

2 交替制ナース・スケジューリングのアルゴリズム改善

池上敦子
成蹊大学

(受理 1999 年 4 月 9 日 ; 再受理 2000 年 1 月 13 日)

和文概要 この論文では 2 交替制ナース・スケジューリング問題に対して提案されているアルゴリズムの改善を試みる。

1998 年に提案された池上・丹羽のアルゴリズムは、看護婦の 1 ヶ月分の実行可能勤務パターンから最適なパターンを選び出すという部分問題を看護婦毎に設定し、これらを繰り返し解くことにより全体として実行可能な勤務表を作成していくものである。しかし部分問題を解くアルゴリズム自体は考えられておらず、実際には実行可能勤務パターンをすべて列挙し、その都度全勤務パターンの評価計算をして比較するというものであった。よって、効率よく実行可能解を見つけ出すと言われるものの実行可能解を与えるまでの時間に問題を抱えていた。

この論文では、部分問題を効率よく解くヒューリスティック解法を構築するために、提案されているアルゴリズムの振る舞いを観察し、解の改善過程において採用される勤務パターンの特徴を明らかにした。そして、比較計算する実行勤務パターンを有効な範囲に絞り込むことによりスピードアップを図った。改善されたアルゴリズムは提案されているアルゴリズムと同じ実行可能解を数倍から数 10 倍の速さで与えることができた。

1. はじめに

我が国におけるナース・スケジューリング問題は、1994 年のある医科大学病院における現場調査の結果 [4] を徹底的に議論することによって、1996 年モデル化された [6]。この問題では、すべての拘束条件を満たす解を得ることが非常に難しく、広い実行可能領域において何かを最適化するというより、存在するのであれば 1 つでも実行可能解を得ようとする問題とされている [6] [8]。1997 年には、このモデルの妥当性を多くの病院において検証するために全国にわたる複数病院を対象に調査がおこなわれた [5] が、勤務表作成において考慮している条件を洗い出すこの調査の結果、モデルのパラメータの値を設定し直すだけでほとんどの病院や部署の問題を扱えることが確認されている [7]。また 1998 年、池上・丹羽 [8] により、このモデルに対して有効なアプローチが提案され、これを実現した 2 交替制用アルゴリズムで実際に使用可能な勤務表が作成できたことも報告されている。

提案されているアプローチでは、膨大な量の拘束条件をもつこの問題に対して看護婦毎に「1 ヶ月分の最適実行可能勤務パターンを見つける」部分問題を設定し、これを繰り返し解くことによって勤務表全体を作成する。目的関数に全体問題の看護婦メンバー構成条件を満たさない度合最小化を組み込み、この値が最小となる部分問題を選び、その解で勤務パターン交換を繰り返している。このアプローチを具体化したアルゴリズムでは、与えられた勤務表に対する近傍を「1 看護婦の勤務パターンを他の実行可能勤務パターンと交換した勤務表」と設定してタブー探索をおこなっている。2 交替制勤務表の例 [8] を図 1 に示す。

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	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					

(-:日勤, Nn:2日間にわたる夜勤, +:その他の勤務, /:休み)

図 1: 2 交替制看護婦勤務表の例

この論文では, [8] のアルゴリズムが実行可能解を得るまでにどのような振るまいをするのかを観察し, 無駄な探索を省くことによりアルゴリズムの改善を図る.

まず, その観察をおこなうために 2 節で, これまでにナース・スケジューリング問題を解くために考えられてきたスケジューリングの単位や局所探索の近傍についてを整理し, 3 節で, 提案されているアルゴリズムの手順を簡単に紹介する. そして 4 節で, このアルゴリズムの中で繰り返し部分問題を解いていく際に実行可能性を 1 番向上させる部分問題が与える解(勤務パターン)がどんな特徴をもっているかを観察し, 5 節で, 提案されているアルゴリズムに対して, 近傍の扱い方を工夫することにより, 大きく処理時間を縮小できた実験結果を紹介する.

2. 勤務パターンの定義

ナース・スケジューリング問題は, これまでに海外でいくつかの研究がなされてきた [1] [2] [11] [12] [13] [14]. これらの研究では, 多くの看護婦が特定の時間帯の勤務のみを専門とする専従看護婦であったり, ローテーションをおこなっていても 1 種類の勤務の期間が 2 週間単位

で続くというスケジューリング単位の大きい問題を扱っている。最近では比較的短い周期でのローテーションにも対応できるモデルが提案されてきている [3] [9] [10] が、実際に解いている問題はスケジュール期間が1週間か高々2週間である。

我が国では対象看護婦のほとんどが全種類の勤務を非常に短い周期でローテーションしているため、スケジューリング単位が1日単位となり非常に小さい。そこで、この小さい単位をどう扱って問題を解くか、または、どのようにまとめて扱うのが効率良いのかを考えなければならない。

「ある看護婦のある日のある勤務に決定したもの」を1つの単位として問題を解く場合、この単位を「看護婦数×スケジュール日数」分だけ組み合わせたものが解になる。この組み合わせは、各勤務の看護のレベルを守るために与えられた条件も看護婦毎に与えられる勤務数や勤務の並びの条件も両方満たすように決定されなくてはならない。

我が国のナース・スケジューリング問題をモデル化した論文 [6] では、この単位で問題を解くことにしてシミュレーテッド・アニーリングを利用している。初期解は毎日の各勤務の合計人数のみ満たすものを用意し、近傍には「ある日のある看護婦とある看護婦の勤務を交換したもの」が設定された。このように勤務表を縦に見て、その要素を交換した「縦交換」を近傍に設定する他にも、各看護婦の勤務の数を満たした初期解を与えて「横交換」を設定する方法や、任意の状態からスタートして「ある日のある看護婦の勤務を他の勤務と交換したもの」を近傍とするもの等が容易に思いつくことができる。しかし、いずれの場合も、解を比較するために全く異なる次元の評価尺度*をすべて一緒に次元で扱わなければならない問題や、多くの条件を満たさない状態での局所解が数多く存在して探索の効率を非常に悪くしてしまうという問題を抱える。

そこで、もう少し大きい単位で扱うことを考えると、勤務表を日毎に縦に見た場合の「あるグループに所属する看護婦のみの勤務を決定したもの」「ある勤務への割当てだけを決定したもの」またはこれらを組み合わせてもっと小さい単位にしたものや、逆にそれらの枠を外して「ある日の勤務すべてを決定したもの」のようにもっと大きい単位を考えることができる。また、勤務表を看護婦毎に横に見た場合にも、期間や勤務を限るなどして同様な単位を考えることができる。極端に言ってしまうと、与えられている拘束条件の部分集合の選び方で、いくらでも単位の設定が可能である。しかし、その選び方で、それが実行可能性に向かいやすいものになるか困難なものになるかが決ってくる。

