

早稲田大学審査学位論文（博士）

鉄道輸送サービスデペンダビリティ向上
のための列車運用整理リアクティブ
スケジューリング技術の研究

佐藤 達広

早稲田大学大学院情報生産システム研究科

2012年11月

目次

第1章	序論	1
1.1	研究の背景と目的	1
1.2	本論文の構成	2
第2章	関連研究および関連技術	5
2.1	鉄道分野におけるスケジューリング技術の研究開発動向	5
2.1.1	列車ダイヤ向けリアクティブスケジューリング	7
2.1.2	資源スケジューリング	9
2.2	リアクティブスケジューリング技術に関する研究開発動向	11
第3章	列車運用整理業務におけるリアクティブスケジューリング	15
3.1	列車運行乱れ時の輸送サービス回復と列車運用整理業務	15
3.2	輸送サービス回復のシステム化の現状	17
3.3	リアクティブスケジューリングの課題	18
第4章	本線運用整理のためのリアクティブスケジューリング方式	21
4.1	研究の背景	21
4.2	問題定義	23
4.2.1	ネットワークフローモデルによる本線運用整理のモデル記述	23
4.2.2	本線運用整理リアクティブスケジューリング問題の定式化	28
4.2.3	定式化の特徴	33
4.3	運用計画変更コストに着目した最適スケジューリング方式	36
4.3.1	基本アイデア	37
4.3.2	ラグランジュ緩和問題	38
4.3.3	ラグランジュ緩和法	40
4.3.4	下界値算出方式	41

4.3.5	実行可能解生成方式	42
4.3.6	提案方式の特徴	44
4.4	運用資源制約と運用計画変更コストに着目した最適スケジューリング方式	44
4.4.1	基本アイデア	45
4.4.2	初期解生成法	47
4.4.3	局所探索法	49
4.5	数値実験	49
4.5.1	プロトタイプシステム	49
4.5.2	実験条件	51
4.5.3	実験結果：スケジューリング方式1	57
4.5.4	実験結果：スケジューリング方式2	60
4.6	考察	63
4.6.1	再作成の効率	63
4.6.2	精度評価	66
4.6.3	デペンダビリティのさらなる向上	67
4.7	本章の総括	70
第5章	車両基地運用整理のためのリアクティブスケジューリング方式	71
5.1	研究の背景	71
5.2	問題定義	72
5.2.1	FJSSPモデルによる車両基地運用整理のモデル記述	72
5.2.2	車両基地運用整理リアクティブスケジューリング問題の定式化	76
5.3	基地構内レイアウト条件を考慮した最適スケジューリング方式	81
5.3.1	基本アイデア	81
5.3.2	部分スケジュール作成処理	83
5.4	数値実験	90
5.4.1	プロトタイプシステム	90
5.4.2	実験条件	91
5.4.3	実験結果：白紙からのスケジューリング	95
5.4.4	実験結果：再スケジューリング	96
5.5	考察	106

5.5.1	再作成の効率	106
5.5.2	評価指標の多様化	106
5.5.3	ジョブ定義の変更	107
5.6	本章の総括	108
第6章	結論と今後の課題	109
6.1	結論	109
6.2	今後の課題	111
6.3	おわりに	116
	謝辞	117
	参考文献	119
	研究業績一覧	131

表目次

4.1	ネットワークモデルの規模	53
4.2	2つの提案方式の比較	67
5.1	各車両基地の番線数と番線種類	93
5.2	白紙からのスケジュール作成の処理時間(秒)	97
5.3	実験結果(再スケジュールの成功回数と比率)	100
5.4	実験結果:全車両基地	100
5.5	実験結果:車両基地A(両サイドオープン型)	102
5.6	実験結果:車両基地B(片サイドオープン型)	103
5.7	実験結果:車両基地C(複合型)	104

目次

2.1	輸送計画の基本構造	6
3.1	列車運行乱れ時の輸送サービス回復業務	16
4.1	ネットワークフローモデル(トリップフローモデル)	24
4.2	スケジュールの部分交換	33
4.3	実行可能解生成手続き	43
4.4	未割当ノードの追加	43
4.5	生成検査法に基づくスケジュール再作成の実行イメージ	46
4.6	プロトタイプシステム	50
4.7	実験データ1:小規模鉄道路線	52
4.8	実験データ2:大規模鉄道路線	53
4.9	データ1(小規模鉄道路線)に対する実験結果	58
4.10	データ2(大規模鉄道路線)に対する実験結果	59
4.11	最短経路探索の実行時間の比較	60
4.12	最良解の目的関数値(小規模鉄道路線)	61
4.13	精度評価結果(小規模鉄道路線)	62
4.14	最良解の目的関数値(大規模鉄道路線)	63
4.15	精度評価結果(大規模鉄道路線)	64
4.16	提案手法を組み合わせたスケジュール再作成モデル	68
5.1	構内入換計画の例	73
5.2	ジョブ定義の例	74
5.3	段階的スケジューリングアーキテクチャ	82
5.4	資源割当(Step1)の処理フロー	85
5.5	ヒューリスティクス	87

5.6	時間シフト量の計算	90
5.7	プロトタイプシステム	91
5.8	実験対象の車両基地	92
5.9	白紙からのスケジュール作成結果（両サイドオープン）	96
5.10	運用乱れケースの例（入区遅れ）	98
5.11	再スケジュールリング結果の変更量の分布	105

第1章

序論

1.1 研究の背景と目的

近年、環境意識の高まりや都市の過密化、資源高騰などの理由から、CO2 排出量やエネルギー効率等の環境負荷が小さく大量輸送が可能な交通機関として、鉄道システム復権の動きが世界的に高まっている [1][2]。そのため、列車ダイヤの超高密度化、列車種別の多様化、線区間相互乗り入れ運転の増加等を要因として各種輸送業務の複雑化が進んでおり、計算機システム導入による業務支援が求められている。一方、信頼性、保全性、可用性等を総合した広義の信頼性を「デペンダビリティ」と呼ぶ。これは数値尺度として厳密に定義することが困難な利用可能性を含めた包括的な概念である。元々はフォールトトレランス研究の分野で 1980 年代から提唱されたものだが [3]、情報技術と社会システムが密接不可分になりつつあることを背景として、情報セキュリティや経済・法律・政治等、様々な分野の知見を取り入れながら幅広い概念として発展しつつある [4]。例えば社会サービスのデペンダビリティの提唱 [5] や、サービスのデペンダビリティ標準化の動き [6] 等が挙げられる。社会活動や経済活動を支える重要な社会インフラである鉄道システムには、列車運行乱れというある種の非定型的な状況において「頼りがいのある」、すなわちデペンダブルなサービスを提供することが求められている。

列車運行が乱れた際に列車ダイヤを再作成し運転回復をおこなう業務を「列車運転整理」と呼ぶ。列車運転整理業務に関しては、支援システムの導入が早くから進められてきた。一方、再作成された列車ダイヤに従って列車を運行するために必要な資源である車両や乗務員の運用計画を再作成する業務を「列車運用整理」と呼ぶ。列車運用整理業務に関しては、未だ殆どの鉄道事業者が人手でおこなっているのが現状である。そのた

め、車両・乗務員の手配ミスや、場当たりの対応による業務の非効率化といった問題が生じている。こうした状況を改善すべく、列車運用整理を支援するシステムも徐々に開発されつつある。ただし、再作成された列車ダイヤの要求を満たしつつ、資源の利用可能性に関する様々な制約に配慮してスケジュールを再構成することの困難さから、意思決定に関わる多くの作業が依然として人手に委ねられているのが現状である。また、システム化事例の多くは、特定の路線を対象とした個別対応的なものであり、それらを他の路線へ適用することが難しいという問題もある。

以上より、本論文は鉄道輸送サービスのデペンダビリティ向上を目指し、列車運行乱れ時の列車運用整理業務に焦点をあて、取り組むべき課題を提示する。そして、それらの課題を解決するために、最適化技法（ラグランジュ緩和法、局所探索法、制約プログラミング）とヒューリスティクスとを組み合わせたリアクティブなスケジューリング方式を提案する。さらに実路線を含むダイヤ再編成データを用いた数値実験により、提案方式の有効性を示す。

1.2 本論文の構成

第1章では、序論として研究の背景と目的を述べると共に、本論文の構成を説明する。

第2章では、リアクティブスケジューリング技術に関する近年の研究開発動向をサーベイする。

第3章では、列車運用整理業務を支援する計算機システムの実現に向けて解決すべき課題を明らかにする。

列車運用整理業務において、旅客・貨物等を輸送する本線上の営業列車の運行に直接関わるものを、特に「本線運用整理」と呼ぶ。運行乱れ時に再作成された列車ダイヤの下で、資源である車両や乗務員の割当スケジュールを再作成することを目的とする。一方、車両基地における車両の清掃・点検といった保全作業に関わる業務を、特に「車両基地運用整理」と呼ぶ。運行乱れ時でも保全作業を円滑に進めるため、車両基地構内に

における車両の移動と配置のスケジュールを再作成することを目的とする。本章では、上記2つの業務におけるスケジュール再作成の特徴と、それらが満たすべき要件を整理する。さらに、整理結果をふまえ、鉄道輸送サービスのデベンダビリティを向上する列車運用整理支援システムの実現において解決すべき2つの課題を示す。

第4章では、一番目の課題に対処するための本線運用整理向けリアクティブスケジューリング技術について明らかにする。

まず、リアクティブスケジューリング問題の構造や制約を陽に扱うためのネットワークフローモデルを提案し、提案モデルで記述したリアクティブスケジューリング問題に対して整数計画問題としての定式化を与える。次に、問題の特徴をふまえたヒューリスティクスを組み込むことで問題を効率的に解くための2通りのスケジューリング方式を提案する。(方式1)は最適化技法としてラグランジュ緩和法を応用し、最適値の下界と上界とのギャップを用いた解の精度評価を行うことにより元のスケジュールからの変更量を最小化した再スケジューリングを求めるものである。(方式2)は最適化技法として局所探索法を応用したものであり、方式1では評価できない非線形な評価指標(全車両の折り返し時間の標準偏差と各車両の走行距離の差分の総和)にもとづく最適化を行うものである。さらに実路線の運行乱れ時の運用データを用いた数値実験結果について述べる。最後に異なる特徴を持つ2つの提案方式を組み合わせることでスケジューリング機能を構成することで、デベンダビリティをさらに向上する可能性について議論する。

第5章では、二番目の課題に対処するための車両基地運用整理向けリアクティブスケジューリング技術について明らかにする。

まず、車両基地のレイアウトを作業に使用する資源と捉えることで、対象問題を生産分野におけるフレキシブルジョブショップスケジューリング問題(FJSSP)の類似問題とみなす。そして、FJSSPを基本としつつ、レイアウト制約、実行可能性判断、元のスケジュールからの変更量の評価、という本問題固有の要素を加えた制約充足最適化問題としての定式化を与え、それにもとづく段階的なスケジュール作成のアーキテクチャを提案する。次に、定式化に基づいたスケジューリング方式として、制約プログラミング(CP)を用い

た方式を提案する。CPに探索の効率化と元のスケジュールからの変更量の最小化に寄与するヒューリスティクスを組み合わせることで方式を構成する。最後に、レイアウト形状が異なる3種類の車両基地を対象とした数値実験をおこない、本提案方式により変更量が少ないスケジュール再作成をレイアウト形状の違いに影響されず効率的に実行できることを示す。

第6章では、本論文で提案したアイデア、および有効性を検証した実験結果について総括し、さらに、鉄道輸送サービスのデペンダビリティのさらなる向上にむけた将来課題について考察する。

第2章

関連研究および関連技術

本章では、リアクティブスケジューリング技術に関する近年の研究開発動向をサーベイした結果を示す。

2.1 鉄道分野におけるスケジューリング技術の研究開発動向

鉄道の様々な計画は、長年にわたって人手主体で作られてきた。しかしながら、近年の計算機ハードウェアの進展等を背景として、計算機システムを用いて計画作成を高度に支援するためのスケジューリング技術の研究開発が進展しつつある [10][15]。

鉄道では、あらかじめ列車の運行に関する計画を作成し、それに従って列車を運行する。この計画のことを特に輸送計画と呼ぶ。輸送計画は、列車計画、車両運用計画、乗務員（運転士・車掌）運用計画、駅・車両基地の構内入換計画からなる [9][10][11]。

図 2.1 に鉄道輸送計画の基本的な構造を示す。図の左側の基本計画と実施計画は、事前に計画を作る段階である。基本計画はダイヤ改正時に主に年 1 回の頻度で作成される。実施計画は、波動的な需要を勘案して基本計画に季節列車や臨時列車等を加えることで毎日作成される。

一方、図の右側は、事故等によって列車運行に乱れが生じた場合に、実施計画に様々な変更を加えて再作成をおこなう段階である。これを広義の列車運転整理と呼ぶ。鉄道におけるリアクティブスケジューリング技術は、この広義の列車運転整理を支援することが目的である。

輸送計画のひとつである列車計画は、列車ダイヤとも呼ばれる。列車ダイヤは列車を「いつ」「どこへ」走らせるかを定めるものであり、輸送サービスの根幹をなす情報とい

える。したがって、鉄道は列車ダイヤを共通の情報として、輸送に関わる様々な部署が分散協調的に使命を遂行していく複雑システムとみなすことができる [12]。図 2.1 における様々な計画も、担当部署はそれぞれ異なるものの、他の部署の計画と関連を取りながら作成されている。

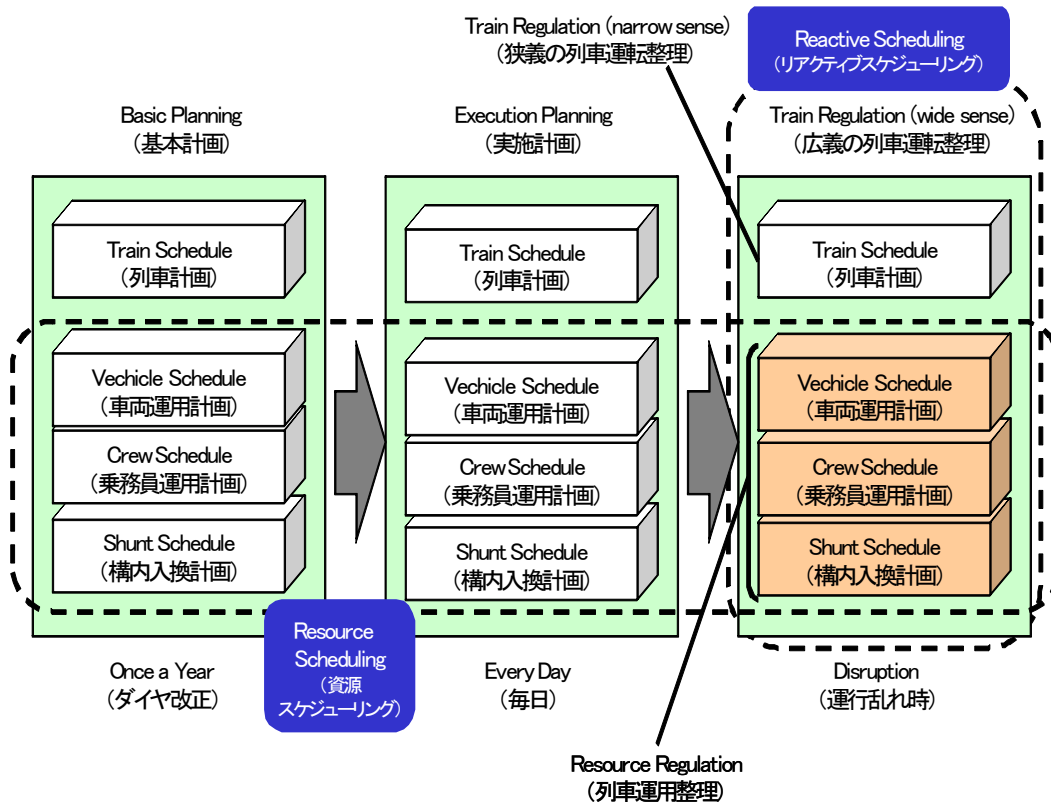


図 2.1: 輸送計画の基本構造

Fig.2.1: Fundamental Structure of Railway Transportation Planning

ここで、列車運転整理（あるいは単に運転整理）という用語は、狭義には列車ダイヤの再作成のことを指す。これに対して、輸送計画の他の要素である、車両運用計画、乗務員運用計画、構内入換計画を再作成することを列車運用整理と呼ぶ。以降では特に断らない限り、列車運転整理という用語は狭義の意味で用いる。

車両運用計画、乗務員運用計画、構内入換計画は、列車を動かすために必要な、車両や乗務員、進路・番線等の設備といった、いわゆる資源（リソース）の運用方法を定める計画である。そこで、これらの計画作成をまとめて資源スケジューリングと呼ぶことと

する。

以降では、鉄道におけるスケジューリング技術の研究開発動向を、リアクティブスケジューリングと資源スケジューリングに分けて説明する。なお、列車運用整理はこの両者にあてはまるが、リアクティブスケジューリングの説明は列車ダイヤの再作成（狭義の列車運転整理）に焦点をあてることとし、列車運用整理については資源スケジューリングの節でまとめて説明する。

2.1.1 列車ダイヤ向けリアクティブスケジューリング

列車ダイヤの再作成に関しては、リアクティブスケジューリング技術の研究開発が比較的盛んにおこなわれている [18]-[26]。

古くは知識工学、あるいは人工知能 (AI) 研究からのアプローチに基づく取り組みがある [18][19]。文献 [18] は運転整理を実行する上での戦略（例えば特急列車を優先させる）を、再作成の目的に応じた単位に細分化してルールを定義し、各目的の協調と統合を行いながらルールを適用して再作成を実行するものである。複雑な戦略であっても目的の単位に落とし込むことで再作成の要素を簡単なルールの組み合わせへと変換し、実時間での再作成を可能としている。文献 [19] は、運転整理の事例をルールとして活用するエキスパートシステムの研究である。IF-Then 形式で表現された列車ダイヤの変更ルールを格納するデータベースと、運転整理に必要な 4 種類の意思決定プロセスとを持ち、ルールを適用しながら 4 つの意思決定プロセスを巡回的に実行することで再作成を実行する。

こうしたエキスパートの知識を活用したアプローチの最近の研究開発事例として、文献 [20][21][22] が挙げられる。文献 [20][21] は、事故の発生地点や時間帯等に応じて頻繁にとられる運転整理手法を「運転整理パターン」として活用するものである。パターンを記述するための R と呼ぶ言語を導入し、パターンを解釈して列車ダイヤに変更を加えるメカニズム (R インタプリタ) を提案している。文献 [22] は、前述のパターンの考え方を踏襲し、運休と折返し変更に特化したアルゴリズムを提案している。高密度線区を

対象とした評価実験により、指令員によるものとほぼ遜色のない再作成結果が得られている。

エキスパートの知識を活用するアプローチは、再作成結果を確認する指令員にとって理解がしやすく、不自然さも少ない計画が得られやすいという利点がある。しかしながら、他の分野と同様に、鉄道分野においてもエキスパートから妥当な知識を獲得することはそもそも非常に困難であり（知識獲得のパラドックス [17]）、また、ルールの適用による問題解決のアプローチが、スケジューリングのような組み合わせ的要素を持つ問題に対しては本質的に組み合わせ爆発を押さえることが困難であることが次第に認識されつつある。上記の最近の研究事例も、ルール適用に最適化技法を組み合わせたり、再作成の手段と範囲を限定することでこの問題に対処している。

一方、近年の計算機性能の向上を背景として、運転整理の問題を組み合わせ最適化問題として捉え、最適化アルゴリズムを構築する試みが数多くおこなわれている [23]-[26]。

文献 [23][24] は、組み合わせ最適化技法としてメタヒューリスティクス [96] を応用したものである。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提唱し、PERT[92] とメタヒューリスティクス的一种であるシミュレーテッド・アニーリングを組み合わせたスケジューリング方式を提案している。首都圏近郊路線の線区を対象として数値実験をおこない、実用的な列車ダイヤの再作成を分単位の世界オーダーで実行可能なことを確認している。文献 [24] は、前記文献 [23] の研究成果を発展させたものである。列車を選択する旅客の行動をシミュレーションを用いて推定し、それを評価に活用することで旅客の不満をよりきめ細かく考慮した再作成をおこなう。

制約充足技法の一種として制約プログラミング (Constraint Programming, CP)[88][89][90][95] が知られているが、これをバックトラックサーチ等の離散型の解探索フレームワークと組み合わせることで組み合わせ最適化技法として用いることができる [91]。このように制約プログラミングを応用した研究開発事例として、文献 [25][26] がある。

文献 [25] は、列車の駅における出発順序を対象としたスケジューリング方式を提案している。列車の走行を計算するための様々な条件を制約として定義し、総遅延時間を最小

化する出発順序を CP を用いて算出する。JR 東日本の新幹線の実データを用いて検証をおこない、総遅延時間を減少できることを確認している。文献 [26] は、CP に知識活用型のアプローチを取り入れたものである。IF-Then 形式で表現された指令員の運転整理ルールを CP で扱うことができる制約に変換し、CP による解探索の中で制約化されたルールを動的に適用することで知識を反映したスケジュールを効率良く算出する。

以上は主に国内の研究開発動向である。海外、特にヨーロッパにおいても運転整理の問題が着目されている [27]。最近では文献 [28]-[32] といった研究がみられる。海外における研究の特徴として、分枝限定法、ラグランジュ緩和法 [77]、列生成法 [82] といった、最適化技法の中でもより厳密解を指向した技法を応用することでスケジューリングの完全自動化を狙うものが多いことが挙げられる。この傾向は後述する資源スケジューリングの研究も同様である。

2.1.2 資源スケジューリング

列車運行に必要な資源の運用方法を定める車両・乗務員運用計画と構内入換計画に対しては、それらの再作成（列車運用整理）に焦点をあてた本格的な研究は未だ殆ど行われていないのが現状である。そこで本節では、再作成だけでなく事前計画の作成（基本計画と実施計画）も含めて資源スケジューリングに関する研究開発動向を幅広く概観する。なお、本分野のより包括的なサーベイとしては、例えば文献 [16][104] を参照されたい。

車両・乗務員運用計画

スケジュールの再作成を対象とした研究は、後述する事前計画に関する研究と比較して圧倒的少数に留まっているのが現状である。例えば文献 [33] は乗務員運用計画を対象としたものである。乗務員運用計画の再作成問題を、乗務員の列車への割り当てを求める組み合わせ最適化問題としてモデル化し、メタヒューリスティクス的一种であるタブーサーチにより準最適解を探索する。実路線データを用いた数値実験により、制約条件を

満たすだけでなく、手作業による再作成結果と同等のスケジュールが得られることを示している。

一方、事前計画の作成に関しては、車両と乗務員の双方共に従来より幅広く研究がおこなわれている [34]-[39]。例えば Cacchiani et al.[36] は、車両運用計画の中でも特に要求座席数を考慮して車両 (train unit) の組合せを決定する必要がある、いわゆる座席要求制約を含むものを対象とし、整数線形計画法 (ILP) としての定式化とヒューリスティック解法を提案している。Caprara et al.[37] は乗務員の運用計画作成を集合被覆問題 (Set Covering Problem, SCP) として定式化し、列挙型アルゴリズム (Enumeration algorithm) とラグランジュ緩和法 (Lagrangian relaxation method)[77] を用いた近似解法を提案している。Fischetti et al.[38] は、車両基地の数に制限があるなど単純化されているが、依然として NP 困難なケースを対象として、乗務員運用計画と車両運用計画の双方に適用可能な 0-1 線形整数計画問題としての定式化と、多面体アプローチに基づく厳密解法を提案している。Haase et al.[39] は、都市交通システムを対象とし、車両と乗務員の運用計画を同時に立案する問題に対して、列生成法 [82] を分枝限定法に組込んだ分枝価格法 (Branch and Price)[83] による厳密解法を提案している。

以上述べた事前計画に関する研究は、いずれも海外、主にヨーロッパの研究である。ヨーロッパでは、上下分離、すなわち公的機関が保有するインフラの上で、複数の鉄道運営会社 (Train Operating Company, TOC) が商業ベースで列車を運行するという図式が定着している。そのような環境においては、資源の運用計画を効率的に作成することと高品質な資源の運用計画を作成することが重要となってきたという事情が研究活性化の背景にある [14]。一方国内においては、例えば乗務員運用計画に関してメタヒューリスティクス的一种である Simulated Evolution 法を用いた手法 [40]、車両運用計画に関して確率的アルゴリズムと最短経路アルゴリズムの組み合わせによる手法 [41] や列生成法を応用した手法 [42][43] の研究がみられる。

構内入換計画

構内入換計画に関しては、その再作成に対して本格的に焦点をあてた研究は見あたらない。一方、事前の計画作成に関しては、これまでにいくつかの関連研究がおこなわれている [44]-[49]。

文献 [44] は、オランダの鉄道路線を対象として、翌日の列車に充当する車両の組合せと構内配置の決定を整数計画問題として定式化し、列生成法を用いて解を求めるものである。文献 [45] は、ファジィ理論によるモデル化と遺伝的アルゴリズム (GA) の応用により、車両基地における車両の組合せとその分解・組立ての構成手順を求めるものである。文献 [46] は、車両基地を出区する際の移動効率を考慮して車両配置を最適化するグラフ理論を用いた研究である。また文献 [47][48][49] は、駅構内における入換計画作成を対象とし、PERT と近似解法の一つである確率的局所探索の組合せによるアルゴリズムを提案している。

なお、駅や車両基地における設備の運用は、海外と国内、鉄道事業者、あるいは鉄道路線で大きな違いがある。そのため、これらの研究は問題の捉え方にバリエーションがあり、扱う問題が必ずしも同一ではないことに注意する必要がある。

2.2 リアクティブスケジューリング技術に関する研究開発動向

本節では、リアクティブスケジューリング技術に関する一般的な研究開発動向として、スケジューリング問題の汎用モデルである資源制約付きプロジェクトスケジューリング問題 (Resource Constrained Project Scheduling Problem, RCPSP) を対象とした研究開発動向を概観する。なお、より包括的なサーベイについては、例えば文献 [50][51] を参照されたい。

RCPSP は、資源に関する条件がある中で各作業の開始時刻および終了時刻を決定する問題である。RCPSP は、フローショップ問題やジョブショップ問題をはじめ、多くのスケ

ジューリング問題を定式化できるという汎用性の高さから、近年改めて注目を集めている [52]。

RCPSP を対象としたリアクティブスケジューリングは、大別して Completely Reactive Scheduling (完全リアクティブスケジューリング) と Predictive-Reactive Scheduling (予測・リアクティブ併用スケジューリング) の2種類に分類することができる [50][51]。Completely Reactive Scheduling は、事前に作成された基本スケジュール (Baseline Schedule) を持たず、プロジェクトの実行時に実行順序や資源割当を戦略に基づき動的に決定する。一方、Predictive-Reactive Scheduling は基本スケジュールを活用するものであり、これはさらに Schedule Repair (修正) と Rescheduling (再作成) の2つに分類される。前者はヒューリスティックルールを用いて基本スケジュールを修正するアプローチを指す。処理時間は早いものの、ルールを固定的に用いる方法のため必ずしも良い結果が得られない可能性がある。一方、後者の Rescheduling は、基本スケジュールを用いつつ必要に応じてスケジュール全体を再構成するアプローチである。以降では、本研究に特に関連が深い Rescheduling について主に説明する。

スケジュールの再作成結果は、しばしば基本スケジュールから大きくかけ離れたものとなる。それではスケジュールの連続性が失われるため、再作成結果の実行現場にとっては不都合なことが多い。そこで、スケジュールの安定性 (stability) に着目し、基本スケジュールからの乖離をできるだけ小さくすることを目的とした研究がおこなわれている [53]-[56]。例えば文献 [53] と文献 [54] は、RCPSP の定式化における目的関数の定義に関する研究である。基本スケジュールからの乖離を表す評価指標として、前者は開始時刻の差、後者は異なる資源が割当てられた作業の数を用いることをそれぞれ提案している。その他、文献 [55] は変更する作業数を最小化するタブーサーチを用いたスケジューリング方式、文献 [56] は、ある時間断面で基本スケジュールと一致 (match-up) するように再作成を行うスケジューリング方式を提案している。

RCPSP をより現実に近い形に拡張したモデルとして、Multi-mode RCPSP (MRCPSP) がある。MRCPSP は RCPSP と異なり、作業の処理時間や要求資源等が固定ではなく、いく

つかのバリエーションの下で実行することができる。この処理時間や要求資源等のバリエーションのことをモードと呼ぶ。モードは、資源を多く使うことでより早く作業を完了できたり、ある作業で利用できる資源が複数種類あるといった、現実のスケジューリングにおいて生じる実際的な条件を扱うために導入された。MRCPSP 向けのリアクティブスケジューリングに関しては、文献 [57] において複数の厳密解法とタブーサーチを応用したスケジューリング方式が提案されている。それらの性能を複数のベンチマーク問題を用いて比較評価し、いくつかのベンチマーク問題に対して有望な結果を得ている。

上述の MRCPSP を含め、RCPSP には様々な派生モデルが存在するが、RCPSP に対するリアクティブスケジューリング方式の構築には、派生モデルに対するものも含め、これまで様々な技法が用いられてきた。主に用いられてきた技法としては、ヒューリスティクス、メタヒューリスティクス、マルチエージェント、知識処理 (AI) が挙げられる。例えば文献 [58] は、RCPSP の一種である複数資源を持つジョブショップスケジューリング問題を対象として、前述の match-up スケジュールを生成するためのヒューリスティクスを用いたスケジュール修正方式を提案している。文献 [59][60][61] は、メタヒューリスティクスの一種である GA を応用したスケジューリング方式を提案している。文献 [62] は、鉄鋼生産ラインを対象としたマルチエージェントによる分散協調型のスケジューリングアーキテクチャを提案している。最後に、文献 [63] は知識ベースシステム (Knowledge-based system) の応用事例である。ユーザとのインタラクション部分に知識ベースシステムを活用することで、リアルタイムなイベントへの対処を効果的に支援できることが示されている。

第3章

列車運用整理業務におけるリアクティブスケ ジューリング

本章では、列車運用整理業務を支援するシステム実現のための課題を明らかにする。

3.1 列車運行乱れ時の輸送サービス回復と列車運用整理業務

鉄道分野においては、輸送力増強や旅客ニーズの多様化を背景とした列車の高速・高密度化、列種・車種の細分化、旅客サービス向上等により、列車ダイヤの策定や運行管理をはじめとする輸送業務全般が複雑化しつつある。

特に、列車運行が乱れた際に輸送サービスを速やかに回復する当日の運行管理に関わる業務の難易度が高まっている。例えば日本の首都圏においては、複数の鉄道路線が交錯し巨大で複雑な交通ネットワークを形成している。そのため、列車運行乱れが一旦発生すると、たとえそれが局所的なものであってもネットワーク全体に影響が波及しやすく、他の多くの列車にも運行の乱れを生じるため、その収束には大きな労力を必要とする。

