

鉄道車両運用におけるインタラクティブな運行計画作成手法

大槻 知史* 愛須 英之 安本 高典 金子 賢一 ((株)東芝)

横山 信行 (東芝ソリューション(株))

中島 正貴 島田 雅和 布施 徹 栗原 茂 鈴木 憲 (東京地下鉄(株))

An Interactive Planning Method in Railway Rolling Stock Allocation

Tomoshi Otsuki*, Hideyuki Aisu, Takanori Yasumoto, Kenichi Kaneko (Toshiba Corporation)

Nobuyuki Yokoyama (Toshiba Solutions Corporation)

Masayoshi Nakajima, Masakazu Shimada, Toru Fuse, Shigeru Kurihara, Ken Suzuki (Tokyo Metro.,Ltd)

Experts working for railway operators still have to devote much time and effort to creating plans for rolling stock allocation. In this paper, we propose an semiautomatic planning method for creating the rolling stock allocation plans. Our scheduler is able to deal with flexible constraint expression inputs interactively and to output falicitous failure messages. Owing to these useful features, the scheduler can provide results comparable to experts faster than before.

キーワード：スケジューリング, アルゴリズム, 鉄道, 車両運用

Keywords: scheduling, algorithm, railway, rolling stock allocation

1. はじめに

鉄道における計画業務には、よく知られたダイヤ作成の他にも、ダイヤ上に乗務員、車両をそれぞれ割当てる乗務員運用計画、車両運用計画や、車両の保守点検のための構内作業計画等がある。これらの計画業務は、現場の熟練者が脈々と蓄積し、受け継がれてきたノウハウを駆使して実施することが多いが、膨大な時間を要している。実用的なスケジューラがあれば、日常の業務効率を大きく改善できる可能性がある。

われわれは、この内、車両運用計画に焦点を当て、実問題に対する高速な求解が可能な解法を提案した⁽¹⁾。一方、スケジューラを実務に活用するためには、解法の高速度だけでなく、運用場面を想定したユーザビリティが重要である。実際、複雑かつ厳しい制約条件の下でユーザの希望にできるだけ適う計画表を作成するためには、インタラクティブに制約条件を変更しながら計画を更新することが不可欠と考えられる。

本稿ではまず、既提案解法が編成毎に判定可能な制約条件に基づいて構築されることに着目し、多様な制約条件を柔軟に変更可能な制約表現の方式について述べる。またインタラクティブな計画作成に不可欠な機能である計画失敗時の理由メッセージ出力を、既提案解法の特長を利用して実現する方式について述べる。最後に以上の方式を活用した運行計画エンジンが、鉄道事業者の実務に有効に活用できることを確認する。

以下、2節では車両運用計画の概要と関連研究について述べ、3節では既提案解法の概要と制約条件の表現方法に

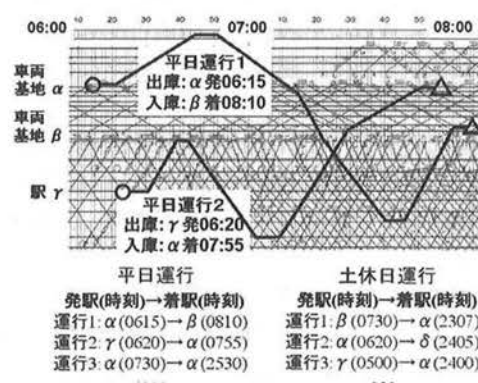


Fig. 1. An instance of routes on a diagram

について述べる。次に4節では、計画作成に失敗した場合のメッセージ出力方式について述べる。最後に5節では実適用例について述べる。

2. 鉄道車両運用における運行計画とは

鉄道車両運用計画は、個々の車両（以下、編成と呼ぶ）に対し、ダイヤ上のどの一連の走行経路（以下、運行と呼ぶ）を割当てるかの計画（以下、運行計画と呼ぶ）や、車両基地に入庫している間に各種の検査/清掃等の構内作業を割当てる計画（以下、検査/清掃計画と呼ぶ）において、時間、場所、作業人員等の全ての諸条件が成立する組合せを総合的に計画することである。本稿では、検査/清掃計画を所与とした場合の運行計画について述べる。

