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

乗務員運用計画の集合被覆問題に対するWedelin解法の適用

申請中　早稲田大学　三浦礼　MIURA Rei
01205890　(財)鉄道総合技術研究所　福村直登　FUKUMURA Naoto
01603200　早稲田大学　森戸晋　MORITO Susumu
01012600　東洋大学　今泉淳　IMAIZUMI Jun

1 研究の背景と目的

欧米ではパイロットや運転士のクルースケジューリングに関する研究ならびに応用が盛んに進められている(たとえば、Barnhart and Cohen[3] 参照) のに対して、国内では関連する手法の研究・適用は限られている。

クルースケジューリング問題は、通常、集合被覆問題 (SCP) または集合分割問題 (SPP) に定式化され、解法は、(1) 最適解法／近似解法、(2) 列をあらかじめ列挙しておく事前列挙／必要に応じて列を生成しながら問題を解く列生成、の 2 つの観点から分類できる。真の意味での最適解法は現在のところ列生成に基づく分枝価格法に限られるのに対して、最適解を保証しない解法として、(a) 事前に列挙された列から構成される大規模 SCP/SPP に対する最適解法の適用、(b) ラグランジュ緩和に基づく解法、(c) メタ解法、などがある。

ここでは、短時間でよい近似解を得るという観点から、SCP 定式化に対するラグランジュ緩和法の適用を考える。ラグランジュ緩和アプローチには、Caprara ら [4] の解法と Wedelin の解法 [1] がある。本研究では後者を取り上げ、主に事前列挙型問題に対して、実際の国内鉄道データをもとに Wedelin 解法の性能を評価するとともに、列生成型への展開について検討する。

2 乗務員運用計画問題

2.1 乗務・行路の定義と問題の提示

ダイヤは所与という仮定の下で、ダイヤ上の列車を乗換可能駅で分割したものを**乗務**、乗務員の 1 回の勤務を構成する一連の乗務の集合を**行路**と呼ぶ。スケジュールの評価はダイヤを運行するのに必要な乗務員の総勤務日数とし、日勤 (夜勤) 行路のコストを 1(2) とする。

このとき、乗務員運用計画問題とは、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である。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し (詳細省略)、これらの条件をすべて満足する行路案が事前に列挙されているものとする。また、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 (SCP) による定式化

M	乗務の集合
N	行路案の集合
c_j	行路 $j \in N$ のコスト
a_{ij}	乗務 $i \in M$ が行路 j に含まれるとき 1, さもなくば 0
x_j	行路 j を選択するとき 1, さもなくば 0

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

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

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

3 Wedelin 解法

3.1 基本的考え方

ラグランジュ乗数 $\pi = (\pi_1, \ldots, \pi_{|M|})$ を用い制約 (2) を緩和して、以下のラグランジュ双対問題を考える。

$$\max_{\pi} \; L(\pi)$$

$$L(\pi) = \min_{x_j} \sum_{j \in N} (c_j - \sum_{i \in M} a_{ij}\pi_i) x_j + \sum_{i \in M} \pi_i$$
$$\text{s.t. } (3), \quad \pi_i \geq 0, \quad \forall i \in M$$

Wedelin 解法の特徴はラグランジュ緩和を下界値算出の手段とは考えずに、上界値すなわち実行可能解算出の手段と考えるところにあり、直接、ラグランジュ双対問題の最適化を目指す訳ではない。乗数の更新は、劣勾配法を用いず、π の特定の要素 π_i に関する 1 次元探索を、各 $i \in N$ に対して繰り返すことにより更新するとともに、実行可能解が出やすいように被約費用に修正を加える (3.2 参照)。

ここで重要な点は、i 番目の制約に対応する乗数 π_i を $\pi_i + q$ と更新した緩和問題の解が、元の問題の i 番目の制約を満たす場合に $L(\pi + qe_i)$ が q に関して最大値をとる (e_i は第 i 要素を 1 とする単位ベクトル)、すなわち、π の 1 次元探索の最大値を与える q が i 番目の制約を満たす q の値と一致する点である (証明略)。以上をもとに、ラグランジュ乗数を以下の手順で更新する。

Step 1 行 i を含む列を被約費用の小さい順に並べる
Step 2 被約費用の小さい順に $r_{[1]}, r_{[2]}$ とする
Step 3 $q = (r_{[1]} + r_{[2]})/2$ を計算する
Step 4 π_i を $\pi_i + q$ に更新する

Wedelin 解法の問題点は、最適性の判定できない点、被約費用を修正しているために元の解法のままでは一般に下界値を提供しない点、事前に最適なパラメータの値を見つけられない点が挙げられる。

3.2 被約費用の修正と解法の流れ

次のラグランジュ乗数 π_{i+1} を更新した際に、それによる緩和問題の解が元の問題の i 番目の制約式を満たさない可能性があるため、以下の可変パラメータ p_{ij} を用いることで被約費用を修正する。

$r'_j = c_j - \sum_{i \in I(j)} (p_{ij} + \pi_i)$

使用する鉄道データは、被約費用の値の差が比較的小さいため $p_{ij} = t/h$ に変更することで Wedelin 解法を適用した [2]。終了判定基準は、上界値が一定時間更新されない、もしくは反復回数 t がパラメータ h を上回った時点で終了とする。

