Journal of the Operations Research
Society of Japan
Vol. 41, No. 4, December 1998

ナース・スケジューリングに有効なアプローチ
—2 交替制アルゴリズムにおける実現—

池上敦子　　丹羽　明
成蹊大学

(受理 1997 年 8 月 19 日；再受理 1997 年 11 月 27 日)

和文概要　この論文ではナース・スケジューリングを解くための有効なアプローチを提案する．

ナース・スケジューリング問題の拘束条件は大きく 2 種類のものに分けることができる．1つは毎日の勤務に支障を起こさないための勤務メンバー構成に対するもの (縦の条件)，もう1つは無理のない勤務の並びや休みや勤務の回数といった各看護婦毎に考慮すべきもの (横の条件) である．提案するアプローチでは，これらを切り分けて看護婦毎に部分問題を定義する．各看護婦に与えられた横の条件を拘束条件とし，毎日の勤務に対する「縦の条件を満たさない度合い」最小化を目的関数として与える．そして，これらの部分問題を繰り返し解くことによって全体としての実行可能解を得ようとする考え方である．具体的には，各看護婦について実行可能勤務パターンを作成し勤務メンバー構成条件にあわせてこれらを組み合わせていく．

提案するアプローチを実現するために 2 交替制の問題に対するアルゴリズムを構築し実際の問題を解いた．このアルゴリズムは数多い拘束条件に対して効率よく実行可能解を与えた．

1.　はじめに

我が国の看護婦交替制勤務には大きく 2 種類，1 日を日勤と準夜勤と深夜勤に分ける 3 交替制と，日勤と夜勤に分ける 2 交替制がある．これまでは 3 交替制が主であったが，現在 2 交替制導入が試みられている*．

この論文では，この交替制勤務における看護婦の勤務スケジュールを決定する問題（ナース・スケジューリング）を扱う．表 1 に勤務表を簡略化したものを示す．ここで挙げた例は 2 交替制部署の勤務表であるが，各セルには各看護婦の各日の勤務の記号が記されている．夜勤は 2 日にわたる勤務であることから連続する 2 つの記号「Nn」で表され，日勤は「–」休みは「/」セミナー等その他の勤務は「+」で表されている．

看護業務は一般の製造業務などと異なり，対象が人であることから決して失敗が許されない．つまり「看護の質」を無視できない業務である．看護の質というのは数値になりにくいが，勤務スケジュールが看護の質に及ぼす要素つまり「看護の質をどう守ることができるか」を考えると，大きく 2 つのことが挙げられる．

1 つめは，毎日の各勤務の看護婦メンバー構成でその質を守ることである．どんな事態にも対応できる要員を揃えておくということであり，勤務表を各列「縦に見た」場合のメンバー構成の充実ということである．これを「縦の条件」と呼ぶことにする。

そして，もう 1 つは各看護婦についての勤務スケジュールである．同じ 1 人の看護婦で

---

* 厚生省が 3 年毎におこなっている「医療施設調査（静態調査）」（厚生省大臣官房統計情報部編）によると，2 交替制勤務を導入している病院の割合は全国で，昭和 62 年 17.6%，平成 2 年 23.7%，平成 5 年 31.3%と増加の傾向にある．平成 8 年には厚生省による国立病院·療養所での 2 交替制導入の試行が実施され，更にこの傾向は進んでいると思われる．

表 1: 2交替制看護婦勤務表の例 (Nn:夜勤,−:日勤,+:他勤務,/:休み)

看護婦番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	/	−	Nn	+
	水	木	休	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	休	金	土	日	月	火	水	木	休み	日勤	夜勤	ほか
1	N	n	/	/	−	−	/	−	N	n	/	+	−	/	−	N	n	N	n	/	−	−	−	N	n	/	/	/	N	n	9	8	6	1
2	−	N	n	/	−	−	−	/	/	N	n	N	n	/	/	/	−	−	N	n	N	n	/	/	/	−	+	−	−	N	9	9	6	1
3	/	−	−	N	n	/	−	−	/	+	+	−	N	n	/	/	−	−	N	n	/	−	/	/	/	N	n	N	n	/	10	8	5	2
4	/	/	/	−	−	−	−	/	−	−	−	/	/	−	/	−	/	−	−	−	+	/	−	−	−	−	−	/	−	−	10	19	0	1
5	n	/	/	−	N	n	N	n	/	−	−	−	/	−	N	n	N	n	/	−	/	−	N	n	N	n	/	/	−	/	9	8	6	0
6	/	−	−	/	/	N	n	N	n	/	−	/	−	N	n	N	n	/	−	+	−	N	n	/	/	/	−	−	−	−	9	10	5	1
7	−	/	N	n	N	n	/	−	−	/	N	n	/	−	−	−	+	/	/	N	n	/	/	−	−	−	N	n	/	−	9	10	5	1
8	−	−	N	n	/	−	/	N	n	/	/	−	−	N	n	/	N	n	/	/	/	/	N	n	N	n	/	−	−	−	10	8	6	0
9	−	−	−	N	n	/	/	−	N	n	N	n	/	/	−	−	−	/	−	/	/	N	n	N	n	/	−	/	/	N	10	9	6	0
10	N	n	/	/	/	N	n	/	−	−	−	−	/	+	N	n	/	−	−	/	N	n	/	−	/	−	−	N	n	/	10	9	5	1
11	+	−	/	−	N	n	N	n	/	/	−	−	/	N	n	/	−	/	/	−	−	−	/	−	/	−	N	n	/	−	10	11	4	1
12	n	/	−	−	−	/	−	/	/	N	n	N	n	/	−	−	−	/	/	N	n	/	/	−	−	/	N	n	N	n	10	9	5	0
13	−	N	n	/	/	/	−	−	N	n	/	−	−	/	−	N	n	/	−	−	N	n	/	−	−	/	−	−	/	/	10	12	4	0
14	+	N	n	/	/	−	N	n	/	−	/	/	N	n	/	+	+	N	n	/	−	/	−	−	−	N	n	/	−	/	10	7	5	3
15	N	n	/	−	−	−	+	/	−	−	/	/	−	−	/	+	−	−	N	n	/	/	/	N	n	/	−	/	N	n	10	10	4	2
16	n	/	−	/	/	−	−	−	/	−	−	N	n	/	−	−	/	−	−	−	/	−	N	n	/	/	/	N	n	N	10	12	4	0
17	−	/	N	n	/	/	+	N	n	/	/	/	N	n	/	−	−	N	n	/	−	−	−	/	N	n	/	−	−	+	10	8	5	2
18	−	−	−	N	n	/	+	−	−	N	n	/	−	−	N	n	/	/	/	N	n	N	n	/	/	−	/	/	/	+	10	8	5	2
19	/	/	/	/	−	N	n	/	−	−	N	n	/	−	−	/	N	n	/	−	−	−	−	/	−	N	n	/	−	−	10	12	4	0
20	+	/	−	N	n	/	−	−	/	−	−	/	N	n	/	−	/	−	−	N	n	N	n	/	/	/	−	−	N	n	9	10	5	1
21	N	n	/	/	/	N	n	/	−	+	+	/	−	N	n	/	−	−	N	n	/	−	/	−	−	N	n	/	−	−	9	9	5	2
22	−	−	/	/	/	−	N	n	/	+	+	−	−	/	N	n	N	n	/	−	−	−	/	/	N	n	N	n	/	−	9	9	5	2
23	/	/	−	−	N	n	N	n	/	−	+	N	n	/	−	−	N	n	/	−	−	/	−	N	n	/	−	+	/	/	9	9	5	2
24	n	N	n	/	−	−	−	N	n	/	−	−	−	−	/	/	/	N	n	/	+	−	−	−	/	/	/	N	n	N	9	10	5	1
25	N	n	/	/	/	−	−	/	N	n	N	n	/	/	/	−	−	N	n	/	+	−	N	n	/	−	−	−	−	−	9	10	5	1
26	−	−	N	n	/	+	+	−	−	N	n	/	/	−	−	N	n	/	−	−	N	n	/	/	−	/	/	/	N	n	9	9	5	2
27	−	N	n	/	/	−	−	−	N	n	/	/	−	−	−	N	n	/	/	−	N	n	/	−	−	−	N	n	/	/	9	11	5	0
28	−	−	/	−	−	+	+	−	/	/	N	n	N	n	/	−	−	/	/	N	n	N	n	/	−	/	−	−	/	N	9	10	5	2
29	−	−	/	/	/	−	N	n	/	−	−	N	n	/	/	−	−	−	/	−	/	N	n	N	n	/	−	−	−	N	9	12	5	0
30	n	/	/	−	−	N	n	/	−	/	−	N	n	N	n	/	−	/	−	+	−	/	N	n	/	−	−	N	n	/	9	9	5	1
31	+	/	−	−	−	/	−	N	n	/	/	−	−	/	−	N	n	N	n	/	−	−	/	N	n	N	n	/	/	−	9	10	5	1
32	−	/	N	n	N	n	/	−	−	−	/	−	+	N	n	/	−	−	N	n	/	/	/	/	N	n	/	−	−	−	9	10	5	1
33	+	N	n	N	n	/	−	+	/	N	n	/	−	−	N	n	/	−	−	N	n	/	−	−	−	−	/	/	/	/	9	9	5	2
34	+	−	/	N	n	N	n	/	−	−	−	/	N	n	N	n	/	/	/	/	−	−	−	−	/	N	n	/	−	+	9	9	5	2
35	N	n	N	n	/	/	/	N	n	N	n	/	+	/	+	+	+	+	/	+	+	+	/	+	+	/	+	N	n	/	9	0	5	11
36	/	−	−	−	N	n	/	−	−	−	/	/	+	−	−	+	/	/	N	n	N	n	/	−	/	/	N	n	N	n	9	9	5	2
37	n	/	/	/	/	−	+	/	N	n	N	n	/	−	−	−	N	n	/	−	−	/	N	n	N	n	/	−	−	−	9	9	5	1
38	/	−	/	/	/	−	−	/	−	/	/	/	−	−	/	−	/	/	/	−	−	/	−	/	/	/	−	−	/	−	17	13	0	0
−:日勤	13	13	10	11	10	13	13	13	13	13	11	10	13	13	13	13	13	11	10	13	13	13	10	13	11	10	13	13	13	13				
Nn:夜勤	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6				

