

鉄道乗務員スケジューリング問題に対する解法とその評価

Solution Methods for Railway Crew Scheduling Problem and their Evaluation

今 泉 淳

(Jun Imaizumi)

福 村 直 登

(Naoto Fukumura)

森 戸 晋

(Susumu Morito)

鉄道乗務員スケジューリング問題に対する解法とその評価

今 泉 淳
福 村 直 登
森 戸 晋

- 1 目的
 - 2 乗務員スケジューリング問題とは
 - 2.1 問題の概要
 - 2.2 問題の詳細
 - 3 定式化
 - 3.1 集合被覆問題と集合分割問題
 - 3.2 定式化における問題点
 - 4 最適化のアプローチ: 事前列挙タイプと列生成タイプ
 - 5 各種解法の性能評価
 - 5.1 事前列挙タイプの解法
 - 5.1.1 初期の試み: Caprara ら[1]の近似解法
 - 5.1.2 初期の試みの結果への対策
 - 5.1.3 Wedelin[7]の近似解法
 - 5.2 列生成タイプの方法
 - 5.2.1 初期の試み
 - 5.2.2 列生成ヒューリスティック解法の性能評価
 - 5.2.3 便乗削減のための定式化の修正と工夫
 - 6 考察
 - 7 おわりに
- 参考文献

1 目的

昨今の運輸業界では、より速く確実な移動手段をより安価に提供することが急務となっている。その実現のためには、所有する各種資源をいかにマネジメントするかが鍵となる。その中でも鉄道会社では、移動の手段を客にどのように提供するかを具体的に示すダイヤグラムの策定とともに、資源のマネジメントという側面からは、そのダイヤグラムと車両や乗務員といった鉄道会社が所有する資源とをどのように対応付けるかの計画立案が必要となる。

乗務員は、列車の運行上欠かすことができない存在であるが、労働・勤務上の規則など様々な

ルールに制約され、各列車に必ず乗務員を充当するという前提の下で、限られた人員を効率的に運用する計画を立案することは容易ではない。その一方で、ことに人件費が高騰している現状を考えると、そのマネジメントの重要性が増している。

本稿では日本の鉄道業における乗務員スケジューリング問題を取りあげ、そのモデル化や最適化の方法に関して、著者らのグループがここ数年間で試みたアプローチやその成果をそれらの数値実験の結果を交えて報告する。

2 乗務員スケジューリング問題とは

2.1 問題の概要

乗務員スケジューリング問題とは、所与のダイヤグラムに対して必要な乗務員を制約条件を満足させつつ割り当てる問題である。

ダイヤグラムは、列車の出発駅と出発時刻、途中停車駅その到着・出発時刻、終着駅とそこへの到着時刻などを与える。一方、乗務員は労働上の条件を満たしながら、列車を乗り継ぎつつ、最終的に勤務の開始地（基地）に戻らなければいけない。当然、全ての列車に乗務員が充当される必要があるが、人的資源の効率的な運用も実現したい。

2.2 問題の詳細

前述のように、ダイヤグラムに基づき対象となる列車の本数やそれぞれの始発駅、終着駅、途中停車駅の発着時刻などが与えられる。これらの駅の中で、乗務員が列車を乗り降りできる乗り換え可能駅が決められている。一般に、乗務員はある列車の始発駅から終着駅までを必ずしも乗務しなくてもよく（また必ず乗務できるとは限らない）、乗り換え可能駅で降りて別の列車に乗ってよい。

ただし、基地から出発して基地に戻るまでの間には労働上の様々な制約が存在する。すなわち、いくつかの評価項目に関してそれぞれに設定された上下限を最終的に守るよう、各乗務員はその所属基地のある出発駅から勤務を開始し、乗り換え可能駅で適宜別の列車に乗り換えつつ、最終的に出発駅に戻らねばならない。この上下限は、勤務が1暦日内に収まる日勤と、2暦日に跨る夜勤によっても違う。

乗務員スケジューリングでは、ダイヤグラムに与えられた全列車に乗務員が充当されるよう、定められた「費用」が最小になるような乗り継ぎパターンの組合せを見つけることが目的となる。

3 定式化

問題を定式化するプロセスは、オペレーションズ・リサーチの枠組で問題解決を行う上でもっと

も重要なステップのひとつである。

3.1 集合被覆問題と集合分割問題

伝統的なアプローチとして、問題を集合被覆問題 (*Set Covering Problem*, SCP) もしくは集合分割問題 (*Set Partitioning Problem*, SPP) のいずれかで捉えるのが一般的である。そこで最初に集合被覆問題としての定式化を示す。

