

Hospital-wide reactive scheduling of nurses with preference considerations

JONATHAN F. BARD* and HADI W. PURNOMO

Graduate Program in Operations Research & Industrial Engineering, 1 University Station, C2200, The University of Texas, Austin, TX, 78712, USA
E-mail: jbard@mail.utexas.edu

Received March 2004 and accepted October 2004

This paper presents a new methodology for reactively scheduling nurses in light of shift-by-shift imbalances in supply and demand. In most hospitals, the nursing staff is given a midterm schedule that specifies their work assignments for up to 6 weeks at a time. However, emergencies, call-outs, and normal fluctuations in personnel requirements can play havoc with the schedule. As a result, it is necessary to make short-term adjustments, either by reallocating resources when shortages exist or by cancelling assignments when demand drops. The need to take into account individual preferences further complicates the process. The problem associated with making the daily adjustments is formulated as an Integer Program (IP) and solved within a rolling horizon framework. The idea is to consider 24 hours at a time, but to only implement the results for the first 8 hours. The IP is then re-solved for the next 24 hours after several hours have elapsed and new data are available, and so on. Initial attempts to solve 50-nurse problems with a commercial code proved to be unsuccessful and led to the development of a branch-and-price algorithm. Included in the algorithm are a feasibility heuristic to find the upper bounds and a cut generation procedure to improve the lower bound computations. A set-covering-type IP was used to find upper bounds and mixed-integer rounding cuts were used to tighten the relaxed feasible region. Although the effectiveness of all but the set covering heuristic proved to be marginal, most problem instances with up to 200 nurses were solved within 10 minutes.

1. Introduction

The growing gap between the supply of nurses and the demand for their services is one of the many factors bidding up the cost of healthcare. In many countries, the situation is now at the point where the rules for good practice are being stretched to their limits and patient care is in jeopardy (Spratley *et al.*, 2000). In response, hospital administrators are forced to rely on more expensive solutions, such as agency nurses and overtime, to meet their needs. The problem is exacerbated by the aging professional population, a shrinking cohort of entry-level graduates, the changing nature of the job, new life and work values, and a historical sense of disenfranchisement of the nursing staff from the decision-making process (Aiken *et al.*, 2002; Kimball and O’Neil, 2002).

As part of the effort to cope with personnel shortages, many hospitals have adopted scheduling policies that give increased weight to the preferences and requests of their nursing staff, often at a considerable cost. The expectation is that a more attractive work environment and increased flexibility to deal with personal matters will lead to higher

retention rates, and ultimately, to lower overall costs when one takes into account required outlays of anywhere from \$30 000 to \$50 000 to hire a nurse.

In the last two decades, most of the published research on nurse scheduling has concentrated on rostering with the aim of accommodating individual preferences. These take the form of requests to work specific shifts or to be given specific days off, and can be measured by various rules related to the number of working hours, shift sequence patterns or even nurse-to-patient ratios (see Cheang *et al.* (2003) for a survey). Typically, nurses are asked to sign up for shifts prior to the beginning of the planning horizon. At that time, they may also submit a list of requests to the nurse manager who decides which to approve immediately and which to defer in light of expected demand. The outcome is a midterm schedule for each nurse in the hospital.

Midterm scheduling fixes the work assignments for the permanent nursing staff for up to 6 weeks at a time. Each unit generates its own rosters independently using some measure of “average” demand as input. In this paper, we begin with the midterm schedule and address the problem of adjusting individual work assignments to account for daily fluctuations in the patient population, absenteeism, and emergencies. Possible options include the use of overtime, calling in nurses on their day off, using outside

*Corresponding author

resources and pool nurses, or living with the shortages. The problem is formulated as an integer program and solved within a rolling horizon framework, once prior to the beginning of each of three 8-hour shifts. Because only limited instances could be solved with a commercial code, we developed a branch-and-price algorithm that was seen to find good solutions to problems with up to 200 nurses in less than an hour, and in most cases, within a few minutes. This is extremely important in an operational environment that may change in a moments notice.

In the next section, we describe the daily adjustment problem and its complexity. The presentation is based on our experience at several medium-sized hospitals in the US but offers as much generalization as possible. In Section 3, we present the mathematical model for the problem. This is followed in Section 4 with a discussion of the solution methodology. Computational results for problem instances with up to 200 nurses working in 14 units are highlighted in Section 5. We close with some remarks on implementation and the effectiveness of the approach.

2. Problem statement

The dynamic nature of the demand for nursing services, coupled with sick leave, personal days, and emergencies, requires that the planned midterm schedule, developed for a month at a time, be adjusted on a shift-by-shift basis. It is either not permitted or extremely undesirable to alter a nurse's midterm schedule without his or her consent. Although this restriction limits the potential cost savings in the short run, it promotes normalcy and stability over time. In most hospitals, the rescheduling is performed throughout the day a few hours before the start of each of three standard 8-hour shifts: day (D = 7:00 a.m.–3:00 p.m.), evening (E = 3:00 p.m.–11:00 p.m.), night (N = 11:00 p.m.–7:00 a.m.).

To enable the process, patient acuity and census data are entered periodically into the hospital's management information system. The principal information includes projected admissions, discharges, and transfers. Also recorded or tracked are the available nurses (on call, pre-approved part-timers, casuals, floaters, agency options), those who are not to be called (no call list) and those who are absent (no show list). The sum of this information comprises the "shift view" of the situation. To make the staffing decisions as the day unfolds, it is necessary to have a "24-hour shift view" of each unit.

In forecasting demand, a major difficulty lies not only in determining the number of patients for whom care must be provided, but the level of care that will be required by each; i.e., the acuity of care. Siferd and Benton (1994) represented the number of nurses needed during a shift as a multiplicative model that takes into account mean patient acuity, number of patients, and the mean rate of change in acuities. Using simulation, they showed that the stochastic interplay of these factors can cause wide swings in coverage

requirements for subsequent shifts. In practice, however, demand is derived primarily from productivity data.

2.1. Roles and responsibilities

There are several levels of authority and responsibility within a hospital with respect to personnel scheduling. At the unit level, the nurse manager, clinical manager, or nurse in charge keeps track of the current situation and assesses whether the level of coverage is too low, appropriate, or too high for the number of patients and their acuity. When more nursing resources are available in a unit than are needed for a particular shift, the nurse manager has several options. Beginning with the least senior staff member, the first is to request a reassignment to another unit. If the nurse in question is not willing to float and is not contractually obligated to do so, her shift is cancelled and her supervisor notified. Generally speaking, nurses would rather be cancelled than floated, although individual preferences may be overridden when the need is critical. If a nurse is cancelled, then one of the following designations is used for the time off: vacation, personal day, holiday, or unpaid leave. Each has different cost consequences. This information is reported to the nursing resources director who is provided with worksheets that are updated by shift and show who is scheduled for duty in each of the units in their clinical areas.

In general, the nursing resources director in the hospital is responsible for ensuring that all units are covered. At his or her control are the float pool and other external resources. The monthly schedules for the outside nurses and internal float pool nurses are readily available to nursing services. The latter are provided by the float pool managers for critical care. Specific unit assignments are not made at this point, but are left to the supervisors as the daily demand comes into focus.

The daily rescheduling problem falls into the area of *disruption management* (Bard *et al.*, 2001; Clausen *et al.*, 2001). It is standard procedure for organizations to make midterm and long-term plans to fix permanent resources and develop an operating plan. Once the plan is implemented, however, demand fluctuations, absenteeism, equipment failures, and other unforeseen events call for periodic realignment. The goal of rescheduling is to reallocate the available resources in a way that minimizes the cost of the disruption. In doing so, it is important to minimize the differences between the new plan and the original plan. In the hospital environment, this often leads to conflict because the optimal course of action may impose undesirable schedules on the permanent staff, such as excessive overtime and long work stretches.

2.2. Daily adjustments

When a unit is short of staff, a number of corrective steps can be taken. Similarly, each has different cost consequences, although in most hospitals, cost is not foremost

on the minds of those directly responsible for a unit. Their primary concern is meeting demand, especially in critical situations. With this in mind, the order of action is typically:

1. Look for a volunteer in the unit to work the next shift (or fraction thereof) as overtime.
2. Try to reach unit staff who are not scheduled to work during the next 24 hours (casual and *per diem* nurses).
3. Find floaters from other units that might be overstaffed or draw on the float pool.
4. Cycle through the on-call list (used mostly in emergencies).
5. Have the nursing resources director call in agency nurses.
6. Cancel nurses whenever necessary.
7. Invoke mandatory overtime by requesting that a nurse on the current shift stay for the next shift. This is done in reverse order of seniority on a rotating basis so the most junior nurse is not necessarily singled out.

Several caveats and qualifications exist for each of these options. For example, when assigning either voluntary or mandatory overtime, there is a distinct possibility that the nurse will call in sick another day during the pay period. Although she will not be paid overtime if her total hours do not exceed 80 in the 14-day period, there are no negative consequences. In effect, she works overtime in exchange for a day off. A similar situation arises in practice when a nurse is called in on her day off to work a shift that is separated from her upcoming shift by only 8 hours (e.g., evening \rightarrow day). In this case, she is likely to call in sick for the day shift so little has been gained. This is called *double back* and should be avoided.

The first two options fall within the purview of the unit manager and are not included in our model. Instead, we focus on the hospital-wide problem of optimally allocating floaters, on-call nurses, and agency nurses. This problem must be solved at least three times per day, and falls within the purview of the nursing resources director.

3. Mathematical model

The majority of rules and constraints that govern the daily adjustment problem for nurses have been mentioned above. The complete set may vary among hospitals but generally reflects institutional policies, union agreements, state or federal statutes, and financial considerations. The overall goal is to satisfy coverage requirements at a minimum cost while taking into account nurse preferences, morale, the need for the perception of fairness, and the expected response of staff members whose work patterns are affected. In light of these considerations, the problem is formulated as an integer linear program for a predetermined planning horizon of, say, 24 hours, or three shifts, and solved using a rolling horizon strategy. That is, solutions are obtained for the three upcoming shifts (say, D, E, N) and the results are implemented for at least the first shift (D) and perhaps for all three shifts

(E, N as well). Then, within two or three hours of the next shift (E), the problem is re-solved for the next three shifts (E, N, D), and so on. It is a simple matter to include 12-hour shifts in the model as long as their start times coincide with one of the 8-hour shifts. We denote them by AM (typically 7:00 a.m.–7:00 p.m.) and PM (7:00 p.m.–7:00 a.m.).

In formulating the model, it is assumed that all cancellation and overtime costs are known, that demand is given or can be estimated accurately for each of the three shifts in the planning horizon, and that the status of all unit nurses, pool nurses, casuals, and agency nurses is known. This means that the nurse managers, the supervisors, and the nursing resources director all have up-to-date information on call outs, shortages, surpluses, floaters, and pool nurses.

The model is designed to reflect the point of view of the hospital and is intended for use by the nursing services office rather than the unit managers. Prior to running the model, the unit managers are expected to evaluate their current staffing needs and take whatever action is necessary and permitted. If a staffing shortage is in view, then they will either ask one or more of their nurses currently on the floor to work overtime, or try to reach casual or *per diem* nurses with whom they have a last minute arrangement. As mentioned, these decisions are not included in the model; however, if a nurse is not needed in her home unit and is willing to float before, during or after a regularly scheduled shift, then this option is included. As such, all overtime is considered to be voluntary. If shortages still exist after the model is run, the nursing services office may impose mandatory overtime, depending on the specific policies of the hospital.

The following notation is used in the developments. A shift can be either 8 or 12 hours and all periods are 4 hours.

Indices

- i = nurses;
- j = units; j_1 and j_2 are two different units;
- p = periods; p_1 = period immediately preceding the shift assignment in the midterm schedule; p_2 = period immediately following the shift assignment in the midterm schedule.

