

# 利用者の不満を最小にする列車運転整理アルゴリズム

富井 規雄<sup>†</sup> 田代 善昭<sup>†</sup> 田部 典之<sup>††</sup>  
平井 力<sup>†</sup> 村木 国満<sup>†††</sup>

運転整理を支援するコンピュータシステムが実用化されるようになってきているが、それらは自動作成機能に欠けるために人間の負担はさほど軽減されていない。本研究では、従来とは異なって、運転整理案の評価尺度として、利用者の不満に着目することを提唱する。ここで、利用者の不満とは、列車の遅延、列車の頻度、接続等に対して、線区・ダイヤ・事故の規模に応じてあらかじめ定義しておくものである。そして、運転整理案の作成問題を、利用者の不満を最小にする組合せ最適化問題ととらえ、メタヒューリスティクスに基づく高度な自動作成機能を備えた運転整理案作成アルゴリズムを導入し、あわせて、実線区に対する本アルゴリズムの実験結果について紹介する。

## Train Rescheduling Algorithm which Minimizes Passengers' Dissatisfaction

NORIO TOMII,<sup>†</sup> YOSHIAKI TASHIRO,<sup>†</sup> NORIYUKI TANABE,<sup>††</sup>  
CHIKARA HIRAI<sup>†</sup> and KUNIMITSU MURAKI<sup>†††</sup>

Although computer systems which assist human experts in rescheduling disrupted train traffic is being practically employed recently, they are not so helpful in decreasing the workload of human experts. This is because they are lacking in intelligence such as to automatically make rescheduling plans. Unlike conventional works, we propose to use passengers' dissatisfaction as a criterion of rescheduling plans. We regard train traffic rescheduling as a combinatorial optimization problem in which dissatisfaction of passengers should be minimized and introduce an algorithm combining PERT and meta-heuristics. We also show some experimental results of the algorithm using actual train schedule data.

### 1. はじめに

鉄道にとって安定輸送は非常に大切であるが、ときとして、天災、事故、車両故障等によって列車ダイヤに乱れが生じることがある。このような場合、列車の運行を正常に戻すために、一連の列車ダイヤの変更が行われる。この業務を運転整理という<sup>1)</sup>。

近年、運転整理案の作成を支援するコンピュータシステム（運転整理支援システム）が実用に供されるようになってきた<sup>2)</sup>。しかし、それらのシステムは、列車順序の変更程度の単純なダイヤ変更提案機能を備えるのみで、それ以外の「運休」、「車両運用変更」、「番線変更」等のダイヤ変更を自動的に行う機能は備えて

いない。そのため、運転整理案を考案する作業は、指令員と呼ばれる専門家にまかされており、大きな負担となっているのが実情である。

この問題を解決するためには、運転整理支援システムに運転整理案の自動作成機能を付加して、機能の高度化、インテリジェント化を図る必要がある<sup>3)</sup>。

しかしながら、運転整理案を自動的に作成することは非常に難しい。2章で詳述するが、その主な理由としては、線区の事情や事故の規模等に応じて評価尺度がケースバイケースにならざるをえないこと、多数の列車を対象とする大規模で複雑な組合せ問題であること、迅速性が要求されること等があげられる<sup>1)</sup>。

本稿では、運転整理案作成問題を組合せ最適化問題ととらえ、計画ダイヤと事故の状況が与えられたときに、近似最適な運転整理案を迅速に作成するアルゴリズムの開発を目指す<sup>4),5)</sup>。そのため、次の2つの問題を解決する必要がある。

- (1) 運転整理案の評価尺度の確立
- (2) 近似最適な運転整理案を迅速に作成するアルゴ

<sup>†</sup> 財団法人鉄道総合技術研究所  
Railway Technical Research Institute

<sup>††</sup> 北海道旅客鉄道株式会社  
Hokkaido Railway Company

<sup>†††</sup> ニューメディア総研株式会社  
New Media Research Institute Co. Ltd.

## リズムの確立

本稿では、運転整理案の評価尺度として、利用者の不満を用いることを提唱する。そして、利用者の不満を最小にする運転整理案を自動的に作成するための PERT とシミュレートド・アニーリングを組み合わせたアルゴリズムを導入する。

ここで、利用者が不満と感じる状況は、あらかじめ洗い出して蓄積しておく（これをクレームファイルと呼ぶ）。利用者の不満としては、列車の発遅延、列車の着遅延、列車の駅外での停車（機外停止）による駅間の走行時分の伸び、列車の運休や遅延にともなう乗車チャンスの減少、列車間の接続がとられないこと等を考慮する。

利用者が何を不満と感じるかは、線区の性格や事故の規模等によって異なるため、状況に応じたクレームファイルをあらかじめ準備しておく。アルゴリズムの実行にあたっては、その状況に応じたクレームファイルを用いる。前述のように、運転整理の評価尺度は、線区の事情や事故の規模等によって異なるが、本研究では、線区や事故の規模等に応じたクレームファイルを個別に準備しておくことでその問題の解決を図る。

この考え方を採用することにより、従来の画一的な評価尺度の下では実現が困難であった、運休、車両運用変更、番線変更等を含んだ運転整理案が自動的に生成でき、また、接続や駅での待ち時間への考慮等も可能となる。これらは、既存の研究では対象とされていなかった事柄である。

アルゴリズムの全体構成は、シミュレートド・アニーリング (SA) に基づく。運転整理案の作成にあたっては、列車の各駅の着発時刻を正確に決定する必要がある（列車ダイヤは、15 秒、線区によっては 5 秒単位で作成されている）ため、それを SA で決定しようとする、探索空間が膨大なものとなる。本稿では、SA では列車の時刻を陽には考慮せず、列車の運休、編成の使用順序、列車の走行順序のみを決定することとし、その結果に基づいて、PERT を用いて、各列車の着発時刻を矛盾のないように定めるという方式を提案する。これによって、SA の探索空間を劇的に縮小することが可能となる。さらに、遅延等の利用者の不満は、クリティカルパスに着目して PERT ネットワークを変形することによって解消できるという性質を応用した効率的なアルゴリズムを提唱する。

本アルゴリズムを PC 上に実装し、現実の線区のダイヤを用いて、その評価を行った。その結果、実用可能な運転整理案を迅速に生成できることを確認した。

以下、2 章では、運転整理業務と現状の運転整理支

援システムの問題点について紹介する。3 章では、本稿で新たに提案する、利用者の不満に基づいた運転整理案の評価手法を導入する。そして、4 章では、利用者の不満を最小にする運転整理案作成アルゴリズムについて述べ、5 章では、実データを用いた実験結果とその評価結果について述べる。

## 2. 運転整理支援システムの現状

### 2.1 運転整理の難しさ

運転整理案を作成する際の具体的なダイヤ変更手段を表 1 に示す。ただし、これらは単独ではなく、事故の規模に応じて、適宜組み合わせで適用される<sup>1)</sup>。

運転整理は、非常に難しい業務である。その主な理由としては、次があげられる<sup>6)</sup>。

- (1) 運転整理案の最適性の評価尺度は、事故の規模や発生時間帯、線区の性格等によって異なる。すなわち、評価尺度を一律に決定することができない。たとえば、一般には、列車の遅延は好ましくないと評価されるが、山手線のような線区では、遅延を回復することよりも列車がある一定の頻度で運行されることの方が重視される。また、同じ線区であっても、朝の通勤時間帯と昼間時間帯とでは、重視する項目が異なることがある。通勤時間帯は、遅延の回復よりも列車の頻度を確保することが優先されるが、昼間時間帯では、夕方の通勤時間帯までに遅延を回復して正常運行にすることが重視される<sup>7)</sup>。このように運転整理案の評価尺度は、ケースバイケースに決定せざるをえないという難しさがある。
- (2) 多数の列車を対象とした大規模な組合せ問題であること。

大都市近郊の線区では、運転整理の対象となる列車は、数百本以上になることも珍しくない。これらの列車相互の関係と列車運行に関する制約を考慮しつ

表 1 運転整理案作成のためのダイヤ変更手段  
Table 1 Methods of rescheduling.