まず、列車を乗務員の乗り降りの対象となる最小の単位に分割したものを乗務 (*trip, task*) と称し、勤務の開始駅から元の駅に戻って来るまでの一連の乗務の繋がり (乗り継ぎパターン) を行路 (*pairing, duty*) と呼ぶ。

ここで、与えられた乗務集合を M 、可能な行路案集合を N とし、これらの要素は適当に番号付けがされているとする。また、 a_{ij} を第 j 番目の行路案が第 i 乗務を含むとき 1、さもなければ 0 とし、 $a_j = (a_{1j}, \dots, a_{ij}, \dots, a_{|M|j})^t$ を第 j 番目の行路案に対応する列ベクトル、 c_j を第 j 番目の行路案のコスト、 x_j を第 j 番目の行路案を選択するとき 1、さもなければ 0 となる 0-1 変数とすれば、乗務員スケジューリング問題は以下のような集合被覆問題として定式化できる[1]。

$$(SCP) \quad \min z = \sum_{j \in N} c_j x_j \quad (1)$$

$$\text{s.t.} \quad \sum_{j \in N} a_{ij} x_j \geq 1, \quad \forall i \in M \quad (2)$$

$$x_j \in \{0, 1\}, \quad \forall j \in N \quad (3)$$

この定式化は、便乗 (*deadhead*) すなわち「乗務員が、運転などの業務をせず移動するためだけに列車に乗る」こと陽に許す。航空と比較すると、鉄道では便乗に対する扱いが比較的緩やかなため、集合被覆問題として考えることはさほど不自然ではない。これに対して航空産業では、業務を行う乗務か便乗かによって行路の制約条件やコストが異なる[2]ことなどを踏まえ、集合分割問題として定式化することが多い。

$$(SPP) \quad \min z = \sum_{j \in N} c_j x_j \quad (4)$$

$$\text{s.t.} \quad \sum_{j \in N} a_{ij} x_j = 1, \quad \forall i \in M \quad (5)$$

$$x_j \in \{0, 1\}, \quad \forall j \in N \quad (6)$$

3.2 定式化における問題点

集合被覆問題で定式化すべきか、それとも集合分割問題で定式化すべきかは、一概にはいえない。もちろん、便乗が許されるとはいえ、それが少ないことが望ましいのは鉄道でも同様である。

集合被覆問題による定式化は便乗を陽に許すとともに、ある乗務に複数の行路（乗務員）が割り当てられた際、どの行路にその乗務の便乗を割り当ててを後で決める必要がある。つまり便乗は結果として発生し、厳密にはさらになんらかの判断をする必要がある。

一方、集合分割問題による定式化は便乗を許さないように見えるが、便乗なしで全乗務を一度だけ被覆するスケジュールが存在するかどうかはインスタンスに依存し、一般にその解の存在は保証されない。そのような意味で集合分割問題による定式化は、本来の乗務と便乗を区別した上で行路を考えて全乗務が厳密に一度だけ被覆されることを指向していると考えるべきである。その一方で、一般に集合被覆問題での定式化に比べて実行可能解が見つけにくくなるというデメリットがある。

いずれの場合でも、これら定式化は「予め行路案が与えられている」という前提で書かれているが、実際にはこの行路案をどのように準備するかが問題となる。すなわち、ルールを満足する行路案は膨大な数にのぼり、後述するようにこれを全て列挙することは現実的でなく、乗務員スケジューリングではそれが問題解決における焦点の一つとなる。

4 最適化のアプローチ：事前列挙タイプと列生成タイプ

乗務員スケジューリングで必要となる、定式化における列に相当する行路案の準備のやり方を分類すると、

- 労働上の制約を満たす行路案をバックトラック法などで予め多数列挙し、これらを集合被覆（あるいは分割）問題の列として最適化する方法（以下「事前列挙タイプ」と呼ぶ）
- 集合被覆（あるいは分割）タイプの問題の主問題に対して、その線形緩和問題の最適解が得られるまで列生成子問題を解き続け、それらの列から整数解を得る、あるいは、そこからさらに列生成法を組み込んだ分枝限定法（分枝価格法[3]）に進むもの（以下これらを総称して「列生成タイプ」と呼ぶ）

