, one week, etc), name of the health unit and number of physicians needed in each health unit/period, name of physicians involved in the allocation problem and number of periods of each physician. All this information is registered in the first block of Fig. 1.

In this paper, to solve the allocation problem, we use the CSP approach associated with the backtracking search algorithm (Russel & Norvig, 2004). This algorithm encompasses a depth search, which, in its turn, is time-consuming, because it explores all the state space looking for a solution. Using heuristics associated with a depth search algorithm can speed up the solution search. Another intended contribution of this paper is the proposal of a new heuristics to the backtracking search algorithm, entitled *domain verification heuristics*. Its performance is compared with

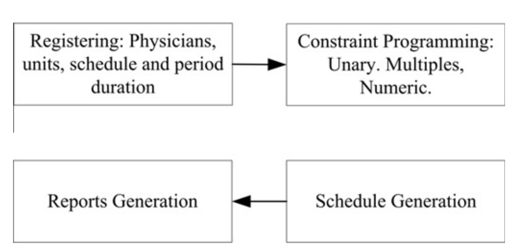

Fig. 1. Block diagram of the developed tool.

other ones previously reported in the literature, as *minimum remaining values*, *grade heuristics* and *forward checking*.

In next section we will discuss about constraints programming and how the schedule is generated using the backtracking algorithm (blocks 2 and 3 of Fig. 1). The logical modeling of this tool was made using Unified Modeling language.

2. Materials and methods

At the beginning of this section it is presented the allocation problem modeling using the CSP technique. It is described the variables, their domains and the proposed constraints. Following, it is presented the backtracking search algorithm with the necessary customization to include the proposed heuristics. As the last topic the heuristics evaluated are presented, with special attention to the *domain verification heuristics* proposed in this paper.

2.1. Problem modeling – Variables

In a health unit the human resource allocation is generally done monthly. Each health unit presents specific variables like the number of physicians needed for a period and the duration of the period (normally 8, 12 or 24 h).

In this paper the variables of the allocation problem are represented as the dimensions of an allocation matrix, $M(d, p, h, v)$, where: month day – d, period – p, health unit – h and vacancies in one period – v. The ranges of values that these variables can assume are:

- d – month day, $d \in D = \{1, 2, 3, \ldots, n_1\}$, where $n_1 \leqslant 31$;
- p – period, $p \in P = \{p_1, p_2, \ldots, p_{n2}\}$, where n_2 is the number of periods. For example, with periods of 12 h, $P = \{p_1, p_2\}$, $n_2 = 2$.
- h – health unit, $h \in H = \{h_1, h_2, \ldots, h_{n3}\}$, where n_3 is the number of health units;
- v – number of vacancies in one period, $v \in V = \{v_1, v_2, \ldots, v_{n4}\}$, where n_4 is the number of physicians needed in one period.

Each element of the matrix M is one allocation unit. The physicians set that can be allocated to an unit $M(d_i, p_i, h_i, v_i)$ (M_i) is the domain S_i of this unit, defined as:

- $S_i = \{s_1, s_2, \ldots, s_{n5}\} \subset S$, where s_n is a physician that can be allocated to the unit M_i, S is the set of all physicians and n_5 the number of physicians that can be allocated to unit M_i.

The solution to an allocation problem is the attribution of a physician s_n for each allocation unit M_i.

2.2. Problem modeling – constraints

The constraints used in this paper are classified in one of the following types: unary, multiple, numeric and implicit constraints.

The unary constraints limit the set of physicians that can be attributed to the domains of one or more allocation units. So, for one allocation unit M_1 it is specified the subset S_1 of physicians that can be attributed to it. The unary constraints could be expressed as the following statement:

Unary constraint statement: "For a unit M_1 defined as: $d \in D_1 = \{d_1, d_2, \ldots, d_x\} \subset D$, $p \in P_1 = \{p_1, \ldots p_n\} \subset P$ and $h \in H_1 = \{h_1, h_2, \ldots, h_w\} \subset H$, $v \subset V$, the physicians that can be attributed to them are limited by $s \subset S_1$".

The specification of a unary constraint is done by defining the sets D_1, H_1, T_1 and S_1. The v variable is not taken into account, because it does not comprehend information of time or place. When more than one unary constraint limits the domain of one unit, the resulting domain is obtained through an intersection operation among the S_k sets of these constraints. For example, if constraints R_1 and R_2 limit the domain of one unit M_i, through the sets S_1 and S_2, respectively, the final domain of the unit is obtained from the intersection $S_1 \cap S_2$. In the sequence it is illustrated the modeling of a real situation through the use of a unary constraint.

Situation 1: physicians s_1 and s_2 could not be allocated in unit 1 in the first period of the second day of the month.

Modeling: this situation is modeled through the use of the constraint 1 shown in the sequence.

Constraint 1: $D_1 = \{2\}$, $H_1 = \{\text{unit } 1\}$, $P_1 = \{p_1\}$ and $S_1 = \{s_3, s_4, \ldots s_n\}$ or $S_1 = S - \{s_1, s_2\}$.

Unary constraints have a specific user interface to be registered in (Fig. 2). In the example of Fig. 2, the unary constraint 1 is registered with the number 36. The set S_1 was as the complement of $\{s_1, s_2\}$: $S - \{s_1, s_2\}$. The term *yes* in the *complement field* of the form indicates this complement operation. The availability of the complement simplifies large domain registrations.

Multiple constraints establish a relationship between two or more allocation units. While unary restrictions limit the variables domain, multiple constraints verify possible conflicts that could occur between a new value attributed to one allocation unit and previous values attributed to other allocation units. Unary constraints act before the allocation process begins, while multiple constraints act while the allocation process takes place.

