OPERATIONS RESEARCH, Vol. 24, No. 5, September-October 1976

Nurse Scheduling Using Mathematical Programming

HOLMES E. MILLER

Rensselaer Polytechnic Institute, Troy, New York

WILLIAM P. PIERSKALLA and GUSTAVE J. RATH

Northwestern University, Evanston, Illinois

(Received original September 5, 1974; final, June 10, 1975)

This paper formulates the nurse-scheduling problem as one of selecting a configuration of nurse schedules that minimize an objective function that balances 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. We present results pertaining to a six-month application to a particular hospital unit and draw comparisons between the algorithm and hospital-generated schedules.

THE NURSE-SCHEDULING process may be viewed as one of generating a configuration of nurse schedules that specify the number and identities of the nurses working each day of the scheduling period. By specifying nurse identities, one creates a pattern of scheduled days off and on for the individual nurses. These patterns, along with the hospital staffing requirements, define the nurse scheduling problem: how to generate a configuration of nurse schedules fulfilling the hospital staffing requirements while simultaneously satisfying the individual nurse's preferences for various schedule pattern characteristics.

This paper formulates the problem and solves it in a mathematical programming format, where we would like to find a configuration of feasible schedules that minimize an objective function trading off staffing costs incurred by the hospital and nurse dissatisfaction costs incurred by the individual nurses.

A number of mathematical programming applications to nurse staffing has appeared in the literature, beginning with Wolfe and Young,[13,14] who constructed mathematical models that minimized the cost of assigning nurses of various classes to do various tasks. Liebman[4,5] also proceeded from a task orientation by assigning nursing tasks in a manner that maximized the effectiveness of nurses performing tasks on various patients. Warner and Prawda[12] sought to minimize a 'shortage cost' of nursing care services for a period of three to four days subject to total personnel capacity, integral assignment, and other relevant constraints. Abernathy, Baloff, Hershey, and Wandel[1,2] considered three different decision levels impinging upon the nurse staffing problem and formulated an interactive model where the outputs of one level (e.g., staffing policies) are the inputs of another.

Much of the work relating to nurse scheduling has concerned cyclical scheduling (see Morrish and O'Conner,[7] Price,[8] Howell,[3] and Maier-Rothe and Wolfe[6]), in which each nurse works a cycle of n weeks, where n is the length of the scheduling period. Cyclical schedules are easily generated but are characterized by excessive rigidity vis-à-vis variations in the supply of and demand for nursing services. Two noncyclical scheduling papers of note have been by Rothstein[9] and Warner.[11] Rothstein's application was to hospital housekeeping operations. He sought to maximize the number of day-off pairs (e.g., Monday–Tuesday) subject to constraints requiring two days off each week and integral assignments. Warner presented a two-phase algorithm to solve the nurse-scheduling problem. Phase I is involved with finding a feasible solution to various staffing constraints, and Phase II seeks to improve the Phase I solution by maximizing individual preferences for various schedule patterns while maintaining the Phase I solution.

1. THE MATHEMATICAL PROGRAMMING MODEL

The mathematical programming model developed here schedules days on and days off for all nurses on a given unit or ward for a given shift for a two-, four-, six-, or eight-week scheduling horizon, subject to certain hospital policy and employee constraints. Because of the large number of constraints, it is possible that no feasible solutions to the nurse-scheduling problem would exist if all the constraints were binding. For this reason we divide the constraints into two classes: feasibility set constraints, which define the sets of feasible nurse schedules, and nonbinding constraints, whose violation incurs a penalty cost that appears in the objective function. Each hospital has the discretion to define which constraints go into each class.

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 that 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., $\tau=1$). (The lower and upper bounds on work stretches are calculated within a scheduling period and also at the interface of a scheduling period with past and future periods.)

Hence one schedule in a π_i satisfying these might be (with $\sigma=7$, $\tau=1$) 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. 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 that

(1) Satisfy a nurse's special requests;

(2) Satisfy as many as possible of the constraints we would like to see binding, given the nurse's special requests.

