

## 中小規模スタッフスケジューリング問題における 調整の容易なスケジュール作成に関する研究

久保 琢磨\* 宇野 毅明\*\*

\*総合研究大学院大学複合科学研究科情報学専攻

\*\*国立情報学研究所

近年、組合せ最適化の実社会への適用を目的とする動きが活発となっているが、一般ユーザが容易に最適化を利用できるまでには至っていない。本研究方針は、中小規模の問題に対し、ユーザ自身が容易に最適化を利用できるしくみづくりとしている。今回はスタッフスケジューリングに焦点をあて、勤務スケジュール作成者のスケジュール作成作業の負担を軽減させる為に、あるビジネスホテルが抱える問題を対象に、スケジュール作成者にとって、容易に調整可能なスケジュールを作成することを目的とする。

### Making an easily adjustable schedule to small or middle scale staff scheduling problem

Takuma Kubo\* Takeaki Uno\*\*

\*The Graduate university for Advanced Studies (SOKENDAI)

\*\*National Institute of Informatics

Recently, the movement to apply combinatorial optimization to practical problem has been active. However, general users can't easily use optimization. The direction of our research is that general users can lightly use optimization. In this paper, we focus the problem called staff scheduling and we want to reduce workload of a staff which make a schedule for every staff. Then, we aim to make an easily adjustable schedule to practical problem which the staff in certain business hotel has.

#### 1. 背景と目的

組合せ最適化問題は実社会の様々な場面で現れる。その為近年、組合せ最適化の実社会への適用を目的とする動きが重要視されている。代表的な研究のひとつに、いくつかの標準問題を定義し、それらに対する問題解決エンジンを準備するアプローチがあり、大きな成果をあげている。また最適化ソルバーを保持するいくつかの企業では、顧客の抱える問題を解決する為に、顧客とのヒアリングを繰返し行い、顧客の要求に合致したシステムを開発するアプローチをとっている。

インターネットが普及し、莫大な情報が存在する現在、ユーザは一般的にサーチエンジンを用いて必要な情報を取得する。ユーザは何か欲しい情報があるならば、関連キーワードをユーザ自身が入力し、もし欲しい情報が見つからなければ、別のキーワードを入力する等、情報入手の為に努力を要する。なぜ Web による情報検索が一般的になったのか？これはユーザ自身がごく簡単に作業ができ、望んでいるものが入手できなかった際でも、ユーザ自身がキーワードの変更する等簡単な工夫をすることが容易に可能である為と考える。

これに対し最適化は、一般ユーザが容易に最適化を

利用できるまでには至っていない。これらの問題は複雑で、ユーザ自身が問題発見、モデル、手法の開発を行なうことができないこと、またそれを支援するための仕組みが存在しないことが原因と考えられる。そこで、ユーザの要求に合致したシステムを開発する為に、多大なコストを費やしてもユーザとシステム開発者との幾度もの打合せを行なう必要がある。資金が豊富であれば、または最適化による利益が大きい大規模な問題を抱える企業であれば、多大なコストを費やすことが可能でなる。しかし、資金が少ない、または最適化による利益が少ない中小規模の問題を抱える企業では、多大なコストを費やすことは難しい。この場合、問題を抱えているにも関わらず、ユーザ自身の要求をすべて満たすようなシステムは開発することができない。

そこで本研究方針は、中小規模の問題に対し、ユーザ自身が容易に最適化を利用できるしくみづくりとする。ユーザが必要事項を定義し、入力データを前提に、計算機が解を探索、出力結果をもとに、ユーザが調整を行うことを考えている。大規模問題に対してはより良い解を得る為に手法を複雑にする必要がある。一方中小規模の問題の際、簡潔な手法で問題をある程度解

決可能であるとする。簡潔な手法であるならば汎用性も向上し、ユーザにとってわかりやすいものとなる。

今回はスタッフスケジューリングに焦点をあてる。スタッフスケジューリングとは、企業において各スタッフが公平な勤務となるように、労働条件やスタッフの希望等の条件を考慮したスケジュールを作成する技術で、企業規模問わず、多くの企業が抱えている。そこで、勤務スケジュール作成者の作成作業の負担を軽減させる為に、今回はあるビジネスホテルが抱える問題を対象に、スケジュール作成者にとって、容易に調整可能なスケジュールを作成することを目的とする。

