

鉄道乗務員スケジューリング問題に対する列生成法によるアプローチ

A Column Generation Approach to the Railway Crew Scheduling Problem

今 泉 淳

(Jun Imaizumi)

鉄道乗務員スケジューリング問題に対する 列生成法によるアプローチ

今 泉 淳

- 1 目的
- 2 問題と背景
 - 2. 1 問題
 - 2. 2 本研究の背景
 - 2. 3 本研究のアプローチと位置づけ
- 3 定式化
 - 3. 1 集合被覆問題としての定式化
 - 3. 2 一般化集合分割問題としての定式化
- 4 解法
 - 4. 1 列生成法
 - 4. 2 列生成子問題
 - 4. 3 列生成子問題の解法における工夫
- 5 数値実験
 - 5. 1 数値実験の概要と目的
 - 5. 2 実験結果
 - 5. 2. 1 集合被覆問題としての定式化
 - 5. 2. 2 一般化集合分割問題としての定式化
 - 5. 2. 3 数値実験全体を通じて
- 6 結論
- 付録
- 参考文献

1 目的

鉄道業などをはじめとする運輸業においては、旅客や貨物の輸送を行う上でそれらの特性を勘案したダイヤグラムを設定している。これら日々の運行を実現する上では、企業が保有するさまざまな資源を効率良く運用することが求められる。具体的には、鉄道業においては車両と乗務員のそれぞれをダイヤグラムに示された列車に充当することが必要になる。これらはいずれも有限資源であり、またその運用計画の立案には、相当の手間と経験や知識を必要とする。

本研究では、鉄道における乗務員スケジューリング問題に着目し、具体的な計画立案の支援を行うための数理的方法を提案する。具体的には、列生成法に基づくヒューリスティック解法を構築し、その性能を計算機による数値実験により評価する。

2 問題と背景

2. 1 問題

乗務員スケジューリング問題とは、ダイヤグラムが与えられた前提下で、ダイヤグラムに示された各列車に対して、さまざまな制約条件を満足するように乗務員を割り当てる問題、ととらえることができる。

ダイヤグラムは、列車の始発駅と出発時間、途中停車駅とその到着・出発時間、終着駅とそこへの到着時間などを与える。これらの駅のうち乗務員は特定の駅（乗り換え可能駅）で列車を乗り降りし、別の列車に乗り換えてよい。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。なお、乗務員が運転などの業務をせず移動のためだけに列車に乗ることを便乗と呼ぶ。

一方、それぞれの乗務員は、その所属する乗務員基地から勤務を開始し、労働上のさまざまな条件を満たしながら列車を乗り継ぎ、最終的に勤務の開始地である自分の乗務員基地に戻らなければいけない。なお、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、勤務の開始地からそこ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）を行路と呼ぶ。

上で触れた「さまざまな条件」の具体的な内容に関して簡単に述べると、勤務の開始から終了までの期間である勤務時間は、

- 実際に列車に乗っている乗務時間
- それ以外に必要な各種時間

からなる労働時間と休憩時間からなるが、これら勤務時間や労働時間、乗務時間には上下限が設定されており、これら以外の属性にも各種制約が存在する。これらは、勤務時間が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を乗務に分解し、すべての乗務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行路の組合せを見つけることが目的となる。

2. 2 本研究の背景

乗務員スケジューリングに対して、諸外国では古くから航空を中心に数理計画によるアプローチ

が試みられ、研究例も比較的多い。それに対して、鉄道に対する数理計画の適用が行われるようになったのは航空に比べて比較的最近であり[1]、とりわけ国内では具体的な研究例は決して多いとはいえない。

モデル化としては、後述する集合被覆問題などによる定式化における列に行路を対応させたものが多く見られ、この定式化を前提にした場合の解法は「事前に行路を列挙するアプローチ」と「列生成法に基づくアプローチ」のふたつに大別できる。

前者のアプローチで問題の全ての行路を陽に対象にするような整数計画問題を考えることは、可能な行路数が膨大になることから事実上不可能である。また、事前列挙する列をある程度限定してそれからなる問題に対して良い解を得たとしても、それが本来の問題から見た場合どの程度良い解なのか不明である点が弱点として残る。すなわち、前者のアプローチではどのような行路群を対象にして問題を解くかが鍵となるが、アプローチとしては単純であり実装の容易さの面からは有望である。

