1S-4

勤務表問題における ILP を用いた制約条件の生成と適用

椎名良久† 金盛克俊†† 大和田勇人†††

†東京理科大学大学院 理工学研究科 経営工学専攻 ††, †††東京理科大学 理工学部 経営工学科

1 はじめに

ナーススケジューリング問題等の勤務表問題の研究が従来から盛んに行われている．作成された勤務表は必ずしも各従業員にとって満足な勤務表であるとは言えない．そこで，なるべく各従業員の好みに合わせた勤務表を作成するべきである．

本研究では，勤務表問題において，各従業員の好みの勤務パターンや人間関係といった情報を考慮するため，帰納論理プログラミング(ILP)[1]によって，過去の勤務表データから，各従業員にとって良いものと判断できる勤務表の学習ルールの生成を行う．ILP を用いることによって，好みの勤務パターンや人間関係のルールを意味の持つ記号で表現することができる．また，得られたルールを出来る限り満たしたい SOFT 制約として扱うことによって，職場の規則や希望シフト等の必ず満たさなければならない HARD 制約を満たしつつ，ルールによる各従業員の好みのシフトの条件をできる限り満たすような勤務表の作成を行う．必要不可欠な制約を満たしつつ，各従業員の好みを出来る限り満たした勤務表が作成できると考えられる．

2 提案手法

本研究の提案手法について説明する．図 1 は本研究の提案手法の流れである．従業員側と作成者側の操作から成る．各従業員の学習ルールを生成する前に，従業員がシステムを通して，いくつかの勤務表に対して，良い勤務表なのか，悪い勤務表なのか，判断を行う．そして，作成者側の操作として以下の 2 つのプロセスからなる．

1. 帰納論理プログラミング(ILP)を用いて，良い勤務表を正事例，悪い勤務表を負事例として各従業員のルールの生成を行う．
2. 1 で得られた学習ルールを新たな SOFT 制約して扱い，HARD 制約を必ず満たし，SOFT 制約をなるべく満たすような新たな勤務表の作成を行う．

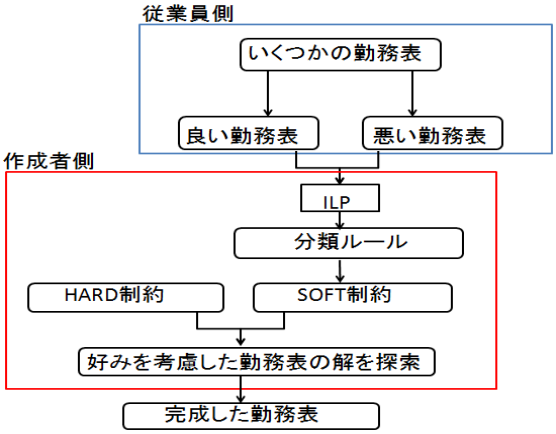

図 1 提案手法の流れ

3 提案手法の実行例

提案手法の実行例を示す際に使用する勤務表として，Ikegami-3Shift-DATA[2]を基に，それに独自に変更を加えたものを利用する．この勤務表では，日勤と準夜勤，夜勤の 3 交替制であり，看護師の人数は 25 人，スケジューリング期間は 30 日となっている．これらの看護師はそれぞれ A もしくは B チームに属し，各グループによるシフトの上下限の制約が示されている．また，各日にち，各看護師の休みや日勤，準夜勤，夜勤がそれぞれのシフトの上下限の制約で示されている．さらに，シフトの連続日数と間隔による上下限の制約，禁止されているシフトパターンが示されている．

3.1 学習ルールの生成

各看護師の好みのルールを生成するために，ILP システムとして GKS[3]を用いた．各看護師のルールを別々で生成している．ここでは，看護師 1 の学習ルールの生成する場合のモード宣言を以下のように表現した．

```
:- modeh(*,roster_nurse1(+term)).
:- modeb(*,sunday(+term,#shift,-outputterm)).
:- modeb(*,monday(+term,#shift,-outputterm)).
:- modeb(*,tuesday(+term,#shift,-outputterm)).
:- modeb(*,wednesday(+term,#shift,-outputterm)).
:- modeb(*,thursday(+term,#shift,-outputterm)).
:- modeb(*,friday(+term,#shift,-outputterm)).
:- modeb(*,saturday(+term,#shift,-outputterm)).
:- modeb(*,nurse2(+outputterm,#shift)).
:- modeb(*,nurse3(+outputterm,#shift)).
```