図 3.1 に列車運行乱れ時の輸送サービス回復業務の全体像を示す。

列車運行乱れ時は、路線全体の運行を監視する指令所において、列車ダイヤの再作成と車両・乗務員の運用計画の再作成が実施される。まず指令所は現業部門である駅、乗務員区所、車両基地等と連絡を取り、事故の状況や復旧の見込み等の情報を把握しながら列車ダイヤを再作成し、さらに列車ダイヤの変更内容に齟齬が生じないように車両と乗務員の運用計画を再作成する。次に再作成結果に基づき、列車時刻等の列車運行内容の変更と、それに伴う予備車両や乗務員の手配といった各種の指示が現業部門へ伝達され

る。次に車両基地では、これらの伝達を受けて、構内の車両の移動と配置に関わる計画である構内入換計画を再作成する。なお、予備車両をすぐに出せないといった理由で現業部門が指令所の指示を実行できない場合は、その旨を指令所へフィードバックし、列車ダイヤと車両・乗務員運用計画を調整する。

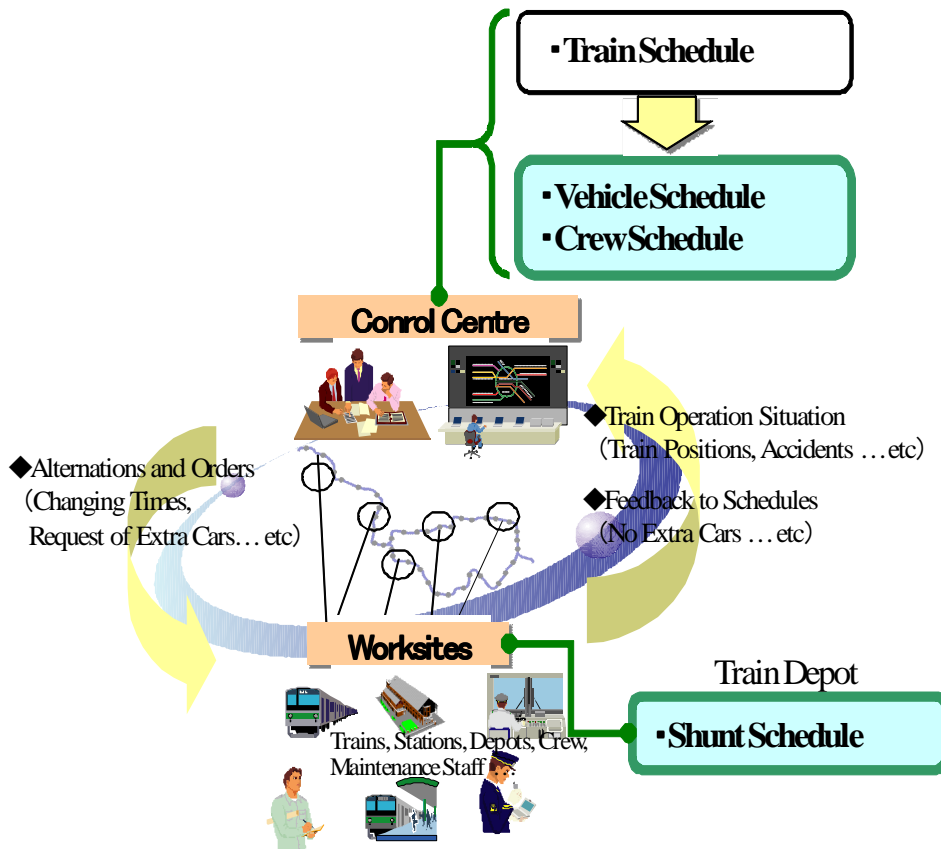


図 3.1: 列車運行乱れ時の輸送サービス回復業務
Fig.3.1: Train Recovery Operations under Disruption

2章で述べたとおり、これらの輸送サービス回復に関わる業務の中で、列車ダイヤを再作成する業務が列車運転整理である。一方、列車を運行するために必要な資源である車両・乗務員、構内設備の運用計画を再作成する業務が列車運用整理である。

ここで、列車運用整理業務において、旅客・貨物等を輸送する本線上の営業列車の運行に直接関わるものを、特に本線運用整理と呼ぶ。運行乱れ時に再作成された列車ダイヤの下で、資源である車両や乗務員の割当スケジュールを再作成することを目的とし、指

令所において実施される。

一方、列車運用整理業務において、車両基地における車両の清掃・点検といった保全作業に関わる業務を、特に車両基地運用整理と呼ぶ。運行乱れ時でも保全作業を円滑に進めるため、車両基地構内における車両の移動と配置のスケジュールを再作成することを目的とし、車両基地において実施される。

3.2 輸送サービス回復のシステム化の現状

運行管理業務を高度化するために、長年にわたって様々な技術が開発されてきた。これらの開発成果の多くは運行管理システムとして実用化されている [64]-[66]。その基本機能のうち、(1) 現場機器、沿線設備等の状態監視、(2) 列車の現在位置の把握、(3) 列車ダイヤに基づく自動進路制御、の3つに関しては、1950年代からの取組みの末に技術的には確立されている [8]。

大手鉄道事業者は、列車運行乱れ時の輸送サービス回復を効率化するため、列車ダイヤを再作成する運転整理業務に焦点をあてて運行管理システムの高度化を進めている [8]。それらは列車の在線位置の表示やシステム入力のグラフィック化、運行遅延の予測といった機能を備えている。また、先進的な一部の鉄道事業者においては、列車ダイヤの再作成案をシステムが提示する自動提案機能を備えているものもある [67][68]。

その一方で、車両や乗務員の運用計画を再作成する運用整理業務は、未だ殆どの鉄道会社において紙とペンを用いて人手にて行われているのが実状である。そのため、車両・乗務員の手配ミスや、場当たりの対応による運用の非効率化といった問題が生じている。列車運転整理のシステム化がある程度進んできたことを契機として、こうした状況を改善すべく、特定の路線を対象とした運用整理業務に関わる試験システムの開発も報告されている [7][69]-[75]。ただし、それらのシステムは、運行乱れや運転整理を契機とした運用上の警報（計画に対する未充当や各種妥当性チェック）の出力、車両や乗務員の現在位置の追跡、割当候補の一覧表示、構内進路のリモート制御といった補助的な機能

を主眼としたものである。スケジュール再作成案の自動提案機能を備えているものもあるが、未だ実用レベルには至っておらず、運用計画の再作成自体はあくまでも人手主体で行うことを前提としている。

また、列車ダイヤの再作成にあたっては、本来は資源の運用計画も併せて再作成する必要があるが[12][13]、運用整理業務のシステム化が進んでいないため、実用化された運転整理の自動提案機能も列車の出発順序の変更といった資源の運用計画への影響が少ない限定的なものに留まっているという現状もある。

3.3 リアクティブスケジューリングの課題

運行乱れの際には、指令員と呼ばれる運行管理業務の担当者（多くは長年の経験を持つベテラン要員）が列車ダイヤを迅速に再作成することで遅延の拡大を防いでいる。指令員は、列車の着発時刻や出発順序の変更、いくつかの列車の運休、臨時列車の設定等をおこなうことで列車ダイヤの回復案を作成するが、それらの変更を実施可能なものとするために本線運用整理が必要となる。すなわち、各列車に対する車両と乗務員の割当スケジュールを、再作成された列車ダイヤと整合性を取りつつ修正を施す必要がある。それを限られた時間の中で実施しなければならない。交通ネットワークの巨大化・複雑化に伴い、人手でのスケジュール再作成が困難となり、本線運用整理の難しさが近年ますます増大しつつある。

駅や乗務員区所といったスケジュール実施現場の混乱を抑えるため、再作成にあたっては元のスケジュールからの変更量を少なくすることが求められる。また、実行可能なスケジュールを得るには、列車ダイヤからの要求と資源の運用に関する制約の双方を満足する必要がある。運転再開までの限られた時間の中でこれらの要件を満たして再作成を実施することが必要である。

一方、車両基地運用整理は各車両基地毎に実施されるが、再作成をおこなう担当者は、保全作業の現場で経験を積み、担当する車両基地の構造や運用を知り尽くしたベテラン

要員が充てられることが多い。保全作業の現場は一般的に若手が少なく、運用整理には相応の技量が求められるため、担当者のなり手が不足している。我が国の就業者構造の変化、特に団塊世代の定年に起因する技能継承問題（2007年問題）[76]が叫ばれて久しいが、ノウハウを持つベテラン要員の引退後も車両基地運用整理業務を円滑に行うことが鉄道事業者にとっての懸案のひとつである。そこで、これまでベテラン担当者のノウハウに支えられてきた人手主体の業務形態を計算機システムを活用した形態へと速やかに移行することが重要な課題になりつつある。

車両基地のスケジュールの再作成にあたっては、本線運用整理と同様に元のスケジュールからの変更量の最小化が求められるが、それに加えて、実行可能性を確保するために車両基地のレイアウト（配線構造）を考慮することが重要である。レイアウトは車両基地毎に異なり、しかも複雑・多岐にわたるため、それらをスケジューリングの制約として共通的に扱うことが求められる。

以上より、鉄道輸送サービスのデペンダビリティ向上のために列車運用整理支援システムを実現するためには、次の課題を解決することが必要である。

課題1 再作成された列車ダイヤからの要求と資源運用制約の双方を可能な限り満たした上で、元のスケジュールからの変更量を最小化した再スケジューリング結果を算出可能な効率的なスケジューリング方式が必要である。

課題2 車両基地の様々なレイアウトに対応し、車両の移動・配置に関わる物理的な制約とその他の資源運用制約とを満足した再スケジューリング結果を算出可能な効率的なスケジューリング方式が必要である。

第4章

本線運用整理のためのリアクティブスケジューリング方式

本章では、課題1に関し、再作成された列車ダイヤからの要求と資源運用制約の双方を可能な限り満たした上で、元のスケジュールからの変更量を最小化した再スケジューリング結果を効率良く算出可能な本線運用整理向けリアクティブスケジューリング技術について明らかにする。

4.1 研究の背景

本線運用整理は、運行乱れが発生し列車ダイヤに変更が生じた際に、車両や乗務員（運転士、車掌）を列車に割り当てる業務である。本線運用整理を対象としたリアクティブなスケジューリング問題を、本章では特に車両リスケジューリング問題 (Vehicle Rescheduling Problem, VRP)、もしくは乗務員リスケジューリング問題 (Crew Rescheduling Problem, CRP) と呼ぶこととする。

VRP/CRPの類似問題として、新規路線の導入や大幅なダイヤ改正時に施策の分析や評価をしたり、日々の列車運行をおこなうために、車両や乗務員の充当スケジュールを事前に作成するものがある。これらを、車両スケジューリング問題 (Vehicle Scheduling Problem, VSP)、もしくは乗務員スケジューリング問題 (Crew Scheduling Problem, CSP) と呼ぶ。

VSP/CSPについては従来より幅広く研究がおこなわれている [34]-[39]。例えば Cacchiani et al.[36] は、VSPの中で特に要求座席数を考慮して車両 (train unit) の組合せを決定する必要がある、いわゆる座席要求制約を含むものを対象とし、整数線形計画法 (ILP) による定式化とヒューリスティック解法を提案している。Caprara et al.[37] は CSP を集合被覆問題

(Set Covering Problem, SCP)として定式化し、列挙型アルゴリズム (Enumeration algorithm) とラグランジュ緩和法 (Lagrangian relaxation method)[77] を用いた近似解法を提案している。Fischetti et al.[38] は、車両基地の数に制限があるなど単純化されているが、依然としてNP 困難なケースを対象として、乗務員スケジューリングと車両スケジューリングの双方に適用可能な 0-1 線形整数計画問題としての定式化と多面体アプローチに基づく厳密解法を提案している。Haase et al.[39] は、都市交通システムを対象とし、車両と乗務員のスケジュールを同時に立案する問題に対して、列生成法 [82] を分枝限定法に組込んだ分枝価格法 (Branch and Price)[83] による厳密解法を提案している。

VSP/CSP に関する従来研究の多くは、事前のスケジュール作成を目的とすることから、基本的にスケジュール全体を「白紙から」作成することを前提としている。事前スケジュールの作成は時間的余裕が十分であることを前提にできるため、妥当な解を得るまでに長時間要しても構わないという立場でスケジューリング方式が設計されているものが多い。したがって、これらを時間的な制約が厳しく、またリアクティブなスケジューリング問題として独自の要件を持つ VRP/CRP へ適用することは不向きと考えられる。

VSP/CSP は、VRP/CRP と同じく列車に対する資源 (車両・乗務員) の充当スケジュールを作成するものである。よって、内容はともかくとしてその出力結果 (スケジュール) の形式は VRP/CRP と全く同じである。したがって、両者は良く似た問題と言えるが、重視すべき評価指標などいくつかの要件が異なる。加えて、VRP/CRP は時間的制約が厳しく、状況が変化する中でリアクティブにスケジュールを再作成する必要がある。そのため両者は本質的に異なる問題と言える。

再作成された列車ダイヤの要求を満たしつつ、資源の利用可能性に関する様々な制約に配慮してスケジュールを再構成することの困難さから、VRP/CRP を対象として自動スケジューリング機能を実用化した例は殆どみあたらない。しかし、運行乱れからの迅速な回復に対する社会的関心がますます高まるにつれ、VRP/CRP 向けのスケジューリング方式が文献 [33] において提案されている。この方式は、まず必要な制約のいくつかを緩和することで初期解を構築し、次にタブー探索を用いて制約違反を減らす方向に解を修

正するものである。運行乱れの実際の事例を対象として数値実験をおこなった結果、提案方式により制約条件を満足する解を算出するだけでなく、エキスパートが作成したスケジュールと同等であることを確認している。

本章では、課題1である、再作成された列車ダイヤからの要求と資源運用制約の双方を可能な限り満たした上で、元のスケジュールからの変更量を最小化した再スケジューリング結果を算出可能な効率的なスケジューリング方式を提案し、実路線を含むダイヤ再編成データを用いた数値実験により提案方式の有効性を示す。

4.2 問題定義

4.2.1 ネットワークフローモデルによる本線運用整理のモデル記述

列車ダイヤに含まれる各列車の走行スケジュールを、乗務員の交代や乗り継ぎが可能な駅毎に区切ったものをトリップ (trip, 旅程) と呼ぶ。複数のトリップをつないで乗務員1人のスケジュールを構成する。これを行路 (pairing) と呼ぶ。車両の場合もこれと同様である。すなわち、車両の折り返しや車両基地への入出区が可能な駅毎に各列車の走行スケジュールを区切ったものをトリップとして、これを複数つないだものを一車両のスケジュール (行路) とする。以降では特に断らない限り、乗務員と車両とを区別せず「資源」と呼ぶこととする。

1個のトリップを頂点、トリップ間の可能な遷移、すなわちトリップの可能な実施順序を有向リンクとするグラフを考えれば、1個の資源に対するスケジュールは、そのグラフ上の資源の流れ (フロー) として表すことができる。車両に対するネットワークフローモデルの例を図4.1に示す。点線の有向リンクがトリップ間の可能な遷移を表す。また、実線の有向リンクが資源の1単位のフローを意味する。すなわち、実線の有向リンクの始点と終点の両方のノード (トリップ) に対して、リンクの向きと同じ順序で同一の資源を割り当てることを意味する。また、あるひとつの資源に着目し、スケジュールの開始を意味する特別なノード (開始ノード) を出発点とし、同じくスケジュールの終了を

意味する特別なノード（終了ノード）に至るまでフローを連続的に辿ることにより、その資源の割当スケジュールの全体が得られる。

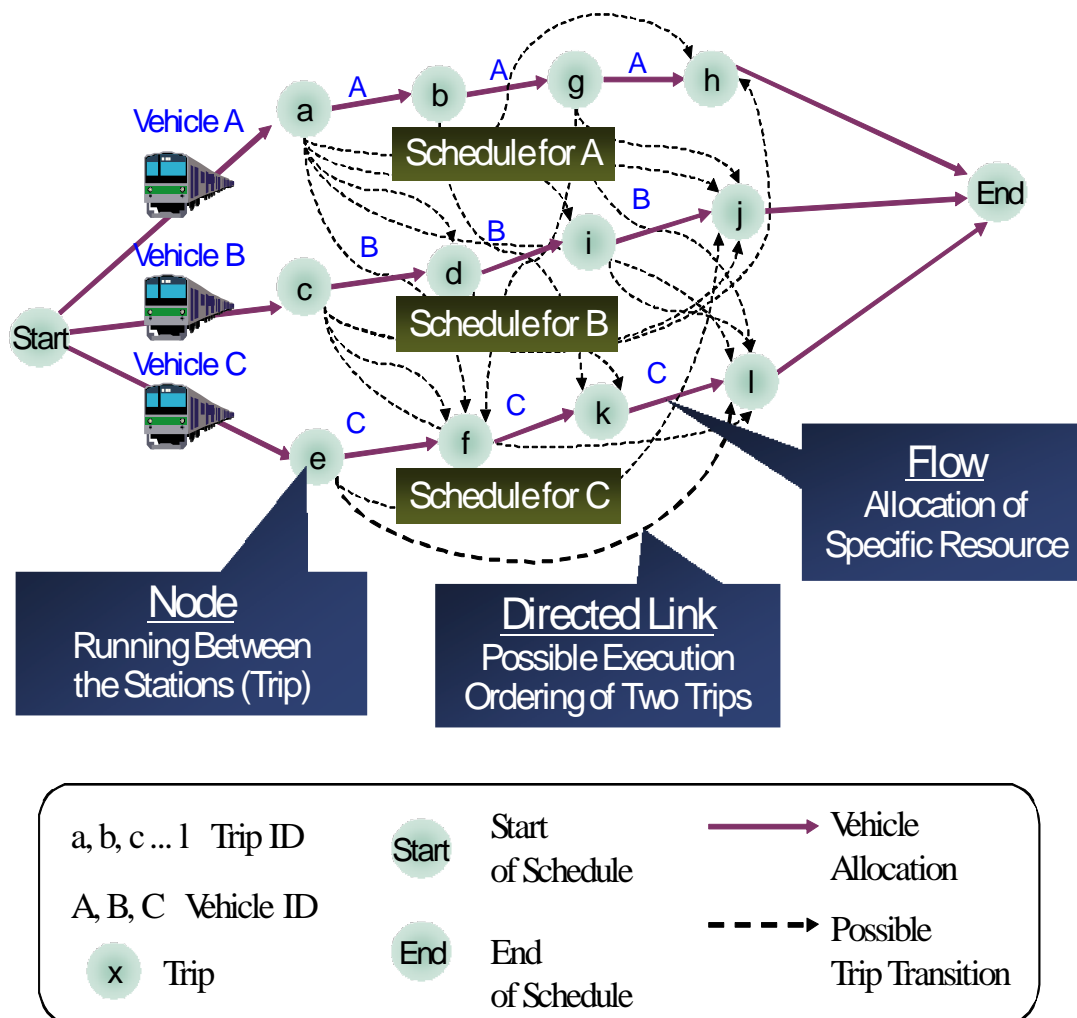


図 4.1: ネットワークフローモデル (トリップフローモデル)

Fig.4.1: Network Flow Model for Vehicle Rescheduling (Trip Flow Model)

上記のネットワークフローモデルにおいて、ノードは列車ダイヤに含まれる列車の運行時刻と位置の情報を属性として持つ。これはすなわち、スケジュールを再作成する上で考慮すべき列車ダイヤからの要求を意味する。

また、点線の有向リンクは、資源をトリップに割当てるとして考慮すべき資源の運用に関する制約を抽象化したものと捉えることができる。例えば、必ず守るべき物理的な制約として、点線有向リンクの始点ノードのトリップ (a) と、終点ノードのトリップ (b) と

の間には、少なくとも次の2つの関係が成立しなければならない。

$$\text{トリップ } a \text{ の終了時刻} \leq \text{トリップ } b \text{ の開始時刻} \quad (4.1)$$

$$\text{トリップ } a \text{ の到着駅} = \text{トリップ } b \text{ の出発駅} \quad (4.2)$$

また、その他の制約としては、駅での車両折り返しや乗務員交代に関わる時間的な制限が挙げられる。例えば特急電車等の長距離列車は、端末駅で折り返しを行う前に清掃や座席転換等の作業が必要である。そのため、その作業に必要な間合い時間を確保しなければならない。乗務員に関しても同様であり、他の車両へ乗務を交代する際には、車両への移動や時刻表の手渡し等に要する時間を確保する必要がある。そこで、物理的な制約を満足する点線有向リンクの中で、車両折り返しや乗務員交代に必要な時間に満たない長さのリンクを削除すれば、残りのリンクは制約を満足することを保証できる。

以上より、本モデルは、本線運用整理に必要な要素である、列車ダイヤからの要求と、資源の運用制約と、スケジュール自身とを統合的に記述することが可能である。

列車運行の乱れに応じて列車ダイヤを変更すると、ネットワークモデルのグラフ構造も変化し、それに伴いスケジュールの実行可能性が損なわれる場合がある。例えば、ある2つのトリップ a, b について、 b の開始時刻は a の終了時刻の後であり、 a から b への点線有向リンクが存在していたが、トリップ a の終了時刻（すなわち列車の到着時刻）に遅れや変更が生じて時刻の逆転が生じると、式(4.1)の制約を満たさないため、 a から b への点線有向リンクが消滅する。したがって、もし a から b の向きにフロー（実線の有向リンク）が存在すれば、そのフローが意味するところの「 a と b に同じ資源を割当て、 a 、 b の順序で実行（列車を運行）する」というスケジュールは物理的に実行不能となる。そこでスケジュールを再作成する必要が生じる。

スケジュールの再作成にあたっては、厳しい時間制約の下で運行回復を迅速におこなうという観点から、実行可能性を確保することが最も重要視される。ただし、資源の運

用効率といった観点からは、実行可能性を確保するだけでなく、鉄道事業者の運用規則や労働協定等に基づく様々な評価指標を満たすことが望ましい。車両における評価指標の例としては、「駅での折り返し時間にばらつきが少ない」、「最終到着駅が翌日の運用開始駅と一致する」、「走行距離の予定との差が少ない」等がある。一方、乗務員に関しては、「乗り換え回数が少ない」、「便乗数が少ない」、「食事や休憩の時間が十分に確保されている」、「超過勤務が少ない」といったものが挙げられる。これらは事前スケジュールの作成である VSP/CSP の評価指標と全く同じである。

上記のように様々な評価指標が存在するが、スケジュール再作成にあたりそれらのどれを採用するか、またどの評価指標をどれだけ重視するかは、一般的に鉄道路線や鉄道事業者毎に異なる。これに対して、鉄道路線、鉄道事業者、車両や乗務員の違い等に依存しない普遍的な評価指標として「元のスケジュールからの変更量を少なくする」が挙げられる。

この評価指標には以下に述べる3つの利点がある。

- 変更量が少なければ、スケジュールの変更に伴う駅や乗務員区所、車両基地といったスケジュール実施現場の混乱を最小限に留めることが可能となる。すなわち、運行乱れ時はただでさえ現場作業が混乱・輻輳しがちだが、変更量が少なければ乗務員への変更連絡や車両の緊急出庫の準備といった変更手配に要する現場作業者の負荷を軽減することができる。さらには連絡ミスといった取り扱い誤りを減らすことも期待できる。
- 変更量が少ないスケジュールであれば、再作成を実施する担当者が再作成の内容を短時間で把握して承認することができる。
- 元のスケジュールは、そもそも様々な評価指標を総合的に考慮して作成されているはずである。したがって変更量を少なくできれば、それらの評価指標を高いレベルに保つことが期待できる。

一方、事前にスケジュールを作成する問題である VSP/CSP は、時間制限が比較的緩やかであり、現場作業におけるスケジュール変更の影響を考慮する必要もないことから、このような評価指標は不要である。したがって、この「元のスケジュールからの変更量を少なくする」という評価指標は、リアクティブなスケジューリング問題である VRP/CRP において特徴的であり、かつ最も重視すべき評価指標といえる。

集合分割 / 被覆によるモデル化との比較

VRP/CRP に良く似ており、事前スケジュールの作成問題である VSP/CSP は、集合分割 (Set Partitioning)、あるいは集合被覆 (Set Covering) の考え方をを用いたモデル化が良く知られている。

すなわち、トリップの集合が与えられたとして、各資源のスケジュール(行路)を、トリップの集合を分割する、あるいは被覆する部分集合として表現する。以降では、この集合分割もしくは集合被覆を用いたモデル化を集合分割 / 被覆モデルと呼ぶ。

集合分割 / 被覆モデルが各資源のスケジュールを記述するものであるのに対して、ネットワークフローモデルは前述のとおりスケジュールだけでなくそれ以外の要素も扱うことができる。すなわち、本線運用整理を扱う上で不可欠な要素である、列車の運行時刻及び位置の情報と資源の運用制約という2つの要素も表現することができる。

また、スケジュールの表現に関しても、集合分割 / 被覆モデルはネットワークフローモデルと比較して限定的である。すなわち、集合分割 / 被覆モデルで表現されたスケジュールからは、それに含まれるトリップは識別することができるが、トリップの実行順序に関する情報は得ることができない。これに対して、ネットワークフローモデルはフローを用いてトリップの実行順序を表現することが可能である。

4.2.2 本線運用整理リアクティブスケジューリング問題の定式化

以下では、前節で定義したネットワークフローモデルを用いて VRP/CRP を整数計画問題として定式化する。

記号の定義

V : ノード集合。 $V = \{1, \dots, n\}$.

トリップを意味する通常ノードと、後述するダミーノードから構成される。なお、以降では特に断らない限り、トリップと通常ノードとを区別せずに扱う。

E : 有向リンク集合。 $E = \{a_{ij} \equiv (i, j) \mid i, j \in V\}$.

i, j がそれぞれ通常ノードである有向リンク (i, j) があり、それが E の要素ならば、トリップ i とトリップ j との間に、少なくとも前述の式 (4.1) 及び式 (4.2) の関係が成立する。すなわち、

(トリップ i の終了時刻) \leq (トリップ j の開始時刻)、かつ

(トリップ i の終了場所) = (トリップ j の開始場所)

でなければならない。

R : 資源集合 $R = \{1, \dots, m\}$.

s, t : スケジュールの開始 (s)、終了 (t) を表すダミーノード。

s から他のノードに接続する有向リンクは全て E の要素である。同じく、 t 以外のノードから t に接続する有向リンクは全て E の要素である。また、 s を終点とする有向リンクは E に含まない。同様に t を始点とする有向リンクは E に含まない。以降では、便宜上 s をノード $1 (\in V)$, t をノード $n (\in V)$ する。

なお、ノードと有向リンクの定義から、ノード集合 V と有向リンク集合 E から構成されるグラフは閉路を持たない。すなわち有向非巡回グラフ (Directed Acyclic Graph, DAG) である。

x_{ij}^k : 有向リンク $(i, j) \in E$ 上の資源 k のフローを表す決定変数。 ($(i, j) \in E, k \in R$).

0か1の離散値を取る。1ならばノード i からノード j に向けて資源 k を流す。すなわち、 i と j に対して資源 k を割当て、かつこの順序でトリップを実行（列車運行）する。

X^k : 資源 k のフロー変数 x_{ij}^k を i 行 j 列の要素とする $n \times n$ 行列。 ($k \in R, (i, j) \in E$).

$$X^k \equiv \begin{pmatrix} x_{11}^k & x_{12}^k & \cdots & x_{1n}^k \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1}^k & x_{n2}^k & \cdots & x_{nn}^k \end{pmatrix}$$

ただし、 $(i, j) \notin E$ ならば、 X^k の i 行 j 列要素は定数0とする。

開始ノードであるノード1を出発点として、終了ノードであるノード n に到達するまで、 $x_{ij}^k = 1$ を満たす有向リンク $(i, j) \in E$ をグラフ上で辿って得られるノードの系列が資源 k のスケジュールを表す。以降では特に断らない限り、 X^k を資源 k のスケジュールと呼ぶ。

X : X^k を全て足し合わせた $n \times n$ 行列。

$$X \equiv X^1 + X^2 \cdots + X^m = \sum_{k=1}^m X^k$$

行列 X の各要素は、対応する有向リンクにおけるフローの合計量を意味する。すなわち X の i 行 j 列要素を x_{ij} とすると、 $(i, j) \in E$ に対しては $x_{ij} = \sum_{k=1}^m x_{ij}^k$ 、それ以外は $x_{ij} = 0$ と定義される。

e_{ij}^k : 割当の可否。 ($k \in R, i, j \in V$).

$$e_{ij}^k = \begin{cases} 1 & i \text{ から } j \text{ へ資源 } k \text{ を流すことが可能} \\ 0 & \text{それ以外} \end{cases}$$

ただし、 $(i, j) \notin E$ ならば、 e_{ij}^k は常に0とする。

有向リンクは、その定義から少なくとも式(4.1)と式(4.2)を満足する。そこで有向リンクを対象として資源を割当てれば、少なくともトリップの時間順序と開始/終了場所のつながりに矛盾のないスケジュールを得ることができる。すなわち、列車ダイヤからの要求を満足するという意味でスケジュールの実行可能性を保障することができる。

しかし、例えば車両形式の違いで実際には割当てられない列車があるといった資源の運用制約が存在するため、資源によっては割当てられないトリップがある。そこで e_{ij}^k を用いて資源毎に割当て可否を定義する。

y_j : ノード i に対する資源の割当て不足数を表す決定変数。 ($i \in V$).

0以上の離散値を取る。

d_i : ノード i が要求する資源数。 ($i \in V$).

各 d_i は1以上、すなわち $d_i \geq 1$ である。

d_i は1の場合が多い。ただし、列車によっては輸送力を増やすために複数の車両を連結したり、作業量が多いため複数の乗務員を必要とするものがある。その場合、 d_i は必要な資源(車両、乗務員)の数に等しくなる。

b_{ij}^k : 資源 k の元のスケジュールのフロー表現。 ($k \in R, i, j \in V$).

元のスケジュールにおいて資源 k をトリップ i とトリップ j にこの順序で割当てるならば1、それ以外は0とする。

c_{ij}^k : 資源 k をリンク (i, j) に流すコスト。 ($k \in R, i, j \in V$).

元のスケジュールからの変更量を評価するために用いる。具体的な定義は後述する。

$c(X^k)$: 資源 k のスケジュールのコストを計算する関数。 ($k \in R$).