もその日のコンディションによってその看護の質も違ってくる．どの看護婦にとっても毎日1番よいコンディションで勤務に出てこれるようなスケジュールを組むことも看護の質を守る大きな要素といえる．これは勤務表を各行「横に見た」場合の無理のない勤務の並びということである．これを「横の条件」と呼ぶことにする（図1参照）．

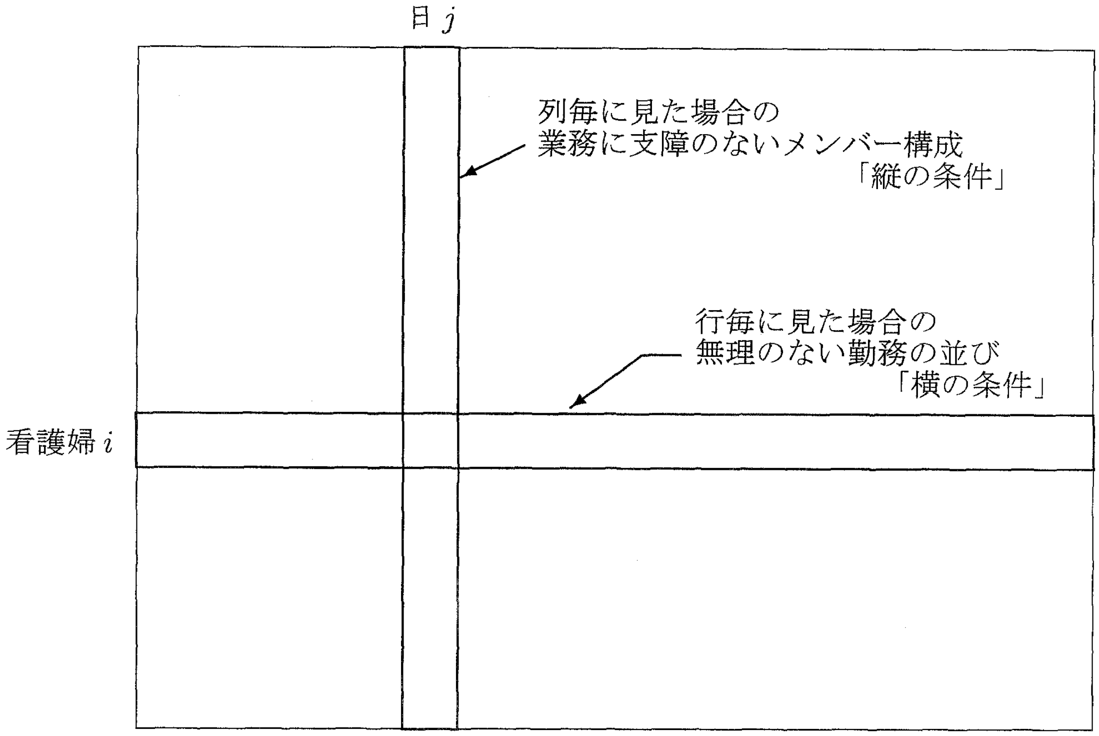

図 1: 勤務表 —— 縦の条件と横の条件

この問題を実際に解いている，つまり看護婦勤務表を作成しているのは，その部署を管理している婦長や主任である．非常に多くの時間を費やし休日のプライベート時間を利用しての作成であるが，毎日の勤務に支障をきたさないよう看護婦を組み合わせながら看護婦1人1人が無理な勤務にならないように勤務表を作成することの難しさは，これまでにも報告されている [1] [2] [3] [4].

各勤務についての看護婦メンバー構成条件が複雑になってしまうことの1つには「看護婦の早期退職者の多さ」から新人看護婦が常に数多く存在してしまうことが挙げられる．また，この退職者を減らすための工夫としての「休みや勤務の希望を受け入れる」ことも勤務表作成をより難しくしてしまう要因になっている．

この論文では，勤務表の縦の条件と横の条件を同時に満たそうとするアプローチをとらず，どちらか一方だけの条件を満たした部分解を組み合わせながら全体としての実行可能解を探していくというアプローチを提案する．具体的には，各看護婦について横の条件のみで実行可能勤務パターンを作成し，毎日の各勤務の看護婦メンバー構成（縦の条件）の実行可能具合を高めていくように組み合わせていくアプローチを示した．このアプローチは一般的なナース・スケジューリングを対象としているが，4節では，その問題の特徴により複雑なテクニックを使わなくても，提案するアプローチを適用できる「2交替制ナース・スケジューリング」を取り上げ，その特徴を利用したアルゴリズムを構築した．そして，実際の2交替制部署勤務表を作成した．

2. 扱うモデルについて

我が国のナース・スケジューリングの定式化 [2] を以下に示す．

《記号説明》
$M = \{$ 看護婦 1, 看護婦 2,..., 看護婦 $m\}$: スケジュール対象となる看護婦の集合
$N = \{1,2,\ldots,n\}$: スケジュール対象となる日の集合
$W = \{$ 勤務 1, 勤務 2,..., 勤務 $w\}$: 勤務の集合
$R = \{r | r$ は看護婦のグループ $\}$
$G_r = \{i | i$ はグループ r に所属する看護婦 $\}, r \in R$
$F_1 = \{(i,j,k), i \in M, j \in N, k \in W |$ 看護婦 i の j 日の勤務がすでに勤務 k に決定している $\}$
$F_0 = \{(i,j,k), i \in M, j \in N, k \in W |$ 看護婦 i の j 日に対して勤務 k が禁止されている $\}$
$P_h = \{(k_1, k_2, \ldots, k_h), k_1, k_2, \ldots, k_h \in W |$
勤務 $k_1, k_2, \ldots, k_h$ の連続勤務が禁止されている $\}, h \in \{2,3,\ldots\}$
$Q_h = \{(k,u,v), k \in W, u, v \in \{0,1,2,\ldots\} |$
勤務 k は, 連続する h 日間に u 回以上 v 回以下 $\}, h \in \{2,3,\ldots\}$
$d_{jk}, j \in N, k \in W$: j 日の勤務 k に必要な人数
$a_{rjk}, r \in R, j \in N, k \in W$: j 日の勤務 k に対するグループ r からの人数の下限
$b_{rjk}, r \in R, j \in N, k \in W$: j 日の勤務 k に対するグループ r からの人数の上限
$c_{ik}, i \in M, k \in W$: 看護婦 i の勤務 k に対する勤務回数の下限
$e_{ik}, i \in M, k \in W$: 看護婦 i の勤務 k に対する勤務回数の上限
$x_{ijk}, i \in M, j \in N, k \in W$: 看護婦 i の j 日の勤務を k にするとき値 1 をとり,
そうでないとき値 0 をとるような 0-1 変数
$S = \{s | s$ は達成したい条件や希望 $\}$
$f_s(x_{ijk}, i \in M, j \in N, k \in W), s \in S$: x_{ijk} の値で与えられる勤務表において
達成したい条件 s に対する未達成度 (達成目標値との差等) に
重要度の重み付けしたペナルティを与える関数

