FACULTEIT ECONOMIE
EN BEDRIJFSKUNDE

TWEEKERKENSTRAAT 2
B-9000 GENT
Tel. : 32 - (0)9 – 264.34.61
Fax. : 32 - (0)9 – 264.35.92

WORKING PAPER

Branching Strategies in a Branch-and-Price Approach for a Multiple Objective Nurse Scheduling Problem

Broos Maenhout [1]

Mario Vanhoucke [2]

January 2008

2008/495

[1] Faculty of Economics and Business Administration, Ghent University, Gent, Belgium
[2] Faculty of Economics and Business Administration, Ghent University, Gent, Belgium
Operations & Technology Management Centre, Vlerick Leuven Gent Management School, Gent, Belgium
broos.maenhout@ugent.be • mario.vanhoucke@ugent.be

D/2008/7012/04

ABSTRACT

The efficient management of nursing personnel is of critical importance in a hospital's environment comprising a vast share of the hospital's operational costs. The nurse scheduling process affects highly the nurses' working conditions, which are strongly related to the provided quality of care. In this paper, we consider the rostering over a mid-term period that involves the construction of duty timetables for a set of heterogeneous nurses. In scheduling nursing personnel, the head nurse is typically confronted with various (conflicting) goals complying with different priority levels, which represent the hospital's policies and the nurses' preferences. In constructing a nurse roster, nurses need to be assigned to shifts in order to maximize the quality of the constructed timetable satisfying the case-specific time related constraints imposed on the individual nurses' schedules. Personnel rostering in healthcare institutions is a highly constrained and difficult problem to solve and is known to be NP-hard. In this paper, we present an exact branch-and-price algorithm for solving the nurse scheduling problem incorporating multiple objectives and discuss different branching and pruning strategies. Detailed computational results are presented comparing the proposed branching strategies and indicating the beneficial effect of various principles encouraging computational efficiency.

Keywords: *Nurse Scheduling, Branch-and-Price, Branching Strategies*

1 Introduction

Recruiting and keeping the right staff are key challenges for the healthcare industry. The performance and quality of a health system ultimately depend on the quality and motivation of health human resources (Zurn, Dolea, and Stilwell, 2005). Therefore, recruitment and retention problems should be appropriately addressed since staff shortages or unmotivated health workforce are likely to have adverse effects on the delivery of health services and outcome of care. A key factor is the organizational support to employees, which is especially revealed in the policies and practices conducted by the health organizations. In this respect, personnel scheduling is a central component and is essential for the delivery of care to patients. On the one hand, it is of critical importance to have suitably qualified staff on duty at the right time since this is a large determinant of service organization efficiency and customers' requirements satisfaction in providing the continuity of care. On the other hand, Cline, Reilly, and Moore (2003) identify unattractive schedules and high workloads as two important factors leading to discontentment and a high nursing turnover. More and more, hospitals adopt scheduling policies that increasingly accommodate preferences and requests of their nursing staff and abandon the more traditional cyclic scheduling. In spite of recent technological advances, it is obvious that health care managers face significant challenges as all these issues congregate at a time when managers are under increasing pressure to control costs while simultaneously ensuring the delivery of high-quality care. One potential way of easing this pressure is to develop better nurse scheduling decision support systems that can help to produce rosters that employ resources more efficiently. The efficient management of nurses is vital to any organization's overall success because nursing labor costs typically represent over 40% of a hospital's total budget (Kazahaya, 2005; Welton, 2006). Hence, the management has interests such as minimizing overtime, maintaining continuity of care, maximizing morale, and minimizing turnover and absenteeism. These are all factored into the scheduling process, at least implicitly. Solving the nurse scheduling problem properly has a positive impact on the nurses' working conditions, which are strongly related to the quality level of health care and the recruitment of qualified personnel (Burke et al., 2004).

This trade-off among roster quality, cost, morale, and performance and its difficulty to model, quantify and parameterize adequately all compromising factors partly explains the wide interest in the literature for the nurse workforce management process. The nurse workforce management is often seen as a three-phase sequential process that basically consists of a staffing, a scheduling, and an allocation phase (Abernathy, Baloff, and Hershey, 1971). In this paper, we present an exact procedure to solve the nurse scheduling problem (NSP), which consists of generating a configuration of individual schedules for all nurses. We concentrate on the scheduling phase rostering the nurses over the mid-term period that involves the construction of duty timetables for a set of heterogeneous nurses over a pre-defined period of, for example, one month. A part of the problem data, such as the number of personnel in a ward, the required qualifications, and the definition of shift types is determined at the strategic level (i.e., the staffing phase). Constructing timetables of work for personnel in healthcare institutions is a highly constrained and difficult problem to solve and is known to be NP-hard (Osogami and Imai, 2000). Each employee of the available nursing staff should be assigned to an individual schedule, which needs to be conform to the applicable nurse scheduling policies. These individual schedules can be viewed as a sequence of days on and days off, where the nurse can be assigned to a particular shift (e.g., early, day, evening, night shift). For the remainder of the paper, we denote such a line-of-work for a nurse as the individual nurse's *schedule* whereas the timetable for all nurses is depicted as the nurse *roster*.

In constructing a nurse roster, nurses need to be assigned to shifts such that the quality of the constructed timetable is maximized while satisfying the case-specific time related constraints imposed on the individual nurses' schedules (Burke et al., 2004). In scheduling nursing personnel, the head nurse is typically confronted with various (conflicting) goals complying with different priority levels, which represent the hospital's policies and the nurses' preferences. In the literature a huge variety of objective function possibilities is considered whereas the imposed constraints can typically be processed as hard or soft constraints with varying constraint parameters and penalty costs (Burke et al., 2004). In the following, the major components of the nurse scheduling problem under study are discussed, i.e.,

- The *hospital objectives* typically consists of ensuring a continuous service with appropriate nursing skills and staffing size, while avoiding additional costs for unnecessary overtime (Azaiez and Al Sharif, 2005; Bard and Purnomo, 2005b). Hence, the hospital must provide some minimum level of care in terms of the number of nurses per skill category for each shift. These minimal coverage requirements express the required number of nurses per shift and per day for all skill categories (e.g., registered nurses, licensed practical nurses, nurse aides, technicians, senior and junior personnel), and are inherent to any shift scheduling problem. In the personnel scheduling literature, various example cases are described with strictly separated skill categories, hierarchically substitutable qualifications, or user definable substitution, which is particularly well suited to real-world practice (Burke et al., 2004; Bard and Purnomo, 2005b). In case of shortages in any ward, the corresponding head nurse can call upon ward nurses doing overtime or may borrow nurses from other wards having a similar specialty. Setting the assignment priorities and the corresponding penalty costs properly, understaffing is compromised first by overtime for part-time nurses and then by floating nurses. Nonetheless, our model (see section 3) is primarily designed for scheduling permanent staff. Insufficient coverage is avoided by adding slack variables that provide an estimation of the required overtime hours or the floating staff size since the slack variables refer to the extra number of nurses per skill level and shift types for each period and each day. Hence, the coverage constraints are handled as soft constraints that can be violated at a certain penalty cost expressed in the objective function. These penalty costs are conceived in a way the personnel shortages are equally distributed over the time horizon. Moreover, the redundancy of nursing personnel is also avoided by penalizing overstaffing such that redundant personnel is equally distributed over the scheduling period.
- The *quality* of a personnel roster in the modern work environment is more and more measured in terms of satisfying the individual nurses' preferences (Ernst et al., 2004). The way these preferences are quantified in the objective function determines the perceived quality of the nurse roster over the scheduling horizon and the perception of fairness between the nurses (Warner, 1976). Azaiez and Al Sharif (2005) conducted an extensive survey to gain understanding on real-world nurses' preferences. All of the identified kind of preferences can be modelled by defining each nurse's preference of working a particular shift on a particular day and sequence dependent preferences (e.g., continuity problems, number of consecutive working days). The latter can be incorporated by defining a (soft) case-specific time related constraint and a corresponding penalty cost that is incurred whenever the nurse's individual schedule violates the case-specific constraint and, hence, does not comply with the nurse's preferred sequence of shifts. According to Zurn, Dolea, and Stilwell (2005), accommodating individual preferences and personal requests will foster a higher morale, a more attractive work environment, increased flexibility to deal with personal matters and higher retention rates. Not considering nurses' preferences causes nurses'

frustration leading to either working under high stresses or quitting their jobs (Azaiez and Al Sharif, 2005).

