

NURSE ALLOCATION WITH STOCHASTIC SUPPLY*

By

Holmes E. Miller

Program in Operations Research and Statistics

Rensselaer Polytechnic Institute

Troy, New York 12180

and

William P. Pierskalla

Department of Industrial Engineering and Management Sciences

Northwestern University

Evanston, Illinois 60201

Research Report No. 37-74-P7

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

Abstract

The nurse allocation problem for a given day and shift is formulated as one which minimizes the cost of allocating nurses among nursing classes and units subject to constraints on the demand for and the supply of nursing services. When the number of nurses reporting for work in the various classes and units form a random vector, the allocation problem becomes a stochastic program with recourse. When the random variables in question are defined on a discrete sample space, the stochastic program may be transformed into a deterministic program which can be solved. A near-optimal cyclic coordinate descent algorithm is presented. Results are given pertaining to some sample problems and the sensitivity of the algorithm's solutions to changes in the probability distribution of the supply random variables is discussed.

INTRODUCTION

Nurse allocation in a hospital occurs over varying time frames. In the long term (e.g. months) "allocation" refers to allocating a fixed supply of nurses employed by the hospital to the various units of a hospital requiring nursing services. The results of such an allocation specify the number of nurses permanently assigned to the various classes and units of a hospital. This form of nurse allocation is treated by Abernathy, Baloff, Hershey and Wandel in (1) and (2). In the very short term (e.g. hours or minutes) "allocation" refers to allocating the nurses already working in a nursing unit among all the nursing tasks that must be performed in that unit. Two examples of this form of nurse allocation are those of Wolfe and Young (12), (13) and Liebman (5).

The nurse allocation problem discussed in this paper is of the short term (e.g. for a given shift on a given day). It may be thought of as a necessary adjustment to the process of scheduling nurses for work, (see (6), (8), and (10)). Regardless of the type of nurse scheduling system used by a hospital, the daily schedules specify how many nurses of the various nursing classes (e.g. RN's, LPN's, Nursing Aides) are scheduled to work in the various nursing units of the hospital. The schedules are generated under certain assumptions regarding estimates of the demand for nursing services on the scheduled days and under the implicit assumption that all of the nurses scheduled to work on a given day will indeed report for work on that day. As the scheduled days approach, however, the supply and demand for nursing services are often different from the supply specified by and the demand estimated in the nurse scheduling process. Thus the allocation problem

is one of how to adjust the realized supply of nurses reporting for work to meet the updated demand for nursing services. This adjustment may involve moving nurses across classes and/or units so that they function as nurses in the same/different classes of the same/different units in a manner consistent with hospital policies.

A number of allocation studies of this type and time frame have appeared in the literature. Warner and Prawda (11) formulate the allocation problem as one of seeking to minimize a "shortage cost" of nursing services while satisfying constraints on personnel capacity, integral assignment, and nursing class substitutability. The problem is posed as a mixed integer quadratic programming problem. The model allows for substitutability between nursing classes by allocating nurses from a known hospital wide supply of nurses of each class. It does not address the problem of allocating nurses when supply is stochastic, nor does it treat the problem of allocating nurses when the supply of nurses in each nursing class is already associated with the various nursing units, via rosters of nurses scheduled to work in those units. Howland (4) also assumes a known supply of nurses but allows only downward substitutability between nursing classes (e.g. RN's for LPN's but not vice versa). His model allocates nurses from a float pool. Both of these assumptions restrict the general applicability of the model. In their paper on cyclical nurse scheduling Maier-Rothe and Wolfe also speak of allocating nurses from a float pool to adjust the nurse schedules properly.

In this paper we shall model the process of nurse allocation to allow allocation of nurses across all nursing classes and units, given a roster of nurses scheduled to work in those classes and units. If

a hospital does not wish cross class and/or unit allocation to take place, penalty costs can be defined so that intra-unit cross-class allocation or float pool allocation are special cases of the model. Moreover the supply of nurses reporting for work is treated as a random variable. Thus the model may deal with situations where the supply of nurses is not known with certainty. When the supply of nurses is known, the model treats it as a special case.