## 2. 問題の難しさ

様々な条件を考慮し、勤務スケジュールを迅速に求める問題は簡単ではない。例えばスタッフが 10 人、期間が 1 ヶ月(30 日)とする。各スタッフに割当可能なシフト(夜勤、日勤、休み)が 3 種類あり、スタッフ  $i$  の  $j$  日のシフトはその中から 1 つ割当てられるとすると、スケジュールパターンは 3 の  $(10 \times 30)$  乗となる。これはスタッフの人数が増加するだけで、スケジュールパターンは大幅に増加することを表す。また、膨大なパターンからなんらかの評価値が最も良いパターンの発見には、基本的に網羅的な探索が必要で、この種の問題は NP 困難な問題に属する。

制約条件には、ある日の日勤が 5 人以上必要等の業務上必須の絶対制約と、スタッフの休日希望日なるべく尊重するといった考慮制約がある。通常、両者を同時に満たすことは困難な為、守れない考慮制約にペナルティを与え、総ペナルティを小さくするスケジュールを求めることが一般的にとられている。解法に関しては、実務において最も良いスケジュールを長時間かけて求めることより、短時間である程度精度の良いものを求めることが必要とされている為、それを可能とする手法を提案している研究が多い。

これらの研究の多くでは、各スタッフの休日希望日が 1 つかなえられないとペナルティを  $x$ 、2 つかなえられないと  $2x$ 、連休が取得できないと  $y$ 、等と独立に付加する。しかし、各制約の相対的な評価は困難であるばかりか、ある種の制約に関しペナルティのとりうる値の範囲が大きいと、その制約を良くする方向ばかりに探索が進んでしまい、探索が偏ってしまう。またこの分野の研究はこれまでににおおいになされているが、ほとんどの研究はペナルティである目的関数の値を最小化することを目標としており、実際にどれほど実務で役立っているかに焦点を当てていない。本研究では、設定した目的関数を改良化していくことで、実際に作成者をどれだけ支援することができたかに注目する。

## 3. 問題概要

各スタッフに関する情報、グループ(ベテラン社員、普通社員、新人社員、アルバイト)、各グループの割当可能シフト、各スタッフの希望を定義し、以下の制約を考慮した 1 ヶ月の勤務スケジュールを生成する問題を扱う。ここで、対象ホテルの勤務形態について説明する。シフトは日勤(A)、夜勤(N)、準夜勤 1(C1)、準夜勤 2(C2)、夜勤明け(O)、休み(H)がある。A は  $j$  日の 9:30~18:00、N は  $j$  日の 17:30~ $j+1$  日の 10:00、C1 は  $j$  日の 17:30~23:30、C2 は  $j-1$  日の 23:30~ $j$  日 10:00 の勤務となり、 $j$  日の勤務が N ならば、 $j+1$  日は必ず O になる。2 交代制の場合は、日勤(A)が 9:30~18:00、17:30 から翌日 10:00 までが夜勤(N)となる。3 交代制の際、日勤は 2 交代制の場合と同様の時間帯を指し、準夜勤 1(C1)が 17:30~23:30、準夜勤 2(C2)が 23:30 から翌日 10:00 までとなる。またアルバイトは C1 または H のシフトのみ割当可能である為、アルバイトが入る場合のみ 3 交代制となる。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。次に絶対制約(H1)~(H8)と考慮制約(S1)~(S3)を示す。尚、絶対制約(H6)は(H6)A か(H6)B のいずれかを満たす。

- ・全員が 7 日に一回は休みを取る . . . (H1)
- ・全員が月に 6 回以上休みを取る . . . (H2)
- ・終了から次勤務まで 8 時間以上必要 . . . (H3)
- ・日勤は必ず 2 人の社員が必要 . . . (H4)
- ・日勤は 1 人以上のベテラン社員が必要 . . . (H5)
- ・夜勤は必ず社員が 2 人出勤 . . . (H6)A
- ・ $j$  日の C1 にアルバイトが 1 人いる . . . (H6)B
- ならば社員が  $j$  日の N に 1 人、 $j+1$  日の C2 に 1 人
- ・新人社員とアルバイトが同時夜勤不可 . . . (H7)
- ・各スタッフは、各日に、1 つのシフト . . . (H8)
- ・スタッフの希望をなるべく遵守したい . . . (S1)
- ・厳しいシフトパターンを少なくしたい . . . (S2)
- ・各社員の準夜勤回数を均等にしたい . . . (S3)

