40. Nurse scheduling: a case of disaggregation in the public sector

HOLMES E. MILLER and WILLIAM P. PIERSKALLA

Abstract

This paper deals with the problem of generating day on-day off patterns for nurses during a two to six week period. The formulation implies selecting a configuration of nurse schedules that minimizes an objective function balancing the trade-off between staffing coverage and schedule preferences of individual nurses, subject to certain feasibility constraints on the nurse schedules. The problem is solved by a cyclic coordinate descent algorithm. Results are presented pertaining to a six month application in a hospital setting.

40.1. Introduction

Because of demands for service seven days a week and around the clock, generating work schedules for nurses in a manner satisfactory to both employer and employee is a difficult task. From the hospital's point of view, the schedules should contain staffing levels satisfying requirements for various nursing classes on the days and shifts in question. The nurses, on the other hand, would like to receive schedules that assure as favorable day on-day off and shift rotation patterns as possible. Unfortunately, neither's desires can be totally satisfied. The hospital must work with a configuration of nurses greater than a hypothetical minimum and the nurses must be willing to accept schedules somewhat less than the 'ideal'.

In this paper, we shall present a mathematical programming based procedure that has generated favorable nurse schedules and has been implemented at a number of hospitals in the United States and Canada. The model is one stage in a disaggregated solution process encompassing three different levels where scheduling occurs.

The highest level involves allocation of nurses to departments over the long term. The output from this phase is often an employee roster. For example, it may specify that three RN's, five LPN's and four Nursing Aides are assigned to a nursing station for the first half of the year.

The employee roster is then used as the input for the second level determining patterns of days on and days off for employees. This phase will be discussed at length in this paper where the final schedules are the solution to a model which trades off employer and employee preferences. The output of this phase, in the case of nursing, is a work schedule for the coming two to six weeks, specifying working and recreation days and, when applicable, shift rotation.

By their nature, day on-day off schedules are made in advance. The requirements on which the schedules are based are forecasts. Moreover it is implicitly assumed all employees scheduled to work on a particular day will actually be working. Reality often does not conform to these assumptions. For example, when the nurse schedules are generated minimum requirements may have specified three LPN's but when the day in question actually arrives the requirements may have risen to four. The number of RN's scheduled to work on a day may be three but one may be sick and only two actually report for work.

To adjust to these realities, the lowest level of the disaggregated solution procedure involves short term personnel allocation. Allocation policies differ in different situations. One common method of short term allocation in nursing involves utilizing a pool of 'float' nurses. They are assigned to various departments to adjust to the changed supply and/or demand conditions.

In Section 2, we shall present the model and solution procedure to a nurse scheduling problem where individual preferences are considered and where the output is a set of personalized nurse schedules. It is an example of intermediate level disaggregation.

Section 3 will discuss how one may use results from lower level disaggregation to examine higher level problems.

40.2. Generating a roster of days on and off

40.2.1. The model

The mathematical programming model to be presented schedules days on and days off for nurses assigned to a given working unit for a two, four, six, or eight week scheduling horizon subject to certain hospital policy and nurse preference constraints. Because of the large number of constraints, no feasible solutions to the scheduling problem would exist if all constraints were binding. We thus divide these constraints into two classes: Feasibility set constraints, which define the sets of feasible nurse schedules, and non-binding nurse and hospital constraints, whose violation incurs a penalty cost which appears in the objective function of the math programming problem. The definition regarding which constraints constitute these classes may change depending upon where the model is being applied.

40.2.2. Constraints: the feasibility set

Because of the possibility of special requests by nurses, no constraints are binding in the sense that they hold under all circumstances except those constraints emanating from the special requests. We do, however, distinguish between constraints we would *like to hold* in the absence of special requests, and those which we shall always allow to be violated while incurring a penalty cost.

The former constraints define what we call the *feasibility set* Π_i, i.e., Π_i = the set of feasible schedule patterns for nurse *i*.