The constraints we would like to satisfy are a function of the hospital in which the model is applied. Thus, for example, we could as easily specify five out of seven days as ten out of fourteen or specify additional constraints we would like to see satisfied, such as one weekend off each pay period.

Constraints: Nonbinding

Each schedule pattern $x^i \epsilon \pi_i$ may violate a number of *nonbinding schedule pattern constraints* while incurring a penalty cost.

Define N_i = the index set of the nonbinding schedule pattern constraints for nurse i. For example, if the hospital in which the model was being implemented deemed them as nonbinding, 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 (4 or 6 weeks);
- No more than W_i consecutive weekends 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).

(These constraints are evaluated within a scheduling period and also at the interface of a scheduling period with past and future scheduling periods.)

In addition to nonbinding schedule pattern constraints, we also have *nonbinding staffing level constraints.* Define d_k as the desired staffing level for day k and m_k as 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.

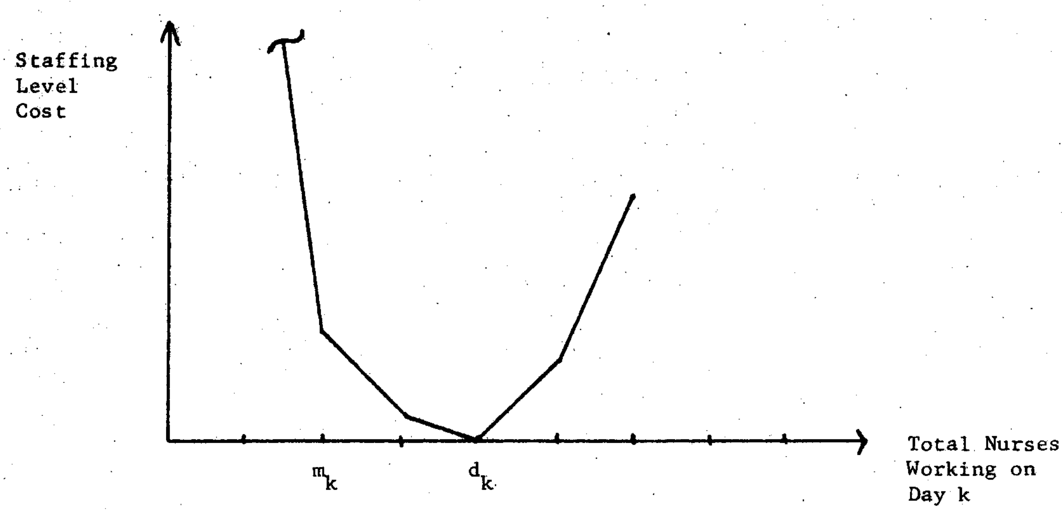

Fig. 1. An example of a daily staffing level cost function for a nursing group.

Objective Function

As was mentioned, the objective function is composed of the sum of two classes of penalty costs: penalty costs caused by violation of nonbinding staffing level constraints and penalty costs caused by violation of nonbinding schedule pattern constraints.

Staffing Level Costs

Define the *group* to be scheduled as the set of all the 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 specified by the hospital. For example, the group to be scheduled may be all those nurses assigned to a nursing unit and the subgroups may be RN's, LPN's, and Nursing Aides. Alternatively, the group may be defined as all RN's and a subgroup might be those capable of performing as head nurses.

Then, for each day $k = 1, \cdots, 14$ (where there are I nurses), the group staffing level costs are given by: $f_k(\sum_{i=1}^{i=I} x_k^i)$, where $x^i=(x_1^i, \cdots, x_{14}^i)$. For example, this function might appear as:

Now define B_j = the index set of nursing subgroups j; J = the index set of all subgroups. If m_k^j and d_k^j are the minimum and desired number of nurses required on day k for subgroup j, we define the staffing cost for violating those constraints on day k for subgroup j as $h_{jk}(\sum_{i\epsilon B_j} x_k^i)$, where $h_{jk}(\cdot)$ is defined similarly to $f_k(\cdot)$.

