

列生成法を用いたナーススケジューリング問題の解法

新 妻 真 輔^{†1} 池 上 敦 子^{†2} 品 野 勇 治^{†1}

医療施設では絶え間なく質の高い医療サービスを提供しなければならない。そのためには適切な看護師の配置を考える必要がある。しかし、勤務の組合せ数や各看護師についての制約条件が多いため、毎日の各勤務に必要なとされる看護師の人数を満たし、各看護師の勤務負荷に関する制約を満たす勤務表の作成は困難である。本稿の目的は次の3つである。まずはじめにナーススケジューリング問題について説明し、我が国のモデルと、海外文献に見られるモデルの2種類に対して定式化を行い、最後に列生成法による解法を提案する。

A Column Generation method for Nurse Scheduling Problem

SHINSUKE NIIZUMA,^{†1} ATSUKO IKEGAMI ^{†2} and YUJI SHINANO^{†1}

Nurse scheduling problem is an important task in order to provide high quality medical services to patients in hospitals. Its solution can be hard to obtain, since there are many roster variations and constraints, for examples, the number of nurses for a shift, the workload for each nurse, the average skills for a shift and so on. This paper introduces two mathematical programming formulations for the nurse scheduling problem: a formulation for Japanese hospital's working environments and for other countries. Then we describe several ideas of column generation technique to solve these formulations.

1. はじめに

医療施設では、24時間絶え間なく質の高い医療サービスを提供する必要がある。そのため、各日・各勤務(シフト:夜勤,日勤など)に必要な看護師数を満たし、各看護師の勤務負荷を考慮した看護師の勤務の配置(ナーススケジューリング)を行わなければならない。看護師の勤務表を作成する際、考慮すべき制約は非常に多いため、勤務表作成は、全ての条件を満足する看護師の勤務配置を求めることも困難な問題である。また、我が国では就業形態による違いも問題を難しくしている要因である³⁾⁷⁾。我が国では、シフト毎に雇用契約をする米国などとは対照的に、各看護師が複数のシフトをローテーションしていたり、個々の看護師のスキルや希望などを考慮していることから、各看護師についての勤務の組合せが多くなる。本研究では、我が国のナーススケジューリング問題を解くことを意識して、モデルを整理し、列生成法を用いた解法を提案する。

2. ナーススケジューリング問題

我が国の看護師の勤務表作成において考慮しなければならない制約条件は、アンケート調査⁸⁾により、次の制約条件が明らかにされている。

- (1) 毎日の各勤務に必要な人数を確保すること
- (2) スキルレベルや業務上の所属チームを考慮して各勤務のメンバーを構成すること
- (3) 各看護師について各勤務の回数が決められた範囲内であること
- (4) セミナ等その他の業務や休日の希望を達成すること
- (5) 禁止される勤務パターンを入れないこと

(1)~(5)よりナーススケジューリング問題の制約条件は(1),(2)の各日・各勤務に関する条件(勤務制約条件)と(3),(4),(5)の各看護師の勤務負荷を考慮する制約条件(看護師制約条件)に大別することができる。

これらの制約条件を考慮して、我が国におけるナーススケジューリング問題は以下のように定義されている⁷⁾。

^{†1} 東京農工大学

Tokyo University of Agriculture and Technology

^{†2} 成蹊大学

Seikei University

ナーススケジューリング問題

看護師の人数 m 、スケジュール日数 n 、勤務の種類の数 w 、スキルレベルやチーム構成等によるグループ、同じ勤務での組合せを避ける看護師ペアまたはグループ、前月の勤務表が与えられ、毎日の各勤務に必要な看護師数と各グループからの人数の上限と下限、各看護師の各勤務に対する回数の上限と下限、それら以外の業務の日程、休日希望日、そして禁止される勤務パターンが明らかであるとき、これらの条件の下で、できるだけ希望目標が達成されるようなスケジュールを組みたい。

3. 定式化

ナーススケジューリング問題の制約条件は、勤務制約条件と看護師制約条件の2つに分類できる。この性質を利用して、各看護師について看護師制約をすべて満たした勤務パターン（看護師の勤務スケジュール）を求める問題（以後は部分問題とよぶ）と、部分問題を解いたときの勤務パターンを組合せて各日・各勤務における制約条件を満たす勤務表を求める問題（以降は主問題とよぶ）に分けて解くことを考える。このことで、問題の探索領域を一度に考慮する必要がなくなるという利点が生じる。

現場で利用する際には、看護師制約をすべて満たした勤務パターンを求める必要があることは、調査から明らかになった⁷⁾。実際に与えられた制約条件をすべて満たすような勤務表が作成できない場合、一般的には看護師制約を緩和して対応することになる⁷⁾。現場では、どの看護師のどのような希望を我慢してもらうかは、各看護師の好み、性格、健康状態をよく把握している看護師長や主任だけである。