THE MODEL

The nurse allocation model presented below is an integer program which minimizes the sum of the direct cost of allocation and the costs of recourse actions due to the random supply, subject to constraints on meeting the demand for nursing services in the various classes and units, and lower bound constraints on the number of nurses allocated across the various classes and units of a hospital. Define:

- y_{ijkm} = The number of nurses of class i , unit j allocated to function as class k , unit m nurses. (integer)
- a_{ijkm} = The number of nurses of class k , unit m equivalent to one class i , unit j nurse. This may be thought of as a full time equivalency ratio.
- u_{ijkm} = The lower bound on the variables y_{ijkm} . (where $u_{ijkm} \geq 0$)
- \bar{r}_{km} = The upper bound on the staffing level for class k , unit m . (known)
- \underline{r}_{km} = The lower bound on the staffing level for class k , unit m . (known)
- c_{ijkm} = The cost of allocating one nurse from class i , unit j to function as a class k , unit m nurse.
- Z_{ij} = The number of nurses available to be allocated from class i , unit j . This is a random variable.

We shall assume the Z_{ij} 's are independent. Also define J = the number of units from which nurses may be allocated; M = the number of units to which nurses may be allocated; I = the number of classes in the J units; and K = the number of classes in the M units. Henceforth we shall assume $J=M$ and $I=K$. If this were not the case, "dummy" classes and/or nursing units could be defined so that the equalities hold.

The nurse allocation problem may be formulated as a stochastic program with simple recourse:

Problem A:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{m=1}^M c_{ijklm} y_{ijklm} \\ & + \sum_{i=1}^I \sum_{j=1}^J E_{Z_{ij}} \min\{q_{ij}(v_{ij}) \mid v_{ij} = Z_{ij} - \sum_{k=1}^K \sum_{m=1}^M y_{ijklm}; \end{aligned}$$

$$v_{ij} \text{ unrestricted}\}$$

$$\text{subject to} \quad \bar{r}_{km} \leq \sum_{i=1}^I \sum_{j=1}^J a_{ijklm} y_{ijklm} \leq \bar{r}_{km}, \text{ for all } k=1, \dots, K$$

$$m=1, \dots, M$$

$$y_{ijklm} \geq u_{ijklm}, \text{ for all } i,j,k,m \text{ and } y_{ijklm} \text{ integer for all } i,j,k,m$$

The first set of terms in the objective function denote the direct allocation costs, i.e. the actual costs of the allocations specified by the y_{ijklm} . The second set of terms in the objective function denote the expected recourse costs. They may be thought of as being the costs of the recourse actions v_{ij} taken to assure that $v_{ij} + \sum_{k=1}^K \sum_{m=1}^M y_{ijklm} = z_{ij}$, where z_{ij} is a realization of the random variable Z_{ij} . Note that $z_{ij} - \sum_{k=1}^K \sum_{m=1}^M y_{ijklm}$ is the difference between the realized supply of nurses in class i , unit j and the total number

of nurses allocated from class i , unit j to all other units. Thus we might intuitively think of $v_{ij} > 0$ as being the cost of asking nurses to work overtime or requesting that nurses scheduled for days off report for work; and $v_{ij} < 0$ as being the cost of nurses who report for work having no classes or units to be allocated to, i.e. "idle time" costs. The first set of inequality constraints specify that the number of "equivalent" nurses (i.e. adjusted by the ratios a_{ijkm}) allocated to each class k in unit m be within the upper and lower bounds specified by \bar{r}_{km} and \underline{r}_{km} . The inequality constraints $y_{ijkm} \geq u_{ijkm}$ place a lower bound on all allocations y_{ijkm} ; and finally all allocations must be integer valued.

Note that the random variables Z_{ij} are defined on a discrete sample space. For example, if n nurses are scheduled to report for work, the number of nurses actually reporting for work will be $n, n-1, \dots, 1$, or 0 . Thus $Z_{ij} \in \{z_{ij}^1, \dots, z_{ij}^{L_{ij}}\}$; $Z_{ij} = z_{ij}^\ell$ with probability P_{ij}^ℓ where $0 \leq P_{ij}^\ell \leq 1$, for all i, j, ℓ and $\sum_{\ell=1}^{L_{ij}} P_{ij}^\ell = 1$, for all i, j . Moreover we note that for each i, j :

$$\begin{aligned} \min\{q_{ij}(v_{ij}) \mid v_{ij} = Z_{ij} - \sum_{k=1}^K \sum_{m=1}^M y_{ijkm}; v_{ij} \text{ unrestricted}\} \\ = q_{ij} \left(Z_{ij} - \sum_{k=1}^K \sum_{m=1}^M y_{ijkm} \right) \end{aligned}$$