《定式化》

$$Minimize \sum_{s \in S} f_s(x_{ijk}, i \in M, j \in N, k \in W) \tag{0}$$

subject to

$$\sum_{i \in M} x_{ijk} \geq d_{jk} \qquad j \in N,\ k \in W \tag{1}$$

$$a_{rjk} \leq \sum_{i \in G_r} x_{ijk} \leq b_{rjk} \qquad r \in R,\ j \in N,\ k \in W \tag{2}$$

$$c_{ik} \leq \sum_{j \in N} x_{ijk} \leq e_{ik} \qquad i \in M,\ k \in W \tag{3}$$

$$x_{ijk} = \tau \qquad (i,j,k) \in F_\tau,\ \tau \in \{0,1\} \tag{4}$$

$$\sum_{\epsilon=1}^{h} x_{i \cdot j+\epsilon-1 \cdot k_\epsilon} \leq h-1 \qquad i \in M,\ j \in \{1,\ldots,n-h+1\},\ (k_1,k_2,\ldots,k_h) \in P_h,\ h \in \{2,3,\ldots\} \tag{5}$$

$$u \leq \sum_{\epsilon=1}^{h} x_{i \cdot j+\epsilon-1 \cdot k} \leq v \qquad i \in M,\ j \in \{1,\ldots,n-h+1\},\ (k,u,v) \in Q_h,\ h \in \{2,3,\ldots\} \tag{6}$$

$$\sum_{k \in W} x_{ijk} = 1 \qquad i \in M,\ j \in N \tag{7}$$

$$x_{ijk} = 0 or 1 \qquad i \in M,\ j \in N,\ k \in W \tag{8}$$

この定式化は, 実際の勤務表作成ではすべての拘束条件を満たすことが非常に難しいことから, その一部を達成目標として表すことを想定している [2]. つまり本来この問題は, 広

い実行可能領域において何かを最適化したいというより，存在するのであれば1つでも実行可能解を見つけたいという性質のものである．

そこで，本論文では拘束条件だけで問題を表し「実行可能解を求めることを目的とする問題」として解くことを考える．そして，すべての条件を縦の条件と横の条件に分類して扱っていく．縦と横に分類された拘束条件式の係数のマトリックスのイメージを図2に示す．

また，実際の問題を解く際に，拘束条件 (1)〜(8) では表せない条件† が存在した場合についても，それらを縦の条件と横の条件に分類して扱うことにより，次節以降のアプローチ，アルゴリズムの展開に対応することが可能である．

次節では横の条件に分類された条件を「部分問題」として定義する．

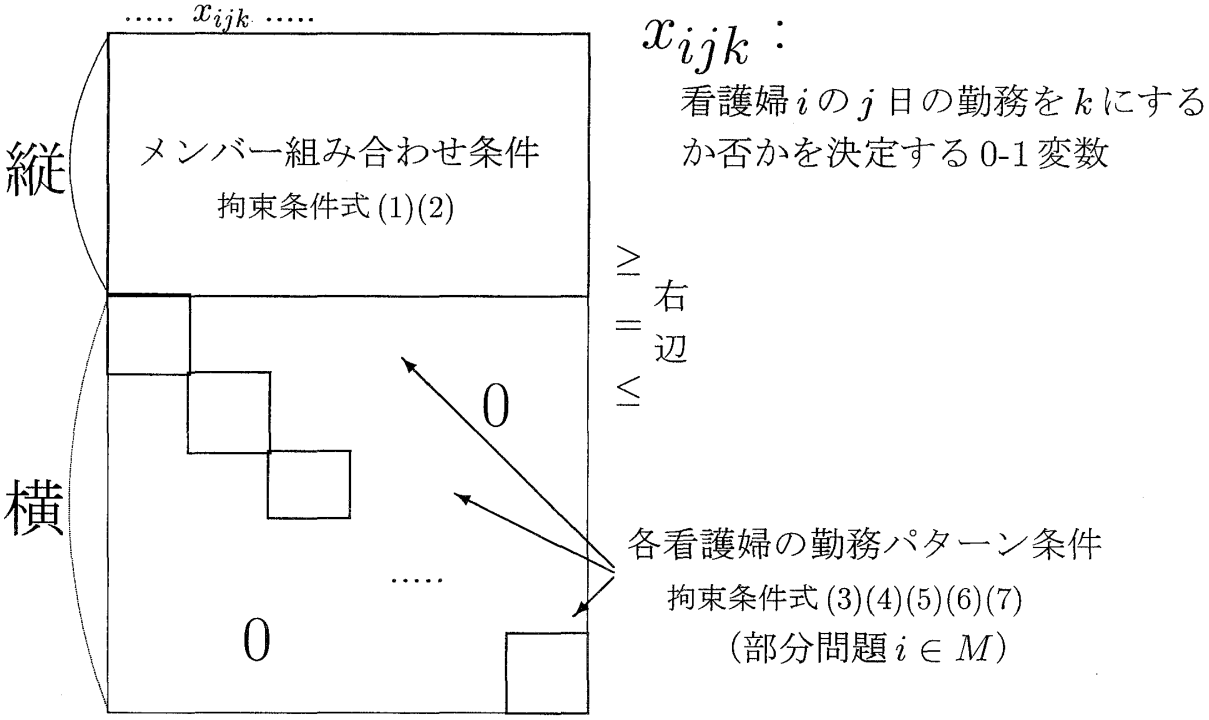

図 2: 拘束条件の係数マトリックスのイメージ

3. 実行可能勤務パターンを利用したアプローチ

勤務表作成の際，1人の看護婦の1日の勤務の決定は，その勤務の他のメンバー選択に大きな縛りを加えるとともに，その看護婦の前日や次の日との並びだけでなく前後数日にわたっての勤務の並び，そして1ヶ月を通してみた場合の勤務の数等，多くの影響を持つ．実際にできあがった勤務表を1ヶ所（1看護婦の1勤務）でも変更することは非常に困難なことであり，初めから作成し直さなくてはならない場合も多い [2]．つまり1つの勤務の決定には，勤務表の縦の条件と横の条件の両方が何重にも重なって拘わっていて，そのすべてを満たさなければ勤務表として成り立たないところにこの問題の難しさがある．

† 拘束条件 (1)〜(8) の意味は，(1) j 日の勤務 k の必要人数確保，(2) j 日の勤務 k におけるグループ r からの人数の上下限，(3) 看護婦 i の勤務 k の数の上下限，(4) 看護婦 i の j 日の勤務を k に固定 ($\tau = 1$) または k を禁止 ($\tau = 0$)，(5) 連続勤務禁止パターンを割り当てない，(6) 連続する h 日間の勤務 k の数の上下限，(7) 看護婦 i の j 日の勤務をちょうど1つ割当てる，(8) x_{ijk}は0-1変数，である．

これらで表すことのできない縦の条件としては「毎日の各勤務の合計人数の上限」「新人看護婦が入る勤務には指導担当看護婦も勤務に組み入れる」，横の条件としては「各看護婦に土曜日曜の2連休を必ず与える」「2連休の数の上下限」といったものが例に挙げられる．

そこで，我々は縦の条件と横の条件のうち，どちらか一方を意識することなく解くことを考える．ここでは，看護婦1人1人に対応した横の条件だけで構成される「部分問題」と，これらの解の結合制約として縦の条件を扱う「勤務パターン組み合わせ問題」とに問題を分けて解く．つまり，各部分問題で各看護婦の1ヶ月分の実行可能勤務パターンを作成し，それらをメンバー組み合わせ条件にあうように組み合わせるのである．