一方、後者の列生成法に基づくアプローチは、列（行路）を逐次生成しながら問題の下界値を得る、さらには、整数実行可能解を得る方法である。下界値を得るプロセスでは線形緩和問題の列を必要に応じて生成するが、下界値が算出された時点までに得られた列を対象にして最適な整数解を得ても、元の問題から見た場合の最適解が得られている保証は無い。もちろん、分枝限定法に列生成法を組み込んだ分枝価格法[2]を用いることで元の問題の最適解を得ることができるが、分枝価格法はその実装が決して容易ではなく、手軽に採用できる方法ではない。

2. 3 本研究のアプローチと位置づけ

本研究では上記のような背景を踏まえ、元の問題の線形緩和問題を最適に解くために列生成法を用い、最適に解けたらばそれまでに生成された列からなる問題を対象に整数実行可能解を得る列生成法によるヒューリスティックなアプローチを採用する。この方法は、元の問題の最適解を提供できる保証はないものの、実装面の手間は分枝価格法ほどではない。一方、生成された列は線形計画問題を解くために生成したものであるものの、整数計画問題の列としてもある程度良質な列を含んでいると期待される。

そして本研究は、Imaizumi et al.[5]の複数の乗務員基地を持つ乗務員スケジューリング問題に対する列生成アプローチと同種の方法を、その問題の一部詳細に関して改めたものに対して適用するとともに、植田ら[10]の列生成子問題の解法の考え方を利用した上で、大規模な数値実験を行うという位置づけになる。さらに、最適化の結果として生じる便乗の対策として、定式化の一部を変更しこれに対しても数値実験を行い、これらを併せて列生成アプローチの性能評価を試みている。

3 定式化

本研究では、過去の研究でも比較的多く採用されている集合被覆問題としての定式化とともに、便乗の抑制を目指した定式化を行う。

3. 1 集合被覆問題としての定式化

乗務の集合を M 、行路の集合を N 、 a_{ij} を行路 j が乗務 i を含むとき 1、さもなければ 0 とし、 $a_j = (a_{1j}, \dots, a_{ij}, \dots, a_{|M|j})^t$ を行路 j に対応する列ベクトル、 c_j を行路 j のコストとすれば、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 (Set Covering Problem, SCP) として定式化できる。

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

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

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

ここで x_j は、行路 j を選択するとき 1、さもなければ 0 となる 0-1 変数である。

式 (2) は不等式であるため、各乗務は複数の列 (行路) で被覆され得る。複数の行路で被覆された場合は、一つの行路 (一人の乗務員) 以外は、その乗務に対しては便乗となる。

集合被覆問題による定式化は過去にも多く採用されており、また鉄道においては便乗に対する扱いが航空に比べて比較的緩やかであるため、それほど不自然ではない。

3. 2 一般化集合分割問題としての定式化

集合被覆問題による定式化は、集合分割問題に比べて実行可能解を得やすいというメリットがあるが、ある乗務を複数の行路で被覆することを許すため便乗が発生する可能性がある。もちろん、現実において便乗が全く許されない訳ではないが、全体としての便乗数や便乗時間が小さいことが望ましい。

ところで Mitra and Darby-Dowman [7] は、乗務員スケジューリングに対して、超過被覆と被覆されないことの両者を許す一般化集合分割問題 (Generalized Set Partitioning Problem, GSPP) としての定式化を示している。

本研究は、集合被覆問題としての定式化とともに、一般化集合分割問題の一バリエーションとし

て、超過被覆のみを許しそれに対するペナルティを目的関数で考慮する問題を考え、これによって便乗の抑制を狙う。具体的には、以下の定式化を採用する。

$$\min z = \sum_{j \in N} c_j x_j + \sum_{i \in M} d_i s_i \quad (4)$$

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

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

$$s_i \geq 0, \quad \forall i \in M \quad (7)$$

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

4 解法

既に述べたように、本研究では、i) 元の問題の線形緩和問題を列生成法で解く、ii) その線形緩和問題が最適に解けたらそれまでに生成された列を対象に整数実行可能解を得る、という手順を採用する。

4. 1 列生成法

列生成法で下界値を求めるまでのやり方は Lavoie et al.[6]と同様のものであり、ここで集合被覆問題を例にその概要を述べる。

まず、 N の部分集合 N' に属する要素に対応する列からなる以下のような限定的な線形計画問題を考える。

$$\min z = \sum_{j \in N'} c_j x_j \quad (8)$$

$$\text{s.t. } \sum_{j \in N'} a_{ij} x_j \geq 1, \quad \forall i \in M \quad (9)$$

$$x_j \geq 0, \quad \forall j \in N' \quad (10)$$

この限定的な問題を最適に解くと、その最適解に対応する双対価格 $\pi_i (i \in M)$ が得られる。この限定的な問題の最適解が、全ての列を持つ元の問題の連続緩和問題の最適解になっているかどうかの判定は、目的関数値を改善するような列、すなわち現在の双対価格の下で負の被約費用を持つ列が存在するかどうかを判定（価格付け）すれば良い。もしそのような列が見つかったならばその