The multiple constraints used in this paper are of two kinds as expressed in the following statements:

Statement of type 1 *multiple restriction: "For n units* $M(d_1, p_1, h_1, v), M(d_2, p_2, h_2, v), \ldots, M(d_n, p_n, h_n, v)$*, it must be* $s_1 \neq s_2 \neq \cdots s_n$*".*

Statement of type 2 *multiple restriction: "For n units* $M(d_1, p_1, h_1, v), M(d_2, p_2, h_2, v), \ldots, M(d_n, p_n, h_n, v)$*, it must not be* $s_1 = s_2 = \cdots = s_n$*".*

A multiple constraint specification is done by defining the units and the constraint type. In the following it is given two examples of multiple constraints modeling.

Situation 2: the physicians allocated in December, 24th could not be allocated in December, 31st.

Modeling: this situation is modeled through four multiple constraints of type 1: Constraints 2, 3, 4 and 5. In this modeling are considered periods of 12 h.

Constraint 2: $M(24, 1, h, v)$; $M(31, 1, h, v)$;
Constraint 3: $M(24, 1, h, v)$; $M(31, 2, h, v)$;
Constraint 4: $M(24, 2, h, v)$; $M(31, 1, h, v)$;
Constraint 5: $M(24, 2, h, v)$; $M(31, 2, h, v)$,

where h stands for any health unit and v stands for any vacancy.

Situation 3: no physician could give three following periods.

Modeling: this situation is modeled through two type 2 multiple constraints, constraints 6 and 7.

Constraint 6: $M(d, 1, h, v), M(d, 2, h, v), M(d + 1, 1, h, v)$;
Constraint 7: $M(d, 2, h, v), M(d + 1, 1, h, v), M(d + 1, 2, h, v)$,

where d stands for any day of the month, h stands for any health unit and v stands for any vacancy.

As the multiple constraints, the numeric constraints act while the allocation process takes place. The numeric constraints used in this paper are classified also in two types. The first one fixes the number of periods of one physician in a schedule. The second type fixes an equal number of periods of a group of physicians in a group of units. Following are the statements of these two constraints:

Statement of type 1 *numeric constraint: "In a schedule the number of periods of physician* s_i *is* x_i*".*

Statement of type 2 *numeric constraint* "For any unit $M(d, p, h, v)$ *defined as:* $d \in D_1 = \{d_1, d_2, \ldots, d_x\} \subset D$, $p \in P_1 = \{p_1, \ldots p_n\} \subset \mathrm{P}$ *and*

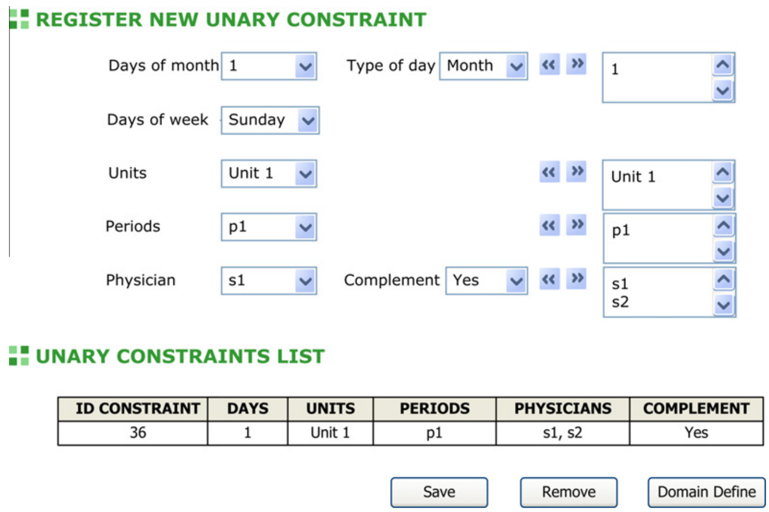

Fig. 2. Interface for unary restrictions registry.

$h \in H_1 = \{h_1, h_2, \ldots, h_o\} \subset H$, each physician of the set $S_1 = \{s_1, s_2, \ldots, s_m\}$ has an equal number of periods".

The specification of type 1 numeric constraint is done fixing the number of periods of each physician. In the developed tool this is done when the physician is registered. The specification of type 2 numeric constraint is done defining the subsets D_1, P_1, H_1 and S_1. Following is given one example of type 2 numeric constraint modeling:

Situation 4: the physicians s_1, s_2, s_3, s_4, $\ldots$ s_{20} must have the same number of periods during Saturdays and Sundays.

Modeling: this situation is modeled through one numeric constraint of type 2: constraint 8. In this modeling we consider periods of 12 h.

Constraint 8: D_1 = {*set of the days of the month corresponding to saturdays and sundays*}

$P_1 = \{1, 2\}$, $H_1 = H$, $S_1 = \{s_1, s_2 \ldots, s_{20}\}$.

The implicit constraints represent physical constraints of the problem. In the allocation problem modeled in this paper, the implicit constraint used is described in the following statement:

Statement of the implicit constraint: "No physicians can have two periods in the same day and at the same time".

The implicit constraints are defined during the development of the algorithm. The user can not change this constraint.