もしも各看護婦についての部分問題の実行可能解（実行可能勤務パターン）があらかじめわかっていたとしたら，以下のような定式化の変換を考えることができる．

看護婦 $i \in M$ に対して1ヶ月分の実行可能勤務パターンの集合を P_i とし，勤務パターン $q \in P_i$ を δ_{iqjk} (j 日の勤務が k であるとき1，そうでないとき0) で表現する．そして，前定式化での x_{ijk} を看護婦 i について勤務パターン q を採用するかどうかを決定する変数 λ_{iq} (勤務パターン q を採用するとき1，そうでないとき0) を使って以下のように表す．

$$x_{ijk} = \textstyle\sum_{q \in P_i} \delta_{iqjk} \lambda_{iq} \qquad i \in M,\ j \in N,\ k \in W$$

$$\textstyle\sum_{q \in P_i} \lambda_{iq} = 1 \qquad i \in M$$

前定式化 (1)～(8) にこれらを代入，追加することで変換された定式化は以下のようになる．

《実行可能勤務パターンを利用した定式化》

$$\sum_{i \in M} \sum_{q \in P_i} \delta_{iqjk} \lambda_{iq} \geq d_{jk} \qquad j \in N,\ k \in W \tag{9}$$

$$a_{rjk} \leq \sum_{i \in G_r} \sum_{q \in P_i} \delta_{iqjk} \lambda_{iq} \leq b_{rjk} \qquad r \in R,\ j \in N,\ k \in W \tag{10}$$

$$\sum_{q \in P_i} \lambda_{iq} = 1 \qquad i \in M \tag{11}$$

$$\lambda_{iq} = 0 or 1 \qquad i \in M,\ q \in P_i \tag{12}$$

非常に多かった拘束条件の数が減少し，逆に変数が増大した非常に横長のマトリックスをもつモデルとなる．拘束条件のマトリックスのイメージを図3に示しておく．

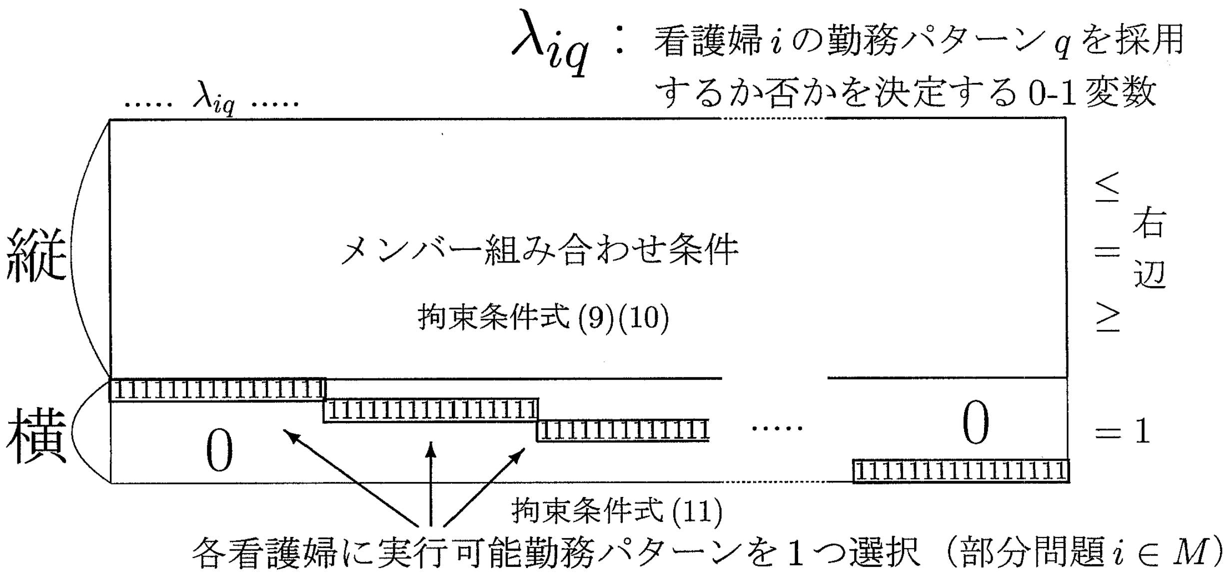

図 3: 拘束条件の係数マトリックスのイメージ

ここで，我々が提案するアプローチの考え方の核となる部分問題を定義する．

部分問題 $i \in M$：
看護婦 i 以外の看護婦のスケジュールがすべて決定している下で看護婦 i のスケジュールを最適に決定する．看護婦 i に対するすべての勤務パターン条件を拘束条件とし，目的関数は毎日の各勤務のメンバー組み合わせ条件を違反する度合いの最小化とする．

拘束条件 (9) と，拘束条件 (10) の下限値条件と上限値条件の重要度の重み付けをそれぞれ w_{jk}, u_{rjk}, v_{rjk} とし，$\alpha^-_{jk}, \alpha^+_{jk}, \beta^-_{rjk}, \beta^+_{rjk}, \gamma^-_{rjk}, \gamma^+_{rjk}$ をそれぞれの式のスラック変数，サープラス変数として，部分問題 i を以下のように定式化する．また，ここでの $\lambda_{i'q}, q \in P_{i'}, i' \in M, i' \neq i$ は 0 か 1 の定数として与えられている．

《部分問題 $i \in M$ の定式化》

$$Minimize \sum_{j\in N}\sum_{k\in W} w_{jk}\alpha^-_{jk} + \sum_{r\in G}\sum_{j\in N}\sum_{k\in W} u_{rjk}\beta^-_{rjk} + \sum_{r\in G}\sum_{j\in N}\sum_{k\in W} v_{rjk}\gamma^+_{rjk} \tag{13}$$

subject to

$$\sum_{i'\in M}\sum_{q\in P_{i'}} \delta_{i'qjk}\lambda_{i'q} + \alpha^-_{jk} - \alpha^+_{jk} = d_{jk} \qquad j \in N,\ k \in W \tag{14}$$

$$\sum_{i'\in G_r}\sum_{q\in P_{i'}} \delta_{i'qjk}\lambda_{i'q} + \beta^-_{rjk} - \beta^+_{rjk} = a_{rjk} \qquad r \in R,\ j \in N,\ k \in W \tag{15}$$

$$\sum_{i'\in G_r}\sum_{q\in P_{i'}} \delta_{i'qjk}\lambda_{i'q} + \gamma^-_{rjk} - \gamma^+_{rjk} = b_{rjk} \qquad r \in R,\ j \in N,\ k \in W \tag{16}$$

$$\sum_{q\in P_i} \lambda_{iq} = 1 \tag{17}$$

$$\lambda_{iq} = 0 or 1 \qquad q \in P_i \tag{18}$$

$$\alpha^-_{jk}, \alpha^+_{jk} \geq 0 \qquad j \in N,\ k \in W \tag{19}$$

$$\beta^-_{rjk}, \beta^+_{rjk}, \gamma^-_{rjk}, \gamma^+_{rjk} \geq 0 \qquad r \in R,\ j \in N,\ k \in W \tag{20}$$

具体的には，看護婦 i 以外全員の勤務がすべて決定している状況で

- 看護婦 i の各勤務や休みの数が許される上下限の幅におさまる
- 看護婦 i の休み希望や勤務希望その他の業務予定等を達成する
- 禁止されている勤務の並びを入れない

という条件を満たした看護婦 i 用勤務パターン $q \in P_i$ の中で，勤務表全体に対する縦の条件である

- 毎日の各勤務の人数の確保（拘束条件 (9)）
- 毎日の各勤務について業務やスキル等で規定されているグループからの人数の上下限（拘束条件 (10)）

といったメンバー組み合わせ条件を違反してしまう度合いが最も少ない（部分問題における目的関数値最小の）パターンを探すことである．

我々が提案するアプローチは，まず最初に各看護婦についてなんらかの勤務パターンを選び，「各看護婦に対応する部分問題すべてを解いて得られた実行可能勤務パターンの中で最もその目的関数値の優れていたものを選択して現在採用されている勤務パターンと交換すること」を繰り返す．この繰り返しによって全体としての実行可能解まで到達しようというものである．

4. 2交替制問題におけるアルゴリズムの構築

提案するアプローチを具体的なアルゴリズムとして実現するために，本論文では2交替制ナース・スケジューリングを対象とした．以下に2交替制勤務の特徴について説明する．

現在，病院現場では3交替制から2交替制への移行が議論されている．2交替制勤務の多くは1日をおおよそ8時間と16時間の2つに分けた勤務帯となっている．これは3交替制の準夜勤と深夜勤をいっしょにあわせて1つの夜勤としたものである．1回が2単位分にあたる長時間夜勤に対して，1ヶ月あたりの夜勤数の制約の他，連続夜勤が可能かどうか，夜勤と夜勤の間は最低何日あけるべきか等，夜勤の並び（夜勤パターン）に対する条件が厳しく与えられる．また，日勤と違って夜勤は勤務者数が少ないことや婦長が不在であることから，あらゆる事態に対応できるようにメンバーが組まれていなければならない．