上記制約について、(H1)~(H3)は労働条件に関する制約、(H3)~(H6)は勤務の質を守る為の制約を示し、他の企業においても一般的な制約である。他との相違点は、アルバイトと社員が混在しており、同時勤務の際の細かな制約が設けられている点があげられる。

スタッフが  $j$  日のシフトが休み(H)であったとしても、 $j+1$  日のシフトが準夜勤 2(C2)であれば、前日の夜から出勤する必要がある為、心理的に普通の休みより損した気持ちを与えてしまう。そこで、 $j$  日が H、 $j+1$  が C2 であるシフトパターンをなるべく少なくすることを考える(考慮制約 S2)。また、スタッフからの不満や緊急時による部分的なスケジュールの修正が必ず生

じる。その際に調整が行ないやすいスケジュールを作成しておくことも重要である。

#### 4. アプローチ

ユーザが条件を入力すると、公平さを保ちつつ、実際に納得して適用できるスケジュールを容易に作成可能なしくみをつくる。まず運用がどれだけ楽になったかが求められる。その要因として入力容易であるか、出力スケジュールが作成者の意図したものに近いか、かつ調整には時間がかからないものであるかが求められる。スタッフからの不満や緊急時による部分的なスケジュールの調整が必ず生じる為、この作業を容易に行えることは重要である。これらを考慮する為以下の3点についても検討する必要がある。

##### ① スケジュール修正の容易さ

スケジュールを作成したところで必ず何人かのスタッフから不満が生じる。スケジュール修正の際、スケジュール表を壁に貼り、それを見ながら何人かのスタッフで話し合いながら対応していくケースが多い。中小規模な問題に対し、再度自動でスケジュール修正をすることは、条件等を入力している時間で、既に話し合いで解決することもしばしばある為あまり効果がないと思われる。そこで、自動修正に関する解法は対象外とする。自動修正の精度向上のための解法よりも、修正しやすいスケジュールを求めることを目標とする。

##### ② 公平さ

企業によって公平さは異なる（この場合 C2 の数を同じにすることが必要(考慮制約 S3)）。公平さ、条件等の定義は各企業で行なうべきである。短期的な公平さを常に維持することは限界があるので、長期的に公平さを守れるものでありたい。そこで、各企業にて公平さの定義を行い、それに基づき作成者が各スタッフに対してどれだけ守らせたいかの希望を設定し、その希望と実際のスケジュールの差異を目的関数とする。

##### ③ 理由づけ

なぜ今回このスケジュールを採用したかをスタッフに説明でき、次回は誰を優先すべきか明確にできるとよい。上記で述べた目的関数にすることで、ある月の勤務スケジュールが公平に作成されることが不可能だったとしても、どれだけ満たせなかったかが明確な為、翌月以降はある考慮制約の重要度を少し高く設定することをスタッフに事前に伝えておくことが可能となる。

#### 5. 評価

従来、評価は短時間で目的関数をどれだけよくできたかが目安となった。本研究では設定した目的関数を改良化していくことで、実際に作成者をどれだけ支援することができたか、つまり作成者の調整の手間をど

れだけ省くことができたかが評価となる。

作成者の要件は、公平であることを明確に説明できるものが欲しいことを挙げている。また研究上の課題として、問題の難しさの項でも述べたが、ペナルティの付加で制御することは困難であるので、より作成者がわかりやすい形にする必要がある。

両者を満たす為に考慮制約について、ランクという概念を定義する。ランクは5段階で設定する。例えば考慮制約(S1)におけるランクの定義例として、希望日が100%満たされれば、ランク1、100%未満~75%ならばランク2、75%未満~50%ならばランク3、50%未満~25%ならばランク4、25%未満~0%ならばランク5とする。考慮制約(S2)の例として、j日がHかつj+1日がC2であるパターンやC2が連続することは嫌がられる為、そのパターンが0回ならばランク1、1回ならばランク2、2回ならばランク3、3回ならばランク4、それ以上ならばランク5とする。なお、考慮制約に関するランクの定義は作成者が意図したように設定可能の為、作成者の希望を考慮しやすい形である。

作成者が各考慮制約においてランクを定義した後、各スタッフがある考慮制約については最低このランク以上守りたいという妥協ランクの設定を行う。スケジュールの評価は考慮制約h、スタッフiの妥協ランクを $Sh_i$ 、実際に求められた勤務スケジュールでの制約h、スタッフiのランクを $Th_i$ とした時、 $\max\{Th_i - Sh_i, 0\}$ の総和を最小にすることである。これにより出力スケジュールが作成者の意図とどの程度違いがあるかを示すことが可能となる。また、ある月で、公平に作成することが不可能であっても、どの程度満たせなかったかが明確な為、翌月以降はある考慮制約は良い妥協ランクを設定してあげることで、長期間での公平さを保つことが可能となる。各制約に対する評価の範囲も同じになるので、探索の偏りもおこらないと考える。

