

介護士スケジュールリングにおける制約の自動抽出に関する基礎検討

末永康貴 永井裕也 柏木一杜 小野智司[†]
鹿児島大学[†]

概要

飲食店や工場等における従業員の勤務シフトを作成するシステムが普及している一方で、介護における職員の勤務表は手動で作成されることが多い。これは、制約条件が介護施設毎に大幅に異なるため、システムを導入する際の制約条件を定義する労力が膨大であることによる。このため本研究では、過去の勤務表から制約条件を自動抽出し、抽出した制約条件を用いて勤務表を生成する方式を提案する。

1 はじめに

近年、介護業界では人手不足が深刻化しており、職員の新規雇用を促進する一方で、離職率の低減を図ることが重大な課題となっている。従業員の希望を満たす勤務表を生成することは、その対策方法の1つであるといえ、施設の管理者が多く時間と労力をかけて手動で作成を行っている。

勤務表を自動で生成する技術が広く研究されているが[1]、介護施設においては、雇用形態や資格が多様であることや条件が施設によって異なるため、施設毎に制約条件を定義する作業が必要であり、システム導入を阻害する要因となっている。

本研究では、過去の勤務表から制約条件を自動抽出し、抽出した制約条件を用いて勤務表を生成する方式を提案する。提案手法は、制約テンプレートをを用いて組合せを抽出し、余裕度を用いることで例外的な組合せを除外して勤務表を生成する。実験により、提案する手法が制約条件を抽出し、勤務表を生成可能であることを示す。

2 関連研究

制約条件を過去のデータから抽出する研究は、テンソル表現を用いた手法[2]や制約テンプレートを

表 1: 適用する制約テンプレート

番号	制約テンプレート	期間	人数	種類	参照
T_1	連続する n 日間のシフトの組合せ	n 日	1 人	pattern	-
T_2	1 か月に出現したシフトの回数	1 月	1 人	count	-
T_3	各曜日のシフトの回数	1 日	全員	count	-
T_4	各日の 2 人の職員の組合せ	1 日	2 人	pattern	-
T_5	連続日勤のシフトの組合せ	mul	1 人	pattern	-

用いた手法[3]が提案されている。テンソル表現を用いた手法では、多次元の設計変数から集約関数を利用して制約条件を抽出することを可能とした。制約テンプレートをを用いた手法では、2次元の表形式データから特定のブロック内のデータを集約して抽出方法を指定することで、要素の数や組合せなどを抽出することを可能とした。

3 提案手法

本論文では、制約テンプレートをを用いて施設の過去の勤務表から制約条件を自動抽出し、抽出した制約条件を基に勤務表を生成する方式を提案する。多様な制約テンプレートをを用いて制約条件の抽出を行うことで、過去の勤務表で考慮されている複雑な制約条件の抽出が可能である。また、スタッフ数が不足している場合などに組まれる例外的なシフトを抽出しないように、スタッフの余裕度を考慮する。

制約条件を抽出する段階では、抽出するシフトや職員の組合せを指定する制約テンプレートをを用いて指定したシフトの組合せを過去の勤務表から取得し、出現した組合せを割当て可能な組合せとする。本手法で適用する制約テンプレートをを表1に示す。制約テンプレートは、期間、人数、種類（組合せまたは回数）および参照情報により構成される。例えば、制約テンプレート T_1 は、ある職員が連続して勤務したシフトの組合せを抽出し、 T_2 は、ある職員が1か月間に各シフトを何回担当したかを計測する。

制約条件の抽出を行う際は、スタッフ数の余裕度を考慮し、例外的な勤務が行われた期間や職員を除外する。ある勤務日 d における余裕度 u_d は、各日の勤務可能な職員の人数 a_d を必要な人数 r_d で割った値 $\frac{a_d}{r_d}$ とする。上記の余裕度が閾値 τ_u を下回った場

A Preliminary Study on Automatic Constraint Extraction in Care Worker Scheduling

[†] Koki Suenaga, Yuya Nagai, Kazuto Kashiwagi, and Satoshi Ono, Kagoshima University

表 2: ヒアリングにより得られた制約条件の例と抽出に利用した制約テンプレートとの対応

区分	番号	ヒアリングで得られた制約	テンプレート
強	H_1	勤務日数は指定する範囲内に収める	T_2
	H_2	各シフトに対して必要な人員数を確保	T_3
	H_3	夜勤入と夜勤明を連続する 2 日間に割当	T_1
弱	S_1	夜勤の翌日は休を割当てて	T_1
	S_2	「夜勤→休」が不可能な場合は「夜勤→C 勤→休」でも可	T_1
	S_3	夜勤の割当て回数を平準化する	T_2
	S_4	日勤を 4 日以上連続で割当てない	T_1
	S_5	人間関係の悪い職員同士は同じ時間帯に割当てない	T_4

