日本オペレーションズ・リサーチ学会
2023 年春季研究発表会

# 混合整数計画法による鉄道乗務割交番の自動作成手法

01111500 (公財) 鉄道総合技術研究所 *加藤 怜 KATO Satoshi
東日本旅客鉄道 (株) 西川 幸男 NISHIKAWA Yukio
東日本旅客鉄道 (株) 芝田 雄吾 SHIBATA Yugo
東日本旅客鉄道 (株) 小林 篤史 KOBAYASHI Atsushi

## 1. はじめに

鉄道の列車運行のためには，各列車の着発時刻を定めた列車ダイヤのみならず，運転士や車掌といった乗務員の運用計画を作成する必要がある [1]．現在でも，乗務員運用計画は熟練者による手作業をベースに作成されているが，技術継承の課題や将来的な労働力不足等を背景に，近年，自動作成のニーズが高まっている．

乗務員運用計画の作成は，乗務員の 1 勤務の行程を示す乗務行路（以下，単に「行路」）の作成と，その遂行順序を示す乗務割交番（以下，単に「交番」）の作成の 2 段階から構成される．本稿では，このうち後段の交番を対象とし，混合整数計画法（以下,「MIP」）に基づく自動作成手法を提案する．

## 2. 交番と問題定義

行路には，1 営業日中に終了する日勤行路と，2 営業日にわたる泊まり行路がある．交番は，各行路の実施順序を定め，必要箇所に休日や連休を挿入して作成するが，乗務員の各月の勤務表の元となるものである．

交番の例を図 1 に示す．上部の数値は日数を示し，下部に各日の勤務内容（時間は勤務時間帯）を示している．この例は 9 日間の交番であり，日勤行路は行路 1，2 の 2 つ，泊まり行路は行路 3，4 の 2 つであり，3 日目が休日，8，9 日目が連休となっている．この交番をもとに，各乗務員の各月の勤務が決められる．

交番の作成にあたっては，以下の条件を考慮する必要がある．

- 各行路間では必要在宅休養時間（ある行路の終了時刻から次の行路の開始時刻までの時間）を確保する（休日，連休を挟むかにより時間は異なる）
- 休日，連休前の退勤時刻は既定時刻以前とする
- 休日，連休後の出勤時刻は既定時刻以後とする
- 必要な休日日数を確保する（必要日数は所与）
- 休日，連休なしの連続勤務日数の下限，上限 (上限は 2 種類) を満たす

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 行路1<br>日勤<br>9:00～<br>17:00 | 行路2<br>日勤<br>8:30～<br>16:00 | 休日 | 行路3<br>泊まり<br>15:00～ | 行路3<br>泊まり<br>～10:30 | 行路4<br>泊まり<br>13:00～ | 行路4<br>泊まり<br>～9:00 | 休日 | 休日 |

図 1: 交番の例

評価指標としては，休日や連休時の在宅休養時間が長い方が望ましいことを踏まえ，以下の 2 つを考える．

(i). 休日，連休時の在宅休養時間の合計を最大化

(ii). 休日，連休時の必要在宅休養時間からの差分の最小値を最大化 (休日，連休が複数含まれるため，必要時間との差分が最も小さい休養時間を最大化)

## 3. MIP による交番作成手法

### 3.1. 巡回セールスマン問題によるモデル化

行路をノード，行路間の接続をアークとしたネットワークで表現することで，巡回セールスマン問題としてモデル化が可能である．なお，各ノード間のアークは，連続勤務となる通常アーク，休日を挟む休日アーク，連休を挟む連休アークの 3 つがある．ただし，必要在宅休養時間を確保できない場合にはアークを張らない．なお，アークのコストは在宅休養時間とする．

解となる巡回路上では，連続勤務日数を考慮する必要がある．そこで，Giacco et al.[2] の研究を参考に，制約を追加することで対応する．

### 3.2. 定式化

記号の定義および定式化を以下に示す．

**集合と定数**

$V$ ノード（行路）の集合
$E$ アークの集合
$R$ 休日アークの集合
$S$ 連休アークの集合
$c_{ij}$ アーク $(i,j)$ の経過日数
$d_{ij}$ アーク $(i,j)$ の休日・連休つなぎの在宅休養時間
$f_{ij}$ アーク $(i,j)$ 採択時に挿入される休日数
$g_{ij}$ 休日・連休アーク $(i,j)$ の必要在宅休養時間からの差分
$l$ 連続勤務日数の下限