できる限り拘束条件を取り込み、それを満たす部分解が容易に得ることができ、それら部分解を縛る拘束条件が比較的緩いものが残されるように設定できれば、全体としての実行可能解が得やすくなるのは明らかである。以上の観点から、アルゴリズムで扱う単位を「各看護婦におけるスケジュール期間の長さを持つ勤務パターン(勤務の並び)」とするアプローチが提案されている [8]。ここでは、1人の看護婦に対して、この勤務パターンを選ぶ問題を「部分問題」†とよぶ。

3. 基本アルゴリズム

[8] で提案されているアルゴリズムを基本アルゴリズムとよぶことにし、この手順を紹介するために記号説明と定式化を以下に示す。この定式化は各看護婦にとっての実行可能勤務パターンの集合を意識して作成されたものである。

*例えば、ある勤務で「ベテランの人数が足りないこと」を評価するものと、ある看護婦について「休みの日数が足りないこと」や「禁止される勤務の並びができてしまった場合」を評価するもの。

†部分問題の解を縛る拘束条件は、人数の過不足に関するものだけに統一される。

記号説明

$M = \{ \text{看護婦 } 1, \text{看護婦 } 2, \dots, \text{看護婦 } m \}$: スケジュール対象となる看護婦の集合

$N = \{ 1, 2, \dots, n \}$: スケジュール対象となる日の集合

$W = \{ \text{勤務 } 1, \text{勤務 } 2, \dots, \text{勤務 } w \}$: 勤務の集合

$R = \{ r | r \text{ は看護婦のグループ} \}$

$G_r = \{ i | i \text{ はグループ } r \text{ に所属する看護婦} \}, r \in R$

$d_{jk}, j \in N, k \in W$: 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

$P_i, i \in M$: 看護婦 i の n 日分実行可能勤務パターンの集合

$\delta_{ijk}, i \in M, q \in P_i, j \in N, k \in W$: 勤務パターン q を表現する要素

(j 日の勤務が k であるとき 1, そうでないとき 0)

$\lambda_{iq}, i \in M, q \in P_i$: 看護婦 i について勤務パターン q を採用するとき値 1 をとり, そうでないとき値 0 をとるような 0-1 変数

$S = \{ s | s \text{ は達成したい条件や希望} \}$

$f_s(\lambda_{iq}, i \in M, q \in P_i), s \in S$: λ_{iq} 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

$$\text{Minimize } \sum_{s \in S} f_s(\lambda_{iq}, i \in M, q \in P_i) \quad (0)$$

subject to

$$\sum_{i \in M} \sum_{q \in P_i} \delta_{ijk} \lambda_{iq} \geq d_{jk} \quad j \in N, k \in W \quad (1)$$

$$a_{rjk} \leq \sum_{i \in G_r} \sum_{q \in P_i} \delta_{ijk} \lambda_{iq} \leq b_{rjk} \quad r \in R, j \in N, k \in W \quad (2)$$

$$\sum_{q \in P_i} \lambda_{iq} = 1 \quad i \in M \quad (3)$$

$$\lambda_{iq} = 0 \text{ or } 1 \quad i \in M, q \in P_i \quad (4)$$

各式の意味は以下の通りである。

(0) 目標値との差の合計最小化。

(1) j 日の勤務 k の必要人数を満たす。

(2) j 日の勤務 k におけるグループ r からの人数が上下限の幅におさまる。

(3) 看護婦 i に対して実行可能勤務パターンをちょうど 1 つ割当てて。

(4) λ_{iq} は 0-1 変数である。

提案されているアプローチ [8] では, この定式化に対して達成したい条件や目標をすべて拘束条件として表し ($S = \phi$) 実行可能解を求めることを目的とする問題として扱っている。そして拘束条件 (1)(2) を満たさない度合 (上下限を超してしまう値) に適当な重みをつけて足しあわせたものを「実行不可能度」とよび, 拘束条件 (3)(4) を満たした状態で, この実行不可能度を最小化するよう λ_{iq} の値を変えながら勤務パターンの交換を繰り返す。

拘束条件 (1) と, 拘束条件 (2) 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 $w_{jk}, u_{rjk}, v_{rjk} \geq 0$ とし, $\alpha_{jk}^-, \alpha_{jk}^+, \beta_{rjk}^-, \beta_{rjk}^+, \gamma_{rjk}^-, \gamma_{rjk}^+ \geq 0$ をそれぞれの式のスラック変数,

サープラス変数として以下の (5)(6)(7) 式のように書き換えると、実行不可能度は (8) 式のように表せる。

$$\sum_{i \in M} \sum_{q \in P_i} \delta_{iqjk} \lambda_{iq} + \alpha_{jk}^- - \alpha_{jk}^+ = d_{jk} \quad j \in N, k \in W \quad (5)$$

$$\sum_{i \in G_r} \sum_{q \in P_i} \delta_{iqjk} \lambda_{iq} + \beta_{rjk}^- - \beta_{rjk}^+ = a_{rjk} \quad r \in R, j \in N, k \in W \quad (6)$$

$$\sum_{i \in G_r} \sum_{q \in P_i} \delta_{iqjk} \lambda_{iq} + \gamma_{rjk}^- - \gamma_{rjk}^+ = b_{rjk} \quad r \in R, j \in N, k \in W \quad (7)$$

$$\sum_{j \in N} \sum_{k \in W} w_{jk} \alpha_{jk}^- + \sum_{r \in G} \sum_{j \in N} \sum_{k \in W} u_{rjk} \beta_{rjk}^- + \sum_{r \in G} \sum_{j \in N} \sum_{k \in W} v_{rjk} \gamma_{rjk}^+ \quad (8)$$

また、提案されている 2 交替制問題のアルゴリズムでは、条件の厳しい夜勤から決定し、その下で日勤を決定する。夜勤スケジューリングにおいては夜勤が固定された後に日勤ように残せる看護婦の数に対して下限値のみを考慮し上限値は設定しない(夜勤パターンにおいて日勤可能日を日勤として扱い実行不可能度中の係数 $v_{rj} \cdot \text{日勤} = 0, r \in R, j \in N$ とする)が、それ以外は夜勤スケジューリングも日勤スケジューリングも全く同じ考え方で解いていく。

