

修正作業を効果的に支援する Excel ベースの スタッフスケジューリングツールの開発

Excel based staff scheduling system for reducing the workload of modification of users

久保 琢磨†

Takuma Kubo

宇野 毅明‡

Takeaki Uno

1. はじめに

ホテルや病院、レストランなどの勤務スケジュールを作成する問題をスタッフスケジューリングという。勤務スケジュールを作成する作業は、出勤人数やスタッフの休日希望などを考慮する必要があるため、手作業では多大な時間を要する。ある程度自動で勤務スケジュールを作成可能であれば、作業者の負担はかなり軽減されると考えられており、これらの研究はおおいにされている。これらの代表的な研究として、看護師における勤務スケジュール作成の状況の現場調査やアンケートなどを行い、その結果を反映したモデルと解法を作成した研究[1][2]が挙げられる。

スタッフ数を m 、スケジューリング数を n とすると、勤務スケジュール作成問題は m 行 n 列の表に、シフトの種類を割当て問題になる。この問題を解く際には、2 種類の制約を考慮する必要がある。1 つは、各日の必要出勤人数等の必ず遵守する絶対制約で、もう 1 つは、スタッフの休日希望や公平性等の必ずしも満たす必要のない考慮制約である。通常、両者を同時に満たすことは困難なため、絶対制約を遵守することを前提として、破られている考慮制約にペナルティを与え、その総ペナルティを最小にする勤務スケジュールを求めることが一般的である。解法に関しては、実務を考えると、最適な勤務スケジュールを長時間かけて求めることよりも、短時間である程度良い解を求める方が好ましい[3][4][7]。

現在、大規模なこれらの問題を解く商用ソルバーは存在し、コストがかけられる場面では適用が可能となっている[5][6]。一方、中小の職場では、大規模な職場以上に個人個人や職場特有の条件がより重要になるため、詳細なカスタマイズが必要になる。しかしカスタマイズには多大なコストや工数を要し、導入への支障となっている面がある。

さらに、カスタマイズせずにソフトウェアを適用しようとする、ユーザは計算機で求めた解をユーザの都合の良いものに修正する作業が必要になる。直接的に計算機で求めた解を用いることは難しく、修正作業は必ず生じると考えてよい。その理由は、モデルから人間の思い描く解が得られる保証はなく、人間の思い描く解と異なる部分を変更しなければ質の良い解が得られないことが多いからである。この修正作業には通常、多大な時間を要し、ボトルネックになりやすいため、実際にあまり導入されていないもう 1 つの要因であると考えられる。そのため、いくら計算機を用いて短時間で勤務スケジュール

を求めたとしても、修正作業に多大な時間を要するならば、作成者の負担は軽減されないのである。つまり、勤務スケジュール作成の作業軽減を考えるならば、計算の効率化ではなく、出力後の修正作業に要する時間の短縮こそが重要となってくるのである。

修正作業に要する時間を効果的に短縮させるためには、2 つのアプローチが考えられる。1 つは、最適化の視点から、計算機を用いて自動で修正しやすい勤務スケジュールを作成するというアプローチである。2 つめは、システムの視点から、修正作業を支援するツールを開発するというアプローチである。

しかし現在、修正しやすい勤務スケジュールとは、どのような特徴を持ち、どのようにして求解すればよいかといった点に関する研究はなされていない。[1][2]では、修正しやすい勤務スケジュールを求めることが重要であると述べられているが、実際の評価には至っていない。そこで、[8][9]では、計算機で求めたいくつかの実問題の解を実際に修正することで、修正作業の難しさを特徴づける要因を調べた。結果、修正しやすい解とは、絶対制約と考慮制約を等価として求めた解、もしくは、ある制約を修正した際に異なる制約が違反するといった制約違反の伝播をおこしにくく、1 つ 1 つの修正作業が容易で、最後まで作業がスムーズで自由度がある解であることが分かった。