Then the total staffing level costs for all 14 days of the pay period are $\sum_{k=1}^{k=14} f_k(\sum_{i=1}^{i=I} x_k^i)+\sum_{k=1}^{k=14} \sum_{j\epsilon J} h_{jk}(\sum_{i\epsilon B_j} x_k^i)$.

Schedule Pattern Costs

For each nurse $i=1, \cdots, I$ 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 nurse i's schedule preferences;

(3) how this cost is weighed in light of nurse i'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 a nurse 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 constraint, i.e., how severely does she view violations of this nonbinding constraint vis-à-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 schedules nurse i has received in the past. Intuitively, if nurse i has been receiving poor schedules, we would want the cost of a given schedule to be relatively higher than the costs for schedules of other nurses in order 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 nonbinding constraint $n\epsilon N_i$ of schedule x^i;

α_{in} = the 'weight' nurse i gives a violation of nonbinding constraint $n\epsilon N_i$, 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-à-vis nurse i's preferences.

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

Problem Formulation

Let $\lambda\epsilon(0, 1)$ be a parameter that weights staffing level and schedule pattern costs. It is chosen such that the weighted staffing and schedule pattern costs are of approximately equal magnitude. Experience has shown a trial-and-error procedure to be effective in arriving at satisfactory values of λ.

Given λ, the problem is to find $x^1, x^2, \cdots, x^I$ that minimize:

$$\lambda\left[\sum_{k=1}^{k=14} f_k\left(\sum_{i=1}^{i=I} x_k^i\right)+\sum_{k=1}^{k=14} \sum_{j\epsilon J} h_{jk}\left(\sum_{i\epsilon B_j} x_k^i\right)\right] +(1-\lambda)\sum_{i=1}^{i=I} A_i \sum_{n\epsilon N_i} \alpha_{in} g_{in}(x^i)$$

subject to $x^i \epsilon \pi_i$, $i=1, \cdots, I$.

2. DESCRIPTION OF THE SOLUTION PROCEDURE

We use an algorithm that finds near-optimal solutions. It starts with an initial configuration of nurse 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, one finds a schedule that results in a lower schedule configuration cost than the lowest cost to date. When all the schedules in π_i have been tested either a lower cost configuration has been found or 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.

Each set π_i will always contain at least one feasible schedule, because of the manner in which the feasibility sets are constructed. To arrive at an initial solution one may select one schedule from each π_i in an appropriate manner (e.g., select the schedule with the lowest dissatisfaction cost). If we view the feasibility region as the cartesian product of the feasibility regions $\pi_1, \pi_2, \cdots, \pi_I$, the algorithm is simply a cyclic coordinate descent algorithm along the coordinate directions π_i. Each π_i contains all feasible schedules for nurse i. When 4 days off are given 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 the cartesian product contains a finite number of points, namely, $\Pi_{i=1}^{i=I} \|\pi_i\|$, where $\|\pi_i\|$ is the number of schedules in the set π_i.

The following steps describe the algorithm in detail.

1. Determine the set of feasible schedules for each employee, π_i. Let $\|\pi_i\|$ denote the number of schedules in π_i.
2. Calculate the schedule pattern costs for each schedule $x^i \epsilon \pi_i$, for $i=1, \cdots, I$.
3. Choose an initial schedule mix (i.e., a schedule for each nurse $i=1, \cdots, I$) 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. Try the kth candidate schedule, x^{ik}, in the schedule mix by temporarily removing the present schedule for nurse i from the current schedule mix and inserting schedule x^{ik}. Let TEST = the cost of this new schedule mix.
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 place of the current schedule for nurse i in the schedule mix. The schedule mix now contains the 'best schedules found so far.' 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.