In the *absence* of special requests, this set might include all schedules satisfying:

- A nurse works ten days every pay period (i.e., 14 day scheduling period)
- No work stretches (i.e., stretches of consecutive days on) are allowed in excess of σ days (e.g., $\sigma = 7$).* No work stretches of τ or fewer days are allowed (e.g., $\tau = 1$).* Hence one schedule in an Π_i satisfying these might be (with $\sigma = 7$, $\tau = 1$, $1 =$ day on, $0 =$

* These are calculated within a scheduling period and also at the interface of a scheduling period and past and future scheduling periods.

day off)

1 1 1 1 1 1 1 0 0 1 1 1 0 0

Now suppose a nurse has special requests. For example, suppose the nurse requests the schedule:

1 1 1 1 1 1 1 1 0 1 0 0 0 B

where the B indicates a birthday off. In this case *all* of the above constraints would be violated and Π_i would consist of only the schedule just given. Thus in the general case, Π_i is the set of schedules which:

1. Satisfies a nurse's special requests.
2. Satisfies as many of the constraints we *would like* to see binding as possible, given the nurse's special requests.

The constraints we would like to hold are a function of the situation in which the model is applied. For example, we could easily specify five out of seven days working as ten out of fourteen or specify additional constraints we would like to see satisfied such as no split days off, i.e., schedules containing 101 patterns.

40.2.3. Constraints: non-binding

Each schedule pattern $x^i \epsilon \Pi_i$ may violate a number of *non-binding schedule pattern constraints* while incurring a penalty cost.

Define

N_i = The index set of the non-binding schedule pattern constraints for nurse i.

For example, if the site in which the model was being implemented deemed them as non-binding, the following constraints might define N_i:

- No work stretches longer than S_i days (where $S_i \leqq \sigma$);*
- No work stretches shorter than T_i days (where $T_i \geqq \tau$);*
- No day on, day off, day on patterns (1 0 1 pattern);*
- No more than κ consecutive 1 0 1 patterns,*
- Q_i weekends off every scheduling period (e.g., 4 or 6 weeks);
- No more than W_i week ends working each scheduling period;

- No patterns containing four consecutive days off;
- No patterns containing split weekends on (i.e., a Saturday on – Sunday off – pattern, or vice versa).

In addition to non-binding schedule pattern constraints, we also have *non-binding staffing level constraints*. Define:

d_k = The desired staffing level for day k; and m_k = the minimum staffing level for day k. Then we have: a) The number of nurses scheduled to work on day k is greater than or equal to m_k and b) The number of nurses scheduled to work on day k is equal to d_k.

40.2.4. Objective function

As was mentioned, the objective function is composed of the sum of two classes of penalty costs; penalty costs due to violation on non-binding schedule pattern constraints and staffing level constraints.

40.2.4.1. Staffing level costs: Define the *group* to be scheduled as the set of all nurses in the unit who are to be scheduled by one application of the solution algorithm. Further define a *subgroup* as a subset of the group. For example, the group to be scheduled may be all those nurses assigned to a nursing unit and the subgroups may be registered nurses, licensed practical nurses, and nursing aides. Alternatively, the group may be defined as all registered nurses and a subgroup might be those capable of performing as head nurses.

Then, for each day $k = 1, \ldots, 14$ (where there are I nurses), the group staffing level costs are given by:

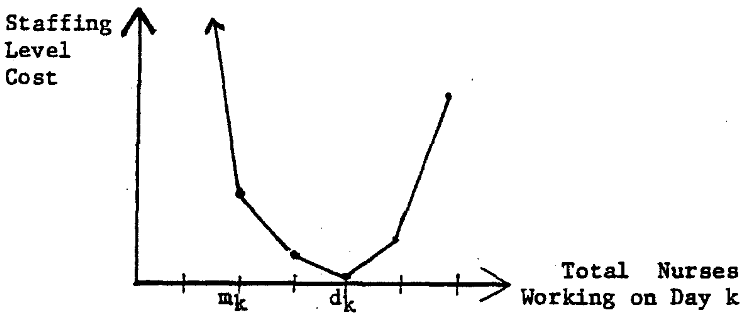
$$f_k(\sum_{i=1}^{I} x_k^i)$$

where $x^i = (x_1^i, \ldots, x_{14}^i)$. For example, this function might appear as seen in the following page.