また、ツールを使って勤務スケジュールを作成する際に、初期の条件入力が多いことは、ユーザがツールを使用する際の大きなハードルとなる。従来は全ての条件の入力後に計算機で勤務スケジュールを作成するが、ユーザがツールを動かす前にかなりの作業を要することになる。それがユーザに使いにくさを与える要因になると考える。ユーザに使いやすさを実感させるために、少ない入力、かつ、短い時間で勤務スケジュールを作成できるものにするべきであろう。また、修正については、修正時の制約違反チェックなどを手作業で行うと、多大な労力をかけ、ミスもおこりやすい。修正の際に、条件を違反している箇所を瞬時にチェックできれば、ユーザの負担はかなり軽減されると考える。

ここで、多くの現場では Excel は通常業務で使用していることが多く、簡単な操作に慣れているユーザがほとんどであると思われる。また、勤務スケジュールは縦にスタッフ、横に時間軸の表形式で表される。Excel も行と列の表形式になっており、行や列に対する簡単な計算も容易に行える。したがって、Excel は、勤務スケジュール作成のプラットフォームには非常に適していると思われる。

†総合研究大学院大学 <kubo_tk@nii.ac.jp>

‡国立情報学研究所(NII) <uno@nii.jp>

し、また、全ての作業が Excel でできれば、導入も容易であり、ユーザにとっても扱いやすいものになるであろう。

そこで、本研究では、ユーザの修正作業の負担を効果的に軽減するために、少ない入力力でツールを扱うことができ、自動的に満たされていない制約を瞬時にチェックする機能を持った Excel ベースの勤務スケジュール作成ツールを開発した。なお、自動で作成する勤務スケジュールは、[8][9]で得られた知見を考慮したもので、修正しやすい勤務スケジュールとなっている。さらに、このツールを実際の運用で適用し、フィードバックを得た。このツールを適用した職場では、初回のみ必要な設定を除いた入力作業と勤務スケジュール修正作業に要した時間は約 30 分であった。これまでの勤務スケジュール作成時間に要した時間は 2~3 時間であったので、ユーザの作業時間を 7 割減らすことができた。

2. 修正しやすさとは

[8][9]では、実際のホテルの問題を計算機で解いて求めたいいくつかの勤務スケジュールについて、被験者が修正作業を行うという実験を実際に行い、結果を分析し、以下の 2 つの解が修正しやすい解であるということが分かった。また、修正しやすい勤務スケジュールを作成するだけでなく、修正を行う際に、ユーザに、どのように修正を行えば都合よく修正できるかを示せば、場当たり的に修正を行うよりも、修正しやすくなると考える。

(1) 絶対制約と考慮制約を等価として求めた解

絶対制約と考慮制約を等価にしているため、各制約がバランスよく違反していると考えられる。そのため、解が込み入った構造になっておらず、作成者の修正の自由度が高いのだろう。これらの解を得るためには、絶対制約と考慮制約の重みを等価とすれば求めることが可能である。様々な制約が少しずつ違反されている勤務スケジュールなので、修正作業の方針を与えることは難しい。

(2) 制約違反の伝播が少なく最後まで自由度がある解

作成者は修正作業の際、主にある日のあるスタッフのシフトを異なるシフトに変更する作業を行う。例えば、スタッフのシフトが出勤から休みに変更される際、出勤人数が減少することを除くと、制約違反は起こらない。一方、休みから出勤に変更すると、“スタッフは 7 日に一回は休みが必要である”制約や“遵守/禁止パターン”制約など、スタッフに関する制約の違反を起こしやすく、これが制約違反の伝播を多く引き起こす原因であると考えられる。従って出勤から休みに変更するような修正のみですむ勤務スケジュールであれば、このような制約の違反をおこさない勤務スケジュールになると考えられる。そこで、制約違反が生じる際、各日の必要出勤人数が過剰である勤務スケジュールを求める。その際、計算機で求めた勤務スケジュールは各スタッフの休みの回数が少なくなっていると考えられる。そこで、作成者は各スタッフの休みを少しずつ増やす方針のもとで、出勤人数を必要人数にまで減らしていけばよい。