Sets

- J = units under consideration;
- T = time periods in planning horizon (6 periods = 1 day);
- $J(i)$ = units in which nurse i is qualified to work;
- $F(i)$ = units to which nurse i can float;
- S = shifts in planning horizon (D, E, N, AM, PM for one day);
- $T(i)$ = periods assigned to nurse i in the midterm schedule (regular time);
- $\bar{T}(i)$ = (4-hour) overtime periods that nurse i is allowed to work (other than assigned periods); $\bar{T}(i) \cup T(i) \subseteq T$;

- R = regular nurses who can float, work overtime, or both;
- P = pool nurses.

Input data

- c_{ijp}^1 = cost of assigning nurse i to unit j in period p (value depends on the type of assignment: either regular time or overtime; the type of nurse: regular and pool nurse; the working location: home unit or float unit)
- c_i^2 = cost of cancelling nurse i ;
- c_{jp}^3 = cost of assigning an on-call nurse to unit j during period p ;
- c_{jp}^4 = cost of assigning an agency nurse to unit j during period p ;
- D_{jp} = incremental number of nurses required in unit j during period p (+ means shortage, - means surplus);
- M = large penalty coefficient;
- p_i^1 = penalty for floating nurse i to another unit;
- p_i^2 = penalty for unproductive assignment (cancellation) of nurse i ;
- p_i^3 = penalty for assigning nurse i a split shift;
- p_i^4 = penalty for an on-call assignment;
- P^{\max} = maximum number of total undesirable patterns allowed;
- OT_i^{\max} = maximum number of overtime periods that nurse i can work;
- O_{jp}^{\max} = maximum number of on-call nurses available for unit j during period p ;
- Z_{jp}^{\max} = maximum number of agency nurses available for unit j during period p .

Decision variables

- x_{ijp} = (binary) 1 if nurse i (regular or pool) is assigned to unit j in period p ; 0 otherwise;
- z_{jp} = number of agency nurses assigned to unit j in period p ;
- o_{jp} = number of on-call nurses assigned to unit j in period p ;
- v_i = (binary) 1 if (either regular or pool) nurse i is assigned to a certain shift in the midterm schedule but is not needed (shift is cancelled);
- b_{ip} = (binary) 1 if nurse i is given a split assignment (regular or overtime) for a shift starting in period p ; 0 otherwise (a 12-hour shift to be split between two or three units);
- g_{jp} = number of gaps (shortages) in unit j during period p .

These parameter and variable definitions imply certain additional assumptions. The first is that we are dealing with several categories of nurses. Unit nurses, pool nurses, and agency nurses are all included in the midterm schedule, but

not all are identified by name. In particular, we only know how many agency nurses have been scheduled for each shift in each unit, and how many on-call nurses are available by period and unit. Because there are upper bounds on these two categories of nurses (denoted by Z_{jp}^{\max} and O_{jp}^{\max} , respectively), gaps or shortages may still exist in the derived schedule. The g_{jp} variables are used to account for unmet demand.

3.1. Formulation

The zero-one integer programming model for a fixed planning horizon and a single skill type is as follows.

$$\begin{aligned} \text{Minimize} \quad & \sum_{i \in R \cup P} \sum_{j \in J(i)} \sum_{p \in T(i) \cup \bar{T}(i)} c_{ijp}^1 x_{ijp} + \sum_{i \in R \cup P} c_i^2 v_i \\ & + \sum_{j \in J} \sum_{p \in T} c_{jp}^3 o_{jp} + \sum_{j \in J} \sum_{p \in T} c_{jp}^4 z_{jp} + M \sum_{j \in J} \sum_{p \in T} g_{jp}, \end{aligned} \quad (1a)$$

subject to

$$\sum_{i \in R \cup P} \sum_{p \in T(i) \cup \bar{T}(i)} x_{ijp} + z_{jp} + o_{jp} + g_{jp} \geq D_{jp}, \quad \forall j \in J, \quad \forall p \in T, \quad (1b)$$

$$\sum_{j \in F(i)} x_{ijp} + v_i = 1, \quad \forall p \in T(i), \quad \forall i \in R \cup P, \quad (1c)$$

$$\sum_{j \in J(i)} x_{ijp} + v_i \leq 1, \quad \forall p \in \bar{T}(i), \quad \forall i \in R, \quad (1d)$$

$$\begin{aligned} x_{ij_1 p} + x_{ij_2 p+1} - b_{ip} &\leq 1, \quad \forall p \in T(i) \cup \bar{T}(i), \\ &\quad \forall j_1 \neq j_2 \in J(i), \quad \forall i \in R \cup P; \end{aligned} \quad (1e)$$

$$\begin{aligned} \sum_{j \in J(i)} x_{ij,p+1} &\geq \sum_{j \in J(i)} x_{ijp}, \quad p = 1, \dots, p_1 - 1, \\ &\quad \forall i \in R \cup P, \end{aligned} \quad (1f)$$

$$\begin{aligned} \sum_{j \in J(i)} x_{ijp} &\geq \sum_{j \in J(i)} x_{ij,p+1}, \quad p = p_2, \dots, |T| - 1, \\ &\quad \forall i \in R \cup P, \end{aligned} \quad (1g)$$

$$\sum_{j \in J(i)} \sum_{p \in \bar{T}(i)} x_{ijp} \leq OT_i^{\max}, \quad \forall i \in R \cup P, \quad (1h)$$

$$\begin{aligned} \sum_{i \in R} \sum_{j \in F(i)} \sum_{p \in T(i)} p_i^1 x_{ijp} + \sum_{i \in R \cup P} p_i^2 v_i + \sum_{i \in R \cup P} \sum_{p \in T(i) \cup \bar{T}(i)} p_i^3 b_{ip} \\ + \sum_{j \in J} \sum_{p \in T} p^4 o_{jp} \leq P^{\max}, \end{aligned} \quad (1i)$$

$$\begin{aligned} x_{ijp} \in \{0, 1\} \forall i, j, p, \quad b_{ip} \in \{0, 1\} \forall i, p, \quad 0 \leq z_{jp} \\ \leq Z_{jp}^{\max} \text{ and integer } \forall j, p, \quad 0 \leq o_{jp} \leq O_{jp}^{\max}, \\ g_{jp} \geq 0 \text{ and integer } \quad \forall j, p, u_i, v_i \in \{0, 1\}, \forall i. \end{aligned} \quad (1j)$$

The objective function (1a) sums the costs of each alternative available for handling shortages. The first term represents the cost of assigning nurse i to unit j in period p . The value of the coefficient c_{ijp}^1 may differ by period and nurse depending on the wage rate, the type of overtime if relevant, and whether a regular nurse or pool nurse is being considered. Some hospitals also pay a differential when a

regular nurse is floated from his or her home unit. The second, third, and fourth terms are self-explanatory. The fifth term penalizes undercoverage when insufficient internal or external resources are available. The value of the coefficient M is set to be greater than $\max \{c_{jp}^A : j \in J, p \in T\}$.

Constraint (1b) tries to ensure that all demand is met in every unit $j \in J$ during each period $p \in T$ in the planning horizon. The cost structure in Equation (1a) guarantees that all options will be considered before a gap is reported. Instead of using shifts as the time unit, the constraint is written in terms of periods. This representation is best whenever two shift-types overlap, which is the case when 8- and 12-hour shifts are included in the model. When $D_{jp} = 0$, no action is necessary; when $D_{jp} < 0$, there is overcoverage in unit j so one or more nurses in that unit may be floated or cancelled. When $D_{jp} > 0$, the shortage can be made up by the various options. Because each shift covers several periods, however, the option to float a nurse is only feasible when all periods covered by her shift are in surplus.

Constraint (1c) restricts the units to which nurse i can float to the set $F(i)$ which is determined by her qualifications and preferences. It is only relevant for regular or pool nurses whose home units have excess coverage in period p . Cancellation is the complementary option and is represented by the variable v_i . Constraint (1d) is similar to Constraint (1c), except that it addresses the overtime periods. Although both constraints limit the assignment of nurse i to at most one unit in period p , the presence of the cancellation variable v_i has different implications for either constraint. In Constraint (1c), if $v_i = 0$, then nurse i will be floated during her regularly scheduled shift; in Equation (1d), if $v_i = 0$, then nurse i may be floated in an overtime period. However, if $v_i = 1$, then nurse i 's regular shift is cancelled without the possibility of overtime. The presence of v_i in Constraint (1d) implicitly prohibits the assignment of overtime without a regular assignment for regular nurses who either float or work overtime.

Care is needed in establishing Constraint (1c). Because pool nurses do not have home units, when $i \in P$, the index j is summed over all possible units for nurse i so $F(i) = J(i)$. In contrast, when $i \in R$ and Constraint (1c) is operative, nurse i is eligible to float so the index j is summed over the set $F(i)$. Pool nurses are not given overtime so Constraint (1d) can be dropped for that category. With regard to regular nurses, several assignments are possible. When a regular nurse is not needed in her home unit, for example, she may be floated or cancelled. If floated, she may work voluntary overtime in another unit. (Recall that voluntary overtime in the home unit is not included in the model.)

Constraint (1e) is used to track the occurrence of split shifts. In particular, the binary variable $b_{ip} = 1$ whenever regular or pool nurse i is assigned to two different units in two consecutive periods, the first being period p . The indices j_1 and j_2 indicate the different units. Split shifts are undesirable and are penalized in Constraint (1e). Note that

it is possible for a 12-hour shift to be split into two or three parts with appropriate penalty.

Constraints (1f) and (1g) ensure that when a nurse works overtime her schedule consists of consecutive periods. A nonconsecutive assignment might be a regular shift followed by an off period, followed by an overtime period. The presence of the cancellation variable v_i in Constraint (1c) automatically prohibits a nonconsecutive assignment during a regular shift; however, the fact that Constraint (1d) is an inequality does not rule out this phenomenon when overtime is included.

The first inequality, Constraint (1f), ensures that overtime assignments are nondecreasing from the first period of the planning horizon up through p_1 , the last period just before the regular shift. The second inequality, Constraint (1g), ensures that the overtime assignments are nonincreasing from the first period p_2 after a regular shift through the end of the planning horizon $|T|$. For example, if three periods follow a regular shift, these constraints rule out the possibility of a nurse working the first and third periods as overtime with the second period being off. For a nurse that may be called in on her day off, we set $p_1 = |T|$ in Constraint (1f), and omit Constraint (1g). In practice, it may be necessary to shorten the range of p in either constraint to account for work stretches immediately preceding and following the current planning horizon.

Labor laws and organizational policies limit the number of hours per day an employee can work. Depending on the length of the regular shift that nurse i is scheduled to work during the planning horizon, the maximum number of overtime periods that can be assigned to her is denoted by OT_i^{\max} . Constraint (1h) enforces this provision.

Constraint (1i) is designed to account for preference violations in the adjusted schedule. The intent is to limit the total number of undesirable patterns to no more than P^{\max} , a user-supplied parameter, as well as discourage the use of on-call nurses (fourth term on the left). Three different types of undesirable patterns are considered in the model: (i) floating a regular nurse from her home unit during a regular or overtime shift; (ii) cancelling a nurse; and (iii) splitting a shift. Finally, Constraint (1j) specifies the domain of the decision variables, including upper bounds on the number of agency and on-call nurses.

4. Solution methodology

The number of variables in model (1) is $O(|R \cup P| \times |J| \times |T|)$ so the complexity of the Integer Program (IP) grows exponentially with each of these parameters. Initial testing showed that 20-nurse instances could be solved in a few seconds with CPLEX 7.1. However, instances with 40 or more nurses failed to converge to within 1% of the LP lower bound after several hours of computations. A major source of difficulty was seen to come from the constraints in Equation (1e), which number $O(|R \cup P| \times |J| \times |T|)$ and

hence grow quadratically with the average number of units in which a nurse is eligible to work. To improve performance, we developed a branch-and-price (B&P) algorithm based on an alternate formulation of the problem.

4.1. Column-oriented model for daily rescheduling

Model (1) is constraint oriented, which means that there is typically one set of constraints for each restriction, rule, and logical relationship in the problem and one decision variable for each possible course of action. The principal decision variables, x_{ijp} , assign a nurse to a unit during a period. As an alternative, we now present a column-oriented model in which the decision variables represent full schedules for the planning horizon (Barnhart *et al.*, 1998; Vanderbeck and Wolsey, 1996). Such models closely resemble set covering formulations and require the enumeration of all possible assignments, at least implicitly. The objective is to select one assignment for each nurse so that the total cost is minimized.