- The *case-specific time related constraints* imposed on individual nurses' schedules are determined by personal time requirements, specific nurse contract stipulations and regulations, specific workplace scheduling policies and practices, collective union agreement requirements, national legislation, etc. These rules define acceptable schedules for the individual nurses and the head nurse and reduce the set of feasible individual roster lines. In practice, hospitals must provide flexibility to define personal work agreements relaxing the hospital- and ward-specific time related constraints and, hence, are confronted with a wide variety of contracts (both full- and part-time) (Burke et al., 2004).

The remainder of the paper is organized as follows. In section 2, we give an overview of the relevant literature applying mathematical programming approaches for solving personnel scheduling problems. In section 3, we discuss the various features of the implemented branch-and-price approach. We describe a method to obtain near-optimal solutions based on column generation, which serves as an upper bound for the branch-and-bound-tree in search for the optimal solution. Furthermore, we suggest several branching strategies and a node pruning strategy to search the branch-and-bound-tree efficiently. In section 4, we present the results of our computational experiments giving insights in the performance of the proposed branching schemes, node pruning strategies, and speed-up techniques. In section 6, conclusions are drawn and directions for future research are given.

2 Literature Overview

Problem descriptions and models vary drastically and depend on the characteristics and policies of the particular business environment. Hence, in the literature many objective function possibilities subject to a huge variety of constraint combinations are explored. Since personnel scheduling problems have this multitude of formulations, many procedures have been proposed to solve personnel scheduling problems in general and the nurse scheduling problem in particular (Burke et al., 2004; Ernst et al., 2004). Exact procedures and in particular mathematical programming techniques have been frequently proposed in nurse scheduling literature for both the cyclical and non-cyclical scheduling of nursing personnel, i.e., assignment programming (Caron, Hansen, and Jaumard, 1999), linear programming (e.g., Morris and Showalter, 1983), integer programming (e.g., Billionnet, 1999), mixed-integer programming (e.g., Beaumont, 1997), network programming (e.g., Balakrishnan and Wong, 1990), non-linear programming (e.g., Warner, 1976), goal programming (e.g., Arthur and Ravindran, 1981; Azaiez and Al Sharif, 2005; Berrada, Ferland, and Michelon, 1996; Brusco and Johns, 1995; Musa and Saxena, 1984), and branch-and-bound approaches (Trivedi and Warner, 1976). In these papers, multiple goal objective models have received considerable attention as they attempt to optimize simultaneously a number of often conflicting objectives due to the different parties in hospital settings. These objectives often involve maximizing utilization of full-time staff, minimizing the number of employees, minimizing understaffing and overstaffing costs, minimizing labour costs as well as minimizing deviations from desired staffing requirements, nurse preferences, nurse special requests, and fair assignment of employees to schedules. The goals with different priority levels and corresponding weights represent the hospital policies and the nurses' preferences.

However, these models are often rather simplified approaches, rely eventually on heuristic principles to achieve near-optimal solutions or consider far more variables that can be dealt with a reasonable computing effort. In order to overcome these difficulties, several authors have solved personnel scheduling problems using column generation and closely connected resulting branch-and-price approaches (e.g., Bard and Purnomo, 2005a, 2005b; Beliën and Demeulemeester, 2005, 2006; Caprara, Monaci, and Toth, 2003; Jaumard, Semet, and Vovor, 1998; Mehrotra, Murphy, and Trick, 2000; Gamache et al., 1999). Jaumard, Semet, and Vovor (1998) were the first to present a basic 0-1 column generation model with a dedicated resource constrained shortest path auxiliary problem for the scheduling of nursing personnel. Motivated by the changing work environment accommodating personnel preferences and requests, Bard and Purnomo (2005a) proposed a multi-objective model formulation for the nurse scheduling problem that is solved with a heuristic column generation approach that combines integer programming and heuristics to generate new columns. Moreover, Bard and Purnomo (2005b) present a column generation approach additionally taking the nurses' competencies into account. They give insight in the beneficial effect of substituting nurses by nurses with lower competencies when there is a critical staff shortage and few alternatives leading to considerable reductions in the need for expensive outside nurses and much better schedules for the regular staff as measured by preference satisfaction. Beliën and Demeulemeester (2005) do not consider the nurse scheduling problem as a separate problem but describe an approach where the minimal coverage constraints are dependent on the master operation room schedule. The goal is to determine the master surgery schedule in the operation room needing the least number of homogeneous nurses. It is outside the scope of their paper to find efficient branching schemes for the separate nurse scheduling problem. Caprara, Monaci, and Toth (2003) present some mathematical models and solution algorithms for a family of staff scheduling problems. The main objective is the minimization of the number of employees needed to perform all daily assignments in the horizon and is solved using a column generation approach. Concerning related personnel scheduling problems, Mehrotra, Murphy, and Trick (2000) present a branch-and-price technique for the tour staff scheduling problem. They devise and implement specialized branching rules suitable for solving the set covering type formulation implicitly using column generation. Gamache et al. (1999) discuss several interesting techniques in order to improve the performance of their branch-and-price algorithm for solving the related airline crew scheduling problem. Furthermore, Beliën and Demeulemeester (2006) develop a branch-and-price procedure for scheduling medical trainees to tasks. In Beliën and Demeulemeester

(2006) the authors compare different branching strategies, i.e., branching on the column variables, branching on the original variables and a problem-specific branching on precedence relations, and conclude the branching scheme on the timetable cells provides consistently the best results.

The contribution of this paper to the nurse scheduling literature is twofold. Firstly, we propose a branch-and-price procedure for the nurse scheduling problem that is able to solve real-world problem instances exactly. The described nurse scheduling problem is very flexible in terms of objective function possibilities, nurses' substitutability, case-specific time related constraints, nurse-specific characteristics and comprehends most characteristics encountered in practice. Secondly, different new and existing branching strategies, node reduction mechanisms, and speed-up techniques have been proposed and/or fine-tuned in order to give insights and improve the computational performance solving the nurse scheduling problem to optimality. We have tested the proposed procedure in a real-world problem environment and investigated the sensitivity of the proposed optimization principles by varying systematically the problem characteristics.

3 Solution procedure

In order to solve the nurse scheduling problem properly, we decompose and reformulate the traditional nurse scheduling problem formulation using the assignment variables (Musa and Saxena, 1984) based on the Dantzig-Wolfe decomposition in order to generate tighter bounds. This reformulation gives rise to an integer master program with a large number of variables, i.e., columns, and a subproblem, which formulation defines the structure of a (feasible) column (Barnhart et al., 1998; Vanderbeck, 2000). The integer master program is dealt using an integer programming column generation procedure, i.e., branch-and-price, which solves the identified pricing problem to check the optimality of an LP solution and branches when the optimal LP solution does not satisfy the integrality conditions. In this section, we discuss the algorithmic details of the branch-and-price approach to solve the nurse scheduling problem. In the following, the pseudo-code for the branch-and-price procedure to solve the NSP is described.

```
Algorithm B&P NSP

    Initialize Restricted Master Problem
    Construct Initial Heuristic Solution

    While level >= 0
        While LP optimal solution is not met (i.e. reduced cost < 0)
            Column Generation Procedure
```

```
        [Solve Restricted Master Problem
         Get Dual Prices
         Solve Pricing Problem
         Apply Pruning Rule ]
     End While

     If LP optimal solution < Best found solution
        If fractional
         Apply Pruning Rule if possible
         Apply Branching Strategy
        Else
         Save New Best found solution
         Backtrack
     Else Backtrack
  End While
```

The branch-and-price algorithm starts with a simple but efficient heuristic method producing good columns to initialize the restricted master problem, which considers only a subset of the schedules nurses can be assigned to (see section 3.1). Next, the algorithm aims to tighten this initial upper bound using a heuristic column generation procedure based on the linear programming relaxation of the master problem (see section 3.2). In order to obtain this LP relaxation of the master problem relaxing the integrality constraints, the search procedure will loop through the column generation procedure. The column generation procedure takes only a feasible subset of the schedules into consideration at each iteration and solves the restricted master problem (z_{RMP}). This restricted master problem can be formulated as follows

Notation

Problem size parameters

N	set of regular nurses to be scheduled (index *i*)
D	set of days in the planning horizon (index *j*)
S	set of shifts adequately covering the demand periods (includes the free shift) (index *k*)