まず、固定された休みや勤務以外は 1 つも勤務(夜勤または日勤)が入っていないダミー・パターン q_0 を看護婦毎に設定する。そして看護婦毎に与えられた実行可能(夜勤または日勤)パターンの集合 P_i の中で、与えられた勤務表に対して実行可能性に 1 番貢献するパターンを選び出す問題を看護婦 i に設定される部分問題 i と定義し、その目的関数を、 i 以外の看護婦のスケジュールは現在の勤務表のまま固定した下での実行不可能度の最小化とする。

また、このアルゴリズムは局所探索を繰り返しおこなうが、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ている。この回数を TL とよぶ。

アルゴリズム実行の際、部分問題 i の最適目的関数値を $z_i, i \in M$, それらの中で最小の値を z^* とよぶことにし、以下にアルゴリズムの手順を示す。

《基本アルゴリズムの手順》

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 $q \in (P_i \setminus TABU_i)$ 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. $z^* = \min_{i \in M} z_i$ 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 $P_i = P_i \setminus \{q_0\}$ とする. $exchanged_{iq'} = counter$. $counter = counter + 1$.

†最小値を与えた i が複数存在した場合、任意のものを選ぶ。[8] では i の値の最小のものを選んだ。一般に勤務表において看護婦の並ぶ順序は上からチーム別にベテランから新人へととなっている。

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. 日勤スケジュールリングにおいて $z^* = 0$ になるまで手順 5 ~ 6 を繰り返す.

ここで, 夜勤スケジュールリングにおける実行可能夜勤パターン $q \in P_i$ 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 P から選び出し, セミナーや休み希望などに不都合なく夜勤が入っているパターンであれば, 夜勤以外の日が日勤可能であるかどうかの情報を書込んで作成される. 実行可能夜勤パターンの例 [8] を図 2 に示す.

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

図 2: 実行可能夜勤パターンの例 (Nn: 夜勤, /: 休み, -: 日勤可能日)

この例は「夜勤数 5 回, 連続夜勤が禁止, 夜勤の翌日は必ず休み, 夜勤と夜勤の間が 3 日以上」といった条件を満たしたパターンを P から選び出し, その他の日 (N, n, / 以外の日) が日勤可能であれば日勤 (日勤可能日) の記号を入れたものである.

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 (パターン上 N で表現する日) j に対して $\delta_{iqj, \text{夜勤}}$ は値 1, そうでない j に対して値 0, そして $\delta_{iqj, \text{日勤}}$ 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.

この論文では, 4 ~ 5 節の議論に入る前に, [8] で実際に問題を解いた際の $q \in P_i, i \in M$ の生成方法に一部修正を加えておく. [8] では各看護婦に対して「土日休日にあたる 2 連休を確実にする」ために上記のように生成されたパターンそれぞれを土日 2 連休を確定した複数のパターンに増幅している. しかし, すでに休み希望などから土日休日の 2 連休が確定されている看護婦についてはパターン増幅が不要であるにもかかわらず全看護婦に対しパターン増幅をおこなっていたので, この場合のパターン増幅をおこなわないよう生成方法を修正した. パターン増幅の例を図 3 に示す. また, この修正により, [8] の夜勤スケジュールリングの段階で得られたもの (図 1 参照) と一部異なる解が得られたので新しい解を図 4 に示す.

図 4 では夜勤が決定している日以外にも, 休み「/」やセミナーなどの勤務「+」が確定している日, また日勤可能な看護婦の数が各チームやスキルレベルでの日勤必要数と等しくなっている日には日勤「-」が確定するので, それぞれのマークが書き込まれている. また合計欄の日勤部分には日勤可能な数が入っている. [8] ではパターン交換の手順を 115 回繰り返すことにより解を得たが, 図 4 の解は 47 回で得ることができた. 9 人 (看護婦 1, 4, 5, 6, 8, 9, 11, 12, 18, 28) に夜勤位置や回数の変更が見られたが, すべての条件を満たした勤務表が得られるという点で同等な解になっている. よって, この新しい解を基本アルゴリズムで解いた解として 4 ~ 5 節の議論を進める.

(Nn:夜勤,/:休み,-:日勤可能日)

パターン増幅する例 :

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
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
-	N	n	/	-	-	N	n	/	-	-	-	-	-	N	n	/	-	-	N	n	/	-	-	N	n	/	-	-	-



2つのパターンに増幅

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
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
-	N	n	/	-	-	N	n	/	/	-	-	-	-	N	n	/	-	-	N	n	/	-	-	N	n	/	-	-	-

2 連休確定

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
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
-	N	n	/	-	-	N	n	/	-	-	-	-	-	N	n	/	-	-	N	n	/	/	/	N	n	/	-	-	-

2 連休確定

パターン増幅しない例 :

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
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
-	N	n	/	-	-	N	n	/	/	-	-	-	-	N	n	/	-	-	N	n	/	-	-	N	n	/	-	-	-

休み希望



パターン増幅しない(そのまま登録)

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
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
-	N	n	/	-	-	N	n	/	/	-	-	-	-	N	n	/	-	-	N	n	/	-	-	N	n	/	-	-	-