since the set of feasible v_{ij} 's contains a single point for each i, j combination. These observations imply that:

$$\begin{aligned} \sum_{i=1}^I \sum_{j=1}^J E_{Z_{ij}} q_{ij} \left(Z_{ij} - \sum_{k=1}^K \sum_{m=1}^M y_{ijkm} \right) \\ = \sum_{i=1}^I \sum_{j=1}^J \sum_{\ell=1}^{L_{ij}} P_{ij}^\ell q_{ij} \left(z_{ij}^\ell - \sum_{k=1}^K \sum_{m=1}^M y_{ijkm} \right) \end{aligned}$$

For example the functions $q_{ij}(\cdot)$ could be defined as:

$$q_{ij}(\alpha) = \begin{cases} e_{ij} \alpha^2 & \text{if } \alpha < 0 \\ b_{ij} \alpha^2 & \text{if } \alpha \geq 0 \end{cases}$$

where $\alpha = z_{ij}^{\ell} - \sum_{k=1}^K \sum_{m=1}^M y_{ijkm}$ and $e_{ij} \geq 0, b_{ij} \geq 0$. This function forces the final allocations y_{ijkm} to result in allocations from classes and units lying "close to" the expected supply of nurses in those classes and units. Moreover it reflects the nonlinear manner in which the recourse costs increase. For example the cost of two nurses working overtime on one day might incur a greater cost than one nurse working overtime on two separate days, given salary and other incidental interactive costs.

Hence Problem A may be reformulated as:

Problem B:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{m=1}^M c_{ijkm} y_{ijkm} + \\ & \sum_{i=1}^I \sum_{j=1}^J \sum_{\ell=1}^{L_{ij}} p_{ij}^{\ell} q_{ij} (z_{ij}^{\ell} - \sum_{k=1}^K \sum_{m=1}^M y_{ijkm}) \\ \text{subject to} \quad & \sum_{i=1}^I \sum_{j=1}^J a_{ijkm} y_{ijkm} \leq \bar{r}_{km} \quad ; \text{ for all } k=1, \dots, K \\ & \phantom{\text{subject to}} \phantom{\sum_{i=1}^I \sum_{j=1}^J} \phantom{a_{ijkm} y_{ijkm}} \leq \bar{r}_{km} \quad ; \text{ for all } m=1, \dots, M \\ & y_{ijkm} \geq u_{ijkm}, y_{ijkm} \text{ integer for all } i, j, k, m \end{aligned}$$

When the functions $q_{ij}(\cdot)$ are convex quadratic functions as defined above, Problem B is a convex integer program subject to linear constraints.

SOLUTION PROCEDURE

The solution algorithm to Problem B presented below is near-optimal rather than optimal. This is due to the extremely large number of possible feasible solutions to Problem B. For example, consider

a problem where nurses are allocated among three classes of four units, i.e. $I=K=3$ and $J=M=4$. Assume that $r_{km} = 5$ and $\bar{r}_{km} = 7$, for each k,m and that $a_{ijkm} = 1$ and $u_{ijkm} = 0$, for i,j,k,m . Even though this problem is "small" in nature, there exist on the order of 10^{54} feasible solutions. If the lower bounds u_{ijkm} are tightened to $u_{kmmk} = 4$, for each k,m and $u_{ijkm} = 0$ otherwise, there still exist over 10^{31} feasible solutions. These facts suggest implicit enumeration algorithms may not be feasible. Moreover the non-linearity of the objective function and the large number of variables casts doubt on using other optimum integer methods, such as cutting planes.

The algorithm that will be used is a cyclic coordinate descent method, where the search directions are coordinate directions and the cartesian product of coordinate directions. The rationale for choosing the search directions stems from representing the variables y_{ijkm} in matrix form:

$$\begin{array}{cccc}
 y_{1111} & y_{1211} & \cdots & y_{IJ11} \\
 y_{1112} & y_{1212} & \cdots & y_{IJ12} \\
 \vdots & \vdots & & \vdots \\
 y_{11KM} & y_{12KM} & \cdots & y_{IJKM}
 \end{array}$$

Note that a) each row represents a fixed class and unit to which nurses are allocated, i.e. the subscripts k and m are fixed; b) each column represents a fixed class and unit from which nurses are allocated, i.e. the subscripts i and j are fixed; and c) the "diagonal" elements, y_{kmmk} , represent the number of nurses of a particular class and unit allocated to function in that same class and unit.