Nurse parameters

G	set of skill categories (index *m*)
F_i	set of feasible schedules for nurse *i* with respect to all hard case-specific time related constraints (index l)

a_{ilmjk}	1 if schedule *l* for nurse *i* covers the required skill competency *m* of shift *k* on day *j*, 0 otherwise
p^1_{ijk}	penalty cost of assigning nurse *i* to shift *k* on day *j*, i.e., inverse to the nurse's preference being scheduled to shift *k* on day *j*
p^2_{il}	total penalty cost of schedule *l* violating the case-specific time related constraints expressing the preferences and contract stipulations of nurse *i*
p^3_l	total penalty cost of schedule *l* violating (soft) time related constraints as a result of the specific workplace conditions (i.e., the practices of the head nurse)
c_{il}	total penalty cost of assigning nurse *i* to schedule *l* (i.e., $c_{il} = \sum_{m\in G}\sum_{j\in D}\sum_{k\in S} a_{ilmjk}\, p^1_{ijk} + p^2_{il} + p^3_l$)

Demand coverage parameters

R_{mjk}	required number of nurses of skill category *m* for shift *k* on day *j*
c^u_{rmjk}	penalty cost of understaffing skill competency *m* on shift *k* on day *j* with *r* extra nurses ($r \in \{1,\dots, R_{mjk}\}$)
c^o_{qmjk}	penalty cost of overstaffing skill competency *m* on shift *k* on day *j* with *q* nurses ($q \in \{1,\dots,\lvert N\rvert - R_{mjk}\}$)
n^u_{rmjk}	1 if r extra nurses are needed to satisfy the coverage requirements of skill competency *m* for shift *k* on day *j*
n^o_{qmjk}	1 if q nurses are scheduled in surplus to perform skill category *m* for shift *k* on day *j*

Decision variables

y_{il}	1 if nurse *i* is assigned to schedule *l*, 0 otherwise
x_{imjk}	1 if nurse *i* is assigned to shift *k* on day *j* requiring skill competency *m*, 0 otherwise

Master Problem Formulation

$$\text{Min} \quad z_{RMP} = \sum_{i \in N} \sum_{l \in F_i} c_{il} y_{il} + \sum_{m \in G} \sum_{j \in D} \sum_{k \in S} \sum_{r=1}^{R_{mjk}} c^u_{rmjk} n^u_{rmjk} + \sum_{m \in G} \sum_{j \in D} \sum_{k \in S} \sum_{q=1}^{n-R_{mjk}} c_{o,qmjk} n^o_{qmjk} \qquad [1]$$

$$\text{s.t.} \quad \sum_{i \in N} \sum_{l \in F_i} a_{ilmjk} y_{il} + \sum_{r=1}^{R_{mjk}} r n^u_{rmjk} - \sum_{q=1}^{n-R_{mjk}} q n^o_{qmjk} = R_{mjk} \qquad \forall m \in G, j \in D, k \in S \qquad [2]$$

$$\sum_{l \in F_i} y_{il} = 1 \qquad \forall i \in N \qquad [3]$$

$$\sum_{r=1}^{R_{mjk}} n^u_{rmjk} + \sum_{q=1}^{n-R_{mjk}} n^o_{qmjk} \leq 1 \qquad \forall m \in G, j \in D, k \in S \qquad [4]$$

The objective [1] of this master problem aims to minimize the weighted sum of penalty costs associated with the individual schedules the nurses are assigned to, the number of times the ward is confronted with a shortage of personnel, and the number of times too many nursing staff is assigned. The penalty costs of the latter two are postulated in a way under- and overstaffing is levelled over the planning horizon, i.e., penalty costs c^u_{rmjk} and c^o_{qmjk} increase exponentially with the deficient or excessive number of nurses. Constraint [2] indicates the required number of nurses for each skill category for each shift on each day. Moreover, formulating the coverage constraints as such allows the very flexible definition of substitutability between the nurses. To ensure mathematical feasibility, two slack variables n^u_{rmjk} and n^o_{qmjk} are associated with constraint [2] representing department under- or overstaffing respectively. Constraint [3] assigns each nurse to a schedule that is feasible to his or her specific time related constraints. Constraint [4] is a supporting constraint linearizing the master problem formulation. More specifically, this constraint determines that maximum one slack or surplus variable can be selected to deal with over- or understaffing. Given the problem structure, the slack variables are bounded, i.e., the slack variable modelling overstaffing can be at most the number of nurses minus the corresponding coverage requirements and the slack variable modelling understaffing can be at most the corresponding coverage requirements.

Next, after solving the restricted master problem, the column generation procedure tries to identify new schedules to enter the basis and adds new schedules if necessary. Since adding a schedule (i.e., a column) can decrease the value of the linear programming solution only if the inserted column has negative reduced cost, we solve the pricing problem that aims to find a schedule with minimum reduced cost. The reduced cost of a new column *l* for nurse *i* is given by

$$\mu_{il} = c_{il} - \gamma_i - \sum_{m \in G} \sum_{j \in D} \sum_{k \in S} \pi_{mjk} a_{ilmjk} = -\gamma_i - \sum_{m \in G} \sum_{j \in D} \sum_{k \in S} (\pi_{mjk} - p_{ijk}^1) a_{ilmjk} + p_{il}^2 + p_l^3 \qquad [7]$$

with μ_{il} reduced cost of column l for nurse i

π_{mjk} dual price of constraint [2] for skill competency m for shift k on day j

γ_i dual price of constraint [3] for nurse i

If this minimum value is nonnegative then the value of the linear programming solution will not decrease by considering other schedules that are not incorporated in the restricted master LP model. This implies that we have found the optimal solution of the linear programming relaxation (z_{LP}). Some specific implementation issues of the column generation procedure are discussed in section 3.3. If the obtained upper bound is not the optimal solution or the LP relaxation of the master problem solved by column generation may does not have an integral optimal solution, we will apply a branch-and-bound depth-first search in order to drive the search process to the optimal integer solution. Branching involves the partitioning of the solution space into disjoint subsets using a particular branching strategy such that the current fractional solution is excluded (see section 3.4). Applying a standard-branch-and-bound procedure to the master problem over the existing columns is unlikely to find an optimal, or even good, or even feasible solution to the original problem. Therefore, it is necessary to apply the column generation procedure and possibly generate additional columns at non-root nodes of the branch and bound tree. At each node, the solution value of the master LP is a lower bound for the problem subject to the active branching constraints. If this lower bound exceeds an already found upper bound, the algorithm back-tracks. If the lower bound is lower than the current best found solution, the algorithm checks whether or not the solution contains fractional columns. If the LP solution is integer, the solution is feasible and thus can be saved as the current best found solution (*UB*). If the obtained LP solution is fractional, further branching is needed. If the node is not fathomed by a pruning rule (see section 3.5), the algorithm will re-enter the LP optimization loop in order to determine a new lower bound given the extra constraint(s). This process continues until backtracking leads back to the root node of the branch-and-bound tree (Barnhart et al., 1998; Vanderbeck, 2000).

3.1 Initialization of the Restricted Master Problem

In order to speed up the convergence to the LP optimal solution of the initial column generation procedure, the algorithm generates high-quality individual nurse schedules for all nurses based on a well performing simple constructive heuristic instead of generating random columns. The constructive heuristic successively generates individual nurse schedules by scheduling the nurses in a random sequence. The identified subproblem is solved for each nurse taking both the nurse's preferences and the penalty costs associated with the coverage constraints and soft regulations into account. The individual nurse's schedule is obtained by calculating the nurse's shortest feasible path using a resource constrained shortest path algorithm (see section 3.2.1).