3. RESULTS

Preliminary tests were conducted on a small sample problem comparing the algorithm presented above with a branch-and-bound algorithm that yielded the optimal solution. These tests showed that our 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 was 0.367 sec vs. 10.509 for the branch-and-bound. Moreover, this time occurred 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 sec. These times are for a CDC 6400.

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 information was used in the application of the algorithm. We could compare the algorithm schedules and the hospital schedules, since both were generated from the same base data.

Algorithm-Generated Schedules

Figure 2 presents some schedules generated by the algorithm for four weeks of the six-month trial period, October 22 to November 18. Note that on fourteen of the twenty-eight days the actual staffing levels were identical with the desired staffing levels. The unit is understaffed by one nurse on two days and overstaffed by one nurse on twelve days.

Table I presents data relating to what percent of the schedules in π_i have employee dissatisfaction costs greater than or equal to that of the chosen schedule.

Note that in all cases except one the nurses were given a schedule better than 90 percent or more of those in the feasible pattern set. Moreover, we see that in most instances the number of schedules in the set of feasible patterns was well over 100; thus there were many schedules to choose from.

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

Fig. 2. A four-week set of nurse schedules generated by the solution algorithm.

We also note how the algorithm schedules equitably over time. In all cases except one a nurse received the lowest cost schedule pattern in the feasible set during one of the two pay periods, and in that one instance the nurse received a schedule in the 99th percentile in each of the two periods. The effect of the aversion index is evident when we note the general pattern of nurses who receive their best schedules during the first two weeks receiving a slightly worse schedule in the second two weeks and vice versa.

More extensive results will now be given for the entire six-month scheduling test. Figure 3 presents a histogram of deviations from desired staffing levels.

On 90 percent of the days the deviation from the desired staffing level was either 0 or ± 1. Moreover, we do not include measures of under- or overstaffing on the units in question. Hence, if a unit was understaffed for a pay period, we would expect a number of negative deviations. Similar results would hold for overstaffed units. In light of these results we see how well the algorithm works in meeting staffing criteria.

TABLE I

RANKING OF SCHEDULES CHOSEN BY SOLUTION ALGORITHM AS JUDGED BY EMPLOYEE DISSATISFACTION COST CRITERIA

Nurse	$\|\pi_i\|$	Pay period 1, percentile	$\|\pi_i\|$	Pay period 2, percentile
1A	15	100	167	92
1B	331	100	233	93
1C	331	100	370	94
1D	302	99	128	99
1E	331	93	166	100
1F	390	94	331	100
1G	156	100	349	100
2A	235	80	1	100
2B	202	100	331	95
2C	163	100	182	92
2D	52	98	331	100
2E	390	100	390	100

Legend: $\|\pi_i\|$ = Number of schedules in feasible schedule set of nurse i; percentile = percent of schedules in feasible schedule set with employee dissatisfaction cost greater than or equal to schedule selected by the solution algorithm.

In Fig. 4 a histogram presents data, taken over the six months, that show what percentile of a nurse's feasible schedule set the chosen schedule pattern fell in (where percentile is defined as in Table I and where π_i such that $\|\pi_i\| \geqq 10$ were the only sets considered).

Note that the algorithm chose the lowest cost schedule from a nurse's feasible schedule pattern set almost 44 percent of the time and chose a schedule that was in the 90th percentile or better almost 88 percent of the time.

The CPU times for the solution algorithm range from about 2½ sec to 8 sec, depending on the number of nurses and the number of schedules in π_i (on a CDC 6400). The average solution time was around 5 sec.

In most instances the groups consisted of from five to seven nurses with an average of about 200 schedules in their feasible schedule sets.

Comparison Between Schedules Generated by the Algorithm and Those Used by the Hospital

As we mentioned, the actual schedules used by the hospital in which the test was made were on record. Table II presents data relating to various schedule characteristics.