It is intuitively clear that any near-optimal allocation will contain relatively high values for the diagonal elements, given a realistic distribution of allocation costs. That is, most nurses will be allocated to function as nurses in their own class and unit, and some will be allocated across classes and/or units to account for changes in supply and demand conditions.

This observation is utilized both in choosing the initial solution and in generating search directions. Specifically the initial solution is chosen to be a feasible "diagonal solution". For example, each diagonal element in row km of the matrix representation of the y_{ijkm} would equal r_{km} and all non-diagonal elements would equal zero (where we assume r_{km} is integer). In generating search directions we will include all directions representing pairwise changes between the diagonal and non-diagonal elements of the various rows and columns of the matrix representation of the y_{ijkm} .

Specifically we search along (where E_{ijkm} refers to the coordinate direction in Euclidian Space corresponding to the variable y_{ijkm}):

- (A) Directions of the form $E_{kmmk} \times E_{ijkm}$,
- (B) Coordinate directions E_{ijkm} ,
- (C) Directions of the form $E_{ijij} \times E_{ijkm}$.

Searching in the directions specified in (A) tends to lower recourse costs by reducing the discrepancies between the number of nurses allocated from classes and units and the expected number of nurses scheduled to work in those classes and units, given a fixed total allocation.

Searching in the directions specified in (B) also tends to

lower recourse costs for the same reason indicated in (A). This time, however, the reduction is facilitated by increasing or decreasing the total number of nurses allocated among all classes and units, as opposed to changing the "mix" of a fixed total allocation.

Searching in the directions specified in (C) tends to lower direct allocation costs by finding feasible allocations with the same recourse costs but lower direct allocation costs.

The algorithm about to be presented searches these directions in a cyclical manner and terminates when no better solution can be found.

Solution Algorithm

- (1) Select an initial feasible solution. For example, let $y_{kmkm} = r_{km}$, for all k, m and $y_{ijkm} = 0$, otherwise.
Let BEST = this solution's cost and set the termination counters,
 $K1 = K2 = K3 = 0$.
- (2) If all termination counters are greater than zero, stop. Otherwise let $i = 1, j = 1, k = 1, m = 1$ and $K1 = 0$.
- (3) Search in direction $E_{kmkm} \times E_{ijkm}$ for a feasible solution yielding a total cost lower than BEST. If one is found, go to (6).
- (4) Let $K1 = K1 + 1$. If $K1 > I \cdot J \cdot K \cdot M$, go to (7).
- (5) Increment subscripts in the following order: j, i, m, k (i.e. go "across the rows"). Go to (3).
- (6) Let BEST = the new total cost and let y_{kmkm} and y_{ijkm} be the allocations that produced it.
Set all termination counters to zero and go to (3).

- (7) If all termination counters are greater than zero, stop. Otherwise let $k = 1$, $m = 1$, $i = 1$, $j = 1$ and $K2 = 0$.
- (8) Search in direction E_{ijkm} for a feasible solution yielding a total cost lower than BEST. If one is found, go to (11).
- (9) Let $K2 = K2 + 1$. If $K2 > I \cdot J \cdot K \cdot M$, go to (12).
- (10) Increment subscripts in the following order: j , i , m , k (i.e. go "across rows"). Go to (8).
- (11) Let BEST = the new total cost and let y_{ijkm} be the allocation that produced it. Set all termination counters to zero, and go to (8).
- (12) If all termination counters are greater than zero, stop. Otherwise let $k = 1$, $m = 1$, $i = 1$, $j = 1$ and $K3 = 0$.
- (13) Search in direction $E_{ijij} \times E_{ijkm}$ for a feasible solution yielding a total cost lower than BEST. If one is found, go to (16).
- (14) Let $K3 = K3 + 1$. If $K3 > I \cdot J \cdot K \cdot M$, go to (2).
- (15) Increment subscripts in the following order: m , k , j , i (i.e. go "down columns"). Go to (13).
- (16) Let BEST = the total cost and let y_{ijij} and y_{ijkm} be the allocations that produced it. Set all termination counters to zero, and go to (13).