$u_1$　休日なしの連続勤務日数の上限
$u_2$　連休なしの連続勤務日数の上限
$n$　交番全体での必要休日日数
$M$　十分に大きな数

**決定変数**

$x_{ij}$　アーク $(i,j)$ を採択する場合 1，さもなければ 0
$y_{ij}$　アーク $(i,j)$ の巡回路上での順序
$z_{ij}$　アーク $(i,j)$ の巡回路上での直近の休日からの経過日数
$\mu$　休日，連休時の必要在宅休養時間からの差分の最小値

$$\max. \sum_{(i,j)\in E} d_{ij}x_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{i\in V} x_{ij} = 1, \quad \forall j \in V \tag{2}$$

$$\sum_{j\in V} x_{ij} = 1, \quad \forall i \in V \tag{3}$$

$$\sum_{j\in V} y_{ij} = \sum_{h\in V} y_{hi} + 1, \quad \forall i \in V \setminus \{0\} \tag{4}$$

$$y_{ij} \le |V| x_{ij}, \quad \forall (i,j) \in E \tag{5}$$

$$\sum_{j\in V} y_{0j} = 1 \tag{6}$$

$$\sum_{j:(i,j)\in E} z_{ij} = \sum_{h:(h,i)\in E\setminus\{R\cup S\}} (c_{hi}x_{hi} + z_{hi}) + \sum_{h:(h,i)\in R\cup S} c_{hi}x_{hi}, \quad \forall i \in V \tag{7}$$

$$l x_{ij} \le z_{ij}, \quad \forall (i,j) \in R \cup S \tag{8}$$

$$u_1 x_{ij} \ge z_{ij}, \quad \forall (i,j) \in E \setminus S \tag{9}$$

$$u_2 x_{ij} \ge z_{ij}, \quad \forall (i,j) \in E \tag{10}$$

$$\sum_{(i,j)\in R\cup S} f_{ij}x_{ij} = n \tag{11}$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in E \tag{12}$$

$$y_{ij} \in Z^+, \quad \forall (i,j) \in E \tag{13}$$

$$z_{ij} \ge 0, \quad \forall (i,j) \in E \tag{14}$$

目的関数 (1) 式は合計の在宅休養時間の最大化，(2), (3) 式は各ノードを 1 度ずつ訪問する制約，(4)〜(6) 式は部分巡回路除去制約，(7)〜(10) 式は連続勤務日数の下限および上限制約，(11) は合計休日日数制約，(12)〜(14) 式は変数の取りうる値を示す．

休日，連休時の必要在宅休養時間からの差分の最小値を最大化するには，(1) 式を以下の式に置き換える．

$$\max. \quad \mu \tag{15}$$

表 1: 実験結果

| データ | 在宅休養時間 | 差分の最小値 | 計算時間 |
|---|---|---|---|
| A-(i) | 750 時間 18 分 | 43 分 | 44 |
| A-(ii) | 736 時間 35 分 | 6 時間 47 分 | 38 |
| B-(i) | 1,431 時間 0 分 | 7 分 | 1,011 |
| B-(ii) | 1,365 時間 55 分 | 8 時間 32 分 | 2,283 |

また，制約条件に以下の式を追加する．

$$\mu - g_{ij}x_{ij} + Mx_{ij} \le M, \quad \forall (i,j) \in E \tag{16}$$

## 4. 数値実験

上記で述べたアルゴリズムの性能を検証するため，実在規模の 2 つのインスタンス A，B を用いて数値実験を行う．A は仮想線区，B は実在線区のデータであり，問題規模として，ノード数はそれぞれ 22，44，アーク数はそれぞれ 1260，4988 である．実験には，Windows 10 Pro, Core i7-8700K, 64GB メモリの PC および数理最適化ソルバー Gurobi Optimizer 9.5.1 を用いる．

表 1 に，数値実験の結果として，各データ（インスタンスと 2 節で述べた評価指標の組合せで示す）の合計在宅休養時間，必要時間からの差分の最小値，計算時間（秒）を示す．評価指標 (i) と (ii) を比較すると，(i) では合計時間は大きくなるが，差分の最小値が小さいため，在宅休養時間のばらつきが大きいことがわかる．一方で，(ii) では差分の最小値が大きく，また合計在宅休養時間も (i) に近い値となっている．

また，計算時間は実用規模の B でも数十分程度であり，十分実用的といえるが，今回対象のインスタンスは中規模程度であるため，より大規模な実在線区での検証が必要と考えられる．

## 5. まとめと今後の展開

本稿では，鉄道乗務員の交番を対象とし，MIP による自動作成手法を提案した．実在規模の問題でも，数十分程度で実用的な交番を作成できることを確認した．今後は，様々な実在線区で検証を進めるとともに，より大規模な線区への適用も可能とする．

## 参考文献


[1] (財) 鉄道総合技術研究所運転システム研究室: 鉄道のスケジューリングアルゴリズム, エヌ・ティー・エス (2005)

[2] G. Giacco et al.: "Rolling stock rostering optimization under maintenance constraints", *Journal of Intelligent Transportation Systems*, Vol. 18, No. 1, pp. 95-105 (2014)