逆に，日勤については勤務者数も多くメンバー組合せ条件も夜勤に比べて緩いこと，日勤と休みの並び（日勤パターン）についての条件も緩いことから，夜勤スケジュールが決定した下で日勤と休みを決定する問題は比較的解き易いものとなる．実際，勤務表作成担当者は夜勤スケジューリングが完成した時点で「勤務表作成は終わったようなもの」と思い，その夜勤にあわせて日勤と休みを，数を調整しながら決定していくという．

つまり，2交替制ナース・スケジューリングでは夜勤のスケジューリングについての拘束条件が全体問題としての解空間を絞り込んでいることから，なんらかの形で日勤メンバーを考慮しながら解くことが可能であれば，夜勤スケジューリング問題を解いて得られた解が全体としての実行可能解の部分解になる可能性が非常に高くなると考えられる．

そこで我々は，夜勤パターンに対する条件が厳しく可能パターンが列挙可能なほど絞り込まれてしまう2交替制夜勤スケジューリングにおいては，前節提案したアプローチが「部分問題を解くための特別なアルゴリズム」を必要とすることなしにそのまま（列挙と比較のみで）適用できることに着目した．そして2交替制ナース・スケジューリングに対して「夜勤スケジューリングを核として解くアルゴリズム」を考案した．

2交替制ナース・スケジューリングに対するアルゴリズムの概要

各看護婦に対して夜勤パターンに対する条件だけの部分問題を作成し，その目的関数には「夜勤メンバー構成条件を満たさない度合い」と「日勤可能な状態の看護婦が日勤メンバー構成での必要人数を下回る度合い」を設定して最小化問題とする．固定勤務として与えられている以外はなにも決定していない状態からスタートし「すべての部分問題を解いて最も目的関数が優れている夜勤パターンを現在採用されているパターンと交換すること」を目的関数値0の夜勤パターン交換が見つかるまで繰り返す．ただし1度はずされたパターンはあらかじめ設定した回数だけ交換の対象としない．

夜勤スケジュールが決定した下では，それらを固定された勤務として各看護婦に対して日勤パターンに対する条件だけの部分問題を作成し，その目的関数に「日勤メンバー構成条件を満たさない度合い」を設定して最小化問題とする．夜勤スケジューリングと同様に部分問題を解くことによって全体としての実行可能解に到達するまで勤務パターン交換を繰り返す．

このアルゴリズムでは，夜勤スケジューリングと日勤スケジューリングの2段階で提案するアプローチを適用する．各段階では実行可能勤務パターン $P_i, i \in M$ の要素と部分問題における目的関数中の重み付け係数 w_{jk}, u_{rjk}, v_{rjk} の値だけが異なる問題を解く．

まず勤務の種類を「夜勤」「日勤」という2種類‡ を設定し $W = \{night, day\}$ と表わす．

‡ 初めの定式化の拘束条件 (7) に対応するためには「夜勤入り（1日目）」「夜勤明け（2日目）」「日勤」「決定しているその他の勤務」「休み」の5種類必要である．ここではアルゴリズム説明をシンプルにするために，部分問題の拘束条件 (14)(15)(16) に関わる勤務だけ取り上げて説明するが，実際には5種類設定したものと同等に問題を解いている．（この説明においては $\sum_{k \in W} \delta_{iqjk} \leq 1, j \in N, q \in P_i, i \in M$.）

夜勤は2日にわたる勤務なので便宜上，夜勤入りの日（夜勤1日目）を「夜勤日」とよぶ．以下に，それぞれの実行可能勤務パターン $q \in P_i, i \in M$ の作成方法について説明する．

看護婦毎の実行可能勤務パターンはそれぞれの横の条件を満たしたものであるが，夜勤スケジューリングにおいては，$(i, j, k) \in F_1$ で決定されているスケジュールの下で（その他の日勤や休みについての条件を無視して）夜勤パターンを作成する．具体的には，計画期間内の夜勤回数，連続して勤務してもよい夜勤数の上下限や夜勤が終了して次の夜勤入りまでの日数の上下限を守った夜勤パターンを作成する．表2，3に夜勤パターンの例（30日に夜勤回数5回の場合の例）を示す．

表 2: 連続夜勤禁止で夜勤の間が3日以上．(Nn:夜勤,/:休み,–:日勤可能日)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
–	–	–	N	n	/	–	–	N	n	/	–	–	–	N	n	/	–	–	N	n	/	–	–	N	n	/	–	–	–

3日以上

表 3: 連続夜勤2回以下で夜勤の間が3日以上，連続夜勤の間は4日以上．

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
–	–	–	–	–	N	n	N	n	/	–	–	–	N	n	N	n	/	–	–	N	n	/	–	–	–	–	–	–	–

4日以上 3日以上

作成された実行可能夜勤パターン $q \in P_i, i \in M$ において δ_{iqjk} の値を以下のように設定する．夜勤日 j に対して $\delta_{iqj \cdot night}$ は値1，そうでない j に対して値0，そして$\delta_{iqj \cdot day}$ の値は夜勤や休み希望やその他の勤務から考えて日勤勤務が可能な日 j に対して値1，そうでない j に対し値0を設定する．したがって実行可能夜勤パターンは夜勤に関する横の条件を満足し，さらに日勤や休みに関する横の条件を無視した日勤数最大のパターンになる．

日勤スケジューリングにおける実行可能勤務パターン $q \in P_i, i \in M$ は，すでに決定された夜勤や休みを固定した下で可能な日勤と休みのパターンを作成したもの（看護婦 i の横の条件すべてを満たしたもの）である．$q \in P_i, i \in M$ における $\delta_{iqj \cdot night}, j \in N$ の値には夜勤スケジューリングで得られたパターンの値が入り，日勤日 j に対し $\delta_{iqj \cdot day} = 1$ となる．

目的関数中の係数 w_{jk}, u_{rjk}, v_{rjk} の値は，各スケジューリングにおける条件の重要度に従って設定するが，夜勤スケジューリングでは日勤人数の上限を考慮する必要がないので $v_{rj \cdot day}$ の値はすべて0とする．また，日勤スケジューリングではすでに夜勤に関する条件が満たされいるので $w_{j \cdot night}, u_{rj \cdot night}, v_{rj \cdot night}$ の値は意味をもたない（値0の項の係数）．

ここで，アルゴリズムを実際に動かすための工夫点として考えたものを以下に示す．

各看護婦の実行可能勤務パターンは各自の勤務条件やスケジュールによって異なってくるが，すべての看護婦について共通の条件（各条件の和）があらかじめ明らかである場合，対象勤務パターンを絞り込んでおくことは可能である．そこで，夜勤のパターンに関しては共通条件のみで与えられる夜勤パターンの集合を P としてアルゴリズム起動の前または初めに作成しておく（$\forall i \in M, P_i \subseteq P$）．そして，アルゴリズム起動時にこれらを入力し，個々の看護婦の条件にあわせて絞り込んで実行可能夜勤パターン集合 $P_i, i \in M$ を設定する．

次に，アルゴリズムの最初で夜勤が決定していない状況を表現するために，各看護婦に対してダミー・パターン q_0 が必要である．これに対しては，決定されている夜勤 $(i, j, night) \in F_1$ が存在した場合のみ $\delta_{iq_0j \cdot night} = 1$ とし，$\delta_{iq_0j \cdot day}$ の値は実行可能パターンと同様に設定することを考える．そして，実行可能夜勤パターン作成時にダミー・パターン q_0 も作成して P_i に加え，パターン交換でダミー・パターンがはずされると同時に P_i からもはずす．日勤実

行可能パターン作成時にも日勤用のダミー・パターンを作成して同様の扱いをする．なお，ダミー・パターンがいつまでも残ってしまう事態（必要人数の拘束条件 (14)(15) が非常に緩い場合には起こりうる）を避けるために，部分問題の目的関数値として与えられる値より十分大きな数 Big を設定し部分問題 i の目的関数に $Big \cdot \sum_{k \in W} \max\{0, c_{ik} - \sum_{j \in N} \delta_{iq_0jk}\} \cdot \lambda_{iq_0}$ の項を加えておく．アルゴリズム中の夜勤日勤両スケジューリングにおける部分問題の目的関数 (13) 式は以下のようになる．

$$Minimize \sum_{j \in N} \sum_{k \in W} w_{jk} \alpha_{jk}^- + \sum_{r \in G} \sum_{j \in N} \sum_{k \in W} u_{rjk} \beta_{rjk}^- + \sum_{r \in G} \sum_{j \in N} \sum_{k \in W} v_{rjk} \gamma_{rjk}^+ + Big \cdot \sum_{k \in W} \max\{0, c_{ik} - \sum_{j \in N} \delta_{iq_0jk}\} \cdot \lambda_{iq_0}$$