2.3. Backtracking search algorithm

To ease the visualization of the solution search process, the allocation problem, as shown in Fig. 3, can be organized as a tree-space state [Deris et al, 2000]. The tree levels correspond to the allocation units of the schedule, M_i, where $1 \leqslant j \leqslant n$. Each level j has m sons that correspond to the m possible attributions to M_i. The problem solution can be represented by a vector $S = [s_1, s_2, \ldots, s_n]$, where s_j corresponds to a M_i attribution. The vector S is a way that connects the level of the root node to the leaves level of the tree. As each node has the same number of sons, the size of the search space is given by m^n, where m is the number of physicians to be allocated. There are some methods to search a solution. The one used in this paper is the backtracking search algorithm. In Fig. 4, it is shown the block diagram of this algorithm, adapted to work with the heuristic procedures used in this paper. The main characteristics of this algorithm are the following: (1) Uses an increase sequence of partial attributions until reaching a complete attribution; (2) Uses a recursive procedure that backtracks when a partial attribution violates any problem constraint.

2.4. Heuristics

Due to the size of the search space, m^n, the solution search was accelerated through the individual or combined use of some of the following heuristics: minimum remaining values – MRV (Kumar, 1992), grade heuristics (Baptiste, 2001), forward checking – FC (Tsang, 1993) and the new heuristics proposed in this paper, the *domain verification* – DV.

The MRV heuristics selects the allocation unit to be attributed as the one with lower number of physicians in its domain. The explored idea is that this allocation unit has the higher probability of failure in its attribution, causing a backtracking in the search tree. This heuristics is applied in the moment of one allocation unit attribution. In Fig. 4, this is done in the block entitled "select allocation unit not attributed". The allocation unit with the lowest number of physicians in its domain is selected. The physician to be allocated is chosen in a random way.

An inconvenience with this heuristics usage occurs when a set of allocation units has the same number of physicians in their domains. This situation is frequently found in the beginning of a search process. This problem was softened through the use of unary constraints and of the grade heuristics.

The grade heuristics is responsible for selecting the allocation unit that is involved in the greatest number of multiple constraints. In Fig. 4, this heuristics is applied together with the MRV heuristics. When there is equality between two or more allocation units after the application of MRV heuristics, the grade heuristics' is used as an inequality criterion.

The FC heuristics optimizes multiple constraints usage in the search process. When a physician is attributed to a allocation unit M_x, the forward checking analyzes each allocation unit not yet filled that is connected through multiple constraints to M_x and excludes from the domain of these units any physician that is not consistent (physician that violates a constraint) with the value attributed to M_x. In the block diagram of Fig. 5, it is presented the insertion of the FC heuristics (traced-contour boxes) in the block diagram of Fig. 4.

The DV heuristics proposed in this paper instead of being concerned with the allocation unit to be filled, as is the case of the MRV and grade heuristics, is focused on the physician to be attributed. In this sense, it acts at the level "attribute physician to allocation unit" shown if the block diagram of Fig. 4. This heuristics consists of selecting inside a unit domain, the physician with the lowest number of attributions given. As a decision criterion it is

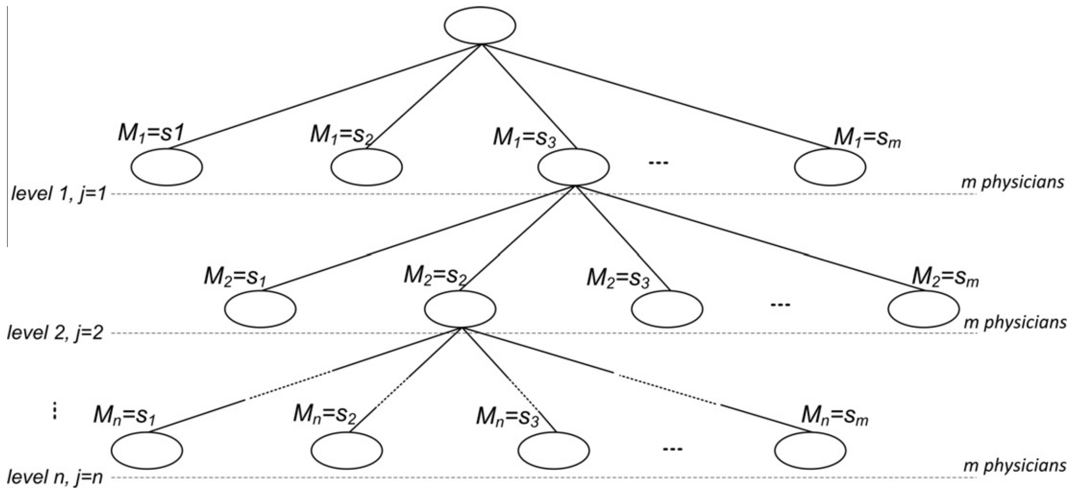

Fig. 3. Search space at an allocation problem.

selected the physician that is present in the smallest number of allocation unit domains. The unit to be allocated is chosen in a random way.

The logic behind this heuristics is to reduce the probability of all the attributions of a given physician to be made at the end of the search process, where the possibility of conflicts is higher. In Fig. 6, it is illustrated an application of DV heuristics. It is assumed the need to attribute physicians s_1, s_2, and s_3 to six allocation units. Each physician is allocated in two periods. It is shown the domains of the allocation units. Fig. 6 presents the moment when the attribution of the fourth allocation unit is to be decided. Applying the first criterion of the DV heuristics is not enough to choose between the three physicians. Nevertheless, through the application of the second criterion, physician s_1 is chosen, because he is found in the domain of the lowest number of allocation units that remain with no attribution.