Fig.1に、縦軸に駅（車両基地も含む）、横軸に時刻を配したダイヤ上に、運行の例を太線で示した。平日運行1は、

編成 名称	04/01(日) 土休日	04/02(月) 平日	04/03(火) 平日
00	運 01.中 検査 八	45.行	運 18.運
01	26.運 24.八	26-2.運	17-2.運
04	運 09.行	行 37.運	運 13.八2
05	運 運日	運 05.運 清掃	運 05.運 検査
		17-2.運	14-2.中1
各日各編成の予定の見方			
06	出庫場所 第1運行		
07	←作業予定		
08	運 01.中 検査	第2運行 第3運行	
09	26.運 24.八		

Fig. 2. An instance of a plan table

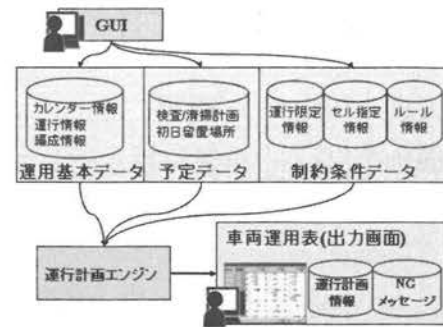


Fig. 3. A framework of the planning engine

車両基地 α を 6 時 15 分に出庫し所定のダイヤを走行した後に車両基地 β に 8 時 10 分に入庫する運行である。

ある編成には、1 日に 2 個以上の運行が割当てられる場合があり、逆に 1 個も運行が割当てられない場合もある。

運行計画は各編成に、平日または土休日ごとに定義された各運行を漏れなく重複なく割当てる形式をとり、Fig.2 のような、各行が編成、各列が日付を表す運用表に、検査/清掃計画や運行計画を記入して作成することが一般的である。

運行計画には、この他にも様々な実用上の制約がある。たとえば東京地下鉄株式会社 (以下、「東京地下鉄」と呼ぶ) の現業区には、以下に示すような制約 (要望) がある。

- 路線ごとに複数の車両基地と留置可能な複数の駅があり、編成は原則的に運行によってのみこれらの車両基地・駅間を移動する
 - ある編成に検査/清掃計画が設定された場合は、所定の時間帯に所定の車両基地に滞在しなければならない
 - 計画期間中に工場入場する編成があり、入場中は運行できない
 - 計画期間中に編成が登録又は廃車になる場合がある
 - 編成によって、走行できる運行が異なる場合がある
 - 検査や清掃を実施した編成は、翌日なるべく長く走行させたい
 - 遅く入庫した編成は、なるべく翌日早く出庫させたい
- なおこれらの要望はごく一部であり、要望の中には車両基地ごとに異なり、車両基地の事情により全く逆の要望となる場合もある。

国内の車両運用計画に関する研究は文献 (2) にまとめられている。最近の解法としては、数日単位の周期的な車両運用計画問題を巡回セールスマン問題に帰着する手法⁽³⁾や、二段階アプローチにより検査清掃と運行を同時計画する解法⁽⁴⁾等が知られている。

ただしこれらの文献では、解法の提案とその性能評価が中心的なトピックであり、制約条件をインタラクティブに変更する仕組みや、計画に失敗した場合に原因を検出しユーザーに指摘する仕組みにはほとんど言及していない。

3. 運行計画エンジンの構成

Fig.3 に、われわれが開発した運行計画作成を自動化する運行計画エンジンの構成を示す。GUI からは、運用基本

データの他、検査/清掃計画、初日留置場所などの予定データ、制約条件データを入力する。また出力画面では計画に成功した場合は作成した運行計画を表示し、失敗した場合は失敗理由に相当する NG メッセージを出力する。

〈3・1〉 運行計画エンジンの解法概要 ここでは、運行計画問題を U 個の運行 $u_1, u_2, \dots, u_U$ を、 H 台の編成 $h_1, h_2, \dots, h_H$ のいずれかに割当てる問題と捉える。なお、各運行 u には、出庫場所 $\text{DEP}(u)$ 、入庫場所 $\text{ARV}(u)$ 、出庫時刻 $\text{dept}(u)$ 、入庫時刻 $\text{arvt}(u)$ が定められているものとする。また各編成 h は、計画初日に最初に出庫すべき場所 (以下、初日留置場所と呼ぶ) に滞在しているが、その初日留置場所を s_h で表す。