名称	内容
運休	列車の運転をとりやめる。
部分運休	列車の一部区間の運転をとりやめる。
臨時列車	臨時列車を運転する。
延長運転	列車の運転区間を延長する。
車両運用変更	車両の使用計画を変更する。
番線変更	列車の番線を変更する。
発順序変更	列車の出発順序を変更する（待避の設定、待避の解除）。
着順序変更	列車の到着順序を変更する（合流駅）。
停車種別変更	通過を停車に変更する。
列車種別変更	列車の種別を変更する（快速列車を普通列車に変更する等）。

つ、各列車の運転区間、運転時刻、番線等を決定していかなければならないという複雑で大規模な問題となる。事実、1時間程度の遅延が発生したときでも、ダイヤの変更件数は、数百件に及ぶこともある。

(3) 走行している列車を対象としているため、迅速性が要求される。

現に走行している列車群に対するダイヤの変更案を作成する業務であるため、迅速に運転整理案を作成する必要がある。

(4) 必要な情報が必ずしもすべて得られるとは限らない。

たとえば、列車がどれくらい混雑しているか/混雑する見込みであるか、あるいは、駅でどれくらいの利用者が列車を待っているか/今後、どの程度の利用者が出現するか等は運転整理の方針を決めるうえで非常に重要な情報であるが、このような情報を精度良く取得/予測することは、現時点では不可能に近い。

2.2 従来の運転整理支援システムとその問題点

運転整理は、きわめて難しい業務であるために、コンピュータによる支援が求められてきており、運転整理案作成支援システムに関する研究が進められてきた<sup>8)~10)</sup>。それらは、あらかじめ保持しておいた計画ダイヤをもとに、その時点までの列車の走行実績と事故の復旧予測時刻を用いて、運転整理案を作成する機能を有する。ただし、これらにおいて考慮されている運転整理手段としては、発順序変更(待避箇所の変更)程度であって、その他のダイヤ変更(表1参照)は、使用者が画面から入力する必要がある。

また、現実に用いられているシステムでは、発順序変更についても、コンピュータが自動的に行うのではなく、そのつど問いかけ画面を出力し、使用者の回答を求めるといったシステムも多い。

このような設計が採用されている理由は、2.1節の(1)~(4)の事情による。すなわち、運転整理は、判断基準が多岐にわたり、典型的な判断では対処できないために、コンピュータは定型的な処理のみを担当し、高度な判断は人間にまかせようとする考え方に基づくものである。

しかし、このような運転整理支援システムは、使用者である指令員の負荷の軽減のためには有用ではない。すなわち、

(1) 前述のように、発順序変更以外のダイヤの変更入力(表1)は、使用者がいちいち入力するため、入力の労力が大きい。場合によっては、入力が間に合わないこともある。

列車本数が多い線区の場合、ダイヤ変更の入力に要する時間は、無視できないものとなる。事実、運転整理案を考えるときには、ダイヤ変更の入力件数となるべく少なくなるような運転整理案を採用する等、本末転倒とでもいべき状態になっているのが実情である。

(2) 局所的な判断を積み重ねて運転整理案を得ようとするアルゴリズムであるため、全体として最適な運転整理案が得られない。

従来の運転整理案作成アルゴリズムは、シミュレーションに基づいている。すなわち、列車運行のルールと物理的制約をもとにして、今後の列車運行を予測し、その過程において、局所的な判断に基づいて列車の順序を決定していくという方式である。そのために、局所的には良い結果であっても、全体として見ると好ましくない運転整理案を生成する場合がある。その例を図1と図2に示す。図1は、列車3が駅Aで遅延した場合の列車運行(ダイヤの変更なし)を示している。なお、図1、図2は、列車ダイヤ図と呼ばれる形式で描かれている。横軸方向に時間、縦軸方向に駅をとり、時間の推移にともなう列車の位置の変化を直線(スジ)で表している。スジの横にある数字は、列車番号である。また、灰色のスジは、計画ダイヤ、黒色のスジは、実績または予測ダイヤを表している。

従来のアルゴリズムでは、図2のように、駅Bにおける列車1と列車3の順序を、「遅延した列車3(列車3')」を邪魔せずに、列車1が駅Cまで到達で

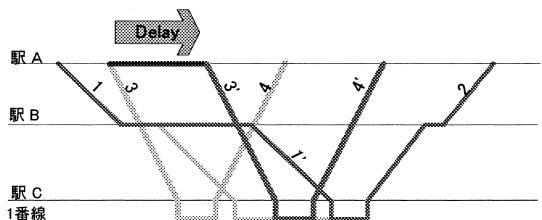


図1 順序を変更しない場合

Fig. 1 Rescheduling plan without modification.

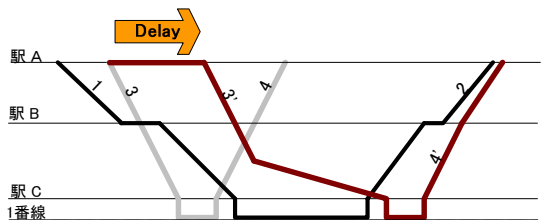


図2 局所的な判断による運転整理案

Fig. 2 Rescheduling plan by local decisions.