3. Results

Initially it is shown the results concerning experiments made to evaluate the performance of the backtracking search algorithm, associated with each one of the heuristics just mentioned. Following, it is presented a case study of a physicians' cooperative of the city of Manaus, State of Amazonas, Brazil that renders services to public health units. The experiments were performed in a personal computer, with 1.6 GHz Intel Core Duo, CPU, 1536 MB RAM and Windows Vista Operational System. The performance of the algorithm was evaluated according to two criteria: (1) the number of allocation units used in the solution search, simply called "units". For this parameter calculation it was considered not only the number of units to be allocated but also the number of backtracks that occurs in the search algorithm. In this sense, if a problem is constituted of 70 allocation units and there were 24 backtracks, the parameter "units" is, in this case, equal to 94; (2) The convergence time of the algorithm, simply called "time".

The following evaluation situations were created with the utilization of (i) no constraints; (ii) unary constraints; (iii) multiple constraints and (iv) both kinds of constraints. In each of these situations the following simulations were done with the utilization of: (i) no heuristics; (ii) MRV and Grade heuristics; (iii) FC heuristics; (iv) DV heuristics; and (v) with all heuristics present. The following parameters of the schedule were set: two periods, one vacancy/period/unit and 35 physicians. For each experiment were obtained the mean value (μ) and the standard deviation (σ) of 40 simulations.

The numeric constraints used were of type 1. The type 2 numeric constraints were used only in the case study. Following are detailed the evaluations of the four situations created. In Table 3, Tables 5 and 6 when it is said that the algorithm does not converge, it means that the convergence does not occur in a period of three hours (after which the simulation process is aborted).

3.1. Situation 1: backtracking search algorithm performance evaluation with no constraints

It was made an attempt of evaluating the backtracking search algorithm varying the complexity of the allocation problem. In this attempt the number of health units was considered a variable. Three different scenarios were created with 5, 10 and 15 health units. In the first one, with 5 health units, there are 70 allocation units to be filled. Each physician must have two periods per week. In the second one, there are 140 allocation units to be filled. Each physician must have four periods per week. In the third one, there are 210 allocation units. Each physician must have 6 periods per week. In Table 1, it is shown the results obtained for this first situation.

3.2. Situation 2: backtracking search algorithm performance evaluation with unary constraints

Unary constraints used are shown in Table 2. Five different scenarios were created. Scenario 1: that uses unary constraints from one to five; Scenario 2: with unary constraints from one to eight; Scenario 3: with unary constraints from one to ten; Scenario 4: with unary constraints numbered from one to twelve; Scenario 5: with unary constraints from one to sixteen. The number of health units was equal to ten. In Table 3 it is shown the results for this second situation.

3.3. Situation 3: backtracking search algorithm performance evaluation with multiple constraints

Multiple constraints used are shown it Table 4. Three different scenarios were created. Scenario 1: that uses the multiple constraint 1; Scenario 2: with multiple constraints 1 and 2; Scenario 3: with multiple constraints 1, 2 and 3. The number of health units

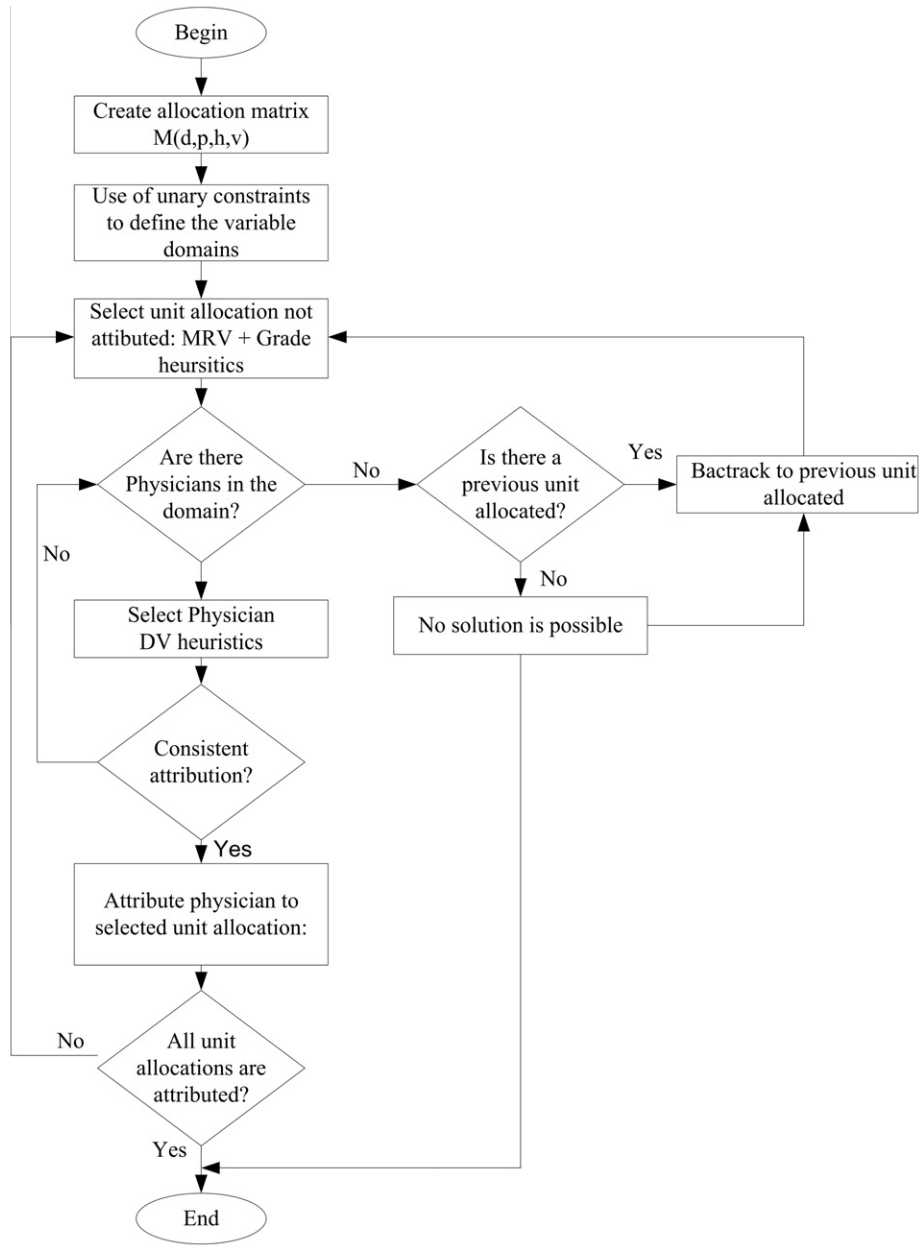

Fig. 4. Block diagram of the backtracking search algorithm, adapted to work with the heuristic procedures used in this paper.