Note that steps (2) - (6) of the algorithm correspond to searches in directions specified in (A) above; steps (7) - (11) correspond to searches specified in (B) above; and steps (12) - (16) correspond to searches specified in (C) above.

Also note that directions (A) - (C) do not include directions such as $E_{ijkm} \times E_{lpkm}$ or $E_{ijkm} \times E_{ijlp}$, i.e. the cartesian product of two non-diagonal directions in a particular row or column of the matrix representation of y . This is due to the observation that the relatively

larger allocations correspond to the "diagonal" components y_{kkmk} . Implicit in this observation is the likelihood that the search directions corresponding to the aforementioned "diagonal" components.

Results

The solution algorithm just presented was applied to five sample problems where: $I = K = 3$, $J = M = 4$ and $q(\alpha) = q_{ij}(\alpha)$, for all i, j where

$$q(\alpha) = \begin{cases} e\alpha^2 & \text{if } \alpha < 0 \\ b\alpha^2 & \text{if } \alpha \geq 0 \end{cases}$$

The sample spaces for the Z_{ij} and the probability distributions on those sample spaces varied for each class i - unit j combination.

Figure 1 presents one of the integer allocations generated:

From To	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,2)	(2,3)	(2,4)	(3,1)	(3,2)	(3,3)	(3,4)	Total Allocated to	Min Req	Max Req
(1,1)	4	2	0	0	0	0	0	0	0	0	0	0	6	6	7
(1,2)	0	4	1	0	0	0	0	0	0	0	0	0	5	5	6
(1,3)	0	0	6	0	0	0	0	0	0	0	0	0	6	4	6
(1,4)	1	0	0	4	0	0	0	0	0	0	0	0	5	5	7
(2,1)	0	0	0	0	6	0	0	0	0	0	0	0	6	4	6
(2,2)	0	0	0	0	0	6	0	0	0	0	0	0	6	4	7
(2,3)	0	0	0	0	0	0	6	0	0	0	0	0	6	6	7
(2,4)	0	0	0	0	0	0	0	6	0	0	0	0	6	5	6
(3,1)	0	0	0	0	0	0	0	0	6	0	0	0	6	4	6
(3,2)	0	0	0	0	0	0	0	0	0	5	0	0	5	5	7
(3,3)	0	0	0	0	0	0	0	0	0	0	6	0	6	5	6
(3,4)	0	0	0	0	0	0	0	0	0	1	0	5	6	6	7

Legend: $(i,j) = (\text{class } i, \text{unit } j)$

FIGURE 1
AN ALLOCATION GENERATED BY THE INTEGER SOLUTION ALGORITHM

Although the optimal integer allocation was unknown in each of the five problems, a lower bound on it's cost was obtained by solving the five problems in continuous variables and using a lower bound on the continuous optimum as a lower bound on the integer optimum (see (7)). The solution algorithm employed was the Frank-Wolfe method. Table 1 presents a comparison of the maximum lower bounds after 100 iterations of the Frank-Wolfe algorithm and the value of the integer allocations generated by the solution algorithm.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Problem Number	e	b	β_{100}	Value of Integer Allocation	(5)/(4)	Value of rounded Frank-Wolfe Allocation
1	6	2	36.8	43.8	1.19	40.4
2	10	4	61.6	72.7	1.18	71.5
3	15	7	94.5	113.9	1.21	116.8
4	30	15	187.9	243.4	1.30	250.4
5	50	25	310.6	375.0	1.21	543.0

TABLE 1

COMPARISON BETWEEN INTEGER SOLUTION ALGORITHM
AND CONTINUOUS FRANK-WOLFE ALGORITHM

Legend:

e = constant in recourse cost function for argument < 0

b = constant in recourse cost function for argument ≥ 0

β_{100} = maximum lower bound in Frank-Wolfe algorithm after 100 iterations

In four out of the five sample problems, the cost of the allocation generated by the integer algorithm was about $1\frac{1}{5}$ times that of the maximum lower bound, β_{100} . Considering the differences in the two algorithms, one being non-integer and the other integer, we conclude the integer algorithm appears to be generating respectable integer

allocations. Later we shall see the integer algorithm generates allocations that are often integer optimal for a different set of integer allocation problems.