いま N_h 個の運行列を $P_h = (p_{h,1}, p_{h,2}, \dots, p_{h,N_h})$ とするとき、各編成の連続する運行は場所と時刻が接続する必要があるため、以下の式 (1)、式 (2) を満たす必要がある。

$$\text{ARV}(p_{h,k-1}) = \text{DEP}(p_{h,k}) \quad (k = 1, 2, \dots, N_h) \quad (1)$$

$$\text{arvt}(p_{h,k-1}) < \text{dept}(p_{h,k}) \quad (k = 1, 2, \dots, N_h) \quad (2)$$

これをつなぎ条件と呼ぶ。ここで便宜上 $\text{ARV}(p_{h,0}) = s_h$ 、 $\text{arvt}(p_{h,0}) = -\infty$ とした。

またつなぎ条件以外の制約条件を満たすための評価指標として、各編成 h の制約条件違反を、必ず守るべき条件 (ハード制約) の違反数 (= $\text{Hard}(h)$) と、できるだけ守りたい条件 (ソフト制約) の違反数 (= $\text{Soft}(h)$) との 2 種類により数値化した。

このとき運行計画は、つなぎ条件と $\sum_h \text{Hard}(h) = 0$ の制約下で、 $\sum_h \text{Soft}(h)$ を最小化する最適化問題となる。

〈3・1・1〉 ネットワークによるモデル化 運行計画は Fig.4 のように、運行をノード、運行間の接続可能性をアークとするネットワークにより表現できる。ここで運行に相当する各ノードの左上には出庫場所を、右上には入庫場所を示す。また左端の h_1 等は編成の番号を示し、太線をたどる経路が各編成の運行計画を表す。なお便宜上、編成 h の最初には s_h を入出庫場所とする初日運行ノードを割当てることで、初日留置場所から出庫することを表現する。

たとえば、編成 h_1 の場合「運行 1 → 運行 4 → 運行 8 →」が編成 h_1 に対する運行計画を表す。

〈3・1・2〉 解法概要 われわれは、このネットワーク

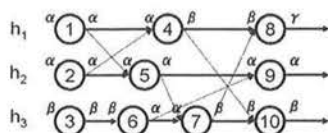


Fig. 4. An instance of a plan on a network

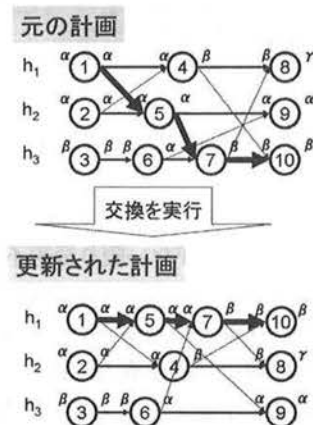


Fig. 5. A path and its corresponding updated plan

上の最適化問題の解法として、SMB(Swap-path Multiple Backtracking) アルゴリズムを提案した⁽¹⁾。

この解法では、たとえば編成 h_1 の初日運行ノードを基点とした経路に基づき、残りの編成の経路の入れ替えを行う。たとえば Fig.5 のように、 $1 \rightarrow 5 \rightarrow 7 \rightarrow 10$ の経路を考えると、 $1 \rightarrow 5$ および $5 \rightarrow 7$ が他編成との入れ替え処理(これを交換と呼ぶ)となる。この場合、最初に $1 \rightarrow 5$ にしたがって編成 h_1 のノード 4 以降と編成 h_2 のノード 5 以降の運行計画を入れ替え、次に $5 \rightarrow 7$ にしたがって編成 h_1 のノード 9 と編成 h_3 のノード 7 以降の運行計画を入れ替えることで Fig.5 下の更新された運行計画が得られる。

〈3・2〉 制約表現方法 われわれは、運行計画の様々な制約条件を以下の、運行限定条件、セル指定条件、ルール条件の 3 種類に整理して表現することとした。これらの条件はいずれも運用表作成結果を見ながら、別の画面からインタラクティブに条件変更が可能である。

運行限定条件 各構内作業に対し、作業実施編成の作業実施日に充当可能な運行を限定する条件である。たとえば「検査に対し、充当可能な運行が 05 運行、12 運行のいずれか」といった形で表現する。

セル指定条件 ある日ある編成に特定の運行を固定する条件である。たとえば、4/3 の編成 01 に 65 運行をできるだけ計画したい場合に、画面から条件設定可能とした。