was equal to ten. In Table 5 is shown the results for this third situation.

3.4. Situation 4: backtracking search algorithm performance evaluation with unary and multiple constraints

Three different scenarios were created. Scenario 1: that uses unary constraints from one to sixteen with multiple constraints 1; Scenario 2: with unary constraints from one to sixteen with multiple constraints 1 and 2; Scenario 2: with unary constraints from one to sixteen with multiple constraints 1, 2 and 3. The number of health units was equal to ten. In Table 6 is shown the results for this fourth situation.

3.5. Case Study

The case study refers to a physician cooperative located at the city of Manaus that renders services to public state health units. The cooperative is formed by 128 physicians initially divided in three categories, according to the affiliation time of the members: (i) "new", with less than five years; (ii) "intermediate", between five and ten years; and (iii), "old" with more 10 y.

The 128 physicians are then divided in four groups, according to the numbers of working periods (12 h): 25 new physicians with 10 periods per month; 10 old physicians with 16 periods per month; 59 physicians (intermediate) with 21 periods per month and 34 physicians (intermediate) with 20 periods per month. The number of health units is 20. Table 7 presents the number of physicians required in each unit, day and period. The schedule is to be planned to August, 2009, performing 1240 allocation units and 2329 vacancies. The 25 new physicians must be allocated only in weekends in complex units: unit 1, 2, 3, 4, 5, and 6. The 10 old physicians must be allocated from Monday to Friday in non-complex units. The intermediate physicians must be allocated in the same number of periods in weekends. No physician can be allocated in more than three consecutive periods.

The following constraints were used in modeling this case study:

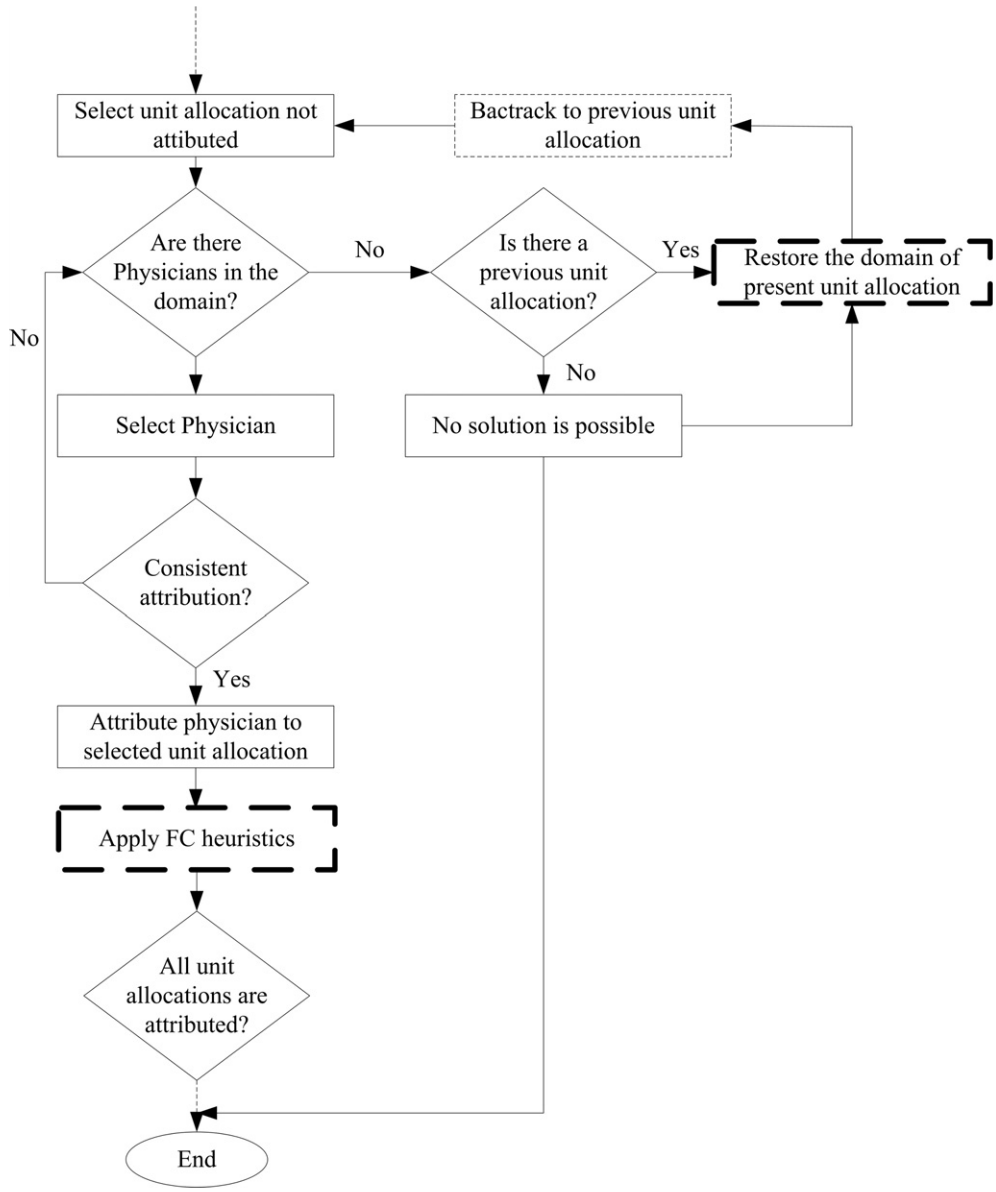

Fig. 5. Block diagram showing the insertion of the forward checking heuristics in block diagram of the backtracking search algorithm of Fig. 4.

