論文・研究レポート

鉄道の乗務員運用計画作成問題に対する列生成法の適用

今泉　淳，植田　達広，森戸　晋

1. 目的と背景

鉄道業では，ダイヤのとおりに列車を運行するために，車両や乗務員をダイヤ上の列車へ対応付ける計画が必要となる．

乗務員は列車の運行に欠かせない一方で，その勤務内容は労働上のルールにより制約が課されており，限られた数の乗務員を各列車に必ず充当しつつ効率的に運用する計画立案（乗務員運用計画作成問題）は実務的にも難しい問題である．この乗務員運用計画の作成は鉄道会社にとって共通の問題だが，日本でのORによる解決例は諸外国に比べてまだ少ない．

この問題を整数計画問題として表現した場合，膨大な変数への対策を要したり，乗務員のスケジュール候補の用意に関する問題が発生する．一方，大規模な整数計画問題に対しては，列生成法のような分解を前提にした解法が有望視されている．特に，諸外国ではこの種の問題に対する列生成法の適用例が多く見られるが，国内での試みは少ない．

本研究では，その試みの一つとして，乗務員運用計画作成問題に対して列生成法を用いたヒューリスティック解法を構築する．そして，日本国内の具体的な線区のダイヤに基づくインスタンスを用いてその性能を検証し，現実的な規模の問題に対する列生成法に基づく方法の可否に関する議論をする．

2. 乗務員運用計画作成問題

以下に，本研究が想定する乗務員運用計画作成問題の定義とその定式化を示す．

2.1 問題と用語の定義

時刻表により始発駅や終着駅，途中停車駅の発車時刻や到着時刻が与えられる「列車」に対して，各列車を乗換可能駅で分割したものを「乗務」（図1），乗務員の基地からその基地までの一回の勤務を構成する一連の乗務の集合を「行路」（図2）と呼ぶ．図1と図2ではA駅からD駅での乗務員の乗り換えが許されており，図2は，ある乗務員はA駅から勤務を開始し，再びA駅に戻って勤務を終了する例である．

行路は，勤務時間が1暦日内に収まる「日勤」と，2暦日に跨る「夜勤」の二種類があり，スケジュール全体の評価はダイヤ上の列車の運行に必要な乗務員の総勤務日数とし，日勤および夜勤の行路のコスト（勤

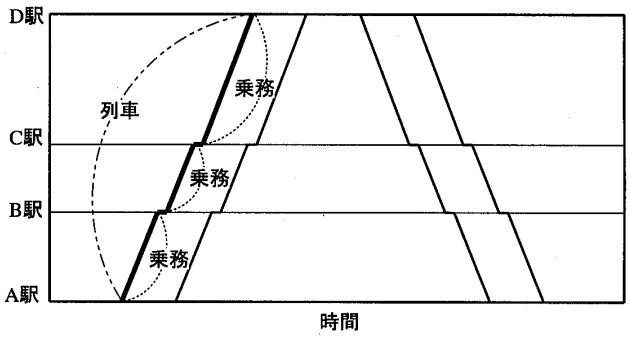

図1　列車と乗務

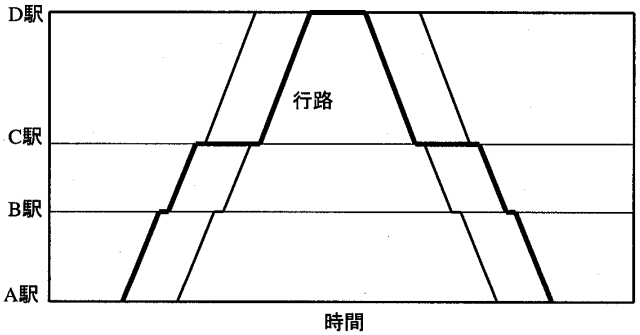

図2　行路

いまいずみ　じゅん
東洋大学 経営学部
〒112-8606 文京区白山5-28-20
うえだ　たつひろ，もりと　すすむ
早稲田大学 創造理工学部
〒169-8555 新宿区大久保3-4-1
受付10.2.24　採択10.12.2