The allocations generated by the integer algorithm shall now be compared with the closest feasible integer allocations obtained by rounding the non-integer allocations of the Frank-Wolfe algorithm obtained after 100 iterations of the algorithm. Although these rounded allocations may not be integer optimal, such a procedure is often used as an approximation to the integer optimum when the true integer optimum is unknown.

Comparing columns (5) and (7) Table 1 indicates the integer algorithm generated allocations with objective function values lower than those of the closest feasible integer allocations to the Frank-Wolfe solutions in three out of the five sample problems. Moreover in the other two cases the costs of the integer algorithm were only slightly higher.

The integer algorithm was then applied over a two week period to allocating nurses among the classes of nine units on the day shift of a 800 bed hospital. The hospital only kept records of the number of nurses scheduled to work in each class (RN, LPN) of each unit on each day. Thus the following assumptions were made:

- (1) Since the hospital's policy was only to allocate nurses among units of the same class, there was no need to determine the full time equivalency ratios, a_{ijkm} (where we assume all nurses in the same class were equivalent). Indeed, separate allocations were run for each class.
- (2) For each class i and unit j we assumed (where n = the number

of nurses scheduled to report for work that day):

a) $n \geq 1$ $P(Z_{ij}=n) = 0.9$; $P(Z_{ij}=n-1) = 0.1$, $P(Z_{ij}=m) = 0.0$, $m \neq n$ or $n-1$

b) $n=0$ $P(Z_{ij}=0) = 1.0$; $P(Z_{ij} \neq 0) = 0.0$

(3) The costs were structured as: $c_{kkmkm} = 0$, $k=1,2$; $m=1,\dots,9$;
 $c_{ijjkm}=1$, $i=1,2$; $j=1,\dots,8$ and $m=1,\dots,9$; $c_{i9im}=999$,
 $i=1,2$; $m=1,\dots,8$; i.e. hospital policy dictated that allocations from unit 9 were not allowed since this was a Medical Intensive Care Unit.

(4) The recourse cost functions were (for all i,j)

$$q_{ij}(\alpha) = \begin{cases} 100 \alpha^2 & \text{for } \alpha < 0 \\ 50 \alpha^2 & \text{for } \alpha \geq 0. \end{cases}$$

(5) The minimum and maximum requirements (i.e. \underline{r}_{km} and \bar{r}_{km}) were chosen in accordance with the general number of nurses scheduled to work over the two week period given in the hospital data.

Table 2 presents the results from these runs.

In each of the 28 problems solved, it was possible to determine the cost of the optimal integer allocation, due to the relative simplicity of the costs and the probability distribution given in the assumptions. These optimal costs are given in columns (3) and (6) for the RN and LPN problems. Comparing the values of columns (3) and (6) with those in columns (2) and (5) (i.e. the costs of the allocations generated by the integer algorithm) reveals the optimal allocation was achieved in 20 out of the 28 cases.

Day (Col.)	Registered Nurses			Licensed Practical Nurses		
	Initial Value (1)	Final Value (2)	Optimal Value (3)	Initial Value (4)	Final Value (5)	Optimal Value (6)
1	940	91	91	400	92	92
2	700	81	81	810	81	81
3	910	92	91	510	90	90
4	1335	85	84	420	91	91
5	875	82	81	280	91	91
6	1065	85	84	760	83	83
7	925	85	84	560	83	83
8	1295	93	92	1190	92	92
9	1700	85	83	405	90	90
10	650	90	90	420	91	91
11	540	90	90	900	84	84
12	400	92	91	610	92	92
13	650	83	83	385	91	91
14	650	83	83	435	92	92

Legend:

Initial Value: Objective function value of initial solution to integer algorithm (minimum requirements solution)

Final Value: Objective function value of final allocation generated by integer algorithm

Optimal Value: Objective function value of optimal integer solution

TABLE 2

COST COMPARISON BETWEEN ALLOCATIONS FROM INTEGER SOLUTION ALGORITHM AND OPTIMAL INTEGER ALLOCATIONS

Columns (1) and (4) give the cost of the initial solutions used in the algorithm. Their presence serves a dual purpose. First, since the initial solutions were chosen to be the minimum requirements solutions (i.e. $y_{kkm} = r_{km}$ for all k, m and all other components of y

equalling 0), it gives some indication of the relatively high costs of such allocations. Second, comparing columns (1) and (2), and (4) and (5) indicates how "far" the algorithm must travel to reach its local optima, as measured by the value of the objective function.