Unary constraint 1: D_1 = {set of the days of the month corresponding to Saturdays and Sundays}, $P_1 = P$, $H_1 = H$ and $S_1 = S$ – {old physicians};
Unary constraint 2: D_1 = {set of the days of the month corresponding to Mondays, Tuesdays, Wednesdays, Thursdays and Fridays}, $P_1 = P$, H_1 = {complex units}, S_1 = S – {{new physicians} ∪ {old physicians}};
Unary constraint 3: D_1 = {set of the days of the month corresponding to Mondays, Tuesdays, Wednesdays, Thursdays and Fridays}, P_1 = P, H_1 = {non complex units}, S_1 = S – {new physicians};
First Multiple constraint type 1: $M(d, 1, h, v)$, $M(d, 2, h, v)$, $M(d + 1, 1, h, v)$;
Second multiple constraint type 1: $M(d, 2, h, v)$, $M(d + 1, 1, h, v)$, $M(d + 1, 2, h, v)$;
Numeric constraints type 1: were used in modeling the number of periods of each physician;
Numeric constraint type 2: D_1 = {set of the days of the month corresponding to Saturdays and Sundays}, $P_1 = P$, $H_1 = H$ and $S_1 = S$ – {{old Physicians} ∪ {new physicians}};

The backtracking search algorithm was used associated with the heuristics that have shown the best results in the simulations of Situation 3 and 4: (1) DV heuristics; (2) MRV + Grade; (3) MRV + Grade + DV. An attempt to use the FC heuristics and all heuristics together was made; nevertheless, due to the high memory usage and the large algorithm convergence time, it was not a viable option. The performance of the algorithm was measured with the criteria "units" and "time" already defined. The best result was with the association of the three heuristics: units = 2239 and time = 46.7 min. A trained professional took about one week (40 h) to manually do the same schedule.

The following outputs are provided by the developed tool: a complete list of all health units with the name of the physician allocated for each one; a list of health units for each physician; a list of physicians allocated to each health unit.

For intermediate physicians the following results were obtained concerning the numeric constraint type 2: to 92 physicians it was allocated 4 periods in weekends and for only one physician it was allocated 3 periods in weekends.

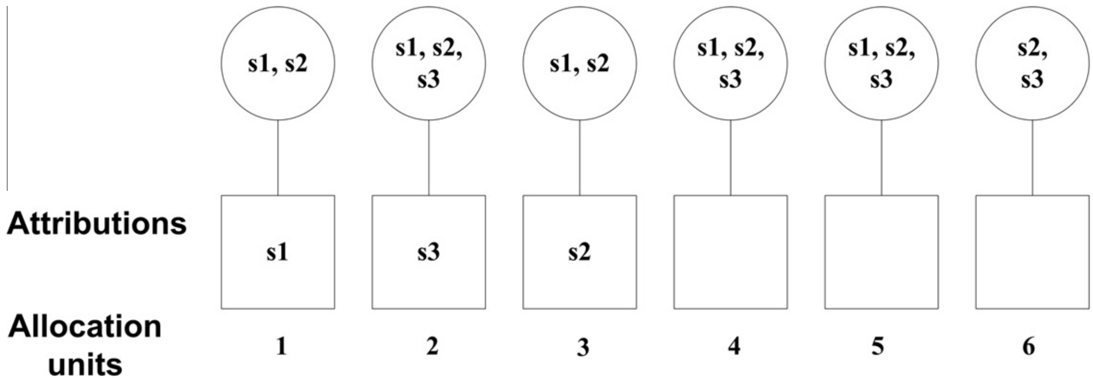

Fig. 6. Example of application of the domain heuristics proposed in this work.

Table 1
Performance of backtracking algorithm with no constraints (Situation 1).

Heuristics	Scenario											
	1				2				3			
	Units		Time (s)		Units		Time (s)		Units		Time (s)	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
No heuristics	70	0	9	1	140	0	25	7	244	63	62	26
MRV + Grade	77	11	10	1	141	1	23	0	211	2	53	11
FC	70	0	147	2	140	1	757	16	271	133	2109	115
DV	70	0	11	0	147	8	29	2	214	6	51	2
MRV + Grade + DV	70	0	12	0	140	0	27	9	214	5	60	3
All	70	0	150	1	140	0	809	5	210	0	2397	47

Table 2
Unary constraints used for performance evaluation of backtracking search algorithm.