の二つに分けられる[4]。いずれのアプローチとも一長一短があり優劣がつけ難い。

我々のグループの初期のアプローチでは、後述するように、行路案を列挙して集合被覆問題として解く方法を採用した。これは、列挙に基づく方法が列生成法よりも実装面で容易であり、現実的な応用を考える上では受け入れ易いであろうことを想定するとともに、実行可能解の求めやすさの観点から集合分割問題よりも扱いやすく、行路案を作成する際に便乗を考慮することについて指針を得る意味でも、まず集合被覆問題による定式化から出発するのが妥当と考えたためである。

5 各種解法の性能評価

本節では、我々のグループが事前列挙タイプと列生成タイプの双方の解法に関して数値実験による評価を行った結果に関して報告する。以下では原則として実在する線区のダイヤグラムを使い、現実規模のインスタンスを用いて数値実験を行っている。また、特に断りが無い場合、行路案のコストはその内容にかかわらず、日勤に対して $c_j = 1$ 、夜勤に対して $c_j = 2$ である。これにより目的関数は総勤務日数となる。

5.1 事前列挙タイプの解法

この方法は、前述したように実装を行う観点から比較的容易に実現可能であることもあり、そのような観点から実務での利用が期待できる。

5.1.1 初期の試み：Caprara ら [1] の近似解法

まず我々のグループが最初に試みたのは、バックトラック法によって行路案の列挙を行った上で、Caprara ら [1] の近似解法を適用することであった [5]。この解法はラグランジュ緩和をベースとし、イタリアの鉄道乗務員スケジューリングで好結果を得た解法である。

前述の通りこれに対しては行路案の列挙手順が必要だが、これは行路に関するルールが明示されていればその実装はさほど難しいものではない。ただしその行路案数は、後述するように、なんらかのメカニズムによって制限しないと膨大になることが予備実験によって判明した。そこで、

- 評価項目の値がルール上の許容内であっても現実的でない値である場合は、それを行路案としては生成しない
- 確率的基準によって、行路の望ましさに応じてその生成の可否を決定する

などの方法を併用して、行路案数をある程度制御できるメカニズムを用いて行路案を生成した。

これらの下準備の後に行った数値実験の結果を表 1、表 2、表 3 に示す。表 1 および表 2 は、Caprara らの解法の基本性能の評価を目的に行った実験の結果であり、表 3 は表 2 で対象とした 486 乗務の問題について行路案数の変化に対する本アプローチの挙動の変化を示したものである。各表内の UB は上界値を、LB は下界値を、GAP は $(UB - LB) \div LB \times 100$ で計算される双対ギャップを意味する。また、計算時間の単位は全て秒、実験は Pentium 4 2.8GHz、メモリ 1.5GB のパーソナルコンピュータで行った。なお、表 2 のみ全ての c_j を 1 として計算を行った結果である。

これらの実験結果から、いくつかの問題点が浮き彫りになった。

■ 解の精度 まず注目すべきなのは、我々のインスタンスに対する解の精度が、期待した水準には至らず、また Caprara らが対象にした行数や列数が同等のインスタンスに対する解の精度に比べ

て劣っている点である (表1、表2)。

これは、インスタンスの性質との関連が推測され、定式化レベルで同じ問題であってもインスタンスの特徴に応じた工夫が場合によっては必要なことを示唆していると考えられる。このことは、近似解法を用いる上では当然覚悟すべき問題であるが、解法内のパラメータの値の調整のみならず解法の一部修正が必要になることも場合によってはあろう。

■ 膨大な数の行路案 前述の通り、ルールが許容する行路案は膨大な数にのぼり、行路が満たすべきルールに何も制限をしないと、行路案は百万以上生成される。行路案が多ければ多いほど全ての行路案を含む「本来の問題」に近づくが、いかに近似解法といえども、この規模の問題に対しては計算時間や解の精度の問題が生じる (表3)。

そのため、実際には行路案数ある程度抑制する必要がある、行路案としてそれが許されるにせよなんらかの基準で排除するようなメカニズムを加えて行路案を生成する必要があった。しかし、本来の整数最適解に採用されるべき行路案を除外することのないような方法は一般に明らかではなく、この部分は事前列挙タイプの弱点である。なお、実務においても、ルール上許されるとしてもなんらかの取捨選択が行われているのが実際のところであろう。

■ 目的関数以外の評価項目 より深刻なのは、目的関数以外にも実際には存在する、解の評価対象になる項目の値が良くないことである。単目的の最適化を行っている以上、目的関数以外が最適化の対象にならないのは当然であるが、「各乗務に対する便乗数」やそれらの総和の「総便乗数」といった実務的に重視される尺度の値が、許容値からかけ離れているという現象が頻繁に発生した。