Weekends of four or more days off were nearly the same in both cases. This is not surprising since most of these resulted from special requests by the individual nurses. The algorithm, however, generated 13 more three-

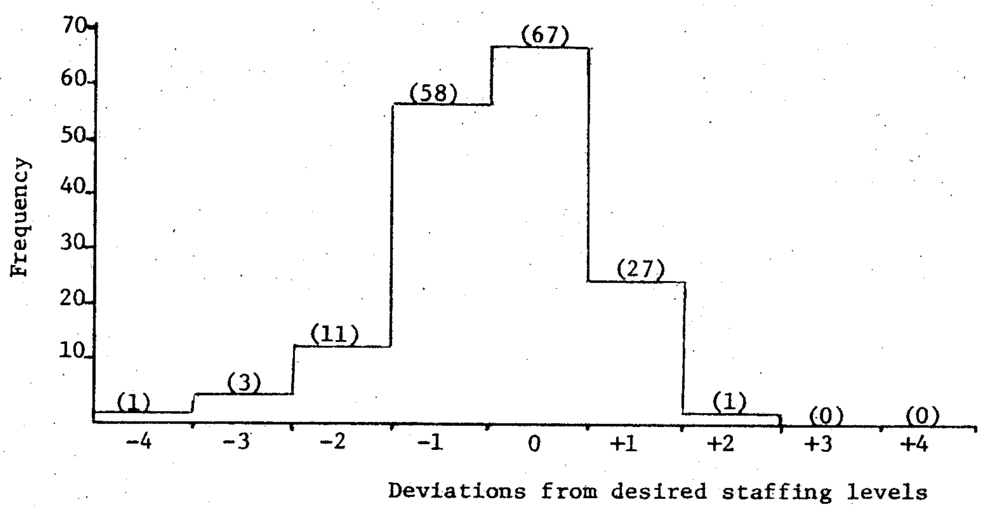

Fig. 3. Distribution of deviations from desired staffing levels by schedules chosen by solution algorithm during the six month test period.

and two-day weekends, than the hospital did. This is a favorable feature since most nurses desire as many weekends off as possible. Moreover, the algorithm generated far fewer split weekends. Again this is favorable since the hospital often does not desire to have such patterns. Both the algorithm and the hospital performed equally well in generating stretches under the individual nurse's minima, but the hospital generated far more stretches over the nurse's maxima. In considering consecutive split days, the algorithm generated more in all instances, although the only significant difference occurred in the generation of two consecutive split days.

We now define $D=|X_k-d_k|$, where $X_k=\sum_{i=1}^{i=I} x_k^i$. This gives us some measure of the deviation between actual and desired staffing levels. These deviations are highlighted in Table III.

In every pay period of every schedule the average daily deviation D of the actual staffing level from the desired staffing level was as small as or