$c(X)$: 全てのスケジュールのコストを計算する関数。

$c(\mathbf{X}^k)$ は、他の資源のスケジュールとは独立に計算可能な関数とする。例えば、食事時間の確保度合い、すなわち労働条件を遵守する観点から適切な時間帯に適切な長さの食事休憩を確保しているかはスケジュール毎に計算可能である。一方、複数の資源のスケジュールが関係する評価指標は $c(\mathbf{X})$ を用いて定義する。例えば「駅での折り返し時間のばらつき」をコストとする場合が該当する。

整数計画問題

以上の準備の下、本問題は以下に示す整数計画問題として定式化できる。

Minimize

$$w_1 c(\mathbf{X}) + w_2 \sum_{k=1}^m c(\mathbf{X}^k) + w_3 \sum_{k=1}^m \sum_{(i,j) \in E} c_{ij}^k x_{ij}^k + w_4 \sum_{i=1}^n y_i \quad (4.3)$$

Subject to

$$\sum_{(1,j) \in E} x_{1j}^k = 1 \quad \forall k \in R \quad (4.4)$$

$$\sum_{(i,n) \in E} x_{in}^k = 1 \quad \forall k \in R \quad (4.5)$$

$$x_{ij}^k \leq e_{ij}^k \quad \forall (i,j) \in E, \forall k \in R \quad (4.6)$$

$$\sum_{(q,i) \in E} x_{qi}^k = \sum_{(i,q) \in E} x_{iq}^k \quad \forall i \in V \setminus \{1, n\}, \forall k \in R \quad (4.7)$$

$$\sum_{k=1}^m \sum_{(q,i) \in E} x_{qi}^k + y_i = d_i \quad \forall i \in V \setminus \{1, n\} \quad (4.8)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i,j) \in E, \forall k \in R \quad (4.9)$$

$$y_i \in \{0, \dots, d_i\} \quad \forall i \in N \setminus \{1, n\} \quad (4.10)$$

ここで、目的関数 (4.3) の w_1, w_2, w_3, w_4 は非負の重み定数である。

上記の各制約式の意味は次のとおりである。

(式 4.4) : 各資源のフローは開始ノード (1) から始まる。

(式 4.5) : 各資源のフローは終了ノード (n) で終わる。

(式 4.6) : 各資源は有向リンクの中でその資源を割当て可能なものの上のみを流れる。

(式 4.7) : 開始及び終了以外の各ノードではフロー量は保存される。

(式 4.8) : 開始及び終了以外の各ノードでは、割当てた資源の数と不足数との合計が資源の要求量に等しい。

(式 4.9) : 各資源の各ノードにおけるフローは0か1の離散値を取る。

(式 4.10) : 各ノードにおける資源不足数は0以上 d_i 以下の離散値を取る。

式 (4.4) ~ 式 (4.7) 及び式 (4.9) ~ 式 (4.10) は、各資源に対して独立に課される制約条件である。一方、式 (4.8) は全ての資源にまたがる制約条件である。

なお、資源が乗務員の場合は、業務を行わず目的地に移動するだけの便乗を許すことが多い。その場合は、トリップに要求量を超えて複数の乗務員を割当てられるので、式 (4.8) の等号を不等号で置き換えた下記を制約式とする。

$$\sum_{k=1}^m \sum_{(q,i) \in E} x_{qi}^k + y_i \geq d_i \quad \forall i \in V \setminus \{1, n\}, k \in R \quad (4.11)$$

目的関数である式 (4.3) の項 $\sum_{k=1}^m \sum_{(i,j) \in E} c_{ij}^k x_{ij}^k$ を、ここではコスト関数 (A) と呼ぶこととする。(A) は資源が流れる有向リンクのコスト c_{ij}^k の総和である。元のスケジュールからの変更量を最小化することを目的として、コスト c_{ij}^k の定義を以下とする。

$$c_{ij}^k = \begin{cases} \alpha & \text{if } b_{ij}^k = 1, j \neq n \\ \beta & \text{otherwise} \end{cases} \quad \alpha, \beta \geq 0 \quad \text{and} \quad \alpha \ll \beta \quad (4.12)$$

この定義の意図するところは、資源 k の再作成されたスケジュールと元のスケジュールとを比較して、異なる部分に大きなコストを与える、というものである。特に $\alpha = 0$ 、 $\beta = 1$ とすれば、コスト関数 (A) は再作成されたスケジュールの中で元のスケジュールに含まれないフローの数を表す。

図 4.2 は 2 つの資源のスケジュールを再作成する例である。スケジュール A に違反が発生し、2 つのトリップを含む 2 つの部分スケジュールをスケジュール B と交換することで新しいスケジュールを作成する。この場合、資源 A の新しいスケジュールの中で元のスケジュールに含まれないフローは c と f の 2 個である。一方、資源 B の新しいスケジュールの中で元のスケジュールに含まれないフローは d と e の 2 個である。したがって、 $\alpha = 0$ 、 $\beta = 1$ とすれば、コスト関数 (A) の値は元のスケジュールに含まれないフロー c, d, e, f の数に等しく 4 となる。

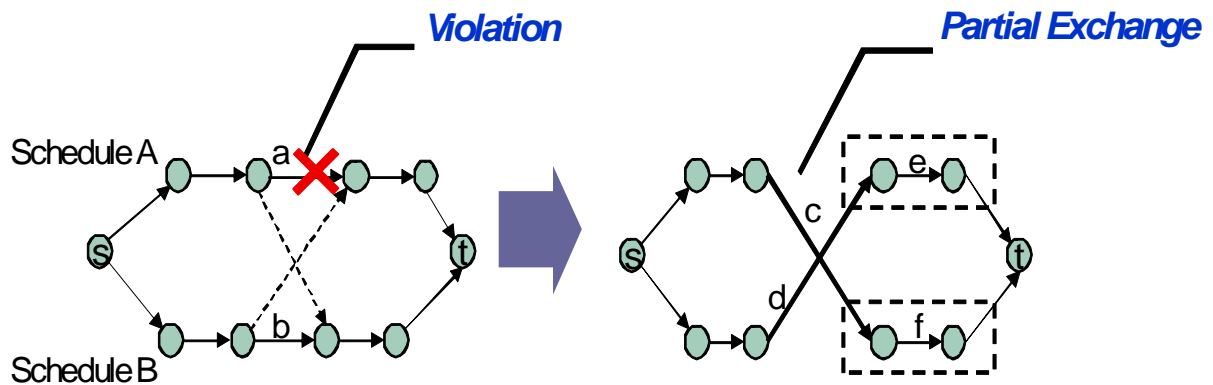


図 4.2: スケジュールの部分交換
Fig.4.2: Partial Schedule Exchange

4.2.3 定式化の特徴

VRP/CRP に良く似ており、事前スケジュールの作成問題である VSP/CSP は、集合分割モデルを用いて以下に示す 0-1 整数計画問題として定式化できる。

[Minimize]

$$\sum_{j \in N} c_j x_j \quad (4.13)$$

[Subject to]

$$\sum_{j=1}^n a_{ij} x_j = 1 \quad \forall i \in T \quad (4.14)$$

$$x_j \in \{0, 1\} \quad (4.15)$$

ここで、 N は行路集合、 j は N の要素、 T はトリップ集合、 i は T の要素、 c_j は行路 j のコストである。なお、トリップはノード集合 V の通常ノードと一対一に対応するが、 V はダミーノードも含むことから、トリップ集合とノード集合とを区別する意味でトリップ集合に対して別の記号 (T) を与えた。また、 a_{ij} は行路 j がトリップ i を含む場合に 1 を、それ以外の場合に 0 を取る定数、 x_j は行路 j を選択する場合に 1 を、それ以外の場合に 0 を取る決定変数である。

一方、集合被覆モデルを用いて定式化する場合は、式 (4.14) に代えて以下を制約式とする。

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad \forall i \in T \quad (4.16)$$

以下では、これらの集合分割 / 被覆モデルに基づく定式化との比較も交えながら、前節で定義したネットワークフローモデルに基づく定式化の特徴を述べる。

資源の扱い

設備投資や延伸等、各種施策の評価・分析のためにスケジュールを作成する際は、資源数を可変とし、資源の導入コストの観点から資源数の最小化を目的とすることが多い。その目的に対しては集合分割 / 被覆モデルの定式化が適している。すなわち、値が 1 である変数 x_j の数が選択した行路の数に等しく、即ちそれが資源数となるので、コスト c_j を 1 とすれば目的関数が資源数の最小化を意味する。

このように集合分割 / 被覆モデルの定式化において資源数は可変であることから、極端な場合は各トリップに対して全て異なる資源を割当てることができる。そのような解は容易に得られ、しかもそれが実行可能解のひとつとなる。

一方、列車運行乱れ時にスケジュールを再作成する場合、資源数の最小化は目的とはならない。そこで本定式化においては、資源は集合 R の要素として固定的に与えられ、それらの資源をトリップに適切に割当てることが必要である。したがって資源を任意に増やすことができないため、集合分割 / 被覆による定式化と比較して実行可能解を得ることが難しいと推察される。

元のスケジュールからの変更

集合分割 / 被覆モデルによる定式化は、個々の資源を他と区別しない。例えば、集合分割 / 被覆モデルによる定式化の解に q 個のスケジュールが含まれていたとすると、そこから q 個の資源が必要なのはわかるが、スケジュールと資源との対応関係、すなわち各資源のスケジュールが元のスケジュールからどう変化したかを知ることはできない。そのため、集合分割 / 被覆モデルによる定式化では、CRP/VRP にとって最も重要な評価指標である元のスケジュールからの変更量を扱うことが本質的に難しいと考えられる。

実行可能性の判定

本定式化における決定変数 y_i は、資源の割当不足数を表す。現実の問題は必ずしも実行可能解が存在することは保証されておらず、列車ダイヤや利用可能な資源の数などの前提条件次第では、全てのノードに対して必要な資源を漏れなく割り当てることができない場合がある。そこで、本定式化は資源割当の不足を定式化に陽に組み込み、割当不足の解を許容する。これにより資源割当が不足するノードを特定できるので、それを実行可能解が得られない原因として示すことができる。また、割当不足があれば前提条件に調整の余地があることがわかることから、これを利用することで、資源の追加といった前提

条件の調整をしながらスケジュールの再作成を繰り返すインタラクティブなスケジューリング機能を実現することが容易になると考えられる。

目的関数

集合分割 / 被覆モデルによる定式化の目的関数は、一般的に線形式のみである。一方、ネットワークフローモデルによる定式化は、線形式だけでなく非線形な目的関数も扱うことができる。

非線形な目的関数の具体例としては、「駅での折り返し時間にばらつきが少ない」、「走行距離の予定との差が少ない」が挙げられる。これら以外にも、対象路線や鉄道事業者の違い等に起因する様々な複雑な評価指標が存在する。例えば、鉄道事業者によっては行路の変更方法に「段落とし」というある種の定石があり、スケジュールがその定石にできるだけ準拠することが求められる。この場合は、定石に含まれる計画作成ノウハウがスケジュールにどれだけ反映されたか、すなわちノウハウの反映度合いが評価指標となるが、複雑なノウハウであっても任意の非線形式を用いれば定式化可能と考えられる。さらに、スケジュールの変更がもたらす個々の乗務員の心理状態や翌日の作業への影響といった、曖昧さや確率的な要素を含む評価指標も考えられる。本研究ではそうした評価指標も非線形な目的関数として扱うことを想定する。

4.3 運用計画変更コストに着目した最適スケジューリング方式

本節では、CRP/VRPにおいて最も重要な評価指標である「元のスケジュールからの変更量」に着目し、その最適化をおこなうための緩和問題を利用したスケジューリング方式を検討する。以下では本提案方式を「スケジューリング方式1」と呼ぶ。

4.3.1 基本アイデア

本提案モデルによる定式化の目的関数（式(4.3)）をあらためて以下に示す。

Minimize

$$w_1 c(X) + w_2 \sum_{k=1}^m c(X^k) + w_3 \sum_{k=1}^m \sum_{(i,j) \in E} c_{ij}^k x_{ij}^k + w_4 \sum_{i=1}^n y_i$$

目的関数は4つの項の線形加重和である。ここで、第4項はスケジュールの実行可能性を判断するための特別な目的関数であり、少なくとも実行可能解を得るためには必ず必要である。

一方、その他の3つの項はそれぞれ異なる評価指標を表している。この中で、本問題において最も重要な評価指標は、第3項の「元のスケジュールからの変更量」である。複数の評価指標がある場合は、評価指標を表す全ての項をこのように線形加重和等で結合し、ひとつにまとめて扱うことが一般的である。ただし、複数の評価指標のバランスを重み付けで調整することは本質的に難しい作業である。そのため、複数の評価指標を重み付けで扱うことを前提としてスケジューリング方式を設計すると、どのように重み付けをしても第3項を高度に最小化することが難しい可能性がある。

そこで本節では、目的関数の第1項と第2項の重み定数 w_1 と w_2 をどちらも0にすることで、評価指標を「元のスケジュールからの変更量」、すなわち運用計画の変更コストだけとし、この単一指標の下での最適スケジューリング方式を設計する。

目的関数において、元のスケジュールからの変更量を表す第3項と、資源の割当不足数を表す第4項は、どちらも線形式である。よって、第1項と第2項の重み定数を0にすれば、目的関数を線形式にすることができる。目的関数が線形であれば、連続変数緩和やラグランジュ緩和を施すことで緩和問題を構成しやすい。緩和問題の解は原問題の最適値の下界値を与える。したがって、その情報を利用することによりスケジューリング方式設計の定石的なアプローチを適用することができる。例えば下界値を用いて探索木を枝狩りする分枝限定法や、最適解が存在する方向を推定し、そこへ向かって探索を進

めていくヒューリスティクス解法等が考えられる。そこで、そうした緩和問題を利用した定石的なアプローチのひとつであるラグランジュ緩和法を適用しスケジューリング方式を設計する。

なお、「元のスケジュールからの変更量」に対応する目的関数の第3項は、コスト定義を変更すれば、各資源のスケジュール毎の評価指標、すなわち本来は第2項で扱うべきもののいくつかを扱うこともできる。それらの例として「予備車両の使用回数」、「最終到着地の変更回数」が挙げられる。「予備車両の使用回数」を扱う場合は、資源 k が予備車両ならば c_{1n}^k を1とし、残りの全ての c_{ij}^k を0とすれば良い。一方、「最終到着地の変更回数」を扱う場合は、あるトリップ i の到着地が資源 k の元のスケジュールの最終到着地と同じであれば $c_{in}^k = 0$ 、そうでなければ $c_{in}^k = 1$ とし、残りの全ての c_{ij}^k を0とすれば良い。

4.3.2 ラグランジュ緩和問題

ここで扱う目的関数は、式 (4.3) に含まれる第3項と第4項からなる以下の式である。

Minimize

$$\sum_{k=1}^m \sum_{(i,j) \in E} c_{ij}^k x_{ij}^k + w \sum_{i=1}^n y_i \quad (4.17)$$

ここで、単純化のために重み定数 (w_3, w_4) を第4項にまとめ、あらためて w とする。

一方、制約条件は元の定式化と全く同一である (式 (4.4) ~ 式 (4.10))。

この定式化に対し、制約条件のひとつである式 (4.8) を、ラグランジュ乗数を用いて目的関数に組込むことでラグランジュ緩和問題を構成する。

Minimize

$$\begin{aligned} & \sum_{k=1}^m \sum_{(i,j) \in E} c_{ij}^k x_{ij}^k + w \sum_{i=1}^n y_i \\ & + \sum_{i=2}^{n-1} \lambda_i \left\{ d_i - \sum_{k=1}^m \sum_{(q,i) \in E} x_{qi}^k - y_i \right\} \end{aligned} \quad (4.18)$$

Subject to

式 (4.4)、式 (4.5)、式 (4.6)、式 (4.7)、式 (4.9)、式 (4.10)

この式のアンダーラインを付した項が、制約条件である式 (4.8) を組み込んだ部分である。以降では、元の定式化を主問題 P 、このようにして構成したラグランジュ緩和問題を $P(\lambda)$ とする。なお、便乗がある場合、制約条件である式 (4.8) は式 (4.11) で置き換えられるが、その際には乗数 λ_{ij} は非負、すなわち $\lambda_{ij} \geq 0$ となる。

さらにこの式は、適当な式変形を施すことにより以下に示す形式となる。

Minimize

$$\underbrace{\sum_{k=1}^m \left\{ \sum_{i=1}^n \sum_{(q,i) \in E} (c_{qi}^k - \lambda_i) x_{qi}^k \right\}}_{(4.19)} + \sum_{i=2}^{n-1} (w - \lambda_i) y_i + \sum_{i=2}^{n-1} d_i \lambda_i$$

この式の第3項は定数項なので、問題を解く上では無視して構わない。アンダーラインを付した第1項の係数 $(c_{qi}^k - \lambda_i)$ を、コスト c_{qi}^k とラグランジュ乗数 λ_i による新たなコスト（調整コスト \hat{c}_{qi}^k ）とみなせば、本項は資源 k のスケジュールに対する調整コストの総和を表している。そして、第2項の係数 $(w - \lambda_i)$ を、コスト w とラグランジュ乗数 λ_i による新たなコスト（調整コスト \hat{w}_i ）とみなせば、本項は資源の不足に対する調整コストの総和を表している。

式 (4.8) を緩和したので、ラグランジュ緩和問題 $P(\lambda)$ の制約式の中に複数の資源に跨るものは存在しない。したがって、 $P(\lambda)$ は次に示す資源 k に関する m 個の部分問題 $P(\lambda, k)$

Minimize

$$\sum_{i=1}^n \sum_{(q,i) \in E} \hat{c}_{qi}^k x_{qi}^k \quad (4.20)$$

Subject to

$$\sum_{(1,j) \in E} x_{1j}^k = 1 \quad (4.21)$$

$$\sum_{(i,n) \in E} x_{in}^k = 1 \quad (4.22)$$

$$x_{ij}^k \leq e_{ij}^k \quad \forall (i,j) \in E \quad (4.23)$$

$$\sum_{(q,i) \in E} x_{qi}^k = \sum_{(i,q) \in E} x_{iq}^k \quad \forall i \in V \setminus \{1,n\} \quad (4.24)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i,j) \in E \quad (4.25)$$

及び、変数 y に関する部分問題 $P(\lambda, y)$

Minimize

$$\sum_{i=2}^{n-1} \hat{w}_i y_i \quad (4.26)$$

Subject to

$$y_i \in \{0, \dots, d_i\} \quad \forall i \in N \setminus \{1, n\} \quad (4.27)$$

とに分割し、それぞれを独立に解くことができる。さらに、定式化の特徴から、部分問題 $P(\lambda, k)$ は、有向リンク (i, j) の長さが c_{ij}^k で与えられるグラフ上で、開始ノード s から終了ノード t までの最短経路を求める問題に帰着される。

4.3.3 ラグランジュ緩和法

ラグランジュ緩和問題において、乗数 λ を固定して得られる最適解は主問題の下界値を与えることが理論的に保証されている。最大の下界値を与えるラグランジュ乗数を求めることができれば、得られる緩和問題の解が主問題の最適解に一致すると期待される。ラグランジュ緩和法は、ラグランジュ乗数を修正しながら緩和問題を繰り返し解くことで、最大下界値を反復的に求めるものである [77]。

ただし、対象とする主問題によっては最大下界値が最適値に一致しない、いわゆる双対ギャップが存在する場合や、両者を一致させるために多くの反復回数を必要とする場合がある。あるいは最適値に一致しても、緩和問題の解が主問題の制約条件を必ずしも満足しないことが起こりうる。そこでラグランジュ緩和法では、反復の度に何らかの手段で主問題の実行可能解をラグランジュ乗数と関連付けて生成し、反復終了後に最良解を近似最適解として出力する。

以降では、本提案方式における下界値の算出法と実行可能解の生成方法について述べる。

4.3.4 下界値算出方式

資源 k に関する m 個の問題 $P(\lambda, k)$ と変数 y に関する問題 $P(\lambda, y)$ とを独立に解き、それらの目的関数値と定数 $\sum_{i=2}^{n-1} d_i \lambda_i$ との合計を求める。これがラグランジュ緩和問題の最適値、すなわち主問題の下界値となる。

$P(\lambda, y)$ は実質的に制約条件を持たないため、最適解を簡単に求めることができる。すなわち、ノード i が $\hat{w}_i \geq 0$ を満たすならば $y_i = 0$ 、そうでなければ $y_i = d_i$ とすれば良い。

一方、 $P(\lambda, k)$ の最適解は s から t までの最短経路問題を解くことで得られる。その際、制約式 (4.23) を満足するため、 $e_{ij}^k = 0$ を満たす有向リンク (i, j) をグラフから削除する。

最短経路探索の具体的な手法としてはダイクストラ法が良く知られている [78]。ダイクストラ法の時間計算量は $O((|V| + |E|) \log V)$ であり、計算効率が高い。しかしながら、適用可能なグラフの条件として全てのリンクの長さが正でなければならぬという制限がある。リンク長にあたる調整コスト \hat{c}_{qi}^k は、その定義 $\hat{c}_{qi}^k \equiv c_{qi}^k - \lambda_i$ より、乗数 λ_i の値次第では負の値を取りうる。したがって、 $P(\lambda, k)$ に対する最短経路探索にダイクストラ法を用いることはできない。

長さが負のリンクを含むグラフに対して適用可能な一般的な最短経路探索アルゴリズムとしては、ベルマン・フォード法が挙げられる [79]。これは時間計算量が $O(|V||E|)$ であり、ダイクストラ法と比較すると問題規模が大きくなるにつれ計算効率が大幅に悪化

する。本提案方式においては、ラグランジュ緩和法の反復毎に資源数 m と同じ回数の最短経路探索を実行しなければならない。問題の規模が大きくなると必然的に m も大きくなるので、提案方式を実用的なものとするためには、最短経路をできるだけ高速に探索する手法が必要となる。

本問題の場合、最短経路を求める対象のグラフが有向非巡回グラフであることから、*shortest path in DAG algorithm*[80] (以下、DAG アルゴリズムと呼ぶ) を用いて最短経路を効率良く求めることができる。このアルゴリズムは、緩和 (relax) と呼ばれる最短経路長の推定値の更新処理を、グラフの各頂点に対して位相的順序 (トポロジカルオーダー) でおこなうものである。ここで位相的順序とは、ノード i, j の間に有向リンク (i, j) があるとき $i < j$ とする順序である。時間計算量は、頂点を位相順に並びかえるトポロジカルソートが $O(|V| + |E|)$ 、最短経路長の更新が頂点 V に関して線形、すなわち $O(|V|)$ である。トポロジカルソートは最初に一度だけ実行すれば良いので、最短経路探索は実質的に $O(|V|)$ の計算量で実行できる。

4.3.5 実行可能解生成方式

最短経路探索をサブルーチンに持つ以下の手続き (図 4.3) により実行可能解を生成する。

本手続きは Step2 において最短経路探索をおこなう。その具体的な手法は下界値算出方式の場合と同じであり、前述の DAG アルゴリズムを用いる。

Step3 におけるグラフ構造の更新は、グラフの各ノードが最短経路に含まれた回数を数え、それが資源要求量 d_i に一致したノードと、そのノードに接続する全てのリンクをグラフから削除する。これにより制約式 (4.23) を満たすことができる。

Step 4 における未割当ノードの追加は、最短経路に含まれるノードを開始ノードから順に辿り、隣接する 2 つのノードを結び、かつ未割当ノードを通る代替経路を探索する (図 4.4)。代替経路があれば、それをあらためて経路の一部とする。こうすることにより

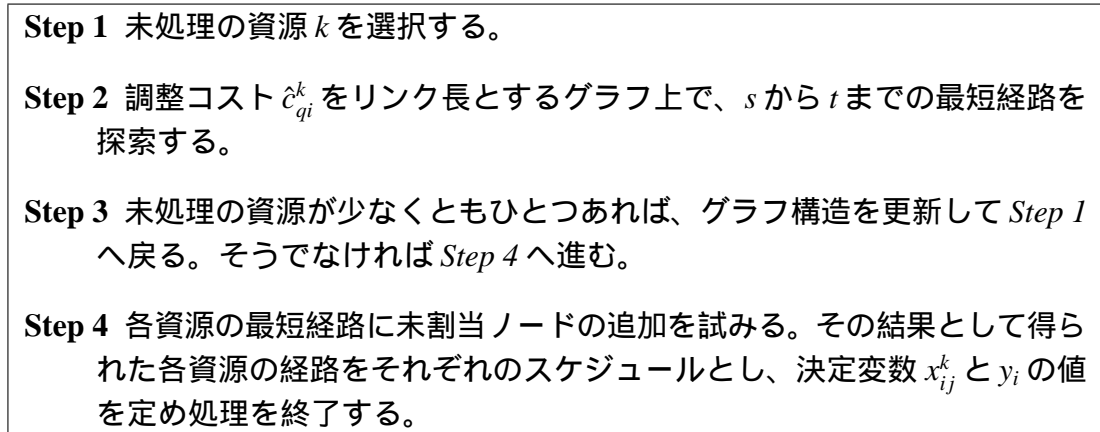


図 4.3: 実行可能解生成手続き
Fig.4.3: Procedure for Feasible Solutions

資源の割当不足を抑えた実行可能解を得ることができる。

なお、乗務員の場合、Step4でノードの追加を行っても未割当ノードが残れば、各ノードの割当数の制限 (d_i) を無視してノードの追加を再度実行する。こうすることにより、便乗をできるだけ少なくしつつ各ノードに必要な数以上の乗務員を割当てることができる。

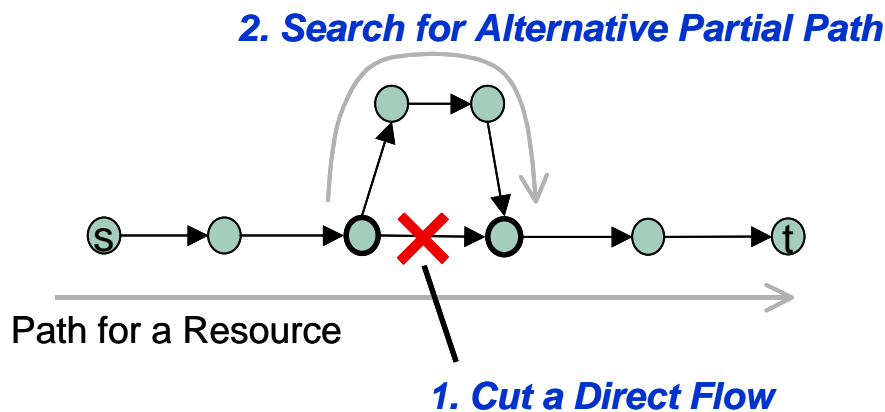


図 4.4: 未割当ノードの追加
Fig.4.4: Insertion of nodes to a Path

決定変数 x_{ij}^k は、リンク (i, j) が資源 k の経路に含まれれば 1、そうでなければ 0 とする。
決定変数 y_i の値は、決定変数 x_{ij} の値を制約式 (4.10) に代入して求める。

4.3.6 提案方式の特徴

以上、スケジューリング方式1について説明した。ラグランジュ緩和法を用いた本提案方式の特徴をあらためて以下に整理する。

下界値算出法 負のリンクを持つグラフの最短経路を、DAG アルゴリズムを用いて探索する。負のリンクを持つグラフに対する一般的な最短経路探索手法としてベルマン・フォード法が知られているが、対象グラフが持つ性質、すなわち有向非巡回グラフである点に着目し、上記アルゴリズムを適用することで計算効率化を図った。

上界値及び実行可能解算出法 ラグランジュ乗数で調整したリンク長を持つグラフ上で各資源の最短経路を探索する。最短経路をひとつ計算する度に、ノードの選択回数が資源の要求量を超えないようにグラフを更新することで制約違反の発生を防ぐ。また、最短経路探索で選択されなかった未割当ノードを、制約違反が発生しないように最後に経路に追加し、それを各資源のスケジュールとする(図4.4)。これにより、資源の割当不足をできるだけ抑えた実行可能解を得ることができる。なお、最短経路は下界値の場合と同様に DAG アルゴリズムを用いて効率良く探索をおこなう。

4.4 運用資源制約と運用計画変更コストに着目した最適スケジューリング方式

本節では、本問題の最も重要な評価指標である「元のスケジュールからの変更量」、すなわち運用計画の変更コストだけでなく、作業効率等、資源の運用に関する様々な評価指標も同時に扱うことが可能な最適スケジューリング方式を検討する。以下では、本提案方式を「スケジューリング方式2」と呼ぶ。

4.4.1 基本アイデア

本定式化の目的関数（式(4.3)）は、4種類のコストの線形加重和である。この目的関数をあらためて以下に示す。

Minimize

$$w_1 c(\mathbf{X}) + w_2 \sum_{k=1}^m c(\mathbf{X}^k) + w_3 \sum_{k=1}^m \sum_{(i,j) \in E} c_{ij}^k x_{ij}^k + w_4 \sum_{i=1}^n y_i$$

このうち、第1項と第2項の目的関数は、対象路線や鉄道事業者の違い、あるいは車両と乗務員の違い等に応じて様々な評価指標を扱う必要があることから、非線形式となり得る。

一方、制約条件はオリジナルの定式化と全く同じである。

Subject to

式(4.4)、式(4.5)、式(4.6)、式(4.7)、式(4.8)、式(4.9)、式(4.10)

目的関数が非線形式を含む場合は、暫定解の評価と更新を繰り返しおこなうことで最適解に接近する生成検査法のアプローチを適用してスケジューリング方式を設計することが考えられる。

生成検査法は、(1)何らかの方法で初期解を生成して、これを暫定解とし、(2)暫定解の評価と、評価に基づく暫定解の更新とを繰り返すことで暫定解を最適解に近づけていくものである。

図4.5に生成検査法に基づくスケジュール再作成の実行イメージを示す。まず(1)として、スケジュールに生じた制約条件の違反を、実際のスケジュール作成者の手法に倣ったヒューリスティックな方法で修正することで初期解を生成する。次に(2)として、探索途中で得られた暫定解を保持しながら暫定解の更新を繰り返すことで代替案を探索する。この際の暫定解の更新も(1)と同様にヒューリスティックスを用いる。これにより、探索

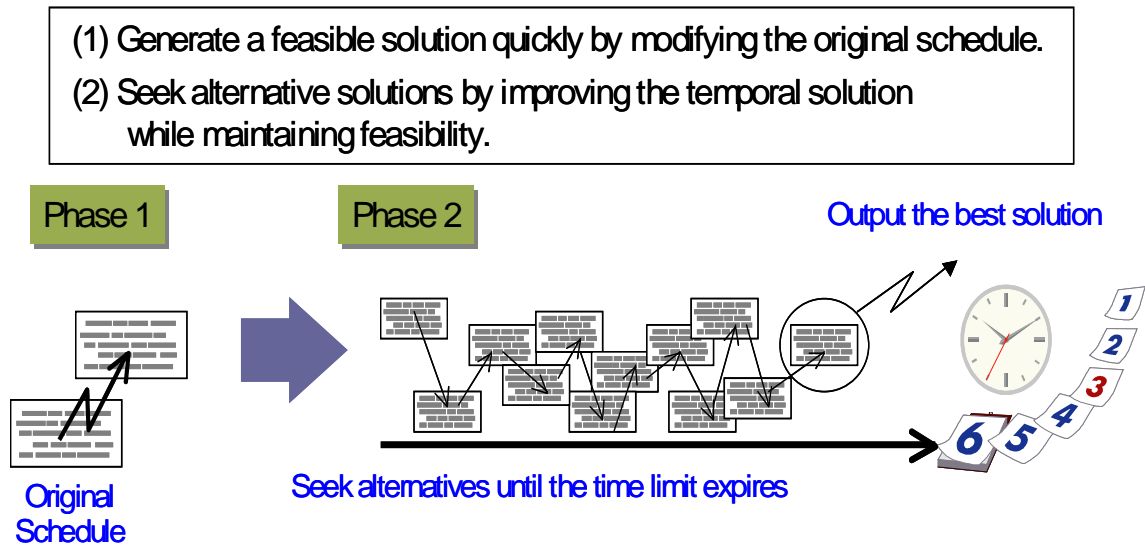


図 4.5: 生成検査法に基づくスケジュール再作成の実行イメージ
 Fig.4.5: Rescheduling Based on Generate and Test Method