■ 非現実的な行路案 さらに得られた実行可能解を精査すると、その行路案が非現実的であることも判明した。すなわち、行路のルール上は許容されるとしても、実務上は採用されることのないような行路が生成されそれが選択されていた。ただし、これは解法側の問題というよりも、むしろ行路案生成における問題であり、行路案数が膨大な数にのぼることと併せてその対策を講じることを考えた。

表1 Caprara らの解法の性能:イタリアの鉄道のデータに対する結果

行	列	UB	LB	GAP(%)	計算時間
507	63,008	175	168	4.2	694
516	47,311	182	182	0.0	218
582	55,151	215	204	5.4	663
2,536	1,081,841	729	585	25.6	46,537
2,586	920,683	986	869	13.5	32,918
4,284	1,092,610	1,114	933	19.4	45,439
4,872	968,672	1,579	1,477	6.9	55,531

表2 Caprara らの解法の性能:日本の鉄道のデータに対する結果

行	列	UB	LB	GAP(%)	計算時間
486	56, 131	93	76	22.4	670
486	121, 119	87	70	24.3	1,332
486	246, 691	86	69	24.6	4,291
486	490, 552	83	70	18.6	6,416

表3 行路案数の増加と計算結果、計算時間の関係

行路数	生成時間	UB	LB	GAP(%)	計算時間
54,202	597	136	112.89	20.5	472
58,912	602	134	111.98	19.7	428
50,943	594	133	113.59	17.1	410
115,931	600	130	106.41	22.2	658
107,172	611	132	105.66	24.9	948
121,519	610	133	90.65	46.7	898
259,294	866	130	87.30	48.8	2,201
265,886	871	128	112.99	13.3	2,294
280,540	894	127	100.85	25.9	2,424
441,135	995	127	87.56	45.0	5,562
472,345	1,033	126	102.56	22.8	4,545
453,891	1,022	126	102.06	23.4	5,201
1,118,310	1,664	132	86.94	51.8	12,337
1,149,612	1,593	126	87.87	43.4	13,401
1,108,646	1,578	124	99.72	24.3	14,964

5.1.2 初期の試みの結果への対策

これらの結果ならびに考察に基づくアプローチの修正として、以下のふたつを試みた。

■ 便乗の抑制に対する試み：一般上限制約の付加 便乗は全く許されないわけではないが、過剰な便乗の発生も避けたい。また、各乗務員基地に所属する乗務員数の上限なども考慮したいという要望もある。そこで、集合被覆問題に一般上限 (*Generalized Upper Bound*, GUB) 制約を付加することが考えられる。

商用の数理計画パッケージであれば付加される制約の種類は問題にならないが、Capraraら[1]の方法はそのままでは使えなくなる。そこで、我々のグループは一般上限制約付き集合被覆問題に対する近似解法をCapraraらの方法を元に考案した[6]。

表4に示したのはその結果の一部であり、5.2.1節で触れる列生成法によって生成された行路を対象にしてPentium 4 2.4GHzのパーソナルコンピュータを用いて解いた結果である。

そして、これを含む各種数値実験の結果から、各乗務に対する便乗数を減らせることが判明したが、便乗の総数を抑制する部分に関しては課題が残った。また、計算時間が極めて長いことも改善すべき点のひとつとして挙げられる。

表4 GUBの有無の比較 (列生成法による行路案)

	GUB 無し	GUB 付き
上界値	135	133
下界値	123	123
GAP (%)	9.8	8.1
総便乗数	108	98
1乗務の最大便乗数	3	3

乗務数:488、行路案数:937
GUB 付き集合被覆問題の計算時間:17.5時間

この結果から考えられる代替的な方向性としては、便乗を考慮した行路案による集合分割問題での定式化や、便乗に対するペナルティの考慮などが挙げられよう。

■ 車両運用の考慮 我々の当初のやり方[5]では、全ての列車を乗務の単位にする際に乗り換え可能駅で一律に分割した。しかし、

- 乗務員は、可能な限り同一列車に続けて乗務させたい
- 終着駅に到着した列車の使用する車両が別の折り返し列車に充当される際、乗務員も同様の運用になっていることで、列車の遅延が生じた場合の対応が簡単になる

などの点から、「できる限り車両運用と乗務員運用を近づけたい」という実務的な要望がある。

そこで、行路案の列挙に車両運用を考慮したところ、行路案を減らしつつ結果的に文献[5]に対して大幅に便乗数が減るという結果を得て、一定の効果を確認できた。しかしそれでもなお便乗数が多く、決め手に欠くとともにその他の方策が必要であった。