きる場合には、駅 B での列車の発順序を変更して、列車 1 を先発させる」といった局所的なルールのみで決めている。すなわち、その先で起こる状況を加味することはしていない。しかし、この例のように、駅 C において、列車 1～列車 2 と列車 3～列車 4 の番線が同じである場合には、駅 B での発順序を変更すると、列車 3 が駅 C の手前で立ち往生（機外停止）して到着が大幅に遅れ、折返しの列車 4 も遅延することになる。よって、このような場合、列車 3 の遅延や列車 1～列車 2 の折返し時間にもよるが、駅 B での順序を変更しない図 1 の方が望ましいこともありうる。従来のアルゴリズムでは、このように局所的な判断に基づいて、列車の順序を決定しているため、全体として最適な運転整理案を作成することについては問題がある（なお、図 1、図 2 は説明のための簡易な図である。実際には、駅 B と駅 C の間には多数の駅が存在するのが普通であること、また、2 本の列車だけでなく多数の列車が関連することが普通であること等の理由により、この判断はきわめて難しくなることに注意する必要がある）。これらの問題を解決し、指令員の大幅な負担軽減を図るためには、なるべく多種類のダイヤ変更手段をカバーする、高度な自動作成機能を持った運転整理案作成アルゴリズムが求められる。

### 3. 利用者の不満による運転整理案の評価

#### 3.1 運転整理の評価に関する従来の研究

運転整理案作成問題を組合せ最適化問題ととらえるためには、運転整理案に対する評価尺度を明確にする必要がある。運転整理案の評価尺度としては、これまで次のようなものが提案されてきた<sup>11)~16)</sup>。

- (1) 遅延時分による評価
- (2) 運休列車本数による評価
- (3) 遅延が収束するまでの時間による評価
- (4) 利用者側の損失・不効用による評価
- (5) 利用者が期待するサービスとの乖離による評価

しかし、運転整理が行われる状況は千差万別であること、必ずしもすべての情報が得られるとは限らないこと等を考慮すると、これらの評価は、いずれも満足のいくものではない。たとえば、遅延時分による評価（全列車の遅延時間の和、あるいは、各列車の遅延時間の重み付き線形和）は、運休との関連が問題となる。すなわち、多数の列車を運休すれば、当然、総遅延時分は小さくなる。極論すれば、遅延している列車すべてを運休とすれば、総遅延時分は 0 となる。しかし、このような運転整理が望まれることは、ありえない。

運休列車本数による評価（運休する列車の数が少ない方がよい）も問題がある。列車の運休は利用者に迷惑をかけるため、なるべく少ない方がよいと考えられる場合もあるが、一方では、必要な輸送力が確保できるのであれば適切な数の列車を運休することによって、混乱したダイヤを早期に正常に復する方が望ましい場合もある。

遅延が収束するまでの時間が短い方がよいという考え方は、これとは逆の問題をかかえている。多くの列車を運休すれば遅延は早期に収束するが、場合によっては、待ち時間や混雑の増加を招くことになり、利用者にとって良い運転整理にはならない可能性がある。

一方、利用者がこらむる不効用をもって、運転整理案を評価しようとする考え方もある<sup>14),15)</sup>。また、本来利用者が期待していたサービスとの乖離を用いて運転整理案を評価しようとする考え方が<sup>16)</sup>。しかし、これらの考え方に基づく評価を実施するためには、作成した運転整理案に対して、利用者の数や利用者の行動（今後どれくらいの数の利用者が、各駅にいつ出現し、どの列車を利用して、どこで乗り換えるのか等）をある程度正確に予測する必要がある。これは、現時点の技術では、非常に困難である。

#### 3.2 利用者の不満による運転整理案の評価

本研究では、現時点の技術においても適用可能な評価尺度を提案することを念頭において、利用者の不満に着目した運転整理案の評価尺度を提唱する。この考え方の背景は、次のとおりである。

- (1) 運転整理が行われるケースは千差万別であり、さまざまなケースに対して、遅延時分や運休本数等の画一的な評価尺度を使用することは適切でない。
- (2) 運転整理が行われる状況は、「異常時」であり、鉄道会社の視点からの評価尺度は、重要ではない。すなわち、評価尺度としては、利用者の視点が圧倒的に重要である。
- (3) 運転整理が行われる状況において、利用者の数や利用者の行動を予測することは容易ではないため、それらを用いた評価尺度を使用することには、現時点では限界があると考えられる。
- (4) 指令員が運転整理案を作成する際には、利用者になるべく迷惑をかけないことを念頭においており、「利用者の不満」という考え方は、指令員の直感によく合致する。

利用者がどういう状況を不満と感じるかは、事故の規模や線区の状況によって異なると考えられる。たとえば、20 分程度列車が停止した等の小規模の事故の場合には、10 分の到着遅れでも不満であろうが、数時

間不通になった場合等には、10分程度遅延することよりも列車の頻度の方に不満を感じると思われる。

本研究では、このような点に配慮し、線区や事故の規模等に応じて、利用者が不満と感じる状況をそれぞれ蓄積しておき（これを、クレームファイルと呼ぶ）、運転整理案を作成するときには、そのときの状況に合致したクレームファイルを選定した後、運転整理案を作成する。なお、クレームファイルは、指令員へのインタビュー等を通じて、あらかじめ作成しておくこととしている。

### 3.3 利用者の不満の定義

本研究では、利用者の不満として、次を考慮する。

- (1) 列車の遅延に関する不満
  - ある列車のある駅への着が、ある値以上遅延した。
  - ある列車のある駅からの発が、ある値以上遅延した。
- (2) 列車の停車時分の増加に関する不満
  - ある列車のある駅での停車時分が、所定の停車時分よりもある値以上増加した。
- (3) 列車の駅間の走行時分の増加に関する不満
  - ある列車のある駅間での走行時分が、所定の走行時分よりもある値以上増加した。
- (4) 列車の運転頻度に関する不満
  - ある駅である時間帯のある方向の停車列車の運転間隔が、ある値以上となった。
- (5) 列車の接続に関する不満
  - ある駅である時間帯のある方向のある2本の列車の間隔が、ある値以上となった。

それぞれの不満には、遅延の程度等を考慮した重みを付しておく。そして、作成された運転整理案に対して、これらの不満の件数をカウントし、それぞれの重みを乗じた線形和を、その運転整理案の評価値とする。

すなわち、本研究では、利用者の不満に関する条件を「制約」ととらえることによって、運転整理案作成問題を「制約違反を最小とするスケジュールを見つけることを目的とする重みつき制約充足問題」と考える。これにより、以下では、クレームファイルで定義されている利用者の不満に関する条件のことを「制約」、利用者が不満と感じる状況を「制約違反」と呼ぶことにする。

## 4. 利用者の不満を最小にする運転整理案作成アルゴリズム

### 4.1 全体構成

近年、組合せ最適化問題に対して、メタヒューリス

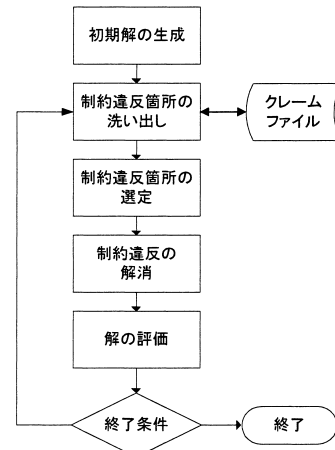


図3 アルゴリズムの全体構成

Fig.3 Configuration of the proposed algorithm.