務日数）を，それぞれ1および2とする．このとき，乗務員運用計画作成問題は「すべての乗務を被覆する，総勤務日数最小の行路の集合を求める問題」と定義できる．

行路には，勤務時間・乗務時間等に関する多数の制約が存在する．本研究で考慮したのは，勤務・乗務・睡眠・休憩のそれぞれの時間の上下限，継続乗務（連続する異なる乗務の時間間隔が一定以下の場合「継続乗務」の扱いになる）の時間および距離の上下限，乗務数の上下限である．

なお，ここでは2人以上の乗務員が同じ乗務に携わる便乗を許すものとする．

2.2 集合被覆問題による定式化

乗務を行に，行路を列に対応させた上で，集合被覆問題（Set Covering Problem, SCP）として定式化する．

ここで，Mを乗務の集合，Nを行路の集合，c_jを行路$j(\in N)$のコスト，a_{ij}（定数）を乗務$i\in M$が行路jに含まれるとき1，さもなくば0とし，変数としてx_jを行路jを選択するとき1，さもなくば0となるような0-1変数とすると，問題は以下に定式化できる．

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

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

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

(1)式は勤務日数最小化を意味する．(2)式は全乗務の被覆制約を表している．

3. 過去の研究と本研究の関係

3.1 過去の研究

乗務員運用計画作成問題は航空や鉄道に発生し，一般に「クルースケジューリング問題」として知られる．

欧米では航空機のクルースケジューリングに関する研究が従来から盛んである．一方，鉄道分野への最適化手法の適用が活発になったのは比較的最近のことである[3]．いずれにせよ，すでにある成果を転用するのは自然な流れではあるが，それにあたっては，鉄道の実際のダイヤに基づくインスタンスにより事前に評価することが必要になることはいうまでもない．

クルースケジューリング問題は，集合被覆問題または集合分割問題（Set Partitioning Problem, SPP）として定式化することが多い．その場合のアプローチは，列をあらかじめ列挙しておく「事前列挙型」と必要に応じて列を生成しながら問題を解く「列生成型」のふたつに分類できる[5]．

すべての列を陽に列挙して最適解を求めることは変数の数の面から困難を伴うため，最適解を得る方法は現実的には分枝限定法に列生成法を組み込んだ分枝価格法[1]に限られる．しかし，その実装は必ずしも容易ではない．

それに対して，許容される列を部分的に列挙して実行可能解を得るのは，手軽な方法であり実務的にも有力である．もちろん，このようにして例えばSCPを最適に解いても，本来の問題に対する最適性は保証されない．しかし，列の数やその選び方の問題が解決すれば，SCPやSPPを解く手段次第では有力な方法にもなり得る．よって，このようなアプローチ（例えば近似解法のCaprara et al.[2]や三浦ら[6]）も代替的手段として考慮すべきであろう．

3.2 本研究のアプローチ

本研究では，すべての列を対象とする問題の下界値が得られるまでを列生成法で，下界値が求まったならば得られた列を対象に整数解を求めるアプローチをとる．これは，列生成型に属するものの，事前列挙型と列生成型の折衷案的な方法ともいえる．

このようなアプローチを採用した理由は，主として以下のようなことが挙げられる．

事前列挙型の代替的方法として　事前列挙型には前述した列の数や選び方の問題がある．そこで本研究では，線形緩和問題を解く際に列を生成するプロセスを列の選び方に対する手段と見なす．そうすることで列数を限定し，列の数に対する間接的な対策とする．

もちろん，元の整数計画問題とその線形緩和問題とでは最適解は一般的に一致せず，また線形緩和問題を解くために生成した列から得た整数解の質に関する問題は残る．この点は後段の実験によって評価する．