図 3: パターン増幅の例

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	/	-	-	N	n	/	-	-	N	n	/	8	10	5	1	
2	/	N	n	/	/	-	N	n	/	/	-	N	n	/	/	-	N	n	/	/	-	N	n	/	/	-	N	n	/	/	-	N	n	/	9	11	5	0	
3	N	n	/	/	-	N	n	/	/	-	N	n	/	/	-	/	-	/	-	-	-	-	N	n	/	-	-	N	n	/	-	-	N	n	/	6	15	5	0
4	/	/	/	/	-	-	N	n	/	-	-	N	n	/	-	-	/	-	-	/	N	n	/	-	-	N	n	/	-	-	N	n	/	8	14	4	0		
5	-	N	n	/	-	-	N	n	/	/	-	-	-	-	N	n	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	13	5	0	
6	/	-	N	n	/	-	-	N	n	/	/	-	-	-	N	n	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	15	4	0	
7	-	-	-	/	-	+	N	n	/	/	-	N	n	/	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	14	4	1	
8		N	n	/	+	+	N	n	/	-	-	N	n	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	11	5	2	
9		/	/	-	N	n	/	/	/	+	-	-	-	-	N	n	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	14	4	1	
10			-	N	n	/	/	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	16	4	0	
11	/	-	N	n	/	/	-	-	N	n	/	+	-	-	N	n	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9	10	5	1	
12	N	n	/	/	/	/	/	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	17	4	0	
13	/	N	n	/	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	13	5	0	
14	-	N	n	/	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	16	4	0	
15	n	/	/	/	/	+	N	n	/	-	-	-	-	-	N	n	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8	12	4	1	
16	N	n	/	-	-	N	n	/	-	-	N	n	/	-	-	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	13	5	0
17	-	-	-	-	N	n	/	/	/	/	+	N	n	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	15	4	1
18	/	-	-	-	N	n	/	/	-	-	-	+	-	-	N	n	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	15	4	1
19	-	/	/	/	/	/	/	/	/	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	25	0	2	
20			N	n	/	/	-	N	n	/	-	-	-	-	N	n	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	13	5	0
21			-	N	n	/	-	-	N	n	/	-	-	-	N	n	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	14	5	0
22			N	n	/	-	-	N	n	/	/	-	-	-	N	n	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	13	5	0
23	N	n	/	/	-	+	-	-	N	n	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	13	5	1
24			N	n	/	+	/	/	N	n	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	15	4	1
25	n	/	/	-	-	N	n	/	-	-	-	-	-	-	N	n	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	14	5	0
26	/	-	-	-	N	n	+	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	14	4	1
27	n	/	/	/	/	-	-	N	n	/	-	+	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8	11	4	2
28			-	-	-	-	/	/	/	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	21	3	0
-:日勤	13	15	9	10	15	14	11	14	12	9	15	10	16	12	15	11	9	16	16	13	15	16	11	10	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	4	4	4	4	4	4	4		

(-:日勤, Nn:2 日間にわたる夜勤, +:その他の勤務, /:休み)

図 4: 夜勤スケジュールリングの結果

4. 有効な勤務パターンの絞り込み

1 人の看護婦に着目した場合, 他の看護婦の勤務パターンの組まれ方によって, どの実行可能勤務パターンが実行不可能度を減らせるかが異なってくるので, 基本アルゴリズムでは勤務パターンが採用される度に各看護婦に対応する部分問題を解いて次の採用勤務パターンを決めている. また, その部分問題を解く際には, 対応する看護婦の実行可能勤務パターン $q \in P_i$ の目的関数値 (実行不可能度) をすべてを計算して, その値の比較によって選択をおこなっている. よって1つのパターン交換の際には, およそ $\sum_{i \in M} |P_i|$ 回の比較計算がおこなわれていることになる. 基本アルゴリズムでは, この比較計算において現在採用されているパターン q^i と評価するパターン q の異なる部分の「勤務が変わることによる目的関数値の変化」のみを計算しているが, その計算時間の合計は解を得るまでの時間の大きな割合を占めている. [8] の問題を解く際には約 97% を占めた.

しかし, 対象となる実行可能勤務パターンの中にも, 全体の実行可能性にとって, よい方

向を持つものと全く反対の方向を持つものが存在するはずである。そこで、どんな勤務パターンがよい方向を持ち得るのかを、実行可能パターン数の多い夜勤スケジューリングにおいて考察する。

ここで、よい方向をもつパターンとは「現在満たしている条件を守ったまま満たしていない条件を改良できる（実行不可能度を減らせる）パターン」と考えると、現在満たされていない条件の1つについて改良できるパターンが存在するならば、それは現在採用されているパターンと高々[§]1ヶ所だけ夜勤位置が異なる（夜勤が増えたり減ったりしている）パターンである。これは「夜勤の合計人数が、ある数ちょうどであるべき」というナース・スケジューリング問題特有の条件から与えられる特徴であり、他のどの夜勤が増えたり減ったりしても必ず実行不可能度に影響が及ぶからである。そして複数ヶ所いっぺんに満たすパターンが存在したとすれば、高々[¶]その数だけ夜勤位置が異なるパターンということになる。

よって、現在採用されている勤務パターンと高々「改良したい条件の数」だけ夜勤の位置が異なるパターンのみを交換の対象として比較計算することが有効であろうと考えた。そして看護婦数分の実行可能勤務パターンが採用されて解が構築された後の各イテレーションにおいては、すでに満たしていない条件数は少なくなっているはずなので、ほとんど夜勤位置がかわらないパターンがその対象になると思われる。

そこで基本アルゴリズムのパターン交換において上記のようなパターンが実際に選択されているかどうかを調べた。パターン交換がおこなわれた2つのパターン（基本アルゴリズムの手順6における現在の q^i と選ばれた q^* ）の夜勤の日の位置の比較をおこなうために、2つのパターンの一方が夜勤であるのもう一方がそうではない日の数を夜勤パターン間の「差」と定義した。パターン q とパターン q' の差を以下の式で与える。

$$\sum_{j=1}^n |\delta_{iqjk} - \delta_{i'q'jk}| \quad (9)$$

夜勤回数が同一であるときのパターン間の差は偶数(0,2,4,...)となるが、1人の看護婦に与えられる夜勤回数の条件には幅(上下限)がある場合が多いので、夜勤回数が1回違いの場合は奇数(1,3,5,...)の差、2回違いの場合には再び偶数(2,4,6,...)の差、というようになる。

調べる対象としては、実際の病院の2交替制部署28人[8]の2ヵ月分のデータにおける夜勤スケジューリングを取り上げた。データ1は[8]で扱った1996年11月の問題(図4)、そしてデータ2は同じ部署の翌年3月の問題である。それぞれの夜勤スケジューリングにおける勤務(夜勤)パターン交換の過程を調べた。基本アルゴリズムでは、データ1に対して47回、データ2に対して972回のパターン交換で解を得ることができた(実行不可能度の重み付け係数 $w_{jk}, u_{rjk}, v_{rjk}, r \in R, j \in N, k \in \{\text{日勤, 夜勤}\}$ は[8]と同様に v_{rj} 日勤, $r \in R, j \in N$ を0とした以外はすべて1に設定)。

各看護婦の実行可能夜勤パターンの数(P_i の要素数)は表1, 2の通りである。パターンの増幅方法の変更のため、データ1の看護婦3と8のパターン数が[8]のものとは異なっている。

そこで、解の構築(看護婦数分の実行可能勤務パターンが採用された)後、パターン交換された2つのパターンの差がいくつであったのかを調べ、表3にまとめた。ここで、各看護婦の