ティクスと呼ばれる枠組みのアルゴリズムが注目を集めている<sup>17)</sup>。本稿で扱う問題と関連の深いスケジューリング問題にも多くの適用例がある<sup>18)</sup>。

メタヒューリスティクスの利点は、精度の高い解を高速に得られることにある。運転整理案作成問題も、きわめて高い迅速性を要求されることから、本研究においても、メタヒューリスティクスの1種であるシミュレーテッド・アニーリング<sup>19)</sup>を採用する。

アルゴリズムの全体構成を図3に示す。また、以下の節で各ステップの詳細を述べる。なお、図3における1回のループを「世代」と呼ぶことにする。

### 4.2 列車運行ネットワーク

列車ダイヤを、列車運行ネットワークと呼ぶ、ある種のPERTネットワークで表現する。既存研究<sup>10),20)</sup>では、PERTネットワークは、列車運行を高速に予測するために使用されているが、本研究ではそれを一歩進め、クリティカルパスを利用して制約違反の解消対象箇所を特定するためにも使用する。

列車運行ネットワークは、次のように構成される。

- ノード：列車の各駅の着・発をそれぞれ1つのノードとする。そのほかに仮想的なノードとして、開始ノードを設ける。
- アーク：ノード間の時間的依存関係をアークで表現する。アークには、両端のノードの事象が連続して発生するのに最低限必要な時間間隔を重みとして付す。

アークの種類とその意味、重みを表2に示す。なお、表2中の「基準運転時分」とは、列車が駅間を運転するのに最低限必要な時間、「最小停車時分」とは、乗降等のために駅に最低限停車しなければならない時

表 2 列車運行ネットワークのアークの種類

Table 2 Types of arcs in train scheduling networks.

名称	意味	設定方法	重み
列車	列車の走行	列車のある駅の発ノードと次駅の着ノードの間	基準運転時分
停車	列車の停車	列車のある駅の着ノードとその駅の発ノードの間	最小停車時分
車両運用	折返し	列車の終着駅の着ノードと折返し列車の発ノードの間	最小折返し時分
同一番線	番線使用順序	ある列車の発ノードと次にその番線を使用する列車の着ノードの間	時隔
発順序	発車順序	ある列車の発ノードと次にその駅から発する列車の発ノードの間	時隔
着順序	到着順序	ある列車の着ノードと次にその駅に到着する列車の着ノードの間	時隔
駅間許容	駅間に同時存在可能な列車数	ある列車の着ノードとその列車が到着後駅間に出発可能な列車の発ノード	時隔
交差支障	進路の競合	ある列車の着/発ノードとその事象と進路が競合する事象ノードとの間	時隔
計画時刻	ダイヤで定められた着・発時刻	開始ノードと各ノードとの間	計画時刻

間、「最小折返し時分」とは、折返し駅において最低限停車しなければならない時間のことをいう。

### 4.3 アルゴリズムの詳細

#### 4.3.1 初期解の生成

計画ダイヤに事故の状況（事故復旧予定時刻等）を入力し、運転整理を実施しないものとして予測を行った列車ダイヤ（単純予想ダイヤ）を初期解とする。

#### 4.3.2 制約違反箇所の洗い出し

クレームファイルとその時点の運転整理案を比較し、制約違反箇所をすべて洗い出す。

#### 4.3.3 解決すべき制約違反箇所の選定

解消を試みる制約違反箇所を1つ選定する。具体的には、4.3.2項で検出された制約違反箇所のうち、「列車の着遅延」、「発遅延」、「停車時分の増加」、「走行時分の増加」の制約違反の中から、1つをランダムに選ぶ。乱数は、発生時刻が早い制約違反箇所の方が高い確率で選ばれるように設定する。これは、発生時刻が早い制約違反を解消すると、その後に発生している他の制約違反もその影響を受けて同時に解消される可能性が高いと考えられるためである。ただし、発生時刻

が最早の制約違反箇所をかならず選定するようにすると、その制約違反がどうしても解消できないときに、プログラムがループに陥ってしまうため、発生時刻が遅い制約違反箇所も確率的に選定するようにしている。

なお、本アルゴリズムでは、「運転頻度」「接続」の制約違反は、直接、解消の対象とはしない。その理由は、「運転頻度」「接続」の制約違反は他の制約違反に起因して発生している場合が多いと考えられるためである。ただし、4.3.5項で述べるように、作成された運転整理案の評価においては、すべての種類の制約違反を考慮することとして、「運転頻度」「接続」の制約違反も少なくなる解が選定されるようにしている。

#### 4.3.4 制約違反の解消

##### (1) 制約違反解消アルゴリズム

4.3.3項で選定された制約違反を解消した運転整理案を作成することを試みる。このアルゴリズムは、「列車の遅延、停車時分・走行時分の増加に関する制約違反は、列車運行ネットワーク中で、その制約違反箇所の事象に対応するノードに至るクリティカルパスを構成するアークのいずれかに原因がある」という性質を用いている。手順を次に示す。

Step1: 4.3.3項で選定された制約違反箇所の事象に対応するノードを決定する。

- 制約違反の種類が列車の着遅延であるとき：当該列車の当該駅の着事象のノードを選定する。
- 制約違反の種類が列車の発遅延であるとき：当該列車の当該駅の発事象のノードを選定する。
- 制約違反の種類が停車時分の増加のとき：当該列車の当該駅の停車アークの終端ノード（当該駅の発事象のノード）を選定する。
- 制約違反の種類が走行時分の増加のとき：当該列車の当該駅間の列車アークの終端ノード（次駅の着事象のノード）を選定する。

Step2: Step1で選定されたノードから初期遅延箇所に対応するノードまで、クリティカルパスをアークと逆方向にたどっていく。クリティカルパスを構成するアークから、車両運用アーク、同一番線アーク、発順序アーク、着順序アークをすべて選び出す。さらに、そのうちの1つをランダムに選ぶ。そして、選ばれたアークの種類に応じて、表3に示すダイヤ変更を実施する。

##### (2) 車両運用アークに対する処理

車両運用アークに対する処理は、やや複雑であるので、図4、図5、図6を用いて、その内容を説明する。

図4は、計画ダイヤに対して、列車3が駅Aで遅延（列車3'）した場合を示している。このまま放置す

表 3 アークの種類に応じたダイヤ変更の実施

Table 3 Schedule modification methods reflecting arc types.

アークの種類	ダイヤ変更手段
発順序アーク	アークの両端の列車の発順序を、当該駅で入れ替える。
着順序アーク	アークの両端の列車の発順序を手前の駅で入れ替える。
同一番線アーク	アークのどちらかの端点の列車の番線を他の番線に変更する。
車両運用アーク	車両運用変更を実施する。
車両運用アーク	運休パターンファイルを参照して、アークの両端の列車を（部分）運休する。

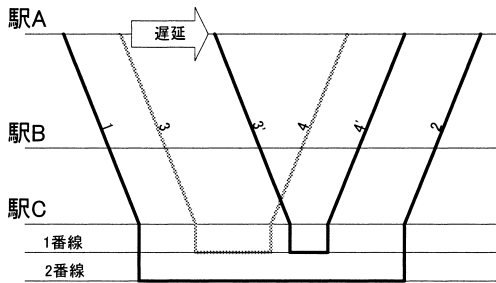


図 4 計画ダイヤ+遅延

Fig. 4 Original schedule + delay.