3.2 Initial heuristic solution

It is well-known that the availability of a good feasible start solution may reduce the size of the branch-and-bound tree considerably. According to Barnhart et al. (1998) and Gamache et al. (1999), a branch-and-price algorithm can be easily turned into an effective approximation algorithm. This is accomplished by branching and searching the tree in a greedy fashion using a depth first strategy. Hence, we aim to improve the initial best solution using a more knowledge-based heuristic based on the column generation procedure and the appropriate branching strategy on the original variables, which is described in section 3.4. The pseudo-code for obtaining this initial heuristic solution is described below.

```
Construct Initial Heuristic Solution
   While Heuristic Solution Not Found
      Column Generation Procedure
      Fix Positive Integer Assignment Variables
      Fix Fractional Assignment Variable
   End While
```

Starting from the initial LP relaxation, we fix all original variables ($x_{imjk} = 1$) for which the sum of the columns that cover the original assignment variable is one, i.e., if $\sum_{l \in F_i} a_{ilmjk} y_{il} = 1$. Moreover, we fix an additional original fractional variable (i.e., $0 < \sum_{l \in F_i} a_{ilmjk} y_{il} < 1$) to one based on the appropriate variable selection strategy (see section 3.4.2). Subsequently, the column generation procedure is invoked again to obtain the LP relaxation under the current shift assignment constraints. All additional integer positive variables are fixed and, again, a fractional variable is set to one. This process of progressively fixing the assignments in the individual schedules continues until a heuristic integer solution is obtained, i.e., $\sum_{l \in F_i} a_{ilmjk} y_{il} = 1$ ($\forall i \in N, m \in G, j \in D, k \in S$).

3.3 Column generation procedure

3.3.1 Subproblem and pricing procedure

The restricted master problem considers a subset of the columns and is solved using the simplex method. In order to obtain the LP relaxation, additional columns are generated as needed by solving the pricing problem, i.e., the subproblem with appropriate dual information (Barnhart et al., 1998; Vanderbeck and Wolsey, 1996). The way of decomposing the original problem formulation leads to a subproblem that consists of generating a feasible individual nurse schedule. The scheduling of a single nurse over the complete scheduling horizon can typically be considered as a minimum cost flow problem on a suitably defined graph (Bard and Purnomo, 2005a). This problem assigns each day j ($j \in D$) a single shift k ($k \in S$) to the nurse, which is denoted by the variables used in the traditional problem formulation of the nurse scheduling problem, i.e., x_{imjk} . These variables are further referred to as the original variables. Moreover, the shift assignment needs to be conform to all (hard) case-specific time related constraints satisfying the nurse's preferences and contract stipulations (p^1_{ijk} and p^2_{il}) and the (soft) regulations of the specific head nurse (p^3_l) as much as possible. Each unit in a hospital may have a different set of hard and soft constraints leading to a different subproblem. Moreover, this subproblem will even differ from employee to employee since the specific nurse preferences, contract stipulations, work regulations, skill competencies, etc vary largely among the nursing staff. Hence, a singular subproblem and associated network is defined for each nurse. All nurses' networks are acyclic having a source and a sink node representing the beginning and end of the scheduling period. In constructing nurses' networks with multiple skill competencies, we duplicate the assignable working shifts by the number of skill competencies such that each shift node corresponds to one specific skill competency. This implies that the shift assignment assigns a nurse to a shift requiring a single specific skill competency and prevents a nurse can provide for multiple skills during a particular shift. The graph used for our algorithm consists of ($|D| * (|S| – 1) * |G| + |D|$) nodes representing the daily shift assignments for the nurse under study.

This pricing problem, i.e., the identified subproblem, is a resource constrained shortest path problem (Jaumard, Semet, and Vovor, 1998), which defines the generated paths to be feasible with respect to different resource constraints and the path structural constraints. Resource constraints can be formulated by means of (minimal) resource consumptions and resource intervals (or resource windows) (e.g., minimal number of working hours). Path structural constraints can model further requirements concerning the feasibility of paths, which are not covered by resources. Such additional requirements might either be an integral part of a feasible path's definition (e.g., complete weekends, 11hrs rest between two working shifts) or be implied by branching rules which come up in the context of branch-and-price and require modifications of the pricing problem. On acyclic graphs, the resource constrained shortest path has been identified to be NP-hard. In nurse scheduling literature, different (pseudo-polynomial) methodologies for solving the resource constrained shortest path problem have been applied. Bard and Purnomo (2005a) use a swapping heuristic to generate new individual nurse schedules. Beliën and Demeulemeester (2005) rely on a dynamic programming approach whereas

Jaumard, Semet, and Vovor (1998) propose a dedicated pseudo-polynomial two-phase algorithm to solve the resource constrained shortest path problem.

The pricing problem defined above finds the column with the lowest reduced cost. Therefore, if a column with negative reduced cost exists, the column generation will always identify a candidate column. This guarantees that the optimal solution to the linear program will be found. However, this may be computationally prohibitive. Fortunately for the column generation scheme to work, it is not necessary to always select the column with the lowest reduced cost, any column with a negative reduced cost will do. In our procedure a two phase approach is implemented. In the first phase, the procedure tries to generate heuristically columns with negative reduced cost. As in Sol and Savelsbergh (1994) several existing columns with a reduced cost close to zero are selected and employ fast local improvement algorithms to construct columns with a negative reduced cost. More precisely, the heuristic pricing phase exploits the single-shift day neighbourhood and the greedy shuffling neighbourhood described in Burke et al. (2003) to alter the selected columns to columns with negative reduced cost as best as possible. In case of failure, the procedure switchs to the second phase where the column with the lowest reduced cost is obtained by a recursive dynamic programming approach (Caprara, Monaci, and Toth, 2003). The optimization algorithm proves optimality or generates an additional column with negative reduced cost. If this pricing procedure finds a negative reduced cost column during the heuristic pricing phase and hence does not enter the optimal pricing phase, the Lagrangean lower bound (see section 3.3.2) cannot be used because the pricing problem has not been solved to optimality.

3.3.2 Lagrangean dual pruning

Lagrangean relaxation can typically be exploited within the framework of a column generation procedure, not only to alleviate the “tailing-off effect” to terminate the column generation method sooner but also to speed up the pricing algorithm. When searching for an integer optimal solution (and, hence, an integer objective function value), Vanderbeck and Wolsey (1996) describe a bound on the final LP value based on the LP value of the restricted master problem and the current reduced costs. Lagrangean relaxation gives a lower bound in each iteration of the column generation process, which needs to be compared to an upper bound on the optimal solution value to check if a node in the branch-and-price tree can be fathomed without any risk of missing the optimum. Hence, the column generation algorithm can be stopped earlier by proving the optimality of the current best solution if the Lagrangean relaxation is higher than the current best solution (UB) minus one (Barnhart et al., 1998; Van den Akker, Hoogeveen and Van de Velde, 2002), i.e.,

$$z_{RMP} - \sum_{i \in N} \mu_i > UB - 1 \qquad [8]$$

Moreover, the Lagrangean lower bound can also be applied to determine if the column generation algorithm has converged to the optimal solution value, i.e.,

$$z_{RMP} - \sum_{i \in N} \mu_i > \lfloor z_{RMP} \rfloor \qquad [9]$$

3.3.3 Upper bound pruning for the pricing problem

This bound prevents the algorithm from generating additional columns for a particular nurse by exploring the possibility of finding a column with negative reduced cost (Beliën and Demeulemeester, 2005). For calculating a lower bound for the pricing problem, we relax the problem formulation of the subproblem by removing all constraints but the path structural constraints and calculate the shortest path over the defined network. No columns need to be generated for the nurse solving the integral subproblem whenever this lower bound exceeds the upper bound for the pricing problem. A column with a reduced cost of zero can be seen as an upper bound for the pricing problem. Moreover, out of [7] the upper bound for this relaxed subproblem can be set equal to the dual price of the constraint associated with the nurse under study (i.e., γ_i).

3.3.4 Partial Column Generation

The pricing step in personnel scheduling branch-and-price procedures is traditionally computationally intensive. Since the subproblems for different nurses use the same dual information the procedure tends to produce schedules covering the same shifts. In order to accelerate the procedure, only a small group of subproblems is solved at each iteration rather than all subproblems for all nurses (Gamache et al., 1999). As the iterations progress, it becomes increasingly difficult to find a column with negative reduced cost. Therefore, more and more subproblems are solved until, ultimately, all subproblems need to be solved.

3.4 Branching strategy

One difficulty in using column generation to solve integer programs is the development of efficient branching rules to ensure integrality. A common branching rule that is appropriate for the master problem, i.e., 0/1 branching on the column variables, is not applicable for the restricted master problem where the columns are generated by implicit techniques. Hence, fixing or bounding a variable of the master is not effective since fixing column variables may destroy the structure of the pricing problem and leads to an unbalanced branch-and-bound tree (Barnhart et al., 1998). In general, this approach comes down to determining the k^{th} best solution to the column generation subproblem at the k^{th} level in the branch-and-bound tree, which is much harder than solving the most enterable column.