Generation and application of constraints using ILP for roster problems
Yoshihisa shiina†, Katsutoshi Kanamori†† , Hayato Ohwada†††
Department of Industrial Administration, Graduate school of Science and Technology Tokyo University of Science†,
Department of Industrial Administration, Faculty of Science and Technology, Tokyo University of Science††,†††

事例は日曜日から土曜日までの 7 日間による小勤務表とする．あらかじめ 102 期間分の小勤務表の事例を用意し，各従業員が勤務パターンや人間関係で，良い勤務表か悪い勤務表か判断させる．良い勤務表なら正事例，そうではない悪い勤務表を負事例とする．本研究では，各従業員が勤務パターンや人間関係から好みの勤務表を判断することを想定し，表 1 の指標を設定した．指標を基に，正事例と負事例を分けた．

ILP による学習を行った結果，看護師 1 の好みのルールは以下のようになった．

```
roster_nurse1(A):-  monday(A, evening, B), nurse13(B,
                    evening).
roster_nurse1(A) :- wednesday(A, evening, B), nurse13(B,
                    evening).
roster_nurse1(A) :- friday(A, rest, B), nurse23(B, evening).
```

1 つ目のルールは，「ある小勤務表 A の看護師 1 の月曜日の勤務が準夜勤であり，かつ看護師 13 の月曜日の勤務が準夜勤であるならば，小勤務表 A は看護師 1 にとって良い勤務表である」という意味を示している．このルールは表 1 による指標と一致している．2 つ目のルールも指標と一致している．3 つ目のルールは，指標と一致していない．看護師 1 は A チームに属しており，看護師 23 は B チームに属しているので，3 つ目のルールは，他の 2 個のルールに比べて，重要度が低いといえる．意味のあるルールを SOFT 制約として適用し，新たな勤務表の作成を行う．

表 1　各看護師の好みの指標

看護師	チーム	勤務パターンの指標	人間関係の指標
1	A	休み→休み	13番の看護師で準夜勤
2	A	準夜勤→休み	4番の看護師で日勤
3	A	日勤→日勤	11番の看護師で夜勤
4	A	準夜勤→休み	8番の看護師で夜勤
5	A	休み→夜勤	12番の看護師で日勤
6	A	日勤→日勤	10番の看護師で夜勤
7	A	日勤→日勤	4番の看護師で準夜勤
8	A	日勤→夜勤	2番の看護師で夜勤
9	A	休み→準夜勤	8番の看護師で日勤
10	A	休み→準夜勤	2番の看護師で日勤
11	A	日勤→日勤	1番の看護師で日勤
12	A	日勤→夜勤	4番の看護師で準夜勤
13	A	休み→休み	4番の看護師で夜勤

3.2　学習ルールの適用例

3.1 で得られた学習ルールを適用し，新たな勤務表の作成を行う．本研究は実際のルールを適用して勤務表を作成する段階に達していないので，ここでは表 1 の指標をもとに，仮のルールを用いて，勤務表の作成例を示す．表 2 は，通常の制約条件で作成した勤務表である．表 3 は，ルールを適用したときの勤務表である．なお，1：休み，2：日勤，3：準夜勤，4：夜勤，5：その他の勤務をそれぞれ表す．2 つの勤務表を比較してみると，表 3 の勤務表は「看護師 7 は看護師 4 と金曜日の準夜勤で働きたい」というルールが適用されている．一方，表 2 の勤務表はどの曜日においても看護師 7 は看護師 4 と準夜勤で働くことができない勤務表となっている．

通常の勤務表は希望シフトしか，従業員の要望しか応えられなかった．それに対して，各従業員のルールを適用した勤務表は，希望シフトを満たしつつ，なるべく従業員好みを多く満たした勤務表を作成することができた．