列に対応する行路を N' に加えて、新たな限定的な線形計画問題を解くことを、そのような列が見つからなくなるまで繰り返す。

4. 2 列生成子問題

限定的な線形計画問題の最適解が全ての列を持つ問題の線形緩和問題の最適解になっているかどうかの判定は、以下のような列生成子問題（価格付け問題）を解いて最適な行路 j^* を見つけることに帰着する。

$$\min_{j \in N} \left\{ c_j - \sum_{i \in M} \pi_i a_{ij} \right\} \quad (11)$$

この問題は、始点を基地からの出発、終点が基地への帰着、ノードが乗務に対応し、乗り継ぎ可能な乗務間をアークで結んだ有向グラフ（ネットワーク）上で、行路に関する制約を守りつつ上記 (11) によって最短なパスを見つける問題、すなわち最短路問題に帰着する。Lavoie et al. [6] の問題ではこの列生成子問題が制約無しの最短路問題になるが、本研究の問題ではこれが資源制約付きの最短路問題 [3] となる。本研究では、この資源制約付きの最短路問題を植田ら [10] の Pull 型ラベリング解法を用いて解いている。

この問題を解いて見つかった最適パスに相当する列が負の被約費用を持つ場合、その列を限定的な線形計画問題に加えて再度解くことを繰り返すが、その際に最適パス以外に負の被約費用を持つ列に相当するパスが見つかる場合がある。線形計画問題の最適化以外に最終的に整数実行可能解を得る必要があることを踏まえ、加える本数を変化させた場合の線形計画問題の最適化への影響、さらには整数実行可能解の精度への影響を観察するために、後の数値実験では追加する列の本数の上限を変えて実験を行う。

4. 3 列生成子問題の解法における工夫

列生成子問題はネットワークのノード数やアーク数が多くなると計算時間がかかる。一方、限定的な線形計画問題を解くプロセスで、列生成子問題を最適に解かなくても、負の被約費用を持つ列が見つければ単体法で目的関数値の改善はできる。負の被約費用を持つ列が存在しないと判定された時点で線形計画問題の解の最適性が保証されるが、その判定のために列生成子問題が最適に解かれている必要はあるものの、列生成の毎反復においては目的関数を改善する意味では必ずしも列生成子問題で最適解を得なくてもよい。

これに関して Barnhart et al. [2] は、まず近似的に価格付けを行い、負の被約費用の列が生成され

なくなったらば列生成子問題を最適に解くという2フェーズのアプローチを提唱している。そこで本研究も、このようなアプローチを採用する。

具体的には、Freling et al.[4]が述べている *dynamic network size* 戦略と、望ましくない組合せの乗務間のアークを取り除く戦略のふたつを併せた上で簡略化し、時間間隔の大きい乗務間のアークを除いたネットワークをまず考え、時間間隔の基準を変えてながら次第に時間間隔の大きい乗務間のアークを加えて、ネットワークの規模を本来のそれに近づけてゆく。数値実験では、このような工夫を施さなかった場合（「列生成子問題の解き方1」と称し「方法1」と略記する）と、ここで述べた方法（「列生成子問題の解き方2」と称し「方法2」と略記する）の比較も行う。

なお、Barnhart et al.[2]は分枝価格法を念頭に置いた上で、価格付けの問題の最適化に手間がかかる場合には、近似アルゴリズムを採用した上で複数の列を追加することを良いと述べている。

5 数値実験

5.1 数値実験の概要と目的

実験のためのプログラムは Microsoft Visual C++ .NET 2003 を用いて作成し、数値実験は HP xw4300 Workstation（Pentium 4 3.8GHz、メモリ2GB）上で行った。また、線形計画問題および整数計画問題は、ILOG 社の CPLEX 9.0 を用いて解いている。なお、整数計画問題は分枝限定法で解いているが、その計算は86400秒で打ち切っている。