さらに、勤務スケジュール作成者が最後の段階まで修正作業をスムーズに行うために、最後まで自由度の高い

勤務スケジュールを作成する必要がある。ここで、公平性の定義は、企業、作成者によって異なる。スタッフの希望の守り具合などのバランスをとることは作成者の判断で行なうべきものであると考える。そこで、スタッフの希望は全て事前に考慮した解を作成する。最初から希望が満たされていない勤務スケジュールよりも作成者の自由度は高いと考える。

以上より、まず計算機で、絶対制約と考慮制約を区別せずに等価に考え、また、制約違反が生じる際には、スタッフの希望を全て満たし、各日の出勤数が過剰である勤務スケジュールを自動生成する。そして、作成者は、過剰である出勤人数を減らしながら、各スタッフの休みの回数を増やしていく方針のもとで、勤務スケジュールの修正を行えば、勤務スケジュールを短い時間で都合よく修正できると考えられる。

各日の出勤人数が過剰である勤務スケジュールを作成するためには、絶対制約と考慮制約を区別せずに、修正のしやすさの視点から重要度を決定するとよい。そして、遵守/禁止パターンの違反した場合、出勤人数の不足の場合、スタッフの希望を守れない場合は、他の条件が違反した場合に比べ、ペナルティを高く設定すると良いと思われる。

3. 開発ツールの特徴

この節では、開発したツールの特徴を述べる。ツールの主な特徴として、以下の 3 点が挙げられる。

(1) 導入に要する時間が短い

最適化エンジンも Excel で開発しているため、全てを Excel で作業することが可能である。導入についても、ファイルのコピーのみでよく、環境設定などが不要である。

(2) 少ない入力で、ツールを容易に扱える

ユーザに、ツールを容易に扱えることを実感させるために、勤務スケジュールを作成する為の最低限必要な情報の入力のみで、とりあえず勤務スケジュールを作成できるものにした。これをもとに、ユーザが必要と思う制約を追加することで、さらに都合のよい勤務スケジュールを作成できる。また、条件などを入力する際に、ユーザが入力する部分、リストから選択する部分、自動で出力される部分を色分けしており、誤った入力を極力避けられるようにした。さらに、詳細な条件設定も簡単に設定、変更できるため、ユーザは様々なパターンを容易に試せる。ツール上での条件設定の一部を図 1 に示す。

(3) 修正作業の効率が良い

2 節で述べた修正しやすい勤務スケジュールを作成するための指針をもとに勤務スケジュールを作成している。

最適化手法は、焼きなまし法(SA)を使用している。計算機ではシンプルなモデルをシンプルな解法で解き、その後の簡単な修正作業で適用する方針のため、汎用性の高い手法として今回は SA を選択した。近傍は、ある 1 つの変数の値を他の値に変えることによってできる割当の全てを指す shift 近傍と、2 つの変数の値を入れ替えることに

よって得られる解集合である swap 近傍を用いた[3][4]。また、修正の際に、手作業で制約違反部分をチェックすると、それだけで多大な時間を費やしてしまい、また制約違反を見落とす可能性も生じる。そこで制約違反を正確かつ迅速にチェックするために、どの制約が現在違反をしているかを瞬時にチェックする機能を設けた。



図 1. ツール上での詳細条件設定画面の一部

次にツールを用いて、勤務スケジュールを完成させるまでの一連の流れを図 2 に示す。はじめに初期設定として、スタッフ、グループ、シフトの定義と簡単な条件設定を行う。次に初期設定をふまえて、勤務スケジュールのフォーマットを作成する。その際に初期設定で設定した条件を考慮したスケジュールが同時に作成される。作成後、詳細な条件設定を行う。ここではスタッフの希望、詳細な出勤必要人数、適切回数、スキル、パターンに関する条件を入力する。次にこれらの条件を考慮した勤務スケジュールを計算機が出力する。最後に、出力された勤務スケジュールに対し、ユーザが修正作業を行う。