Some additional notation is needed to describe the column-oriented version of Equations (1a)–(1j).

- k = index for alternative schedules;
- N = set of nurses to be scheduled; $N = R \cup P$;
- $K(i)$ = set of alternative schedules for nurse i ;
- c_{ik} = cost of assigning schedule k to nurse i (this value is a function of the number of regular hours assigned, the number of overtime hours assigned, or whether the nurse is cancelled);
- p_{ik} = penalty coefficient associated with schedule k for nurse i ;
- X_{ijp}^k = parameter equal to 1 if schedule k for nurse i covers period p on unit j , 0 otherwise;
- y_{ik} = binary decision variable equal to 1 if nurse i is given schedule k ; 0 otherwise.

We can now write the problem as:

$$\begin{aligned} \text{Minimize } & \sum_{i \in N} \sum_{k \in K(i)} c_{ik} y_{ik} + \sum_{j \in J} \sum_{p \in T} c_{jp}^3 o_{jp} + \sum_{j \in J} \sum_{p \in T} c_{jp}^4 z_{jp} \\ & + M \sum_{j \in J} \sum_{p \in T} g_{jp}, \end{aligned} \quad (2a)$$

subject to

$$\sum_{i \in N} \sum_{k \in K(i)} X_{ijp}^k y_{ik} + z_{jp} + o_{jp} + g_{jp} \geq D_{jp}, \quad \forall j \in J, \quad p \in T, \quad (2b)$$

$$\sum_{k \in K(i)} y_{ik} = 1, \quad \forall i \in N, \quad (2c)$$

$$\sum_{i \in N} \sum_{k \in K(i)} p_{ik} y_{ik} + \sum_{j \in J} \sum_{p \in T} p^4 o_{jp} \leq P^{\max} \quad (2d)$$

$$\begin{aligned} y_{ik} \in \{0, 1\}, \quad \forall i \in N, \quad k \in K(i), \quad 0 \leq z_{jp} \leq Z_{jp}^{\max}, \\ 0 \leq o_{jp} \leq O_{jp}^{\max}, \quad g_{jp} \geq 0 \text{ and integer}, \quad \forall j \in J, p \in T. \end{aligned} \quad (2e)$$

In this model, pool nurses and regular nurses belong to the same set N . The distinction between the two appears as part of the input. A regular nurse has a home unit as reflected in the midterm schedule, and is allowed to work overtime. In contrast, pool nurses do not have a home unit and are not given overtime. Their daily shift assignments are part of the midterm schedule but their unit assignments are generally not made until they report for work.

The objective function (2a) is designed to minimize the total personnel costs. The first term sums the cost of each alternative schedule k for nurse i , which is denoted by c_{ik} . The value of this coefficient depends on the number of working periods (regular or overtime) in the schedule as well as the base wage. The remaining terms are identical to the last three terms in the constraint-based objective function (1a).

Constraint (2b) is similar to Constraint (1b) and requires that the demand in unit j during each period p be covered whenever possible. The first term represents an assignment of a regular or pool nurse; the next two terms represent alternative resources: agency nurses (z_{jp}) and on-call nurses (o_{jp}); the last term is associated with a gap (g_{jp}). The parameter X_{ijp}^k is used to map the decision variable y_{ik} into the appropriate unit and period assignment. The set $K(i)$ contains all “legal” schedules for nurse i as well as the cancellation option. Each schedule in $K(i)$ must comply with all the rules embodied in Constraints (1c)–(1h). The domain definitions in Constraint (2e) place upper bounds on the number of on-call and agency nurses available in each unit. When gaps exist in a solution, the decision to fill them with expensive or unpopular options such as mandatory overtime is left to the nursing services office.

Equation (2c) ensures that each nurse $i \in N$ is given exactly one schedule. It is written as an equality because the cancellation option is included in the definition of $K(i)$. Constraint (1d) limits the total penalty of the adjusted assignments to the parameter value P^{\max} . The penalty for schedule k associated with nurse i is given by p_{ik} . This value is the sum of the penalties that result from undesirable patterns, as explained in the discussion of Constraint (1g). For nurse i and schedule $k \in K(i)$, it is computed as follows:

$$p_{ik} = \sum_{j \in F(i)} \sum_{p \in T(i)} p_i^1 x_{ijp} + p_i^2 v_i + \sum_{p \in T(i) \cup \bar{T}(i)} p_i^3 b_p. \quad (3)$$

The penalty associated with the use of on-call nurses is treated separately and is represented by the second term on the left-hand side of Equation (2d). Constraint (2e) indicates that all variables are restricted to be integral.

4.2. Column generation

An upper bound on the total number of schedules or columns for nurse i in model (2) is $O(|J(i)|^{|\bar{T}(i) \cup \bar{T}(i)|})$. When

a nurse is eligible, for example, to work in six units for up to four periods a day, the bound is $6^4 = 1296$. Because a medium-sized hospital may have to schedule anywhere from 100 to 200 nurses at a time over a 24-hour planning horizon, a more efficient procedure is needed than trying to solve model (2) directly with all feasible columns. A common alternative is to use Dantzig-Wolfe (D-W) decomposition with the necessary adjustments for integer variables (Wolsey, 1998). In this approach, a master problem is created from Equation (1a), (1b), (1g) and (1h) in our case, but with only a subset of feasible columns, which we denote by the set $\bar{K}(i)$. This gives rise to model (2) but with Constraint (2b) replaced with:

$$\sum_{i \in N} \sum_{k \in \bar{K}(i)} X_{ijp}^k y_{ik} + z_{jp} + o_{jp} + g_{jp} \geq D_{jp}, \quad \forall j \in J, \quad p \in T. \quad (3b)$$

4.2.1. Derivation of reduced costs

In the algorithm, additional columns are generated for each nurse by solving a series of pricing subproblems at each iteration. Constraints (1c)–(1f) along with the binary variables x_{ijp} and v_i define the feasible region for the subproblems. Because these constraints decompose by i , there will be one subproblem per nurse. The corresponding objective functions depend on the reduced costs associated with the current master problem. To see how to construct these functions, consider the relationship between c_{ik} and X_{ijp}^k in model (2). The parameter X_{ijp}^k is equivalent either to a decision that assigns nurse i to unit j in period p , which is denoted as x_{ijp} , or cancelling nurse i , which is equivalent to v_i . So, for nurse i working schedule k , we have:

$$c_{ik} = \sum_{j \in J(i)} \sum_{p \in T(i)} c_{ijp}^1 x_{ijp} + c_i^2 v_i.$$

If the dual variables for Constraints (2b), (2c) and (2d) are denoted by μ_{jp} , σ_i and τ , respectively, then for each nurse i , the reduced cost in the master problem for column k is:

$$\begin{aligned} \bar{c}_{ik} = c_{ik} - \pi \mathbf{A}_{ik} &= \sum_{j \in J(i)} \sum_{p \in T(i) \cup \bar{T}(i)} (c_{ijp}^1 - \mu_{jp}) x_{ijp} \\ &+ c_i^2 v_i - \sigma_i + \tau p_{ik} \end{aligned} \quad (4)$$

where $\pi = (\mu, \sigma, \tau)$, \mathbf{A}_{ik} is the ik th column of Constraints (2b)–(2d), and the parameter p_{ik} is given by the right-hand side of Equation (3).

At optimality, all reduced costs must be non-negative. To check this condition, we implicitly evaluate all \bar{c}_{ik} by maximizing the right-hand side of Equation (4) over Constraints (1c)–(1f) for each nurse i separately. If the result gives a negative objective value, the corresponding solution vector, call it $(x_{ijp}^k, v_i^k) \equiv (X_{ijp}^k)$, represents a new column to be added to the master problems. For each nurse i , we solve:

Subproblem i:

$$\begin{aligned} \text{Minimize} \quad & \sum_{j \in J(i)} \sum_{p \in T(i) \cup \bar{T}(i)} (c_{ijp}^k - \mu_{jp}) x_{ijp} + c_i^2 v_i \\ & + \tau \left(\sum_{j \in F(i)} \sum_{p \in T(i)} p_i^1 x_{ijp} + p_i^2 v_i + \sum_{p \in T(i) \cup \bar{T}(i)} p_i^3 b_{ip} \right) - \sigma_i, \end{aligned} \quad (5a)$$

subject to

$$\sum_{j \in F(i)} x_{ijp} + v_i = 1, \quad \forall p \in T(i), \quad (5b)$$

$$\sum_{j \in J(i)} x_{ijp} + v_i \leq 1, \quad \forall p \in \bar{T}(i), \quad (5c)$$

$$\begin{aligned} x_{ij_1 p} + x_{ij_2 p+1} - b_{ip} &\leq 1, \quad \forall p \in T(i) \cup \bar{T}(i), \\ &\forall j_1 \neq j_2 \in J(i), \end{aligned} \quad (5d)$$

$$\sum_{j \in J(i)} x_{ij,p+1} \geq \sum_{j \in J(i)} x_{ijp}, \quad p = 1, \dots, p_1 - 1, \quad (5e)$$

$$\sum_{j \in J(i)} x_{ijp} \geq \sum_{j \in J(i)} x_{ij,p+1}, \quad p = p_2, \dots, |T| - 1, \quad (5f)$$

$$\sum_{j \in J(i)} \sum_{p \in \bar{T}(i)} x_{ijp} \leq OT_i^{\max}, \quad (5g)$$

$$\begin{aligned} x_{ijp} &\in \{0, 1\}, \quad \forall j \in J(i), \quad \forall p \in T(i) \cup \bar{T}(i), \\ v_i &\in \{0, 1\}, \quad b_{ip} \in \{0, 1\}, \end{aligned} \quad (5h)$$

where the last term in the objective function (5a), is a constant.

4.3. Overview of the B&P algorithm

The first step in applying the B&P algorithm is to set up a restricted linear Master Problem (MP) with a subset of all feasible schedules, $\bar{K}(i)$, and to remove the integral restrictions on the variables in Equation (2e). The restricted MP is then solved. To check whether the LP solution obtained is optimal to the full MP, the pricing subproblems of Equations [(5a)–(5h) for $i \in N$] are solved sequentially. If the objective function value, Equation (5a), of subproblem i is less than zero, then the solution (x_{ijp}, v_i) is converted to a column of the form (X_{ijp}^k) in the MP and added to $\bar{K}(i)$. The restricted MP is then re-solved with the updated sets $\bar{K}(i)$, and the process is repeated until all columns price out favorably. This is the D-W component of the algorithm.

At this point, if all of the decision variables $(y_{ik}, z_{jp}, o_{jp}, g_{jp})$ are integral, then we have solved the integral version of the full MP as well as the original problem constituted by Equations (1a)–(1j). If any of these variables are fractional, then it is necessary to apply a Branch-and-Bound (B&B) algorithm the MP. Because the LP solved at the root node does not necessarily contain the columns that comprise the optimal solution to Equations (2a)–(2e) when branching constraints are imposed, the pricing subproblem of Equations (5a)–(5h) must be solved to determine whether additional columns should be added to the MP at each node of the B&B tree.