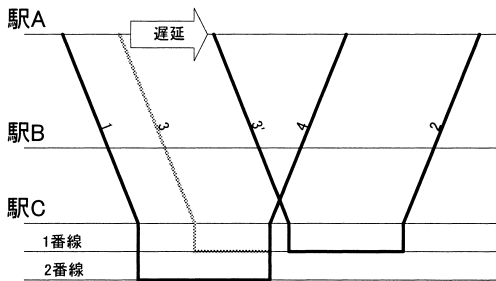


図 5 車両運用変更

Fig. 5 Change of vehicle utilization schedule.

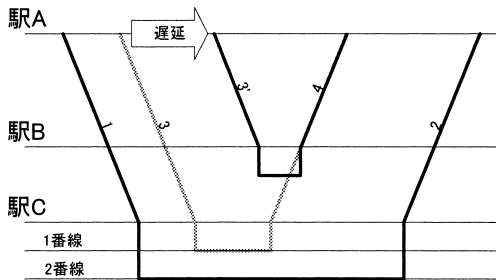


図 6 列車の運休

Fig. 6 Cancellation of trains.

ると、列車 3 の折返し列車である列車 4 も遅延する (列車 4')。しかし、ここで、図 5 のように、列車 4 に、駅 C に列車 1 として到着している編成を充当し、

列車 2 に列車 3 として到着する編成を充当することとすれば (このような変更を車両運用変更という)、列車 4 を遅延なしで運転することができる。

一方、図 6 は、図 4 に対して、列車の運休を実施した例である。列車 3、列車 4 をそれぞれ、駅 B ~ 駅 C 間を運休することによって、列車 4 を駅 B 以降、遅れなしで運転することができる。

本アルゴリズムでは、図 4 から作成された列車運行計画ネットワーク内のクリティカルパスから、列車 3 と列車 4 の間の車両運用アークを検出する。そして、車両運用変更が可能な列車が存在しないときには、運休を行い、存在する場合には、車両運用変更を実施するか運休を実施するかをランダムに決定する。なお、車両運用変更の可否を判定する際には、編成が物理的に存在するかどうかだけでなく、編成種別が一致しているかどうかのチェックも行う (たとえば、特急列車と普通列車の間の車両運用変更はできない)。

(3) 運休パターンファイル

運休に関しては、運休パターンファイルと称するデータをあらかじめ準備しておき、これを参照して運休区間を決定する。運休パターンファイルは、「駅 1、駅 2、番線」の形式で作成しておく。ここで、「駅 1」は、クリティカルパスから検出された車両運用アークに対応する駅 (図 6 の場合、駅 C)、「駅 2」は、駅 1 ~ 駅 2 までを運休として、列車を折り返させる駅 (図 6 の場合、駅 B)、「番線」は、駅 2 での折返し番線を指す。ここで、駅 2 が列車の始発駅と一致する場合には、全運休となり、列車の中間駅 (始発駅、終着駅以外の駅) と一致する場合には、部分運休となる。

なお、このように、運休についてのみ、あらかじめ運休パターンファイルを用意しておくのは、列車の運休は、輸送需要の段差や折返し番線の有無等の条件を加味して決定する必要があり、やみくもに (たとえばランダムに) 運休区間を決定する方式は、効率的でないと考えられるためである。

(4) 探索の進行に応じたダイヤ変更手段の制限

表 3 から分かるように、本アルゴリズムには、列車を運休する機能はあるが、アルゴリズム実行途中で一度実施した運休を取り消す機能はない。その意味で、本アルゴリズムは、完全 (complete) ではない。運休を取り消す機能を設けていない理由は、とりうるダイヤ変更手段の種類を多くすると、選択肢が増えるために探索空間が膨大となり、アルゴリズムの性能が落ちると考えられるためである。

しかし、一方では、この方式は次のような問題点を有することが想定される。

● 運休以外のダイヤ変更(たとえば、順序変更)を行えば解消できる制約違反を、運休によって解消した解を作成してしまう。そして、この解は運転頻度や接続に関する制約違反を有するために、評価値が低い。しかし、運休を取り消す機能がいないために、これらの制約違反は最後まで解消されない。本研究では、この問題に対処するために、制約違反を解消するときに用いるダイヤ変更手段を、探索の進行度合いに応じて制限する戦略をとる。具体的には、次の手順による。

- 探索開始後、一定世代が経過するまでは、ダイヤ変更手段として「順序変更」、「番線変更」のみを用いる。
- さらに、一定世代が経過するまでは、ダイヤ変更手段として、上記に加えて「車両運用変更」を用いる。
- さらに、一定世代が経過した後は、上記に加えて「運休」を用いる。

このように、探索の進行にともなって、用いるダイヤ変更手段に制限を加えるという方式を採用することにより、精度の高い解を効率良く求めることを目指している。

4.3.5 解の評価

検出された制約違反に対して、クレームファイル中であらかじめ定められている重みを乗じ、それらをすべて加算したものを、解の評価値とする。具体的には、次による。

$$\text{評価値} = \sum w_i \text{発遅延制約違反}(i) + \sum w_j \text{着遅延制約違反}(j) + \sum w_k \text{停車時分制約違反}(k) + \sum w_l \text{走行時分制約違反}(l) + \sum w_m \text{運転頻度制約違反}(m) + \sum w_n \text{接続制約違反}(n)$$

ここで、発遅延制約違反  $(i)$  は、 $i$  番目の発遅延制約違反、 $w_i$  は、それに対する重みを表す。他にも同様である。

4.3.6 新たな解の採否の決定

4.3.4 項の操作によって新たに得られた解  $S'$  の評価値を直前の解  $S$  の評価値と比較し、改良されていれば、 $S'$  を新たな解として採用して、そこから探索を続ける。改悪となっている場合には、改悪の程度と探索の進行状況に応じた確率で  $S'$  を採用するかどうかを決める。この確率は、一般的なシミュレイトド・アニーリングの考え方に基いて決定する。 $S'$  を採用しないと判定された場合には、 $S$  に戻って探索を再開する。

4.3.7 終了条件

その時点までに得られている最良解の評価値を観察

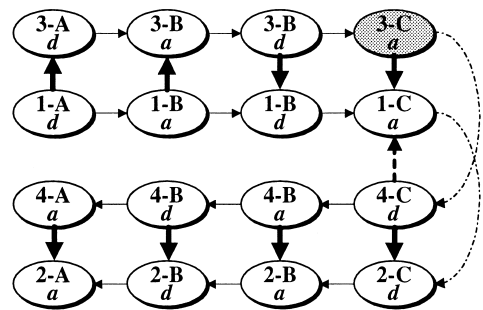
し、一定世代以上改善が見られない場合、その時点までの最良解を出力して終了する。

4.4 アルゴリズムの実行例

図1の例を用いて、本アルゴリズムの実行の経過を説明する。図7に図1のダイヤから構成された列車運行ネットワークを示す。ただし、図が煩雑になることを避けるために、開始ノード、計画時刻アーク、各アークの重みは省略している。また、ノードの中の記号は、列車名-駅名-着発区分を示している( $a$  は着、 $d$  は発を表す)。