表 2　通常の勤務表

看護師	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
	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M
1	3	1	2	2	1	1	3	1	4	1	2	2	2	3	1	2	4	1	2	3	1	4	1	2	2	3	4	1	2	2
2	3	4	1	2	2	1	2	3	1	4	1	2	1	3	1	2	1	4	1	3	1	2	2	2	2	3	1	2	2	2
3	1	3	4	1	1	2	2	3	1	4	1	2	2	2	3	1	1	2	2	4	1	2	3	1	2	2	3	4	1	2
4	2	3	4	1	2	2	1	2	3	1	4	1	2	2	3	1	1	2	2	2	3	1	1	4	1	2	3	4	1	2
5	4	1	3	1	2	2	1	2	3	1	2	4	1	2	1	3	1	2	2	2	3	1	2	2	1	2	2	3	4	1
6	2	2	3	4	1	2	1	2	1	3	1	4	1	2	2	3	1	2	4	1	2	3	1	2	4	1	2	3	1	2
7	2	2	1	1	3	4	1	2	2	3	1	2	4	1	2	1	3	1	2	4	1	3	4	1	2	2	2	2	3	1
8	2	2	1	3	4	1	2	1	2	1	3	1	4	1	2	2	3	1	2	1	4	1	3	1	2	2	2	2	3	4
9	1	2	1	3	4	1	2	2	2	1	1	3	1	4	1	2	2	3	1	2	4	1	2	3	1	2	2	2	2	3
10	1	5	2	2	3	4	1	1	1	2	3	1	2	4	1	1	2	3	1	2	2	1	2	3	4	1	2	2	2	3
11	1	1	2	1	2	3	4	1	1	2	2	3	1	1	4	1	2	2	3	1	2	2	2	2	3	4	1	2	2	2
12	1	1	2	1	1	3	4	1	2	2	1	2	3	1	4	1	2	2	2	2	1	1	2	2	2	2	2	1	2	2
13	1	1	1	2	1	2	3	4	1	2	2	2	3	1	2	4	1	2	3	1	2	2	2	2	3	1	2	1	2	4

表 3　ルールを適用した勤務表

看護師	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
	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M
1	3	1	1	2	2	2	2	4	1	1	2	2	2	2	3	1	1	2	4	1	2	4	1	1	4	1	2	2	3	3
2	2	1	2	2	3	1	4	1	2	2	3	3	1	4	1	2	2	1	2	4	1	1	2	2	3	3	4	1	1	2
3	2	2	4	1	1	2	2	2	1	4	1	3	1	2	2	1	4	1	1	2	2	2	2	3	1	2	2	3	1	3
4	4	1	2	3	1	3	4	1	2	2	4	1	3	1	2	2	2	3	1	3	4	1	1	1	3	1	2	2	2	2
5	1	3	1	4	1	2	2	3	3	3	1	2	2	2	1	1	1	4	1	2	2	3	4	4	1	2	2	2	4	1
6	1	2	2	2	4	1	3	1	2	2	3	4	1	1	1	2	2	2	2	1	3	2	2	2	2	1	3	4	1	2
7	2	2	1	1	2	3	1	2	2	2	2	1	3	3	4	1	3	1	2	3	1	2	2	2	2	1	1	2	4	1
8	2	1	3	1	3	1	2	2	3	3	1	2	4	1	2	2	3	1	2	4	1	3	1	1	2	2	4	1	2	2
9	2	4	1	3	1	2	2	2	2	1	1	2	4	1	2	2	1	3	3	1	2	4	4	1	1	2	3	1	2	2
10	1	5	2	2	4	1	3	3	1	2	2	4	1	3	1	2	2	2	2	1	3	1	2	2	2	4	1	3	1	1
11	2	1	4	1	2	2	2	1	4	4	1	2	2	2	3	3	4	1	3	2	2	1	2	3	1	3	1	2	2	1
12	1	2	2	2	4	1	2	1	1	2	2	1	2	2	3	1	1	2	2	1	1	2	3	1	2	2	1	2	2	4
13	3	3	3	4	1	2	1	1	1	2	4	1	2	2	2	4	1	2	1	2	1	1	1	3	5	2	2	2	1	3	5

4　まとめ

本研究では，勤務表問題において ILP に用いることによって，過去の勤務表から，各従業員の勤務パターンや人間関係の好みのルールを生成し，得られたルールを SOFT 制約として適用し，新たな勤務表を作成する方法を提案した．過去の経験に基づいた勤務表を作成することができるといえる．今後は，実際の学習ルールを用いて，勤務表の作成を行い，通常の勤務表との比較検証を行う．また，これまで勤務表は手作業で作成していたが，自動的に作成できるシステムを目指す．

参考文献

[1] S. Muggleton: Inverse entailment and progol, New Generation Computing, Volume 13 Numbers 3-4, pp. 245-286, 1995.

[2] 池上敦子: ナース・スケジューリング - 調査・モデル化・アルゴリズム - , 数理統計, Vol.53, No.2, pp.231-259, 2005.

[3] F. Mizoguchi, H. Ohwada: Using inductive logic programming for constraint acquisition in constraint-based problem solving, In Proceeding of the 15th International Workshop on Inductive Logic Programming, pp. 297-332, 1995.