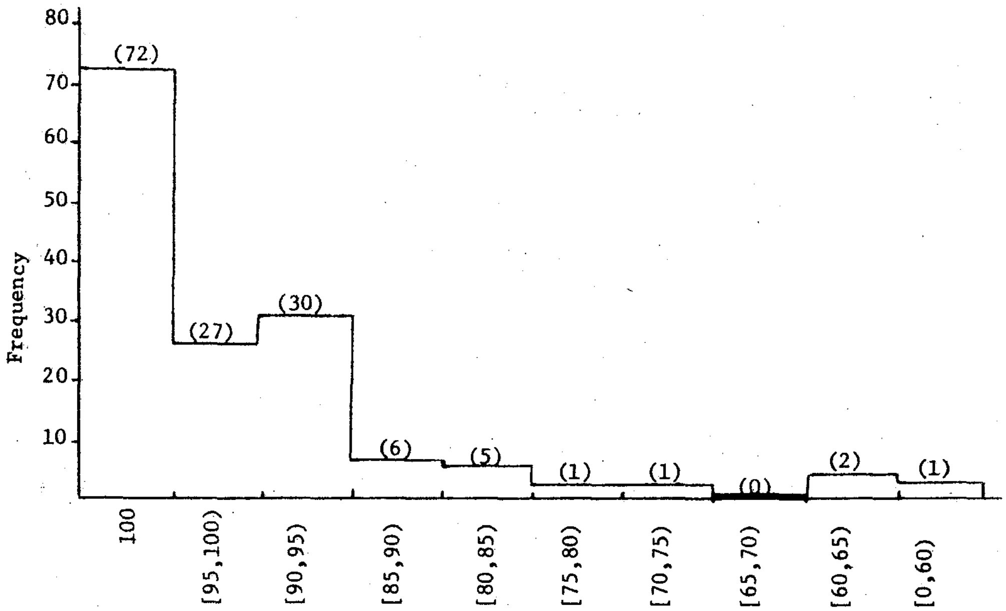

Fig. 4. Distribution of the number of schedules in the feasibility sets with employee dissatisfaction costs greater than that of the schedule chosen by the algorithm (over a six-month period).

smaller for the schedules generated by the algorithm than those generated by the hospital. Moreover, in all cases but two the variance of D was smaller for the algorithm-generated schedules. One of these two occasions occurred in the pay period containing New Year's Day, and in the second occasion the variances differed by only 0.04.

Another measure of the variability of the actual vs. desired staffing

TABLE II

COMPARISON OF SCHEDULE PATTERN CHARACTERISTICS BETWEEN HOSPITAL AND ALGORITHM GENERATED SCHEDULES

	Weekends off				Working stretches		Consecutive split days off (101 pattern)		
	$\geqq$ 4 Day	3 Day	2 Day	Split	Under min	Over desired	1	2	3
Hospital schedules	11	17	43	25	19	57	26	13	5
Algorithm schedules	9	21	52	10	21	44	27	21	6

levels is given in the last two columns of Table III. This is the sum of the squares of the deviations. In all cases the sum of the squared deviations arising from the algorithm-generated schedules is less than those from the hospital-generated schedules.

Extensions

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

TABLE III

COMPARISON OF SOME STAFFING LEVEL STATISTICS FROM HOSPITAL AND ALGORITHM GENERATED SCHEDULES

Schedule period	Pay period	Average D		Variance of D		$\sum_{k=1}^{k=14} (X_k - d_k)^2$	
		Alg	Hos	Alg	Hos	Alg	Hos
9/24	1	0.786	0.929	0.169	0.352	11	17
	2	1.071	1.071	0.209	0.495	19	23
10/22	1	0.357	0.786	0.229	0.454	5	15
	2	0.643	0.786	0.230	0.454	9	15
11/19	1	0.357	0.786	0.229	0.312	5	13
	2	0.429	0.571	0.245	0.387	6	10
12/17	1	1.000	1.286	0.714	1.061	24	38
	2	1.143	1.429	1.694	1.673	42	52
1/14	1	0.214	0.786	0.168	0.454	3	15
	2	1.000	1.143	0.429	0.551	20	26
2/11	1	1.071	1.214	0.495	0.455	23	27
	2	0.643	0.786	0.230	0.454	9	15

Legend: $D = |X_k - d_k|$; Hos = hospital generated schedule; Alg = Algorithm Generated Schedule.

1. We 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, we 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. We schedule the day shift treating these rotation patterns as fixed conditions.

The problem of part-time employees is handled in a way analogous to full-time employees. Feasibility sets π_i are constructed for part-time employees 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 nonbinding constraints. We then proceed in the same manner as with full-time employees, 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.

4. SUMMARY AND CONCLUSIONS

We have presented a mathematical programming model formulating the nurse-scheduling problem as one where we find a set of feasible nurse schedules that seek to minimize the sum of schedule pattern aversion costs incurred by the nurses and staffing level pattern costs incurred by the hospital. The model is realistic, in the sense that it can handle any number of any type of schedule pattern and staffing level constraints, and is general enough to be applied in a number of different hospital settings with differing operating policies.