具体的な問題例としては、現実に基づく、乗務数が概ね500前後の4つの異なるデータ（乗務数488、501、503、および526）を用いている。この問題における列のコストは日勤が1、夜勤が2であり、集合被覆問題における目的関数は総勤務日数となる。また、乗り換え可能駅は4つ存在し、そのうちの3ヶ所に乗務員基地がある。したがって、列生成法の各反復において日勤と夜勤それぞれ3、合計6個の列生成子問題を解くことになり、それぞれの子問題毎に与えられた上限数まで被約費用の小さい順に列を選んで限定的な線形計画問題に追加している。

数値実験では、集合被覆問題としての定式化の結果を基点に、総勤務日数、便乗数、便乗時間という評価項目に着目し、

- 列の追加本数の上限が線形計画問題の最適化と整数実行可能解の精度に及ぼす影響の観察
- 一般化集合分割問題での定式化による便乗の抑制の可否および集合被覆問題の場合との比較
- 列生成子問題の最適化の方法を変えた効果に関する観察

を行う。

これらを通じて、本研究が採用した列生成ヒューリスティック解法の全体的なパフォーマンスを評価する。

5. 2 実験結果

実験結果を、表1から表16までにまとめた。それぞれの表内の項目は以下の通りである。

列追加上限：列生成の一反復で、子問題毎に追加する列数の上限。

列本数：線形計画問題が最適に解けて下界値が得られるまでに生成された列数。

反復回数：列生成の反復回数。

LP最適値：線形計画問題の最適値。列の追加本数上限に依存せず一致する。

時間：列生成が終了するまでに要した時間で、単位は秒。

IP：分枝限定法の計算を打ち切った時点での整数計画問題の解の目的関数値。ただし、集合被覆問題の場合は総勤務日数と同一なので省略した。

総勤務日数：整数実行可能解の総勤務日数。集合被覆問題による定式化においては目的関数値である。

便乗時間：発生した便乗によって生じた便乗時間の総和で、単位は分。

便乗数：発生した便乗の総数。

また、これら各表の中の総勤務日数、便乗、便乗時間の問題・定式化・方法別の平均を表17に示した。

5. 2. 1 集合被覆問題としての定式化

表1から表4に方法1の結果を、表5から表8に方法2の結果を示した。

a) 列の追加本数上限の変化の影響

まず、表1から表4に注目する。列の追加本数の上限が1の場合、列生成法の各反復では各乗務員基地の日勤および夜勤それぞれの被約費用のもっとも小さい列のみしか追加されない。その結果、線形計画問題の最適化に多くの反復を必要とし、線形計画問題の最適化にかかる時間は長くなっている。逆に、負の被約費用を持つ列を限定的な線形計画問題により多く追加することで、問題によってその割合は異なるものの、1本しか追加しない場合に対して反復回数が減り計算時間が大幅に短縮していることがわかる。

その一方で、列が多くなると変数が多くなることで整数計画問題としての規模が大きくなり、その問題が行路の組合せとして良いものを含む可能性が高くなるものの、その良い組合せを見つけ難くなると予想される。これに対しては、

- 追加列数の上限が1や5の場合は整数解の精度は良いとはいえない
- それに対して、上限をさらに大きくした場合ではそれぞれのケース間で目的関数値に顕著な違

いがあるとはいえ、また追加本数の上限として何本が最善かの判定もできないなどのことが観察される。すなわち、線形計画問題の最適化を高速化する上では負の被約費用を持つ列の追加を1本に限定しないほうが良く、またこのことが良い整数実行可能解を問題に持たせる結果に至らせものと考えられる。

b) 列生成子問題の解き方の影響

さらに、表5から表8にも注目し比較する。方法1と方法2とでは追加する列数の上限が同一でも、列生成子問題の解き方が違うので、生成される列の内容が異なる。したがって、生成される列の本数や線形計画問題の最適化に要する時間も異なるが、いずれの問題でも線形計画問題の最適化の計算時間が短縮していることが確認できる。

一方、表17の評価項目の値を見ると、SCPの方法1と方法2とでは全体としては方法2のほうが優れている。特に、便乗時間や便乗数は方法2の優位が明白である。

c) 便乗に対する考察

便乗を抑制することが定式化には考慮されていないため、結果として生じる便乗時間や便乗数は方法に依存せず大きい。ただし、中には便乗時間や便乗数が少ないケースもあり、そのような場合は総勤務日数も良い。