のどの時点においても最良の暫定解を出力できるようにしながら、スケジューリングの時間的な余裕に応じて目的関数を漸進的に最適化することが期待できる。

生成検査法は問題解決の一般概念であるが、これを実現するための具体的な手法として、メタヒューリスティクス [96] と総称されるものがある。メタヒューリスティクスは、解の探索に関わるマクロな発見的知識であるヒューリスティクスを適応的に用いることを特徴とする各種の近似解法の総称である。遺伝的アルゴリズム (Genetic Algorithm,GA)、シミュレーテッド・アニーリング (Simulated Annealing,SA)、タブーサーチ (Tabu Search)、アントコロニー最適化 [97] 等の手法がある。適用にあたっての厳密な定式化を必ずしも必要とせず、計算機への実装も比較的容易であることが応用上の大きなメリットであり、プロトタイピングから実用システムの開発まで幅広く活用されている。特に我が国においては GA を用いた実システムの構築が非常に盛んであり、上水道プラントの送水計画、バス乗務員の仕業ダイヤ作成、航空機スケジューリング等、様々な分野において実適用の事例がある。

以上の考察より、スケジューリング方式2の設計にメタヒューリスティクスを用いることとする。適用する手法は、アプローチの原理的な有効性を確認することを目的として、メタヒューリスティクスの中で最もシンプルな局所探索法を採用する。

4.4.2 初期解生成法

ここでは、列車運行の乱れと列車ダイヤの変更のためにグラフ構造が変化し、制約条件に違反が発生する場合を想定する。これに対する初期解生成法として、ヒューリスティクスを用いたスケジュール修正方法を定義する。違反を生じる制約条件は、具体的には式(4.6)である。すなわち、それまで存在していたグラフ上のある有向リンクが、列車の到着時刻の遅れや出発時刻の変更等の影響を受けて消滅し、そのリンク上のフローが無効になったことを意味する。

このような場合、スケジュール作成担当者は一般的にスケジュールの一部を他と交換することで違反を修正しようとする。交換の例は図4.2に示したとおりである。この初期解生成法は、スケジュールの部分的な交換をおこなうヒューリスティクスを用いてネットワークフローを修正することで初期解を作成する。その具体的な処理は以下に示すとおりである。

Step1 制約条件に違反するフローの中で時刻が最も早いものを選択する(フローaとする)。

Step2 他の資源のスケジュールからフローをひとつ選択する(フローbとする)。

Step3 フローa以降の部分スケジュールとフローb以降の部分スケジュールとを交換する(図4.2)。

Step4 未処理の違反フローがあればStep1に戻る。全ての違反フローを処理したら違反の有無をチェックする。違反がない場合は実行可能解が得られたので全体処理を終了する。違反がある場合は、規定の試行回数を超えていなければ違反フローを全て未処理に設定した上で再びStep1から実行する。規定の試行回数を超えていれば全体

処理を終了する。この場合は違反が残るので実行可能解は得られない。そこで、違反フローがある資源のスケジュールを違反フローの始点ノードで打ち切り、それ以降のスケジュールに含まれるノード i の変数 y_i (割当て不足数) の値を 1 にする。これにより実行可能ではないが定式化を満たす解を強制的に生成する。

Step2 で選択するフロー b は、スケジュールの部分的な交換によりフロー a の違反が解消し、なおかつ新たな違反が発生しないものとする。この条件を満たす交換を特に通常交換と呼ぶ。交換によってフロー a はなくなり、その代わりにフロー a の始点ノードから伸びる新たなフローが生成される (図 4.2 のフロー c)。この条件を満たす候補が複数ある場合は、新たなフローの時間的長さが最も短いものを選択する。通常交換は常に可能であるとは限らない。そこで、それが不可能な場合はフローの選択条件を緩和し、交換相手のフローに新たな違反が生じることを許容してフローを選択する。これを強制交換と呼ぶ。複数の候補がある場合の選択基準は通常交換と同じだが、交換の履歴を参照して違反状態が元に戻る候補は除外する。

強制交換も不可能な場合は、違反フロー (フロー a) を切断してスケジュールを 2 つに分割し、予備、すなわちスケジュールが空の資源にその一方を割り当てる。予備がない場合は、任意の 2 つの資源のスケジュールを結合することで予備の生成を試みる。

なお、列車ダイヤの変更により臨時列車が設定された場合は、上記の処理を実行する前に臨時列車をトリップに分割し、それらを既存のスケジュールの任意の箇所に挿入する。また、運休列車はあらかじめスケジュールから取り除いておく。これらの前処理でスケジュールの一部に制約条件違反を生じることになるが、発生するのはいずれも式 (4.6) の違反である。したがって、上記の初期解生成処理で元々の違反とあわせて違反解消がおこなわれる。

4.4.3 局所探索法

前節の方式で得られた実行可能解を初期解として、局所探索法 [84] を用いて代替案探索をおこなう。局所探索法は生成検査法の一つであり、元の解に少しの変形を加えて複数の解（近傍と呼ぶ）を生成し、その中から評価値の良いものを選んで再び変形・評価・選択を繰り返すことにより、評価値の高い解を段階的に得る手法である。

ここで近傍は、制約式 (4.6) の新たな違反が生じないスケジュールの部分交換で得られる解の集合である。全てのフローを対象として、交換する相手のフローを変えながら部分交換をおこなうことで近傍を生成する。

探索の停止件は、反復回数が一定値に到達するか、近傍の中に暫定解よりも評価値が良いものが存在しない場合とする。

4.5 数値実験

4.5.1 プロトタイプシステム

提案スケジューリング方式を組み込んだプロトタイプシステムを開発した。

図 4.6 にプロトタイプシステムの画面例を示す。本システムは、列車ダイヤと資源のスケジュールをダイヤ図と呼ばれる形式で表示する。すなわち、横軸が時刻、縦軸が駅の並びを意味する 2 次元平面上に、列車の走行予定（ダイヤ）を駅間を結ぶ線として表す。そして、駅での折り返し等、同一の資源を連続して割当ててる 2 つの走行予定の間を水平線で結ぶ。

本システムは、以下に示す 3 つの主要なソフトウェア機能から構成される。(a) グラフィカルユーザインターフェース (GUI)、(b) データベース及び通信ネットワーク、(c) 提案スケジューリング方式を実装したスケジューリング機能。GUI はいくつかのインタラクティブな機能を提供する。すなわち、システムがスケジュールの再作成結果とその評価値を、前述のダイヤ図やグラフ等を用いてグラフィカルに表示し、ユーザがそれを受けて再作成

の前提条件や各種パラメータの設定値を変更することができる。例えば列車の遅延時間、運休や臨時列車、再作成の時間範囲等である。また、ユーザはシステム画面上でマウスを用いてスケジュールを直接修正したり、スケジュールの一部を固定することもできる。

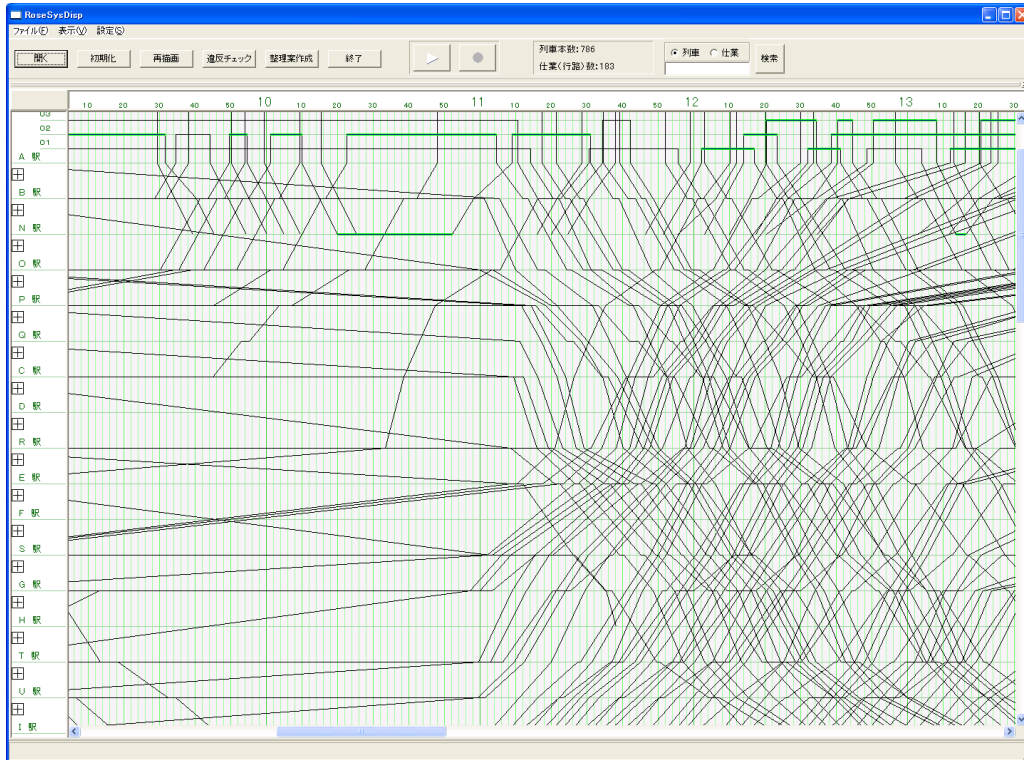


図 4.6: プロトタイプシステム
Fig.4.6: Prototype System

こうしたインタラクティブな機能の大部分は、提案したネットワークフローモデルを利用することで比較的容易に実現できる。すなわち、スケジュールの修正といった各種の変更の多くは、スケジュールを構成するトリップと資源の運用制約の変更とみなすことができるので、システム内部にネットワークフローモデルを保持し、その構造を変更する操作として上述のインタラクティブな機能を設計・実装することができる。

以下にインタラクティブ機能とモデル操作との対応例のいくつかを挙げる。

- 列車の時刻変更は、その列車に該当するノードの属性値である開始もしくは終了時刻の変更に対応する。また、時刻変更によって有向リンクに時刻の逆転が生じた場

合は、その有向リンクを削除する。

- 列車の運休は、その列車に該当するノードと、そのノードにつながる有向リンクの削除に対応する。
- スケジュールの修正は、そのスケジュールを表すネットワークフローの削除やつなぎ替えといった変更操作に対応する。システム画面上では、フローはダイヤ図上で2つの列車走行予定を結ぶ水平線として示される。
- スケジュールの固定は、資源の割当て可否を指定する定数 e_{ij}^k の修正に対応する。例えば、資源 a のスケジュールにおいて、有向リンク (i, j) 上のフローを固定したい場合は、 e_{ij}^a の値を1（割当可）とし、資源 a 以外の残り全ての資源 k について $e_{ij}^k = 0$ （割当不可）とすれば良い。

ネットワークフローモデルの代わりに集合分割 / 被覆モデルを用いるのであれば、集合分割 / 被覆モデルがスケジュールを限定的な形でしか扱えず、資源運用制約も表現できないことから、上記のインタラクティブな機能はスケジューリング機能と切り離して独立に設計・実装する必要がある。そのため、システムの構造が複雑になり、機能改良や機能拡張等が困難になる恐れがある。

4.5.2 実験条件

実験データ

実路線の運行乱れ時の運用データを用い、車両スケジューリングを対象とした実験をおこなった。実験用のデータは2種類用意した。ひとつは小規模路線を対象としたものである。対象路線の列車本数は142、車両数は11である。以降ではこれをデータ1と呼ぶ。もうひとつは大規模路線を対象としたものである。列車本数は786、車両数は185である。以降ではこれをデータ2と呼ぶ。

運行乱れの状況と列車ダイヤの変更内容は、どちらのデータも概要はほぼ同じである。朝のラッシュ時間帯において、ある列車が駅間を走行中に故障を発生して停止し、後続する複数の列車が大幅に遅延した、という運行乱れの下で、大幅に遅延したいくつかの列車を運休させることで回復を図るものである。

図4.7はデータ1の一部である。プロトタイプシステムの画面上に、列車ダイヤと車両の運用スケジュールをダイヤ図として表示している。

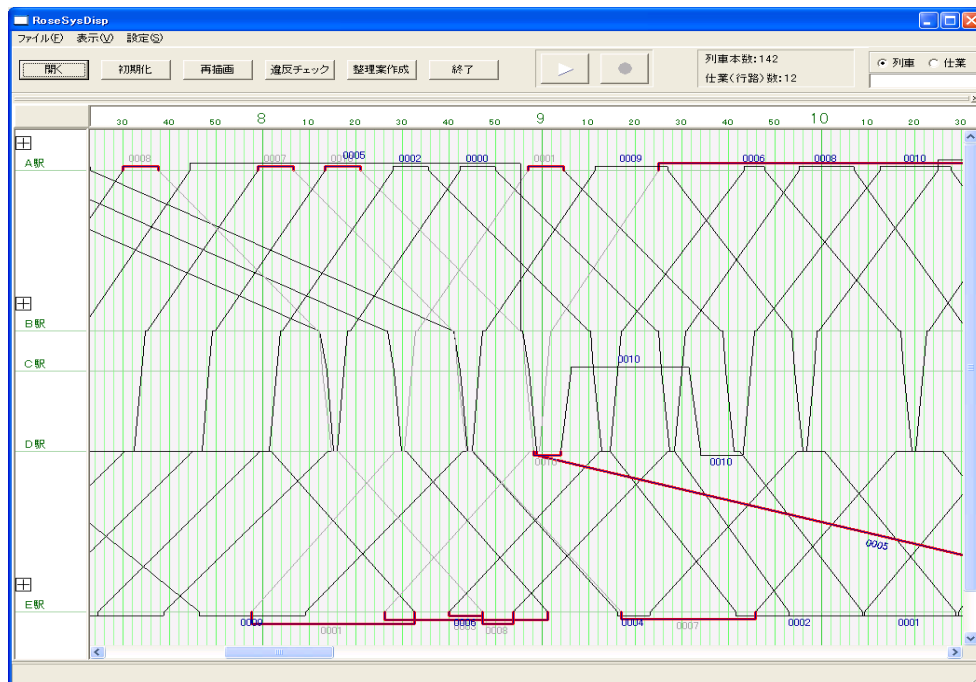


図 4.7: 実験データ 1 : 小規模鉄道路線

Fig.4.7: Experimental Data 1 : Small Railway Line

同様にデータ2の一部を図4.8に示す。

データ1と2はどちらも実際の事例であり、比較的大きな運行乱れを扱ったものである。

両者のネットワークモデルの規模を表4.1に示す。どちらのデータも、1本の列車を1個の独立したトリップ(スケジュールの構成単位)とした。したがってグラフのノード数は列車本数と一致する。

これらのデータに対して、列車ダイヤを変更した時点から当日の全列車の運行が終了

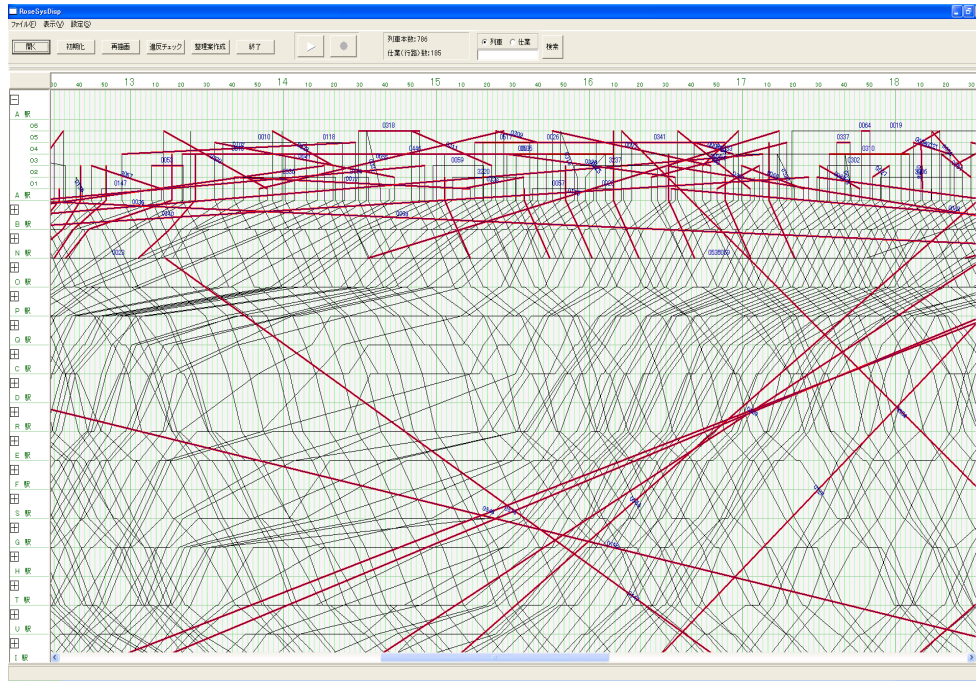


図 4.8: 実験データ 2 : 大規模鉄道路線

Fig.4.8: Experimental Data 2 : Large Railway Line

する時点までの時間幅でスケジュールの再作成をおこなう。再作成の時間幅は約 18 時間である。

表 4.1: ネットワークモデルの規模

Table4.1: Graph Size of Network Model

	Data 1	Data 2
Node	142	786
Link	3115	6212
Resource	11	185

目的関数

本問題の目的関数（式 (4.3)）をあらためて以下に示す。

$$w_1 c(\mathbf{X}) + w_2 \sum_{k=1}^m c(\mathbf{X}^k) + w_3 \sum_{k=1}^m \sum_{(i,j) \in E} c_{ij}^k x_{ij}^k + w_4 \sum_{i=1}^n y_i$$

この式において、最も重要な評価指標である「元のスケジュールからの変更量」に対応するのは第3項である。第3項のコスト c_{ij}^k の具体値を設定しやすくするため、第3項の重み定数 w_3 を1とした。この前提の下で、予備実験の結果もふまえて第3項のコスト c_{ij}^k の具体値を次のとおりとした。

$$c_{ij}^k = \begin{cases} 1 & \text{if } b_{ij}^r = 1, \exists r \in R \\ 10 & \text{otherwise} \end{cases}$$

元のスケジュールにおいて、ある資源 r がトリップ i と j に対してこの順番で割り当てられることになっていれば、 c_{ij}^k の値は1を取る。そうでなければそれよりも大きな値である10を取る。したがって、コスト c_{ij}^k の合計が大きなスケジュールであれば、元のスケジュールからの変更量が多いことを意味する。

また、第3項の重み w_3 は、他の項の重みの基準値とすることを目的として、 $w_3 = 1$ とした。

第1項と第2項は非線形式となり得ることから、提案スケジューリング方式1の場合は w_1 と w_2 のどちらも0とし、第1項と第2項を対象外とする。

一方、提案スケジューリング方式2の場合は、第1項と第2項をそれぞれ次の評価指標に対応させる。

- 第1項：全車両の駅での折り返し時間の標準偏差
- 第2項：各車両の走行距離の差分の総和

ここで、折り返し時間とは、車両のスケジュールを構成する隣り合う2つのトリップの間合い時間、すわなわち両者を結ぶフローの長さを意味する。なお、一般的に折り返しは終着駅に到着した列車を反対方向へ出発させることを意味するが、ここでは同一方向へ出発する場合も折り返しとみなして目的関数値の計算に含むものとする。

一方、走行距離の差分とは、スケジュール再作成の実施前後での走行距離の差の絶対値を意味する。車両保守においては、車両の走行距離の実績が定められた距離以下となる周期で定期点検を行うことが法律で定められている。一定期間の車両のスケジュールは事前に作成されており、これに基づいて将来の定期点検の実施日も決まっている。そのため、スケジュールの再作成によって走行距離が変化してしまうと定期点検実施日の変更が発生する可能性がある。そこで、第2項において各車両の走行距離の差分をできるだけ小さくすることをスケジュール再作成の目的のひとつとし、点検実施日の変更可能性をできるだけ減らすこととする。

第1項のコスト関数 $c(X)$ 、第2項のコスト関数 $c(X^k)$ の具体的な定義を以下に示す。

第1項

$$c(X) = \sqrt{\frac{S}{N-1}} \quad (4.28)$$

ここで、 $S = \sum_{k=1}^m \sum_{(i,j) \in E} (x_{ij}^k p_{ij} - \mu)^2$ (折返し時間の平均値からの差分の2乗和)、 $\mu = \sum_{k=1}^m \sum_{(i,j) \in E} x_{ij}^k p_{ij} / N$ (折返し時間の平均値)、 $N = \sum_{k=1}^m \sum_{(i,j) \in E} x_{ij}^k$ (駅での折返し回数)、 p_{ij} は有向リンク $(i, j) \in E$ の長さ、すなわち駅での折返し時間とする。

第2項

$$c(X^k) = |U^k - V^k| \quad (4.29)$$

ここで、 $U^k = \sum_{(i,j) \in E} x_{ij}^k q_i$ (車両 k のスケジュール再作成後の走行距離)、 $V^k = \sum_{(i,j) \in E} b_{ij}^k q_i$ (車両 k のスケジュール再作成前の走行距離)、 q_i はノード $i \in V$ が表すトリップの走行距離とする。

上記から明らかとなおり、どちらの項も非線形式である。また、第1項は全ての車両をまとめて計算するものであり、第2項は各車両毎の計算結果を足し合わせることで値が得られる。

各項の重み定数 (w_1, w_2) は、恣意的な設定を避けることを目的として、次の方法を用いて動的に定めることとした。

Step1 提案スケジューリング方式2の初期解の評価値を各項毎に算出する。第1,2,3項の値をそれぞれ v_1, v_2, v_3 ($v_3 \neq 0$) とする。

Step2 上記の値を用いて、第1項の重み定数 w_1 を $w_1 = \frac{v_3}{v_1}$ 、第2項の重み定数 w_2 を $w_2 = \frac{v_3}{v_2}$ とする。

この方法を用いれば、目的関数の各項の値の水準が第3項に揃えられるので、いずれかの評価指標に偏ることなく、それらを平等に扱いながら解探索をおこなうことが期待できる。

最後に、第4項は資源の割当不足に対するペナルティを意味する。第4項は実行可能解を得るためのものであることから、ペナルティを他の項と比較して非常に大きな値にする必要がある。そこで予備実験の結果も踏まえ、第4項の重み定数 w_4 を1000とした。

測定項目

2つの提案スケジューリング方式に共通する測定項目を以下とした。

- 探索結果：実行可能解（資源割当不足のない解）が得られたかどうか。
- 探索効率：最良解を出力するまでの処理時間。
- 探索品質：最良解の評価値の値。最も重要な評価指標である「元のスケジュールからの変更量」については精度評価もおこなう。この精度評価には、提案スケジューリング方式1を適用することで得られる最適値の下界の情報を利用する。

計算機環境

実験に使用した計算機は、CPU 3.0GHz、メモリ 3.5GB の一般的なPCである。

その他

例えばラグランジュ緩和法の下界値の推移やラグランジュ乗数の初期値等、2つの提案スケジューリング方式のいずれかに固有の測定項目や条件等の各種設定については、各提案方式の実験結果の節で個別に説明する。

4.5.3 実験結果：スケジューリング方式1

各種設定

ラグランジュ乗数 λ_i の初期値は全て0、各反復における更新には劣勾配 (Subgradient) の概念に基づく汎用的な手法である劣勾配法 (Subgradient Method)[81] を用いた。反復回数は、予備実験の結果に基づき、データ1に対しては100回、データ2に対しては1000回とした。

データ1: 小規模鉄道路線

上記の設定の下で、データ1に対して提案方式を実行した結果を図4.9に示す。

実験の結果、資源割当不足を含まない実行可能解を生成することができた。実行時間は4.73秒であった。これは十分実用的な実行時間であるといえる。

図4.9において、 LB は下界値である。すなわち、各反復で得られたラグランジュ緩和問題の最適値を意味する。また、 UB は上界値である。同様に各反復で得られた主問題の目的関数値を意味する。一方、 $Best\ LB$ と $Best\ UB$ は、探索のその時点の反復までに得られた全ての下界値と全ての上界値の中での最良値をそれぞれ表す。また、相対双対ギャップ (relative duality gap) は、最良上界値 f_{best} と最良下界値 ϕ_{best} とのギャップ $\Delta_{best} \equiv f_{best} - \phi_{best}$ を用いて、 $\frac{\Delta_{best}}{\phi_{best}} \times 100$ と定義される。

この結果より、収束の度合いを示す相対双対ギャップが約3%であることから、得られた最良解の評価値が、最適値から少なくとも3%以内にあると推定できる。

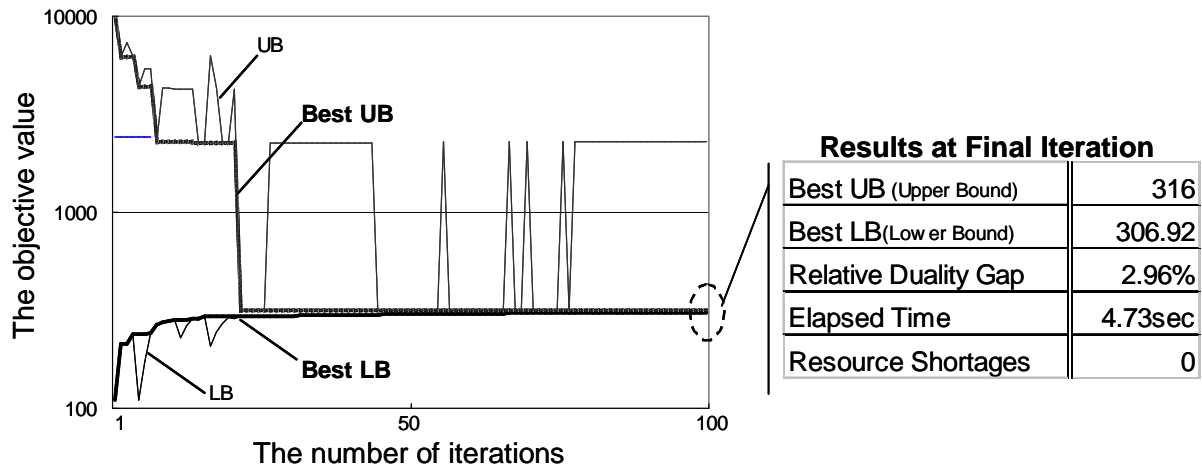


図 4.9: データ 1 (小規模鉄道路線) に対する実験結果
 Fig.4.9: Experimental Results for Data 1 (Small Railway Line)

データ 2: 大規模鉄道路線

図 4.10 にデータ 2 に対する実験結果を示す。

データ 1 と同じく資源割当不足数 0 の実行可能解が得られた。相対双対ギャップはデータ 1 よりも小さく、1.56% である。したがって、この場合も最適解に非常に近い解を得ることができた。一方、実行時間は約 40 分である。これはデータ 1 よりも大きな値である。

また、同じく図 4.10 より、下界値の立ち上がりは比較的早く、少数の反復 (約 300 回) で最良値である 4703.67 の近傍に到達していることがわかる。下界値は最適値の推定値とみなすことができるので、この結果は精度の高い推定値が探索の早い段階で得られたことを示している。その一方で上界値の改善は緩やかである。最良値 4777 までに多くの反復 (約 800 回) を必要としている。

最短経路探索の実行効率

以下では補助的な評価として、本提案方式における最短経路探索の実行効率を評価する。

最短経路探索のアルゴリズムとして DAG アルゴリズムを用いる本提案方式と、その代替

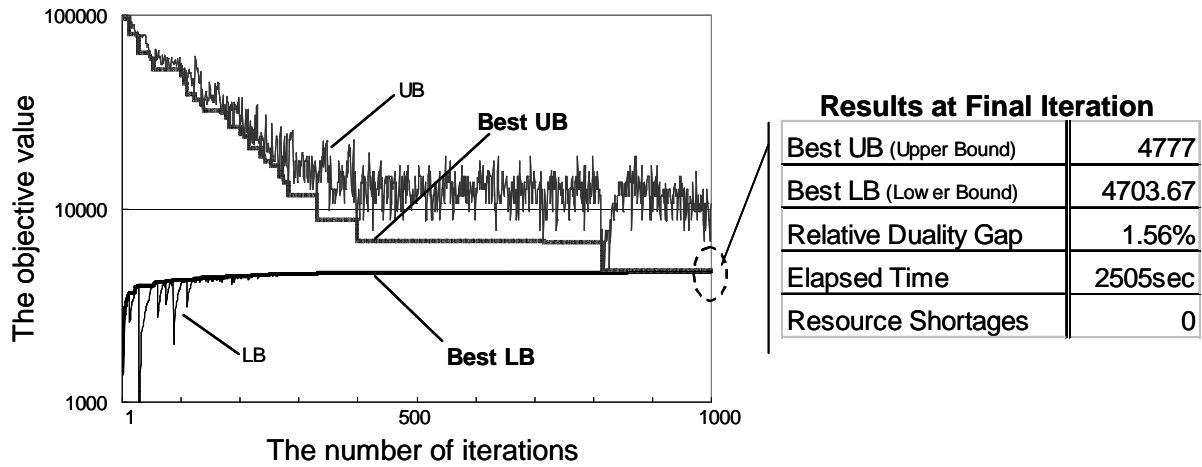


図 4.10: データ 2 (大規模鉄道路線) に対する実験結果
 Fig.4.10: Experimental Results for Data 2 (Large Railway Line)

手段としてベルマンフォード法を用いた場合の実行時間との比較を図 4.11 に示す。

これよりあきらかなとおり、両者の実行時間の差は非常に大きい。加えて、これは両者の時間計算量の違いから自明であるが、ベルマンフォード法の方がデータ量の増大に伴う実行時間の増加割合がはるかに大きい。データ 1 とデータ 2 の実行時間を比較すると、DAG アルゴリズムではその差は約 53 倍なのに対して、ベルマンフォード法は約 244 倍となる。また、大規模路線であるデータ 2 の場合、ベルマンフォード法を使用するとラグランジュ緩和法の 100 回の反復計算に 8 時間以上を要する。データ 2 の実験結果より、最適解に近い実行可能解を得るためには 1000 回の反復が必要なことから、ベルマンフォード法を用いて同等の解を得るには 80 時間を要することになる。以上の結果より、ベルマンフォード法を用いると実行効率の観点で本提案方式の実用性を著しく損なうことになるといえる。

