

# 介護士スケジューリングにおける過去の勤務表からの制約条件の抽出に関する研究 A Study on Constraint Extraction from Past Schedules in Care Worker Scheduling

末永 康貴<sup>1)</sup> 小野 智司<sup>1)</sup>  
Koki Suenaga Satoshi Ono

## 1 はじめに

介護業界では少子高齢化の進展に伴い人手不足が深刻化しており、職員の新規雇用促進と離職率低減が重要な課題となっている。従業員の勤務希望を反映した勤務表の生成はその対策方法の1つであるが、現在は施設の管理者が多くの時間と労力をかけて手動で作成している。

介護施設における勤務表作成は、看護師スケジューリング (Nurse Scheduling Problem: NSP) [1-3] と比較して、雇用形態が多様で、資格保有状況に応じたシフト制限が多いほか、シフトの組み方における要件が施設によって大きく異なる。施設毎に制約条件を定義することから、複数回にわたるヒアリング作業が必要となる。これは施設にとって大きな負担となり、システム導入を阻害する要因の一つとなっている。

本論文は、介護施設の過去の勤務表から制約条件を抽出し、抽出した制約条件を用いて勤務表を生成する手法を提案する。提案手法は、過去の勤務表に含まれる、人手不足により生じた望ましくないシフト割り当てを除外することに特徴がある。また、制約テンプレートを用いて抽出した制約条件で勤務表が生成されない場合、制約条件を段階的に緩和する。評価実験により、制約テンプレートで施設の制約条件を抽出し、制約条件の緩和により現実的な勤務表が生成できることを確認した。

## 2 関連研究

NSP を対象として、過去の勤務表から制約条件を抽出する研究が行われている [4-6]。Kumar らは、集約関数を用いることで制約条件を抽出する COUNT-OR を提案した [4]。Ben らは、NSP の動的特性、特に事故や看護師の病欠といった予測不能な事象に焦点を当て、事前知識なしに過去の勤務表から制約条件を自動的かつ暗黙的に学習する手法を提案した [6]。これらの手法と比較して、提案手法は例外的なシフトを除外して制約条件の抽出を行える点に特徴がある。

## 3 提案手法

本研究では、制約テンプレートを用いて過去の勤務表から制約条件を抽出し、抽出した制約条件から勤務表を生成する手法を提案する。提案手法の構成を図 1 に示す。本手法は、抽象化されたシフトの勤務表 [7] を生成したのち、具体化したシフトの勤務表を生成する。これにより、夜勤など制約条件の充足が難しいシフトを先に割り当てることができ、実行可能な勤務表の生成が可能となる。

本手法で適用する制約テンプレートを表 1 に示す。制約テンプレートは、期間、人数、一般性、種類、参照情報で構成され、着目する期間や職員数、種類を変更することで多様な制約条件を抽出することができる。

制約条件の抽出段階では、事前に設計した制約テンプレートを用いて過去の勤務表からシフトの回数や勤務パターンなどの制約条件を抽出する。介護施設の勤務表は

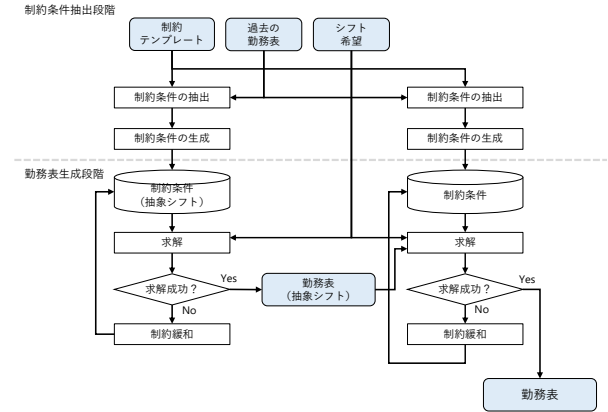


図 1 提案手法の処理手順

表 1 制約テンプレート

ID	制約テンプレート	期間	人数	一般性	分類	参照
T <sub>1</sub>	連続する $n$ 日間の勤務パターン	$n$ 日	各職員	特定	パターン	-
T <sub>2</sub>	連続する $n$ 日間の勤務パターン	$n$ 日	各職員	一般	パターン	-
T <sub>3</sub>	1か月に出現したシフトの回数	1か月	各職員	特定	回数	-
T <sub>4</sub>	各曜日のシフトの回数	1日	全職員	特定	回数	曜日