5.1.3 Wedelin[7]の近似解法

集合被覆問題に対する近似解法はCapraraら[1]以外にも存在し、様々な近似解法が我々が対象とするインスタンスに対してどのような挙動や性能を示すのかを知っておくことは意義がある。

そのような近似解法のひとつとして、Wedelin[7]はラグランジュ緩和をベースにしたヒューリスティック解法を提案しているが、上で述べたような背景から、この解法が我々が対象としているインスタンスに対してうまく働くかを検証した[8]。

Wedelinの解法の特徴は、ラグランジュ緩和を下界値の算出のための手段とは考えずに、上界値すなわち実行可能解の算出の手段と考えるところにあり、ラグランジュ双対問題の最適化を直接的に目指す訳ではない。その他、実行可能解の最適性を判定できない点や下界値を提供しない点、さらにCapraraの方法と同様に解法内のパラメータの値の設定が必要な点が弱点として挙げられる。

表 5 Wedelin の解法と分枝限定法の性能比較

乗務数	Wedelin の解法		分枝限定法	
	最良値	時間 (秒)	目的関数値	時間 (秒)
488	122	203	122	32,594
501	130	505	130	9,562
503	134	460	134	5,660
526	141	281	141	8,433

表 5 は、そのパラメータの値を適切に設定した上で事前に列挙した10万程度の列を持つ問題に対する分枝限定法(数値計画パッケージの Xpress-MP を使用)と Wedelin の解法の結果を比較している(使用したのは Pentium 4 3.2GHz、メモリ 2 GB のパーソナルコンピュータ)。これらデータ全てにおいて、分枝限定法の結果は全て最適解だが、Wedelin の解法は分枝限定法に対して圧倒的に短い時間で同じ値を算出していることに注目したい。

もちろん、本アプローチが事前列挙タイプを前提にしている以上、手元にある列に対して良い解を得たという位置づけに過ぎず、解の精度が行路案群をどのように選ぶかに依存する問題点は残る。その意味で、この解法は、十分良い解を与えると信じられる行路案群が手元にあり、かつ、全体のスケジュールを繰り返して求めることを必要とするようなケースにおいて威力を発揮するものと考えられる。

5.2 列生成タイプの方法

最適な整数解を求めるには分枝価格法が必要であるが、その実装が容易ではないことを考慮し、また理論的な基盤を固める意味で、列生成タイプの近似解法を指向した。

5.2.1 初期の試み

このタイプの方法に対する我々のグループの最初の試み[9]では、列生成法の性質の理解と基本的な挙動の把握を目指し、集合被覆タイプの問題に再定式化をした上で、線形計画主問題が最適に解けるまでに生成された列を対象に整数解を求める方法を採用した。ただし、その整数解を得るために用いた方法が近似解法であったため、生成された列の整数解を出す意味での善し悪しの評価がはっきりしなかった。

そこでその後、整数実行可能解を得る部分を近似解法ではなく分枝限定法(数値計画パッケージ)で置き換え[10]、列生成法の挙動や特徴をある程度把握できたが、解法全体としては列生成子問題の最適化の手順の効率や整数解の精度に課題が残っていた。

5.2.2 列生成ヒューリスティック解法の性能評価

このような経緯を経て、新たに列生成子問題の解法を全面的に見直した上で、列生成ヒューリスティック解法の性能評価を行った[11]。この解法は、多資源制約付最短路問題に帰着される列生成子問題を pull 型ラベリング解法で解き、前述の方法と同様に、主問題の下界値が得られた時点までに生成された列を集合被覆問題の列とし、その問題を対象に数理計画パッケージの Xpress-MP を用いて整数解を得ている。なお、主問題の LP の最適化には数理計画パッケージの CPLEX を使用し、Pentium 4 3.2GHz、メモリ 2 GB のパーソナルコンピュータ上で実験を行っている。

表 6 では、乗務数が488の問題に対して、列生成の際の子問題毎に同時に追加する列の本数上限の違いによる、LP 解が得られるまでの時間や整数解の精度の違いを比較している。すなわち、単体法では被約費用が負で最小の列を一本追加しておけば目的関数の値を改善できるが、本アプローチでは被約費用が負の列が複数生成されることが多く、複数の列を限定線形計画主問題に追加することが下界値や整数実行可能解の計算にどのような影響を与えるかが興味の対象のひとつとなる。