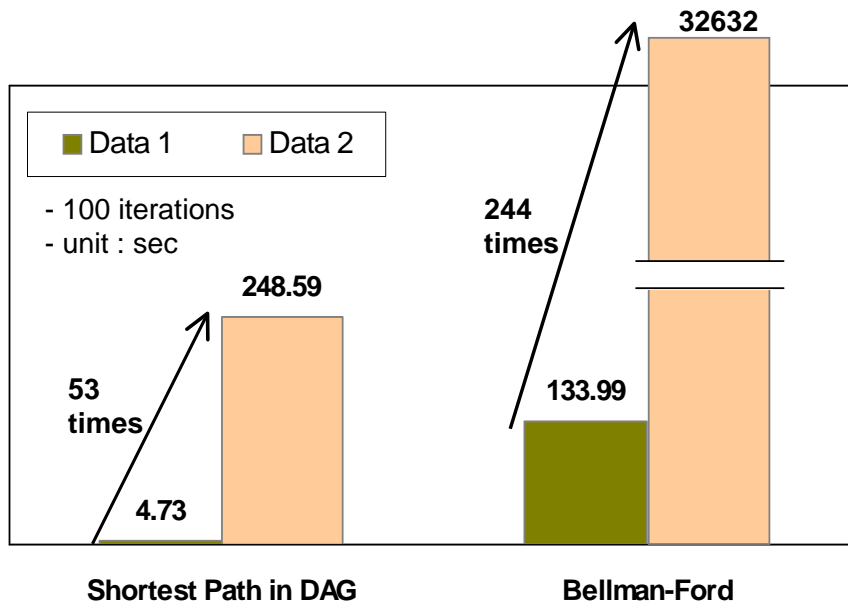


図 4.11: 最短経路探索の実行時間の比較

Fig.4.11: Execution Time : Two Cases Using Different Shortest Path Algorithms

4.5.4 実験結果：スケジューリング方式2

データ 1: 小規模鉄道路線

実験の結果、資源割当不足のない実行可能なスケジュールが得られた。実行時間は0.66秒であった。内訳は、初期解生成が0.05秒、局所探索の反復処理が0.61秒である。なお、初期解生成の実行時間には、ネットワークフローモデルの生成と条件違反判定の処理時間を含む。局所探索における反復回数は13回、反復1回あたりの近傍サイズ（解候補の個数）の平均値は159個であった。また、動的に定めた目的関数の第1,2項の重みは、 $w_1 = 20.2$, $w_2 = 142.9$ である。

図 4.14 に目的関数の各項の値を示す。なお、評価指標「資源割当不足」に対応する第4項は自明（値0）のため省略した。これより、程度に差はあるものの、スケジューリング方式2の特徴である暫定解の反復的な修正により初期解の全ての評価指標を改善することができた。具体的には、目的関数第1項の「車両の折り返し時間の偏差」について35.0%、同じく第2項の「車両の走行距離の差分」について63.9%、第3項の「元のスケ

「元からの変更量」について 34.3%、それぞれ初期解を改善することができた。

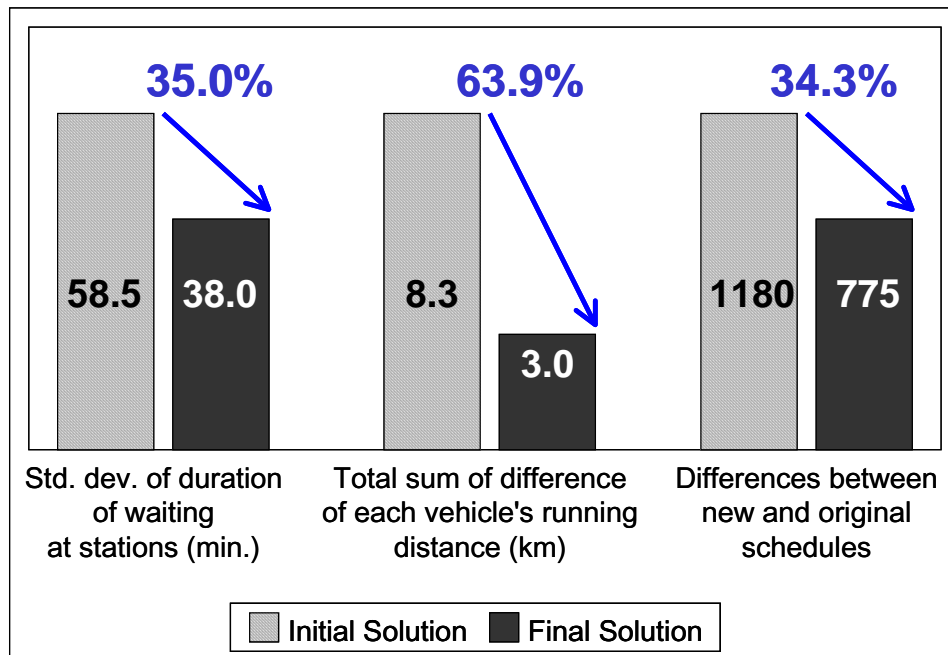


図 4.12: 最良解の目的関数値 (小規模鉄道路線)

Fig.4.12: Evaluation Values of the Solution (Small Railway Line)

図 4.13 に、評価指標「元からのスケジュールからの変更量」(目的関数第 3 項)に対する精度評価の結果を示す。ここでの精度評価とは、提案スケジューリング方式 1 を適用して得られた最適解の最良下界値と評価値とのギャップ(乖離度合い)を計算することを意味する。すなわち、ギャップは、評価値 f と最良下界値 ϕ_{best} より、 $\frac{f - \phi_{best}}{\phi_{best}} \times 100$ (%) と定義される。

比較の参考のため、図 4.13 は提案スケジューリング方式 1 の精度評価の結果も含む。これより、提案スケジューリング方式 2 のギャップの値は 152.51% である。これは、最適値に非常に近いと考えられる最良下界値に対して、評価値はその約 2.5 倍の大きさであることを意味する。一方、提案スケジューリング方式 1 のギャップは 2.96% である。この精度評価により、両者は評価指標「元からのスケジュールから変更量」の充足度合いに差異があることを客観的に示すことができた。

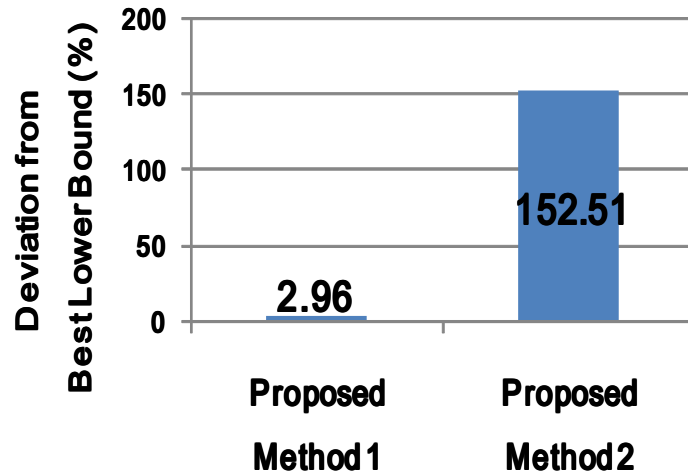


図 4.13: 精度評価結果 (小規模鉄道路線)

Fig.4.13: Deviation from the Best Lower Bound (Small Railway Line)

データ 2: 大規模鉄道路線

実験の結果、資源割当不足のない実行可能なスケジュールが得られた。実行時間は376.2秒であった。内訳は、初期解生成が5.5秒、局所探索の反復処理が370.7秒である。なお、初期解生成の実行時間には、ネットワークフローモデルの生成と条件違反判定の処理時間を含む。局所探索における反復回数は123回、反復1回あたりの近傍サイズ(解候補の個数)の平均値は2099個であった。また、動的に定めた目的関数の第1,2項の重みは、 $w_1 = 106.0$, $w_2 = 43.3$ である。

図4.14に各評価指標の評価値を示す。目的関数第4項の「資源割当不足」は自明(値0)なので省略した。程度に差はあるものの、初期解の全ての評価指標を局所探索法により改善した。具体的には、目的関数第1項の「車両の折り返し時間の偏差」について52.0%、同じく第2項の「車両の走行距離の差分」について32.9%、第3項の「元のスケジュールからの変更量」について6.9%、それぞれ初期解を改善することができた。

データ1の場合と同じく、図4.13に、評価指標「元のスケジュールからの変更量」(目的関数第3項)に対する精度評価の結果を示す。比較の参考のため、提案スケジューリン

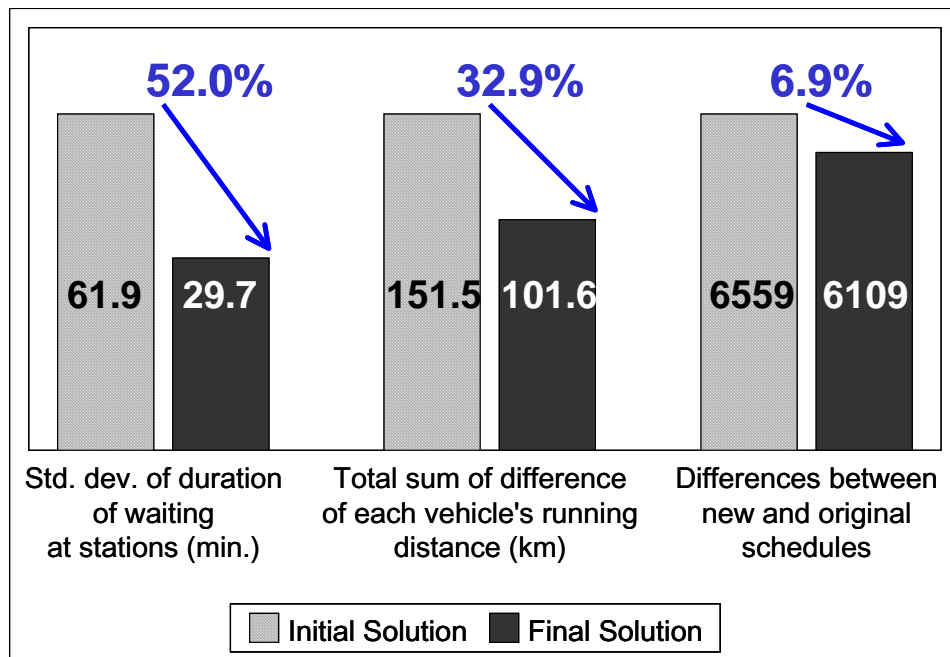


図 4.14: 最良解の目的関数値 (大規模鉄道路線)

Fig.4.14: Evaluation Values of the Solution (Large Railway Line)

グ方式 1 の精度評価の結果も併せて示す。

これより、提案スケジューリング方式 2 のギャップの値は 29.83% である。これは評価値が最良下界値の約 1.3 倍の大きさであることを意味する。小規模鉄道路線 (データ 1) の場合と比較すると乖離の度合いは小さい。一方、提案スケジューリング方式 1 のギャップは 1.56% であった。したがって、データ 2 においても両者は評価指標の充足度合いに自明な違いがあることを客観的に確認することができた。

4.6 考察

4.6.1 再作成の効率

提案スケジューリング方式 1 の計算時間は、小規模鉄道路線の場合は約 5 秒、大規模鉄道路線の場合は約 40 分であった。一方、提案スケジューリング方式 2 の計算時間は、小規模路線の場合は約 0.7 秒、大規模鉄道路線の場合は約 6 分であった。

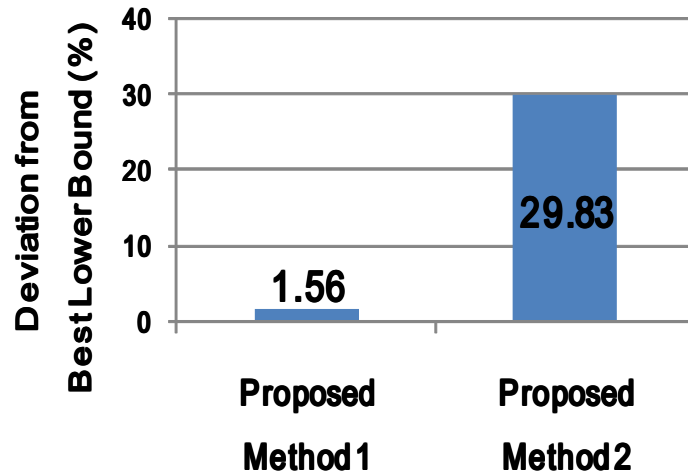


図 4.15: 精度評価結果 (大規模鉄道路線)

Fig.4.15: Deviation from the Best Lower Bound (Large Railway Line)

本線運用整理において、スケジュールの再作成に許容される時間は、運行乱れの状況や鉄道事業者の業務フローの違い等に応じて様々に異なる。例えば朝ラッシュ時に30分程度の運行支障が発生した場合は、以下に示す3つのステップで本線運用整理をおこなうことが多い。

1. 初動対応 (時間幅: 30分) 運転再開に向けた対応である。運休や臨時列車の設定といった、比較的大きな列車ダイヤの変更が行われる。よって、運転再開までの短い時間の中で列車ダイヤの大きな変更が五月雨式に発生するため、スケジュールの再作成を何度も素早く繰り返す必要がある。許容時間の目安としては秒単位が望ましく、長くても1分程度と考えられる。
2. 回復 (時間幅: 2~3時間) 運転再開後から列車運行が通常状態に戻るまでの対応である。駅での追い越しや退避順序の変更といった比較的小さな列車ダイヤ変更が多い。よって、列車ダイヤが車両や乗務員の運用スケジュールに与える影響は限定的であり、影響を与える頻度も少ないことから、初動対応と比べるとスケジュール再作成に許容される時間的余裕は大きい。許容時間の目安としては数分程度と考えられる。

3. 調整（時間幅：12時間以上） 当日の全列車の運行が終了するまでの対応である。夕方のラッシュ対応や終電の接続調整、翌日の運行との整合性を確保する等の目的で列車ダイヤの小規模な修正がおこなわれる。時間幅の長さも考慮すれば、列車ダイヤが車両や乗務員の運用スケジュールに与える影響はさらに限定的であり、影響を与える頻度も少ない。したがってスケジュール再作成に許容される時間的余裕は最も大きい。許容時間の目安としては数十分程度と考えられる。

小規模路線の場合は、提案方式1と2の双方共に秒単位以下で結果が得られている。したがって本線運用整理の3つの段階全てに対してどちらの提案方式も有効と考えられる。

一方、大規模路線の場合は、どちらの提案方式も初動時に求められる秒単位の応答性を確保するのは困難である。ただし、対応の時間幅が30分程度と短いことから、スケジュール再作成の範囲を狭めることで必要な応答性を満足することが可能である。数値実験ではスケジュール再作成の範囲を約18時間としたので、これを30分とすれば、応答性を約1/36にできる。こうすることにより、提案方式1であれば約1分、提案方式2であれば約10秒でスケジュール再作成を実行することができる。

同様に、数分での応答性が要求される回復時においても、スケジュール再作成の範囲を狭めることで提案方式1を有効に適用できると考えられる。すなわち、再作成の範囲を3時間とすれば、提案方式1の実行時間を1/6の約7分に短縮することができる。

ただし、最適化の観点からは、スケジュール再作成の範囲を狭めず、スケジュール全体を対象として再作成を実行することが望ましい。そこで提案方式の解探索効率の向上が今後の課題である。例えば提案スケジューリング方式1の場合、ラグランジュ緩和法の一回の反復は最短経路探索にDAGアルゴリズムを用いることで十分に効率化しており、これ以上の計算効率向上はあまり期待できない。そこで実行可能解の生成法とラグランジュ乗数の更新方法とを改良し、少ない反復回数でギャップを収束させることが有望と考えられる。一方、提案スケジューリング方式2は、生成検査法のインスタンスとして用いた局所探索法を、より洗練された他の手法で置き換えることが考えられる。例えば、メ

タヒューリスティクス的一种であり、大域的な探索制御メカニズムを備えたGA[96]やアントコロニー最適化[97]の適用等が考えられる。

解の継続的改善

提案方式1と2は、どちらも解を漸進的に改善する方式であることから、探索を途中で打ち切ってもその時点での最良解を得ることができる。したがって、計算時間が長いといった理由でスケジュールの再作成を途中で打ち切っても、何らかの再作成結果は算出可能である。このような特徴を備えることから、提案方式を用いることで許容時間に応じたスケジュール再作成を柔軟におこなうことができると考えられる。

4.6.2 精度評価

提案スケジューリング方式1の特徴のひとつとして、下界値を用いた精度評価が挙げられる。すなわち、探索で得られた最良下界値を最適値の推定値とみなし、そこからのギャップ(乖離度合い)を測ることで解の精度を定量的に見積もることができる。応用の観点からは、この特徴には以下に示す2つの利点がある。

1. 輸送現場のスタッフやスケジュール作成担当者の理解や承認を得やすい。
2. 手作業で作成した、あるいは他の方式を適用して得られたスケジュールの有効性を定量的に示すことができる。

1. に関しては、列車の運行に応じて遅れが拡大する等、状況が変化の中で、スケジュール作成者はスケジュールの再作成を繰り返す必要がある。また、列車運行が乱れると、輸送現場では複数の作業が輻輳すると共に情報も錯綜し混乱状況に陥りやすい。これらを考え合わせると、スケジュールの有効性を定量的に示すことは、限られた時間の中でスケジュール作成担当者がスケジュールの妥当性を判断し、現場スタッフが円滑に輸送回復作業を進める上で重要な役割を果たすと考えられる。

一方、2. に関しては、提案方式1で得られた最良下界値を最適値の推定値とみなすことで、他の方式で得られたスケジュールの評価値の精度を最適値からの乖離度合いにより推定することができる。同様のことは手作業で作成されたスケジュールに対してもおこなうことができる。この精度情報を用いれば、スケジュールの良し悪しを客観的に判断することができる。よって、スケジューリング方式の改善、ダイヤ乱れ収束後の関係者による振り返り、スケジュール作成担当者の教育等、様々な場面で精度情報を有効活用できると期待される。

4.6.3 デペンダビリティのさらなる向上

数値実験結果をふまえ、2つの提案スケジューリング方式の比較を表4.2に示す。

表 4.2: 2つの提案方式の比較

Table4.2: Comparison of two Proposed Methods

Comparison Items	Proposed Method 1	Proposed Method 2
Optimization of Differences between New and Original Schedules	○	△
Non Linear Objectives	×	○
Accuracy Evaluation of Solution	○	×
Rescheduling Efficiency	△	○

提案スケジューリング方式1は、運用計画の変更コストである元のスケジュールからの変更量の最適化を行うものである。最適化技法としてラグランジュ緩和法を応用し、最

適値の下界と上界とのギャップを用いた解の精度評価を行うことにより、元のスケジュールからの変更量を最適値の近くまで最小化した再スケジューリングをおこなう。

一方、提案スケジューリング方式2は、運用計画の変更コストに加え、資源の運用に関する非線形な評価指標も対象とした最適化を行うものである。数値実験結果より、元のスケジュールからの変更量に関しては提案方式1より最適化の度合いは低いものの、非線形な評価指標の値を初期解から改善することができる。また、再作成の実行効率は提案方式1の約7倍である。

以下では、このように特徴が異なる2つの提案方式を組み合わせることでより効果的なスケジュール再作成を実現し、デペンダビリティをさらに向上する可能性について検討する。

図4.16は、2つの提案方式を組み合わせさせたスケジュール再作成モデルの一例である。

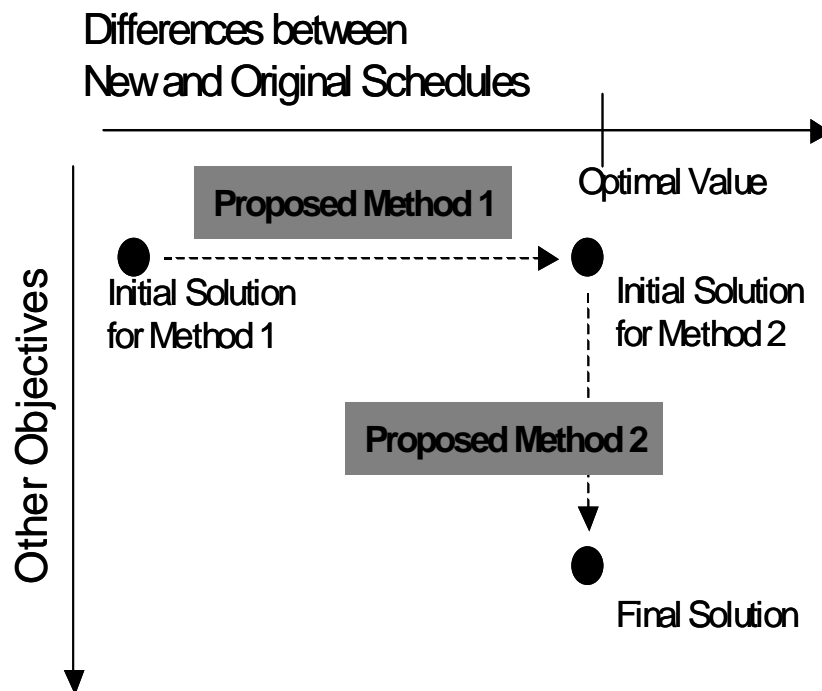


図 4.16: 提案手法を組み合わせさせたスケジュール再作成モデル
 Fig.4.16: Hybrid Solution Model of two Proposed Methods

このモデルは、提案方式1と提案方式2を続けて実行するものである。まず、提案方式

1により、元のスケジュールからの変更量を最適値の近くまで最小化したスケジュール再作成をおこなう。そしてその結果を初期解として提案方式2に入力し、スケジュール再作成を再度おこなう。その際、元のスケジュールからの変更量だけでなく他の評価指標も目的関数に加えることで、元のスケジュールからの変更量を初期解と同等の水準に保ちながら他の評価指標の値を改善することができる。

提案方式2は提案方式1と比較して再作成の実行効率が高い。したがってこのモデルを用いれば、提案方式1の実行時間にわずかな時間を追加するだけで、元のスケジュールからの変更量を高度に最適化するだけでなく、他の評価指標の最適化も考慮したスケジュール再作成を実現できる。また、最終的に得られた再作成結果に対しても精度評価ができることから、その評価結果を再作成されたスケジュールと共に出力すれば再作成担当者の意思決定を効果的に支援することも期待できる。

4.7 本章の総括

本章では、(課題1)に対処するための本線運用整理向けリアクティブスケジューリング技術について明らかにした。まず、リアクティブスケジューリング問題の構造や制約を陽に扱うため、列車ダイヤからの要求である列車の運行時刻及び位置の情報と、資源の運用制約と、スケジュール自体とを統合して扱うことが可能なネットワークフローモデルを提案した。そして、提案モデルで記述したリアクティブスケジューリング問題を整数計画問題として定式化し、問題の特徴をふまえたヒューリスティクスを組み込むことで問題を効率的に解くための2通りの方式を提案した。(方式1)は最適化技法としてラグランジュ緩和法を応用し最適値の下界と上界とのギャップを用いた解の精度評価を行うことにより元のスケジュールからの変更量を最小化した再スケジューリングを求めるものである。(方式2)は最適化技法として局所探索法を応用したものであり、(方式1)では評価できない非線形な評価指標(全車両の折り返し時間の標準偏差と各車両の走行距離の差分の総和)にもとづく最適化を行うものである。これらの提案方式を組み込んだプロトタイプシステムを実装し、実路線の運行乱れ時の運用データを用いた数値実験を実施した。その結果、方式1により、元のスケジュールからの変更量の最適値に対して、評価値の乖離が約1.6%以内の実行可能解を数十分のオーダの計算時間で算出できることを示した。さらに、方式2により、元のスケジュールからの変更量に加え、非線形な評価指標である全車両の折り返し時間の標準偏差と各車両の走行距離の差分の総和について、従来方式と比較して前者は約52%、後者は約33%、それぞれ改善できることを示した。最後に、異なる特徴を持つ2つの提案方式を組み合わせることでスケジューリング機能を構成することで、デペンダビリティをさらに向上する可能性を議論した。

第5章

車両基地運用整理のためのリアクティブスケ ジューリング方式

本章では、課題2に関し、車両基地の多様なレイアウトに対応し、複数の物理制約や資源運用制約を同時に満足しつつ良好な応答性を実現する車両基地運用整理向けリアクティブスケジューリング技術について明らかにする。

5.1 研究の背景

車両基地運用整理の目的は、車両基地における車両の清掃・点検といった保全作業を列車運行乱れの際にも円滑に進めるため、車両基地構内における車両の移動と配置のスケジュールを再作成することである。この車両の移動と配置のスケジュールのことを、以降では構内入換計画と呼ぶ。

構内入換計画の計算機によるスケジューリング方式の関連研究としては、オランダの鉄道路線を対象として、翌日の列車に充当する車両の組合せと構内配置の決定を整数計画問題として定式化し、大規模組合せ問題の解法の一つである列生成法を用いて解を求めるもの [44]、ファジィ理論によるモデル化と遺伝的アルゴリズム (GA) の応用により、車両の組合せとその分解・組立ての構成手順を求めるもの [45]、車両基地を出区する際の移動効率を考慮して車両配置を最適化するグラフ理論によるアプローチ [46]、等が挙げられる。また、類似問題である駅構内における入換計画作成を対象として、PERT と近似解法の一つである確率的局所探索の組合せよるアルゴリズムの研究等もおこなわれている [47][48][49]。

上記の従来研究は、対象とする車両基地の運用形態が様々に異なることから、スケジュー

リングの前提条件や目的等、対象問題の捉え方自体にいくつかの相違がある。また、これらはいずれも所与の前提条件と入力データの下で、制約条件や評価指標に沿ったスケジュールの全体を、いわば白紙から作成することに焦点をあてたものであり、列車運行乱れ時の再スケジューリングに求められる応答性や評価指標（元のスケジュールからの変更量最小化）を満足することが困難と考えられる。

スケジュールの再作成にあたっては、本線運用整理と同様に元のスケジュールからの変更量の最小化が求められる。それに加えて、実行可能性を確保するために車両基地のレイアウト（配線構造）を考慮することが重要である。レイアウトは車両基地毎に異なり、しかも複雑・多岐にわたるため、それらをスケジューリングの制約として共通的に扱うことが求められる。

以上をふまえ、本研究では、車両基地の多様なレイアウトに対応し、複数の物理制約や資源運用制約を同時に満足しつつ良好な応答性を実現する構内入換計画のリアクティブスケジューリング方式を検討する。

5.2 問題定義

5.2.1 FJSSP モデルによる車両基地運用整理のモデル記述

構内入換計画は、検査・点検・清掃といった各種保全作業を実施するための車両基地構内における車両の移動と配置に関するスケジュールである。本線上の営業運転の合間や営業終了後の時間帯に予定された作業を漏れなく実施するために、車両を車両基地構内で「いつ」「どこへ」移動するかを定めたものである。これは図 5.1 に例を示すように、時間帯×番線のダイヤ図形式で表現することができる。なお、ここでの車両とは、実際には1列車の構成単位である連結された車両の集まりを指すものとする（一般的には、これを「車両編成」、あるいは単に「編成」と呼ぶことが多い）。

構内入換計画の作成にあたり事前に与えられる情報としては、(1) 車両基地構内の番線・進路等のレイアウト、作業や車両についての情報等の各種基礎データ、(2) 各車両の作業

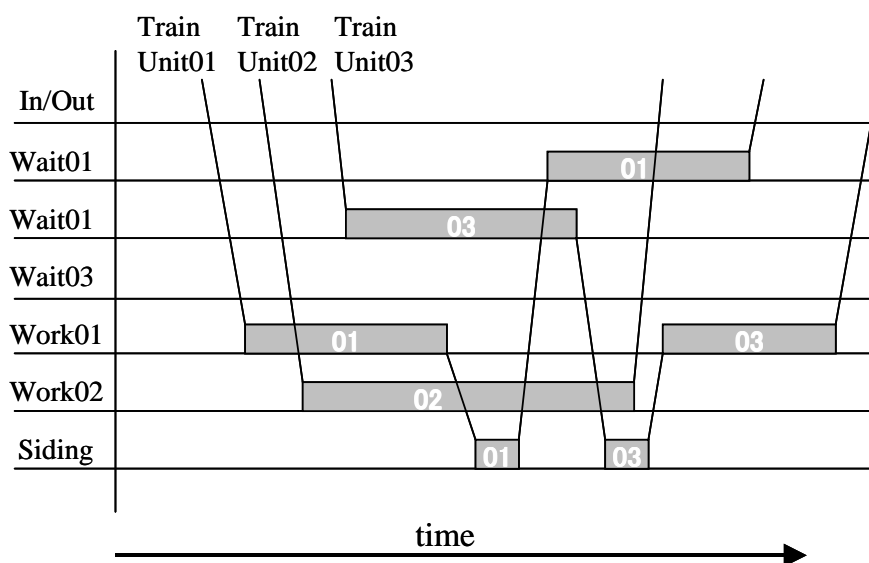


図 5.1: 構内入換計画の例

Fig.5.1: An Example of Shunting Schedule

予定、(3) 同じく各車両の車両基地への入出区時刻、が挙げられる。

スケジュール作成において満たすべき代表的な制約条件としては、(1) 他の車両との間で進路、番線の時間的競合が発生しない、(2) 予定作業の設備要件や車両の長さなどに応じた適切な番線に車両を在線させる、(3) 出区時刻を超過せずに予定された作業を漏れなく実施する、といったものが挙げられる。これら以外にも移動の所要時間や、進路支障といった信号設備上の制約など、様々な条件を考慮することが求められる。

列車運行乱れへの迅速な対応を可能とする上で、スケジューリング方式には高い応答性が求められる。加えて、現場作業の安全性と効率の観点から、再作成された構内入換計画は元のスケジュールからの変更量ができるだけ少ない方が望ましい。変更量が少ないほど作業手配や事前準備等への影響が小さい上、作業者の思い違い等に起因するミスを防ぐことも期待でき、総じて現場の混乱を最低限に抑えることができる。

車両基地のレイアウト（番線、進路）を作業に使用する資源と捉えることで、本問題は生産分野におけるフレキシブルジョブショップスケジューリング問題 (Flexible Job Shop Scheduling Problems, FJSSP)[85][86][87] の類似問題とみなすことができる。