また，局所解に陥ることを避けるためにパターン交換で1度はずされたパターンについては，あらかじめ設定した回数だけ交換の対象としないことを考える．この回数を TL とよぶことにする．

アルゴリズム実行の際，部分問題 i を解いたときの目的関数値を z_i，それらの中で最小の値をそのときの「実行不可能度」 $z^*(\leq \forall z_i)$ とよぶことにし，以下にアルゴリズムの具体的手順を示す．

《手順》

0. すべての看護婦に共通な条件を満たす夜勤パターンをすべて列挙し（集合 P の作成）ファイル P に保存しておく．
1. ファイル P から 集合 P を入力し，各看護婦 $i \in M$ に与えられた条件によって実行可能夜勤パターンを選び出し（P_iの作成），$exchanged_{iq} = -\infty, q \in P_i$ と設定する．
2. 各看護婦 $i \in M$ についてダミー・パターン q_0 を作成し $P_i = P_i \cup \{q_0\}$ とする．
3. 各看護婦 $i \in M$ にダミー・パターン q_0 を割り当てる（$q^i = q_0$）．
4. $counter = 1$.
5. 各部分問題 $i \in M$ において，集合 $TABU_i = \{q | exchanged_{iq} \geq counter - TL\} \cup \{q^i\}$ を設定し，現在割り当てられているパターン q^i とすべての $q \in (P_i - TABU_i)$ を交換してみた中で目的関数値を最小にするパターンを選び，その目的関数値を z_i とする（部分問題 i に$\lambda_{iq} = 0, q \in TABU_i$ を拘束条件として加えて解く）．
6. $z^* \leq \forall z_i$ を与えた部分問題 i で選ばれたパターン q^* を現在のパターン q^i と交換する（$q' = q^i, q^i = q^*$）．$q' = q_0$ ならば $P_i = P_i - \{q_0\}$ とする．$exchanged_{iq'} = counter$．$counter = counter + 1$.
7. 実行不可能度 $z^* = 0$ ならば，現在の $q^i, i \in M$ を夜勤スケジュールとして決定する．そうでないならば手順5へ．
8. 各看護婦 $i \in M$ の夜勤スケジュールに対して実行可能日勤パターンを作成し（P_iの作成），$exchanged_{iq} = -\infty, q \in P_i$ と設定する．
9. 各看護婦 $i \in M$ についてダミー・パターン q_0 を作成し $P_i = P_i \cup \{q_0\}$ とする．
10. 各看護婦 $i \in M$ にダミー・パターン q_0 を割り当てる（$q^i = q_0$）．
11. $counter = 1$.
12. 日勤スケジューリングにおいて $z^* = 0$ になるまで手順5〜6を繰り返す．

5. 勤務表の作成

前節で紹介したアルゴリズムで，ある大学病院における1996年11月の2交替制勤務表を作成した．

対象部署は業務内容からABCの3チームに分かれており，対象看護婦は合計28人，各チームそれぞれ10人，9人，9人で構成されている．そのうち3年目以上が，それぞれ3人，

2人，2人，2年目が，それぞれ4人，5人，5人，新人が，それぞれ3人，2人，2人となっている．

すべての月のすべての看護婦に共通な夜勤パターンについての条件は以下の通りである．

- 夜勤は連続できない
- 夜勤のあとは必ず休みを1日入れる
- 夜勤と夜勤の間は3日以上あける
- 夜勤回数は0〜5回．

各看護婦についての夜勤回数の上下限は，

- 「0回」がBチーム新人に1人
- 「2〜3回」がCチーム新人に1人
- 他26人は「4〜5回」．

さらに，その月のすべての看護婦に共通な条件として「土曜日曜または日曜祭日にあたる2連休が必ず1回は保証されること」があったので，個々の実行可能夜勤パターンの集合 P_i を設定する際に，集合 P で与えられた1夜勤パターンに対し必ず1回は土日祭日にあたる2連日が連休になるように実行可能夜勤パターンを作成する．よって土日祭日の2連休可能な箇所が複数存在すればその数だけ「$\delta_{iqj \cdot night}, j \in N$ の値は同じだが $\delta_{iqj \cdot day}, j \in N$ の値が一部異なる」実行可能夜勤パターンが作成される[§]．

日勤パターンについての条件は各看護婦共通で，

- 日勤は連続3日まで
- 7日に1日は休みを入れる
- 「休み・日勤・休み・日勤・休み」のパターンを入れない
- 休みを9〜10日．

夜勤メンバー組合せ条件は，

- 必要人数は4人
- 各チームから2年目以上が1人以上
- 全体で3年目以上が1人以上

日勤メンバー組合せ条件は，

- 必要人数はその日の業務内容により異なっており，6日と19日が12〜14人，26日が12〜16人，その他の平日が10〜11人，日曜祭日は9人
- 各チームからは，6日と19日が4〜5人，26日が4〜6人，その他の平日が3〜4人，日曜祭日は3人
- 各チームから2年目以上が，26日が2〜5人，その他の日が2〜4人
- 全体で3年目以上が1人以上

前月末の勤務表と当月の勤務希望，休み希望，セミナー等固定された勤務，曜日が書き込まれた勤務表を表4に示す（各チームとも，上から3年目以上，2年目，新人の順になっている）．そして，それらを考慮して各看護婦について選択された実行可能夜勤パターンの数を表5に示す．

ここで，部分問題における目的関数中の重み付け係数 w_{jk}, u_{rjk}, v_{rjk} は，夜勤スケジューリングにおける $v_{rj \cdot day}, r \in R, j \in N$ を0に設定した以外はすべて1に設定した．1度はずしたパターンを交換の対象としない期間 TL のサイズは問題の大きさによってチューニングが必要と思われるが，この問題では30回とした．

これらの条件の下で夜勤スケジューリングした結果，全員がダミー・パターンからスタートして33回のパターン交換まで実行不可能度を増やすことなく値2まで解を改善できた．そして，115回目の交換で1個目の実行可能解が得られた．

[§] 脚注†で述べた「拘束条件 (1)〜(8) で表せない」横の条件であったので，集合 P_i 作成において対応した．

表 4: 前月末勤務表と勤務希望等 (Nn:夜勤,−:日勤,+:他勤務,/:休み,*:夜勤不可,x:日勤不可)

	看護婦番号	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		土	日	月	火	水	木	金	土	日	休	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	休	日	月	火	水	木	金	土
A	1	/	/	/	-	-	N	n	/										+															/			
A	2	/	/	-	-	-	/	/															*											/	/		*
A	3	-	N	n	/	-	-			/	/													/													
A	4	N	n	/	/	N	n	/																			/										
A	5	-	-	N	n	/	-																														
A	6	n	/	-	-	N	n	/																													
A	7	-	-	+	N	n	/							+			/	/																			
A	8	/	/	N	n	/	-						+		+															/	/	x					
A	9	N	n	/	-	-	+												+										N	n	/	/					
A	10	/	-	-	-	+	+							/																							
B	11	/	/	/	-	N	n	/											+														/	/			
B	12	N	n	/	/	/	-			/																											
B	13	/	/	N	n	/	/	/																							N	n	/				
B	14	-	-	-	N	n	/														/	/															
B	15	n	/	-	-	-	N	n	/					+													/								N	n	/
B	16	-	N	n	/	-	-											N	n	/									N	n	/						
B	17	/	/	-	N	n	/												+																		
B	18	-	-	-	-	+	-														+																
B	19	-	-	-	/	-	-			/									+		+																
C	20	-	N	n	/	/	-						/																								
C	21	-	-	-	/	-	/																														
C	22	/	/	N	n	/	-																				/										
C	23	N	n	/	-	-	-							+																/	x						
C	24	n	/	+	N	n	/							+																							
C	25	/	-	-	+	-	N	n	/				N	n	/																N	n	/				
C	26	/	/	/	/	N	n	/				N	n	+																							-
C	27	/	-	-	-	-	N	n	/							/			+		/							+									
C	28	-	N	n	/	+	-																														

表 5: 各看護婦の実行可能夜勤パターン数

Aチーム										
看護婦番号	1	2	3	4	5	6	7	8	9	10
パターン数	1965	2752	6822	4222	9164	6187	1906	1593	782	6502
Bチーム										
看護婦番号	11	12	13	14	15	16	17	18	19	—
パターン数	1949	10173	1200	4269	134	235	3951	4614	5	—
Cチーム										
看護婦番号	20	21	22	23	24	25	26	27	28	—
パターン数	9260	13150	6171	5420	3767	177	519	354	6601	—