In the following, we indicate how the proposed branch-and-price procedure branches to integrality using two different branching methodologies, i.e., branching on the original variables and branching on the residual problem. The combination of these two branching methodologies results in effective branches and pseudo branches. If the algorithm fails to find the optimum because it focused on the wrong residual problem at a node and its offspring of the search tree, it can backtrack on the branch that defines this residual problem, thereby creating an effective branching on an original fractional variable. In figure 1 we illustrated the different branching methodologies on an example problem instance where 5 nurses need to be scheduled over a period of 3 days and 4 shifts (early shift (s_1), late shift (s_2), night shift (s_3), and free shift (s_4)) requiring all the same skill competency. The matrices display the values of the original variables (i.e., $\sum_{l \in F_i} a_{ilmjk} y_{il}$, $\forall i \in N, m \in G, j \in D, k \in S$).

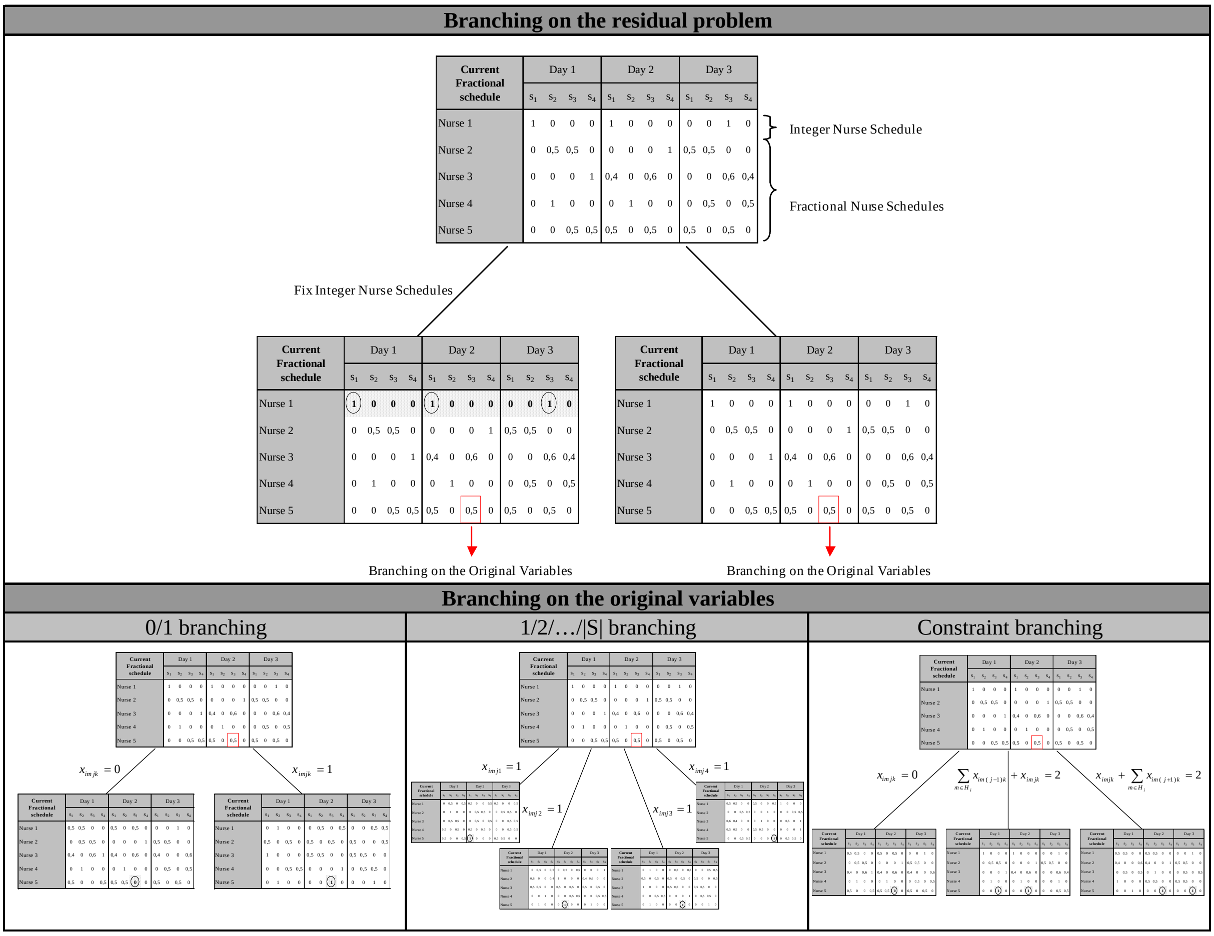

Figure 1. Different branching methodologies for the nurse scheduling problem

3.4.1 Branching on the residual problem

Before selecting an original fractional variable to branch on, we can first focus on the residual problem. The residual problem is the problem that remains after retaining the integer individual nurse schedules and is comprised of the nurses with fractional nurse schedules who are required to provide for the residual required coverage. In other words, two branches are created where the one branch fixes the integer individual nurse schedules and the second branch does not and applies one of the effective branching strategies proposed in section 3.4.2 on the created branches. This branching methodology strongly reduces the search space, accelerating the finding of integer nurse schedules.

3.4.2 Branching on the original variables

As the proposed optimization problem is inherently a set partitioning model, Ryan and Foster (1981) suggested a more suitable branching strategy intrinsically based on the original variables (Barnhart et al., 1998). Moreover, they pointed out whenever the LP solution of a 0-1 restricted master problem is fractional, it is always possible to find a pair of fractional original variables to initiate a new branch. Personnel scheduling literature has shown that such branching strategies fixing the original variables are compatible with the pricing problem. The optimal configuration of schedules will be constructed by progressively fixing the assignments in the individual schedules. Whenever a fractional variable is detected, the branching on the original variables can basically be performed using different strategies, i.e.,

- The *first* possible strategy (i.e., 0/1 branching) determines the obligate assignment of a nurse to a particular shift requiring a specific skill competency on a particular day ($x_{imjk} = 1$) or explicitly prohibits the specific assignment ($x_{imjk} = 0$). This branching strategy is often applied in personnel scheduling literature (e.g., Jaumard, Semet, and Vovor, 1998; Mehrotra, Murphy, and Trick, 2000; Beliën and Demeulemeester, 2005, 2006).

- In a *second* branching strategy (i.e., 1/2/…/|S| branching) we construct as many branches in each node as the number of working shifts multiplied by the number of skill competencies compatible with the nurse's abilities plus one for the rest period. In each of these branches, we explicitly assign the nurse to a shift on a particular day performing a particular skill or to the rest period ($x_{imjk} = 1; \forall k \in S, m \in G$).

- As a *third* branching strategy (i.e., Constraint branching) we implement a constraint based branching rule. More precisely, the branching strategy is based on a very common constraint in practice advocating the quality of care, i.e., the constraint 'minimal consecutive working days of the same shift type'. This constraint typically hinders frequent schedule changes and advocates the continuity of care (Fitzpatrick, Farrell, and Richter-Zeunik, 1987; Haggerty et al., 2003). When a fractional variable is detected, the number of branches to be created is equal to the number of combinations the detected variable as part of the minimal consecutive shift pattern is assigned to the nurse and the branch prohibiting the assignment of the nurse to that shift (e.g., when the minimal consecutive working days is 2 for the detected fractional variable for shift type *k*, three branches can be created, i.e., $\sum_{m \in G} x_{im(j-1)k} + x_{imjk} = 2$, $x_{imjk} + \sum_{m \in G} x_{im(j+1)k} = 2$, and $x_{imjk} = 0$). When the constraint 'minimal consecutive working days of the same shift type' is 1 for the particular shift type, this branching scheme is identical to the 0/1 branching scheme on the original variables. Neighboring branching strategies, referred to as branch-on-follow-on strategies, are already explored by Falkner and Ryan

(1987) and Vance et al. (1997) for the bus driver scheduling problem and the airline crew pairing problem respectively.

Apart from these three branching strategies, the branching variable can be selected in several ways. In our first variable selection strategy (i.e., 'Most fractional variable'), the fractional variable x_{imjk} with its fractional part closest to 0.5 is chosen. In case of ties, the variable with the smallest nurse penalty cost p^1_{ijk} is selected. The second variable selection strategy (i.e., 'Worst assignment'), the algorithms selects the fractional variable that is worst in terms of objective function value relative to the nurse's best shift assignment that day. In case of ties the variable with its value closest to 0.5 is selected. The third variable selection strategy (i.e., 'Variable close to one') selects the fractional variable x_{imjk} closest to 1. In case of ties, the variable with the smallest nurse penalty cost p^1_{ijk} is selected.