表 6 の結果は、列生成の各反復における子問題毎の追加列数の上限の設定によって IP 解の値に若干のばらつきはあるが、おおむね良好な結果である。特に、追加する列の本数を 1 本に限定せず複数にすることによって、列生成の時間すなわち LP 最適解が得られるまでの時間が大幅に削減されている。また、5.1.3節の表 5 で乗務数488の結果が最適解でありながら、表 6 の IP 最適値はそれに勝っている。すなわち、表 5 の結果はあくまでも列挙された列からなる問題に対して最適だったに過ぎなかったことに注目されたい。その意味で、列生成法では線形緩和問題の最適化のために列を生成するものの、それら列に基づいて得られた整数解の精度を観察する限り、事前列挙タイプの代替的方法に十分なり得る可能性を示しているともいえる。

もちろん Wedelin の解法に比べると主問題の連続緩和問題の最適化ならびに IP 最適化に要する時間の合計ははるかに長いが、列生成法によるアプローチの意義はその下界値によって上界値の評価を行える点にあるといえよう。

表 6 列生成法による行路案群の生成と整数解の精度:追加列数上限の影響

列数上限	1	5	10	なし
反復回数	509	172	125	40
総列数	2622	4386	5756	19287
LP 最適値	114.317			
列生成計算時間(秒)	5398.3	1942.2	1234.0	354.2
IP 解の値	119	118	116	116

※計算時間の単位は秒、IP 解の値は10,000秒で計算を打ち切った結果である

5.2.3 便乗削減のための定式化の修正と工夫

Wedelin の解法と列生成ヒューリスティック解法は、別々な意味で有望な結果を示したが、便乗については目的関数になんら反映させていないことから当然であるが、実務的には改善が望ましい値であった。

より細かく考えると、便乗が生じたことで便乗をどの乗務員に任せるかの決定を行う必要があるとともに、その結果、本来業務を行う意味での乗務として採用された行路中の乗務が便乗になったことで、当該行路の実行可能性が破壊される可能性、すなわち厳密には労働条件を満足しなくなるという問題が発生し得る。

これに対して我々は、集合被覆問題を解いた答を出発点に便乗をどの乗務員に行わせるかをなるべく労働条件を満たすように決定する手順を試みるとともに、集合分割問題の定式化で便乗を考慮した事前列挙タイプの解法も試みたが、実用的な意味で結果は不十分であった。

このような経緯を経て、Graves ら [12] が示した *Elastic Embedded Set Partitioning Problem* (EESPP) として定式化して列生成ヒューリスティック解法を試みるとともに、Barnhart ら [13] の提案する便乗選択方法のアイデアの適用も行った [14]。

■ EESPP による定式化の効果 EESPP の定式化は以下の通りである。

$$(\text{EESPP}) \min z = \sum_{j \in N} c_j x_j + \sum_{i \in M} d_i s_i \quad (7)$$

$$\text{s.t. } \sum_{j \in N} a_{ij} x_j - s_i = 1, \quad \forall i \in M \quad (8)$$

$$x_j \in \{0, 1\}, \quad \forall j \in N \quad (9)$$

$$s_i \geq 0, \quad \forall i \in M \quad (10)$$

ただし、 s_i は第 i 乗務の便乗数、 d_i は第 i 乗務の 1 便乗当りのペナルティである。

数値実験は Pentium 4 3.2GHz、メモリ 2 GB のパーソナルコンピュータ上で行い、主問題の線形緩和問題の最適化には CPLEX を、IP 最適化には Xpress-MP を使用した。また、IP 最適化は 10,000 秒で計算を打ち切っている。

表 7 では、総勤務日数と総便乗数について、集合被覆問題での定式化 (SCP) と EESPP で便乗ペナルティを変化させたものとを比較しているが、これから分かるように総勤務日数と総便乗数にはトレードオフの関係がある。すなわち、総便乗数を減らそうとすると総勤務日数は増加する。このペナルティを大きく設定することで、集合被覆問題の解に比べて総勤務日数は悪化するものの、総便乗数はかなりの減少が見込める。

表7 EESPP と SCP の比較

乗務数	尺度	SCP	便乗ペナルティ		
			0.1	0.5	1.0
488	総勤務日数	116	116	117	121
	総便乗数	16	12	8	2
501	総勤務日数	118	118	119	123
	総便乗数	17	11	7	1
503	総勤務日数	122	122	125	125
	総便乗数	19	15	3	1
526	総勤務日数	126	127	129	130
	総便乗数	18	14	2	0

表8 便乗選択の効果

乗務数	尺度	便乗の対象	
		選択をする	全て許可
488	列生成計算時間(秒)	2,434	3,006
	総勤務日数	121	119
	総便乗数	2	6
501	列生成計算時間(秒)	595	2,776
	総勤務日数	124	120
	総便乗数	1	7