Step 1 $i := 1$

Step 2 ラグランジュ緩和問題を解き (r'_j が負なら $x_j = 1$、正ならば $x_j = 0$)、$\pi_i := \pi_i + q$、$p_{ij} = \pm t/h$ とする

Step 3 解が制約を満たしていたら、上界値を算出

Step 4 $i := i + 1$、$i \le |M|$ ならば Step 2 に戻る

Step 5 終了判定基準を満たせば終了、さもなくば Step 1 に戻る (1 反復終了)

4 計算実験による性能評価

4.1 実験のねらい

実験は、4 種類の実在線区のデータ (乗務数 488,501, 503,526) を用いる。実験目的は以下の通り：

1. パラメータ値により解の精度・計算時間に影響を与えることが予想されるため、鉄道データにおけるパラメータ分析、Wedelin 解法の特徴を調べる
2. 分枝限定法の結果と比較することで Wedelin 解法の性能を評価する

(使用マシン：Pentium4 3.20GHz 2GB RAM)

4.2 Wedelin 解法の実験結果

4.2.1 パラメータ感度分析

パラメータ h を変化させたときの最良値・計算時間の変化を図 1 に示す。右下がりの変化をしているのが最良値であり、右上がりの変化は計算時間である。

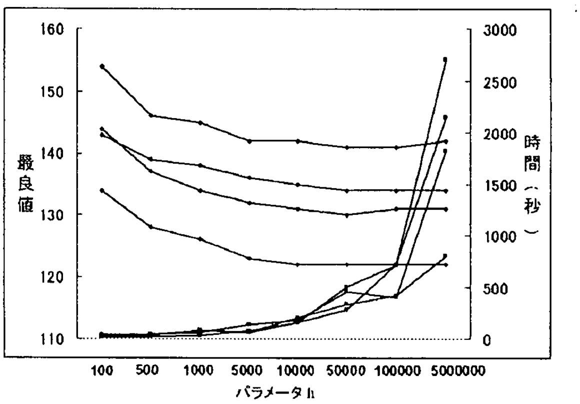

図 1: 最良値・計算時間の変化

一般にパラメータ h を大きく設定することで良い解が得られるが、一方で計算時間がかかる。しかし、h を大きくしすぎると逆に解が悪化する可能性もある。なお、今回の実験では 4 データともバックトラック法で生成した行路 (10 万弱) を使用した (表 1,2 も同様)。

また、パラメータ h の最適な値は行路数 (列数) によっても変化する (表 1 参照)。行路数が増えると h の最適な値が低下する傾向があるのは興味深く、行路数が膨大でも適正なパラメータ値を設定すれば比較的短時間で最良値算出が可能である。

表 1: 行路数による最適パラメータ値の変化

行路数	パラメータ h	最良値	計算時間
1611	100000	142	59
106830	50000	141	281
795607	10000	137	890

4.2.2 分枝限定法との比較

表 2: Wedelin 解法と分枝限定法の性能比較

乗務数	Wedelin 解法		分枝限定法	
	最良値	時間 (秒)	最良値	時間 (秒)
488	122	203	122	32594
501	130	505	130	9562
503	134	460	134	5660
526	141	281	141	8433

表 2 は、列数 10 万程度の問題に対する分枝限定法 (商用パッケージ) と Wedelin 解法の計算時間を比較している。Wedelin 解法は、分枝限定法の高々1 割程度の時間で「最適値」を算出している。これら 4 データではすべて最適解が算出されていることに注目したい。これは、Wedelin 解法が分枝限定法の分枝操作の代わりに、被約費用を修正することで状況に応じた最適な列を選択しているためと考えられる。また、表 1 の最後の行のように、分枝限定法で解を出せなかった問題においても、目的関数値をさらに改善する解が得られている。

4.2.3 Wedelin 解法による解の更新状況

表 3 は Wedelin 解法の解の更新状況をまとめている。
改善率 (%)=(初期値-最良値)/最良値× 100

表 3: Wedelin 解法の暫定解の変化 (h=50000)

行路生成	乗務数	暫定解の個数	改善率	反復回数	
				初期解	最良解
BT	488	3	3.3	374	508
	501	3	1.5	473	850
	503	6	4.5	255	647
	526	2	0.7	921	932
CG	488	3	1.6	5187	5189
	501	3	1.5	4861	4865
	503	2	2.2	5617	5617
	526	3	2.9	5649	5651

改善率に関しては、バックトラック (BT)・列生成 (CG) ともに低く、初期解算出時点で良い値が得られている。CG では初期解と最良解との差がほとんど見られず、初期解算出後に数秒の間に暫定解が複数算出されている。

5 結論と今後の課題

本研究では、実際の国内鉄道データをもとに Wedelin 解法を適用し、より大規模な問題においても短時間で精度の良い実行可能解算出に成功した。また、今後の課題として列生成型 Wedelin 解法への展開を検討する。

参考文献

[1] D. Wedelin, *Annals of OR*, Vol.57, pp.283-301, 1995.

[2] A.J.Mason, *Annals of OR*, Vol.108, pp.239-276, 2001.

[3] C.Barnhart and M.Cohn, *M & SOM*, Vol.6, pp.3-22, 2004.

[4] A.Caprara, M.Fischetti, and P.Toth, *Opns.Res.*, Vol.47, pp.730-743, 1999.