この場合、一個の車両が車両基地に入区してから出区するまでの移動と配置の一連の繰り返しが「ジョブ」、個々の移動と配置が「オペレーション」にあたる。

図 5.2 にジョブの定義例を示す。この例は、「車両基地に入区後、作業番線へ移動して

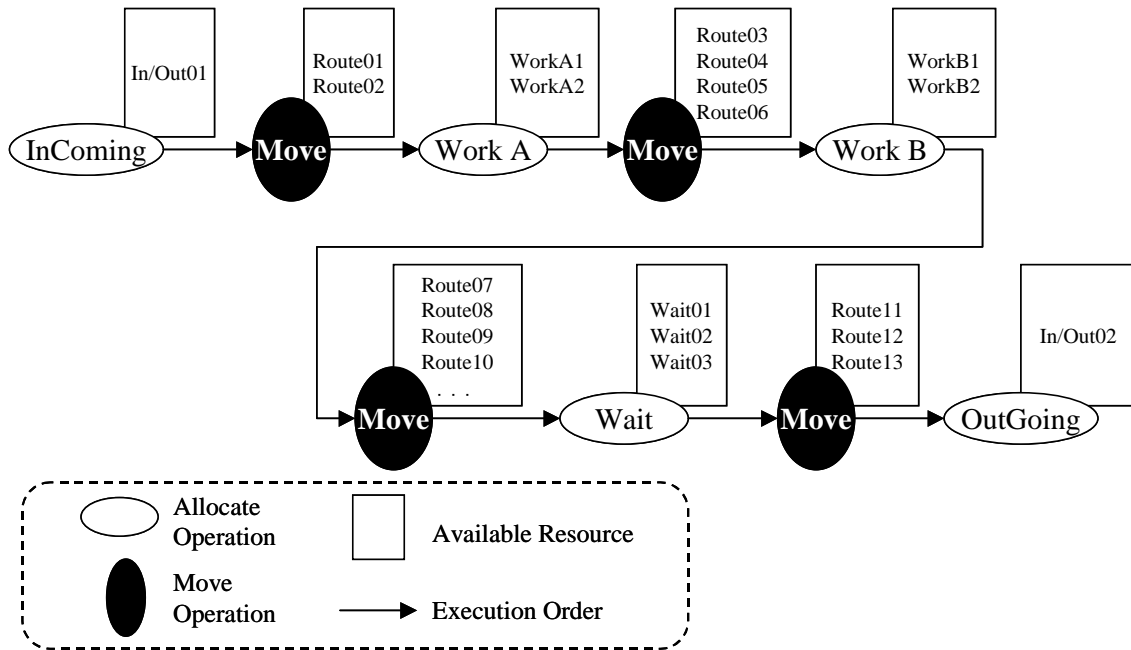


図 5.2: ジョブ定義の例

Fig.5.2: An Example of Job Definition

作業 A を実施する。その後は別の作業番線へ移動して作業 B を実施し、終了後は留置番線に移動して出区時刻までそこで待機する」というオペレーションの系列を一個のジョブとするものである。

白い楕円が車両の配置、黒い楕円が番線間の移動を表す。各オペレーションに付された四角の中には使用可能な資源が示される。配置に対しては番線が、移動に対しては進路が与えられる。なお、車両の入区と出区は、入出区線（車両基地の外部とつながる通路）を使用する仮想的な番線配置とみなす。

FJSSP は、ジョブショップスケジューリング問題 (Job Shop Scheduling Problems, JSSP)[93] を、生産現場等における現実のスケジューリングの実態に合わせて拡張したより実際的な問題である。JSSP においては、各オペレーションを処理する資源（機械にあたる）が

あらかじめ決められており、納期遅れといった何らかの評価基準の下で、各資源上でのオペレーションの処理順序を求めることが主眼となる。これに対して、FJSSPは各オペレーションを処理可能な資源が複数存在し、オペレーションの処理順序のみならず資源の割当も決定することが求められる。

JSSPは、組合せ最適化の研究分野における、いわゆるNP困難(NP Hard)[94]なクラスに属する問題であり、最適解を多項式時間で求める一般的な厳密解法が存在しないことが強く示唆されている。JSSPの拡張問題であるFJSSPも当然NP困難な問題であるが、オペレーションへの資源の割当をも決定要素に含むことから、可能な組合せの数がはるかに多く、この意味でFJSSPはJSSPよりも困難な問題であるといえる。

本問題もFJSSPと同様に各オペレーション(移動、配置)の資源(進路、番線)が所与ではなく、複数の候補の中から適切な資源を選択する必要がある。ただし、各車両が車両基地へ入出区する時刻があらかじめ決められており、その固定された時間枠の中でスケジューリングをおこなうことから、FJSSPと比較すると探索空間が狭いと推察される。

その一方で、本問題は(1)各車両の車両基地への入区から出区までの時間枠が固定されるため、レイアウトを構成する資源(進路、番線)が車両の移動・配置の際に競合しやすい、(2)移動と配置を矛盾無く繰り返す上で、必要な資源の組合せがレイアウトに強い制約を受ける、(3)時間軸に沿って移動と配置を切れ目なく連続的に実行する必要がある、等の制約から、スケジュールの局所的な修正の影響が広範囲に影響しやすいという特徴を持つ。

例えば、ある車両の配置時間を延長したことで資源の競合が生じると、それがさらに他の資源競合や時刻の矛盾が連鎖的に発生しスケジュールの実行可能性を大きく損なうことが考えられる。したがって、本問題は資源競合等を含まない実行可能解を求めることが困難な問題と考えられる。また、オペレーションの実行順序だけではなく、その実行時刻まで決定する必要がある点も一般的なFJSSPとは異なっている。

5.2.2 車両基地運用整理リアクティブスケジューリング問題の定式化

FJSSP としてのモデル化に基づき、本問題を制約充足最適化問題として定式化する。ここでは、各車両のジョブ定義が、作業内容や構内レイアウト等の情報に基づき事前に与えられるものとする。

[記号の定義]

集合

J : ジョブ集合。 $J = \{1, \dots, n\}$.

O : オペレーション集合。 $O = \{1, \dots, m\}$.

M : 資源集合。 $M = \{0, 1, \dots, l\}$.

0 は車両基地のレイアウトに存在せず、全てのオペレーションに割当可能で、かつ競合を考えない特別な資源とする。これをバッファ資源と呼ぶ。バッファ資源がひとつでも割当てられたスケジュールは実行可能解とはならないが、車両基地に実在する他の資源を割当てると競合が避けられない場合にバッファ資源が代わりに割当てられる。すなわちバッファ資源は資源競合の発生の有無を識別する役割を持つ。なお、4章で示した定式化では資源集合を R としたが、本定式化においては通常の資源だけでなくバッファ資源を含めたものを資源集合とすることから、両者を区別するためにあらためて別の記号 (M) を与えた。

O_j : ジョブ j に属するオペレーションの集合。 ($j \in J$).

O_j は O の第 $N_{j-1} + 1$ 番目から N_j 番目までの連続する要素から構成され、この順番で実行されるものとする。ここで $N_0 = 0$, $N_j = \sum_{k=0}^j m_k$, m_k は O_k の要素数である。したがって、 $O \equiv \bigcup_{j=1}^n O_j$ である。

M_i : オペレーション i に割当て可能な 0 以外の資源の集合。 ($i \in O$).

O^{Rt}, O^{Tr} : 移動 (Rt), 配置 (Tr) オペレーションの集合。 ($O \equiv O^{Rt} \cup O^{Tr}$)

M^{Rt}, M^{Tr} : 進路 (Rt), 番線 (Tr) の集合。 ($M \equiv M^{Rt} \cup M^{Tr} \cup \{0\}$)

M_k^{src}, M_k^{dst} : 番線 $k \in M^{Tr}$ を発番線 (着番線) とする進路の集合。

定数

r_j, h_j : ジョブ j の開始時刻と終了時刻。 ($j \in J$).

p_i : オペレーション i の最小処理時間。 ($i \in O$).

δ_{ij} : 進路競合フラグ。 ($i, j \in M^{Rt} \cup \{\mathbf{0}\}$).

$$\delta_{ij} = \begin{cases} 1 & \text{進路}i\text{と進路}j\text{は競合する} \\ 0 & \text{それ以外} \end{cases}$$

i, j の少なくとも一方が $\mathbf{0}$ ならば, 常に $\delta_{ij} = 0$ (競合なし) とする。

u, q : 進路支障時分、番線続行時隔。

C_{ik} : オペレーション i に資源 k を割当ててコスト。 ($i \in O, k \in M$).

$$C_{ik} = \begin{cases} 1 & k=0\text{のとき} \\ 0 & \text{それ以外} \end{cases}$$

決定変数

s_i, e_i, z_i : オペレーション i の開始時刻、終了時刻、使用資源。 ($i \in O$).

各決定変数のドメイン (取り得る値の範囲) は、 $s_i \geq 0, e_i \geq 0, z_i \in M_i \cup \{\mathbf{0}\}$ である。

論理記号

&&, ||, \Rightarrow : かつ、または、もし~ならば

$$Or_{k=1}^n A_k: A_1 || A_2 || \cdots || A_n$$

[制約条件]

(1) 番線競合制約

2つの番線配置オペレーション $i, i' (\in O^{Tr})$ に対してバッファ資源以外の同一番線が割当てられている場合、両者の実行時間帯は重複しない。

$$\begin{aligned} z_i = z_{i'} \ \&\& \ z_i \neq \mathbf{0} \Rightarrow s_i \geq e_{i'} + q \ \parallel \ s_{i'} \geq e_i + q \\ \forall i, i' \in O^{Tr}; \ i' > i \end{aligned} \quad (5.1)$$

(2) 進路競合制約

2つの移動オペレーション $i, i' (\in O^{Rt})$ に対して競合する進路が割当てられている場合、両者の実行時間帯は重複しない。

$$\begin{aligned} Or_{k \in M_i} \{z_i = k \ \&\& \ \delta_{kz_{i'}} = 1\} \Rightarrow \\ s_i \geq e_{i'} + u \ \parallel \ s_{i'} \geq e_i + u \ \forall i, i' \in O^{Rt}; \ i' \neq i \end{aligned} \quad (5.2)$$

(3) レイアウト制約 1

同じジョブの中で、番線配置オペレーション $i (\in O^{Tr})$ の次の移動オペレーション $i+1 (\in O^{Rt})$ に対しては、 i に割当てられた番線を発番線とする進路を割当てる。

$$\begin{aligned} z_i = k \Rightarrow Or_{l \in M_k^{src} \cup \{\mathbf{0}\}} \{z_{i+1} = l\} \\ \forall i \in O^{Tr}; \ i \neq N_j; \ \forall j \in J \\ \forall k \in M_i \end{aligned} \quad (5.3)$$

(4) レイアウト制約 2

同じジョブの中で、番線配置オペレーション $i (\in O^{Tr})$ の1つ前の移動オペレーション $i-1 (\in O^{Rt})$ に対しては、 i に割当てられた番線を着番線とする進路を割当てる。

$$\begin{aligned}
 z_i = k &\Rightarrow Or_{l \in M_k^{dst} \cup \{0\}} \{z_{i-1} = l\} \\
 \forall i \in O^{Tr}; i \neq N_{j-1} + 1; \forall j \in J \\
 \forall k \in M_i
 \end{aligned} \tag{5.4}$$

(5) 開始時刻制約

各ジョブの先頭のオペレーション（入区）は、開始時刻がジョブの開始時刻に一致する。

$$s_i = r_j \quad \forall i = N_{j-1} + 1; \forall j \in J \tag{5.5}$$

(6) オペレーション連続性制約

各ジョブの先頭以外のオペレーション（入区以外の全て）は、開始時刻が1つ前のオペレーションの終了時刻に一致する。

$$s_i = e_{i-1} \quad \forall i \neq N_{j-1} + 1; \forall j \in J \tag{5.6}$$

(7) 終了時刻制約

各ジョブの最後のオペレーション（出区）は、終了時刻がジョブの終了時刻に一致する。

$$e_i = h_j \quad \forall i = N_j; \forall j \in J \tag{5.7}$$

(8) 在線時間制約

番線配置の終了時刻は、（開始時刻 + 最小処理時間）以上でなければならない。

$$e_i \geq s_i + p_i \quad \forall i \in O^{Tr} \tag{5.8}$$

(9) 移動時間制約

移動の終了時刻は、（開始時刻 + 最小処理時間）に一致しなければならない。

$$e_i = s_i + p_i \quad \forall i \in O^{Rt} \quad (5.9)$$

[目的関数]

実行可能解な解を得るためには、「バッファ資源の使用回数の最小化」を意味する以下の目的関数の下で解を求めれば良い。

(1) バッファ資源使用回数最小化

$$\min \sum_{i=1}^m C_{iz_i} \quad (5.10)$$

この目的関数の値が0以外の場合、少なくとも1ヶ所は資源競合が発生していることを意味し、得られた解は実行不可能である。一方、目的関数の値が0ならば、バッファ資源が割当てられたオペレーションは存在しないので、得られた解は資源競合を含まない実行可能解となる。なお、実行不可能な場合には、バッファ資源が割当てられたオペレーションから番線競合を生じる車両(のスケジュール)を特定することができるので、これを手掛かりとして、後述するVSUからISUへと変更する車両を決定することができる。

入換業務に従事する現場担当者は、「次にどの車両を(順序)」「どの番線へ(場所)」(移動させるか)という点を常に意識しながら業務を進めている。よって、番線と移動順序の変更回数は乱れに伴う現場の混乱を最小化する上で重要な指標と考えられる。すなわち、元のスケジュールからの変更量に関しては、使用番線と移動の実施順序という2つの観点から、以下に示す2種類の目的関数を定義する。

(2) 番線変更回数最小化

$$\min \sum_{i \in O^T} \mu(i) \quad (5.11)$$

ここで、 $\mu(i)$ は番線配置オペレーション $i(i \in O^T)$ を引数として取る関数であり、その使用資源 (z_i の値) が元のスケジュールと同じであれば0を、異なっていれば1を返す。なお、白紙からのスケジュール作成の場合には元のスケジュールが存在しないので、常に $\mu(i) = 0$ とする。

(3) 移動順序変更回数最小化

$$\min \sum_{i=1}^{m'-1} \phi(i, \delta, \delta') \quad (5.12)$$

ここで、全ての移動オペレーションを開始時刻の昇順に並べた系列を移動実施系列と呼ぶ。 δ, δ' をそれぞれ再スケジュール後及び元のスケジュールの移動実施系列とし、 $\delta \equiv \sigma_1 \sigma_2 \cdots \sigma_{m'}$, $\delta' \equiv \sigma'_1 \sigma'_2 \cdots \sigma'_{m'}$ ($\sigma_i, \sigma'_i \in O^{Rt}$; $m' = |O^{Rt}|$; $\forall i, j (i \neq j) \sigma_i \neq \sigma_j, \sigma'_i \neq \sigma'_j$) と表す。さらに $ord(\sigma_i, \delta')$ 、 $ord(\sigma_{i+1}, \delta')$ を移動オペレーション σ_i と σ_{i+1} の δ' (元のスケジュール) における序列 (順番) とする時、現場担当者がスケジュールに沿って車両の移動を順次実施していく際に認識する、「元のスケジュールからの差異」を表すものとして $\phi(i, \delta, \delta')$ を次のように定義する。 $ord(\sigma_i, \delta') > ord(\sigma_{i+1}, \delta')$ 、すなわち再スケジュール後と元のスケジュールとでは序列の大小関係に逆転が生じている場合に $\phi(i, \delta, \delta') = 1$ 、それ以外は $\phi(i, \delta, \delta') = 0$ 、ただし白紙からのスケジュール作成の場合は常に $\phi(i, \delta, \delta') = 0$ とする。

以上の通り、本問題は複数の目的関数を同時に満足する多目的最適化問題として定式化される。

5.3 基地構内レイアウト条件を考慮した最適スケジューリング方式

5.3.1 基本アイデア

探索領域を効果的に絞込み解探索を効率良くおこなうことと、元のスケジュールからの変更量を最小化することの2点を目的として、図5.3に示す段階的なスケジューリング

のアーキテクチャを採用する。

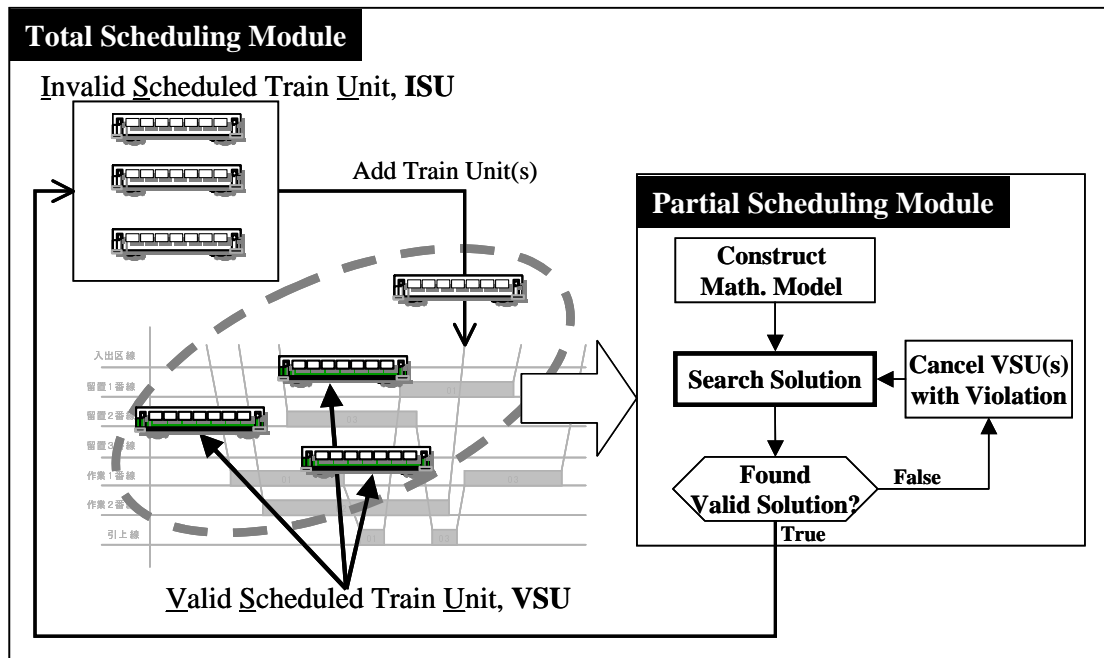


図 5.3: 段階的スケジューリングアーキテクチャ
Fig.5.3: Basic Idea of the Proposed Method

このアーキテクチャは、全体処理と、その構成要素である部分スケジュール作成処理とに分けて説明できる。全体処理は、まず制約条件を満たすスケジュールを持つかどうかで車両を2種類に分類する。そして、制約条件を満たすスケジュールを持つ、すなわち有効な車両の全てと、制約条件を満たすスケジュールを持たない、すなわち無効な車両の一部を選択する。一方、部分スケジュール作成処理は、全体処理で選択された車両を対象としてスケジュールを作成する。その結果スケジュールが得られれば、全体処理に戻り無効な車両を追加して部分スケジュール作成処理を再度実行する。スケジュールが得られなければ、作成対象とした一部の車両を無効にした上で全体処理に戻り、無効な車両を追加して部分スケジュール作成処理を再度実行する。上記の一連の処理を全車両のスケジュールが得られるまで繰り返し実行するものである。

以降では、上記の有効な車両を、VSU(Valid Scheduled Train Unit, VSU)、無効な車両をISU(Invalid Scheduled Train Unit, ISU)と呼ぶ。VSUは制約条件を満足した実行可能なス

スケジュールを既に持っている。そこで、上記の部分スケジュール作成処理において、VSU については保持しているスケジュールを極力維持するものとする。一方、ISU は、列車運行乱れとそれに伴う列車ダイヤや車両運用計画の変更により、スケジュールに制約条件違反を生じたものである。そのため、部分スケジュール作成処理においてスケジュールを白紙から作りなおす必要がある。

以上がスケジューリングアーキテクチャの概要である。なお、スケジュール再作成の初期段階において、全ての車両を ISU に設定すればスケジュール全体を白紙から生成することも可能である。また、スケジュール作成担当者によるスケジュールの部分的な修正が、列車運行乱れそれ他によるものと同様な各種の制約条件違反を引起こす可能性が考えられる。その場合は、制約条件に違反するスケジュールを持つ車両を ISU に設定することで、違反の解消を自動的におこなわせることも可能である。これによりある種のインタラクティブなスケジューリング支援機能を実現することができる。

5.3.2 部分スケジュール作成処理

全体構成

部分スケジュール作成処理は、VSU の全てと追加された一部の ISU とを対象として、制約条件違反のないスケジュールを作成する。具体的には、まず本問題の定式化（5.2.2 節）における定数や変数を具体化し、組合せ解探索アルゴリズムを用いてその解を探索する。これにより部分スケジュールが得られた場合は、追加された ISU を VSU に変更して処理を終了する。部分スケジュールが得られない場合は、番線の競合といった不具合の原因となる VSU を解除、すなわち該当車両を ISU（スケジュールを白紙から作成）へと変更した上で、解の探索を再び実行する。これを違反のない部分スケジュールが得られるまで繰り返した後に、部分スケジュールに含まれる全ての ISU を VSU へ変更して全体処理へ制御を移す。

組合せ解探索アルゴリズムは、後述する本問題の数理モデルに含まれる多様な制約式を直

接的に扱うことが可能な制約プログラミング (Constraint Programming, CP)[88][89][90][95] をベースとして、探索効率化と元のスケジュールからの変更量最小化に寄与する複数のヒューリスティクス (後述) との組み合わせから構成される。

CP は、対象問題を制約条件の集合として定式化し、制約伝播 (Constraint Propagation) と呼ばれる制約式同士の相互作用により決定変数のドメイン (取り得る値の集合) を絞り込んでいく探索手法である。(1) 非線形や論理型、あるいは問題独自の多様な制約条件を柔軟に扱うことができる、(2) 解探索の途中で制約条件を動的に追加したり、メタヒューリスティクス等の他の探索手法とのハイブリッドが比較的容易である、といった性質も備えていることから、スケジューリングをはじめとする離散型の問題領域における実用的な解法として近年注目されつつある [95]。

本問題の定式化は、論理記号や変数添え字 (Variable Subscript)[88] (式 (5.10) の C_{iz_i}) 等、線形計画法をはじめとする一般的な数理計画法の枠組みでは扱うことができない特別な表記法を含んでいるが、CP であれば原理的にはこれを直接扱うことが可能である。CP を用いて離散型の探索問題を解く場合、探索処理の基本的なフレームワークは探索木を用いたバックトラック探索が一般的である。探索木のノードを辿る (決定変数のいずれかを選択して値を割当て) 度に制約伝播を実行し、残りの決定変数のドメインをそれまでの値割当て結果と制約条件とに矛盾がないように動的に縮小していく。

探索効率化のアイデア

ただし、NP 困難な FJSSP の類似問題である本問題の定式化に対して CP をそのまま適用するだけでは、組合せ爆発を抑えて実時間で有効な解を得ることは困難と考えられる。そこで、以下に述べる 2 つのアイデアを適用することで解探索のさらなる効率化を図る。

1. 問題の分割 本問題を「資源割当」と「時刻決定」の 2 つの部分問題に分割し、2 段階のステップで解を探索する。

まず Step1 では資源変数 (番線及び進路) の割当を実行し、各オペレーションの資

源変数の値が定まる度に、その資源が既に割当てられている他のオペレーションとの間で資源競合が生じないように各オペレーションの実行順序関係を動的に決定する。次に、Step2 では Step1 の探索結果に基づき時刻変数（開始及び終了時刻）の値を決定する。

Step1 において競合オペレーション同士の実行順序がもれなく定まることから、Step2 の時刻決定は一般的な PERT 計算 [92] と同等な問題に帰着できる。したがって、CP をそのまま適用する、あるいは既存の PERT 計算アルゴリズムを応用することで Step2 の解を効率良く求めることができる。

以上を整理した Step1 の処理フローの概略を図 5.4 に示す。

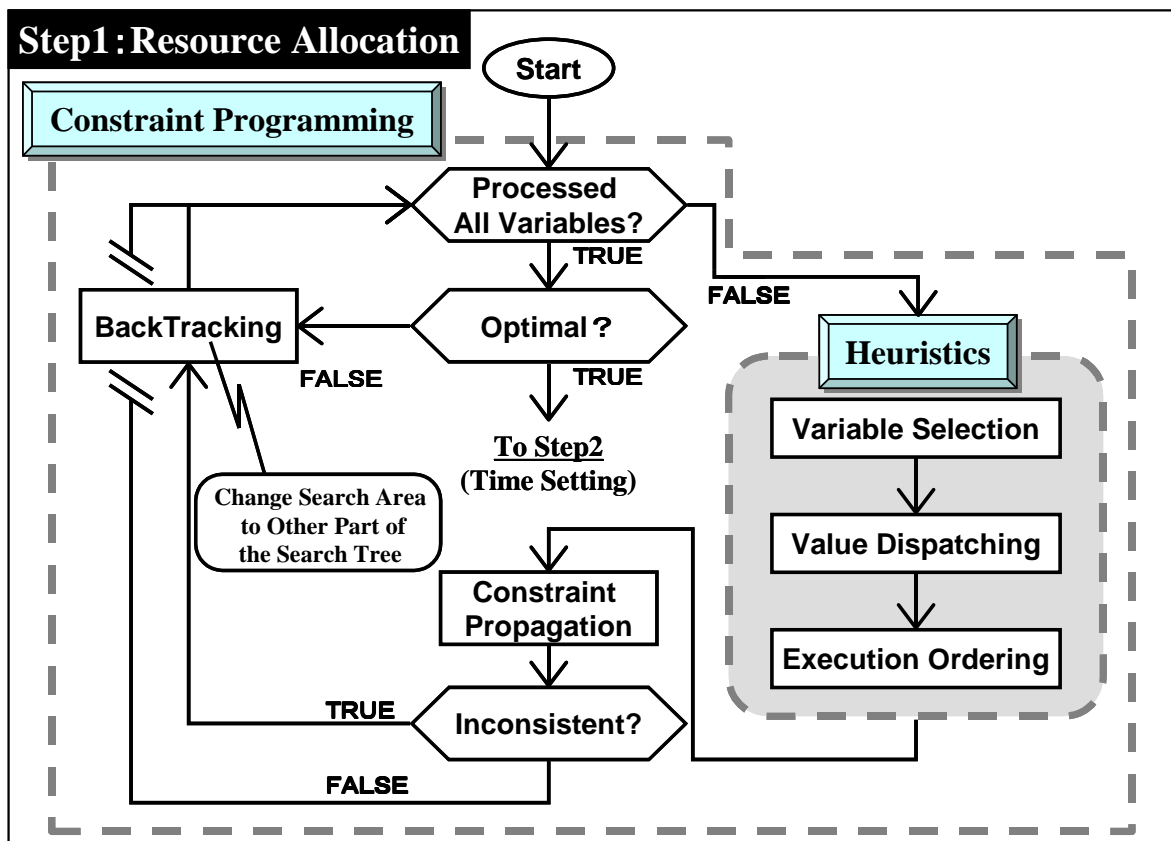


図 5.4: 資源割当 (Step1) の処理フロー

Fig.5.4: Flow Diagram of the Search Algorithm for Resource Allocation

2. 目的関数の単一化 本問題は複数の目的関数を同時に満足する多目的最適化問題として

定式化される(5.2.2節)。多目的な問題に対しては、複数の目的関数を線形加重和等で結合しひとつに纏めた上で解を求めるというアプローチが一般的である。ただし本問題は、バッファ資源の使用回数を0にすることが実行可能解を得る上での必須条件であるため、他の目的関数と結合してしまうとその条件を満足するように探索を制御することが難しくなる恐れがある。

そこで、元のスケジュールからの変更量の最小化に関する2つの目的関数(式(5.11)、式(5.12))を定式化から外し、解探索アルゴリズムは「バッファ資源使用回数最小化」(式(5.10))だけを目的関数として扱う。そして、残りの2つの目的関数の最小化は、元のスケジュールからの変更量を考慮しながら効率良く探索を進めるためのヒューリスティクスを解探索アルゴリズムに埋め込むことで実現する。

ヒューリスティクス

資源割当をおこなう Step1 は、資源変数の値を定めるだけでなく、オペレーションの実行順序も決定する必要がある。そのため、CPの制約伝播によるドメインの絞込みを考慮しても探索すべき解空間は非常に大きなものとなる。そこで、実行可能かつ元のスケジュールからの変更量が少ない有望な解を探索の早い段階で得ることを狙いとして、探索木を辿る際に実行される「資源変数の選択」、「値の決定」、並びに「実行順序付け」に対して、上記の狙いを実現するためのルール、すなわちヒューリスティクスを適用して処理方式を構成する(図5.5)。

すなわち、ヒューリスティクスはスケジュール作成の対象としてVSU(元のスケジュールを引き継ぐ)とISU(白紙から作成)という2種類の車両が存在することを踏まえ、次に示す3つの方針に沿って解探索を方向付けするために用いられる。(1)VSUのスケジュールは元のスケジュールとできるだけ一致する。(2)ISUのスケジュールはVSUのスケジュールと資源競合をできるだけ生じない。(3)無駄な領域の探索や探索の途中失敗が極力発生しない。

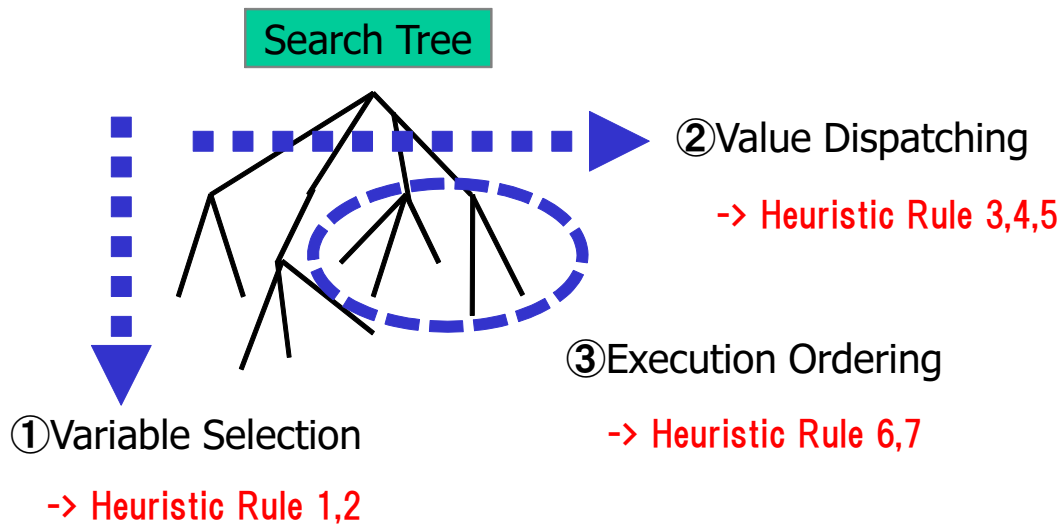


図 5.5: ヒューリスティクス
Fig.5.5: Heuristics

方針(1)は、元のスケジュールに準拠すべきVSUのスケジュールを変更量を抑えて生成することを意味する。一方、方針(2)と(3)は探索効率の向上を意図したものである。方針(2)により資源競合が発生して部分スケジュールの再作成(5.3.2節)が発生する可能性を抑え、方針(3)により探索ツリー上でノードを辿る回数を低減する。