#### 6. 解法

実社会への適用の為には、長時間かけて大域最適解を求めることよりも短時間で精度のよい解を求めることが望まれる。また大規模問題に対しては、より良い解を得る為に手法を複雑にする必要がある。しかし、問題が中小規模の場合、簡潔な手法で問題をある程度解決可能であると考える。簡潔な手法であるならば汎用性も向上し、またユーザにとってもわかりやすいものとなる。そこでまず、局所探索法を考える。しかし、この手法は簡潔である反面、局所最適解に陥ってしまう可能性がある。より効率のよい解法として、焼きなまし法(Simulated Annealing)を用いる。焼きなましとは、固体に熱を加えて溶解し、その状態からゆっくり

と温度を下げることで、安定した分子構造をもつ品質のよい物質を生成するプロセスで、焼きなまし法(以下SA)はこの挙動と組合せ最適化問題を対応させ、温度を少しずつ下げ、安定した解を得る為の手法である。近傍解を生成し、解のよさに応じた遷移確率を設定し、それに従い次の解を選択する。改善解であっても遷移確率を与えることで、局所最適解からの脱出を図ることが可能であることがSAの特徴である。温度が高い状態では改善解であっても、解を受理する確率は高く、温度が低くなるにつれて、解が改善された時のみ解を受理する為、局所探索法のような振る舞いをする。初期解生成はランダムに生成、近傍解生成はシンプルに行う為、あるスタッフのある日の勤務を別のスタッフに割り当てることのみ行なう。もし絶対制約を違反すれば大きなペナルティを付加する。

### 7. 実験

最適化の研究では絶対制約を遵守し、考慮制約をなるべく守ることを目標とするが、後の調整を考えると絶対制約の遵守された解の作成より、絶対制約は多少違反しても考慮制約を遵守する方が調整しやすいとも考えられる。そこで、SAを用いて絶対制約と考慮制約の重要度を変化させた各場合の問題を解き、次に出力スケジュールに対し、実際に調整作業を行う。対象は、スタッフが9名、期間は30日とする。図1は考慮制約より絶対制約に重要度をおく際の作成スケジュールの一例を示し、図2は絶対制約より考慮制約の方が重要度をおく際の作成スケジュールの一例を示す。

	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	4	C2
S1	H	C2	A	A	A	A	A	H	C2	A	H	A	A	H	C2	N	O	A	N	O	H	N	O	A	A	A	A	A	H	7	8	
S2	A	A	A	A	H	C2	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	7	7	
S3	H	N	O	N	O	N	O	H	N	O	N	O	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	6	3
S4	H	H	A	N	O	H	C2	N	O	N	O	C2	H	H	N	O	N	O	N	O	H	N	O	A	A	A	A	A	A	9	2	
S5	N	O	N	O	H	N	O	H	C2	N	O	N	O	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	8	3
S6	H	C2	H	C2	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	9	5	
S7	A	A	N	O	H	A	A	N	O	H	A	A	N	O	H	A	A	N	O	H	A	A	N	O	H	A	A	N	O	6	2	
S8	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	2	0	
S9	C1	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	2	0	

図1.絶対制約を重要視した際の出力結果の一例

	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	4	C2
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	5	0	
S2	A	A	A	A	H	C2	H	N	O	N	O	H	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	2	9	
S3	H	N	O	N	O	N	O	H	N	O	N	O	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	7	3
S4	N	O	H	A	N	O	A	H	A	N	O	H	C2	N	O	N	O	H	C2	C2	C2	H	N	O	N	O	N	O	6	4		
S5	H	C2	C2	H	C2	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	9	4
S6	H	H	C2	H	A	N	O	A	A	A	H	C2	A	N	O	H	C2	C2	A	H	C2	A	H	C2	A	H	C2	A	H	9	4	
S7	H	N	O	N	O	H	N	O	N	O	H	N	O	N	O	H	A	H	A	H	A	H	A	H	A	H	A	H	A	H	9	3
S8	H	H	C1	C1	C1	H	H	H	C1	C1	H	H	H	C1	C1	H	H	H	C1	C1	H	H	H	C1	C1	H	H	H	1	0		
S9	C1	C1	H	H	H	H	H	H	C1	H	H	H	H	C1	H	H	H	H	C1	H	H	H	H	C1	H	H	H	H	2	1	0	