[§]夜勤位置が同じでもパターン増幅等により休みの位置が異なるパターンが存在するならば、その差が日勤可能看護婦の人数に影響する場合がある。

[¶]さらに現在採用されているパターンに1ヶ所夜勤が増えたり減ったりした場合に複数の現在満たしていない条件を満たす場合があるからである。例えば、ある日の夜勤の全体人数とグループに必要な人数の両方を満たしていない状況で、そのグループに属している看護婦の「現在採用されているパターンにその日の夜勤をつけ加えた」パターンを採用した場合。

実行可能パターン中の夜勤の数は標準で4回もしくは5回だが、データ1では2回もしくは3回が1人、0回が1人、データ2では3回もしくは4回が4人、0回1人、4回1人、5回5人が存在する。

表 1: 各看護婦の実行可能夜勤パターン数 (データ 1)

看護婦番号	1	2	3	4	5	6	7	8	9	10
パターン数	1965	2752	1925	4222	9164	6187	1906	710	782	6502
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	—
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	—

表 2: 各看護婦の実行可能夜勤パターン数 (データ 2)

看護婦番号	1	2	3	4	5	6	7	8	9	10
パターン数	124	286	1233	4694	3093	3087	2362	247	452	3894
看護婦番号	11	12	13	14	15	16	17	18	19	—
パターン数	407	219	4734	211	780	1537	466	2030	1	—
看護婦番号	20	21	22	23	24	25	26	27	28	—
パターン数	283	28	190	9854	30	6480	766	19140	408	—

表 3: パターン交換されたパターン間の「差」の分布

差	データ 1	データ 2
(解構築イテレーション)	28	28
0	11	816
1	7	87
2	0	41
3	1	0
4 ~ 10	0	0
合計	47	972

この結果を見ると、パターン交換の多くが差0のパターン間での交換だったことがわかる。差0のパターンとは夜勤の位置が全く同じで土日休日2連休の位置だけが異なるパターンのことなので、この交換は日勤に残せる人数の調整だけをおこなっていることになる。また、それ以外の場合でも共通部分を多くもつ(差が1か2)パターンの間で交換が起きている。

次に、実行可能パターンの中で、差の値の分布がどのようになっているかを調べてみた。ここでは、データ1において実行可能夜勤パターンの数が標準的な看護婦17の3951個の実行可能パターンを例にとり、イテレーション29(人数分の実行可能パターンを選択できた直後)において採択されていたパターンに対する残りの3950個のパターンの差の値の分布の累積グラフを図5に示す。

この例では、差3までのパターンの数89は全体の2%である。逆に言うと、パターンの比較計算の98%が無駄に終わっていたことになる。

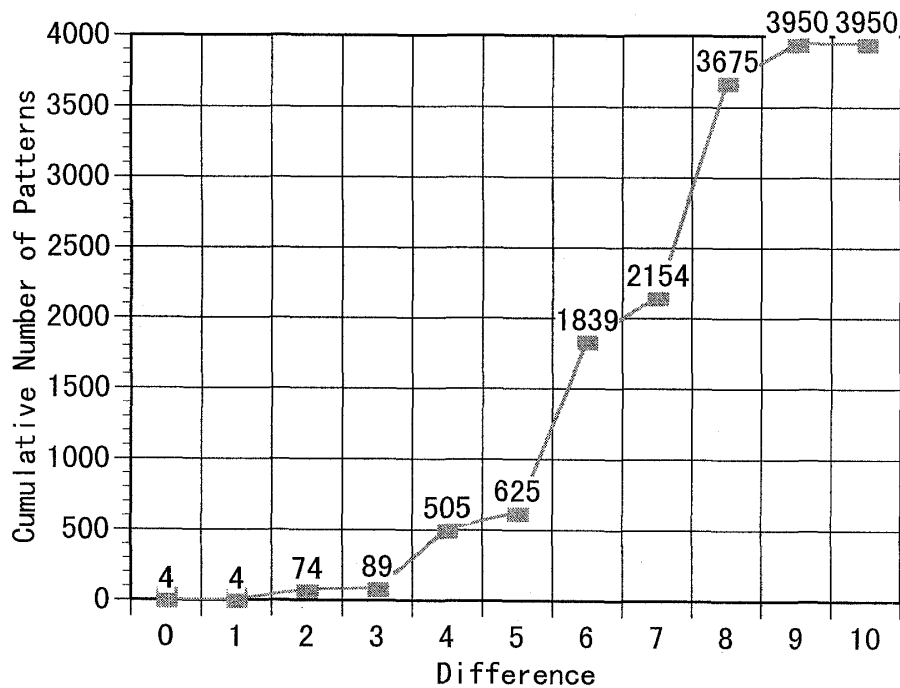


図 5: 差の値によるパターンの分布の例

(データ 1, イテレーション 29, 看護婦 17 の採用パターンに対するその他のパターンの分布)

そこで、次節では、アルゴリズムにおいて勤務パターンの交換対象を現在採用されている勤務パターンと差が少ないものだけに絞った場合、どの程度スピードアップが図れるのかを実験する。部分問題を解く際に、提案されているアルゴリズムがすべての解（勤務パターン）の目的関数値を計算比較することによって部分問題の最適解を求めているのに対し、この交換対象の絞り込みは部分問題を解くためのヒューリスティック解法といえる。

5. 計算実験結果

変更を加えられた新しいアルゴリズムの夜勤スケジューリング部分を示す。パターン交換の対象を絞り込む範囲を「現在採用されているパターンとの差」で規定し、その差の値を DIFF とよぶことにする。また、看護婦 i の実行可能夜勤パターンの集合 P_i の中から現在採用されているパターン q^i と DIFF の値から絞り込まれるパターン交換対象集合を \tilde{P}_i とし、実際にパターン交換がおきた看護婦について、その都度あらたに作成するものとする。

$k =$ 夜勤として、集合 \tilde{P}_i の定義を以下に示す。

$$\tilde{P}_i = \left\{ q \mid \sum_{j=1}^n |\delta_{iqjk} - \delta_{i^qjk}| \leq DIFF, q \in P_i \right\} \quad (10)$$

ここでは、すべての $q \in P_i$ を調べて q^i との差が DIFF 以下であるパターンを選び出して \tilde{P}_i を作成することにする。

また、実行可能解が短時間で得ることができなかった場合の終了条件として、イテレーション数の上限 ITE を導入し、ここでは $ITE=100000$ と設定した。以下に手順を示す。

《改善アルゴリズムの手順》

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 を作成して割り当てる ($q^i = q_0$).
3. 各 $i \in M$ について $\tilde{P}_i = P_i$. $counter = 1$.
4. 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 $q \in (\tilde{P}_i \setminus TABU_i)$ 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.
5. $z^* = \min_{i \in M} z_i$ 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 ($q' = q^i, q^i = q^*$). そして \tilde{P}_i を作成し直す ($\tilde{P}_i = \{q | \sum_{j=1}^n |\delta_{iqjk} - \delta_{iq'jk}| \leq DIFF, q \in P_i\}$). $exchanged_{iq'} = counter$. $counter = counter + 1$.
6. 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, $counter \geq ITE$ の場合には終了し, それ以外の場合は手順 4 へ.