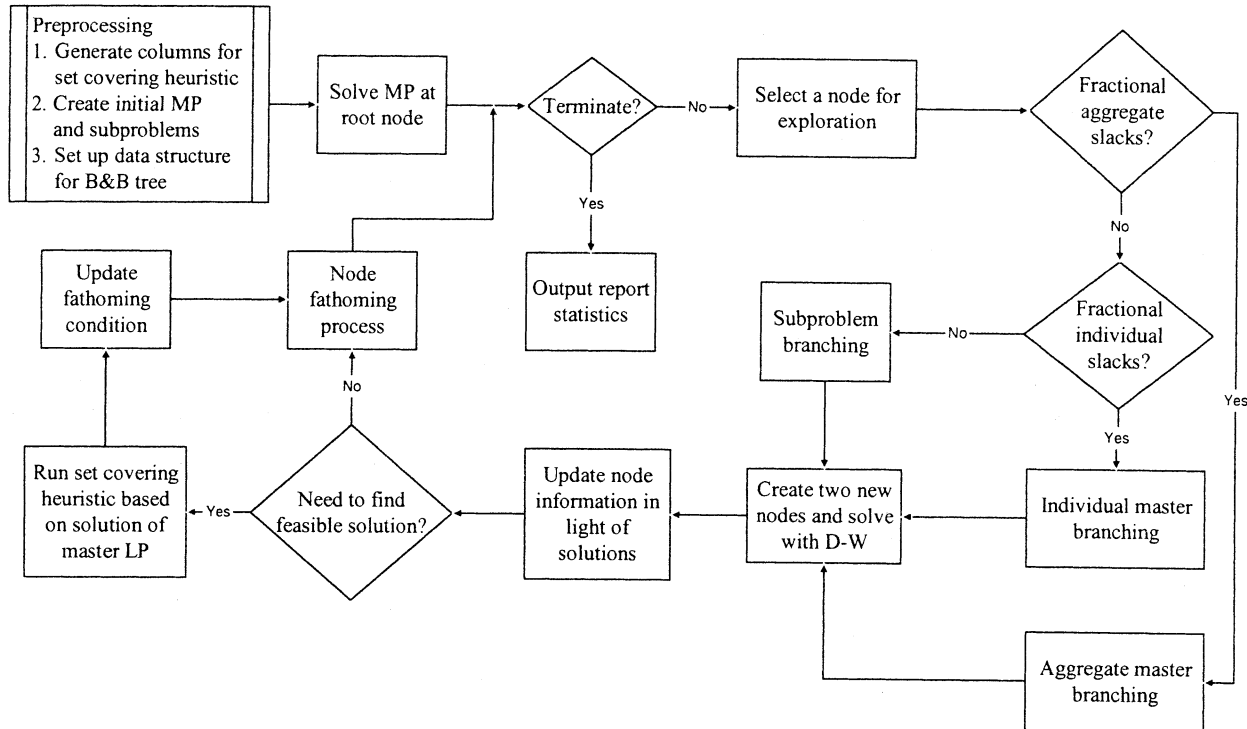


Fig. 1. Flowchart of the B&P algorithm.

Figure 1 identifies the major components of the methodology. In the preprocessing step, three operations are performed: (i) initial columns $\bar{K}(i)$ are generated for the restricted MP; (ii) a restricted set covering problem is set up to be used as a heuristic for finding feasible solutions; and (iii) the data structures are initialized for the B&B tree and the subproblems. The details are discussed in the remainder of this section. How each step is implemented is critical to the success of the overall approach.

In general, when one or more of the $(y_{ik}, z_{jp}, o_{jp}, g_{jp})$ variables are fractional, branching is required. Because of the known difficulties associated with branching on the y_{ik} variables in the MP, we first branch on the (z_{jp}, o_{jp}, g_{jp}) variables (master branching rule) and then on the subproblem variables (x_{ijp}, v_i) (subproblem branching rule). A depth-first search strategy is used to minimize the amount of data that must be stored. At the root node and at every 10 nodes thereafter, a heuristic is called to find a feasible solution ζ_v (see Section 4.5). The incumbent, call it $\bar{\zeta}$, is then updated in the usual manner by putting $\bar{\zeta} \leftarrow \min\{\bar{\zeta}, \zeta_v\}$.

The algorithm terminates when either a prespecified node limit is reached or the B&B tree is completely explored. Assuming that neither of these termination criteria is satisfied and node v is selected for branching, the fractional solutions in MP^v are identified, and depending on the circumstances, one of several branching rules is selected (see Section 4.4). Two descendent nodes, v_1 and v_2 , are created, and each is resolved using D-W decomposition. The following fathoming rules are applied to each:

1. MP is infeasible.
2. Within some tolerance, the LP solution is no better than the incumbent.
3. Solution is integral.

Unfathomed nodes are placed in a stack. This facilitates the implementation of depth-first search; i.e., last-in, first-out.

The principal attributes of the restricted MP^v that are stored at each node v include the LP objective function value, ζ_v , and the branching constraints imposed so far from the root node to node v . The former is used to determine whether the node should be fathomed and how the search will proceed otherwise. Storing branching constraints eliminates the need to maintain a backtracking data structure. Note that all columns are held in a single location and when MP^v is solved, the bound data on the subproblem variables at node v are used to set the corresponding MP variables, y_{ik} , to zero.

4.4. Branching rules

Applying standard B&B techniques to the restricted MP alone after the LP solution is found is inappropriate because there may exist columns not in the MP that are in the optimal solution to the original IP. Therefore, branching rules should be designed so that new columns can be easily priced out and when appropriate, added to the MP. This consideration makes it difficult to branch directly on the y_{ik} variables. Although it is straightforward to impose the restriction $y_{ik} = 1$ on subproblem i , imposing the

complementary restriction $y_{ik} = 0$ is all but impossible without major modifications to the subproblem at each iteration (Vanderbeck, 2000).

With this in mind, we have developed two alternative branching procedures. The first addresses fractional y_{ik} values in the MP, but is implemented through the subproblem variables (x_{ijp}, v_i) . This is a common approach and is discussed at length by Savelsbergh (1997) and Barnhart *et al.* (1998). The second is associated with the “slack” variables (z_{jp}, o_{jp}, g_{jp}) in the master problems.

4.4.1. Subproblem branching

When the restricted MP solution for nurse i is fractional, the logic of Constraint (2c) requires that at least two of the y_{ik} variables be fractional. Because no columns in the MP are identical for nurse i , the fractional variables represent alternative assignments, say $(X_{ijp}^{k_1})$ and $(X_{ijp}^{k_2})$. This means that in general nurse i has been assigned to two different units j_1 and j_2 in at least one period, p , so two different variables x_{ij_1p} and x_{ij_2p} , $j_1 \neq j_2$, will be positive in either Constraint (5b) or Constraint (5c). Rather than using the standard binary branching rule $x_{ijp} = 0$ on one branch and $x_{ijp} = 1$ on the other, we have adopted an SOS type-I strategy in which the variables in Constraints (5b) and (5c) are evenly divided into two subsets; e.g., $B(i)$ and $F(i) \setminus B(i)$ for Constraint (5b). This leads to a more balanced tree.

In general, the subproblem rule chooses a pair of fractional assignment variables y_{ik} from the MP and then identifies two corresponding subproblem variables x_{ij_1p} and x_{ij_2p} that are fractional by implication in, say, Constraint (5b). These are called the core variables. In our implementation of SOS branching, we assign $j_1 \in B(i)$ and $j_2 \in F(i) \setminus B(i)$, and partition the feasible region as follows for p fixed:

$$\sum_{j \in B(i)} x_{ijp} + v_i = 0 \text{ and } \sum_{j \in F(i) \setminus B(i)} x_{ijp} = 0. \quad (6)$$

The same partitioning scheme is applicable for Constraint (5c) but with one caveat. Constraint (5c), which limits the overtime assignment to at most one unit in period p , allows the possibility that only one core variable exists at the current node in the B&B tree. This happens, for example, when nurse i is assigned one period of overtime in schedule $(X_{ijp_1}^{k_1})$ and two periods in schedule $(X_{ijp_1}^{k_2})$ both in unit $j \in J(i)$, as well as the same regular shift. In this case, the branching strategy given in Equation (6) fails to partition the feasible region correctly. To see this, replace $F(i)$ with $J(i)$ in Equation (6), and let $x_{ij_1p_1} = 0$ in schedule k_1 and $x_{ij_1p_1} = 1$ in schedule k_2 for $j_1 \in B(i)$. Now, because there is only one core variable, imposing either constraint in Equation (6) at a node in the B&B tree will not eliminate schedule k_1 in the MP as required for a complete partitioning of the feasible region. To circumvent this difficulty, when only one core variable can be found, we use standard binary branching on the variable; i.e., $x_{ij_1p_1} = 0$ and $x_{ij_1p_1} = 1$.

A second issue that requires some discussion is the presence of the cancellation variable v_i in Constraints (5b) and (5c). Although this would seem to violate the principles of SOS branching, the follow result states otherwise.

Lemma 1. *The inclusion of the cancellation variable v_i in both Constraints (5b) and (5c) does not invalidate the use of SOS branching.*

Proof. It is sufficient to show that all feasible solutions to problem (5) will be explored when SOS branching is used. When $v_i = 0$, nurse i must be assigned a regular shift and may be assigned overtime. Standard SOS branching applies. When $v_i = 1$, nurse i is cancelled so there can be no regular or overtime assignments. These two cases allow for all possible assignment for nurse i . ■

Because cancelling a nurse greatly reduces the size of the feasible region, we first check the cancellation columns in the MP to see if any of them are fractional. This situation arises most frequently when nurse i is working the current shift and has volunteered for overtime, so Constraint (5b) does not apply to her. In this special case, the cancellation cost and penalty coefficients for nurse i are both zero. Therefore, when v_i is one of the core variables, the LP solution associated with the left-hand side of Equation (6) will usually be much smaller than the solution associated with the right-hand side which favors $v_i = 1$. Often, the left-hand node can be fathomed.

With subproblem branching, the MP and subproblems can be modified simply by changing the bound of the associated variables. In subproblem i , this means setting the branching variables in Equation (6) to zero in model (5). The modification of the MP is based on the term $(X_{ijp}^k)y_{ik}$. Fixing the branching constraint $\sum_{j \in B(i)} x_{ijp} + v_i = 0$ for some p and i , is equivalent to restricting columns in the MP to only those that have $v_i = 0$ and $x_{ijp} = 0$ for all $j \in B(i)$. This is enforced by fixing the upper bound of the corresponding y_{ik} variables to zero.

Example 1. Supposed at a given node in the B&B tree, two fractional MP variables, $y_{ik_1} = \frac{1}{2}$ and $y_{ik_2} = \frac{1}{2}$, have been identified for nurse i . Let $F(i) = J(i) = \{1, 2, 3, 4\}$. Using a six-component vector to represent the six periods in a day, let the assignments be $(X_{ijp}^{k_1}) = (100 \ 100 \ 1 \ 1 \ 2 \ 2)$ and $(X_{ijp}^{k_2}) = (100 \ 100 \ 1 \ 1 \ 3 \ 100)$, where the 100 indicates no assignment in the corresponding period. If the three shifts under consideration are D, E, N, then the first solution means that nurse i is assigned to unit 1 for 8 hours during E, followed by 8 hours of overtime in unit 2 during N. The second solution also has her working in unit 1 during E, but now her overtime is in unit 3 and is limited to 4 hours.

This situation gives rise to two branching possibilities, the first associated with period 5 and the second with period 6. When period 5 is selected, the SOS constraint,

Constraint (5c), is $x_{i15} + x_{i25} + x_{i35} + x_{i45} + v_i \leq 1$ and the core variables are x_{i25} and x_{i35} . Arbitrarily partitioning the remaining variables leads to the following constraints: $x_{i15} + x_{i25} + v_i = 0$ for the left-hand node and $x_{i35} + x_{i45} = 0$ for the right-hand node, where $B(i) = \{1, 2\}$. In addition, all columns in which nurse i is assigned to work on units 1 or 2 during period 5 must be removed from the left-hand node MP along with the cancellation column. Similarly, all columns in which nurse i is assigned to work on units 3 or 4 during period 5 must be removed from right-hand node of the MP.

When period 6 is selected, the SOS constraint is $x_{i16} + x_{i26} + x_{i36} + x_{i46} + v_i \leq 1$ but now there is only one core variable, x_{i26} . As a consequence, standard branching must be used. The subproblem associated with the left-hand node has the restriction $x_{i26} = 0$, and all columns that assign nurse i to unit 2 during period 6 must be removed from the MP. The subproblem associated with the right-hand node has the restriction $x_{i26} = 1$, implying that all columns that do *not* assign nurse i to unit 2 during period 6 must be removed from the MP.