また，実行不可能度 $z^* = 0$ となったあとも実行可能解を30個見つけるまでアルゴリズムを延長させたところ，197回目までの交換でそれらの解が得られた（ただしこれらの解には土日祭日にあたる2連休の位置が異なるだけで夜勤自体は同じパターンで与えられているものも多く含むので，異なる夜勤のパターンのスケジュールは4種類である）．全看護婦に共通な条件だけを満たす夜勤パターン（夜勤回数0〜5：6958パターン）をすべて列挙してファイル P に保存するのに4.9秒，ファイル P から夜勤パターンを入力し個々の看護婦の実行可能パターンを選び出して1個目の実行可能解を探すのに368.7秒，そして2個目から30個目までを探すのに259.8秒を要した（使用計算機は Sun SS20）．

残された日勤と休みのスケジューリングには，1個目に得られた解（夜勤スケジュール）を利用した．利用した夜勤スケジュール（確定している夜勤，休み，固定勤務）を表6に示す．また，各グループにおいて日勤可能看護婦の数が日勤必要数（下限値）と等しくなっている部分は日勤が確定するので日勤マーク「−」が書き込まれている．

これに対する各看護婦についての実行可能日勤パターンの数を表7に示す．

表 6: 夜勤スケジューリングの結果

看護婦	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	/	−	Nn	+
番号	金	土	日	休	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	休	日	月	火	水	木	金	土	休み	日勤	夜勤	ほか
1	n	/	/	−		−				−		+	N	n	/	−	−	N	n	/			N	n	/		/	N	n	/	7	13	4	1
2	/	N	n	/		−	N	n	/	/		N	n	/		−	N	n	/			N	n	/			/	/			9	11	5	0
3	N	n	/	/		N	n	/		−	N	n	/			/	/							N	n	/	−			N	7	14	5	0
4	/		−	N	n	/			N	n	/	−		N	n	/	−			/	N	n	/	/		N	n	/			8	12	5	0
5	−		N	n	/	−				−		−			N	n	/			N	n	/	/	/	N	n	/				6	16	4	0
6	/		/	/	N	n	/			N	n	/				N	n	/						−			N	n	/		7	15	4	0
7	−		−	−	/	−	+	N	n	/	/	−	N	n	/	/	/		N	n	/			−			−		N	n	7	14	4	1
8			−			+		+	N	n	/	−		N	n	/	/			N	n	/	/	/	N	n	/				7	13	4	2
9	N	n	/			N	n	/	/	/		+			N	n	/					N	n	/	/		N	n	/		8	11	5	1
10		/	/	N	n	/	/			N	n	/				−	−	N	n	/				−			−		N	n	6	16	4	0
11	/	−	−	N	n	/	−		N	n	/	+		N	n	/	−			N	n	/	/	/		/	/		N	n	9	10	5	1
12		/	/		N	n	/			N	n	/		−		N	n	/						−	N	n	/			N	6	15	5	0
13	/	N	n	/	−		N	n	/	/		N	n	/		−	N	n	/					N	n	/					7	13	5	0
14		−	N	n	/		−		/	/		−		/	/	−	−	N	n	/			N	n	/	−				N	7	16	4	0
15	n	/	−		−		+	N	n	/		−		−	N	n	/			/	N	n	/	/		−		N	n	/	7	13	4	1
16	N	n	/		−	N	n	/		−	N	n	/	−		/	/					N	n	/		−	N	n	/		7	13	5	0
17		−	−		N	n	/			−		+	N	n	/	/	/		N	n	/			−		N	n	/			6	15	4	1
18	/	N	n	/				N	n	/		−		+		N	n	/			N	n	/	/		N	n	/			7	12	5	1
19		/	/							−		+		+		−	−							−		−					2	26	0	2
20			N	n	/	/	−	N	n	/		−			N	n	/		−		N	n	/	/		N	n	/			7	13	5	0
21			−	N	n	/	−		N	n	/	−		N	n	/	/		N	n	/			−			N	n	/		6	14	5	0
22		N	n	/	−		N	n	/	/		N	n	/			N	n	/	/		N	n	/							7	13	5	0
23	N	n	/	/	−		+		−	N	n	/				N	n	/	−				/	N	n	/			N	n	6	13	5	1
24			N	n	/		+		/	/	N	n	/				−		−				N	n	/			N	n	/	6	15	4	1
25	n	/	/	−	−	N	n	/	−	−		−	N	n	/		−	N	n	/				N	n	/				N	6	14	5	0
26	/		−	−	N	n	+	/	−	−	N	n	/			/	/		−	N	n	/		−	N	n	/			−	7	14	4	1
27	n	/	/	/			N	n	/	−		+		/			N	n	/		+		N	n	/			N	n	/	8	11	4	2
28			−	−			−		/	/		N	n	/			−		N	n	/			−							4	22	2	0
−:日勤	13	13	9	12	15	14	11	14	12	9	15	10	16	12	15	11	9	16	16	13	15	16	12	9	15	15	13	15	16	16				
Nn:夜勤	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4				

日勤スケジューリングにおいては，日勤パターンを作成（全列挙）したあと，61回のパターン交換により実行可能な勤務表が得られた（10.9秒）．その結果を表8に示す．

また，30個目までの実行可能解を求めたところ，1個目の解が見つかってからのパター

表 7: 各看護婦の実行可能勤務パターン数

Aチーム										
看護婦番号	1	2	3	4	5	6	7	8	9	10
パターン数	25	10	136	37	155	75	7	44	37	81
Bチーム										
看護婦番号	11	12	13	14	15	16	17	18	19	—
パターン数	7	533	104	101	22	94	105	136	3374	—
Cチーム										
看護婦番号	20	21	22	23	24	25	26	27	28	—
パターン数	109	149	136	136	121	56	56	55	396	—

表 8: 日勤スケジューリングの結果

看護婦番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	/	−	Nn	+
	金	土	日	休	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	休	日	月	火	水	木	金	土	休み	日勤	夜勤	ほか
1	n	/	/	−	−	−	/	/	−	−	−	+	N	n	/	−	−	N	n	/	−	−	N	n	/	−	/	N	n	/	9	11	4	1
2	/	N	n	/	−	−	N	n	/	/	−	N	n	/	−	−	N	n	/	/	−	N	n	/	−	−	/	/	−	−	10	10	5	0
3	N	n	/	/	−	N	n	/	/	−	N	n	/	−	−	/	/	−	−	−	/	−	−	N	n	/	−	−	/	N	10	11	5	0
4	/	−	−	N	n	/	−	−	N	n	/	−	−	N	n	/	−	−	−	/	N	n	/	/	−	N	n	/	/	/	10	10	5	0
5	−	−	N	n	/	−	−	−	/	−	−	−	/	−	N	n	/	/	/	N	n	/	/	/	N	n	/	−	−	−	10	12	4	0
6	/	−	/	/	N	n	/	−	−	N	n	/	−	−	−	N	n	/	−	−	−	/	/	−	−	−	N	n	/	/	10	12	4	0
7	−	/	−	−	/	−	+	N	n	/	/	−	N	n	/	/	/	−	N	n	/	−	−	−	/	−	−	/	N	n	10	11	4	1
8	−	/	−	/	/	+	−	+	N	n	/	−	−	N	n	/	/	−	−	N	n	/	/	/	N	n	/	−	−	−	10	10	4	2
9	N	n	/	−	−	N	n	/	/	/	−	+	−	−	N	n	/	/	−	−	−	N	n	/	/	−	N	n	/	/	10	9	5	1
10	−	/	/	N	n	/	/	−	−	N	n	/	/	/	−	−	−	N	n	/	/	−	−	−	/	−	−	−	N	n	10	12	4	0
11	/	−	−	N	n	/	−	−	N	n	/	+	/	N	n	/	−	−	−	N	n	/	/	/	−	/	/	−	N	n	10	9	5	1
12	−	/	/	/	N	n	/	−	/	N	n	/	−	−	−	N	n	/	/	−	−	/	−	−	N	n	/	−	−	N	10	11	5	0
13	/	N	n	/	−	−	N	n	/	/	−	N	n	/	/	−	N	n	/	/	−	−	−	N	n	/	−	−	/	−	10	10	5	0
14	−	−	N	n	/	−	−	−	/	/	/	−	−	/	/	−	−	N	n	/	/	−	N	n	/	−	−	/	−	N	10	13	4	0
15	n	/	−	−	−	/	+	N	n	/	−	−	/	−	N	n	/	−	−	/	N	n	/	/	−	−	−	N	n	/	9	11	4	1
16	N	n	/	−	−	N	n	/	−	−	N	n	/	−	−	/	/	−	/	−	−	N	n	/	−	−	N	n	/	/	9	11	5	0
17	/	−	−	−	N	n	/	−	−	−	/	+	N	n	/	/	/	−	N	n	/	−	−	−	/	N	n	/	/	−	10	11	4	1
18	/	N	n	/	/	−	−	N	n	/	−	−	/	+	−	N	n	/	−	−	N	n	/	/	−	N	n	/	−	/	10	9	5	1
19	−	/	/	/	/	−	−	/	−	−	/	+	−	+	−	−	−	/	−	−	/	−	−	−	/	−	−	/	−	−	10	18	0	2
20	−	−	N	n	/	/	−	N	n	/	−	−	/	−	N	n	/	−	−	−	N	n	/	/	−	N	n	/	−	/	9	11	5	0
21	−	−	−	N	n	/	−	/	N	n	/	−	−	N	n	/	/	/	N	n	/	/	−	−	/	−	N	n	/	−	10	10	5	0
22	−	N	n	/	−	−	N	n	/	/	−	N	n	/	/	−	N	n	/	/	−	N	n	/	−	−	−	/	/	−	10	10	5	0
23	N	n	/	/	−	/	+	−	−	N	n	/	−	/	−	N	n	/	−	−	/	−	/	N	n	/	−	−	N	n	9	10	5	1
24	/	−	N	n	/	−	+	−	/	/	N	n	/	−	−	/	−	−	−	/	−	−	N	n	/	−	−	N	n	/	9	12	4	1
25	n	/	/	−	−	N	n	/	−	−	/	−	N	n	/	−	−	N	n	/	−	/	−	N	n	/	/	−	−	N	9	11	5	0
26	/	−	−	−	N	n	+	/	−	−	N	n	/	−	/	/	/	/	−	N	n	/	−	−	N	n	/	/	−	−	10	11	4	1
27	n	/	/	/	−	−	N	n	/	−	−	+	−	/	−	−	N	n	/	−	+	/	N	n	/	−	−	N	n	/	9	10	4	2
28	/	/	−	−	/	−	−	−	/	/	−	N	n	/	/	−	−	−	N	n	/	−	/	−	−	−	/	−	−	−	10	16	2	0
−:日勤	10	10	9	9	11	12	10	11	9	9	11	10	10	10	11	10	9	11	13	10	10	11	10	9	10	15	11	10	11	10				
Nn:夜勤	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4				