以上に従い, DIFF の値を 1 ~ 10 に設定して前節で調べた夜勤スケジュールリング問題を解いた. 扱う夜勤パターン中の夜勤の数の最大が 5 であることから, パターン間の差は高々 10 である. よって, このアルゴリズムを DIFF=10 に設定して解くことは, 3 節で紹介した基本アルゴリズムで解くことに等しい. これらの設定によって実行可能解が得られるまでの時間を比較した結果を DIFF=1,2,3,10 の場合について表 4 に, DIFF=1 ~ 10 についてのグラフを図 6 に示す. 使用計算機は GATEWAY G6-266(OS:FreeBSD2.2.2) である.

表 4: パターン交換の対象の違い (DIFF=1,2,3,10) による実行時間の比較
(かっこ内は実行可能解を得るまでにおこなわれたパターン交換の回数)

DIFF	データ 1	データ 2
1	10.8 秒 (78)	——
2	10.8 秒 (34)	18.0 秒 (972)
3	11.5 秒 (47)	31.4 秒 (972)
10(全て)	71.3 秒 (47)	500.7 秒 (972)

データ 1 に関しては DIFF が 3 以上, データ 2 に関しては DIFF が 2 以上に設定された場合には同じパターン交換過程となり全く同じ解が得られるが, 比較計算の量が異なってくるので DIFF の値によって実行時間も大きく異なる. 基本アルゴリズムと同じ解をデータ 1 では約 6 倍, データ 2 では約 28 倍の速さで解くことができている.

DIFF の値を, さらに小さく DIFF=1 と設定し, データ 1 のように各イテレーションでの計算時間を減らすかわりにパターン交換回数を増やす可能性をもたせることで, 高速化を図ることも考えられる. しかし, 逆に実行可能解に到達できない状況に陥る危険性もでてくる. データ 2 を DIFF=1 で解いた場合には, イテレーション (パターン交換)100000 回以内で実行可能解を得ることができなかった (パターン交換 100000 回の時点で $z^* = 7$). これは, 差 1 のパターン (夜勤が 1ヶ所増えたり減ったりしたもの) というのは, 夜勤回数に許される幅が小さい場合や休み希望やセミナー等が数多く設定されている状況では, 存在しにくいからである. 例えば, 夜勤回数が「ある数ちょうど」であるべき (データ 2 では 6 人存在) ならば, 差 1 の

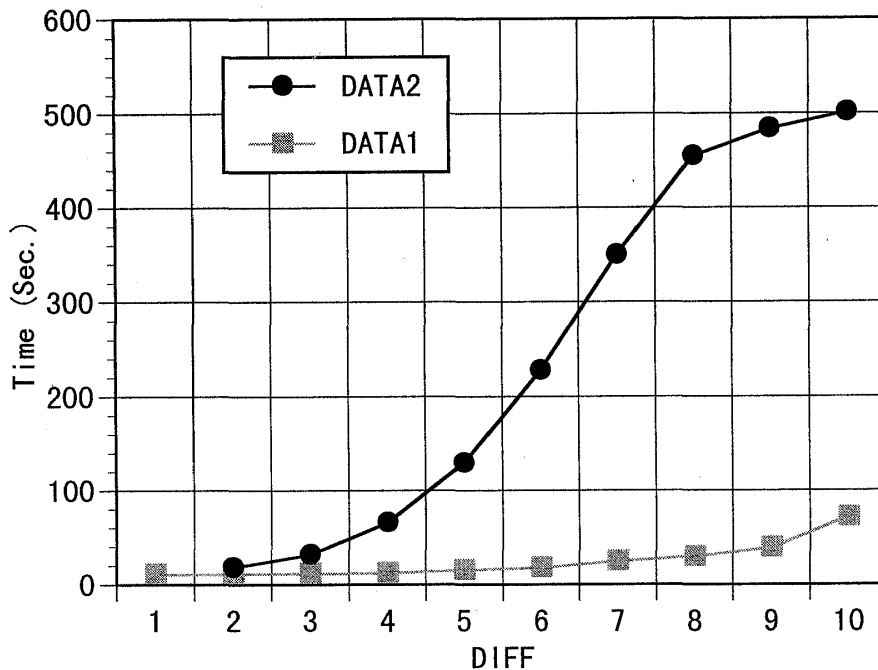


図 6: パターン交換対象の違い (DIFF=1 ~ 10) による実行時間の比較

パターンは存在しないので、差 0 のパターンとの交換だけとなり夜勤位置が初めに選ばれたパターンのものに固定されてしまう。また、1 より大きい奇数差のパターンも夜勤回数が異なるので、パターンの数は少ない (図 5 参照)。よって、これらの「差の値によるパターンの分布」の特徴と、以上の調査・実験結果から、DIFF の値を 2 ~ 3 に設定することは有効であると考えられる。

また、データ 1 で DIFF を 1 や 2 に設定した場合の解は、それぞれ基本アルゴリズムと異なっていたが、夜勤に対する条件のすべてを満たすという点では同等なものであった。ただし DIFF を 2 に設定した場合の解は、日勤スケジューリングにおいて「日勤は連続 3 日まで」という条件を満たせない看護婦がでてしまったので、夜勤スケジューリングで複数個得た解の 10 番目のものを使って勤務表を完成させることになった。ナース・スケジューリングにおいて日勤に対する条件は比較的緩いが、このような状況に陥る可能性は基本アルゴリズムにおいても改善アルゴリズムのどんな DIFF の値を設定したものにおいても同じように残されている。これらを回避する手段としては、上記で述べたように、夜勤スケジューリングにおいて得られる複数の解を利用することが考えられる。アルゴリズムは 2 番目以降の解を非常に速く得ることができる [8] からである。データ 1 で DIFF を 2 に設定した場合には、2 ~ 10 番目の解を 0.1 秒、11 ~ 30 番目の解を 0.5 秒で得ている。