5. 2. 2 一般化集合分割問題としての定式化

乗務 i に対する便乗のペナルティ(d_i)は、「乗務 i の乗務時間(分) $\div 10$ 」として計算実験を行った。そして、表9から表12に方法1の結果を、表13から表16に方法2の結果を示した。

a) 目的関数の値

まず、方法1で解いた場合の結果である表9から表12に注目しよう。目的関数は総勤務日数と便乗に対するペナルティの和であるので、その値(表内のIP)を実務的な観点から解釈することはできない。しかし、線形計画問題の最適値(表内のLP最適値)すなわち下界値と比較をすると、集合被覆問題の場合に比べてその開きが大きく、解の精度は劣っている。これは、整数解を求める観点からは、集合被覆問題とはなんらかの意味で問題の性質が異なることを示唆していると考えられる。

表 1 SCP による定式化 (乗務数488、方法1)

列追加上限	1	5	10	50	100	250	500	1000
列本数	1938	3852	4881	8504	10316	11414	12837	13301
反復回数	324	133	88	48	51	44	42	38
LP 最適値	114.667	114.667	114.667	114.667	114.667	1114.667	114.667	114.667
時間	10167	5686	3071	2182	2540	2224	2637	2132
総勤務日数	128	124	123	121	120	122	122	118
便乗時間	5191	3754	2956	2240	2455	3105	2276	1543
便乗数	76	56	46	34	34	46	36	22

表 2 SCP による定式化 (乗務数501、方法1)

列追加上限	1	5	10	50	100	250	500	1000
列本数	1704	3525	4864	8250	9300	10175	10708	11772
反復回数	285	120	86	45	41	41	37	40
LP 最適値	118.333	118.333	118.333	118.333	118.333	118.333	118.333	118.333
時間	5882	2849	1953	1114	1099	1187	1183	1218
総勤務日数	133	129	127	126	126	125	127	127
便乗時間	6191	4072	3304	3279	2531	2532	3625	6225
便乗数	91	63	53	49	43	43	53	51

表 3 SCP による定式化 (乗務数503、方法1)

列追加上限	1	5	10	50	100	250	500	1000
列本数	1887	3340	4314	8533	8620	9409	10250	13175
反復回数	316	114	77	49	42	37	39	44
LP 最適値	119.333	119.333	119.333	119.333	119.333	119.333	119.333	119.333
時間	6155	2728	1639	1415	961	1009	1295	1460
総勤務日数	133	130	129	128	127	126	126	126
便乗時間	5274	3732	3714	3243	2651	2251	2356	2578
便乗数	79	57	55	51	43	37	37	41

表 4 SCP による定式化 (乗務数526、方法1)

列追加上限	1	5	10	50	100	250	500	1000
列本数	2172	4097	5592	9728	11645	13559	12787	13449
反復回数	363	138	98	55	52	62	49	45
LP 最適値	123.667	123.667	123.667	126.667	126.667	123.667	123.667	123.667
時間	9805	4017	2696	1915	1841	2422	1702	2128
総勤務日数	138	135	131	130	130	129	130	130
便乗時間	5188	3647	2812	2335	2562	2015	2526	2407
便乗数	86	68	50	42	42	36	42	38

表 5 SCP による定式化 (乗務数488、方法 2)

列追加上限	1	5	10	50	100	250	500	1000
列本数	3895	5845	5849	11434	10709	9256	10644	13008
反復回数	798	255	138	146	113	96	113	126
LP 最適値	114.667	114.667	114.667	114.667	114.667	114.667	114.667	114.667
時間	6942	2744	1487	2017	1575	1286	1238	1573
総勤務日数	121	120	120	118	119	120	119	118
便乗時間	2698	2382	2253	1584	1856	2246	1873	1491
便乗数	38	36	32	24	26	34	30	24

表 6 SCP による定式化 (乗務数501、方法 2)

列追加上限	1	5	10	50	100	250	500	1000
列本数	2477	4051	4615	7820	7247	8010	8124	8980
反復回数	426	154	98	77	72	68	68	71
LP 最適値	118.333	118.333	118.333	118.333	118.333	118.333	118.333	118.333
時間	1879	1170	884	820	722	716	716	730
総勤務日数	128	125	125	125	125	126	126	124
便乗時間	4209	2697	2949	2523	2739	3000	3027	2419
便乗数	61	45	45	41	41	45	47	37

表 7 SCP による定式化 (乗務数503、方法 2)

列追加上限	1	5	10	50	100	250	500	1000
列本数	3892	6161	5560	9926	8025	9091	9352	9742
反復回数	816	286	136	115	82	95	92	92
LP 最適値	119.333	119.333	119.333	119.333	119.333	119.333	119.333	119.333
時間	4440	2001	884	863	722	628	659	661
総勤務日数	125	123	126	123	126	125	125	125
便乗時間	2771	1869	3135	1757	2574	2197	2449	2672
便乗数	39	27	45	27	39	37	39	37