人員不足により例外的なシフトが割り当てられる場合があり、このシフトを含むパターンを抽出した場合、勤務表生成時に避けるべき勤務パターンも許容される。このため、パターンの抽出において余裕度と柔軟性、出現頻度を考慮し、例外的なパターンを除外する。余裕度  $u_d$  は、ある勤務日  $d$  における勤務可能な職員数  $a_d$  を必要な人数  $r_d$  で割った値  $\frac{a_d}{r_d}$  であり、閾値  $\tau_u$  を下回った場合、その日を含むパターンの抽出を行わない。柔軟性  $u_f$  は、ある職員の1か月に於ける休暇希望日数  $N_d^{(r)}$  を実際の休暇日数  $N_d^{(a)}$  で割った値  $N_d^{(r)}/N_d^{(a)}$  であり、閾値  $\tau_f$  を下回った場合、その月の勤務パターン抽出を行わない。出現頻度  $u_c$  は抽出したパターンの出現頻度であり、閾値  $\tau_c$  を下回ったパターンは制約条件から除外する。

勤務表生成段階では、制約条件の抽出段階で抽出した制約条件と施設側からの希望シフトをもとに勤務表を生成する。提案手法は、実行可能な解が見つからない場合、強い制約条件を弱い制約条件に緩和し、繰り返し解を求める。制約テンプレート T<sub>2</sub> で抽出した制約条件は、強い制約条件として扱う。しかし、過去の勤務表に一度も出現しないパターンをすべて禁止するため、制約条件が過剰になり、他の強い制約条件と干渉する可能性がある。このため、解が見つからない場合、長いパターンから順に弱い制約条件に緩和し、再度解を求める。

## 4 評価実験

提案手法の有効性を検証するため、鹿児島市内の介護施設における40か月分の勤務表を用いて実験を行った。当該施設は約20人の職員ならびに2つのユニットにより構成される。シフトの対応を表3に示す。抽象化されたシフト (抽象シフト) の種類を6種類 (-: 休, G: 日勤, D: D勤, N: 夜勤) とし、具体化されたシフト (詳細

1) 鹿児島大学 Kagoshima University

表 2 制約テンプレートと制約条件の関係

区分	ID	管理者へのヒアリングで得られた制約条件	制約テンプレート
強	H <sub>1</sub>	勤務日数は指定する範囲内に収める	T <sub>3</sub>
	H <sub>2</sub>	各職員に対し、勤務可能なシフトのみ割当て	T <sub>3</sub>
	H <sub>3</sub>	勤務時間は指定する範囲内に収める	T <sub>3</sub>
	H <sub>4</sub>	各シフトに対して必要な人員数を確保	T <sub>4</sub>
	H <sub>5</sub>	夜勤入と夜勤明けを連続する 2 日間に割当て	T <sub>2</sub>
弱	S <sub>1</sub>	夜勤明け翌日は休を割当て	T <sub>1</sub>
	S <sub>2</sub>	夜勤、休が不可能な場合、夜勤、日勤、休でも可	T <sub>1</sub>
	S <sub>3</sub>	夜勤、休、夜勤を禁止	T <sub>1</sub>
	S <sub>4</sub>	夜勤の割当て回数を平準化	T <sub>3</sub>
	S <sub>5</sub>	日勤を 4 日以上連続で割当てない	T <sub>1</sub>
	S <sub>6</sub>	希望シフトを満たす	-
	S <sub>7</sub>	連続勤務のうち、A 勤は 1 回のみ	T <sub>1</sub>
	S <sub>8</sub>	B 勤を連続で割当てない	T <sub>1</sub>

表 3 シフトの対応

シフト	抽象	詳細	シフト	抽象	詳細	シフト	抽象	詳細
休	-	-	1A	G	1A	2A	G	2A
年休	-	-	1B	G	1B	2B	G	2B
			1C	G	1C	2C	G	2C
			1D	1D	1D	2D	2D	2D
			1E	G	1E	2E	G	2E
			1夜入	1N	1N	2夜入	2N	2N
			1夜明	1N	1N	2夜明	2N	2N

シフト)の種類を 12 種類とした。シフトの前の数字はユニットを表す。なお、日勤の一種である D 勤は勤務可能な職員が限定されており、詳細シフトを生成する段階で制約条件を満たすことが難しいため、抽象シフトとした。すなわち、詳細シフトを作成する段階では、抽象シフトにおける日勤を「A 勤、B 勤、C 勤、E 勤」に詳細化するとともに、ユニットの割り当てを決定した。