4. 実例説明

開発したツールの有効性を示すために、某ホテルでツールを適用し、実際の勤務スケジュールを作成した。

一般的にホテルや看護師の職場などの業務形態は、2交代制や3交代制、または両方が混在している場合がほとんどである。対象ホテルは2交代制と3交代制が混在している。

また、スタッフのスキルレベルでは、ベテラン社員、普通社員、新人社員、アルバイトのグループに分かれている。各マスに割当てられるものであるシフトには日勤(A)、夜勤(N)、準夜勤1(C1)、準夜勤2(C2)、夜勤明け(O)、休み(H)があり、もしj日がNならば、j+1日は必ずOとなる。2交代制の場合、Aがj日の9:30~18:00、Nがj日の17:30からj+1日の10:00までとなる。3交代制では、Aは2交代制の場合と同様の時間帯を指し、C1がj日の17:30~23:30、C2がj-1日の23:30~j日の10:00までとなる。アルバイトはC1又はHのみ割当可能であるため、アルバイトが入る場合のみ2交代制と3交代制が混在する。ベテラン社員、普通社員、新人社員はA、N、C2、H、Oが割当可能である。

勤務スケジュールを作成する際には様々な制約を考慮する必要がある。以下に絶対制約(H1)~(H8)と考慮制約(S1)~(S3)を述べる。また(H1)~(H3)は各スタッフに関する労働条件に関する制約、(H4)~(H7)は各日における業務に必要な出勤人数に関する制約、(S1)~(S3)はなるべく守りたい制約である。なお、(H6)A、(H6)Bはどちらか一方のみ満たせばよい。これらの制約は、ほとんどの職場で現れる一般的なものである。

ツール上でのこれらの制約の設定方法は、例えば制約(H1)の場合、図1の「X日間にシフトYがZ回以上ある」欄で、Xに8、YにH、Zに1とすることで、制約を表現できるようになっている。他の制約についても同様の方法で設定できる。