今、遅延した列車1(列車1')の駅Cへの着が修復すべき制約違反であると決定されたとする。このとき、ノード1-C- $a$  から初期遅延箇所まで、クリティカルパスを逆にたどって、パス:1-C- $a$  (同一番線アーク) 4-C- $d$  (車両運用アーク) 3-C- $a$  (列車アーク) 3-B- $d$  (停車アーク) 3-B- $a$  (列車アーク) 3-A- $d$  を得る。このパスから、同一番線アークと車両運用アークを集め、いずれかをランダムに選ぶ。ここでは、同一番線アークが選ばれたとする。表3の手順に従い、駅Cにおける列車3~列車4または列車1~列車2の番線の変更を試みる。確率によって、列車3~列車4の番線が、番線2に変更されたとする(このときの状況を図8に示す)。

次に、列車1の駅Bの発遅延が解消すべき不満箇所であると決定されたとする。このとき、列車1の駅Bの発ノード(1-B- $d$ ) からクリティカルパスを逆にたどり、1-B- $d$  (発順序アーク) 3-B- $d$  (停車アーク) 3-B- $a$  (列車アーク) 3-A- $d$  というパスを得る。これから、発順序アークを選び、表3に従って、駅Bでの列車1と列車3の発順序を入れ替える。このようにして、図9の運転整理案を得る。



→ 列車アーク・停車アーク    → 発順序・着順序アーク  
 --- 同一番線アーク    ····· 車両運用アーク

図7 図1のダイヤに対応する列車運行ネットワーク  
 Fig. 7 Train schedule network constructed from Fig. 1.



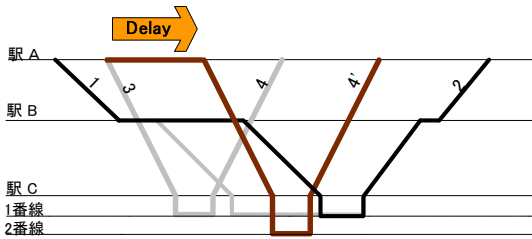


図 8 列車 3～列車 4 の番線を変更  
Fig. 8 Change of the track of Train3 ~ Train4.

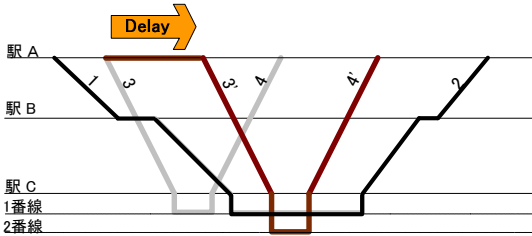


図 9 駅 B で列車 1 と列車 3 の順序を変更  
Fig. 9 Exchange the order of Train1 and Train3.

## 5. 実験結果とその評価

### 5.1 実験結果

本稿で導入したアルゴリズムを PC に実装し、実際のダイヤデータを用いて、その性能を検証した。

#### (1) 用いたデータ

対象とした線区は、首都圏近郊の約 40 km の路線で、19 の駅が存在し、特急や普通列車が数分間隔で運転されている。また、途中には、他の線区との分岐駅となっている駅があり、一部の列車は直通運転を行っている。実験では、この線区の 24 時間分の列車ダイヤ(列車本数は総計 564 本)を用いた。

#### (2) クレームファイルの構成

クレームファイルは次のように構成した。

- 主要駅(計 9 駅)での上下方向の発遅延が 2 分以下でなければならない。
  - 各駅間の走行時分の増加が 1 分以下でなければならない。
  - 主要駅(計 9 駅)での停車時分の増加が 4 分以下でなければならない。
  - 主要 4 駅における列車頻度に関する制約。
  - 快速待避駅(2 駅)における快速と普通列車の接続時間に関する制約。
  - 分岐駅における乗入れ列車との接続時間の制約。
- 遅延, 走行時分, 停車時分の制約の重みは、それぞれ 1 とし、列車頻度, 列車の接続の制約の重みは、それぞれ 50 とした。なお、クレームファイル中のレコード件数は、計 265 件であった。この実験では、「遅延」

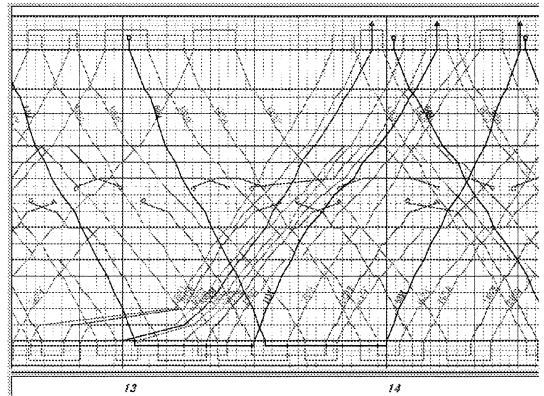


図 10 運転整理なし(実験 1 ケース 1)  
Fig. 10 Without rescheduling (Exp. 1 Case 2).

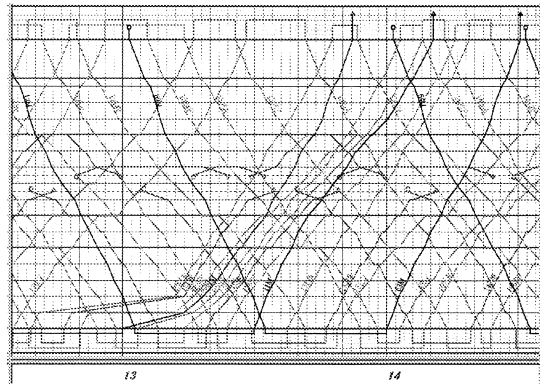


図 11 実験結果(実験 1 ケース 1)  
Fig. 11 A rescheduled plan (Exp. 1 Case 2).

に関する制約は 1 種類(2 分以下)のみとし、遅延の度合いに応じた重みを付した制約を複数準備することはしていない。その理由は、本クレームファイルは、初期遅延が比較的小さいケースに適用することを念頭において作成したものであり、その場合、大きく遅延する列車はほとんど存在しないため、遅延時分の多少の差を反映した制約を準備することには、さほど意味がないと考えられるためである。走行時分, 停車時分の制約についても同様である。

(3) シミュレートド・アニーリングのパラメータ  
初期温度は、20 度、減衰率は、0.97 とし、同一温度での繰返し回数は 10 回とした。

#### (4) 終了条件

実験のために、4.3.7 項で述べた終了条件を無視し、つねに 400 世代まで実行することとした。

#### (5) 実行世代によるダイヤ変更手段の制限

次のとおりとした。  
100 世代まで：順序変更, 番線変更

表 4 実験 1 のケース 1 の結果  
Table 4 Results of Case 1 of Exp. 1.