下界値による評価が可能　線形緩和問題の最適解は元の問題に対する下界になり，整数解の評価の材料になる．

商用パッケージの利用　実務においてこの種の方法を活用するためには，ゼロから実装を行うのではなく商用パッケージの併用も手段の一つとなろう．昨今のパッケージの性能は極めて高いが，このような用途にどの程度耐え得るかを検証することも必要である．

列生成法そのものの特性分析　列生成法は，筆者らの知る限り，国内の現実規模の問題例に対する実験的分析は必ずしも十分ではなく，実用上の目安を示す意味でも，具体的な問題例に対する解の精度や計算時間など数値結果を示すことには十分な意義があると考

える．

このような観点から，本研究は，既存の解法を日本の具体的事例に対して適用したものであり，方法論的な意味での新規性はないが，現実に基づく鉄道乗務員運用計画作成問題に対して，列生成に基づく方法を実装しまたその性能を数値実験で分析し，実務での使用に資することを目指して，解法の特徴に関して明らかにする．

4. 解法の詳細

4.1 解法の概要

前述の通り，本研究の解法は，「元の問題の線形緩和問題を列生成法で解く」「線形緩和問題の最適解を得るために生成された列に基づき整数解を得る」の二つの部分からなる．

前者は，列を生成するための子問題の最適化を必要とする．以下にその概要を記す．後者では解法の選択には自由度があるが，ここでは商用パッケージ（分枝限定法ベース）を用いて整数解を得る．

4.2 列生成法

問題（P）の線形緩和問題（LP）を考える．

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

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

$$x_j \geq 0, \quad \forall j\in N \tag{6}$$

さらに，Nの部分集合$\bar{N}(\subseteq N)$とその要素に対応する列が手元にある前提の下で，それらの列からなる問題（限定線形緩和問題，RLP）を考える．

$$\text{(RLP)} \quad \min \quad z=\sum_{j\in \bar{N}} c_j x_j \tag{7}$$

$$\text{s.t.} \quad \sum_{j\in \bar{N}} a_{ij}x_j \geq 1, \quad \forall i\in M \tag{8}$$

$$x_j \geq 0, \quad \forall j\in \bar{N} \tag{9}$$

RLPのある実行可能基底解のLPに対する最適性は，(5)式に対応する双対変数をπ_iとしたとき，被約費用の符号，すなわち，

$$\xi=\min_{j\in N}\{c_j-\sum_{i\in M}\pi_i a_{ij}\} \tag{10}$$

のξの符号から判定できる．これを負にする$j\in N$が見つかれば目的関数の値は改善が可能，さもなくば現在のRLPに対する実行可能基底解はLPに対して最適である．

もちろん(10)においてNのすべての要素に対応する列すべてが列挙されているわけではないので，その最小値は直ちには求め得ない．しかし，各列のa_{ij}の内容（0もしくは1）は行路が満たすべき条件によって制約されているため，ξを求めることは最適化問題に帰着する．これを「列生成子問題」と呼ぼう．

4.3 列生成子問題

ネットワーク上の最短路問題　列生成子問題は，多資源制約付の最短路問題に帰着できる．すなわち，乗務員の基地からの出発（行路の開始）と基地への帰還（行路の終了）をそれぞれ始点sと終点tに，乗務を点に対応させ，乗り継ぎが可能な乗務に対応する点の間を有向枝で結んだネットワークを考える．

すると，ξを求める問題は，2.1節で述べた行路に関する各種項目それぞれを資源と見なして，枝(i, j)の通過に各資源の消費を伴いつつ，頂点i（乗務iに対応）の通過に$-\pi_i$のコストがかかる，各資源の使用量の上下限を守るようなsからtへの最小コストのパスを求める問題となる．