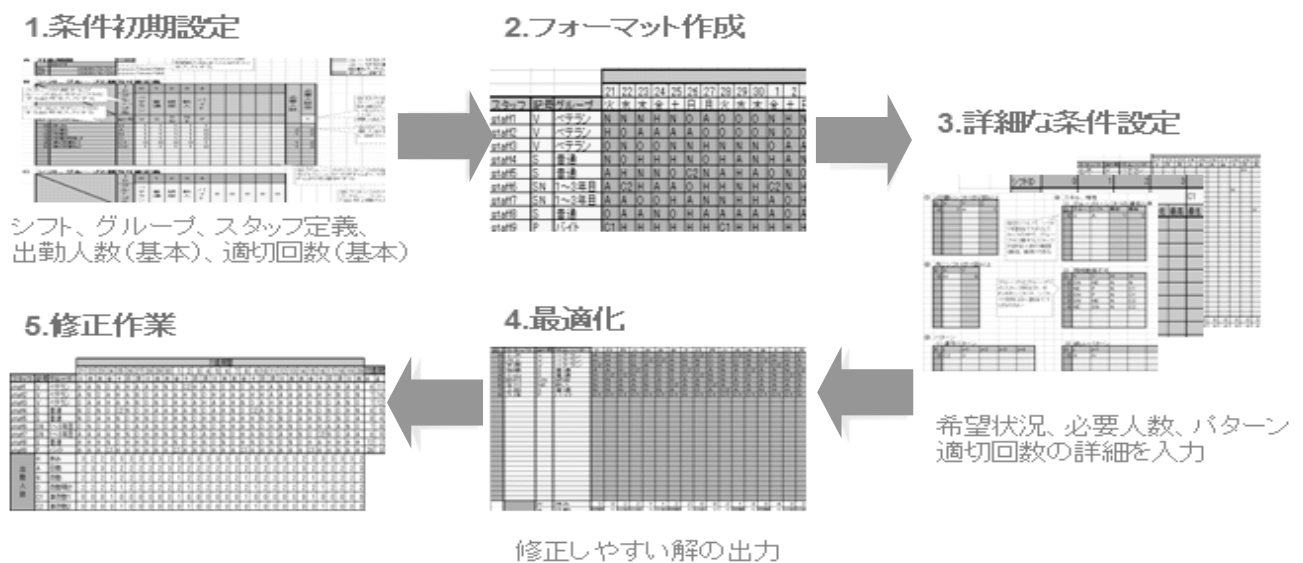


図 2. 勤務スケジュール作成までの一連の流れ

絶対制約

- (H1) 全員が8日に一回は休みを取る
- (H2) 全員が月に6回以上休みを取る
- (H3) 終了から次勤務まで8時間以上必要である
(満たさないパターンは禁止)
- (H4) 日勤は必ず2人の社員が出勤
- (H5) 日勤は1人以上のベテラン社員が出勤
- (H6)A 夜勤は必ず社員が2人出勤
- (H6)B j 日の準夜勤1にアルバイトが1人いるならば社員が j 日の夜勤に1人, $j+1$ 日の準夜勤2に1人必要
- (H7) 新人社員とアルバイトが同時に勤務不可
- (H8) 各スタッフは各日に1つのシフトが割当てられる

考慮制約

- (S1) スタッフの希望をなるべく遵守したい
- (S2) スタッフのシフトを適切な回数にしたい
- (S3) 嫌なシフトパターンはなるべく避けたい

一般的には、絶対制約と考慮制約を区別するが、本研究では、絶対制約と考慮制約の区別は行わない。もし、制約(H3), (S1)を違反した場合は、大きなペナルティを与える。さらに、(H4), (H5), (H6)については、出勤人数が不足の際には大きなペナルティを与え、逆に過剰な場合のペナルティは小さくしている。また、本来、絶対制約となる(H1), (H2)を違反した場合のペナルティは小さく設定した。その結果、制約違反が生じる場合、スタッフの希望は遵守しつつ、出勤人数が過剰で、各スタッフの休みの回数が少なくなっている勤務スケジュールを求めることが可能になると考える。

5. 適用結果

今回開発したツールを用いて、実際に、勤務スケジュール作成者が、勤務スケジュールを作成した。尚、作成者は、Excelの一般的な操作は行える。スタッフ数は9人、対象期間は1ヶ月(30日)である。

(1) ユーザの条件設定作業について

初期設定と初期条件を考慮した勤務スケジュール作成までには約10分を要した。スタッフの希望、詳細な出勤必要人数、適切回数、スキル、パターンに関する条件の設定には約20分を要した。ここまでの作業は、一度設定してしまえば、今後変更する必要がない情報が多いため、初回以降はここまでの作業時間は極力抑えることができ

る。入力データのスタッフの希望に関しては、各日の各スタッフに対応するマスに希望するシフトを入力する形式である。スタッフの希望の申請は54箇所、全体の2割に相当した。スタッフの希望の申請状況を図3に示す。

(2) ユーザの修正作業について

自動で勤務スケジュールを作成する処理に約1分費やした。スタッフの希望は遵守しつつ、出勤人数が過剰で、各スタッフの休みの回数が少なくなっている勤務スケジュールに対するユーザの修正作業時間は約25分であった。これまでの勤務スケジュール作成作業時間は2~3時間要していた。ツールを適用することで、初回のみ必要な設定を除いた入力作業と勤務スケジュール修正作業に要した時間は約30分となり、ユーザの作業時間を7割減らすことができた。修正作業を経て完成した勤務スケジュールを図4に示す。