ン交換毎に1個ずつ得ることができた（2.5秒）．このことからもわかるように，日勤に関しては拘束条件が緩やかなことから，あとから人による手直しも可能である．

6. おわりに

本論文では，勤務表における縦の条件と横の条件を分けて解くことを考えた．そして縦の条件を反映できるような目的関数をもつ部分問題を設定し，それらを繰り返し解いて全体としての実行可能解を効率よく探すアプローチを提案した．この考えを実現させた2交替制のアルゴリズムにおいては，実際のデータに対して実行可能な勤務表を作成することができた．

2交替制問題の核となる夜勤スケジューリングについては，同じ部署の他の月の勤務表やそれらの条件を緩めたものなど複数のデータで勤務表を作成した．横の条件のきつさによって実行可能夜勤パターンの数が異なることや，各スキル・レベルの構成人数によっても縦の条件のきつさが変わってくるので解が得られるまでの時間にばらつき（数秒～30数分）はあるものの，すべての場合において与えた条件を満たす勤務表を得ることができた．このことは，全条件を満たす勤務表の作成が達成できないため休み希望をあきらめてもらったり日勤のメンバー構成等のいくつかの条件を緩めて作成している現在の状況に対して，大きく貢献できるものと考える．実用に際しては，まだ実行時間のばらつきや大きさが問題となるが，部分問題を解く際の工夫によって「時間短縮の可能性」があると思われる．

これらの実行において興味深かったのは，1つ目の解を探すのに多少時間を要しても，その周辺に実行可能解が複数存在していたことである．このことは，複数の勤務表提示の可能性を与えるが，現在与えられている勤務表に一部修正を加えた上での再スケジューリングの可能性として拡張できるようその利用を考えていきたい．

また，スケジューリングの最中にどの条件を満たせないのかを表示させたので，理屈にあわない条件（例えば，休み希望が同じ日に重なりすぎているときなど）やネックになっている条件を発見することもできた．

横の条件つまり勤務パターン作成の条件を切り出して考えることは，問題を解きやすくするばかりでなく病院特有の条件やその部署の特徴，さらに個々の看護婦の特徴等を考慮した勤務パターンを意識においたまま問題を解くことを可能にする．つまり実行可能な勤務パターンしか対象としないで問題を解いていくことは，効果的に解空間を絞り込むばかりでなく，解の構築や改善の段階においても人間が利用可能なたたき台を提供できるということである．

本論文では2交替制ナース・スケジューリング問題を対象にしたが，3交替制の問題に対しても同じ考え方で解くことが有効であると考える．ただし，3交替の場合は2交替と違って深夜勤または準夜勤だけのスケジューリングを切り出して解くことが難しい．同一勤務間における拘束条件は比較的緩いが，3種類の勤務の並び方に対する条件が厳しいからである．また，すべての勤務を含んだ実行可能勤務パターンを考えた場合，考慮すべき解空間が非常に大きくなるので部分問題を解くのに実行可能勤務パターンをすべて列挙することは不可能である．つまり，部分問題を解くためのアルゴリズムが必要になる．

今後は，提案するアプローチを3交替制の問題で実現するために部分問題を効率よく解けるアルゴリズムの開発をおこなっていきたい．また，現場の勤務表作成担当者が自分で条件を何度でも設定変更をおこないながら勤務表作成できるようなインタフェースの開発もおこなっていきたい．

謝辞

本論文について貴重な御助言を頂いた査読委員の方々，勤務表作成を御指導下さった東京女子医科大学病院松平信子婦長，山田照婦長，秋山恵美主任に心から感謝します．

参考文献

[1] 池上, 相澤, 大倉, 若狭, 松平, 越河: ナース・スケジューリング・システム構築のための基礎的調査研究. 労働科学, **71** (1995) 413-423.

[2] 池上, 丹羽, 大倉: 我が国におけるナース・スケジューリング問題. オペレーションズ・リサーチ, **41** (1996) 436-442.

[3] 竹本, 宇都, 佐々木, 櫻井, 加藤: コンピュータによる看護婦勤務表作成の隘路 1-5. 看護管理, **3** (1993) 39-42, 120-125, 171-176, 261-265, 319-325.

[4] 宮子あずさ監修: まるごと一冊勤務表. *Nursing Today*, **12** (1997).

池上敦子
成蹊大学工学部経営工学科
180-8633 武蔵野市吉祥寺北町 3-3-1
E-mail: atsuko@is.seikei.ac.jp

ABSTRACT

AN EFFICIENT APPROACH TO NURSE SCHEDULING —IMPLEMENTATION FOR A 2-SHIFT CASE—

Atsuko Ikegami Akira Niwa
Seikei University

This paper deals with the scheduling of nurses to staff shifts at a hospital. The basic difficulty is the necessity of maintaining a certain level of service and skill in the makeup of every shift, while balancing the workload among the nurses involved. As a result it is usually impossible to develop a schedule which satisfies all the requirements, in spite of the time and resources spent in the effort.

In this paper we present an efficient approach to this scheduling problem whose constraints are of block-angular structure: it consists of blocks of constraints which can be dealt with independently without a set of coupling constraints. Each of these blocks corresponds to a set of requirements for a specific nurse, and the coupling constraints are associated with the requirements in developing the overall makeup of each shift.

An objective function is first defined to measure the degree of violation caused by a schedule. We set out to optimize a problem defined by this objective function and the block of constraints for one nurse, given that the other nurses' schedules are fixed as specified in the current trial schedule. (For the first trial schedule, we used one which leaves all the nurses unassigned.) Using this trial schedule throughout, we optimize every nurse's schedule in turn by fixing those of the other nurses. Out of the resulting schedules, every one of which differs from the trial schedule only in assignments for one nurse, we choose the one with the minimal value for the objective function. This becomes the new trial solution for the next iteration, and we repeat this iterative process until a satisfactory and hopefully feasible schedule is obtained.

We have implemented this approach and constructed an algorithm for a 2-shift case. As the night shifts present more stringent and tighter constraints, if first finds a schedule to satisfy them, and then seeks a schedule to satisfy the daytime constraints. This approach is particularly effective where there are many constraints.