The subproblem branching procedure can be summarized as follows:

- Step 1.* Find the fractional cancellation variable closest to 0.5; call it $y_{i_1k_1}$. Then, for nurse i_1 who is associated with this variable, find the next closest variable to 0.5; call it $y_{i_1k_2}$. If no cancellation variables are fractional, find the two fractional assignment variables that are closest to 0.5, say $y_{i_1k_1}$ and $y_{i_1k_2}$.
- Step 2.* Identify the core variables and the corresponding branching constraint, either Constraint (5b) or Constraint (5c). If only one core variable exists, go to Step 4, otherwise go to Step 3.
- Step 3.* Assign one core variable to the set $B(i)$ and the other to $F(i) \setminus B(i)$. Arbitrarily assign the remaining variables in the branching constraint to one of these sets as evenly as possible. Create two descendent nodes, one for each constraint in Equation (6). Modify the MP and subproblem accordingly.
- Step 4.* For the core variable x_{ijp} , create two descendent nodes by imposing the restriction $x_{ijp} = 0$ on the first and the restriction $x_{ijp} = 1$ on second. Make the appropriate modifications to the MP and subproblem.

If there is a weakness in B&B methods, it is revealed when symmetric solutions exist. In B&P, when several subproblems are similar, eliminating a particular solution in one does not eliminate its counterpart in the others (Barnhart *et al.*, 2000; Vanderbeck, 2000). As a result, little or no change occurs in the MP objective function values as the depth of the B&B tree increases. In the daily adjustment problem, symmetry is evidenced most visibly when the same period is repeatedly selected for branching. If nurse i_1 is excluded from working in some period as a result of the branching decision, then she is likely to be replaced by nurse

i_2 in that period when the two have similar profiles. Assuming that their wage rates differ marginally, there will be little or no change in the LP solution. The same fractional schedules may result in two different levels in the B&B tree, with the only difference being the actual nurses assigned to a unit.

In the daily adjustment problem, symmetric solutions are unavoidable because most demand periods can be covered by a majority of the available nurses, most having similar or identical profiles. To reduce the effects of symmetry, branching rules that are not based on the individual subproblems should be introduced. In our application, we consider branching on the slack variables in the MP, as well as the generation of cuts from the MP.

4.4.2. Master branching

Even when the y_{ik} variables are integral in a solution to the restricted MP, the slack variables (z_{jp} , o_{jp} , g_{jp}) in Constraint (2b') may be fractional. This follows because the on-call variables, o_{jp} , are likely to be fractional when the penalty constraint, Constraint (2d), is tight in an LP solution. We have implemented two master branching rules to take this situation into account. The first is called the aggregate master rule and forces the sum of the slack variables that cover the same unit $j \in J$ to be integral. The following two branching constraints are used to extend the B&B tree to the next level: $\sum_{p \in P} (z_{jp} + o_{jp} + g_{jp}) \leq \lfloor a \rfloor$ and $\sum_{p \in P} (z_{jp} + o_{jp} + g_{jp}) \geq \lceil a \rceil$, where a is the (fractional) sum of the slack variables at the current node for a given j .

The second is called the individual master rule and forces the integrality of individual variables. Applying it to the on-call variables, for example, gives the branches $\lfloor o_{jp} \rfloor \leq a$ and $\lceil o_{jp} \rceil \geq a$. Our empirical results indicate that the individual master rule should be used only when it is no longer possible to apply the aggregate master rule.

These rules provide several advantages for the B&P algorithm. First, they are easy to implement and second, they do not affect the pricing subproblems. Because the slack variables appear only in the restricted MP, the dual variables associated with the master branching constraints play no part in the modification of the subproblem objective functions. Finally, these rules target a particular covering constraint in Equation (2b') rather than a particular nurse. Because these constraints (as well as the on-call variables) apply to multiple nurses, the symmetry effect is eliminated.

4.4.3. Implementation

Initial testing suggested the following order for branching: (i) aggregate master rule; (ii) individual master rule; and (iii) subproblem rule. Applying the master rules first, forces integrality of the slack variables in Constraint (2b') and reduces the chances that the same constraints are repeatedly selected during subproblem branching. Empirically, this led to smaller search trees and more diverse feasible solutions when the heuristics were applied. Also, because the cost

coefficients of the slack variables are much larger than the cost coefficients of column variables, y_{ik} , the lower bounds increased rapidly as the tree was extended. This increased the likelihood that the corresponding nodes would be fathomed.

By using depth-first search and giving priority to master branching, fractional solutions at a parent node are noticeably different than the LP solutions at subsequent nodes. As a consequence, the feasible solutions found by our heuristic are noticeably different as well, providing the diversification necessary to broadly explore the feasible region.

4.5. Set covering heuristic

The ability to find high-quality feasible solutions is crucial to the successful implementation of the B&P algorithm. If the algorithm should terminate before the convergence criteria are met, at least a feasible solution is available. Moreover, good feasible solutions facilitate fathoming by bounds, thus reducing the size of the B&B tree. As a rule, complementary heuristics should have short computation times and make use of the fractional solution at a node. We have developed two heuristics for the daily adjustment problem, the first a tabu search algorithm and the second a restricted set covering IP. The discussion here is limited to the latter due to its superior performance. For a description of the tabu search algorithm, see Purnomo (2005).

The set covering heuristic is based on the observation that when the D-W procedure terminates with an LP solution to problem (2), if a subset of the columns contained in the restricted MP correspond to the optimal solution of the daily adjustment problem, then solving the restricted MP as an IP will yield the solution to the original problem (1). If some but not all of the columns are in the optimal set, then solving the restricted MP as an IP may still yield a good feasible solution to problem (1).

With this in mind, we found it advantageous to initialize model (2) with a set of “good” columns. After solving several restricted MPs starting with the slack variables only, we observed, not surprisingly, that most columns in an optimal basis had low cost and penalty coefficients. We also noticed that columns in which nurses were assigned to more than two units had high cost and penalty coefficients. Therefore, we define “good” columns for nurse i as those in which she is assigned to the same unit during her regular 8-hour shift but can work overtime in any unit $j \in J(i)$. However, if two contiguous periods of overtime are assigned, then they must both be in the same unit. Assuming that periods 3 and 4 correspond to the regular shift which is to be worked in unit j_1 , this definition allows schedules of the following form: $(0, 0, j_1, j_1, 0, 0)$, $(0, j, j_1, j_1, 0, 0)$, $(0, 0, j_1, j_1, j, 0)$, $(j, j, j_1, j_1, 0, 0)$, $(0, 0, j_1, j_1, j, j)$, $(0, j_2, j_1, j_1, j_3, 0)$ for $j \in J$. For a problem with 70 nurses, the total number of good columns can grow to 8 000 depending on the nurse profiles considered.

Set-Covering-Algorithm

Input: Set of available nurses ($\hat{N} \subseteq N$), solution of the MP $\{\hat{y}_{ik} : \forall i \in \hat{N}, k \in K(i)\}$, demand over the planning horizon by unit $\{D_{jp} : \forall j \in J, p \in T\}$, incumbent $(\sigma^{\text{inc}}, \bar{\zeta}^{\text{inc}})$, where $\sigma(i) = (j_1(i), \dots, j_{|T|}(i))$ is a $|T|$ -component vector that represents a schedule for nurse i over the planning horizon.

Output: Improved solution, $\sigma^{\text{inc}} = \{\sigma(i)^{\text{inc}} : \forall i \in \hat{N}\}$ and
Step 0. (Initialization.) Set $N_u = \emptyset$ (unscheduled nurses) and $N_s = \emptyset$ (scheduled nurses). If at root node, generate “good” schedule for all nurses and build the restricted Set covering (SC) problem (2a)–(2e); otherwise, use existing columns.

Step 1. (Fix schedules of some nurses.) For all $i \in \hat{N}$ and $k \in K(i)$. If $(\hat{y}_{ik} > 0.99$ for $i = \hat{i}$ and $k = \hat{k})$, then select schedule \hat{k} and put:

$$\begin{aligned} N_s &\leftarrow N_s \cup \{\hat{i}\} \\ \sigma(i)^{\text{feas}} &\leftarrow \{X_{ijp}^{\hat{k}} : \forall j \in J, p \in T\} \\ D_{jp} &\leftarrow D_{jp} - X_{ijp}^{\hat{k}}, \forall j \in J, p \in T \\ P^{\text{max}} &\leftarrow P^{\text{max}} - p_{i\hat{k}} \end{aligned}$$

Else, put $N_u \leftarrow N_u \cup \{\hat{i}\}$.

Step 2. (Remove scheduled nurses.) For all $i \in N_s$, set the right-hand side of the SOS constraint, Constraint (2c), to zero.

Step 3. Solve updated version of problem (2a)–(2e) as an IP to get $(y_{ik}^*) \forall i \in N_u$ and objective function value $\bar{\zeta}^*$. Put $\sigma(i)^{\text{feas}} \leftarrow \{X_{ijp}^{k^*}, \forall j \in J, p \in T\} \forall i \in N_u$.

Step 4. If $(\bar{\zeta}^* < \bar{\zeta}^{\text{inc}})$, then put $\bar{\zeta}^{\text{inc}} \leftarrow \bar{\zeta}^*$ and $\sigma(i)^{\text{inc}} \leftarrow \sigma(i)^{\text{feas}} \forall i \in \hat{N}$.

At the root node, the restricted SC problem (2a)–(2e) is built with “good” columns and solved as an IP. No columns are added at subsequent nodes, but some may be removed as a result of branching restrictions. As the computations continue, after the restricted SC is solved, the columns associated with schedule variables $y_{ik} > 0.99$ are selected as part of the solution and removed from the IP by setting the right-hand side of Equation (2c) to zero.

4.6. Column management and cuts

A number of researchers have suggested that it may be more efficient to try to generate several columns from each subproblem (5a)–(5h) when checking for optimality, rather than only the one that has the most negative reduced cost (Barnhart *et al.*, 1998). However, because we solve subproblem (5a)–(5h) as an IP, it is not practical to try to identify more than one column at a time. When we perturbed an optimal schedule by adding, removing or switching a unit assignment, we found that it was nearly impossible to obtain a new schedule that had a negative reduced cost. Empirically, we found that only a small fraction of feasible solutions

priced out negatively. As a consequence, at most one column from a subproblem was added to the MP during the pricing operation. However, once a column was added, it was only removed when it violated a branching constraint.

The efficiency of implicit enumeration depends largely on the ability to fathom nodes by bounds high in the tree. Whereas heuristics help to reduce the upper bound in general, the lower bound at a node is most often determined by solving the corresponding LP. One way to improve the lower bound is to incorporate strong valid inequalities in the original formulation. This tightens the LP relaxation. Unfortunately, there are no specialized valid inequalities suitable for Constraints (2b)–(2d). Although the knapsack-like structure of the penalty constraint of Equation (2d) suggests that a generalized cover cut may be effective, we could not find any such violations in our initial testing. Instead, we investigated the use of Mixed-Integer Rounding (MIR) cuts, which were added to the restricted MP after the LP solution was obtained. To obtain the new bound, it was necessary to reoptimize MP with the D-W procedure.

For the purpose of generating MIR cuts, we used the penalty constraint of Equation (2d) only and treated the on-call variables, o_{jp} , as continuous. In simplified form, this constraint is:

$$y + o \leq b \tag{7}$$

where y corresponds to the first summation on the left-hand side of Equation (2d) and o corresponds to the second summation. Note that if all the penalty coefficients p_{ik} are integer, then y must be integral at optimality. Based on Proposition 8.6 in Wolsey (1998), a valid inequality for Equation (7) is $y + (1 - f)o \leq \lfloor b \rfloor$, where $f = b - \lfloor b \rfloor$. Of course, for this inequality to be useful, b must be fractional. To create fractional right-hand side values, we serially divide Equation (7) by the coefficients p_{ik} , starting with the smallest and then round down all the coefficients of each y_{ik} . For the trial coefficient $p_{i^*k^*}$, we get the cut:

$$\sum_{i \in N} \sum_{k \in K(i)} \left\lfloor \frac{p_{ik}}{p_{i^*k^*}} \right\rfloor y_{ik} + (1 - f) \left(\frac{p^A}{p_{i^*k^*}} \right) \times \sum_{j \in J} \sum_{p \in T} o_{jp} \leq \left\lfloor \frac{p^{\max}}{p_{i^*k^*}} \right\rfloor, \tag{8}$$

where

$$f = \frac{p^{\max}}{p_{i^*k^*}} - \left\lfloor \frac{p^{\max}}{p_{i^*k^*}} \right\rfloor,$$

If Equation (8) is violated at the current LP solution $(\hat{y}_{ik}, \hat{o}_{jp})$, it is added to the restricted MP.