列生成子問題の定式化と解法　T_{ij}を枝(i, j)の通過に対して伴う資源消費量を表すベクトル，RをT_{ij}と同じ次元のベクトルと約束し，$\mathrm{Cost}(i, R)$を「始点sから頂点iまでの，資源消費量Rのパスの最小コスト」とすると，以下の漸化式が成立する．

$$\mathrm{Cost}(s, \mathbf{0})=0 \tag{11}$$

$$\mathrm{Cost}(j, R)=\min\{\mathrm{Cost}(i, R')-\pi_j \mid R'=R\ominus T_{ij}, (i, j)\in \text{枝集合}\} \tag{12}$$

ここで，$\mathbf{0}$は各資源の消費量がいずれも0であることを表すベクトル，$\ominus$は，継続乗務以外の資源に関しては単純な引き算として，継続乗務制約に関しては乗り継ぎが継続乗務になるか否かによって適当に処理するものとする．

さらに$R^{\min}$と$R^{\max}$はそれぞれ行路制約の資源消費量の下限および上限を表すベクトルとすると，(10)のξは$R^{\min}\leq R\leq R^{\max}$を満たす$s$から$t$への最小コストパスによって定まるので，

$$\xi=\min\{\mathrm{Cost}(t, R) \mid R^{\min}\leq R\leq R^{\max}\} \tag{13}$$

となる．

本研究ではpull型ラベリング解法[4]を採用してこれを解いている．pull型ラベリング解法とは，先行する頂点までの部分パスをもとに順次パスを拡大してゆく方法である．すなわち，この漸化式を満たす部分パスに対応する$\mathrm{Cost}(j, R)$を開始時刻の早い乗務（頂点）から順次拡張し，最小コストパスを求める．

なお，資源制約のない最短路問題をDijkstra法で解く場合には頂点とラベルが一対一対応するが，複数

の資源を考慮する必要がある本問題では，各頂点に複数のラベルがつく．このような解法の性質上，始点から終点へは最小費用のパス以外にも複数のパスが生成される．後の数値実験では，このことを利用してLP最適解を得るまでの時間を短くし得るかを考察する．

5. 数値実験

5.1 実験の概要

数値実験は，実際の線区のダイヤに基づく複数のインスタンスを使用した．これらは同一線区でダイヤ改正によって内容が異なるインスタンスであり，乗務数はいずれも500前後である．

実験は四種類のインスタンスに対して行い，全体的傾向はおおむね同一であることを把握している．しかし，この種の数値実験の結果の詳細を示すことには同様の方法を採用する他のケースの参考になり得ると考え，詳細を示すためにそれらのうちの二つを表として表2と表3，ひとつを図として図3と図4に示した．

本線区には乗務員の乗り換えが可能な駅が4つあり，そのうちの三カ所に乗務員基地がある．行路には日勤と夜勤の二種類があるため，六つの列生成子問題を解くことになる．

数値実験は，Pentium 4 3.2 GHz（メモリ2Gバイト，Windows XP Professional）のパーソナルコンピュータ上で行い，線形計画問題の最適化（単体法）にはCPLEX 10.1を用い，整数解はXPRESS-MP Release 2006を用いて算出した．以下，各インスタンスはその乗務数（例えば「501」など）をもって識別することにする．

数値実験では

- 線形緩和問題の最適化に要する時間
- 整数解の精度

が興味の対象となる．

なお，実験結果の表の項目に関しては表1に説明を記した．

5.2 実験結果と考察

一般に，単体法の各反復では負の被約費用の列に対応する変数を基底として加えれば目的関数値を改善できる．ところで，本研究の子問題に対する解法はその性質上，生成されるパス（列）は最小費用の列に限られず，他にも実行可能なパスが得られることが多い．

一方，線形緩和問題が最適に解ければ生成された列数に無関係に理論的な下界は保証されるが，そのための計算時間は各反復での列の追加の方策に依存する可能性がある．

また，線形緩和問題が解けた時点までに生成された列が良い整数解を生むかも分からない．一般に，列数が少ないほど整数解を得る意味で元の問題との乖離が大きくなる．一方，列数が多くなると元の問題に近づ