■ 便乗選択の効果 便乗選択の基本的な考え方は、集合分割問題で定式化した上で「効果的な乗務」にのみ便乗を許すというものである。具体的には、列生成のプロセスにおいて、乗務 $i(\in M)$ を便乗したと仮定した上での行路の最小被約費用を近似的に求め、値が小さいならば便乗する価値があると判断するものである。このことによって、全ての乗務に便乗を許可することを防ぎ、列生成子問題の最適化の負担が大きくなるようにする。

便乗選択の効果を検証するため、比較対象として、便乗を全ての乗務に許可した場合を考える。表8に示したのは、計算機的环境や条件を「EESPPによる定式化の効果」の実験の場合と同一に行った実験の結果だが、便乗選択を行うことで列生成子問題への負担は軽くなり、便乗が減っている。しかし、表7のEESPPと比較すると、総勤務日数、総便乗数に大差はなかった。

6 考察

事前列挙型としては、Capraraらの解法は若干の課題を残したが、Wedelinの解法は良好な結果を示した。特に、このような問題を繰り返し解く場合には、この種の高速な解法は有効であろう。

また列生成ヒューリスティック解法は、事前列挙型に対して計算時間において劣っているものの、解の精度はいずれも実用的な域に達していると考えられる。もちろん、実務的に不可欠な労働に関する条件は現状において可能な限り制約に盛り込んであるものの、さらに細かい要因を組み込む要

請に対処できるようにすることを考えると、列生成ヒューリスティック解法には列生成子問題の最適化をより短時間でやるなどの課題は存在する。

このほか、以下のような課題が挙げられる。

■ 行路のコストの扱い 我々が対象とした問題では、行路の内容にかかわらず日勤か夜勤かによってコストが定まってしまう。すなわち、目的関数の行路案の採否に関する変数の係数は1もしくは2のいずれかであり、これが求解においてなんらかの影響を及ぼしている可能性がある。たとえば Wedelin の解法では、これを踏まえた上での修正を施している。

元々の問題に行路の内容を評価してコストとするようなメカニズムが存在しない以上、これをそのまま扱うのがある意味で正しいが、その一方で、行路の内容を反映するような簡単な評価関数、例えば乗務の間の時間などを評価対象にするような可能性はあろう。もちろん、乗務の間の時間が短ければ短いほど良いわけではなく、例えば食事や仮眠などの時間などをどう考慮するかという問題もある。いずれにせよ、行路に対する妥当な評価関数を設定することが必要ではないかと考えられる。

■ 分枝価格法 最適性を保証する解法という意味では、分枝価格法を指向する方向性が考えられる。分枝価格法の適用においてその実装上もっとも重要なのは、分枝の方法と分枝内容に応じた列生成子問題におけるネットワークや主問題の修正である。これらの事柄に関してはすでに先行する研究で一定の成果があり、Barnhart ら [3] が述べているように、特に分枝の方法としては Ryan and Foster [15] の制約分枝の延長上の branch-on-follow-on [16] が知られている。

7 おわりに

本稿では、鉄道における乗務員スケジューリング問題に対する我々のグループのここ数年の試みを報告した。そこでは、

- 既存の方法の中で、我々が対象とする問題に対して極めて高速化かつ良好な精度の解を与える近似解法の存在を確認した
- 事前列挙タイプの解法は近似解法の選択が重要であるとともに、インスタンスに応じた解法内のパラメータの値の調整、あるいは解法の部分的な修正の必要があるものの、それらを適切に行うことで良好な精度の解を短時間で提供し得ることが判明した
- 列生成タイプのヒューリスティック解法は計算に時間がかかるものの、実用的な精度の解を提供し得る
- 便乗に関する問題は、定式化の工夫や便乗の選択に関する考慮を適切に行うことで解決できる可能性を示唆した

などの成果を得た。

これまでのこれら試みによって、乗務員スケジューリングの解法のための骨格部分に関する方向性が与えられたと考える。特に列生成法に関するさまざまな基礎的な知見が得られたことは、極めて貴重だったといえる。

一方、前述の通り、これらの研究が対象にした問題では現実に存在する労働に関する条件のかなりの部分を盛り込んでいるが、簡略化した部分がないわけではない。そこで今後は、それらのある意味で極めて実務的な条件をどのように反映するかと、実務的な意思決定にこれらの成果をどのように利用していくかを検討することが課題として挙げられよう。