ルール条件 予め用意されたルールのひな形にパラメータを設定し、登録する条件である。たとえば、「平日の運行 u_1 の後は、必ず運行 u_2 とする」というルールに対し、 u_1, u_2 に具体的な値を設定し、ルールを登録する。

4. 計画作成不可能な場合の理由の提示手法

運行計画エンジンの運用場面を想定すると、基本データ

Table 1. An instance of routes

運行番号	日	出庫時刻(場所)	入庫時刻(場所)
運行 1	4/1	5:00(深川)	9:30(深川)
運行 2	4/1	5:15(行徳)	23:00(中野)
運行 3	4/1	5:30(中野)	10:00(深川)
運行 4	4/1	16:00(深川)	23:15(行徳)
運行 5	4/1	16:30(深川)	23:30(深川)
運行 6	4/2	5:30(深川)	8:50(行徳)
運行 7	4/2	5:45(中野)	23:45(深川)
運行 8	4/2	6:00(行徳)	23:00(中野)

や制約条件の入力ミスにより計画作成不可能となる場合が想定される。また予め入力した条件の下では計画を作れず、検査/清掃計画やその他のハード条件を変更せざるを得ない場合がある。

一方計画不可能となった際に、もし理由が分からないとすると、どの制約条件を修正すべきか分からず、計画作成作業が停滞してしまう可能性が高い。

そこで、われわれは与えられた条件の下で運行計画の作成が可能かを判定し、不可能性を証明できた場合(以下、このような状態を NG と呼ぶ)や、NG 理由を推定できた場合は、適切なメッセージによりユーザに不可能であることを通知する方針を採る。

なお本節においては、01, 02, 03 の 3 編成に、4/1, 4/2 の二日間につき、Table 1 の 8 運行を割当てて例を用いて、代表的な NG 検出手法を説明する。

〈4・1〉 構内作業計画等により計画作成不可能となる場合の検出 Fig.6 に、構内作業計画等による計画作成不可能の検出フローを示す。

まず、運行限定条件やセル指定条件に基づき、 d 日目の編成 h に割り当て可能な運行 $\text{unkouList}(d, h)$ を全ての d と h の組に対し求めておく。次に、編成 h の初日運行ノードから $\text{unkouList}(d, h)$ に含まれるノードのみを辿った経路により、何日目まで到達できるか(D 日目とする)を探索する。

この結果、 D が最終日と一致する場合は NG を検出しない。一方 D が最終日前日以前の場合は、 $D+1$ 日目の編成 h に構内作業が設定された場合は、その構内作業が NG の原因であると推定できるため、構内作業により運行計画が作成できない旨のメッセージを出力する。その他の場合も、 $D+1$ 日目と h との組を出力しユーザに通知する。

Fig.7 に、2 日目の編成 02 に設定された清掃計画のために運行計画作成が不可能である例を示す。なお清掃実施日には、割当て可能な運行が運行 6 のみという運行限定条件が設定されているとする。

この場合 Fig.6 のフローに従うと、まず運行限定条件から、 $\text{unkouList}(1, 02) = \{6\}$ となる。次に、編成 02 の初日運行ノードから経路を辿ると、運行 6 を通る経路は存在しない。したがって、経路は 2 日目には到達できず、運行計画作成は不可能であることが分かる。