以下、「資源変数の選択」、「値の決定」、並びに「実行順序付け」の各処理におけるヒューリスティクスをルールとして整理して述べる。なお、以下では説明のわかりやすさのため、「VSU」及び「ISU」を、それらに含まれるオペレーション(移動と配置)、もしくはその資源変数と同一視することとする。また、各ルールの説明に付された角カッコ[]の中に関連する方針を示す。

(1) 変数選択

- (ルール1) VSUとISUの双方が存在する場合にはVSUを先に選択する。これはVSUの資源割当と実行順序付けの結果に基づき、それらと干渉しないようにISUの探索をおこなわせるためである。[方針(1)(2)(3)]

- (ルール2) 対象がISUだけの場合は、まず番線配置の資源変数を先に選択し、その後移動の資源変数を選択する。この理由は、配置先の番線を全て決定すれば番線間の移動に用いる進路は一意に定まるため、番線配置を優先することで無駄な領域の探索を省略できるからである。[方針(3)]

(2) 値決定

- (ルール3) VSU に対しては元のスケジュールと同じ値を常に割当てて。[方針(1)]
- (ルール4) ISU に対しては、バッファ資源(0)を用いた解は実行可能解とはならないため、できるだけバッファ資源を割当てないこととし、他の資源が候補にない場合に限りバッファ資源を選択する。[方針(2)]
- (ルール5) ISU に対しては、バッファ資源以外の候補については、割当てにより競合を生じる可能性を各候補毎に推定し、その推定値が最も小さいものを選択する。
[方針(3)]

ある資源 a を選択した場合の競合可能性の推定方法は以下の通りである。まず、その時点で資源 a が既に割当てられているオペレーションを全て取得し、選択対象の資源変数のオペレーションと上記の各オペレーションとの間で時間的な重複部分を算出し、その合計を推定値とする。なお、この時点ではオペレーションの時刻変数の値は確定していないが、取り得る値の下限值と上限値は存在し、探索過程の中でCPの制約伝播により動的に変化する(5.3.2節)。そこで、開始時刻変数の下限値(開始可能な最も早い時刻)が開始時刻に、終了時刻変数の上限値(終了可能な最も遅い時刻)が終了時刻にそれぞれ等しい、すなわち両者の実行時間が最大であると仮定してその重なりを求める。

(3) 実行順序付け

- (ルール6) 2つのオペレーション j, j' がどちらも VSU であれば、元のスケジュールを参照してそれと同じ実行順序、すなわち $j \rightarrow j'$ もしくは $j' \rightarrow j$ を設定する。[方針(1)]

ここで、 $j \rightarrow j'$ は「オペレーション j の終了後にオペレーション j' を実行する」ことを意味し、 $j' \rightarrow j$ はその逆を意味する。CP の枠組みにおいてこれを実現するためには、例えば $j \rightarrow j'$ の場合であれば、 $(e_j + d \leq s_{j'})$ なる制約式を動的に設定すれば良い(ここで d は任意の間合い時間)。

- (ルール7) それ以外の組合せの場合は、実行順序 $j \rightarrow j'$ と $j' \rightarrow j$ のそれぞれについて、実行順序を満足するために必要なオペレーションの時間シフト量を算定し、その値が小さい方の順序を選択する。[方針(3)]

例えば図 5.6 の場合、2つのオペレーション j 及び j' が重複しないように時間後方へシフトする量を、 j をシフトする場合 ($j' \rightarrow j$) と、 j' をシフトする場合 ($j \rightarrow j'$) の双方について計算し、その値が小さい $j \rightarrow j'$ を採用する。ただし、(2) の値決定と同様に、この時点ではオペレーションの時刻変数の値は確定していないため、時刻変数の上下限值(開始時刻の下限値と終了時刻の上限値)を仮に用いて計算をおこなうこととする。

なお、上記の実行順序付けに伴う制約式設定により制約伝播処理が実行されるが、その結果、矛盾が発生する、すなわち値が未だ定まっていない決定変数のドメインが空となる可能性がある。その場合には、最初に設定した実行順序とは逆順の制約式を生成して制約設定を再びやり直す。

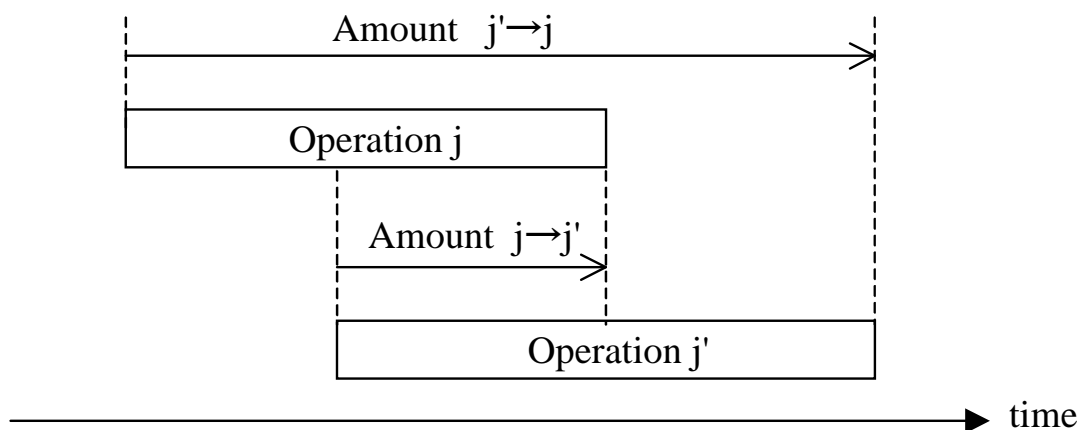


図 5.6: 時間シフト量の計算
Fig.5.6: Calculation of Time Shifting Amount

5.4 数値実験

5.4.1 プロトタイプシステム

提案方式を計算機上に実装し、プロトタイプシステムを開発した(図 5.7)。本プロトタイプは、作成結果をグラフィック表示する静的な機能だけでなく、車両の入区時刻等の変更情報を外部から取得したり、スケジュール作成担当者からのスケジュール修正や条件の追加変更指示等を受け付けて再スケジュールを実行する動的な機能を備えている。組合せ解探索アルゴリズムの実装には、仏 ILOG 社の CP ライブラリである ILOG Solver を使用した。これはオブジェクト指向に基づくクラスライブラリであり、提供されているソフトウェア部品(クラス)を継承することで、対象問題に合わせた探索処理のカスタマイズを比較的容易に行うことができる。

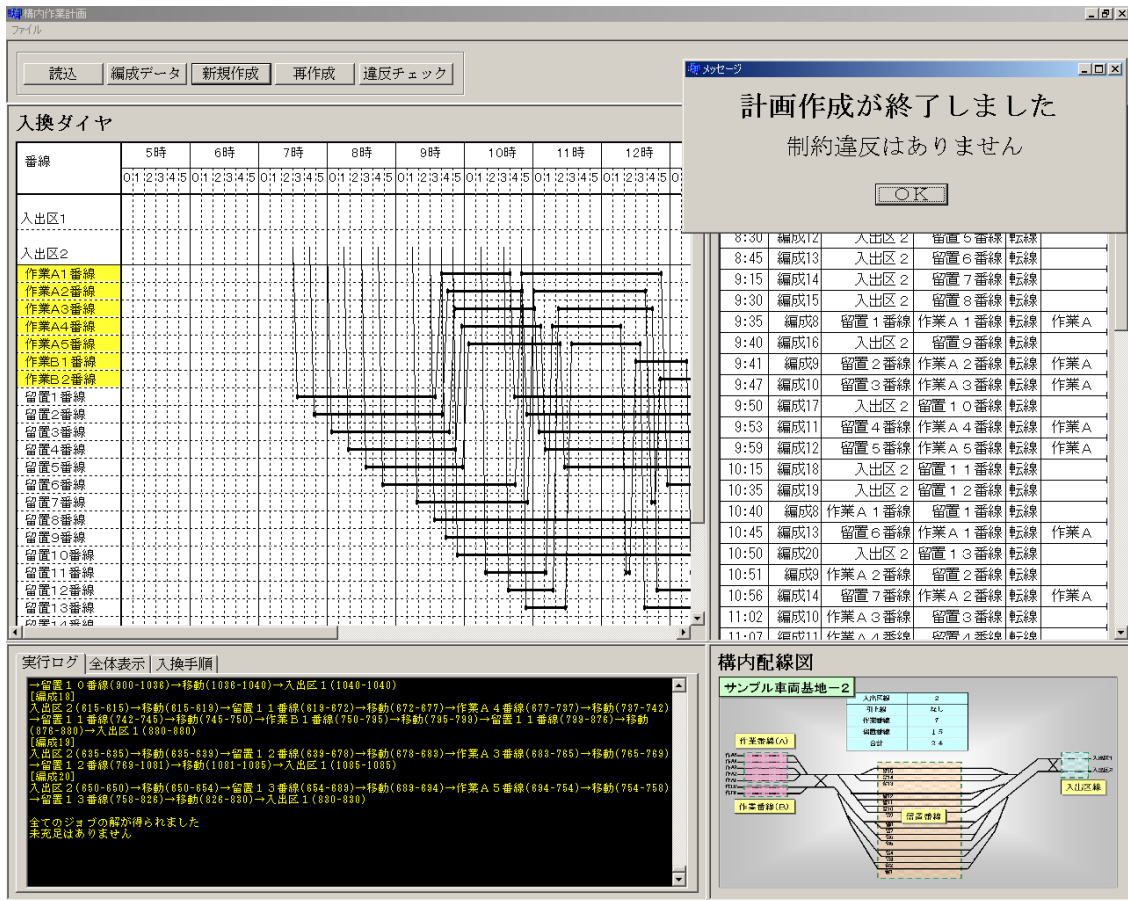


図 5.7: プロトタイプシステム
Fig.5.7: Prototype System

5.4.2 実験条件

実験データ

適用対象の車両基地のレイアウトを図 5.8 に示す。これらはいずれも仮想的な車両基地であり、実際の車両基地の特徴をふまえてレイアウト形状を定めた。また、実験結果の比較を容易にするため、番線の総数とそれらの種類（作業、待機等）の内訳はほぼ同じとし、中程度の規模の車両基地に相当するものとした（表 5.1）。

これらの車両基地はレイアウト形状に特徴的な違いがある。

車両基地 A は、作業用の番線群（レイアウトの左側）と待機用の番線群（レイアウト

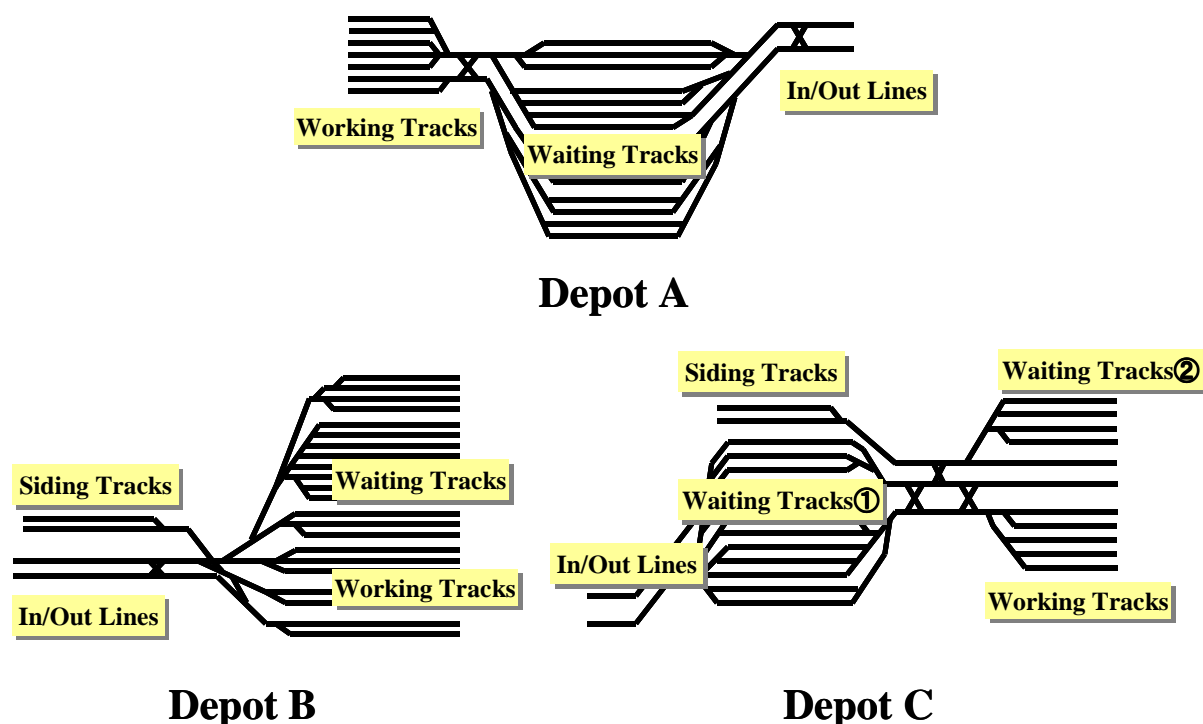


図 5.8: 実験対象の車両基地
Fig.5.8: Depots for the Numerical Experiments

の中央)が直列に配置されている。そして2本の入出区線(レイアウトの右側)が本線の駅と接続している。本線から入区する車両は、まず待機番線に留置され、作業を実施するために作業番線へ移動する。作業終了後は再び待機番線に戻り、所定の時刻に入出区線を通り駅へ出区する。複数の作業を行う場合は、待機番線からさらに他の作業番線へ移動する。このレイアウトの特徴は、待機番線に留置された車両が左右両方向へ移動可能な点である。この特徴を持つレイアウト形状を「両サイドオープン型」と呼ぶ。

車両基地Bは、作業番線群(レイアウトの右下)と待機番線群(レイアウトの右上)が並列に配置されており、それらが2本の入出区線(レイアウトの左下)と直接結ばれている。よって駅から入区した車両は、作業番線と待機番線のどちらへも直接移動することができる。そこで、出区迄に時間的な余裕があれば待機番線にいったん待機させ、時間的余裕がなければ作業番線に直接留置させるといった運用が一般的におこなわれている。番線間の移動は引上線(レイアウトの左上)を介しておこなう。例えば、複数の作業を

続けて行う場合は、最初の作業を実施した番線から引上線に一旦移動し、さらに次の作業をおこなう番線へ移動させる。両サイドオープン型と異なり、このレイアウトは全ての番線の移動方向が左右どちらかに限定されている。そこでこの特徴を持つレイアウト形状を「片サイドオープン型」と呼ぶ。

最後に、車両基地Cは、上記「両サイドオープン型」と「片サイドオープン型」の両方の特徴を備えたレイアウト形状である。左右両方向に移動できる作業番線群（レイアウトの中央下）と、一方向のみ移動が可能な引上線（レイアウトの中央上）の両方が存在し、どちらを用いても並列に配置された作業番線群（レイアウトの右下）あるいは待機番線群（レイアウトの右上）へ移動することができる。この特徴を持つレイアウト形状を「複合型」と呼ぶ。

番線数などの規模に大小の違いはあるものの、実際の車両基地の多くは上記3種類のレイアウト形状のいずれかに分類することができる。

表 5.1: 各車両基地の番線数と番線種類

Table 5.1: Number and Kind of Tracks at Each Experimental Depot

	Depot A	Depot B	Depot C
In/Out	2	2	2
Siding	0	2	2
Waiting	15	15	16
Working	7	7	7
Total	24	26	27

スケジュール再作成の対象である各車両の入出区時刻と作業予定内容は全車両基地共通である。朝ラッシュ時の営業運転を終了した複数の車両が車両基地に一旦戻り、必要な作業を実施した後に夕刻の営業運転のために再び車両基地を出て行くという、典型的な運用パターンを想定してそれらのデータを作成した。スケジュール再作成の時間枠は上述のとおり昼間の作業を対象とした約12時間、車両数はこの規模の車両基地では標準

的な13、作業種類は4種類、作業の総数は16である。なお、上記より各編成毎の作業数は平均すると一日あたり1個強（作業総数16/車両数13）となるが、これは一定間隔で実施が義務付けられている定期的な検査や、数日周期で実施される清掃作業、突発的な故障の修繕等をおこなう臨時の検査といった実際の作業の実施頻度に沿った値である。その他、作業や移動の所要時間、在線時間の最小値、作業可能な番線や車両の長さに応じた収容可能な番線等の各種基礎条件に関しても、全ての車両基地ができるだけ同じとなるように設定した。

実験対象データのジョブ数は車両数と同じ13、オペレーション数は129である。オペレーションは作業だけでなく入出区や留置番線の配置、番線間の移動も該当することから、ジョブ数と比較して数が多い。FJSSPの近似解法に関する理論的な研究[87]では、規模が異なる複数のデータセットを対象として近似解法の評価をおこなっているが、ここではオペレーション数が100以上の問題を平均以上の規模と位置付けている。これより、対象データはFJSSPとしてもnon-trivialな問題と考えられる。

目的関数

5.2.2節で定義したとおり、目的関数は以下の3種類である。

(1) バッファ資源使用回数最小化

$$\min \sum_{i=1}^m C_{iz_i}$$

(2) 番線変更回数最小化

$$\min \sum_{i \in O^T} \mu(i)$$

(3) 移動順序変更回数最小化

$$\min \sum_{i=1}^{m'-1} \phi(i, \delta, \delta')$$

実験概要

「白紙からのスケジューリング」と「再スケジューリング」の2通りの実験をおこなった。

「白紙からのスケジューリング」は、提案スケジューリング方式の基本な性能を評価するものである。3種類の車両基地の実験データに対して白紙からのスケジュール作成を実行し、実行可能解生成の有無と処理時間を測定する。なお、5.3.1節で示したように、白紙からのスケジュール作成は、図5.3の全体処理において全ての車両を初期状態でISUに設定することで実行可能である。

一方、「再スケジューリング」は本研究の目的であるリアクティブスケジューリングの観点から提案方式を評価する。前述の「白紙からのスケジューリング」にて作成したスケジュールを実施中に列車運行乱れや車両基地構内の作業遅れが発生し、スケジュールの再作成が必要となったケースを想定する。運行乱れを生じた車両や、作業の遅れ時間等をパラメータとし、それらを変えることで各ケース毎に複数の運用データを生成した。それらに対してスケジュール再作成をおこない、処理時間や目的関数の値を測定する。

両者の実験条件、特に「再スケジューリング」については、実験結果の節でさらに詳しく説明する。

計算機環境

実験に使用した計算機は、CPU 3.6GHz、メモリ 2.0GBの一般的なPCである。

5.4.3 実験結果：白紙からのスケジューリング

全ての車両基地に対して資源競合のない実行可能な構内入換計画が得られた。図5.9に、両サイドオープン型のレイアウト形状である車両基地Aに対して白紙からのスケジュール作成を実施した結果を示す。

第5章 車両基地運用整理のためのリアクティブスケジューリング方式

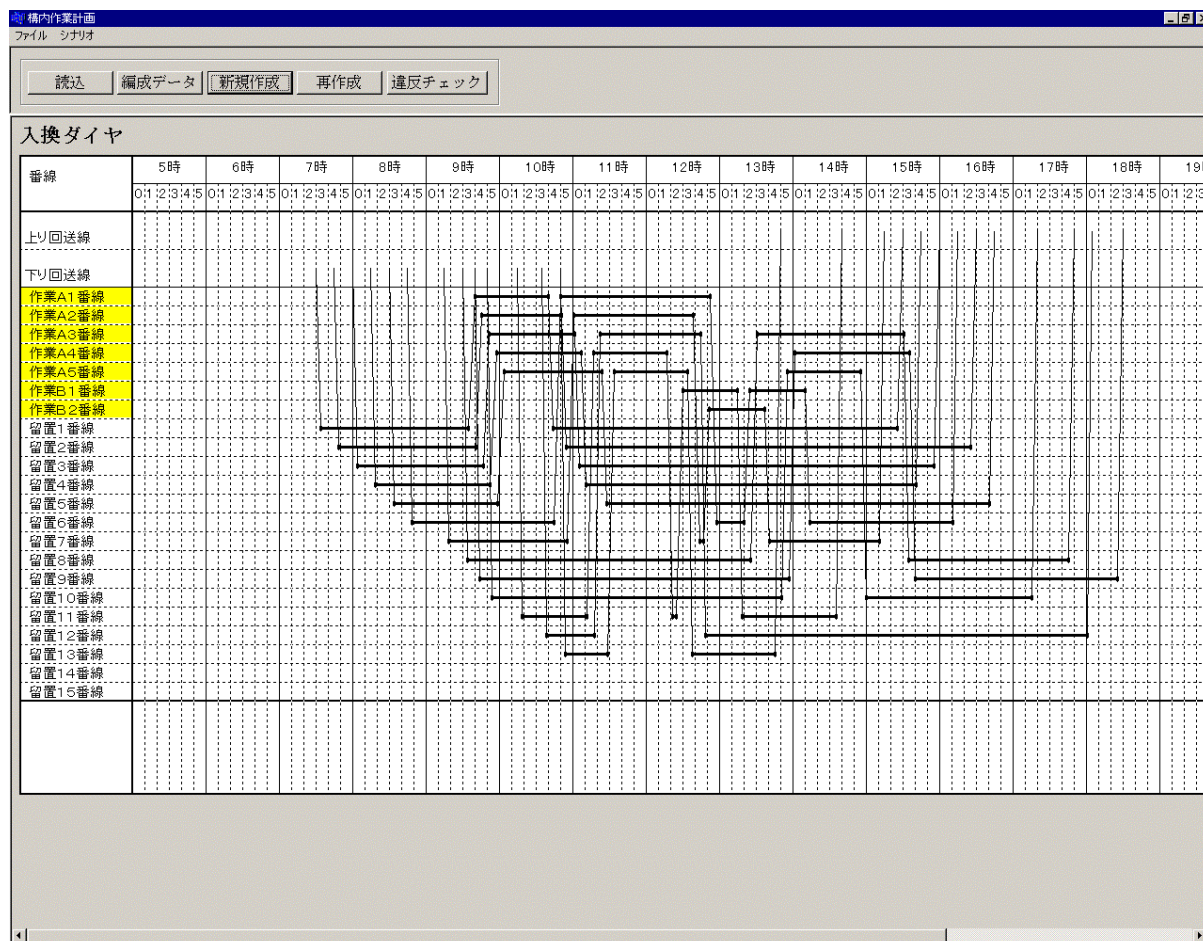


図 5.9: 白紙からのスケジュール作成結果 (両サイドオープン)

Fig.5.9: Scheduling Result from Scratch (Depot A)

表 5.2 は各車両基地毎のスケジュール作成の所要時間である。どの車両基地もほぼ同じであり、3 秒程度でスケジュールが得られた。

5.4.4 実験結果：再スケジューリング

提案方式を用いて白紙から作成したスケジュールを実施中に、列車運行乱れや車両基地構内の作業遅れが発生したという運用乱れのケースを想定し、再スケジューリングを実施した。

運用乱れのケースは以下の 3 種類とした。いずれも実際の車両基地において日常的に

表 5.2: 白紙からのスケジュール作成の処理時間 (秒)
Table5.2: Processing Time from Scratch (sec)

Depot A	Depot B	Depot C
3.10	3.42	3.24

発生する典型的なものである。

ケース1 作業遅れ 作業の完了時刻が予定より遅れる。

ケース2 入区遅れ 本線の列車運行乱れのため、車両基地への入区時刻が予定より遅れる。

ケース3 出区遅れ 作業の大幅な遅延や車両故障等の要因で、車両基地からの出区時刻が予定より遅れる。

図 5.10 は運用乱れのケースの一例である。これはケース2の「入区遅れ」を表している。複数の車両が一斉に入区遅延を生じていることが見て取れる。なお、この例において灰色のマスクが掛けられた領域は実績を表している。すなわち、マスク領域はスケジュールに沿って作業を実施済の部分である。それ以降の領域がスケジュール再作成の対象となる。

各ケース毎に、運行乱れを生じた車両や作業の遅れ時間といったパラメータを変えることで設定が異なる運用データを複数個生成した。データの生成方法は以下のとおりである。

(1) 作業遅れ スケジュールに含まれる作業を1つ選択し、5分刻みで最大60分の遅れを設定する。これを全ての作業を対象におこなう。

(2) 入区遅れ 入区が遅い順に車両を n 個選択し、その入区時刻に対して5分刻みで最大60分の遅れを設定する。これを n が1から全ての車両数まで変化させておこなう。

第5章 車両基地運用整理のためのリアクティブスケジューリング方式

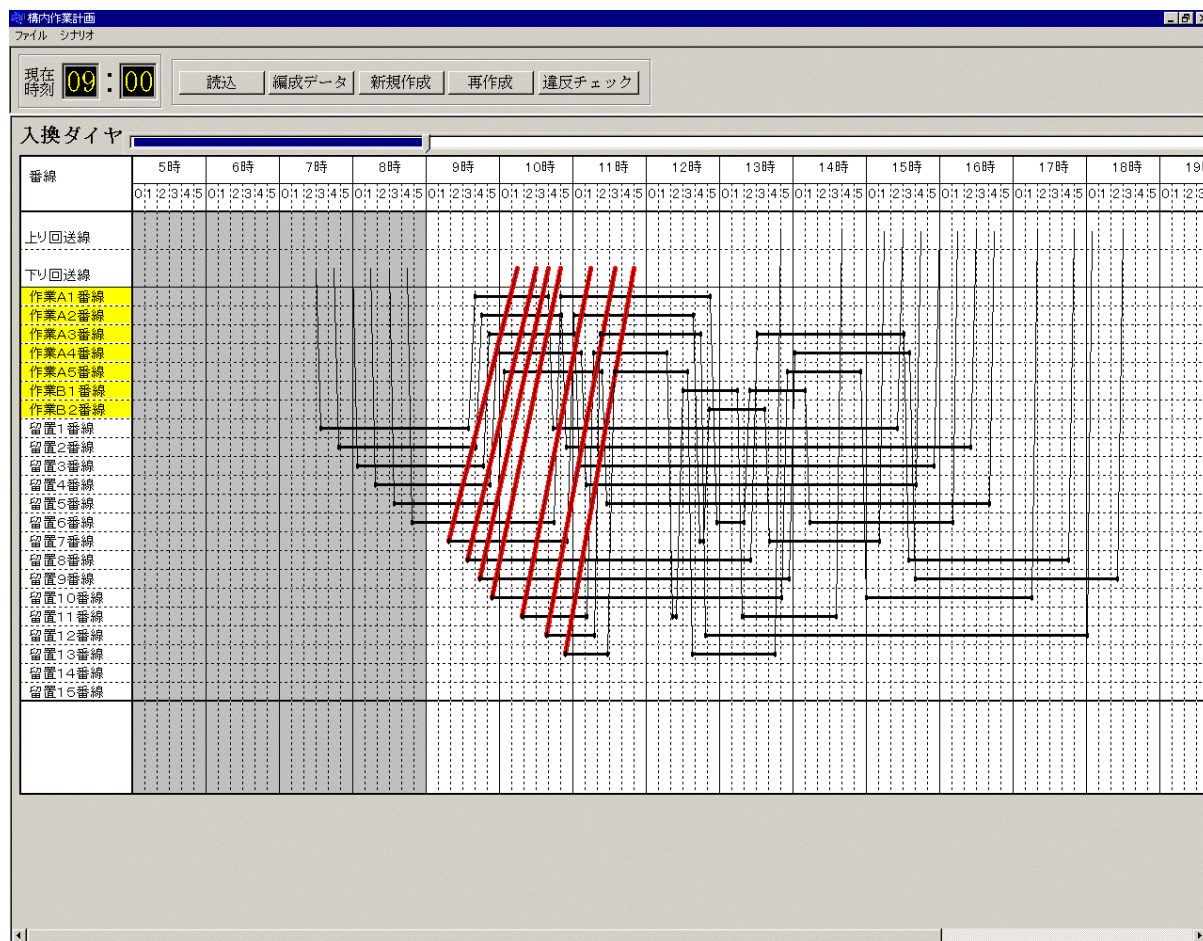


図 5.10: 運用乱れケースの例（入区遅れ）

Fig.5.10: An Example of Setting Changes in Operational Conditions (Incoming Delay)

- (3) 出区遅れ 出区を遅らせる車両を1つ選択する。これをAとする。Aよりも出区時刻が遅い車両を全て選択しグループ化する。これを全ての車両を選択しておく。

ケース(1)及び(2)は遅れの最大幅を一律に60分としたが、実際には列車運行の大規模乱れ等の要因でそれ以上の大幅な遅れが発生する可能性がある。ただし、そうした場合には車両の出区時刻を単純に遅らせることで作業に必要な時間を確保できることから、「出区を遅らせること無しにスケジュール再作成が可能な最大の遅れ時間」として、予備実験の結果を踏まえつつこの値とした。

一方、ケース(3)は出区遅れが避けられない状況を想定したものである。出区が遅れる

と通常は駅の始発時刻も遅れることになるが、本線の列車運行は車両基地の作業よりも優先度が高いことから、車両基地構内の遅れの影響が本線へ波及しないことが最も重視される。そこで、出区予定時刻が遅い他の車両と出区時刻の交換を試みることで、スケジュール再作成の影響を車両基地構内に限定し、本線に波及しないようにすることが一般的に行われている。ケース(3)はこの運用を想定したものである。出区を遅らせる車両に対して、それと可能な交換相手を全て抽出したものを1個の運用データとする。このデータを用い、交換相手の車両を順次選択して出区予定時刻を交換した上でスケジュールの再作成をおこなう。その結果、交換可能、すなわち実行可能なスケジュールが得られる車両があればスケジュール再作成を成功とみなす。

なお、各ケース共に、スケジュールの中で制約条件違反が生じた箇所から時間的に最も早いものを取り出し、一律にその10分前からスケジュール再作成をおこなうこととした。すなわちその時刻以前が実績領域となる。

運用データの数は、ひとつの車両基地あたり360個である。その内訳は、(1)作業遅れが192個、(2)入区遅れが156個、(3)出区遅れが12個、である。

3種類の車両基地に対してそれぞれ同じ数だけ運用データがある。したがって、運用データの総数は上記の3倍の1080個である。

実験結果

上記の条件の下で数値実験をおこなった結果を表5.3に示す。表5.3は、スケジュール再作成が成功した運用データの数とその全体に対する比率を示している。ここで、スケジュール再作成が成功とは、バッファ資源の使用回数が0の解が得られたことを意味する。

3種類の車両基地のいずれに対しても90%を超える高い割合でスケジュールの再作成に成功した。全データをまとめると、スケジュール再作成の成功率は約98%である。特に車両基地Aについては、全運用データに対してスケジュール再作成に成功する結果が得られた。

表 5.3: 実験結果 (再スケジュールの成功回数と比率)
Table5.3: Experimental Results (Schedule Feasibility)