まず実験 1 として、36 か月分の勤務表と対応する希望シフト情報をもとに、抽象シフトと詳細シフトのそれぞれに関する制約条件の抽出を試みた。制約テンプレート T<sub>1</sub> で抽出した職員 10005 の勤務パターン例を図 2 に示す。図 2(1) は抽象シフトの形式で抽出したパターンであり、夜勤後に休日を割当て、3 日連続勤務した後に休日を割当てなどのパターンが得られたことが分かる。また、図 2(2) は詳細シフトの形式で抽出したパターンであり、A 勤、B 勤、C 勤を連続で割当て、A 勤後に C 勤を連続で割当てなどのパターンが得られたことが分かる。一方で、余裕度や柔軟性、出現頻度で例外として除外された勤務パターン例を図 3 に示す。図 3(1) をみると、夜勤直後に日勤を割当て、日勤の前後に休を割当てなどのパターンが除外されたことが分かる。また、図 3(2) をみると、C 勤の前後に A 勤を割当てなどのパターンが除外されたことが分かる。

次に実験 2 として、実験 1 で抽出した制約条件を用いて勤務表の生成を試みた。4 か月を対象とし、抽出した制約条件や希望シフト情報をもとに勤務表生成を試みたが、過去の勤務表に含まれない職員の制約条件は、類似する職員と同様の制約条件を適用した。

提案手法で生成した勤務表の例と違反数をそれぞれ図 4 と図 5 に示す。図 5 より、提案手法により生成された勤務表は、すべての強い制約条件を満たし、また、S<sub>2</sub> を除くすべての弱い制約条件に関して管理者が作成した勤務表よりも違反数が同程度以下であることを確認した。

### 5 結論

本研究は、制約テンプレートを用いて過去の勤務表から制約条件を抽出し、抽出した制約条件を用いて勤務表を生成する手法を提案した。実際の介護施設の勤務表を用いて実験を行った結果、制約テンプレートにより職員の勤務パターン施設の制約条件を抽出し、制約条件の緩

(10005, 1N, -)	(10005, 1A, -, 1N)
(10005, G, 1N)	(10005, 1D, -, 1N)
(10005, 1N, -, G)	(10005, 1A, 1B, 1C)
(10005, G, G, G)	(10005, 1A, 1C, 1C)
...	...

(1) 抽象シフト

(2) 詳細シフト

図 2 テンプレート T<sub>1</sub> を用いて抽出した勤務パターン

(10005, 1N, G)	(10005, 1B, 1D)
(10005, 1N, 1N, G)	(10005, 1B, 1D, 1N)
(10005, -, -, G, -)	(10005, 1A, 1C, 1A)
(10005, -, G, -, G, 1N)	(10005, 1B, 1A, 1B)
...	...