表 8 SCP による定式化 (乗務数526、方法 2)

列追加上限	1	5	10	50	100	250	500	1000
列本数	2952	4873	6214	8795	8562	9265	9354	9374
反復回数	509	202	142	115	103	106	101	96
LP 最適値	123.667	123.667	123.667	123.667	123.667	123.667	123.667	123.667
時間	2302	1418	997	810	889	797	839	799
総勤務日数	134	131	130	128	129	129	128	131
便乗時間	3864	2878	2969	1792	2389	2277	2121	3107
便乗数	62	48	46	32	38	40	32	48

表 9 GSPP による定式化 (乗務数488、方法1)

列追加上限	1	5	10	50	100	250	500	1000
列本数	2356	4155	5580	10317	10126	11853	12868	14420
反復回数	395	142	105	69	62	66	60	61
LP 最適値	131.7	131.7	131.7	131.7	131.7	131.7	131.7	131.7
時間	10792	3831	2922	2154	1779	2206	1691	1966
IP	497.6	405.1	360.6	254.1	270.8	278.5	257.6	261.4
総勤務日数	133	130	129	130	133	130	130	128
便乗時間	3650	2752	2317	1242	1378	1485	1276	1334
便乗数	62	48	40	24	24	28	26	24

表 10 GSPP による定式化 (乗務数501、方法1)

列追加上限	1	5	10	50	100	250	500	1000
列本数	2388	4422	5618	9834	10637	10923	11370	12009
反復回数	399	152	101	67	64	61	60	56
LP 最適値	131.967	131.967	131.967	131.967	131.967	131.967	131.967	131.967
時間	5644	2559	1400	1138	1015	930	1102	886
IP	563.5	383.2	358.9	269.4	272.1	271.2	276.9	265.5
総勤務日数	140	131	134	133	135	136	133	134
便乗時間	4235	2522	2249	1364	1371	1352	1439	1315
便乗数	75	47	43	27	29	29	29	25

表 11 GSPP による定式化 (乗務数503、方法1)

列追加上限	1	5	10	50	100	250	500	1000
列本数	2279	3986	4914	8623	9811	10965	11141	11503
反復回数	381	137	90	60	64	61	60	56
LP 最適値	142.367	142.367	142.367	142.367	142.367	142.367	142.367	142.367
時間	4084	1507	1085	800	897	918	992	777
IP	582.0	450.7	386.9	318.1	303.2	280.8	236.0	295.5
総勤務日数	142	137	138	138	134	139	131	134
便乗時間	4400	3137	2489	1801	1692	1418	1050	1615
便乗数	77	55	41	33	31	27	21	31

表 12 GSPP による定式化 (乗務数526、方法1)

列追加上限	1	5	10	50	100	250	500	1000
列本数	2406	4405	5639	10470	11392	11243	12497	14443
反復回数	402	154	100	71	64	62	62	62
LP 最適値	141.55	141.55	141.55	141.55	141.55	141.55	141.55	141.55
時間	6165	3012	1734	1637	1445	1690	1318	1383
IP	563.3	463.9	354.1	350.7	303.0	299.4	286.8	302.8
総勤務日数	146	142	143	148	143	142	143	140
便乗時間	4173	3220	2113	2027	1600	1574	1440	1632
便乗数	76	62	42	40	32	30	32	38

表 13 GSPP による定式化 (乗務数488、方法 2)

列追加上限	1	5	10	50	100	250	500	1000
列本数	5768	7754	10015	12284	12250	11006	13954	15274
反復回数	1124	321	252	179	170	130	179	182
LP 最適値	131.7	131.7	131.7	131.7	131.7	131.7	131.7	131.7
時間	12658	3271	2775	1567	1909	1778	1675	1914
IP	283.1	251.6	219.1	253.7	227.8	240.2	226.3	246.9
総勤務日数	127	123	123	125	122	127	124	127
便乗時間	1564	1286	961	1287	1058	1133	1024	1199
便乗数	30	22	16	24	20	20	18	24

表 14 GSPP による定式化 (乗務数501、方法 2)