また、各イテレーションで比較計算されたパターンの数を DIFF=1,3,5,7,10 について表したものをデータ 1 について図 7、データ 2 について図 8 に示す。これらの数が実行時間に直接影響しているといえる。

6. おわりに

この論文では、ナース・スケジューリング問題を解くアルゴリズムで扱う部分解の単位やそこから考えられる解の近傍、そして局所探索をおこなう際の近傍の扱いについて述べた。

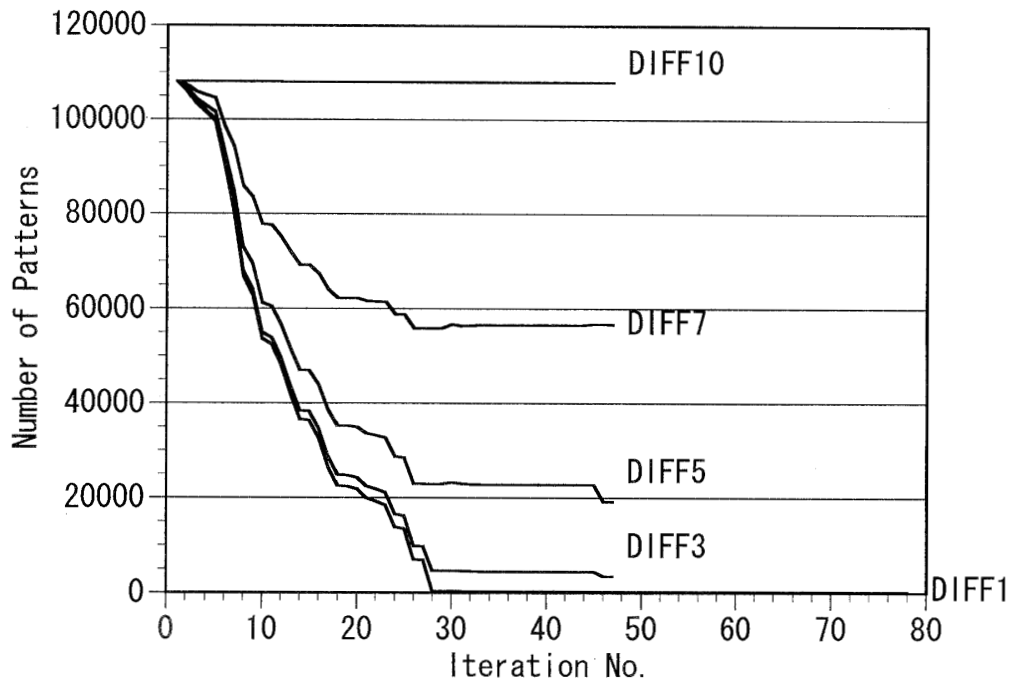


図 7: DIFF の値によるパターン比較計算量の比較 (データ 1)

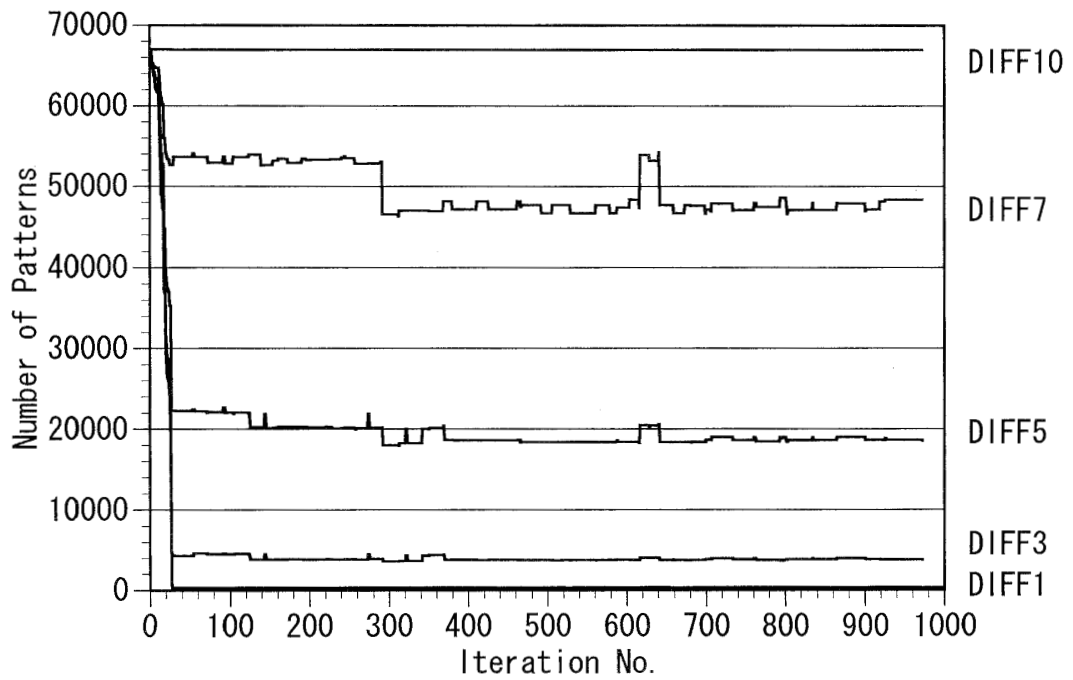


図 8: DIFF の値によるパターン比較計算量の比較 (データ 2)

扱ったアルゴリズムは、1人分の看護婦の勤務パターンを部分問題の単位として各日各勤務の実行可能性を最も向上させるように勤務パターンを選んでいく。この実行可能性は各勤務のメンバー構成に依存するものであり、具体的にはベテラン看護婦の確保ということになる。よって解構築のフェーズでは、各勤務に必要なベテラン看護婦ができる限り配置されるようなパターン選択がおこなわれていく。この際、ベテラン看護婦の数がぎりぎり少ない数(彼女達の勤務可能日数の合計がその月に必要な「のべ人数」に近い値)であっても、全ての日が勤務可能であれば実行可能勤務パターンが数多く存在するので、比較的簡単に実行可能解を得ることができる。しかし現実には、会議やセミナー等、日常業務以外の勤務の他にも、それぞれが希望する休み等が存在するので、実行可能解を見つけることが困難になってくる。極端な例としては、同じ日に休み希望を出す看護婦の数がある数を超した場合には、それだけで実行不可能な問題となる。