試行	最良解の評価値	最良解が得られた世代
1	346	361
2	344	363
3	348	271
4	349	230
5	349	289
6	351	132
7	345	306
8	344	282
9	348	312
10	349	239

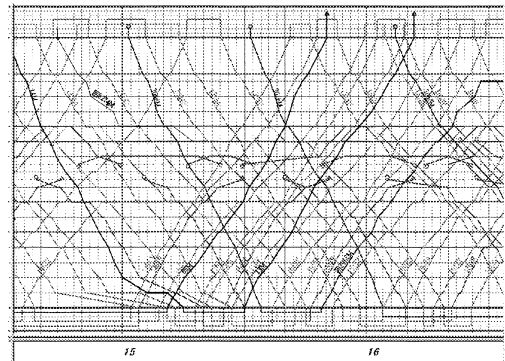


図 13 運転整理なし (実験 1 ケース 2)  
Fig. 13 Without rescheduling (Exp. 1 Case 2).

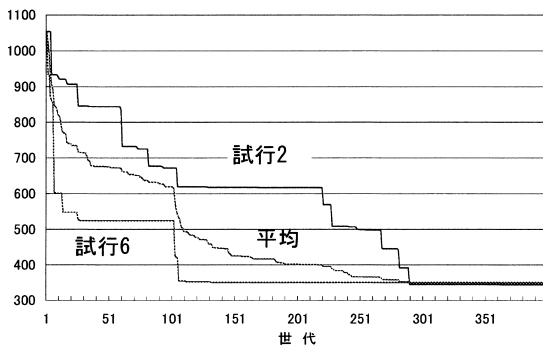


図 12 解探索の推移 (実験 1 ケース 1)  
Fig. 12 Search process (Exp. 1 Case 1).

表 5 実験 1 のケース 2 の結果  
Table 5 Results of Case 2 of Exp. 1.

試行	最良解の評価値	最良解が得られた世代
1	160	288
2	154	219
3	153	219
4	153	213
5	153	207
6	153	234
7	153	224
8	153	215
9	153	225
10	153	225

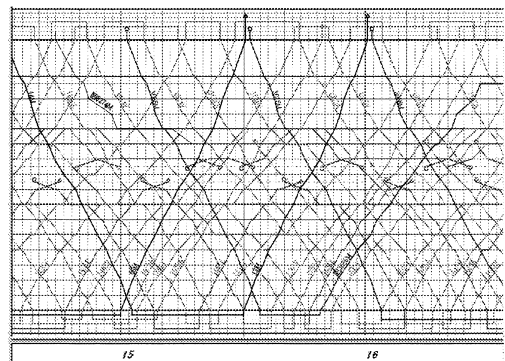


図 14 実験結果 (実験 1 ケース 2)  
Fig. 14 A rescheduled plan (Exp. 1 Case 2).

200 世代まで：順序変更，番線変更，車両運用変更  
201 世代以降：制限なし

(6) 実験内容

・実験 1：性能評価のための実験

次の 2 つのケースに対して，それぞれ 10 回ずつアルゴリズムを試行した。

ケース 1：駅間で列車が約 30 分停止した例（踏切事故を想定）

ケース 2：始発駅で 1 本の列車が約 20 分遅延した例（車両故障を想定）

・実験 2：探索戦略評価のための実験

4.3.4 項 (4) で述べた，用いるダイヤ変更手段を世

代に応じて制限するという戦略の効果を評価するために，実験 1 のケース 1 と同じ乱れのケースに対して，用いるダイヤ変更手段を制限しない場合の実験を行った。

(7) 実験 1 の結果

ケース 1：実験 1 のケース 1 において，運転整理をしない場合の列車ダイヤと本アルゴリズムで生成された運転整理案の 1 つをそれぞれ図 10，図 11 に示す（図 10，図 11 には，ともに約 2 時間分のダイヤを示している）。また，10 回の試行の結果を表 4 に示す。あわせて，解探索の推移を図 12 に示す。図 12 は，横軸方向を探索の進行過程とし，その世代までに得られている最良解の評価値を示したグラフである。ただし，煩雑になることを避けるために，最も早く収束した試行 6，最も収束が遅かった試行 2，および 10 回の試行の平均のみを示す。

ケース 2：実験 1 のケース 2 の結果を表 5 に示す。また，運転整理をしない場合の列車ダイヤと本アルゴリズムで生成された運転整理案の 1 つをそれぞれ図 13，図 14 に示す。解の探索の推移は，ケース 1 と

表 6 実験 2 の結果  
Table 6 Results Exp. 2.

試行	最良解の評価値	最良解が得られた世代
1	479	342
2	392	369
3	480	190
4	441	353
5	345	388
6	347	163
7	354	298
8	398	9
9	472	9
10	586	182

同様であるので省略する。

#### (8) 実験 2 の結果

実験 1 のケース 1 と同じ乱れのケースに対して、用いるダイヤ変更手段を世代によって制限しない場合の結果を表 6 に示す。

#### 5.2 評価と考察

##### (1) 実験 1 について

実験 1 については、ケース 1、ケース 2 のすべての試行において、実用上問題のない運転整理案を生成している。ケース 1 は、駅間で列車が停止しているため後続の列車がすべて遅延することになる難しいケースであるが、図 11 のダイヤ図を図 10 のダイヤ図と比較すると、列車の順序の変更や車両運用変更を適切に行うことによって、特に事故発生直後の列車の頻度を確保した運転整理案を生成していることが分かる。なお、図 11 において実施されたダイヤの変更は、車両運用変更 4 件、発順序変更 13 件、着順序変更 8 件、番線変更 13 件の計 38 件であった。

ケース 2 は、始発駅で 1 本の列車のみが遅延して長時間停止したケースである。この場合、後続の列車を先に発車させるために列車の順序を適切に変更することや、機外停止を防止するために番線を適切に変更する等の方策が必要になる。図 13、図 14 を比較すると、図 14 では、適切な数の列車を運休し、端末駅での番線と途中駅での列車順序を適切に変更することによって、特に遅延発生直後の列車頻度を確保するとともに、機外停止を防止した運転整理案の作成に成功していることが分かる。図 14 において実施されたダイヤの変更は、運休 2 件、車両運用変更 1 件、発順序変更 11 件、着順序変更 4 件、番線変更 12 件の計 30 件であった。

これらの運転整理案を、当該線区の指令員に示し、十分実用的であるという確認を得ている。

また、表 4、表 5 からは、10 回の試行において、最終的に得られる解の評価値は各ケースともほぼ一致し

ていることが分かる。図 12 からは収束の過程に若干の差はあるものの、安定して収束に向かっていくことが見てとれる。これらのことから、本アルゴリズムは、安定して良好な解を生成可能であると判断してよいと考えられる。

実行時間は、解が収束する 300 世代までで、約 1 分 (PentiumIV 3.06 GHz) であり、十分実用的な値であった。

##### (2) 実験 2 について

表 4 と表 6 を比較すると、表 6 では、得られた解の質にばらつきがあること、探索の初期で局所最適解に陥っていると思われる場合があることが分かる。実験 2 で作成された運転整理案の内容を観察したところ、待避を解除すれば制約違反を解消できる列車を運休にしてしまう、車両運用変更を行えば制約違反を解消できる列車を運休にしてしまう等の現象が見られた。一方、実験 1 で作成された運転整理案ではそのような現象は観察されなかった。この結果は、4.3.4 項 (4) で導入した、用いるダイヤ変更手段を世代に応じて制限するという戦略が良好に動作していることを示していると考えられる。