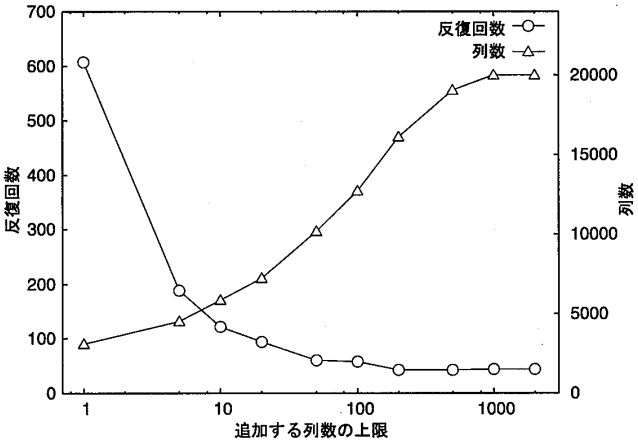

図3 追加する列数の上限と反復数・列数（501）

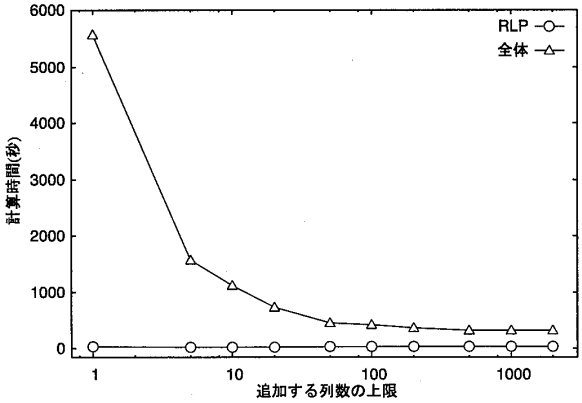

図4 追加する列数の上限とLP最適化にかかる時間（501）

表1 実験結果の表内の項目の説明

上限	列生成法における一反復において一子問題から追加する列数の上限
反復	列生成法の反復回数
列数	生成された総列数
LP最適値	線形緩和問題の最適値
計算時間	線形緩和問題の最適値を得るのに要した時間（内訳と合計）
RLP	RLPの最適化に要した時間
CGSP	列生成のために最短路問題を解いた時間
合計	上記二つの合計
IP	整数解の目的関数値
算出時間	LP最適値を得た後，分枝限定法を開始して10000秒を計算時間の上限とした際，上記の値を算出するまでにかかった時間（秒）

くがゆえに元の問題の最適解を持つ可能性が高まるものの変数が増えるというジレンマが生じる．

よって，列生成のプロセスで被約費用が負で最小の列のみならず他の負の費用の列があった場合にそれも同時に追加するか否か，またその本数の多寡は，線形緩和問題の最適化のための時間のみならず，整数解の精度やその計算時間へも影響が及ぶ可能性があり，これらの点に検証を要する．

これらに関して，以下の二段階で考察する．

a）追加する列数の上限のLP最適化への影響

i）列数の上限と反復回数および総列数　図3は，横軸が追加する列数の上限，左側の縦軸が列生成法の反復回数，右側の縦軸が線形緩和問題が解けるまでに生成された総列数だが，この図からは，

- 上限が低い（横軸の左側）と，合計の列数（△）は少ないが列生成の反復回数（○）は多い
- 上限が高くなる（横軸で右に行く）に連れて，合計の列数が多くなる反面，列生成法の反復回数は少なくなる