After cycling through all possibilities, if at least one cut is found, MP is reoptimized with the D-W procedure. In the process, all that is necessary is to replace the pricing subproblem objective function coefficient τ in Equation (5a) with $\tau + \gamma/p_{i^*k^*}$, where γ is the dual variable associated with constraint Equation (8).

Lemma 2. *When an MIR cut is added to the restricted MP constituted by Equations (2a)–(2e), the new solution will have $\tau = 0$ and the added cut will be binding.*

Corollary 1. *Let ζ^{MP} be the solution to the restricted MP before an MIR cut is added and let ζ_{cut}^{MP} be the solution after reoptimization. If the coefficient $\gamma/p_{i^*k^*}$ derived from the dual variable associated with the cut is not included in the objective function of the pricing problem when the D-W procedure is applied, then the new solution is the same as the old; that is, $\zeta^{MP} = \zeta_{cut}^{MP}$.*

Proof. When a cut is added to the restricted MP, say of the form given by Equation (8), all columns in the LP are augmented by one component. When reoptimization begins, the objective function will initially increase because the current solution is no longer feasible. However, the D-W procedure will reintroduce all columns in the original optimal basis of the restricted MP that were modified when Equation (8) was added. In particular, those columns with a nonzero entry in Equation (8) that were in the original optimal basis will price out negatively because the coefficient $\gamma/p_{i^*k^*}$ is not included in objection function (5a). The additional basic column, needed because of the additional row, will be identified when feasibility is first attained. At optimality, the corresponding basic variable will be zero so the original solution will be feasible, giving $\zeta^{MP} = \zeta_{cut}^{MP}$. ■

5. Computational experiments

The column generation approach was implemented in Visual C++ and linked to the CPLEX 7.1 callable libraries which were used to solve the LP MP constituted by Equation (2a)–(2e) and the IP subproblems (5a)–(5h). All computations were performed on a 1.1 GHz PC.

To test the methodology, five “difficult” problem sets were created based on operational data obtained from several US hospitals. Table 1 identifies the parameter settings for these instances, which are divided into five groups ranging in size from 20 to 200 nurses. Column 2 indicates the total number of nurses and column 3 gives a breakdown by type (X1 – X2 – X3 – X4), where X1 = number of nurses who can be floated during their regular shift and who are available for overtime, X2 = number of nurses who only float, X3 = number of nurses who are available for overtime but do not float during their regular shift, and X4 = pool nurses. The number of units in which a nurse is qualified to work, $|F(i)|$ or $|J(i)|$, was randomly determined by sampling from a uniform distribution, $U(3,7)$. The average number of units is given in column 4. For problems 1–10, a total of 10 units was considered; for problems 11–25 we increased this number to 14.

The remaining input characteristics given in Table 1 are the supply-demand relationship and the maximum penalty value, P^{\max} . For the 24-hour planning horizon, the

Table 1. Summary of the problem instance characteristics

<i>Problem no.</i>	<i>Total no. of nurses</i>	<i>Composition of nurse types</i>	<i>Average units per nurse</i>	<i>Excess supply (hours)</i>	<i>Demand (hours)</i>	<i>Maximum penalty</i>
1	20	10-0-0-10	4.30	244	204	80
2	20	5-5-0-10	4.30	204	184	80
3	20	5-5-5-5	4.55	196	184	70
4	20	10-5-0-5	4.55	236	236	80
5	20	10-5-0-5	4.55	236	184	80
6	50	10-10-20-10	4.56	456	588	250
7	50	20-15-10-5	4.56	544	588	275
8	50	30-0-0-20	4.56	628	568	225
9	50	30-0-0-20	4.56	628	588	225
10	50	10-10-20-10	4.35	456	568	225
11	80	25-10-25-20	5.11	1376	860	250
12	80	25-10-25-20	5.11	1376	672	215
13	80	30-0-25-25	4.86	1200	624	200
14	80	20-20-20-20	4.86	1536	720	300
15	80	20-20-20-20	4.86	1536	780	300
16	150	30-30-60-30	4.23	1352	860	185
17	150	25-25-80-20	4.23	1296	628	300
18	150	40-10-60-40	4.23	1472	892	285
19	150	25-25-80-20	4.23	1296	892	250
20	150	30-30-60-30	4.23	1352	904	220
21	200	30-30-110-30	4.07	1656	860	151
22	200	50-25-75-50	4.11	1948	936	275
23	200	20-20-60-100	4.29	1844	1032	245
24	200	30-30-110-30	4.07	1656	936	150
25	200	50-50-50-50	4.12	2020	1428	500

“excess supply” in column 5 indicates the maximum possible working hours for all nurses, as determined by their profiles. Some nurses are only available during their regular shift and others only for overtime. In addition, a nurse cannot be scheduled for more than 16 hours a day. The “demand” in column 6 identifies the extent of undercoverage in the hospital. This requirement is satisfied from a combination of internal and external resources, or is represented by a “gap” in the solution. Internal resources include floaters, overtime, and pool nurses, whereas external resources include agency and on-call nurses. The fact that supply exceeds demand in all problem instances simply indicates that there is a mismatch between the two.

The total penalty in the last column of Table 1 determines the tightness of Constraint (2d). When this constraint is tight in an LP solution, many of the y_{ik} and o_{jp} variables are fractional, thus making the original IP more difficult to solve. Before fixing P^{\max} , we first solved between five and 10 instances in each of the five groups cases and selected the values that led to the most difficult problems. The other penalty coefficients used in Constraint (2d) are listed in Table 2.

In addition to the tightness of Constraint (2d) and the number of nurses under consideration, there are two other input factors that make a problem difficult. The first is the proportion of assignment types given in column 3 of Table 1.

In general, the more periods that a nurse may be scheduled, the larger the feasibility region of the associated subproblem (5a)–(5h) and hence the more difficult it is to solve. Nurses who float and are available for overtime may be assigned between two to four periods, whereas nurses who are only available for overtime may be assigned either one or two periods.

The second factor is the number of units to which a nurse may be assigned. This value determines the cardinality of $F(i)$ and $J(i)$, and hence the number of x_{ijp} variables in subproblem i . Taken together, these two factors give an upper bound on the total number of possible schedules for a nurse. For example, a nurse who can work in any of five units in four different periods has $5^4 = 625$ possibilities that must be considered.

Implementation: Preliminary testing of the B&P algorithm suggested that several components be eliminated. The first was the tabu search heuristic, which gave disappointing

Table 2. Penalty settings

<i>Source of violation</i>	<i>Penalty</i>
Floating a regular nurse from home unit for 4 hours, p_i^1	3
Cancelling a nurse, p_i^2	3
Splitting a shift during consecutive working hours, p_i^3	8
Using an on-call nurse, p^4	14

results. For an 80-nurse problem, for example, it took 45 seconds to perform 50 iterations, whereas the D-W procedure took only 6 seconds on average to solve the restricted MP at each node in the search tree. Moreover, tabu search converged slowly and rarely found a feasible solution that was more than 1% below the incumbent. In B&P, the level of effort required for such small gains cannot be justified. Therefore, we only used the set covering heuristic for the upper bound computations. It was run at every five nodes for up to 30 seconds, but rarely reached this limit. At the root node, the resultant optimality gap was always within 7% and usually much less.

A second component that we eliminated was cut generation. Adding cuts improved the LP bound initially, but when the restricted MP was reoptimized, the new bound was only a fractional percentage above the original. In view of these results, we did not feel that further investigation was warranted. Some statistics are provided at the end of the section.

5.1. Solution characteristics of test problems

The performance of the B&P algorithm was measured by the solution quality and computational effectiveness. Table 3 summarizes the solution quality for the adjusted schedules in terms of total cancellations, required overtime

hours, on-call nurses, and unmet demand (gap). The computations were halted when all nodes were fathomed due to infeasibility or bounds, or when the total number of nodes explored reached a threshold value of 1000.

Column 2 shows the total number of cancelled (regular or pool) nurses who are scheduled to work a regular period but are not needed in their home unit and are not floated. These nurses receive monetary compensation equal to two hours of their basic wage. Column 3 (not needed) indicates the number of nurses who have volunteered for overtime in another unit but are not scheduled and do not have to be compensated. Nurse-unit compatibility determines, in part, whether overtime is assigned.

The next five columns in Table 3 deal with assigned internal resources. There are two possibilities: overtime and float. For the former, we report total overtime hours in column 4 and average overtime hours in column 5. The average is calculated by using only overtime nurses in the denominator. A nurse can work for up to eight hours of voluntary overtime. Column 6 gives the total hours of demand covered by floaters, whereas “total floats” in column 7 gives the number of nurses who are floated to other units. The next three columns (agency, on-call, gap) report the number of hours filled by outside resources (agency, on-call) and the total number of uncovered hours (gap). The last column gives the average penalty violation per

Table 3. Solution quality for test problems

<i>Problem no.</i>	<i>Cancelled</i>	<i>Not needed</i>	<i>Total overtime hours</i>	<i>Avg overtime hours</i>	<i>Floaters (hours)</i>	<i>Total floats</i>	<i>Agency hours</i>	<i>On-call hours</i>	<i>Gap hours</i>	<i>Average penalty</i>
1	5	0	8	8	132	15	48	0	0	5.42
2	7	0	0	0	112	13	32	0	4	4.87
3	3	3	28	7	92	11	80	0	8	6.00
4	4	3	28	5.60	92	11	96	0	36	5.75
5	0	0	20	6.67	124	17	48	0	4	8.67
6	3	2	176	6.52	204	23	144	12	80	7.31
7	5	0	148	6.73	276	32	144	0	56	7.22
8	1	0	124	6.53	360	43	144	0	4	9.08
9	1	0	128	6.40	376	44	128	0	16	9.08
10	2	2	172	6.37	220	25	128	12	84	7.50
11	7	4	240	7.50	452	46	32	0	64	6.52
12	7	12	172	7.82	352	38	32	8	36	5.69
13	7	7	323	6.84	396	48	48	0	8	5.74
14	12	3	188	6.06	388	44	112	0	40	5.08
15	0	0	228	6.71	592	65	48	0	60	6.67
16	1	31	284	6.45	520	60	32	0	24	5.71
17	0	32	352	6.18	500	54	32	0	28	6.12
18	15	38	172	5.93	624	65	16	0	0	5.35
19	10	35	352	6.29	364	38	32	0	36	5.41
20	7	26	296	6.44	496	56	64	0	12	5.27
21	1	70	328	6.69	484	55	32	0	12	5.77
22	30	60	112	5.60	716	78	0	0	0	4.55
23	35	50	40	4	1020	105	0	0	0	4.40
24	6	58	388	6.06	440	49	32	0	36	5.00
25	0	15	464	6.27	980	120	32	0	8	9.71

nurse. Only nurses with nonzero penalty are included in the calculation.

The solution quality, of course, is useful for nurse managers because it informs them of the adequacy of their resources to meet uncovered demand. In general, the interaction of such factors as the types of nurses available, excess supply hours, demand hours, and maximum penalty determine the allocation of resources. When demand hours, for example, are substantially lower than excess supply hours, there is likely to be a surplus of nurses. This translates into a high number of cancellations and limited overtime. Moreover, the cost structure favors pool nurses, floaters, and overtime hours in that order when demand exceeds the supply provided by the midterm schedule in any period. A final point about these results is that the number of nurses cancelled (column 2) or who have volunteered for overtime but are not needed (column 3) is, for the most part, a function of the excess supply and demand in the data sets. In the 200-nurse instances (nos. 21–24), for example, there are a relatively large number of not needed nurses compared with the smaller instances.