列追加上限	1	5	10	50	100	250	500	1000
列本数	5961	6581	9329	14413	12492	10604	12801	11785
反復回数	1231	262	236	226	184	138	168	137
LP 最適値	131.967	131.967	131.967	131.967	131.967	131.967	131.967	131.967
時間	8835	1241	1332	1761	1142	667	914	809
IP	301.1	333.6	265.9	241.6	235.9	242.2	245.0	253.0
総勤務日数	129	133	129	127	128	130	130	129
便乗時間	1721	2006	1369	1146	1079	1122	1150	1240
便乗数	35	37	25	23	23	25	23	23

表 15 GSPP による定式化 (乗務数503、方法 2)

列追加上限	1	5	10	50	100	250	500	1000
列本数	4992	5778	7202	10536	11327	10720	11265	12000
反復回数	1028	240	175	141	142	133	129	137
LP 最適値	142.367	142.367	142.367	142.367	142.367	142.367	142.367	142.367
時間	5582	1297	982	908	933	921	830	931
IP	345.8	358.9	327.4	281.6	268.4	262.6	253.5	250.6
総勤務日数	133	136	132	131	130	130	131	131
便乗時間	2128	2229	1954	1506	1384	1326	1225	1196
便乗数	39	41	37	29	23	25	25	21

表 16 GSPP による定式化 (乗務数526、方法 2)

列追加上限	1	5	10	50	100	250	500	1000
列本数	3311	5836	6904	9895	10264	10197	10569	11086
反復回数	561	224	159	139	137	130	119	125
LP 最適値	141.55	141.55	141.55	141.55	141.55	141.55	141.55	141.55
時間	1889	885	718	833	755	798	876	940
IP	443.6	341.0	329.7	278.5	289.9	290.4	309.7	312.6
総勤務日数	145	136	140	137	141	136	139	139
便乗時間	2988	2052	1897	1418	1490	1544	1707	1736
便乗数	62	40	36	28	32	28	36	34

表 17 実験結果のまとめ (評価項目の平均)

問題(乗務数)	評価項目	SCP		GSPP	
		方法 1	方法 2	方法 1	方法 2
488	総勤務日数	122.250	119.375	130.375	124.750
	便乗時間	2940.000	2047.875	1929.250	1189.000
	便乗数	43.750	30.500	34.500	21.750
501	総勤務日数	127.500	125.500	134.500	129.375
	便乗時間	3969.875	2945.375	1980.875	1354.125
	便乗数	55.750	45.250	38.000	26.750
503	総勤務日数	128.125	124.750	136.625	131.750
	便乗時間	3224.875	2428.000	2200.250	1618.500
	便乗数	50.000	36.250	39.500	30.000
526	総勤務日数	131.625	130.000	143.375	139.125
	便乗時間	2936.500	2674.625	2222.375	1854.000
	便乗数	50.500	43.250	44.000	37.000

b) 総勤務時間および便乗の抑制効果

表17内の SCP の方法 1、方法 2 と GSPP の方法 1 を比較しよう。集合被覆問題のふたつの方法の結果に比べて、一般化集合分割問題の総勤務日数は悪化している。便乗時間は一般化集合分割問題のほうが確実に小さいが、その差が小さい場合もある。便乗数は、平均でみると集合被覆問題の方法 1 に対しては一般化集合分割問題が勝っているが、集合被覆問題の方法 2 と比べると問題によって優劣が異なり便乗数の抑制効果が顕著であるといえない。

また、表 9 から表12より、列追加上限が小さい場合は、一般化集合分割問題としての定式化であっても便乗時間や便乗の絶対数は必ずしも小さいとはいえず、「より多くの列から選ぶ」ような状況でないと良い整数解を得るのが難しいことが分かる。

c) 列生成子問題の解き方の影響

方法 2 にしたがった場合の結果である表13から表16も注目の対象に含め、まず一般化集合分割問題の方法 1 との比較をし、次いで集合被覆問題から得られた結果とも比較する。

表13から表16を見ると、線形計画問題の最適化にかかる時間に関する方法 1 との優劣は、問題や列の追加本数上限によって結果が異なり、その傾向は把握しづらい。その意味では、計算時間の短縮に関して良い影響を与えているといい切れないが、方法 2 を採用することを否定するほどの結果ではない。

一方、表17において、総勤務日数や便乗時間、便乗数は、平均レベルにおいて方法1に比べて良い。もちろん、列の追加上限が同じ場合の個々のケースにおいて方法 2 が必ずしも良いとはいえない

場合も見受けられるが、一般的に一般化集合分割問題の方法2が安定的に良い答を算出していると評価できる。さらに、表17において、集合被覆問題の方法2と比べた場合、総勤務日数は劣るものの便乗時間や便乗数では確実に勝っている。すなわち、方法2同士で比べた場合、