## 6. おわりに

運転整理案の評価尺度として、利用者の不満に着目する考え方を新たに提唱し、その考え方に基づく運転整理案作成アルゴリズムを導入した。本アルゴリズムは、PERT とシミュレーテッド・アニーリングを組み合わせたもので、これまでの研究では実現されていなかった、列車の運休や、車両運用変更、番線変更を自動的に実施する機能を持つほか、そのときの状況に応じた運転整理案を迅速に作成する機能を有する。また、運休パターンファイルとクレームファイルを除いては、線区やダイヤ固有の条件を使っていないため、多くの線区に汎用的に適用可能であるという特徴を有する。

本アルゴリズムを PC 上に実装し、実際のダイヤデータを用いた実験を行うことにより、その有効性を実証した。

なお、本稿で提案したアルゴリズムは、表 1 中のすべてのダイヤ変更を自動的に行うわけではない。臨時列車の運転、延長運転、停車種別変更、列車種別変更等は、利用者の入力にまかされている。その理由は、これらのダイヤ変更の実施には、それらの列車を運転する運転士・車掌の手配が必要となる、そのときの旅客流動の状況等に基づく高度な判断が必要となる等による。これらを自動的に実施するアルゴリズムの開発については、今後の課題としたい。

謝辞 本研究を実施するにあたり、貴重なご意見をいただいた、JR 東日本先端鉄道システム開発センター 辺田文彦、佐藤伊佐美の両氏、ならびに、JR 東日本千葉支社の関係者の方々に深く感謝します。

### 参考文献

- 1) 富井規雄 (編): 鉄道システムへのいざない, 共立出版 (2001).
- 2) 特集「運行管理」, JR ガゼット, Vol.57, No.3 (1999).
- 3) 中村達也, 井原恭平: 運行管理システムの現状と課題, 特集「鉄道における情報処理技術の現状と今後」, 電気学会誌, Vol.124, No.5 (2004).
- 4) 富井規雄, 田代善昭, 田部典之, 平井 力, 村木国満: 利用者の不満を最小にする列車運転整理アルゴリズム, 情報処理学会研究報告 知能と複雑系 2003-ICS-121 (7) (2003).
- 5) 富井規雄, 田代善昭, 田部典之, 平井 力, 村木国満: 利用者の不満を最小にする列車運転整理案作成アルゴリズム, 情報処理学会第 46 回数理モデル化と問題解決研究発表会, 2003-MPS-46 (11) (2003).
- 6) 富井規雄: 輸送計画・運行管理システムの高度化, 電気学会誌, Vol.124, No.5 (2004).
- 7) 富井規雄: ダイヤの乱れを克服する—鉄道の運行管理システムの現状と今後, 情報処理, Vol.44, No.8 (2003).
- 8) 駒谷喜代俊, 匹田志郎, 安部恵介: 知識工学的アプローチによる列車運転整理支援システムの開発, 電気学会論文誌, Vol.107-C, No.2 (1987).
- 9) 荒木 大ほか: マンマシン協調モデルに基づく運転整理エキスパートシステム, 電気学会論文誌, Vol.115-C, No.5 (1995).
- 10) 富井規雄, 池田 宏: ハイブリッド型列車運転整理シミュレータ, 電気学会産業システム情報化研究会 IIS-96-14 (1996).
- 11) 長谷川豊, 近谷英昭, 篠原専二: 運転整理評価基準の実験的検討, 第 15 回鉄道におけるサイバネティクス利用国内シンポジウム論文集 (1978).
- 12) 長井忠昌, 海老原浩一, 東海林保, 中田裕友: 運転整理の評価, 第 14 回鉄道におけるサイバネティクス利用国内シンポジウム論文集 (1977).
- 13) 岡本正巳, 矢部雄次: シミュレーションによる運転整理手法の評価, 第 15 回鉄道におけるサイバネティクス利用国内シンポジウム論文集 (1978).
- 14) 小林里紗: 利用者の利便性から見た非常時の運転整理ダイヤの評価, 東京大学工学部土木工学科卒業論文 (2000).
- 15) 長崎祐作, 古関隆章: 都市近郊鉄道における運転整理案の作成と評価, 電気学会交通・電気鉄道研究会, TER-02-63 (2002).
- 16) 村田 悟, Goodman, C.: 乗客の満足度を考慮した列車群制御方式, 第 34 回鉄道におけるサイバネティクス利用国内シンポジウム論文集 (1997).
- 17) 柳浦睦憲, 茨木俊秀: メタヒューリスティクスの枠組, 電気学会論文誌, Vol.121-C, No.6 (2001).
- 18) Anderson, E.J., et al.: Machine Scheduling, Aarts and Lenstra (Eds.), *Local Search in Combinatorial Optimization*, John Wiley & Sons Inc. (1997).
- 19) Aarts, E. and Korst, J.: *Simulated Annealing and Boltzman Machines*, John Wiley & Sons Inc. (1989).
- 20) 安部恵介, 荒屋真二: 最長径路法を用いた列車運行シミュレーション, 情報処理学会論文誌, Vol.27, No.1 (1986).

(平成 15 年 8 月 21 日受付)

(平成 15 年 10 月 10 日再受付)

(平成 15 年 11 月 1 日採録)



富井 規雄 (正会員)

1978 年京都大学大学院工学研究科情報工学専攻修士課程修了。同年日本国有鉄道。1987 年 (財) 鉄道総合技術研究所。現在, 同所輸送情報技術研究部運転システム研究室長。鉄道のスケジューリングアルゴリズムの研究に従事。電気通信大学大学院客員助教授。京都大学博士 (情報学)。人工知能学会, 電気学会, 日本 OR 学会各会員。



田代 善昭

1990 年九州工業大学大学院工学研究科設計生産工学専攻修士課程修了。同年九州旅客鉄道株式会社。2002 年より (財) 鉄道総合技術研究所に外向。鉄道のシミュレーションモデル, スケジューリングアルゴリズムの研究開発に従事。現在, 同所輸送情報技術研究部運転システム研究室副主任研究員。



田部 典之

1996 年東京都立科学技術大学大学院工学研究科電子情報系システム工学専攻修士課程修了。同年北海道旅客鉄道株式会社入社。2000 ~ 2003 年 (財) 鉄道総合技術研究所に外向。外向中, 運転整理アルゴリズムの研究に従事。2004 年まで, 北海道旅客鉄道株式会社勤務。



平井 力

1996年東京工業大学大学院情報理工学研究科数理・計算科学専攻修士課程修了。同年(財)鉄道総合技術研究所。現在、同所輸送情報技術研究部運転システム研究室副主任研究員。

鉄道のシミュレーションモデル、スケジューリングアルゴリズムの研究開発に従事。日本OR学会会員。

---



村木 国満(正会員)

2000年豊橋技術科学大学大学院工学研究科知識情報工学専攻修了。同年(株)ニューメディア総研。運転整理アルゴリズムの研究に従事。