Now define:

B_j = The index set of nurse subgroups j, where
J = The index set of all subgroups.

If m_k^j and d_k^j are the minimum and desired number of nurses re-



quired on day k for subgroup j, we define the staffing cost for violating those constraints on day k for subgroup j as: $h_{jk}\ (\Sigma\, x_k^i)$ where $h_k(\cdot)$ is defined similarly to $f_k(\cdot)$ $i \in B_j$

Then the total staffing level costs for all 14 days of the pay period are:

$$\sum_{k=1}^{14} f_k\left(\sum_{i=1}^{I} x_k^i\right) + \sum_{k=1}^{14} \sum_{j \in J} h_{jk}\left(\sum_{j \in B_j} x_k^i\right)$$

40.2.4.2. Schedule pattern costs: For each nurse $i = 1, \ldots, I$, the schedule pattern costs for a particular pattern x^i measure:

1. The costs inherent in that pattern in relation to which constraints in N_i are violated.
2. How nurse i perceives these costs in light of that nurse's schedule preferences.
3. How this cost is weighed in light of the nurse's schedule history.

For example, for (1) the pattern,

1 1 1 1 1 0 0 1 1 1 0 0 1 1

may incur a cost for nurse i whose minimum desired work stretch is 4 days. This is a cost inherent in the pattern. Considering (2), we next ask how nurse i perceives violations of the minimum desired stretch constraints, i.e., how severely are violations of this non-binding constraint viewed vis-a-vis others in N_i. Finally (3), gives us some indication of how we should weigh this revised schedule pattern cost in light of the schedule employee i has received in the past. Intuitively, if nurse i has been receiving bad schedules, we would want the cost to he higher to cause a good schedule to be accepted when the solution algorithm is applied and vice versa.

Thus, we define:

$g_{in}(x^i)$ = the cost of violating non-binding constraint $n \in N_i$ of schedule x^i.

a_{in} = the 'weight' nurse i gives a violation of non-binding constraint $n \in N_i$, which we shall call the *aversion coefficient.*

A_i = the *aversion index* of nurse i; i.e., a measure of how good or bad nurse i's schedules have been historically vis-a-vis nurse i's preferences.

Then the total schedule pattern cost to nurse i for a schedule pattern x^i is: $A_i \ \Sigma_{n \in N} \ \alpha_{in} g_{in}(x^i)$, and the sum of these costs for all nurses $i = 1, \ldots, I$ is the total schedule pattern cost.

40.2.5. Problem formulation

The nurse scheduling problem may now be formulated as: (where $0 < \lambda < 1$ weighs staffing level and schedule pattern costs). Find $x^1, x^2, \ldots, x^I$ which minimize:

$$\lambda \left[\sum_{k=1}^{14} f_k\left(\sum_{i=1}^{I} x_k^i \right) + \sum_{k=1}^{14} \sum_{j \in J} h_{jk}\left(\sum_{i \in B_j} x_k^i \right) \right] +$$

$$(1 - \lambda) \sum_{i=1}^{I} A_i \sum_{n \in N_i} a_{in} g_{in}(x^i) \quad \text{s.t.} \quad x^i \in \Pi_i, \ i = 1, \ldots, I.$$

40.2.5.1. Description of the solution procedure: The solution procedure used is a near-optimal algorithm. It starts with an initial configuration of schedules, one for each nurse. Fixing the schedules of all nurses but one, say nurse i, it searches Π_i. The lowest present cost and best schedule configuration are updated if, when searching Π_i, a schedule is found which results in a lower schedule configuration cost than the lowest cost to date. When all the schedules in Π_i have been tested either 1) A lower cost configuration has been found or 2) No lower cost configuration has been found. The process cycles among the I nurses and terminates when no lower cost configuration has been found in I consecutive tests.