(3) ユーザからのその他の意見

瞬時に制約をチェックできる機能は修正を行うにあたって非常に便利で、さらに、条件設定を簡単に入力、変更できるため、様々なパターンを試せておもしろいという意見があった。これらの意見からも、ユーザが使いやすさを実感し、容易にツールを扱うことができたことがうかがえる。

(4) 考察

本ツールでは中小規模のスタッフを対象としているため、スタッフが100人いるような大規模な勤務スケジュールの作成は考えていない。しかし、スタッフの人数が多くなれば、問題規模も大きくなる。良質な解を求めるためには、問題の規模に応じて、SAのパラメータである温度調整などを自動で行えるようにする必要があると考えている。

また、勤務スケジュール作成問題を抱える中小の現場は、ホテルや病院以外にも非常に多いであろう。例えば、喫茶店などは、ホテルや病院などとは異なり、アルバイトの人数が非常に多いケースが頻繁にある。アルバイトが希望を申請する頻度は非常に高く、またこれらの条件が非常に重視されるため、通常のホテルや病院などの問題とは少し異なる構造であると思われる。本ツールでは、制約違反が生じる場合、スタッフの希望は遵守しつつ、出勤人数が過剰で、各スタッフの休みの回数が少なくなっている勤務スケジュールを求めているため、たとえ、

			対象期間																															
			21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
スタッフ	記号	グループ	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水		
staff1	V	ベテラン			N	O	A	H				H	N	O					N	O	N	O	A									N	O	A
staff2	V	ベテラン																	H														H	H
staff3	V	ベテラン	O											H							H													
staff4	S	普通										A								H	N	O					H	H	N	O				
staff5	S	普通								N	O									O														
staff6	SN	1~3年目	O		O							N	O		H						H						N	O						
staff7	SN	1~3年目					H	N	O			H	N	O													H							
staff8	S	普通																															H	
staff9	P	バイト				C1									C1												C1						C1	

図3. スタッフの希望申請状況

スタッフの数が多このような現場においても、作成者の修正作業の負担を効果的に軽減させることは可能であると考えている。

6. まとめと今後の課題

修正作業を前提とした Excel ベースのツールを作成し、実際の職場で適用した。その結果、これまで勤務スケジュール作成に 2~3 時間要していた作業を、初回のみ必要な設定を除いた入力作業と勤務スケジュール修正作業に要した時間は約 30 分となり、ユーザの作業時間を 7 割減らすことができた。今後の課題は、他の職場でツールを適用し、より汎用性の高いものにする事である。

また、勤務スケジュール作成問題とは異なる問題、例えば、スポーツに関する地域のリーグ戦スケジュール作成問題、時間割作成問題、人員配置問題などを抱えている中小の現場は少なくない。しかし、勤務スケジュール作成問題と同様に、これらの問題を解決するシステムは、あまり導入されていないと思われる。本研究でのアプローチ、つまり、計算の効率化ではなく、出力後の修正作業に要する時間を効果的に短縮させることに重点をおいたアプローチは、これらの問題についても有効であると考えている。