しかし、そのような「明らかに実行不可能と判断できる状況」を排除した上で勤務表を作成する際には、実行可能解がなければ、ネックになっている日または看護婦を明らかにする必要があり、もしも1つでも実行可能解が存在するのであれば、数多くのパターン交換をおこなってでも見つけ出さなければならない。池上・丹羽 [8] のアルゴリズムは、理屈にあわない条件やネックになっている条件を知ることが可能であるが、解を得るまでの時間に問題を残していた。そこで本論文では、短い時間内で最大限のパターン交換が可能となる改善をおこなった。具体的には、定義した近傍の中に実行可能性にとって有効なパターンとそうでないものが存在することから、勤務パターン交換の対象を絞り込んだ。解の近傍の中のすべてのパターンを比較評価して次に交換するパターンを選んでくるやり方に対して、この絞り込みはアルゴリズムの実行速度を数倍から数10倍速くする。これは実際に利用できる看護婦勤務表作成ソフトの実現に大きく貢献するものと考えられる。

また、改善アルゴリズムの \tilde{P}_i の生成では、新しく q^i が採用された際に、すでに列挙されているすべての実行可能パターン $q \in P_i$ との差を調べて選び出しているが、与えられた勤務パターン q^i と DIFF の値だけで \tilde{P}_i を生成できる簡便な方法¹¹で置き換えることにより更に速度向上が可能と思われる。

今後の課題としては、3交替制の問題が挙げられる。3交替制の問題においては、同一勤務間における拘束条件は比較的緩いが異種勤務の並びについての条件が厳しいため、勤務の次元で問題を切り分けることが難しい。しかし、日勤、準夜勤、深夜勤すべてが埋まった実行可能勤務パターンを看護婦毎にすべて列挙して扱うことは、そのパターンの数が膨大な数となるため現実的には不可能である。よって、これらを列挙せずに扱うためには、看護婦毎に与えられる部分問題(全体問題の実行可能性を向上させるような実行可能勤務パターンを選ぶ問題)を解くヒューリスティック解法が必要となる。本論文で提案した「アルゴリズムの改善」は、このようなヒューリスティック解法の構築にも応用できると考える。今後は、提案されているアプローチにのっとった3交替制問題用のアルゴリズムを開発していくために「勤務パターン交換の交換対象の絞り込み」のための「勤務パターン間の差」の設定方法や初期解を得るための方法について研究を進めていきたい。

謝辞

貴重な御助言を頂いた査読委員の先生方に心より感謝します。

¹¹本論文では利用していないが DIFF の値が小さいときには簡単な「夜勤位置の移動」で可能と思われる。

参考文献

- [1] J.Arther and A.Ravindran: A multiple objective nurse scheduling model. *AIIE transactions*, **13**(1981)55-60.
- [2] P.Bell,G.Hay and Y.Liang: A visual interactive decision support system for workforce (nurse) scheduling. *INFOR*, **24**(1986)134-145.
- [3] K.Dowland: Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research*, **106**(1998)393-407.
- [4] 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.
- [5] 池上, 越河: 看護婦勤務表作成におけるアンケート調査. 私立医科大学病院看護部長会 (総婦長会) 調査報告書資料 (1997).
- [6] 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.
- [7] 池上, 丹羽: ナース・スケジューリング・モデルの補足. *OR 学会春季研究発表会アブストラクト集* (1998)52-53.
- [8] 池上, 丹羽: 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. *Journal of Operations Research Society of Japan*, **41**(1998)572-588.
- [9] B.Jaumard,F.Semet and T.Vovor: A generalized linear programming model for nurse scheduling. *European Journal of Operational Research*, **107**(1998)1-18.
- [10] H.Millar and M.Kiragu: Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. *European Journal of Operational Research*, **104**(1998)582-592.
- [11] H.Miller,W.Pierskalla and G.Rath: Nurse scheduling using mathematical programming. *Operations Research*, **24**(1976)857-870.
- [12] I.Ozkarahan and J.Bailey: Goal programming model subsystem of a flexible nurse scheduling support system. *IIE Transactions*, **20**(1988)306-316.
- [13] S.Randhawa and D.Sitompul: A heuristic-based computerized nurse scheduling system. *Computer & Operations Research*, **20**(1993)837-844.
- [14] M.Warner: Scheduling nursing personnel according to nursing preference:A mathematical programming approach. *Operations Research*, **24**(1976)842-856.

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

IMPROVEMENT ON THE 2-SHIFT NURSE SCHEDULING ALGORITHM

Atsuko Ikegami
Seikei University

This paper proposes an improvement of the algorithm by Ikegami and Niwa (1998) for 2-shift nurse scheduling.

In their algorithm, we prepare beforehand a set of all the schedules for each nurse which specify which days she or he is feasibly assigned to the nighttime shift with an extra information that the nurse can or cannot work for the remaining days. Also we define the objective function to measure the degree of violations of the nighttime-related constraints which are caused by a solution.

Then we set out to compare all the prepared solutions for each nurse in terms of the value of the objective function given that the other nurses' schedules are fixed as specified by the current trial solution. Finally we choose the best solution from these individual best ones. This solution in turn will be used as the next trial solution to define a new optimization problem to be solved and so on.

This creates an iterative process once the first trial solution is given. For it we use a solution that assigns nobody to the nighttime shift on any day. Also we keep that specific schedule which produced the current trial solution out of the comparison process to avoid a creation of a loop. The iteration is continued until we have a schedule whose value is zero, i.e. a schedule which satisfies the nighttime constraints.

Starting from this schedule, we go to the second phase of satisfying the daytime constraints through the same process after we prepare a set of schedules for each nurse which satisfy the daytime constraints like in the first phase. We again use as the initial solution a schedule which assigns nobody to the daytime shift on any day.

This algorithm obtains a good solution, but is often slow as it basically checks all the alternative schedules for each nurse. The main contribution of this paper is to reduce the time taken in solving the subproblem by curtailing the number of alternative schedules to be looked at in each iteration.

The observation of a sequence of the trial solutions in the first phase of the algorithm to satisfy the nighttime constraints shows the following. The two adjacent trial solutions have a small difference where the difference is defined to be the number of days when the night shift is assigned in one solution while not in the other: The difference is 0 to 3.

This leads us to the modified algorithm which focuses only on the schedules that have a little difference from the current solution to be examined and results in a remarkable improvement.