表 3: 抽出した制約条件の例

(a) T_1 により抽出された制約条件				(b) T_2 により抽出された制約条件		
Staff	d	$d+1$	$d+2$	Staff	Shift	Count
10001	H	H	H	10006	H	15
10001	D	1N	1N	10006	D	15
10001	D	D	1N	10006	1N	0
10001	1N	H	D	10006	2N	0
10002	D	H	D	10018	H	17
⋮	⋮	⋮	⋮	⋮	⋮	⋮

合は、制約条件の抽出を行わない。1日より長い期間を対象とする制約テンプレートを利用する際は、各日の余裕度の最小値を適用する。加えて、抽出したパターンの出現回数が閾値 τ_c を下回る際は、制約条件として抽出しない。

4 評価実験

提案する方式の有効性を評価するため、鹿児島市内の介護施設における勤務表を用いて実験を行った。まず実験 1 として、2020 年から 2022 年の 3 年分の実際の勤務表とシフト立案時に調査した希望休暇日の情報を用いて制約条件の抽出を試みた。次に実験 2 として、実験 1 で抽出した制約条件を用いて、2023 年の 7 ヶ月分の勤務シフト表の作成を試みた。当該施設は職員が 20 人で、2つのユニットにより構成されている。ユニット毎に日勤が 5 種類、夜勤が 2 種類存在するが、シフトの種類を 4 種類（休、日勤、ユニット 1 および 2 の夜勤）に一般化した。実験 2 では、CP-SAT(Constraint Programming-SAT)*1 を用いて勤務表の作成を試みた。実験 1 で抽出した制約条件の強弱の分類は手動で行うこととした。

ヒアリングにより得られた制約条件と、実験 1 で制約条件の抽出に用いた制約テンプレートとの対応を表 2 に示す。また、実験 1 において、制約テンプレート T_1, \dots, T_5 を用いて抽出した制約条件の例を表 3 に示す。表 3(a) より、スタッフ 10001 は日勤を

*1 https://developers.google.com/optimization/cp/cp_solver?hl=en

Staff	date															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
10001:	1N	H	D	D	H	D	1N	1N	H	1N	1N	H	D	1N	1N	
10002:	D	H	1N	1N	H	1N	1N	H	D	D	H	1N	1N	H	D	
10003:	D	H	H	D	D	H	D	D	D	D	H	D	H	D	H	
10005:	D	D	D	H	D	H	D	D	1N	1N	H	D	H	D	D	...
10006:	H	H	D	D	H	D	D	H	H	H	H	D	H	D	D	
10011:	1N	1N	H	2N	2N	H	H	H	H	H	H	H	2N	2N	H	
10050:	H	1N	1N	H	1N	1N	H	1N	1N	H	2N	2N	1N	1N	1N	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

図 1: 生成した勤務表の例

表 4: 各月の勤務表における違反数の平均

	S_1	S_2	S_3	S_4	S_5
管理者が作成した勤務表	2.2	1.2	0.6	4.0	4.6
提案手法 ($\tau_u = 1.3, \tau_c = 4$)	2.4	0.0	0.2	4.2	3.8

2 日行った後に夜勤を行えることや、表 3(b) より、スタッフ 10006 は日勤のみ行うことができ、1 ヶ月の休暇が 15 日であることがわかる。

次に、実験 2 において、実験 1 で抽出した制約条件を用いて勤務表を生成したところ、7 か月のうち 5 か月において勤務表を生成することに成功した。生成した勤務表の例を図 1 に、管理者が作成した場合、余裕度と出現回数を考慮した場合の 5 ヶ月分の勤務表における違反数の平均を表 4 に示す。表 4 より、管理者が手動で作成した勤務表と比較して同程度の違反数であることがわかった。

5 結論

本論文は、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自動抽出し、抽出した制約条件を用いて勤務表を生成する方式を提案した。多様な制約テンプレートを用いることで多くの制約条件が抽出でき、過去のシフトの組合せを考慮した勤務表が生成されることを確認した。今後、より実用的な勤務表を生成するための制約条件の抽出方法や、解が得られない場合に制約条件を緩和する処理について検討を行う。

参考文献

- [1] Sara Ceschia, et al. Solving a real-world nurse rostering problem by simulated annealing. *Operations Research for Health Care*, Vol. 36, p. 100379, 2023.
- [2] Kumar, et al. Automating personnel rostering by learning constraints using tensors. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 697-704. IEEE, 2019.
- [3] S Kolb, et al. Learning constraints in spreadsheets and tabular data. *Mach Learn* 106, 1441 - 1468, 2017.