3.5 Subset dominance rule

In order to further prune certain nodes of the branch-and-bound tree we have implemented the so-called subset dominance rule of De Reyck and Herroelen (1998). This dominance rule can be applied when the set of branching constraints of a previously examined node in the tree is a subset of the set of branching constraints in the current node. Obviously, this dominance rule can only occur when branching on the residual subproblem.

4 Computational Results

In order to test the performance of the branch-and-price procedure, we carried out a case study in a ward of a Belgian hospital and validated our algorithm on a set of artificially generated problem instances of the NSPLib dataset of Vanhoucke and Maenhout (2005a) integrating the characteristics of the real-life problem environment. In the ward, the head nurse constructs each month a nurse roster starting from an empty roster, which implies that the scheduling process is acyclically organized. In order to guarantee the quality of the constructed individual schedules, this process is influenced by the (real-life) characteristics of the available nurses (e.g., incidental scheduling preferences, nurse competencies) and guided by a set of (hard) case-specific time related ward rules displayed in table 1, which prevail in the computational experiments. In order to solve instances under varying assumptions and circumstances (i.e., roster construction for another month (e.g., other nurse preferences, other minimal coverage requirements)) and verify the robustness of the different proposed solution approaches, we downloaded a subset of nurse scheduling problem instances of the NSPLib dataset from www.projectmanagement.ugent.be/nsp.php, i.e, the N30 subset where 30 nurses need to be scheduled over a period of 28 days (one month) and three working shifts and a free shift. These files are artificially generated based on a set of complexity indicators for the nurse scheduling problem. More precisely, the nurse preferences working shift *k* on day *j* and coverage requirement information have been generated under a controlled design based on six indicators presented in Vanhoucke and Maenhout (2005b) resulting in 96 complexity classes containing each 10 instances. We have randomly selected 1 instance from each complexity class and extended them with the set of hard case-specific time related constraints displayed in table 1 imposed by the ward. Furthermore, individual sequence-dependent nurse preferences are defined for each nurse. We test, without loss of generality, our procedure with all nurses as full-time personnel having a single skill competency.

Description	**Parameters**	
	Minimum	**Maximum**
A rest of 11 hours between working shifts needs to be respected	-	-
Maximal one assignment per day	-	-
Number of Working hours	144	180
Number of Working days	20	20
Consecutive number of working days	2	5
Consecutive number of assigned early shifts	2	5
Consecutive number of assigned late shifts	2	5
Consecutive number of assigned night shifts	2	5
Consecutive number of rest periods	1	2
Number of free days after series of night shifts	2	2

(*) This data is obtained by interviewing the head nurse of the ward involved in the study.

Table 1. Work regulations of full-time nursing personnel

The coverage penalties have been defined as exponentially increasing penalties as follows, i.e., the penalty cost c^u_{rmjk} for encountering a staff shortage of one workforce is 100, of two workforces is 200, etc. which is multiplied by the number of nurses in shortage. Hence, for one workforce short a penalty of 100 is accounted in the objective function, for two nurses 400 is accounted, for three 900, etc. The penalty cost c^o_{qmjk} for encountering a staff surplus of one workforce is 5, of two workforces is 10, etc. multiplied with the excess number of nurses.

In this section, we give insight in the computational performance tested on the aforementioned 96 problem instances analyzing different aspects of the branch-and-price procedure. In section 4.1, we compare the performance of the branching schemes proposed in section 3.4 for the nurse scheduling problem. In section 4.2, we analyze the impact of the proposed strategies limiting the size of the branch-and-bound tree, i.e., the initial heuristic upper bound of section 3.2 and the pruning rule indicated in section 3.5. In section 4.3, we test the impact and relevance of the different speed-up techniques from sections 3.1, 3.3.1, 3.3.2, 3.3.3, and 3.3.4 for the nurse scheduling problem. All tests were carried out on a Dell computer with a Dual Core processor 2.8 Ghz and 2 Gb RAM. The procedure has been linked with the industrial LINDO optimization library version 5.3 (Schrage, 1995).

In the remainder, we report the results on all our tests using

- parameters describing the search for the LP relaxation of the master problem at the root node (index LP),
- parameters describing the search for the knowledge-based heuristic upper bound at the root node (index UB),
- parameters describing the search for the IP solution (index IP), and
- parameters providing global information (no index)

with the following symbols, i.e.,

# iter	number of column generation iterations
# columns	number of columns generated
CPU_{MP}	required CPU time to solve the master problem
CPU_{SP}	required CPU time to solve the subproblem
CPU	total required time to solve the linear relaxation
# $nodes_{effective}$	number of effective branching nodes
# $nodes_{total}$	total number of nodes
% optimal	percentage of solutions whose optimal solution is obtained within a time limit of 3600s

4.1 Performance analysis of different branching strategies for the NSP

In this section we compare the eighteen different branching strategies comprising the three variable selection strategies (‘Most fractional variable’, ‘Worst assignment’, and ‘Variable closest to one’), the possible pseudo-branching on the residual problem (‘Branching on the residual problem’ and ‘Without branching on the residual problem’), and the three effective branching strategies on the original variables (‘0/1 branching’, ‘1/2/…/$|S|$ branching’, and ‘Constraint branching’). Table 2 reports the best found results for our branch-and-price procedure for the different branching strategies incorporating all features exposed in section 3.

Effective branching strategy	0/1 branching					
Pseudo branching strategy	Branching on the residual problem			Without branching on the residual problem		
Branching selection strategy (*)	(1)	(2)	(3)	(1)	(2)	(3)
# $columns_{IP}$	895.96	490.01	387.12	857.67	553.04	437.69
% $optimal_{IP}$	96.88%	100.00%	92.31%	92.71%	100.00%	96.71%
CPU_{MP}	79.22	46.44	25.21	128.75	72.30	60.42
CPU_{SP}	469.69	347.98	442.63	433.12	353.14	403.84
CPU	548.91	394.42	467.84	561.87	425.44	464.27
Effective branching strategy	**1/2/…/|S| branching**					
Pseudo branching strategy	Branching on the residual problem			Without branching on the residual problem		
Branching selection strategy (*)	(1)	(2)	(3)	(1)	(2)	(3)
# $columns_{IP}$	1887.85	1626.08	2224.08	1747.98	1116.15	1012.35
% $optimal_{IP}$	88.54%	92.71%	53.85%	88.54%	92.71%	73.96%
CPU_{MP}	51.42	48.04	44.10	202.64	145.19	145.98
CPU_{SP}	643.23	599.12	1324.77	429.99	465.35	738.30
CPU	694.65	647.16	1368.86	632.63	610.54	884.28
Effective branching strategy	**Constraint branching**					
Pseudo branching strategy	Branching on the residual problem			Without branching on the residual problem		
Branching selection strategy (*)	(1)	(2)	(3)	(1)	(2)	(3)
# $columns_{IP}$	1584.46	1716.47	999.62	677.04	1224.69	320.19
% $optimal_{IP}$	92.71%	84.38%	80.77%	92.71%	88.54%	92.71%
CPU_{MP}	41.56	47.73	30.24	239.72	229.69	123.08
CPU_{SP}	555.42	636.50	755.31	384.10	555.10	284.90
CPU	596.98	684.24	785.55	623.82	784.78	407.98

(*) The branching selection strategy is performed by the 'Most fractional variable' (1), 'Worst assignment' (2), or 'Variable close to one' (3).

Table 2. Performance of the different branching strategies

The table reveals that combining the branching on the residual problem and the 0/1 effective branching strategy that selects the worst fractional assignment outperforms all other strategies, both in terms of the percentage of problems solved to optimality and the required CPU time. For the 0/1 branching strategy and the 1/2/…/|S| branching strategy, the selection strategy on the worst fractional assignment outperforms both other variable selection strategies. The strategy that selects the fractional variable closest to one performs consistently well for the constraint branching strategy, which confirms the findings of Vance et al. (1997). In general, the 0/1 branching strategy performs better than the other two effective branching strategies. Finally, the inclusion of the branching strategy on the residual problem is beneficial over the direct branching on the original variables without the creation of pseudo branches except for the branching strategy creating as many branches as the possible number of assignments and the constraint branching strategy selecting the variable closest to one. Branching on the residual problem typically requires less time for solving the master problem since a fraction of the nurses have their schedules fixed.