開発ツールは、以下のサイトからダウンロードできる。
<http://research.nii.ac.jp/~uno/staffschedule.html>

参考文献

[1] 池上 敦子, “ナース・スケジュールリング—調査・モデル化・アルゴリズム—,” 統計数理第 53 巻第 2 号, P231-P260, 2005.
 [2] Ikegami A. and Niwa A, “A sub-centric model and approach to the nurse scheduling,” Mathematical Programming 97, 517-541, 2003.
 [3] Nonobe K. and Ibaraki T, ”A tabu search approach to the constraint satisfaction problem as a general problem solver,” European Journal of Operational Research, Vol.106, 599-623, 1998.
 [4] 野々部 宏司, 茨木 俊秀, “汎用アルゴリズムとしての CSP (制約充足問題に対するタブー探索アプローチ),” 数理解析研究所講研録 1015 巻, P124-136, 1997.
 [5] 新田 俊博, 嶋田 佳明, 田辺 隆人, “実務的なシフトスケジュール作成問題へのメタヒューリスティクスアルゴリズムの応用,” 第 19 回 RAMP シンポジウム論文集, P131-137, 2007.
 [6] 嶋田 佳明, 新田利博, 多田 明功 “汎用的なシフトスケジュールリングツール~モデリングとその開発~, ” スケジュールリング・シンポジウム 2008 講演論文集, P177-P182, 2008.
 [7] 柳浦 睦憲 茨木 俊秀, “組合せ最適化-メタ戦略を中心として-”朝倉書店, 2001.
 [8] 久保 琢磨, 宇野 毅明, “スタッフスケジュールリングにおける修正しやすい解を知る為の実験とその考察,” スケジュールリング・シンポジウム 2008 講演論文集, P123-P126, 2008.
 [9] 久保 琢磨, 宇野 毅明, “スタッフスケジュールリングにおける修正しやすさを考慮した解の分析,” 数理解析研究所講究録(1629) P56-58 2009/2”

ID	スタッフ	記号	グループ	対象期間																				対象期間内のシフト回数															
				21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	H	A	N	O	C1	C2
0	staff1	V	ベテラン	H	A	N	O	A	H	A	A	H	N	O	C2	H	A	N	O	N	O	A	H	A	N	O	N	O	A	H	A	A	6	11	6	6	0	1	
1	staff2	V	ベテラン	A	N	O	A	H	A	N	O	A	A	A	H	N	O	H	A	A	A	H	H	A	A	A	A	A	H	N	O	N	7	14	5	4	0	0	
2	staff3	V	ベテラン	O	A	A	H	A	A	N	O	N	O	H	A	A	H	A	A	H	N	O	A	H	H	A	A	H	N	O	A	N	O	7	12	5	6	0	0
3	staff4	S	普通	N	O	N	O	C2	N	O	H	A	A	H	A	N	O	A	H	N	O	C2	A	N	O	H	H	N	O	N	O	H	N	6	5	9	8	0	2
4	staff5	S	普通	N	O	A	H	N	O	H	N	O	H	H	N	O	H	N	O	A	H	H	N	O	N	O	A	H	N	O	N	O	A	8	4	9	9	0	0
5	staff6	SN	1~3年目	O	N	O	H	N	O	A	A	N	O	A	H	A	N	O	H	H	A	N	O	H	H	N	O	A	H	A	A	N	O	7	8	7	8	0	0
6	staff7	SN	1~3年目	A	A	A	A	H	N	O	H	H	N	O	N	O	A	H	N	O	H	A	N	O	A	H	N	O	C2	N	O	A	A	6	9	7	7	0	1
7	staff8	S	普通	H	H	H	N	O	H	H	N	O	A	N	O	H	N	O	N	O	H	N	O	N	O	N	O	H	A	H	H	H	12	2	8	8	0	0	
8	staff9	P	バイト	H	H	H	C1	H	H	H	H	H	C1	H	H	H	H	H	C1	H	H	H	H	H	H	C1	H	H	H	H	H	26	0	0	0	4	0		
出勤人数	H	休み		3	2	2	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2		
	A	日勤		2	3	3	2	2	2	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3		
	N	夜勤		2	2	2	1	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
	O	夜勤明け		2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2		
	C1	準夜勤1		0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
C2	準夜勤2		0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
必要人数①	H	休み																																					
	A	日勤		2	3	3	2	2	2	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3		
	N	夜勤		2	2	2	1	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
	O	夜勤明け		2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
	C1	準夜勤1		0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
C2	準夜勤2		0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

図 4. 完成した勤務スケジュール