- 総勤務日数では、集合被覆問題による定式化の結果が良い
- 便乗時間と便乗数では、一般化集合分割問題による定式化の結果が勝るといえる。

5. 2. 3 数値実験全体を通じて

一般に、整数実行可能解を得る観点からは列が少なければ問題の規模は小さくなり計算の手間は少なくなるはずだが、その代償として元の問題から見た場合に良い整数解が含まれない可能性が高まる。その一方で、問題の列が多ければ組合せ数は多くなるものの、選択の余地が広がるため良い解が含まれる可能性が高くなるが、それを得るために手間がかかる可能性がある。

総列数が最大でも概ね15000程度の整数計画問題を解いている今回の実験からは、変数が多いことが結果的に得られる整数解の精度に悪影響を与えたようなケースは明瞭には見られず、列を多く追加したり、結果的に列数が多い問題を解くことの弊害は顕在化しなかった。しかし、より大規模な問題を解く場合において追加する列数の上限をどう設定すべきかに関しては、別途実験を行って検証する必要がある。また、集合被覆問題と一般化集合分割問題とでは、整数解を得るための計算時間が同じにもかかわらず、後者における整数実行可能解の目的関数値の下界値からの乖離が大きく、問題がなんらかの意味で難しいことを示唆している。その意味で、より良い整数解をどのように見つけるかが今後の検討課題の一つになると考えられる。

6 結論

本研究では、複数の乗務員基地をもつ鉄道乗務員スケジューリング問題に対する列生成法に基づくヒューリスティック解法の性能を、現実に基づく問題を用いた数値実験によって評価した。さらに、定式化の変更による便乗を抑制の可否に関しても検討を行った。

本研究が提示した方法は最適解を提供する保証は無いものの、線形緩和問題を解く際に列を生成させることにより、事前に行路を列挙して解くアプローチに必須の列を選択する基準や手順を代替させて、整数計画問題で対象とする列を限定しつつ下界値を得ることで整数解の評価を可能にしている。また、この解法は実装の面からも分枝価格法ほど難しくなく、実用性は高いと考えられる。

今後の課題としては、一般化集合分割問題においてさらに良い実行可能解を求めることの他に、

- 列生成子問題をさらに高速化し、全体の計算時間を短縮する

- 行路のコストの精密化を行い、このことでどのようなスケジュールが得られるかを観察する
 - 分枝価格法を実装し、その性能の評価を行う
- などのことが挙げられる。

付録

5. 2. 2節で一般化集合分割問題の定式化での目的関数について述べたが、問題によっては時間が15秒単位で与えられているため、それを分単位に表現し直した時点で小数点以下を切捨てて整数化した上で10で割ったものを、各便乗に対するペナルティとしている。

(本研究は、東洋大学平成17年度海外特別研究の成果の一部である。)

参考文献

- [1] E. Abbink, M. Fischetti, L. Kroon, G. Timmer, and M. Vromans. Reinventing crew scheduling at Netherlands railways. *Interfaces*, Vol. 35, No. 5, pp. 393–401, 2005.
- [2] 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.
- [3] J. Desrosiers, Y. Dumas, M. Solomon, and F. Soumis. Time constrained routing and scheduling. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Network Routing*, Handbooks in Operations Research and Management Science, chapter 2. North-Holland, 1995.
- [4] R. Freling, R. Lentink, and A.P.M. Wagelmans. A decision support system for crew planning in passenger transportation using flexible branch-and-price algorithm. *Annals of Operations Research*, Vol. 127, pp. 203–222, 2004.
- [5] J. Imaizumi, H. Arai, K. Ogawa, S. Morito, and N. Fukumura. Column generation approaches to scheduling problems in logistics. In *Proceedings of the 1st International Congress on Logistics and SCM Systems*, pp. 136–143, 2004.
- [6] S. Lavoie, M. Minoux, and E. Odier. A new approach for crew pairing problems by column generation with and application to air transportation. *European Journal of Operational Research*, Vol. 35, pp. 45–58, 1988.
- [7] G. Mitra and K. Darby-Dowman. Cru-Sched - a computer base bus crew scheduling system using integer programming. In J. M. Rousseau, editor, *Computer Scheduling of Public Transport 2*, pp. 223–232. North-Holland, 1985.
- [8] 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. 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.
- [9] 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. スケジューリング学会, 2004.
- [10] 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. 日本オペレーションズ・リサーチ学会, 2005.

(2007年1月16日受理)