という傾向が認められる．

これらは表2と表3の左から2列目と3列目を上から下に見ても確認でき，上限の変化を1から2000で見た場合，反復数は10分の1程度に減るが列数は数倍に増加する．

表2　488の問題に対する結果

上限	反復	列数	LP最適値	計算時間（秒）			整数解	
				RLP	CGSP	合計	IP	算出時間
1	509	2622		36.6	5361.7	5398.3	119	5742.8
5	172	4386		22.5	1919.8	1942.3	118	2370.2
10	125	5756		25.2	1208.8	1234.0	116	1636.7
20	86	7347		25.8	803.3	829.1	116	3644.4
50	60	9748		29.6	593.0	622.6	116	5648.8
100	48	12950	114.317	31.4	452.3	483.7	116	8511.7
200	48	15601		31.0	436.7	467.7	116	2506.3
500	51	17584		27.8	373.7	401.5	116	2718.2
1000	53	18185		28.3	420.4	448.7	116	1123.8
2000	50	18102		27.8	368.2	396.0	116	1653.3
なし	40	19287		28.9	325.3	354.2	116	1830.7

表3　503の問題に対する結果

上限	反復	列数	LP最適値	計算時間（秒）			整数解	
				RLP	CGSP	合計	IP	算出時間
1	634	3191		47.8	11818.5	11866.3	126	3843.1
5	193	4699		34.1	3235.6	3269.7	125	5010.8
10	125	5907		43.9	2110.2	2154.1	124	9270.3
20	91	6869		39.0	1216.7	1255.7	124	315.0
50	76	10158		45.8	936.2	982.0	123	2235.6
100	57	12637	119.983	42.0	648.8	690.8	122	5205.1
200	57	16800		44.3	650.3	694.6	122	1438.2
500	42	19389		38.3	563.5	601.8	122	4432.6
1000	60	21997		48.8	633.9	682.7	122	674.7
2000	59	22677		46.0	592.6	638.6	122	4693.8
なし	53	22667		46.9	621.7	668.6	121	9892.4

ii）列数の上限と計算時間　計算時間に関する図4の横軸を左から右に見ると，

- LP最適解が得られるまでの時間全体（△）は激減し，その時間と反復回数（図3の○）との間には強い関係がある
- 単体法の実行時間（図内の○，表でのRLP）の計算時間の全体（△，表では「合計」）に対して占める割合は極めて小さく，またRLPのグラフは極めて平坦である

と認められる．これらのことは，表2と表3それぞれの「計算時間」の3列を上から下へ見た場合も同様である．

これらから分かるように，単体法の計算時間の合計は追加列数上限にかかわらず安定的で，それは列生成の時間に比べてはるかに短い．つまり，線形緩和問題の最適化におけるネックは列生成子問題にある．しかし，追加列数の上限を大きくすることで一反復でより多くの列が追加され，結果的に反復回数が少なくなり，列生成子問題を解く回数が減るため計算時間全体は短くなる．

すなわち，列生成子問題を解いた際，列の被約費用が負であるが最小でなくても，それらをなるべく多く列集合に加えることで線形緩和問題を解くための時間を短くできる．列生成子問題の最適化の負荷が大きいのは，資源の種類が多いがゆえであるが，これを短くすることが計算時間の短縮のための次の課題になろう．

b）追加する列数の上限の整数解の質と計算時間への影響　ここでは，追加する列数の上限が，結果として解く対象となる集合被覆問題から得られる整数解の質とそれを得るための計算時間にどのような影響を与えるかが興味の対象である．

表2と表3の右端の二列を見ると，いずれのインスタンスでも，追加列数の上限が低い場合の整数解よりも上限が高い場合のほうが整数解の目的関数値が良い．特に，整数解の目的関数値が整数しかとり得ないことを考慮すると得られている下界は実質切り上げて良く，その場合の表内の最良の整数解（列の追加上限がもっとも大きい場合）の下界からの乖離は1程度である．一方，その整数解を求めるために要する時間と列数との間には関係があるとはいえず，個々のケースによるばらつきが大きい．