また 2 日目の編成 02 には、清掃が設定されているため、

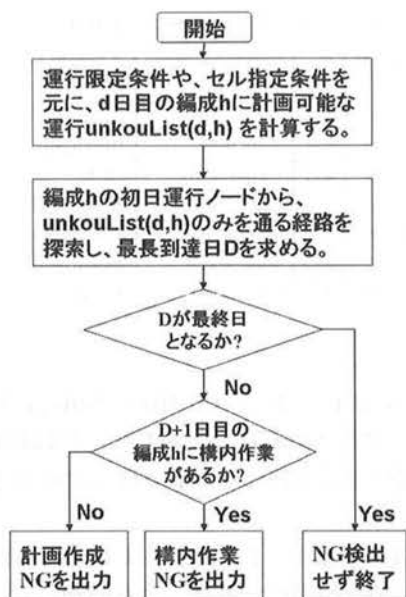


Fig. 6. A flow chart for judging infeasibility by the path search procedure

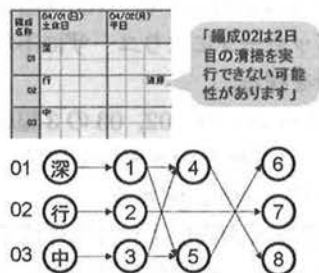


Fig. 7. An instance of detecting infeasibility from a cleaning plan

2日目の02編成の清掃を実行できない可能性がある旨のメッセージを出力できる。

これを見たユーザは、2日目の清掃予定を変更することにより、次の試行では計画作成に成功できる可能性が高まる。

〈4・2〉その他の場合 その他のケースとして、原因が特定できないが、計画作成に失敗するケースがあり得る。山登りベースの既提案解法では、この場合ハード制約の違反数最小の運行計画が最終的に得られる。

そこで制約条件の違反数最小の計画と、当該計画における制約違反箇所(日と編成の組)をメッセージとしてユーザに提示する。

たとえばこの制約違反箇所に検査/清掃計画が設定されている場合、ユーザは次の試行で当該計画を削除することにより、計画作成に成功する可能性が高い。

5. 運行計画エンジンを用いた実際の運行計画作成

われわれは運行計画エンジンを、実際の東京地下鉄の車両基地に導入し、1ヶ月分の運行計画作成の試験運用を行った。試験運用を行った車両基地では、従来1ヶ月分の車両運用表作成作業に10日程度要していた。

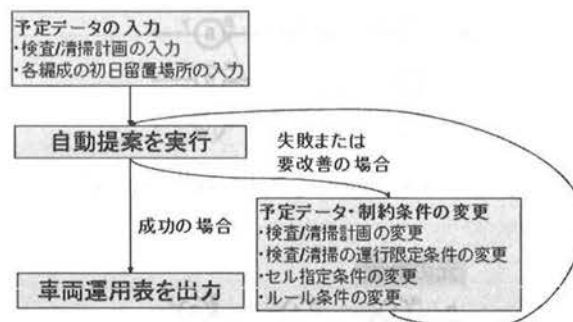


Fig. 8. A trial flow of planning

試験運用にあたって、平日/土休日の区分の情報であるカレンダー情報、ダイヤ改正がない限り変更不要な運行情報、編成の追加・削除がない限り変更不要な編成情報については、予め入力した。その上で Fig.8 のような手順で、車両基地の作業の方々に運行計画作成を実施してもらった。

具体的には、最初に予定データ(検査/清掃計画、初日留置場所)を入力し、自動提案により作成可能であることを確認しながら、インタラクティブに制約条件を調整・追加し、最終的に全ての制約条件を満足する計画を作成することとした。この条件を変更する過程で、誤った制約条件や強すぎるハード制約を入力した場合に、本稿で述べた NG 理由提示機能の効果により的確に NG 原因となる制約条件と原因となる日、編成の組を出力できることが分かった。

最終的な計画表出力までに要する時間は、制約条件の入力時間を含めても数時間程度であり、従来よりも十分短時間で車両運用表の作成が可能となることが分かった。また以上のプロセスにより得られた運行計画は、熟練者による軽微な手直しにより、実際に運用可能となることを確認した。

6. まとめ

本稿では、既提案解法の特性に着目し、インタラクティブに変更可能な制約表現方式と、計画に失敗した際の適切な NG 理由検出・提示方式をもつ、運行計画エンジンを提案した。また提案したエンジンは、東京地下鉄における試験運用を通して、熟練者に迫る計画を高速に作成できることを確認した。提案した手法は評価関数の表現力が高く、多くの鉄道事業者の実務への適用が期待できる。

文 献

- (1) 大槻知史, 愛須英之, 田中俊明: 車両運用計画に対する制約充足解法の提案, 日本オペレーションズ・リサーチ学会和文論文誌, Vol. 53, pp. 30-55(2010).
- (2) 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, 株式会社エヌ・ティー・エス (2005).
- (3) 福村直登, 中村達也, 西森進矢, 坂口隆: 鉄道の車両運用計画作成アルゴリズム, スケジューリングシンポジウム 2008 予稿集, pp. 167-172(2008).
- (4) 今泉淳, 山岸雄樹, 森戸晋: 二段階数理計画アプローチによる鉄道車両運用計画の策定, 日本オペレーションズ・リサーチ学会和文論文誌, Vol. 53, pp. 14-29(2010).