The CPU times for the five sample problems presented earlier ranged from 7.79 seconds to 9.46 seconds, averaging 8.65 seconds. For the 28 problems just presented they ranged from 2.26 seconds to 5.88 seconds, averaging 4.59 seconds. All runs took place on a CDC 6400.

Sensitivity Analysis

In this section we shall consider 4 out of the 28 problems just presented and solve each of these for ten different discrete probability distributions. From the results some observations shall be made regarding the sensitivity of the problems chosen and of the algorithm to changes in p .

The probability distributions chosen will be denoted by (i,j,k) . Let n be the number of nurses scheduled to report for work. Then we choose (i,j,k) such that:

$$\text{a) For } n > 1: P(Z_{ts}=n) = i; P(Z_{ts}=n-1) = j; P(Z_{ts}=n-2) = k;$$

$$P(Z_{ts}=m) = 0. \text{ For all } m \neq n, n-1, \text{ or } n-2.$$

$$\text{b) For } n=1: P(Z_{ts}=1) = i; P(Z_{ts}=0) = j + k; P(Z_{ts}=n) = 0., \text{ and } n \neq 0 \text{ or } 1.$$

The case $n=0$ did not present itself in these runs. Table 3 presents the results of this analysis.

In each of the four problems it was possible to calculate the value of the optimal integer allocation. In all cases the optimal integer allocation was generated by the solution algorithm.

The results of Table 3 clearly indicate the optimal integer allocations to the four problems are relatively insensitive to changes

in p. In problems #1 and #4 one allocation is integer optimal for the first four probability distributions listed and another is integer optimal for the last six. In problems #2 and #3 the same allocation is integer optimal for all ten probability distributions.

Probability Distribution	Prob.#1 (RN)		Prob.#2 (LPN)		Prob.#3 (RN)		Prob.#4 (LPN)	
	Pattern	Cost	Pattern	Cost	Pattern	Cost	Pattern	Cost
(1.,0.,0.)	A	1	C	2	D	2	E	2
(.9,.1,0.)	A	91	C	92	D	92	E	92
(.8,.2,0.)	A	181	C	182	D	182	E	182
(.7,.3,0.)	A	271	C	272	D	272	E	272
(.6,.4,0.)	B	301	C	362	D	362	F	332
(.5,.5,0.)	B	301	C	452	D	452	F	377
(.8,.1,.1)	B	391	C	422	D	362	F	422
(.7,.2,.1)	B	391	C	512	D	452	F	467
(.6,.2,.2)	B	481	C	842	D	722	F	692
(.5,.3,.2)	B	481	C	932	D	512	F	737

Legend:

Pattern: Distinct allocation pattern generated by the integer algorithm

Cost: The cost of this pattern

All of the costs given in this table are integer optimal for the problems and probability distributions.

TABLE 3

SENSITIVITY ANALYSIS OF FOUR PROBLEMS TO CHANGES IN p.

We now present the following Proposition regarding the sensitivity of an integer allocation $y = (y_{111}, \dots, y_{IJ11}, \dots, y_{11KM}, \dots, y_{IJKM})$ to changes in the probability distribution $p = (p_{11}^1, \dots, p_{11}^L, \dots, p_{IJ}^1, \dots, p_{IJ}^{LIJ})$. It's proof is in (7) and follows directly from the objective function of Problem B being convex in the variables y_{ijkm} :

Proposition:

If allocation y is integer optimal for any n probability distributions p^1, \dots, p^n ; it is integer optimal for any convex combination of those n distributions (where all other parameters are fixed).

For example, allocation C in problem #2 would be the integer optimum for all convex combinations of any subset of the ten probability distributions specified. Thus allocation C is also the integer optimum for

$$\frac{1}{2}(1.,0.,0.) + \frac{1}{2}(.5, .3, .2) = (.75, .15, .10).$$

The above discussion indicates the optimal solution sets are quite stable, even in the integer case and that slight errors in estimating the probability distribution p do not drastically affect the optimal integer allocation.

The results in Table 3 also indicate the ability of the integer algorithm to "switch" away from an allocation when it is no longer integer optimal. For example, in problem #1 allocation A is integer optimal for $(.7, .3, 0.)$ and allocation B is integer optimal for $(.6, .4, 0.)$. The cost of allocation A when $(.6, .4, 0.)$ holds is 362, which indicates it is not integer optimal. The algorithm recognized this fact by choosing allocation B rather than allocation A .