本研究では、あらかじめ求めてある各看護師制約条件を全て満足する実行可能な勤務パターンが、主問題の勤務制約条件を記述する係数行列の「列」に対応するように問題を定式化した。ここでは、看護師人数 m は固定されているものとし、看護師数の最少化は行わない。そのかわりに、実行可能解を得る（つまり、各日・各勤務における不足人数をなくす）ことを目的とする。そこで、4節で提案する手法での手順をわかりやすくするためにも、不足人数を表す変数 μ_{jk} を定式化に組み込んでおき、目的関数には、各日・各勤務における不足人数の総和の最小化を明示した。

公開されているベンチマーク問題⁹⁾では、各看護師

についての休み希望や勤務希望は考慮されていないものもある。提案手法の有効性を確かめるために勤務パターンへの看護師の割当て方が異なる、2つの主問題の定式化を考案した。定式化1は Millar のモデル²⁾等に見られるような、同じ勤務パターンに複数の人が割当てられるモデルである²⁾⁶⁾。このモデルは、各看護師についての休み希望や勤務希望などが考慮されていない。定式化2は、我が国の現状に即して、各看護師ごとに異なる勤務パターンを求める必要がある場合のモデルである。入力として与えられる各看護師に対して、個別に勤務パターンが割当てられる。つまり、各看護師について休みの希望や勤務希望が考慮できる。

定式化1

$$\text{minimize} \quad \sum_{j \in N, k \in W} \mu_{jk} + \sum_{q \in P} 0 \cdot \lambda_q \quad (1)$$

subject to

$$a_{jk} \leq \mu_{jk} + \sum_{q \in P} \delta_{qjk} \lambda_q \leq b_{jk}, \quad j \in N, k \in W \quad (2)$$

$$\sum_{q \in P} \lambda_q = m \quad (3)$$

$$\lambda_q \in \mathbb{Z}^+, \quad q \in P \quad (4)$$

$$\mu_{jk} \in \mathbb{Z}^+, \quad j \in N, k \in W \quad (5)$$

ここでの記号の定義は以下である。

m : 看護師の人数

N : スケジュール対象となる日の集合

W : 勤務の集合

P : 看護師制約を満たした勤務パターン集合

a_{jk} : 看護師が j 日の勤務 k で必要となる人数の下限値を示す定数

b_{jk} : 看護師が j 日の勤務 k で必要となる人数の上限値を示す定数

δ_{qjk} : 看護師が勤務パターン q で j 日の勤務 k が勤務なら1、勤務でないなら0となる定数

μ_{jk} : 看護師が j 日の勤務 k に不足する人数を示す変数

λ_q : 勤務パターン q に割当てられる人数を示す変数

定式化1では、(1)式の目的関数は、各日・各勤務で不足する看護師の合計人数の最小化を示し、(2)式が看護師が j 日の勤務 k において必要となる人数の上下限値におさまる条件、(3)式は勤務パターンに割当てられる看護師の人数は m 人である、つまり、勤務制約条件を示す。定式化1では、(4)(5)の決定変数 μ_{jk} 、

λ_q がいずれも非負整数となる。

看護師制約は、勤務パターンを生成する際に考慮済みであることに注意されたい。3 日間のスケジュールを考え、勤務が日勤 (1)、夜勤 (2) の 2 交代制の場合を例とすると、看護師の勤務パターン q は、次の列ベクトルで表される。

$$[\delta_{q,1,1}, \delta_{q,1,2}, \delta_{q,2,1}, \delta_{q,2,2}, \delta_{q,3,1}, \delta_{q,3,2}]^T$$

定式化 1 においては、(2) 式の係数ベクトルの $\delta_{q,*}$ 列は、看護師制約を満足する勤務パターンのみしか利用されていない。例えば、全 64 パターンの内、看護制約を満足するのは以下の 3 パターンだけかもしれない。このパターン生成時に、看護師制約は考慮済みである。

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

定式化 2

$$\text{minimize} \quad \sum_{r \in R, j \in N, k \in W} \mu_{rjk} + \sum_{i \in M, q \in P_i} 0 \cdot \lambda_q \quad (6)$$

subject to

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

$$\sum_{q \in P_i} \lambda_{iq} = 1, \quad i \in M \quad (8)$$

$$\lambda_{iq} \in \{0, 1\}, \quad i \in M, q \in P \quad (9)$$

$$\mu_{rjk} \in \mathbb{Z}^+, \quad r \in R, j \in N, k \in W \quad (10)$$

ここでの記号の定義は以下である。

M : 看護師の集合

N : スケジュール対象となる日の集合

W : 勤務の集合

R : 看護師グループの集合

P_i : 看護師 i について看護師制約を満たした勤務パターン集合

G_r : グループ r に属する看護師の集合

a_{rjk} : グループ r からの看護師が j 日の勤務 k で必要となる人数の下限値を示す定数

b_{rjk} : グループ r からの看護師が j 日の勤務 k で必要となる人数の上限値を示す定数

δ_{iqjk} : 看護師 i が勤務パターン q で j 日の勤務 k が勤

務なら 1、勤務でないなら 0 となる定数

μ_{rjk} : グループ r からの看護師が j 日の勤務 k に不足する人数を示す変数

λ_{iq} : 看護師 i に勤務パターン q が割当てられれば 1、そうでなければ 0 となる変数

定式化 2 では、定式化 1 と同様 (6) 式の目的関数は、各日・各勤務で不足する看護師の合計人数の最小化を示し、(7) 式が勤務制約条件を示す。ここで、(8)(9) 式が看護師 i の取りえる勤務パターンを 1 種類に規定している。(10) の決定変数 μ_{rjk} は、定式化 1 と同様、非負整数となる。