4.2 Assessment of node pruning strategies

In this section we analyze the contribution of the pruning rule of section 3.5 with and without the use of the initial heuristic solution approach of section 3.2. In table 3 we only display results for the branching strategy combining the branching on the residual problem and the 0/1 branching selecting the worst fractional assignment since this strategy outperforms all other branching schemes (see table 2).

		No pruning rules	Subset Dominance Rule (see section 3.5.2)
UB Search (see section 3.2)	# $nodes_{effective}$	18.35	12.85
	# $nodes_{total}$	24.42	18.92
	CPU_{IP}	543.29	394.42
	% $optimal_{IP}$	93.75%	100.00%
Without UB Search	# $nodes_{effective}$	29.00	17.69
	# $nodes_{total}$	39.58	24.58
	CPU_{IP}	1027.26	709.30
	% $optimal_{IP}$	84.38%	92.71%

Table 3. Performance of the different pruning strategies

Table 3 reveals the beneficial effect of incorporating the initial heuristic solution in the branch-and-price procedure in terms of all displayed parameters. The solution quality provided by this heuristic is much better than the solution quality of the constructive heuristic (see section 3.1) providing a much tighter bound during the search for the IP solution. Incorporating the pruning rule leads to more solutions solved to optimality within the time limit and requires significantly less CPU time and fewer nodes in the tree.

4.3 Incorporation of different speed-up techniques

Table 4 shows the contribution of the various speed-up techniques of sections 3.1 and 3.3 for the best performing procedure identified in section 4.1.

Speed-up techniques	Without speed-up	Constructive heuristic (section 3.1)	2 phased pricing (section 3.3.1)	Lagrange dual pruning (section 3.3.2)	Subproblem pruning (section 3.3.3)	Partial CG (section 3.3.4)	Combined speed-up
LP search							
# $iter_{LP}$	22.69	19.35	22.50	21.19	22.69	44.58	27.92
# $columns_{LP}$	680.77	580.38	675.00	635.77	678.65	494.85	470.31
$CPU_{LP,MP}$	5.62	2.50	5.33	5.01	5.61	7.39	5.93
$CPU_{LP,SP}$	171.94	147.79	147.53	149.48	175.08	143.31	138.73
CPU_{LP}	177.57	150.29	152.85	154.49	180.69	150.69	144.66
UB search							
# $iter_{UB}$	9.65	11.58	9.65	9.31	9.65	12.85	11.23
# $columns_{UB}$	289.62	347.31	295.38	279.23	120.73	372.27	110.12
$CPU_{UB,MP}$	0.10	0.09	0.10	0.07	0.11	0.10	0.10
$CPU_{UB,SP}$	10.10	10.94	11.12	8.32	8.48	12.19	8.59
CPU_{UB}	10.20	11.04	11.22	8.39	8.58	12.29	8.68
IP search							
# $iter_{IP}$	77.46	92.04	79.19	69.31	77.46	118.62	90.50
# $columns_{IP}$	916.22	978.46	909.98	541.82	899.23	845.77	490.01
$CPU_{IP,MP}$	19.55	23.78	22.58	17.57	19.55	59.83	40.41
$CPU_{IP,SP}$	374.72	382.04	370.18	235.41	364.81	341.33	200.66
CPU_{IP}	394.26	405.81	392.75	252.98	384.35	401.16	241.07
% $optimal_{IP}$	91.67%	92.71%	93.75%	96.88%	91.67%	92.71%	100.00%
Global information							
CPU_{MP}	25.27	26.37	28.00	22.64	25.27	67.32	46.44
CPU_{SP}	556.76	540.77	528.82	393.22	548.36	496.83	347.98
CPU	582.03	567.14	556.83	415.86	573.63	564.15	394.42

Table 4. Effects of the different speed-up techniques

In the following, we compare the results of each specific speed-up technique with the results of the base case, i.e., 'Without speed-up'.

The gain of implementing the *constructive heuristic* (see section 3.1) lies especially in the search for the LP relaxation of the different problem instances. The procedure needs fewer iterations and, hence, has to solve fewer subproblems to obtain the LP relaxation. The beneficial effect on the total needed CPU time can be entirely attributed to the time savings in the search for the LP relaxation. No effect (or even a small negative effect) can be observed in the search to the initial heuristic and the optimal solution.

The effect of employing the *two phase pricing method* (see section 3.3.1) can be mainly encountered in the search for the LP relaxation and the search for the optimal solution. In both steps, less time is needed to solve the pricing step although nearly the same amount of columns needs to be generated in the base case. Moreover, fewer iterations are needed of the column generation procedure to obtain the LP relaxation. Hence, selecting "good" columns speeds up the convergence of the column generation loop. The effect of the speed-up technique cannot be found back in the search for the initial heuristic solution due to the many 0/1 constraints on the individual nurse schedules causing the heuristic neighbourhoods to perform badly.

The *Lagrangean dual pruning* (see section 3.3.2) tails off the column generation loops which can be detected in all stages of the search. Fewer iterations are needed to obtain the LP relaxation in each node, which leads to a smaller number of columns to be generated. A huge reduction in CPU time can be detected for both the master problem and the subproblem.

The effect of the *upper bound pruning for the pricing problem* (see section 3.3.3) can mainly be observed in the search for the initial heuristic where a far higher amount of subproblems can be pruned compared to the other stages (i.e., ‘IP Search’ and ‘LP Search’). In this stage, a single nurse’s network is heavily constrained, which implies that the optimal solution of the relaxed problem deviates only little from the individual nurse’s schedule obtained by solving the pricing problem. However, the time savings are limited since the time to solve the constrained nurse’s network optimally using the pricing procedure is relatively small. The performance of this upper bound is typically strongly dependent on the case-specific time related constraints that need to be incorporated in a nurse’s network. The less resource constraints the better the pruning rule.

Implementing *partial column generation* (see section 3.3.4) leads to a higher number of master problem iterations, which causes the higher CPU time to solve the master problem in each stage of the search. However, this speed-up technique reduces drastically the number of subproblems needed to be solved especially in the search for the LP relaxation. Overall, this method has a positive effect on the required CPU time.

All described effects have a positive effect on the required CPU time. However, due to interactions between these speed-up techniques (e.g., Lagrangean dual pruning and two phase pricing) the total time gain is less than the sum of the time savings of all these stand-alone speed-up techniques as can be observed in the column ‘Combined speed-up’.

6 Conclusion

In this paper an exact branch-and-price procedure has been proposed for the mid-term nurse scheduling problem, i.e., assigning nurses to shifts in order to maximize the quality of the constructed roster, satisfying the hospital objectives and meeting the legal, union, hospital, and personal constraints imposed on the nurses’ individual schedules. The proposed objective function aims to minimize the weighted sum of penalty costs associated with the individual schedules the nurses are assigned to and with over- and understaffing the ward during a particular shift on a particular day. The latter are formulated in such a way the staff shortages and the staff abundances are levelled over the time horizon. The described nurse scheduling problem is very flexible in terms of objective function possibilities, nurses’ substitutability, case-specific time related constraints, and comprehends many real-world problem characteristics. Different branching strategies and node reduction mechanisms have been proposed and fine-tuned in order to improve the computational performance solving the nurse scheduling problem to optimality. Our computational results indicate that combining the 0/1 branching strategy selecting the worst fractional assignment with the branching strategy fixing the integer nurse schedules performs particularly well. Moreover, we have shown how the size of the branch-and-bound tree can be reduced considerably using different pruning techniques. We have also demonstrated the benefits of several speed-up techniques proposed in literature on the nurse scheduling problem in particular. We have tested all these theoretical principles on the problem structure of a ward in a Belgian hospital. Moreover, we have also investigated the sensitivity of the proposed optimization principles by varying systematically the problem characteristics in a controlled way testing the proposed procedure on an artificially generated dataset NSPLib.

As for future research, we have the intention to compare the obtained results for the case study with the procedure at hand with the results obtained by various commercial software packages, which typically thrive on heuristic procedures. In this way, we could benchmark various software packages in terms of computational performance whereas the packages now are mainly evaluated on their functionalities and compatibility with other hospital information systems. Furthermore, the dynamic nature of the demand for nursing services, coupled with absenteeism, personal days, and emergencies raises the need for constructing more robust rosters. Instead of adjusting and rescheduling the planned mid-term roster frequently during the course of the planned month, we aim to incorporate a part of this uncertainty upfront while constructing the mid-term nurse roster. In this way, we want to reduce the probability more costly options are needed in order to overcome staff shortages.