Unary constraint	Description
1	Physician 1 can work only at first period
2	Physicians 2 and 3 can not work at Saturdays on second period
3	Physicians 4 and 5 can not work day 22 at unit 1
4	Physicians 6 and 7 can not work on mondays
5	Physicians 8, 9 and 10 can not work day 24, unit 3 on first period
6	Physicians 11 and 12 can not work day 25 at unit 4
7	Physicians 13 and 14 can not work day 26 at unit 9 on first period
8	Physician 15 can not work day 27 at unit 5
9	Physicians 16, 17 and 18 can not work day 28 at unit 6 on second period
10	Physicians 19 and 20 can not work day 22 on unit 7
11	Physicians 21, 22 and 23 can not work day 23 at unit 8 on first period
12	Physicians 24 and 25 can not work day 24 on unit 9
13	Physicians 26,27 and 28 can not work day 25 at unit 10 on first period
14	Physicians 29 and 30 can not work day 26 at unit 1
15	Physicians 31 and 32 can not work day 27 at unit 2 on second period
16	Physicians 33, 34 and 35 can not work day 28 at unit 3

4. Discussion

This paper describes the human resources allocation applied to a health area as a CSP problem. This approach requires constraints modeling of the desired application area. It was also studied the behavior of backtracking algorithm when used to solve the CSP problem and associated with some heuristics proposed in literature and with a new heuristics proposed in this paper, named *domain verification* heuristics.

The CSP technique was also employed by Weil et al. (1995) to human resources allocation in nursing area. Other papers, like the one of Oughalime et al. (2008) and the one of Dowsland (1998), used tabu-search to solve the same problem. There are some differences between the solution employed by Weil et al. (1995) and the one now presented, concerning problem modeling, application objective, the used heuristics and the developed tool.

Concerning the problem modeling, the main differences are the problem variables and their domains. In the last mentioned paper the problem variables were the pair (day, nurse). The domain of allocation units are the values: 0 – nurse with no activity in the day, 1 – nurse with activity in the morning period, 2 – nurse with activity in the afternoon period, 3 – nurse with activity in the night period. In the present paper the problem variables are the quadruple (day, period, health unit, vacancy). The domain of an allocation unit is composed by the physicians that can work in the unit. The constraints are also different and obey the correspondent models. In the paper from Weil et al. the constraints are classified as optional or compulsory, whereas in this paper the constraints are classified as unary, multiple, numeric or implicit.

Concerning the application, the mentioned paper works with only one unit, whereas the present paper works with several ones. In both papers the backtracking algorithm was employed, with the difference that in the mentioned paper this algorithm was associated with MRV and Arc-consistency heuristics, while in the present paper this algorithm was associated with MRV, Grade and DV heuristics (this last one proposed in this work). In the present paper the performance of all the heuristics was studied concerning algorithm convergence time and number of allocation units used. Only after this study was performed, it was chosen the best set of heuristics to be used in the case-study.

Concerning the developed tool, the mentioned paper used a Ilog-Solver Library, that is a library of the Le-Lisp and C++ languages, designed to solve CSP problems. The present paper used

Table 3
Performance of backtracking search algorithm with unary constraints (Situation 2).

Heuristics	Scenario																			
	1				2				3				4				5			
	Units		Time (s)		Units		Time (s)		Units		Time (s)		Units		Time (s)		Units		Time (s)	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
No heuristics	150	10	31	4	168	48	34	12	140	8	28	0.2	160	39	32	11	152	15	30	4
MRV + Grade	142	5	28	1	141	13	28	1	144	13	29	2	143	12	28	1	142	6	28	1
FC	170	68	919	37	140	1	880	7	NC	–	–	–	141	1	890	23	377	528	1058	375
DV	141	9	28	1	146	14	29	4	145	28	29	3	141	12	27	1	141	15	27	1
MRV + Grade + DV	140	57	37	2	140	45	35	0	144	39	38	3	148	37	37	5	144	19	35	3
All	140	0	997	9	140	0	1039	5	140	0	1057	9	140	0	1095	9	NC	–	–	–

NC: no convergence.

Table 4
Multiple constraints used for performance evaluation of backtracking search algorithm.

Multiple constraint	Description
1	No physician can be scheduling for more than three consecutive periods
2	The physicians who have being scheduling on Saturday can not be schedule for next Sunday
3	The physicians who have being scheduling in the first period on Tuesday can not be schedule for the second period on Thursday

general-purpose tools as Java, MySQL and Tomcat. A special characteristic of the developed tool presented in this paper is the flexibility offered to the user. New constraints can be inserted or removed whenever needed, making it possible the employment of this tool to solve different problems (of similar nature) with no need that a new code be written. In papers already published concerning nurse scheduling, this flexibility is not present in the work of Weil et al. (1995) and in the work of Dowsland (1998). In the work of Oughalime et al. (2008), nevertheless, it is present.

The results for the four simulated situations suggest that: (1) When using the backtracking algorithm with no constraints or with unary constraints, the performance when using no heuristics, or the DV heuristics, or MRV + Grade heuristics, or MRV + Grade + DV heuristics is equivalent when considering the parameter "time". On the other side, the performance of this parameter when considering the FC heuristics or of all combined heuristics is not adequate, because convergence takes too long. The performance of the algorithm when considering the parameter "units" is better when all heuristics are combined. (2) When using the backtracking algorithm with multiple constraints or with multiple and unary constraints together, the DV heuristics was the only one that converges in all situations. The best results are obtained when using the DV heuristics, the association of MRV + Grade heuristics and the association of MRV + Grade + DV heuristics. The FC heuristics and the association of all heuristics converge in a few situations, with large convergence times.

Table 5
Performance of the backtracking search algorithm with multiple constraints (Situation 3).

Heuristics	Scenario											
	1				2				3			
	Units		Time (s)		Units		Time (s)		Units		Time (s)	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
No heuristics	NC	–	–	–	NC		–	–	NC	–	–	–
MRV + Grade	145	8	65	6	150	16	70	15	NC	–	–	–
FC	NC	–	–	–	NC	–	–	–	196	97	2478	137
DV	135	10	64	4	157	27	75	26	165	6	76	7
MRV + Grade + DV	126	11	58	4	NC	–	–	–	143	4	65	27
All	140	2	1514	12	NC	–	–	–	140	0	2709	25

NC: no convergence.