図2.考慮制約を重要視した際の出力結果の一例

図1, 図2どちらも目的関数の評価は450であった。丸で囲んだ部分は希望日を満たしていない箇所、四角部分は絶対制約を違反している箇所である。図3は図1の結果の調整後のスケジュール、図4は図2の調整後のスケジュールを示す。尚、灰色部分は希望を示す。

	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	4	C2
S1	H	C2	A	A	A	A	A	H	C2	A	H	A	A	H	C2	N	O	A	N	O	H	N	O	A	A	A	A	A	H	7	4	
S2	A	A	A	A	H	C2	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	7	2	
S3	H	N	O	N	O	N	O	H	N	O	N	O	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	6	4
S4	H	H	A	N	O	H	C2	N	O	N	O	C2	H	H	N	O	N	O	N	O	H	N	O	A	A	A	A	A	A	9	2	
S5	N	O	N	O	H	N	O	A	N	O	H	C2	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	7	1
S6	H	H	C2	H	C2	A	N	O	H	C2	C2	A	A	A	H	N	O	A	A	A	A	A	A	A	A	A	A	A	H	8	5	
S7	H	A	H	N	O	A	A	H	A	N	O	N	O	H	N	O	A	N	O	A	N	O	A	N	O	A	N	O	A	N	7	1
S8	H	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	H	2	0	
S9	C1	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	C1	H	2	0	

図3.図1の調整後のスケジュール

	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	4	C2
S1	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	7	1	
S2	A	A	A	A	H	C2	H	N	O	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	8	3	
S3	H	N	O	N	O	N	O	H	N	O	N	O	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	7	3
S4	N	O	H	A	N	O	A	H	A	N	O	H	C2	N	O	N	O	H	C2	C2	C2	H	N	O	N	O	N	O	6	2		
S5	H	C2	C2	H	C2	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	C2	A	H	9	4
S6	H	H	C2	H	A	N	O	A	A	A	H	C2	A	N	O	H	C2	C2	A	H	C2	A	H	C2	A	H	C2	A	H	9	2	
S7	H	N	O	N	O	H	N	O	N	O	H	N	O	N	O	H	A	H	A	H	A	H	A	H	A	H	A	H	A	H	9	3
S8	H	H	C1	C1	C1	H	H	H	C1	C1	H	H	H	C1	C1	H	H	H	C1	C1	H	H	H	C1	C1	H	H	H	1	0		
S9	C1	C1	H	H	H	H	H	H	C1	H	H	H	H	C1	H	H	H	H	C1	H	H	H	H	C1	H	H	H	H	2	1	0	

図4.図3調整正後のスケジュール

色が濃い箇所が調整箇所であり、丸で囲んだ部分は希望日を満たせなかった箇所である。手作業での調整に図3には約20分、図4には約10分の時間を要した。絶対制約を2箇所満たしていなかったにも関わらず、考慮制約S1の希望日をすべて満たしていた図2のスケジュールの方が、調整作業が容易であった。この一例に限らず、希望日をほぼ満たしている際のスケジュールに関する調整作業は比較的行いやすい傾向にあった。しかし現在、調整のしやすさを図る指標は実際の調整作業時間のみで、調整のしやすいスケジュールやそれを計る指標の定義にはまだ至っておらず、今後の課題である。また図4における公平さについて、Staff5のC2が多くなっているが、翌月はStaff5の考慮制約S3に関する妥協ランクを1と設定し、重点的に考慮することで翌月へ公平さを維持することが可能となる。また翌月への重要視する点を明確にできたことがスケジュールの採用の理由付けにもつながる。

### 8. おわりに

あるホテルが抱える問題を対象に、SAを用いて作成者の希望と実際の差異を指標とし、長期的な公平さを維持することが可能で、かつ容易に調整可能なスケジュールの作成を行なった。今後の課題として、調整のしやすさやその指標の定義、対象ホテルでの実業務で利用可能なツールの作成、そして汎用性のあるしくみにすることがあげられる。

### 参考文献

- 1) 茨木 俊秀, “問題エンジンとしてのメタヒューリスティクス・アルゴリズム,” 第19回 RAMP シンポジウム論文集, P117-130, 2007.
- 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-,” 統計数理第53巻第2号, P231-P260, 2005.