Naturally there is no guarantee that the algorithm will always generate an integer optimal allocation. When it does not, statements similar to the above cannot be made with the same mathematical precision. The results do suggest, however, that a near-optimal integer allocation for one probability distribution will still be near-optimal for another distribution "close" to it. This is comforting in light of slight errors that may be present in estimating the probability distributions.

SUMMARY AND CONCLUSIONS

In this paper the daily process of allocating nurses among the various nursing classes and units of a hospital was formulated as a stochastic program with simple recourse. Because of the discrete nature of the random variables Z_{ij} , the number of nurses reporting for work in class i of unit j , the problem was transformed into a deterministic program. A near-optimal cyclic coordinate descent algorithm was presented, exploiting the fact that a characteristic of low cost solutions is the presence of a large number of nurses allocated to function in their own classes and units. In the sample problems where the integer optimum was unknown, the algorithm generated solutions having costs only 20% higher than those of the maximum lower bound obtained from the Frank-Wolfe algorithm, applied to a continuous variable version of the same problem. When the integer optimal solutions were known, the solution algorithm found them in 20 of the 28 problems tested. Finally, the optimal integer allocations chosen by the algorithm appear to be quite insensitive to changes in the probability distributions of the random variables Z_{ij} .

The model may be applied in many different nurse allocation settings. It can treat allocation among different classes within the same unit, allocation among different units within the same class, allocations among different classes and units, allocation from a float pool or among a subset of the total number of nursing classes and units. Moreover it can be applied to problems where the supply of nurses is stochastic or where it is deterministic.

References

1. Abernathy, W., N. Baloff, and J. Hershey, "A Variable Nurse Staffing Model". Decision Sciences, Vol. 5, No. 1, (January 1974), pp. 58-72.
2. Abernathy, W., N. Baloff, J. Hershey, and S. Wandel, "A Three Stage Manpower Planning and Scheduling Model-A Service Sector Example," Operations Research, Vol. 21, No. 3 (May-June 1973).
3. Frank, M. and P. Wolfe, "An Algorithm for Quadratic Programming," Naval Research Logistics Quarterly, Vol. 3 (1956), pp. 95-110.
4. Howland, H., "Nurse Allocation from a Pool". Paper presented at the 44th National ORSA Meeting, San Diego, California (November 1973).
5. Liebman, J., J. Young and M. Bellmore, "Allocation of Nursing Personnel in an Extended Care Facility", Health Service Research Journal, Vol. 7, No. 4 (Fall, 1972) pp. 209-220.
6. Maier-Rothe, C. and H. Wolfe, "Cycle Scheduling and Allocation of Nursing Staff", Socio-Economic Planning Sciences (1973).
7. Miller, H., Mathematical Programming Models of Nurse Scheduling and Allocation, Doctoral Dissertation, Northwestern Univ.; Evanston, Illinois; (August, 1974).
8. Miller, H., W. Pierskalla, and G. Rath, "Nurse Scheduling Using Mathematical Programming," Department of Industrial Engineering and Management Sciences, Northwestern University (1974).
9. Reiter, S. and G. Sherman, "Discrete Optimizing," Journal of the Society of Industrial and Applied Mathematics, Vol. 13, No. 3, (September 1965).
10. Warner, D., "Scheduling Nursing Personnel According to Nursing Preference-A Mathematical Programming Approach." Paper presented at 45th National ORSA Meeting, Boston, Mass. (April 1974).
11. Warner, D. and J. Prawda, "A Mathematical Programming Model for Scheduling Nursing Personnel in a Hospital," Management Science, Vol. 19, No. 4 (December, 1972), pp. 411-422.
12. Wolfe, H. and J. Young, "Staffing the Nursing Unit: Part I," Nursing Research, Vol. 14, No. 3, (1965) pp. 236-243.
13. Wolfe, H. and J. Young, "Staffing the Nursing Unit: Part II," Nursing Research, Vol. 14, No. 4, (1965) pp. 299-303.
14. Ziemba, W., "Stochastic Programs with Simple Recourse," Stanford University, Dept. of Operations Research Technical Report No. 72-15 (August, 1972).