さきほど501の問題について具体的な数値やグラフを示さなかったが，同様の傾向を示している．

総じて，良い整数解を生み出すためには整数解を得

る対象になる集合被覆問題の列数（変数の個数）が大きくならないようにコントロールする必要性は小さいと考えられる．

5.3 補足

本論文で示した計算時間は，本研究で使用した環境（ハードウェア，ソフトウェア）に依存する．

ハードウェアについては，最新の高性能CPUを用いれば，同一の解をより短い時間で得られる，あるいは，同一の計算時間上限でより良い解が得られる，などが期待できる．

一方ソフトウェアに関しては，線形計画問題や整数計画問題の最適化に商用パッケージを用いた．線形計画問題は最適解法が必要だが，整数計画問題に対して，ユーザーが信頼する他の最適化エンジンや，精度と計算時間のトレードオフを勘案して近似解法を用いることも十分考えられよう．

また，本研究が用いた枠組みでは，整数解を求める際の計算時間の上限を問題の規模に応じて設定することになるが，問題規模の変化に伴う解法の挙動の観察やそれにより必要となる対策は，今後の検討課題とする．一般に乗務数の増加に伴い，列数が多くなり線形計画問題や整数計画問題の最適化の時間が増加し，多資源制約つき最短路問題のネットワークの大規模化により列生成の手間が増えると予想される．全体の計算時間を勘案した場合には，上記の近似解法の採用も有力な選択肢となろう．

6. 結論と今後の課題

本研究を通じて，列生成法による方法がおおむね実用に耐え得る計算時間で，下界を得つつ良い精度の整数解を提供することが判明した．その上で，

- 線形緩和問題の最適化の際に追加する列の本数は，一本に限定するのではなく複数追加することで，線形緩和問題の計算時間を大幅に短縮できる．
- より多くの列を追加して得た列群に基づく整数解の精度は良く，その一方で列数の増加による計算負担は認められない．すなわち，この問題規模では，列の数に関する問題は顕在化せず，整数解の精度のみに良い影響を与える．
- これらの結果からみて，線形緩和問題の最適解が得られるまでのプロセスにおける列生成法は，実行可能な行路集合から行路（列）を選別する方法の一つとして成立する．

などの知見を得た．

今後の課題としては，

1. 列生成子問題の最適化の高速化
2. 行路の特性や条件が顕著に異なるような線区での本解法の性能の評価，その結果に応じた対策
3. 分枝価格法の適用

などが挙げられる．

本研究では他の線区のインスタンスでの実験まで及んでおらず，インスタンスの特性によっては計算時間に影響を与えないとも限らない．それらの結果次第では，なんらかの対策の必要性を惹起し，さらには一番目や三番目の課題に対する現実的要請にもつながる可能性がある．

謝辞　本研究を行うにあたり，㈶鉄道総合技術研究所の協力を得ました．

参考文献

[1] 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*, 46 (1998), 316-329.

[2] A. Caprara, M. Fischetti and P. Toth, "A heuristic method for the set covering problem," *Operations Research*, 47 (1999), 730-743.

[3] A. Caprara, L. Kroon, M. Monaci, M. Peeters and P. Toth, "Passenger railway optimization," In C. Barnhart and G. Laporte, editors, *Transportation*, Vol. 14 of *Handbooks in Operations Research and Management Science*, chapter 3. North-Holland, 2007.

[4] J. Desrosiers, Y. Dumas, M.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*, Vol. 8 of *Handbooks in Operations Research and Management Science*, chapter 2. North-Holland, 1995.

[5] 今泉淳，「鉄道の運用計画問題に対する整数計画法によるアプローチ」，オペレーションズ・リサーチ，53 (2008), 439-445.

[6] 三浦礼，今泉淳，福村直登，森戸晋，「鉄道における乗務員運用計画の集合被覆問題に対するWedelinの解法の適用」，電気学会論文誌C, 129 (2009), 1958-1967.