Table 6
Performance of the backtracking search algorithm with unary and multiple constraints (Situation 4).

Heuristics	Scenario											
	1				2				3			
	Units		Time (s)		Units		Time (s)		Units		Time (s)	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
No heuristics	NC	–	–	–	NC	–	–	–	NC	–	–	–
MRV + Grade	148	15	64	7	192	88	107	70	NC	–	–	–
FC	264	237	2736	552	NC	–	–	–	NC	–	–	–
DV	140	1	59	1	173	56	93	53	143	6	74	9
MRV + Grade + DV	161	28	73	18	185	14	101	13	NC	–	–	–
All	NC	–	–	–	NC	–	–	–	NC	–	–	–

NC: no convergence

Table 7
Case study – number of physicians needed per unit/day/period.

Unit	Period	Week day						
		Mon	Tue	Wed	Thr	Fri	Sat	Sun
1	1°	4	4	4	4	4	1	1
	2°	0	1	1	1	0	1	1
2	1°	4	4	4	4	4	4	4
	2°	3	3	3	3	3	3	3
3	1°	5	5	5	5	5	1	1
	2°	1	1	1	1	1	1	1
4	1°	3	3	5	3	3	0	0
	2°	0	0	0	0	0	0	0
5	1°	3	3	3	3	3	3	3
	2°	2	2	2	2	2	2	2
6	1°	4	4	4	4	4	4	4
	2°	4	4	4	4	4	4	4
7	1°	2	2	2	2	2	2	2
	2°	2	2	2	2	2	2	2
8	1°	2	2	2	2	2	2	2
	2°	2	2	2	2	2	2	2
9	1°	2	2	2	2	2	1	1
	2°	1	1	1	1	1	1	1
10	1°	2	2	2	2	2	1	1
	2°	1	1	1	1	1	1	1
11	1°	2	2	2	2	2	2	2
	2°	2	2	2	2	2	2	2
12	1°	2	2	2	2	2	2	2
	2°	2	2	2	2	2	2	2
13	1°	2	2	2	2	2	2	2
	2°	2	2	2	2	2	2	2
14	1°	2	2	2	2	2	2	2
	2°	2	2	2	2	2	2	2
15	1°	2	2	2	2	2	2	2
	2°	2	2	2	2	2	2	2
16	1°	5	5	5	5	5	0	0
	2°	0	0	0	0	0	0	0
17	1°	2	2	2	2	2	2	2
	2°	2	2	1	2	2	2	2
18	1°	0	1	0	1	0	0	0
	2°	0	0	0	0	0	0	0
19	1°	1	1	1	1	1	0	0
	2°	0	0	0	0	0	0	0
20	1°	2	2	2	2	2	2	1
	2°	1	1	1	1	1	1	1
	Total	80	82	82	82	80	63	62

For the case-study, the heuristics and association of heuristics that has given the best results with multiple and multiple and unary constraints were employed. The association of the MRV + Grade + DV heuristics can be interpreted in the following way: the association of MRV + Grade heuristics selects the best unit to be allocated and the DV heuristics selects the best physician to allocate. The results obtained in the case-study are satisfactory, mainly when it is considered that the same schedule took about 50 h to be made manually. This shows the effectiveness of the heuristics set chosen for the case-study.

5. Conclusion

A class of human allocation problem in health area, cooperative services, was modeled through the CSP approach. The solution employing the backtracking algorithm was improved by the proposal of a new heuristics (the *DV* heuristics). While the other heuristics already published in literature work are based on unit selection, this new heuristics is based on selection of the professional to be allocated. The isolated performance of the proposed heuristics was better than the isolated one of the other ones when complex constraints were present. The association of MRV + Grade heuristics to select the allocation unit and the *DV* heuristics to select the professional to be allocated seems to be a good choice to solve a real problem. The developed tool enables that new constraints be inserted or removed every time it is needed, making it possible the employment of this tool to solve different problems with no need of a new written code.

References

Dowsland, K. (1998). Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research, 106*, 393–407.
Kumar, V. (1992). Algorithms for constraint – satisfaction problems: A survey. *AI Magazine, 13*, 32–44.
Oddi, A., & Cesta, A. (2000). Toward interactive scheduling systems for managing medical resources. *Artificial Intelligence in Medicine, 20*, 113–138.
Oughalime, A., Ismail, W., & Yeun, L. (2008). A tabu search approach to the nurse scheduling problem. *International Symposium on Information Technology, 1*, 1–7.
Russel, S., & Norvig, P. (2004). *Artificial intelligence*. New York: Pearson Education.
Spyropoulos, C. D. (2000). AI Planning and scheduling in the medical hospital environment. *Artificial Intelligence in Medicine, 20*, 101–111.
Tsai, C. C., & Li, S. (2009). A two-stage modeling with genetic algorithms for the nurse scheduling problem. *Expert Systems with Applications, 36*, 9506–9512.
Tsang, E. (1996). *Foundations of constraint satisfaction*. London: Academic Press.
Valouxis, C., & Housos, E. (2003). Hybrid optimization techniques for the work shift and rest assignment of nursing personnel. *Artificial Intelligence in Medicine, 20*, 155–175.
Weil, G., Heus, K., François, P., & Poujane, M. (1995). Constraint programming for nurse scheduling. *IEEE Engineering in Medicine and Biology, 14*, 417–422.