The solution algorithm given is a cyclic coordinate descent method that generates a solution specifying which days are on and off for each individual nurse being scheduled. The algorithm runs relatively quickly on the computer and generates solutions in a number of ways superior to those presently used in the hospital with which it was compared. Moreover, the algorithm-generated solutions compared favorably with the optimal solution for a test problem.

A nurse-scheduling algorithm based on the ideas presented in this paper has been implemented in a number of hospitals in the United States and Canada. The results from these implementation sites have been most favorable.

ACKNOWLEDGMENT

The authors wish to thank Drs. Ronald Gue, Richard Jelinek, James Brya, R. K. D. Haussmann, and Mr. Tim Zinn for their cooperation, encouragement, and the six months of actual staffing data used in this study.

This research was partially supported by NSF Grant GU-3857.

Note: This paper resulted from a technical report, "Nurse Scheduling Using Mathematical Programming," Northwestern University, Evanston, Illinois, March, 1973.

REFERENCES

1. W. ABERNATHY, N. BALOFF, AND J. HERSHEY, "A Variable Nurse Staffing Model," *Decision Sci.* **5**, 58–72 (1974).
2. W. ABERNATHY, N. BALOFF, J. HERSHEY, AND S. WANDEL, "A Three Stage Manpower Planning and Scheduling Model—A Service Sector Example," *Opns. Res.* **21**, 693–711 (1973).

3. J. P. Howell, "Cyclical Scheduling of Nursing Personnel," *Hospital J.A.H.A.* **40,** 77–85 (1966).
4. J. S. Liebman, "The Development and Application of a Mathematical Programming Model of Personnel Allocation in an Extended Care Nursing Unit," Ph.D. Thesis, Department of Operations Research and Industrial Engineering, Johns Hopkins University, Baltimore, Maryland, 1971.
5. J. S. Liebman, J. P. Young, and M. Bellmore, "Allocation of Nursing Personnel in an Extended Care Facility," *Health Service Res. J.* **7,** 209–220 (Fall, 1972).
6. C. Maier-Rothe and H. Wolfe, "Cycle Scheduling and Allocation of Nursing Staff," Arthur D. Little & Co., Inc., Cambridge, Massachusetts (1972).
7. A. Morrish and A. O'Connor, "Cyclic Scheduling," *Hospitals J.A.H.A.* **14,** 66–71 (1970).
8. E. Price, "Techniques to Improve Staffing," *American J. Nursing* **70,** 2112–2115 (1970).
9. M. Rothstein, "Hospital Manpower Shift Scheduling by Mathematical Programming," *Health Service Res. J.* **8,** 60–66 (1973).
10. D. M. Warner, "A Two-Phase Model for Scheduling Nursing Personnel in a Hospital," Doctoral Dissertation, Tulane University, New Orleans (1971).
11. D. M. Warner, "Scheduling Nursing Personnel According to Nursing Preference: A Mathematical Programming Approach," *Opns. Res.* **24,** 842–856 (1976).
12. D. M. Warner and J. Prawda, "A Mathematical Programming Model for Scheduling Nursing Personnel in a Hospital," *Management Sci.* **19,** 411–422 (1972).
13. H. Wolfe and J. P. Young, "Staffing the Nursing Unit: Part I," *Nursing Res.* **14** (3), 236–243 (1965).
14. H. Wolfe and J. P. Young, "Staffing the Nursing Unit: Part II," *Nursing Res.* **14** (4), 299–303 (1965).
15. T. Zinn, R. Jelinek, G. Delon, and R. Aleman, "A Nurse Schedule and Allocation System in Operation," *Proc. 23rd Annual American Institute of Industrial Engineers Conferences*, pp. 273–282, June 1972.