(1) 抽象シフト

(2) 詳細シフト

図 3 例外として除外された勤務パターン

Staff	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		
I0001	1N	-	1D	1A	-	1B	1D	1N	-	1A	1N	-	1B	1N	-	1N	1N	-	1N	1N	-	1A	1N	-	1D	1N	-	1D	1N	
I0002	1D	1N	-	1C	-	1A	1C	-	1B	1N	-	1B	1A	1D	-	1B	1A	1N	-	1C	1B	1N	-	1D	1N	-	1D	1N		
I0003	-	1D	-	1D	1A	1D	1N	-	1D	1D	1A	-	1D	1D	1A	-	1N	1N	-	1D	1D	1A	-	1D	1N	1C	-	1D	1N	
I0005	1N	1N	-	1B	1N	-	1A	1B	1C	-	1C	1B	1C	1N	-	1A	-	1A	1B	1N	-	1C	1A	1B	-	1A	-	1A	-	
I0006	1A	-	1B	1C	-	1C	1B	-	1A	-	-	1C	-	1B	1C	-	1C	1B	-	1C	1B	-	1C	1B	-	1A	-	1A	-	
I0018	1B	1A	-	-	-	1C	-	-	1B	-	-	1B	-	-	1A	1C	1B	-	1C	1B	-	1C	1B	-	1A	1B	1B	-	1B	-
I0008	-	1B	1C	-	1B	1A	-	1D	1D	1A	-	1D	1A	1C	-	1D	1D	-	1D	1A	-	1D	1D	-	1C	1A	-	1C	1A	-
I0007	1C	1C	1A	-	-	-	1B	1C	-	1C	1B	-	-	1C	1B	-	-	1C	1B	-	-	1C	1A	-	-	1C	1A	-	-	-
I0010	-	-	1N	1N	-	1N	1N	-	1N	1N	-	1N	1N	-	2N	2N	-	-	-	-	-	-	-	-	1N	1N	-	1N	1N	
I0011	-	-	1N	1N	-	-	1N	1N	-	-	1N	1N	-	1N	1N	-	1N	1N	-	1N	1N	-	-	2N	2N	-	2N	2N	-	-
I0012	2C	2C	2A	-	-	2B	2C	-	2C	2A	2B	-	2B	2C	2N	2N	-	2C	2C	2B	2N	2N	-	-	2B	2N	2N	2C	-	
I0014	2A	2N	2N	-	-	2D	2D	2D	2A	2N	2N	-	2A	2B	-	-	2D	2D	2A	2N	2N	-	-	2D	2A	2B	2N	2N	-	
I0013	2D	2A	2N	2N	-	2A	2N	2N	-	2D	2C	2N	2N	-	2A	2D	2B	2N	2N	-	2C	2B	2A	2C	-	2A	-	2A	-	
I0015	2B	-	2B	2A	2C	-	2B	2C	2B	-	2A	-	2C	-	2B	2A	2C	-	2B	2C	2A	-	2C	2B	2C	-	2B	2A	-	
I0044	2N	-	2C	2N	2N	-	2A	2N	2N	-	2D	2B	2D	2N	2N	-	2A	2C	-	2D	2D	2A	2N	2N	-	2A	-	2N	-	
I0045	-	2D	2D	2D	2N	2N	-	2B	2N	2N	-	2C	2N	2N	-	2B	2N	2N	-	2A	2N	2N	-	2D	2D	2N	2N	-	-	-
I0046	2N	2N	-	2C	2B	2N	2N	-	2D	2B	2N	2N	-	2D	2D	2N	2N	-	2C	2B	2N	2N	-	2D	-	2D	-	2D	-	
I0019	-	2B	-	2B	2A	2C	-	2A	-	2C	2C	2A	-	2A	2C	2C	-	2B	2C	2A	-	2C	2B	2A	-	2C	2C	2B	-	
I0020	2E	-	2E	2E	-	-	-	2E	-	2E	-	2E	-	-	-	2E	2E	-	2E	2E	-	-	2E	2E	-	2E	-	2E	2E	2E

図 4 抽出した制約条件で生成した勤務表例

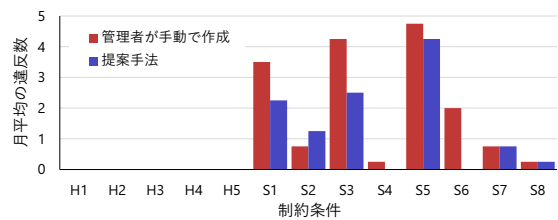


図 5 違反数の比較

和により、現実的な勤務表が生成できることを確認した。今後、他の施設においても実用的な勤務表が生成可能であることを検証する。

### 参考文献

- [1] S. Ceschia, et al. Solving a real-world nurse rostering problem by simulated annealing. *Operations Research for Health Care*, Vol. 36, p. 100379, 2023.
- [2] OM Agboola, et al. Application of metaheuristic algorithms to personnel scheduling problem in healthcare management: A. *Technical & Industrial Sponsors*, p. 81, 2016.
- [3] T. Kurokawa, et al. Meta-heuristic solution of supply unit sufficient type care service for senior housing. *J. Japan Soc. Welfare Engineering*, Vol. 19, No. 1, pp. 14–20, 2017.
- [4] M. Kumar, et al. Automating personnel rostering by learning constraints using tensors. *Int'l Conf. Tools with Artificial Intelligence*, pp. 697–704, 2019.
- [5] M. Kumar, et al. Learning max-sat from contextual examples for combinatorial optimisation. In *AAAI*, pp. 4493–4500, 2020.
- [6] A. B. Said, et al. An implicit learning approach for solving the nurse scheduling problem. *Int'l Conf. Neural Information Processing*, pp. 145–157, 2021.
- [7] K. Suenaga, T. Furuta, S. Ono. A Preliminary Study on Constraint Extraction and Exception Exclusion in Care Worker Scheduling. *Int'l Symp. Artificial Life and Robotics*, 2025.