The algorithm is:

1. Determine the set of feasible schedules for each nurse's Π_i. Let $|\Pi_i|$ denote the number of schedules in Π_i.

2. Calculate the schedule pattern costs for each schedule $x^{ik} \in \Pi_i$, for $i = 1, \ldots, I$.
3. Get initial schedule configuration and let BEST = its cost (e.g., choose the lowest cost schedule from each Π_i).
4. Let $i = 1$, $K = |\Pi_i|$, $k = 1$, and CYCLE = 0.
5. Insert schedule x^{ik} in schedule mix and let TEST = the cost of this configuration.
6. If TEST < BEST go to step. 8.
7. Let $k = k + 1$. If $k = K + 1$ go to step 9. Otherwise go to step 5.
8. Let CYCLE = 0 and BEST = TEST. Insert x^{ik} in complement of 'best schedules to date'. Go to step 7.
9. If CYCLE = I stop. Otherwise let $i = i + 1$ (if $i > I$, let $i = 1$) and let $K = |\Pi_i|$, $k = 1$, and CYCLE = CYCLE + 1. Go to step 5.

If we view the feasibility region as $\Pi_1 x \ldots x\Pi_I$ the algorithm is simply a cyclic coordinate descent algorithm along the coordinate directions Π_i. Each Π_i contains all feasible schedules for employee i. When 4 days are given off every 14 day pay period, Π_i contains at most $\binom{14}{4} = 1001$ schedules. This number is reduced considerably when previous schedules, special requests, and other feasibility set constraints are considered. The convergence of the algorithm is assured since $\Pi_{1x} \ldots x\Pi_I$ contains a finite number of points, namely, $\Pi_{i=1}^{I} |\Pi_i|$.

40.2.5.2. Results: Preliminary tests were conducted for scheduling nurses on a small sample problem comparing the algorithm presented above with a branch and bound algorithm which yielded the optimal solution. These tests showed the algorithm generated schedules almost as good as the optimal ones in far less computer time. For example, in one run on a 4 nurse, 20 schedule problem the cost of the algorithm generated schedule was 12.3 while the optimal cost was 7.55 (the initial cost of the algorithm solution was 239.45). The CPU time for the algorithm on a CDC 6400 was .367 seconds vs. 10.509 for the branch and bound. Moreover, this was when the initial upper bound in the branch and bound was the final solution generated by the algorithm. Arbitrarily large upper bounds yielded running times on the order of 30 seconds.

More extensive tests were run for the day shift of a unit in a large 800 bed hospital. The hospital had collected historical data regarding nurse schedule preferences and minimum and desired staffing levels. This data was used in the application of the algorithm. Because the algorithm schedules and the hospital schedules were generated from the same base data, it enabled us to compare the algorithm schedules and hospital schedules.

40.2.6. Algorithm generated schedules

Figure 1 presents some schedules generated by the algorithm for 4 weeks of the 6 month trial period: October 22 to November 18. Note that on 14 of the 28 days the actual staffing levels were identical

Group 1

RN's	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
1A	V	V	R	1	1	1	1	R	M	1	1	1	0	0	0	1	M	1	1	1	1	0	M	1	1	1	0	0
1B	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	0	1	1	1	1	M	1	1	0	0	0
1C	1	1	1	1	0	0	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	0
1D	1	0	0	1	1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	1	1	0	1	1	1
1E	1	1	1	1	1	1	1	0	0	1	1	1	0	0	1	1	1	1	0	B	0	1	1	1	0	0	1	1
1F	1	1	1	0	1	1	1	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1	1	1
1G	V	V	1	1	1	0	0	1	1	0	0	1	1	1	1	0	1	1	1	0	0	1	1	0	1	1	1	1
Group 2																												
LPN																												
2A	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	R	R	R	1	1	1	R	1	1
2B	1	1	0	B	1	1	1	1	0	1	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	0	0	0
2C	1	1	1	C	0	0	0	1	1	0	C	1	1	1	1	1	0	C	1	1	1	1	0	1	C	1	0	0
2D	0	1	1	1	1	1	1	V	V	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	0	1	1	1
2E	0	1	1	1	1	0	0	1	1	1	0	1	1	1	0	0	1	1	1	0	0	1	1	1	1	1	1	1
Total Desired	9	9	9	9	9	5	6	9	9	9	8	8	6	5	9	9	9	9	9	5	6	9	9	9	8	8	7	6
Total Actual	8	9	9	9	9	6	6	9	9	10	9	8	6	6	9	10	10	10	9	5	5	10	10	10	9	8	7	7