5.2. Computational statistics for test problems

Table 4 presents the computational results for the B&P algorithm, including run times, several measures of the integral-

ity gap, the extent of the search trees, and related statistics. Column 2 gives the MP size in terms of number of rows and columns at the final iteration. Because no columns are ever deleted, this is the largest LP solved. However, the number of eligible columns at each node depends on the branching constraints imposed at that point in the tree. The number of rows is a reflection of Constraints (2b)–(2d) and does not change from one iteration to the next. The branching constraints associated with the slack variables are not included in this statistic.

The cumulative clock time (in seconds) is given in column 3 and accounts for all the steps in the algorithm from preprocessing to output generation. Although larger problems, as measured by the number of nurses, tend to take longer to solve, the 150-nurse instances appear to be the most difficult. Column 4 reports the total number of nodes generated and fathomed by the B&P algorithm. In the implementation, a node is explored only if the difference between the LP solution and incumbent is within the optimality gap given in the last column and discussed below.

The average time per node in seconds and the average number of new columns added per node excluding the root node are reported in columns 5 and 6, respectively. As can be seen, there is a wide variation in the number of nodes per problem, both within each group of five instances and

Table 4. Computational results

<i>Problem no.</i>	<i>Problem size (rows × columns)</i>	<i>Time (seconds)</i>	<i>Nodes</i>	<i>Avg time per node (seconds)</i>	<i>Avg no. columns generated</i>	<i>Best solution</i>	<i>Best solution node</i>	<i>Initial LP solution</i>	<i>Initial gap (%)</i>	<i>Optimality gap (%)</i>
1	80 × 1673	128	48	2.7	4.80	4372.9	20	4313.5	1.4	0.1
2	80 × 796	77	50	1.4	2.86	4760.2	0	4695.6	1.4	0.1
3	80 × 1090	199	141	1.4	1.22	6559.2	110	6441.7	11.8	0.1
4	80 × 1363	355	298	1.2	0.52	14 335.3	0	13 918.5	3.0	0.1
5	80 × 1373	231	138	1.7	1.39	5557.7	130	5090.4	12.8	0.1
6	110 × 2066	2201	1000	2.2	0.12	33 150.5	15	32 609.2	1.8	1.1
7	110 × 2474	86	22	3.9	1.05	29 253.2	20	28 994.2	2.0	1.0
8	110 × 3992	257	60	4.3	0.72	15 042.2	30	14 923.0	5.2	1.0
9	110 × 3933	94	21	4.5	1.24	18 156.7	10	18 025.4	4.2	1.0
10	110 × 1786	102	39	2.4	0.95	33 390.8	30	33 107.0	1.4	1.0
11	165 × 4955	382	91	4.2	0.27	53 296.7	90	52 891.3	1.4	1.0
12	165 × 5067	354	63	5.6	2.63	24 628.0	50	24 322.8	1.7	1.0
13	165 × 5133	481	80	6.0	0.66	18 570.8	75	18 386.5	4.9	1.0
14	165 × 4146	379	70	5.4	0.69	27 733.4	50	27 316.2	1.7	1.0
15	165 × 4195	341	52	6.6	2.10	31 991.5	50	31 506.2	4.8	1.0
16	235 × 4749	2432	657	4.4	0.09	29 690.8	315	29 156.1	2.5	1.0
17	235 × 3884	131	21	6.2	1.71	29 694.8	20	29 453.7	1.7	1.0
18	235 × 4625	13	1	13.0	0	20 418.6	0	20 308.2	0.5	1.0
19	235 × 3885	65	21	3.1	0.47	32 350.8	10	31 121.2	1.6	1.0
20	235 × 4744	690	129	5.4	0.42	27 714.8	35	27 218.3	1.9	1.0
21	285 × 5255	807	165	6.2	0.19	27 638.0	150	27 129.2	2.2	0.5
22	285 × 5974	120	17	7.1	0.53	22 623.2	10	22 554.2	1.0	0.5
23	285 × 3968	123	21	5.9	1.86	21 448.6	10	21 350.8	5.9	0.5
24	285 × 5244	129	22	5.9	0.50	35 330.8	20	32 451.6	1.3	0.5
25	285 × 5946	336	36	9.3	1.69	39 061.2	29	38 820.9	1.6	0.5



between groups. No trend is discernable. The same can be said for the average number of columns per node, although this statistic provides some insight on the effectiveness of the initial columns. The poorer the quality of the initial columns, the greater the number of new columns that are generated at each node. This translates into an increase in the computational time per node.

Empirically, the number of columns added per node decreased as the search tree grew. Because columns are never deleted from the MP, at some point, few if any implicit columns price out favorably. Moreover, the number of fractional variables remained relatively constant even though the branching restrictions changed. In light of these observations, we tried an alternative pricing strategy in which the search for new columns was halted as soon as the first was found. Without exception, this approach led to a slight increase in overall run times and so was abandoned.

The last five columns in Table 4 provide insight into the efficiency of the algorithm and the tightness of the LP solution at the root node. The objective function value of the best feasible solution found is reported in column 7. This is an aggregate monetary value that includes the cost of nurses plus a penalty for uncovered demand associated with the gap variables, g_{jp} . Column 8 identifies the node at which the best solution was found. Although there does not appear to be any discernable pattern within a group of problems, or with respect to the overall size of the search tree, for a large number of cases, the best node was not encountered until late in the enumerative process. A closer look at the individual trees revealed that many were not balanced. Because of the depth-first nature of the search, it was often necessary to go deep into the tree to find, what turned out to be, the best feasible solution. Once encountered, fathoming was quick.

The tightness of the LP solution at the root node is reported in column 9 and the initial gap is given in column 10. The latter is the percent difference between the integer solution obtained with the set covering heuristic and the LP solution at the root node; that is, $(z_{IP} - z_{LP}) / z_{LP} \times 100\%$. The initial gap indicates the effectiveness of the heuristic at the root node in converting the fractional solution obtained with D-W decomposition to a feasible integer solution.

The last column in Table 4 lists the optimality gap used for termination. This value is the worst possible percent-

age difference between the (unknown) optimal integer solution and the smallest lower bound associated with the problem. It also serves as a tolerance to determine when to fathom a node with a fractional solution. More precisely, when $(1 + \text{optimality gap}) \times z_{LP} < \text{best integer solution}$, the node is fathomed. Based on the realized difficulty in solving problems of various sizes, we used a 0.1% tolerance for the 20-nurse instances, a 1% tolerance for the 50-, 80-, and 150-nurse instances, and a 0.5% tolerance for the 200-nurse instances. With the exception of problem no. 25, the best solution was always found by the heuristic rather than by integrality of the MP.

5.3. Summary by group

The overall performance of the B&P algorithm can be measured by its running time for a prespecified optimality gap. For the 25 test problems, the computation times varied from 77 seconds to 2432 seconds, or a bit more than 1 minute to 40 minutes. Most large problems were solved within 16 minutes, whereas the smaller ones with 20 and 50 nurses were solved in less than 5 minutes. Table 5 summarizes the results by group. As expected, the computational times grow with the number of nurses (the exception being the 200-nurse instances), as do the total columns generated and the average time to solve each node. In contrast, the average number of columns generated per node does not exhibit any consistent behavior.

Two factors contribute to the size of the search tree: (i) the given optimality gap; and (ii) the quality of the heuristic solution. Somewhat surprisingly, the number of nurses is not correlated with the total number of nodes. Problem nos. 1–5 with 20 nurses have a higher average node count than problem nos. 11–15 with 80 nurses and 21–25 with 200 nurses.

Although there is no obvious relationship between the size of the tree and the characteristics of a problem, the quality of the solution provided by the set covering heuristic plays an important role in fathoming. If good feasible solutions are found early on, then many nodes are likely to be fathomed high in the tree, thus limiting its size. The performance of the heuristic is closely related to the number of fractional variables in the master LP solution. As the number of columns in the MP grows, so does the possibility of fractional solutions and the effort required to achieve convergence.

Table 5. Summary statistics for computational results

Total no. of nurses	Avg no. of columns	Avg time (seconds)	Nodes	Avg time per node (seconds)	Avg no. of columns per node	Best solution node	Initial gap (%)
20	1259.0	198.0	135.0	1.68	2.158	52.0	6.08
50	2850.2	307.8	468.6	3.46	0.816	21.0	2.92
80	4699.2	387.4	71.2	5.56	1.270	63.0	2.90
150	4377.4	666.2	165.8	6.42	0.538	76.0	1.64
200	5277.4	303.0	52.2	6.88	0.954	43.8	2.40

For those problems in which the best node was identified retrospectively to be close to the root node and far from the terminal node, the heuristic solutions were seen to be of poor quality, that is, far from the LP solutions. This led to relatively deep trees in about 25% of the instances. In problem no. 2, for example, the optimal solution was found at the root node, but it was necessary to explore 50 more nodes before the search could be terminated.

The average time to solve a node is determined primarily by the size of the MP, which, in turn, grows with the complexity of the pricing problems. The latter is a function of the nurse types given by the X1 entry in column 3 of Table 1 and the number of units to which they can float. The more nurses that are available to float and work overtime (X1), the greater the number of columns that can be generated in the pricing problems. For example, problem no 22 has 50 nurses who can float and work overtime and who can be assigned to an average of 4.11 units. The total number of columns generated was 5974. In contrast, problem no. 23 has 20 nurses of the same type but with slightly more units to which they can be assigned (4.29). The total number of columns generated in this instance was only 3968.

5.4. Sensitivity analysis

During our experiments using different settings for the aggregate penalty parameter, P^{\max} , we found that there was little correlation between this value and algorithmic performance. Nevertheless, there is a strong negative relationship between P^{\max} and cost because increasing P^{\max} allows columns with higher penalties for regular and pool nurses to be included in a solution. These columns have a much lower cost than those associated with agency

nurses, on-call nurses, and uncovered demand, which would otherwise have to be used to obtain feasibility. The net effect is an overall reduction in the objective function value.

To illustrate this relationship, we solved a 70-nurse problem (50-0-0-20) for 12 values of P^{\max} ranging from 200 to 576. In all cases, the B&P algorithm was terminated only after all nodes were fathomed using a 0.1% optimality gap as the stopping criterion. The total cost results are plotted in Fig. 2, which shows that as P^{\max} increases, the objective function first decreases rapidly and then levels off after 320. This exponential-type behavior was typical of all problems investigated. Note that beyond 500, Constraint (2d) is non-binding in the optimal solution.

The results for the run time as a function of the aggregate penalty are plotted in Fig. 3. Very little can be said about the relationship between these two measures except that there appears to be a slight upward trend for values of P^{\max} between 250 and 470. For values below 250, the computation times are relatively high, and for values above 470, solutions are obtained within a matter of seconds at the root node.

The latter observation suggests that the major source of difficulty in solving the daily adjustment problem is the tightness of the penalty constraint. As the data in Fig. 3 illustrate, however, some problems are more difficult than others even though their penalty values are close. A more detailed analysis revealed that this observation can be partially explained by the amount of slack in the penalty constraint, Constraint (2d), when evaluated at the solution obtained with the IP set covering heuristic. For all but the largest values of P^{\max} , the LP solution to the MP is always tight with respect to Constraint (2d). However, we were only able to identify a weak correlation between this

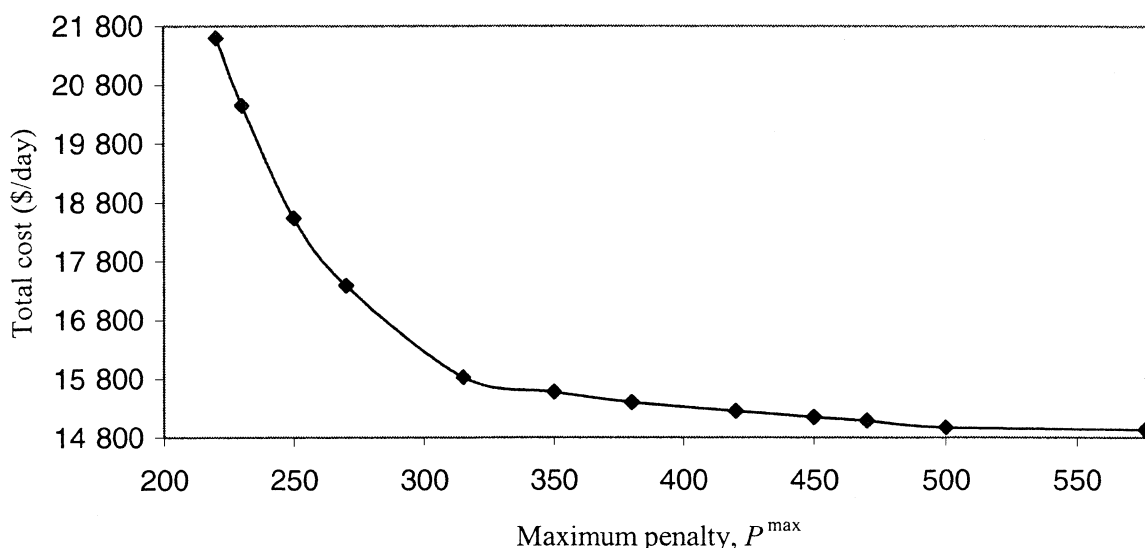


Fig. 2. Relationship between the objective function and the aggregate penalty.