これら知見の延長上により実務的な要望を盛り込んだ高速かつ高精度の解法を提供できることを今後の目標として、本稿を終える。

参考文献

- [1] A. Caprara, M. Fischetti, and P. Toth. A heuristic method for the set covering problem. *Operations Research*, Vol. 47, No. 5, pp. 730–743, 1999.
- [2] 田島玲. パイロットの乗務パターン最適化. 杉原厚吉, 茨木俊秀, 浅野孝夫, 山下雅史 (編), アルゴリズム工学—計算困難問題への挑戦—, pp. 266–267. 共立出版, 2001.
- [3] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, Vol. 46, No. 3, pp. 316–329, 1998.
- [4] 今泉淳. 運輸業における数理計画の応用—乗務員スケジューリングのモデル化と解法. 東洋大学経営論集, 68号, pp. 35–48, 2006.
- [5] 今泉淳, 鷺見亮, 斎藤秀和, 森戸晋, 富井規雄, 福村直登. 鉄道乗務員運用計画へのバックトラック法による行路候補列挙と集合被覆問題の近似解法. 統計数理研究所共同研究リポート168 最適化:モデリングとアルゴリズム17, pp. 172–179. 統計数理研究所, 2004.
- [6] 斎藤秀和, 今泉淳, 森戸晋, 福村直登. 一般上限制約付き大規模集合被覆問題—鉄道の乗務員運用計画に対するラグランジュ緩和アプローチ—. 日本オペレーションズ・リサーチ学会 2004年春季研究発表会 アブストラクト集, pp. 138–139, 2004.
- [7] D. Wedelin. An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Annals Operations Research*, Vol. 57, pp. 283–301, 1995.
- [8] 三浦礼, 福村直登, 森戸晋, 今泉淳. 乗務員運用計画の集合被覆問題に対する Wedelin 解法の適用. 日本オペレーションズ・リサーチ学会 2005年春季研究発表会 アブストラクト集, pp. 96–97, 2005.
- [9] 小川健一, 今泉淳, 森戸晋, 福村直登. 鉄道の多拠点乗務員運用問題に対する列生成アプローチ. 日本オペレーションズ・リサーチ学会 2004年春季研究発表会 アブストラクト集, pp. 136–137, 2004.
- [10] J. Imaizumi, H. Arai, K. Ogawa, S. Morito, and N. Fukumura. Column generation approaches to scheduling problems in logistics. *International Journal of Logistics and SCM Systems*, Vol. 1, No. 1, pp. 7–13, 2006.
- [11] 植田達広, 森戸晋, 今泉淳, 福村直登. 乗務員運用計画問題の列生成子問題に対する Pull 型ラベリング

- 解法と性能評価. 日本オペレーションズ・リサーチ学会 2005年秋季研究発表会 アブストラクト集, pp. 160–161, 2005.
- [12] G.W. Graves, R.D. McBride, I. Gershkoff, D. Anderson, and D. Mahidhara. Flight crew scheduling. *Management Science*, Vol. 39, No. 6, pp. 736–745, 1993.
- [13] C. Barnhart, L. Hatay, and E.L. Johnson. Deadhead selection for the long-haul crew pairing problem. *Operations Research*, Vol. 43, No. 3, pp. 491–499, 1995.
- [14] 加藤 怜, 植田達広, 森戸 晋, 福村直登. 乗務員運用計画問題に対する便乗削減の定式化と解法. 日本オペレーションズ・リサーチ学会 2006年春季研究発表会 アブストラクト集, pp. 178–179, 2006.
- [15] D. Ryan and B.A. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*, pp. 269–280. North-Holland, 1981.
- [16] C. Barnhart, E.L. Johnson, G.L. Nemhauser, and P.H. Vance. Crew scheduling. In R.W. Hall, editor, *Handbook of Transportation Science*, pp. 493–521. Kluwer Academic, 1999.
- [17] 今泉 淳, 福村直登, 森戸 晋. 鉄道クルースケジューリングに対する数理計画アプローチモデル化と最適化の観点から. スケジューリング・シンポジウム2004 講演論文集, pp. 77–82, 2004.
- [18] 今泉 淳, 福村直登, 森戸 晋. 鉄道における乗務員スケジューリング問題に対する数理計画法の適用. スケジューリング・シンポジウム2006 講演論文集, pp. 99–104, 2006.

※福村直登 ((財) 鉄道総合技術研究所 運転システム研究室長)、森戸 晋 (早稲田大学 創造理工学部 教授)

(2008年 1 月 8 日受理)