Legend:

1 = Day Scheduled On	R = Requested Day Off
0 = Day Scheduled Off	B = Birthday Off
M = Day on for Meeting	C = Day on for Class
V = Vacation Day Off	

Figure 1. A two-week set of nurse schedules generated by the solution algorithm.

with the desired staffing levels. The unit is understaffed by 1 nurse on 2 days and overstaffed by 1 nurse on 12 days.

When the schedule pattern costs for the schedule in Figure 1 were examined, we found that in all cases but one the nurses were given a schedule better than 90 percent or more of those in the feasible pattern set. Moreover, over half the time the number of feasible patterns in the sets Π_i were well in excess of 200 so there were many schedules to choose from.

When the entire 6 month period was considered, it was found that in 90 percent of the days the deviation from the desired staffing level was 0 or +1. This was unadjusted for aggregate unit under or overstaffing which would necessitate some deviations. These same algorithm generated schedule configurations yielded the lowest cost schedule in a nurse's feasible schedule pattern almost 44 percent of the time and/or schedules with a cost less than or equal to 90 percent of the schedules in the nurse's feasible schedule set almost 88 percent of the time. The algorithm generated schedules compared favorably to those actually used by the hospital according to schedule patterns as well as staffing level inturn. See [1] for more extensive results.

40.2.7. Extensions

The model may be extended to include shift rotation and part-time nurses by redefining the feasibility sets Π_i in an appropriate manner. For example, if we consider shift rotation, we:

1. Schedule night and evening shifts first.
2. If the staffing level patterns require shift rotation to reduce staffing costs, and if the day shift has nurses available to be rotated, select nurses from those available to rotate and have them rotate to the night and evening shifts. The exact rotation patterns selected must conform with various rotation constraints and must result in reduction of staffing costs on the shifts rotated to.
3. Schedule the day shift treating these rotation patterns as fixed conditions.

The problem of part-time nurses is handled in a way analogous to full-time nurses. Feasibility sets Π_i are constructed for part-time nurses depending on appropriately defined constraints (e.g., a nurse must work four days out of every fourteen). Then the schedules are listed according to how they meet a set of appropriately defined non-binding constraints. Then we proceed in the same manner as with full-time nurses choosing schedules from the sets Π_i where now some of these sets contain part-time nurses schedules and some contain full-time nurses schedules.

40.3. Discussion

The nurse scheduling procedure just presented is an example of intermediate level scheduling in a three level disaggregated scheduling process. The natural flow of things indicates one solves the long range, intermediate range, and short term problems in sequential order. This pertains to the operational mode. This natural order may be reversed when one considers the planning mode, where the planning may be carried out in concert with a simulation experiment.

For example, suppose a hospital is interested in determining the effects of a long-term nurse allocation policy. Using the lower levels of the disaggregation procedure, day on-day off schedules as well as short-term allocation needs may be simulated indicating the effects of the long-term procedure.

We see the presence of a higher to lower level flow in the operational modes and lower to higher level flows in the planning mode.

In both cases a large complex problem has been broken up into smaller, solvable components. The preceding has discussed how one of those components may be defined in the case of nurse scheduling. Because of its generality, the model may also be applied to other situations where tradeoffs between employer and employee preferences must be made.

References

1. Miller, H. E., Pierskalla, W. P., and Rath, G. J., 'Nurse Scheduling Using Mathematical Programming', *Operations Research*, vol. 24, no. 5 (September–October 1976), pp. 857–870.