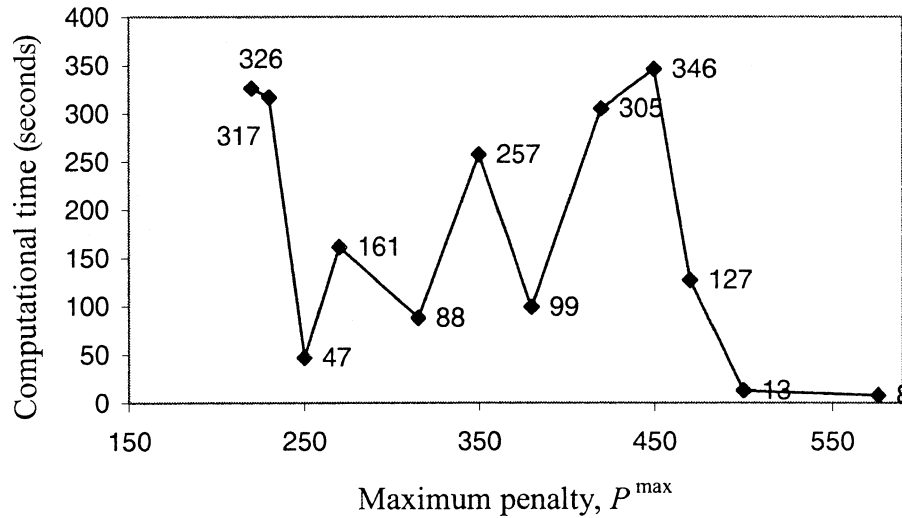


Fig. 3. Relationship between the algorithmic efficiency and the aggregate penalty.

slack and the optimality gap in the early stages of the computations.

The relevant data are presented in Table 6 for the 70-nurse problem for the first 11 values of P^{\max} . In each case, the computations were halted after five nodes were reached or when the optimality gap was zero. The second column in the table reports the smallest lower bound of all unfathomed nodes, the third column gives the better of the two integer solutions found by the heuristic at the root node and node 5, the fourth column gives the optimality gap between the lower bound and the integer solution, and the fifth column reports the slack in Constraint (2d). The last column gives the solution times to reach convergence (same as in Fig. 3). The results suggest that there exists a weak relationship between the slack values and gaps. Empirically speaking using a cutoff point of 100 seconds, if the

slack is >2 , then the problem is difficult; for values <2 , it is easy; when the slack = 2, there does not seem to be any correlation.

5.5. MIR cuts

Adding valid inequalities to the MP offers the possibility of obtaining improved lower bounds, but at the expense of greater run times. Unfortunately, our experience with a subset of the instances in Table 1 showed that when MIR cuts were added, the LP solution increased only marginally or not at all after reoptimization. Results for problem nos. 5, 10, 12, 16, 21, and 25 are highlighted in Table 7. The second column identifies the LP solution at the root node before MIR cuts were added; the third column gives the LP solution after the cuts were added and the MP reoptimized with the D-W procedure (the last column indicates that no new columns were ever generated). The fourth column gives the percentage improvement. When no cuts were found, as indicated in the sixth column, the third and fourth columns were left blank. From the sixth column, we see that between zero and 30 cuts were added to the MP with a maximum improvement of 0.99%.

The fifth column in Table 7 reports the integer solution found by the heuristic at the root node after the MIR cuts were added and the D-W procedure reapplied. These values are nearly identical to those in Table 4 for the same problems, so once again, no improvement. The amount of time required to reoptimize the restricted MP and the number of new columns generated are reported in the last two columns of the table. Although these results only reflect the performance of the cut generation procedure at the root node of the B&B tree, they are typical of our experience when a problem is solved to optimality.

Table 6. Optimality gap for different penalty values for the 70-nurse problem

Maximum penalty, P^{\max}	LP lower bound	Best integer solution	Optimality gap (%)	Slack in Constraint (2d)	Overall time (seconds)
220	21 327.4	22 111.5	3.67	2	317
230	20 223.1	20 453.4	1.14	2	326
250	18 458.8	18 538.8	0.43	2	47
270	17 298.0	17 618.8	1.85	1	161
315	15 821.0	15 825.0	0.02	0	88
350	15 498.9	15 762.7	1.70	2	257
380	15 363.4	15 895.8	3.46	2	99
420	15 222.3	16 519.5	8.58	3	305
450	15 121.2	15 652.2	3.50	3	346
470	15 060.6	15 954.6	5.90	5	127
500	14 974.0	14 976.0	0	0	13

Table 7. Results for experiments using MIR cuts

<i>Problem no.</i>	<i>Best LP before MIR cuts added</i>	<i>LP after MIR cuts added</i>	<i>Initial improvement (%)</i>	<i>Integer solution at root node</i>	<i>Total cuts added</i>	<i>Extra time (seconds)</i>	<i>Additional columns generated</i>
5	5090	—	—	5741	0	10	0
10	33 107	33 176	0.20	33 581	30	15	0
11	52 891	—	—	53 645	0	25	0
16	29 156	29 446	0.99	29 882	1	40	0
21	27 129	27 314	0.68	27 739	30	50	0
25	38 820	—	—	39 448	0	35	0

6. Summary and conclusions

The computational results show that the daily adjustment problem can be solved efficiently for up to 200 nurses using the B&P algorithm. The success of the algorithm can be largely attributed to the branching scheme developed to construct the search tree and to the set covering heuristic that provided high-quality feasible solutions. With respect to branching, two rules were applied. The first was designed to exploit the presence of the SOS-type constraints in the subproblems. The second was aimed at reducing the effects of symmetry and focused on the slack or outside nurse variables in the demand constraint, Constraint (2b).

As part of the research, two heuristics were developed to find feasible solutions. The first, tabu search, works by directly manipulating the incumbent to arrive at a local optimum. Although extensive testing was done with various neighborhood definitions, list sizes, and diversification strategies, we were never able to achieve more than a 1 or 2% improvement. The second, a set-covering-type approach, involves solving an IP whose columns correspond to “good” schedules. The IP heuristic proved to be much more effective than tabu search and was incorporated in the B&P algorithm.

In an effort to strengthen the lower bound provided by the solution of the MP at each node in the B&B tree, general M/R cuts were added to the model. Initial testing showed that they were not effective, however, primarily because of the need to reoptimize the MP using the D-W procedure after being added. The amount of time spent checking for new columns was out of proportion to the improvement in the bound, which, in most cases, was negligible. Our limited experience with Gomory cuts showed the same to be true.

In reality, specifying a value for P^{\max} is likely to be problematic. As an alternative, it might be better and even more appropriate to consider an aggregate penalty for the on-call nurses only and to define a separate penalty constraint that can be individually tailored for each regular and pool nurse. In the MP, Constraint (2d) would be much simpler because it would only contain the term associated with the on-call nurses. However, each subproblem would now include a penalty constraint equivalent to Constraint (1i) but without the on-call term on the left-hand side.

A practical advantage of this formulation is that it is likely to produce much faster run times. When Constraint (2d) is not binding at optimality, the B&P algorithm almost always converges at the root node. Moreover, the sensitivity analysis results showed that when the slack in Constraint (2d) induced by a feasible solution is small, convergence is likely to be quick. By removing the first term in Constraint (2d) and modifying P^{\max} accordingly, the chances of realizing either of these situations is greatly increased, especially when either P^{\max}/p^4 is integral or the optimal solution does not contain on-call nurses.

Acknowledgement

This work was supported in part by the National Science Foundation under grant DMI-0218701.

References

- Aiken, L.H., Clarke, S.P., Sloane, D.M., Sochalski, J. and Silber, J.H. (2002) Hospital nurse staffing and patient mortality, nurse burnout, and job dissatisfaction. *Journal of the American Medical Association*, **288**(16), 1987–1993.
- Bard, J.F. and Purnomo, H.W. (2005). Preference scheduling for nurses using column generation. *European Journal of Operational Research*, (to appear).
- Bard, J.F., Yu, G. and Argüello, M.F. (2001) Optimizing aircraft routings in response to groundings and delays. *IIE Transactions on Operations Engineering*, **33**(10), 931–947.
- Barnhart, C., Hane, C.A. and Vance, P.H. (2000) Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, **48**(2), 318–326.
- Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P. and Vance, P.H. (1998) Branch-and-price: column generation for solving huge integer programs. *Operations Research*, **46**(4), 316–329.
- Cheang, B., Li, H., Lim, A. and Rodrigues, B. (2003) Nurse rostering problems—a bibliographic survey. *European Journal of Operational Research*, **151**(1), 447–460.
- Clausen, J., Hansen, J., Larsen, J. and Larsen, A. (2001) Disruption management. *OR/MS Today*, **28**(5), 40–43.
- Kimball, B. and O’Neil, E. (2002) *The American Nursing Shortage*, The Robert Wood Johnson Foundation, Princeton, NJ.
- Purnomo, H.W. (2005) Solving the mid-term and short-term nurse scheduling problems. PhD dissertation, Graduate Programming in Operations Research & Industrial Engineering, The University of Texas, Austin, TX 78712, USA.

- Savelsbergh, M.W.P. (1997) A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, **45**(6), 831–841.
- Siferd, S.P. and Benton, W.C. (1994) A decision modes for shift scheduling of nurses. *European Journal of Operational Research*, **74**(4), 519–527.
- Spratley, E., Johnson, A., Sochalski, J., Fritz, M. and Spencer, W. (2000) The registered nurse population, findings from the national sample survey of registered nurses, US Department of Health and Human Services, Washington, DC. Available at <http://www.ruralnursing.org/rnsurvey00-1.pdf>
- Vanderbeck, F. (2000) On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, **48**(1), 111–128.
- Vanderbeck, F. and Wolsey, L.A. (1996) An exact algorithm for IP column generation. *Operations Research Letters*, **19**, 151–159.
- Wolsey, L.A. (1998) *Integer Programming*, Wiley, New York, NY.

Biographies

Jonathan F. Bard is a Professor of Operations Research & Industrial Engineering in the Mechanical Engineering Department at the University of

Texas at Austin. He holds the Industrial Properties Corporation Endowed Faculty Fellowship, and serves as the Associate Director of the Center for the Management of Operations and Logistics and as the Area Coordinator for the OR&IE Graduate Program. He received a D.Sc. in Operations Research from The George Washington University. He has previously taught at the University of California—Berkeley and Northeastern University. His research interests are in airline operations, the design and analysis of manufacturing systems, postal operations, and vehicle routing. Prior to beginning his academic career, he worked as a program manager for the Aerospace Corporation and as a systems engineer for Booz, Allen & Hamilton. He currently serves on the editorial boards of five journals and is a fellow of IIE.

Hadi W. Purnomo is a Ph.D student in the Operations Research & Industrial Engineering Program at the University of Texas at Austin. He received an M.Sc. in Operations Research from the same university and a B.Sc. in Industrial Engineering from the Tenth November University in Indonesia. His research interests are in mathematical programming and optimization, especially solving large-scale problems that arise in manufacturing, logistics and transportation, and service operations. He has participated in several research projects involving personnel scheduling for healthcare facilities.

Contributed by the Applied Optimization Department