Acknowledgements

We acknowledge the support given to this project by the 'Bijzonder Onderzoeksfonds (BOF), Ghent University, Belgium' under the contract number B/04668/01. Furthermore, we would like to thank Filip Demeyere, nursing director, and Denis Goeminne, department head internal medicine, for their collaboration and useful information and giving us the permission to obtain insight in a real-life problem environment.

References

Abernathy, W. J., N. Baloff, J. C. Hershey. 1971. The Nurse Staffing Problem: Issues and Prospects. *Sloan Management Rev.* **12** 87-99.

Arthur, J. L., A. Ravidran. 1981. A multiple objective nurse scheduling model. *IIE Trans.* **13** 55-60.

Azaiez, M. N., S. S. Al Sharif. 2005. A 0-1 goal programming model for nurse scheduling. Comput. *Oper. Res.* **32** 491-507.

Balakrishnan, A., R. Wong. 1990. A network model for the rotating workforce scheduling problem. *Networks* **20** 25-42.

Bard, J. F., H. W. Purnomo. 2005a. Preference scheduling for nurses using column generation. *Eur. J. Oper. Res.* **164** 510-534.

Bard, J. F., H. W. Purnomo. 2005b. A column generation-based approach to solve the preference scheduling problem for nurses with downgrading. *Socio-Econ. Plann. Sci.* **39** 193-213.

Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, P. H. Vance. 1998. Branch-and-Price: Column Generation for solving huge Integer Programs. *Oper. Res.* **46** 316-329.

Beaumont, N. 1997. Scheduling staff using mixed integer programming. *Eur. J. Oper. Res.* **98** 473-484.

Beliën, J., E. Demeulemeester. 2005. Integrating nurse and surgery scheduling. DTEW Research Report 0526, K.U.Leuven, Belgium.

Beliën, J., E. Demeulemeester. 2006. Scheduling trainees at a hospital department using a branch-and-price approach. *Eur. J. Oper. Res.* **175** 258-278.

Berrada, I., J. A. Ferland, P. Michelon. 1996. A Multi-objective Approach to Nurse Scheduling with both Hard and Soft Constraints. *Socio-Econ. Plann. Sci.* **30** 183-193.

Billionnet, A. 1999. Integer programming to schedule a hierarchical workforce with variable demands. *Eur. J. Oper. Res.* **114** 105-115.

Brusco, M. J., T. R. Johns. 1995. Improving the Dispersion of Surplus Labor in Personnel Scheduling Solutions. *Comput. Ind. Eng.* **28** 745-754.

Burke, E. K., P. De Causmaecker, S. Petrovic, G. Vanden Berghe. 2003. Variable neighbourhood search for nurse rostering problems. M. G. C. Resende, J. P. de Sousa, eds. *Metaheuristics: Computer Decision-Making*, Kluwer, Mechelen, Belgium.

Burke, E. K., P. De Causmaecker, G. Vanden Berghe, H. Van Landeghem. 2004. The state of the art of nurse rostering. *J. Scheduling* **7** 441-499.

Caprara, A., M. Monaci, P. Toth. 2003. Models and algorithms for a staff scheduling problem. *Math. Program.* **98** 445-476.

Caron, G., P. Hansen, B. Jaumard. 1999. The Assignment Problem with Seniority and Job Priority Problems. *Oper. Res.* **47** 449-453.

Cline, D., C. Reilly, J. F. Moore. 2003. What's behind RN turnover?. *Nurs. Management* **34** 50-53.

De Reyck, B., W. Herroelen. 1998. A branch-and-bound procedure for the resource constrained project scheduling problem with discounted cash flows and generalized precedence relations. *Comput Oper. Res.* **25** 1-17.

Ernst, A. T., H. Jiang, M. Krishamoorty, B. Owens, D. Sier. 2004. Staff scheduling and rostering: A review of applications, methods and models. *Eur. J. Oper. Res.* **153** 3-27.

Falkner, J., D. Ryan. 1987. A bus crew scheduling system using a set partitioning model. *AsiaPac J Oper Res* **4** 39-56.

Fitzpatrick, T., L.Y. Farrell, M. Richter-Zeunik. 1987. An automated staff scheduling system that minimizes payroll costs and maximizes nurse satisfaction. *Comput Nurs* **5** 10-14.

Gamache, M., F. Soumis, G. Marquis, J. Desrosiers. 1999. A Column Generation Approach for large-scale Aircrew Rostering Problems. *Oper. Res.* **47** 247-263.

Haggerty, J.L., R.J. Reid, G.K. Freeman, B.H. Starfield, C.E. Adair, R. McKendry. 2003. Continuity of care: a multidisciplinary review. *Brit Med J* **327** 1219-1221.

Jaumard, B., F. Semet, T. Vovor. 1998. A generalized linear programming model for nurse scheduling. *Eur. J. Oper. Res.* **107** 1-18.

Kazahaya, G. 2005. Harnessing technology to redesign labor cost management reports. *Healthc. Financ. Management* **59** 94-100.

Mehrotra, A., K. E. Murphy, M. A. Trick. 2000. Optimal shift scheduling: A branch-and-price approach. *Nav. Res. Log.* **47** 185-200.

Morris, J. G., M. J. Showalter. 1983. Simple approaches to shift, days-off, and tour scheduling programs. *Management Sci.* **29** 942-950.

Musa, A. A., U. Saxena. 1984. Scheduling Nurses Using Goal Programming Techniques. *IIE Trans.* **16** 216-221.

Osogami, T., H. Imai. 2000. Classification of Various Neighbourhood Operations for the Nurse Scheduling Problem. *Lect. Notes Comput. Sci.* **1969** 72-83.

Ryan, D. M., B. A. Foster. 1981. An Integer Programming Approach to Scheduling. A. Wren, ed. *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*, North-Holland, Amsterdam, The Netherlands.

Schrage, L. 1995. *LINDO: Optimization software for linear programming*. LINDO Systems Inc., Chicago, IL.

Sol, M., M. Savelsbergh. 1994. A branch-and-price algorithm for the pickup and delivery problem with time windows. Memorandum COSOR 94-22, TUE, The Netherlands.

Trivedi, V. M., M. Warner. 1976. A branch and bound algorithm for optimum allocation of float nurses. *Management Sci.* **22** 972-981.

Vance, P.H., A. Atamturk, C. Barnhart, E. Gelman, E.L. Johnson, A. Krishna, D. Mahidhara, G.L. Nemhauser, R. Rebello. 1997. A heuristic branch-and-price approach for the Airline Crew Pairing Problem. Technical Report TLI/LEC-97-06, Georgia, Institute of Technology, Atlanta, GA.

Van den Akker, M., H. Hoogeveen, S. Van de Velde. 2002. Combining Column Generation and Lagrangean Relaxation to Solve a Single-Machine Common Due Date Problem. *INFORMS J. Comput.* **14** 37-51.

Vanderbeck, F., L. A. Wolsey. 1996. An exact algorithm for IP column generation. *Oper. Res. Lett.* **19** 151-159.

Vanderbeck, F. 2000. On Dantzig-Wolfe Decomposition in Integer Programming and Ways to Perform Branching in a Branch-and-Price Algorithm. *Oper. Res.* **48** 111-128.

Vanhoucke, M., B. Maenhout. 2005a. Characterisation and Generation of Nurse Scheduling Problem Instances. Working Paper 05/339, Ghent University, Belgium.

Vanhoucke, M., B. Maenhout. 2005b. NSPLib – A tool to evaluate (meta-)heuristic procedures. *Proceedings of the 31st Meeting of the European Working Group on Operational Research Applied to Health Services (ORAHS),* Southampton, Great Britain, Forthcoming.

Warner, H. W. 1976. Scheduling Nursing Personnel According to Nursing Preference: A Mathematical Approach. *Oper. Res.* **24** 842-856.

Welton, J. 2006. Paying for Nursing Care in Hospitals. *Am. J. Nurs.* **106** 67-69.

Zurn, P., C. Dolea, B. Stilwell. 2005. Nurse retention and recruitment: developing a motivated workforce. *Global Nurs. Rev. Initiative (WHO)* **4** 1-31.