4. 提案手法

Subproblem-Centric-Approach⁴⁾ では部分問題の解を列挙し、看護師制約条件を満たす勤務パターン集合から目的関数値に貢献する勤務パターンを 1 つ選び、選ばれた勤務パターンに該当する看護師の勤務パターンの交換を繰り返して解を改善しながら探索していった。しかし、スケジュール日数と看護師の人数が多くなると、列挙にかかる時間は増大してしまう。また、列挙した勤務パターンを交換をしていく方法では、局所解に陥ってしまう可能性がある。

この 2 つの問題を解決するために、通常の branch-and-price アルゴリズムが行っているように、部分問題として、新しく生成する勤務パターンの列の主問題における被約費用を最小化する問題 (PP: Pricing Problem) を整数計画問題として定式化する。この PP を用いて、次のように列生成し、生成された列を列プールへ保存する。列プールの初期状態は空集合を仮定する。

Step1: 主問題の線形緩和問題を解いた結果の双対変数を用いて PP を定式化する

Step2: PP の線形緩和問題 RPP を解く

Step3: もし、RPP の最適解の目的関数値が非負なら、それを示して終了。

Step4: もし、RPP の解が整数解なら、その列を列プールへ保存して終了。

Step5: 整数でなかった変数に対する、適当な丸め方のパターンに対して、看護師制約を満たすパターンがあるか調べる。看護師制約を満たす全てのパターンでは、被約費用をチェックし負である列をすべて列プールへ保存し終了。

Step6: 実行可能パターン未発見で終了

ここで、列プールが空でなく終了した場合は、生成された列を主問題に追加し、主問題の線形緩和問題を

解いて、上記 Step1 から繰り返す。一番最初の主問題は適当に l 列生成する。例えば定式化 1 の場合であれば、 $\lambda_q = 1 (q=1, \dots, l)$ を仮定して、 $\delta_{q,*} (q=1, \dots, l)$ を生成する。(2) 式を満足するように μ_{jk} の値を調整することで実行可能解を構成する。

Step6 において、実験的には PP を解くことも試み、線形緩和問題のみで求めた場合との違いを調べたい。

5. 今後の課題

現状では、branch-and-price アルゴリズムは構成できておらず、主問題のルート問題での処理しか考えられていない。分枝操作と、分枝後の列生成手法を設計する必要がある。分枝操作を行うことも、適当ではない可能性がある。その場合は、上下界値のチェックにより、分枝操作は行わずに、終了判定できるようなアルゴリズムとしたい。本稿で、あえて部分問題として整数計画問題を生成しているのは、我が国の現状の厳しい制約条件を十分に表現できることに起因している。しかし、列生成法では、部分問題はヒューリスティック解法で解くのが一般的である。適当なヒューリスティックの構築も検討したい。

残念ながら、実装が追いつかなかったもので、実験結果については可能な範囲で発表時に紹介したい。

参 考 文 献

- 1) Miller, Pierskalla, Rath: Nurse scheduling using mathematical programming. Operational research, Vol.24, pp.857-870, 1976.
- 2) Millar, H.H., Kiragu, M: Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. European Journal of Operational Research 104, pp.582-592, 1998.
- 3) 岡本文朗: 看護配置基準の問題点とその背景-国際比較をふまえて-, Institute for Technology Enterprise and Competitiveness, 2005.
- 4) Atsuko Ikegami, Akira Niwa: A Subproblem-centric Model and Approach to the Nurse Scheduling Problem, Mathematical Programming, Vol.97, No.3, pp.517-541, 2003.
- 5) Jonathan F. Bard, Hadi W. Purumo: Preference scheduling for nurses using column generation, European Journal of Operational Research, 164, pp.510-534, 2005.
- 6) Joroen Belien, Erik Demeulemeester: A Branch-and-price approach for integrating nurse and surgery scheduling, European Journal of Operational Research, 189, pp.652-688, 2008.
- 7) 池上敦子: ナース・スケジューリング-調査・モデ

リング・アルゴリズム, 統計数理, Vol.53, No.2, pp.231-259, 2005.

- 8) 池上敦子, 丹羽明, 大倉元宏: 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, Vol.41, No.8, pp.436-442, 1996.
- 9) Personnel Scheduling Data Sets and Benchmarks: <http://www.cs.nott.ac.uk/tec/NRP/>