	Num. of Success	Success Ratio (%)
Depot A	360	100.0
Depot B	341	94.7
Depot C	353	98.1
Total	1054	97.6

表 5.4: 実験結果 : 全車両基地
Table5.4: Experimental results : All Depots

All Depots & All Data	Avg.	Std. Dev.	Max	Min
Processing Time (sec)	0.16	0.80	10.83	0.02
Num. of Track Changes	0.06	0.43	7.00	0.00
Track Change Ratio (%)	0.09	0.61	9.86	0.00
Num. of Move Order Changes	1.57	1.93	9.00	0.00
Move Order Change Ratio (%)	2.70	3.33	15.52	0.00

表 5.4 は、スケジュール再作成に要する処理時間と2つの目的関数の値を、全車両基地の全運用データについて集計したものである。2つの目的関数とは、「番線変更回数」と「移動順序変更回数」である。なお、目的関数については値だけでなく比率も示す。番線変更回数の比率とは、スケジュールに含まれる全ての配置オペレーションの中で番線が変更されたものの割合である。これを以降では「番線変更比率」と呼ぶ。同様に、移動順序の変更比率とは、スケジュールに含まれる全ての移動オペレーションの中で順序が変更されたものの割合である。これを以降では「移動順序変更比率」と呼ぶ。

スケジュール再作成においては、乱れにより不具合を生じた車両を ISU に設定し、残

りの車両はVSUとして元のスケジュールをできるだけ保持するように解探索をおこなうことから、白紙からのスケジューリングの場合よりもさらに高速な処理が期待できる。処理時間の平均は0.16秒、標準偏差は0.80秒であった。すなわち、データの約7割は平均値から1標準偏差離れた約1秒未満、9割以上は2標準偏差離れた約2.0秒未満で再作成を実行した。

また、処理時間の最大値は10.83秒である。これは白紙からのスケジュール作成の3倍程度の処理時間であった。このデータ以外にも処理時間が平均値を大幅に超えるデータがいくつかあったが、それらはいずれもスケジュール再作成に失敗したものであった。提案方式は車両を追加しながらスケジュールの部分的な作成を試行錯誤的に繰り返すが、再作成に失敗する場合は必然的に繰り返し回数が増加するため処理時間を要したと推察される。

同じく表5.4より、全体として番線及び移動順序の双方共に少ない変更回数でスケジュール再作成をおこなっていることが伺える。番線変更回数は平均値が0.1を下回っており、変更比率は0.1%未満である。また、番線変更回数が0回のデータは1046個で全体の96.9%を占める。これより、ほとんどのデータに対して番線の変更なしにスケジュール再作成が実行できている。

一方、移動順序の変更回数は平均値が1.57回である。これを比率に換算すると2.7%である。移動順序の変更回数が平均値1.57の3倍以内、すなわち変更回数が4回以下のデータは984個であり、全体の91.1%を占める。これより、移動順序に関しても少ない変更回数でスケジュール再作成をおこなうことができた。

次に、車両基地A、B、C毎の個別の実験結果を表5.5、5.6、5.7にそれぞれ示す。これらは表5.4と同じく、スケジュール再作成に要する処理時間と2つの目的関数の値を集計したものである。全運用データの集計結果だけでなく、3種類の運用乱れケース毎の集計結果も示す。いずれの結果も特定の車両基地や特定のケースに大きく偏ることなく、全体の実験結果と同様の傾向を示している。

図5.11は、横軸に番線変更回数、縦軸に移動順序変更回数を取り、実験結果の分布を

表 5.5: 実験結果：車両基地 A (両サイドオープン型)

Table 5.5: Experimental results : Depot A

Depot A		Avg.	Std. Dev.	Max	Min
Case 1 Work Delay	Processing Time (sec)	0.05	0.11	1.59	0.02
	Num. of Track Changes	0.01	0.10	1.00	0.00
	Track Change Ratio (%)	0.01	0.14	1.41	0.00
	Num. of Move Order Changes	1.83	1.51	7.00	0.00
	Move Order Change Ratio (%)	3.16	2.59	12.07	0.00
Case 2 Incoming Delay	Processing Time (sec)	0.17	0.08	0.44	0.03
	Num. of Track Changes	0.05	0.22	1.00	0.00
	Track Change Ratio (%)	0.07	0.31	1.41	0.00
	Num. of Move Order Changes	3.46	2.15	8.00	0.00
	Move Order Change Ratio (%)	5.96	3.71	13.79	0.00
Case 3 Outgoing Delay	Processing Time (sec)	0.05	0.01	0.06	0.05
	Num. of Track Changes	0.00	0.00	0.00	0.00
	Track Change Ratio (%)	0.00	0.00	0.00	0.00
	Num. of Move Order Changes	2.08	0.51	3.00	1.00
	Move Order Change Ratio (%)	3.59	0.89	5.17	1.72
All Cases	Processing Time (sec)	0.10	0.12	1.59	0.02
	Num. of Track Changes	0.03	0.16	1.00	0.00
	Track Change Ratio (%)	0.04	0.23	1.41	0.00
	Num. of Move Order Changes	4.39	3.38	13.79	0.00
	Move Order Change Ratio (%)	2.54	1.96	8.00	0.00

表 5.6: 実験結果：車両基地 B (片サイドオープン型)

Table 5.6: Experimental results : Depot B

Depot B		Avg.	Std. Dev.	Max	Min
Case 1 Work Delay	Processing Time (sec)	0.37	1.84	10.83	0.03
	Num. of Track Changes	0.08	0.37	3.00	0.00
	Track Change Ratio (%)	0.12	0.53	4.23	0.00
	Num. of Move Order Changes	0.42	1.01	5.00	0.00
	Move Order Change Ratio (%)	0.72	1.75	8.62	0.00
Case 2 Incoming Delay	Processing Time (sec)	0.16	0.07	0.34	0.03
	Num. of Track Changes	0.00	0.00	0.00	0.00
	Track Change Ratio (%)	0.00	0.00	0.00	0.00
	Num. of Move Order Changes	0.00	0.00	0.00	0.00
	Move Order Change Ratio (%)	0.00	0.00	0.00	0.00
Case 3 Outgoing Delay	Processing Time (sec)	0.08	0.04	0.19	0.05
	Num. of Track Changes	0.33	0.65	2.00	0.00
	Track Change Ratio (%)	0.49	0.97	3.03	0.00
	Num. of Move Order Changes	2.25	0.45	3.00	2.00
	Move Order Change Ratio (%)	3.90	0.77	5.17	3.45
All Cases	Processing Time (sec)	0.27	1.35	10.83	0.03
	Num. of Track Changes	0.06	0.30	3.00	0.00
	Track Change Ratio (%)	0.08	0.43	4.23	0.00
	Num. of Move Order Changes	0.51	1.47	8.62	0.00
	Move Order Change Ratio (%)	0.30	0.85	5.00	0.00

表 5.7: 実験結果：車両基地 C (複合型)
Table5.7: Experimental results : Depot C

Depot C		Avg.	Std. Dev.	Max	Min
Case 1 Work Delay	Processing Time (sec)	0.08	0.38	5.16	0.03
	Num. of Track Changes	0.15	0.75	4.00	0.00
	Track Change Ratio (%)	0.21	1.06	5.63	0.00
	Num. of Move Order Changes	1.79	2.21	9.00	0.00
	Move Order Change Ratio (%)	3.09	3.81	15.52	0.00
Case 2 Incoming Delay	Processing Time (sec)	0.16	0.08	0.52	0.03
	Num. of Track Changes	0.00	0.00	0.00	0.00
	Track Change Ratio (%)	0.00	0.00	0.00	0.00
	Num. of Move Order Changes	1.89	1.71	6.00	0.00
	Move Order Change Ratio (%)	3.26	2.94	10.34	0.00
Case 3 Outgoing Delay	Processing Time (sec)	0.06	0.04	0.17	0.05
	Num. of Track Changes	0.58	2.02	7.00	0.00
	Track Change Ratio (%)	0.82	2.85	9.86	0.00
	Num. of Move Order Changes	2.58	1.68	7.00	1.00
	Move Order Change Ratio (%)	4.45	2.89	12.07	1.72
All Cases	Processing Time (sec)	0.11	0.28	5.16	0.03
	Num. of Track Changes	0.10	0.66	7.00	0.00
	Track Change Ratio (%)	0.14	0.94	9.86	0.00
	Num. of Move Order Changes	3.21	3.43	15.52	0.00
	Move Order Change Ratio (%)	1.86	1.99	9.00	0.00

示したものである。番線変更が1回以下のものが全データの99%を占める。変更回数
 双方共に多いはデータはきわめて少ない。番線変更回数が1回以下で、移動順序変更回
 数が4回以下のデータは981個である。これは全データの90.8%である。また、番線と移
 動順序の変更が共に0回のパターンは42%あるが、これは番線も移動順序も変更せず
 時刻の微調整のみで再作成が済んだものである。以上より、本提案方式により元のスケ
 ジュールからの変更量が少ないスケジュール再作成を安定的に実行することができた。

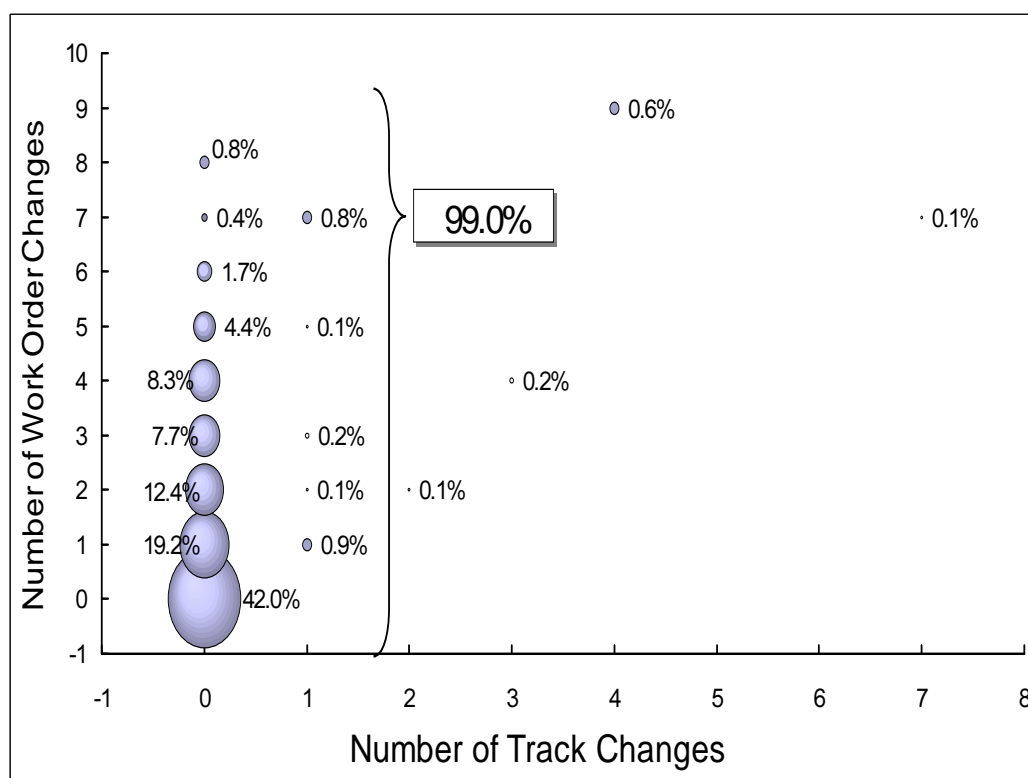


図 5.11: 再スケジューリング結果の変更量の分布

Fig.5.11: Range of Rescheduling Results in terms of the Change Amount

5.5 考察

5.5.1 再作成の効率

数値実験結果より、本提案方式は秒単位の時間オーダー（0.02～1.59秒）でスケジュールの再作成が可能であることを示した。車両基地運用整理は、列車運行の乱れや車両基地構内の作業遅れといった突発的な事象に即応的に対処する必要があることから、スケジュールの再作成は早ければ早いほど望ましい。上記の結果から本提案方式はその要件を満たしているといえる。

なお、本研究では番線数が30未満の中規模車両基地を対象として数値実験をおこなったが、実存する最大の車両基地であっても番線数は高々100を超えないことから、本提案方式はレイアウト規模の増加に十分対応可能との感触を得ている。

ただし、規模が増大すれば車両基地における運用が多様化・複雑化する懸念がある。例えば大規模な車両基地の中には、ひとつの番線に複数の車両を同時に配置するところも存在する（これを番線の区分使用と呼ぶ）。その場合は車両が番線に出入りする順番を考慮しながら衝突が発生しないように車両配置を決定する必要がある。こうした運用の多様性や複雑性への対応は今後の課題である。

5.5.2 評価指標の多様化

車両基地運用整理において最も重要な評価指標は、元のスケジュールからの変更量の最小化である。ただし、それ以外にも資源の有効活用や作業の効率化等を目的とした様々な評価指標がある。例えば「作業者の移動時間を最小化する」といったものである。本提案方式が秒単位の時間オーダーでスケジュールの再作成が可能であることから、そのような評価指標は、プロトタイプに実装したようなGUIを介して再作成の担当者がシステムと対話的なプロセスを素早く繰り返すことで充足できると考えられる。元のスケジュールからの変更量以外の評価指標も考慮したスケジューリング方式の実現は今後の課題で

ある。例えば、多様な評価指標を柔軟に扱うことが可能で、CPとの親和性も高いGA等のメタ解法[18][96]を本提案方式に組み込む、あるいは本提案方式と組み合わせる、といったアプローチが考えられる。

5.5.3 ジョブ定義の変更

ジョブ定義とは、一個の車両が車両基地に入区してから出区するまでの移動と配置の一連の繰り返しのことである(5.2.1節参照)。5.2.2節で述べたように、本問題を定式化するにあたっては各車両のジョブ定義が事前に与えられることを前提とした。しかし、実際の車両基地においてはジョブ定義は必ずしも固定されたものではなく、スケジュールの再作成においてジョブ定義自体を変更する必要がある場合がある。例えば、数値実験で説明した3種類の車両基地のレイアウト形状のうち、片サイドオープン型の車両基地(車両基地B)は、作業を終了した車両を作業番線から直接出区させることもできるが、引上線を介して待機番線へ移動し、そこで待機させた後に出区させることもできる。一般的に出区までの時間的余裕がある場合は、作業番線を空ける目的で待機番線へ移動させることが多い。ただしその場合でも、何らかの理由で出区が早まると、ジョブ定義を変更して待機番線への移動を取りやめなければ実行可能なスケジュールの再作成ができないことが起こり得る。数値実験の結果、全データの約2%にあたる26個のデータに対しては実行可能なスケジュールの再作成ができなかった。それらの大部分(19個)は車両基地Bのものであったことから、多くのデータは上記のようなジョブ定義の変更が必要であったと推察される。ジョブ定義を可変とすることで問題の複雑さがさらに増すため取扱いが困難であるが、スケジュール再作成の成功率をさらに向上するためには、本提案方式をジョブ定義の変更に対応することが求められる。

5.6 本章の総括

本章では、(課題2)に対処するための車両基地運用整理向けリアクティブスケジューリング技術について明らかにした。車両基地のレイアウトを作業に使用する資源と捉えることで、本問題は生産分野におけるフレキシブルジョブショップスケジューリング問題(FJSSP)の類似問題とみなすことができる。ただし、(1)車両基地の入区から出区までの時間枠が固定されるため、レイアウトを構成する資源(進路、番線)が車両の移動・配置の際に競合しやすい、(2)移動と配置を矛盾無く繰り返す上で、必要な資源の組合せがレイアウトに強い制約を受ける、(3)レイアウト上で時間軸に沿って移動と配置を連続的に実行する必要がある、等の制約から、配置時間の延長といった部分的な変更が他へ連鎖的に影響しやすいという特徴を持つ。これらの制約を記述可能な制約充足最適化問題としての定式化とそれにもとづく段階的なスケジュール作成のアーキテクチャを提案した。制約充足最適化問題の定式化はFJSSPを基本としつつ、(1)レイアウトの扱いを一般化した制約式を定義し、(2)実行可能性を判断するため競合を考えない特別な資源(バッファ資源)を導入し、(3)スケジュールの質を評価するため、「使用番線」と「移動の実施順序」に着目して最適化の評価指標である元のスケジュールからの変更量を定義した。次に、定式化に基づいたスケジューリング方式として、制約プログラミング(CP)を用いた方式を提案した。CPに探索の効率化と元のスケジュールからの変更量の最小化に寄与するヒューリスティクスを組み合わせることで方式を構成する。提案方式を組み込んだプロトタイプシステムを実装し、レイアウト形状が異なる3種類の車両基地を対象として数値実験を実施した。3通りの運用乱れケースを表す1080個の運用データを生成し実験をおこない、(1)秒単位の時間オーダー(0.02~1.59秒)で運用計画の再作成が可能であり、(2)全データの約98%についてバッファ資源を使用しない実行可能解が得られ、(3)番線変更回数は全データのうち96.9%が0回、(4)移動順序変更回数は平均値が1.57回で全移動回数の2.7%に相当し、その3倍以内のものが全データの91.1%を占め、(5)時刻の修正のみで再作成に成功したものが全データの42%であることを示した。

第6章

結論と今後の課題

6.1 結論

本論文は、鉄道輸送サービスのデペンダビリティ向上を目指し、列車運行乱れ時の列車運用整理業務に焦点をあて、最適化技法とヒューリスティクスとを組み合わせたリアクティブなスケジューリング方式を検討し、実路線を含むダイヤ再編成データを用いた数値実験により提案方式の有効性を評価した結果をまとめたものである。

第2章では、リアクティブスケジューリング技術に関する近年の研究開発動向をサーベイした結果を示した。

第3章では、列車運用整理業務を支援するシステム実現のための2つの課題を明らかにした。すなわち、列車運用整理業務を構成する要素である、旅客・貨物等を輸送する本線上の営業列車の運行に直接関わる「本線運用整理」と、車両基地における車両の清掃・点検といった保全作業に関わる「車両基地運用整理」のそれぞれに関し、解決すべき課題として以下を提示した。

(課題1) 本線運用整理に関し、再作成された列車ダイヤからの要求と資源運用制約の双方を可能な限り満たした上で、元のスケジュールからの変更量を最小化した再スケジューリング結果を算出可能な効率的なスケジューリング方式が必要である。

(課題2) 車両基地運用整理に関し、車両基地の様々なレイアウトに対応し、車両の移動・配置に関わる物理的な制約とその他の資源運用制約とを満足した再スケジューリング結果を算出可能な効率的なスケジューリング方式が必要である。

第4章では、(課題1)に対処するための本線運用整理向けリアクティブスケジューリング技術について明らかにした。リアクティブスケジューリング問題の構造や制約を陽に扱うためのネットワークフローモデルを定義し、そのモデルを用いて対象問題を整数計画問題として定式化した。そして定式化された問題の解を算出する2通りのスケジューリング方式を提案した。(方式1)は最適化技法としてラグランジュ緩和法を応用し、最適値の下界と上界とのギャップを用いた解の精度評価を行うことにより元のスケジュールからの変更量を最小化した再スケジューリングを求めるものである。(方式2)は最適化技法として局所探索法を応用したものであり、方式1では評価できない非線形な評価指標(全車両の折り返し時間の標準偏差と各車両の走行距離の差分の総和)にもとづく最適化を行うものである。本提案方式を組み込んだプロトタイプシステムを実装し、実路線の運行乱れ時の運用データを用いた数値実験をおこなった結果、方式1により、元のスケジュールからの変更量の最適値に対して、評価値の乖離が約1.6%以内の実行可能解を数十分のオーダの計算時間で算出できることを示した。さらに、方式2により、元のスケジュールからの変更量に加え、非線形な評価指標である全車両の折り返し時間の標準偏差と各車両の走行距離の差分の総和について、従来方式と比較して前者は約52%、後者は約33%、それぞれ改善できることを示した。最後に、異なる特徴を持つ2つの提案方式を組み合わせることでスケジューリング機能を構成することで、デベンダビリティをさらに向上する可能性を示した。

第5章では、(課題2)に対処するための車両基地運用整理向けリアクティブスケジューリング技術について明らかにした。車両基地のレイアウトを作業に使用する資源と捉えることで、生産分野におけるフレキシブルジョブショップスケジューリング問題(FJSSP)の類似問題として対象問題をモデル化した。そしてレイアウトの扱いを一般化した制約を記述可能な制約充足最適化問題としての定式化と、段階的なスケジュール作成のアーキテクチャを提案した。定式化された問題の解を算出するスケジューリング方式は制約プログラミング(CP)を応用したものであり、CPに探索の効率化と元のスケジュールからの変更量の最小化に寄与するヒューリスティクスを組み合わせることで方式を構成す

る。本提案方式を組み込んだプロトタイプシステムを実装し、レイアウト形状が異なる3種類の車両基地を対象として数値実験を実施した結果、3通りの運用乱れケースを表す1080個の運用データに対して(1)秒単位の時間オーダー(0.02~1.59秒)で運用計画の再作成が可能であり、(2)全データの約98%について実行可能解が得られ、(3)番線変更回数は全データのうち96.9%が0回、(4)移動順序変更回数は平均値が1.57回で全移動回数の2.7%に相当し、その3倍以内のものが全データの91.1%を占め、(5)時刻の修正のみで再作成に成功したものが全データの42%であることを示した。

6.2 今後の課題

現在、本線運用整理と車両基地運用整理の双方共に、本研究で提案したスケジューリング方式を組み込んだ実用化システムの研究開発を推進中である。

本線運用整理に関しては、本研究で実装したプロトタイプシステムをベースとして汎用パッケージソフトウェアを開発し、ある鉄道路線への適用を検討している。

車両基地運用整理に関しては、鉄道事業者の御協力のもと、実際の車両基地を対象としたプロトタイプシステムの実証実験に取り組んでいる[103][104]。

こうした鉄道輸送現場での実適用を通し、本研究成果に一定の有効性があることを実地に確認しつつある。例えば、ある車両基地では翌日のスケジュール作成に担当者が毎日1~2時間程度を要している。一般的に、翌日のスケジュール作成は基本スケジュールの修正で済むことが多く、その意味でスケジュールの再作成に良く似ている。列車運行の乱れや作業の遅れ等が発生した場合は、スケジュールの再作成が必要となるが、再作成に時間がかかることから実際にはスケジュールに基づかず目先のトラブルに場当たり的に対処することが多い。そのため、作業の輻輳や連絡漏れ等がしばしば発生して現場が混乱し、乱れの収束に時間がかかる事例がしばしば生じている。そこで、本研究成果を組み込んだ計画支援システムを用いることにより、再作成の素案を秒単位の時間で立案し、さらには元のスケジュールからの変更を抑えることで、乱れからの回復を素早く

かつ円滑におこなうことができる」と期待されている。

その一方で、鉄道輸送サービスのデペンダビリティをさらに向上するためには、システムの操作性改善といった従来課題だけでなく、以下に示す新たな課題を解決すべきことが明らかとなってきている。

情報収集と不確定情報の管理

スケジュールの再作成を実態に即した形でおこなうためには、列車運用整理で扱う様々な情報を適切に管理することが必要である。

例えば本線運用整理であれば、列車ダイヤ、車両・乗務員の現在位置、障害復旧時刻、車両基地運用整理であれば、列車の入区予定時刻、車両基地構内の車両在線状況、作業の遅れ時間といったものがそうした情報に該当する。これらの中で車両の現在位置に関しては、列車の走行を監視制御する運行管理システムが列車の在線位置を追跡しているので、列車番号と車両番号とを突き合わせることで最新情報を取得することが可能である（ただし本線上の位置のみ）。

しかしながら、その他の大部分の情報に関しては、オンラインでの取得手段が確立されていないのが現状である。例えば多くの鉄道路線では、輸送混乱時に乗務員の所在がわからなくなってしまう、駅や乗務員区所等と何度も電話連絡をすることで現在位置をようやく特定している。

乗務員の現在位置については、運転中や待機中といった乗務員の勤務状況に応じて複数の手段（ICカード、無線LAN、PHS）を組み合わせる位置を特定するシステムの開発が試験的に進められている [74]。また、鉄道に関する様々な情報システムが持つデータを一元的に管理するため、標準データ記述言語であるXMLを活用した統合情報管理モデルが提案されている [98]。こうした技術を活用することで列車運用整理に適した情報収集機能の実現が期待される。

また、列車ダイヤや車両基地への入区予定時刻は上述の運行管理システムが保持して

いるが、それらのデータが運行管理システムにおいて更新されるのは実際に変更をおこなう直前である。そのため、そのタイミングで運用整理システムがデータを取得しても運用整理をおこなう時間的余裕がないことが多い。障害復旧時刻や作業の遅れ時間についても同様である。それらが確定した段階で運用整理をおこなっても大抵は遅きに失することになる。そこで現在の人手での運用整理では、状況変化を先読みすることでこの問題に対処している。例えば、列車ダイヤの変更方針やその内容がある程度固まった段階でスケジュールを仮作成し、それを実際の変更内容に合わせて適宜修正しながら徐々に確定させていく、という方法が取られている。現在は人間系でおこなわれているこのような不確定な情報を適切に扱うことで、より実態に即したスケジューリングを実現する技術の確立が必要である。そのためには、不確定情報の下で最適化を扱う技法として、例えば確率計画法 [99] の応用等が考えられる。さらには、障害復時刻といった不確定な情報を精度良く推定する技術を確立し、不確定さの度合い自体を減らすことも今後の重要な課題である。

カスタマイズの容易化

本研究では、列車運用整理に関わる各種の資源運用制約や車両基地のレイアウトを汎用的に扱うことが可能なスケジューリング方式を提案した。それらのスケジューリングに関わる諸条件は、特定の鉄道路線に依存しない共通的なものであり、多くの鉄道路線に対して適用可能と考えられる。

しかし、例えば本線運用整理であれば、乗務員の労働規則は鉄道事業者毎に異なる上、例えば「長い休憩時間よりは、早く勤務を終了して帰宅させるといった要望を考慮する」といった明文化されていない暗黙のルールやノウハウも存在する [15]。車両基地運用整理であれば、一般的な分類から外れる変則的なレイアウト形状を持つ車両基地や、作業の安全性確保や効率化のため車両配置や作業順序に独自の運用ルールを持つ車両基地もある [103][104]。

したがって、特定の鉄道路線を対象として列車運用整理業務をシステム化する際には、共通的条件は当然漏れなく満たしつつも、このような個別条件にきめ細かく対応することが重要となる。この課題に対処するためには、本研究におけるスケジューリングの最適化というアプローチを土台にしつつ、そこに人間の知識を入れ込む形を目指すことが大切である。例えば人手による運用のパターン化や、ルールやノウハウの解探索メカニズムへの埋め込みといった様々な手段により、スケジューリング方式を適用対象に合わせてカスタマイズしやすくすることがひとつの方向性と考えられる。

本研究で示したとおり、列車運用整理はある種の最適化問題として扱うことが可能である。したがって、実用化システムの構築にあたっては、本質的にノーフリーランチ定理 [100] が示すところの「問題領域の知識を可能な限り使用して最適化すべき」を、個々の適用対象毎に地道に行うべきものといえる。しかしながら、エンジニアリングの観点からは個別化の労力はできるだけ減らすべきなのは言うまでも無い。情報システムの一般的な設計者や開発者であっても個々のシステム開発を容易におこなうことができるように、個別条件の抽出やカスタマイズに関わるガイドラインの策定、そのための支援ツールの整備、スケジューリング方式のカスタマイズ容易な形でのソフト部品化等が今後望まれる。

また、鉄道事業者や鉄道路線毎の個別性が強い現在の列車運用整理業務のあり方をみなおし、これを標準化することや、スケジューリング機能にインタラクティブ性を積極的に取り入れ、その代わりスケジュールの自動作成で扱う条件を限定するといった、ある種の「割り切り」を入れて個別化の必要性自体を減らすことも、複雑な列車運用整理業務のシステム化を目指す上でのひとつの重要な方向性と考えられる。

個別支援から全体支援へ

鉄道は、列車ダイヤを共通の情報として、様々な業務の担当者が分散協調的に職務を遂行していく使命を持った複雑なシステムと考えることができる [12]。

本研究では、列車運用整理業務の構成要素である本線運用整理と車両基地運用整理に対し、それらを支援するためのスケジューリング方式を提案したが、輸送サービスのデペンダビリティをさらに向上させるためには、この両者を密に連携させることで列車運用整理業務全体を効果的に支援する取り組みが今後重要になると考えられる。さらに視点を広げれば、列車運行乱れ時における他の業務である「列車運転整理」との連携や、より将来的には他の鉄道路線やバス・タクシーといった異なる輸送モード間の連携を実現することも求められるであろう。

これらの課題に対しては、統合最適化、多目的最適化、協調スケジューリングといった視点でのアルゴリズム研究がますます重要になると考えられる。ただし、従来よりも遙かに大規模で複雑な対象を扱う必要があることから、アルゴリズム研究からのアプローチだけでその解決を図るのは困難と考えられる。これは目的が異なる複数のシステムの連携や協調を扱う問題であり、システム工学分野におけるいわゆる System of Systems (SoS) 研究 [101] と密接な関わりがある。例えば SoS においては、近年、目的が異なる複数のシステムを連携させるための新しいシステムコンセプトとして、共生自律分散 [102] という考え方が提唱されている。こうしたシステム工学からのアプローチも取り入れながら、列車運行乱れ時における輸送業務・輸送モードの全体支援に向けて課題解決を図っていくことが重要である。

6.3 おわりに

地球規模で見れば今後も都市への人口流入が続き、いわゆるメガシティ（巨大都市）が世界中で増加すると予測されている [105]。この流れの中で、都市における生活の質向上、運営の効率化、そして省資源などの環境にも配慮したスマートシティと呼ばれる構想が提示され、その実現に向け情報技術 (IT) を活用した社会インフラの高度化が求められている [106]。スマートシティにおいては、環境負荷が小さく大量輸送が可能な唯一の交通機関として鉄道が果たす役割はきわめて大きい。また、鉄道大国である我が国は、スマートシティを支える鉄道システムの研究開発を推進する上で格好のフィールドである。情報工学を修得し企業研究者でもある筆者は、鉄道輸送サービスのさらなるデペンダビリティ向上を使命として、本研究の成果を礎とし、実フィールドでの経験を活かしながら引き続き研究開発に取り組んでいきたいと考えている。

謝辞

本論文は、筆者が1995年から現在に至るまで(株)日立製作所システム開発研究所、ならびに横浜研究所にて従事してきた鉄道分野における計画支援システムの研究開発業務での経験に基づき、2008年から現在まで早稲田大学大学院 情報生産システム研究科において行ってきた列車運用整理リアクティブスケジューリング技術に関する研究成果をまとめたものです。早稲田大学ならびに(株)日立製作所内外の多くの方々にご協力、ご支援を賜りました。ここに深く感謝の意を表します。

はじめに、本研究を推進するにあたり、全過程を通じて終始懇切なご指導とご鞭撻を賜りました、早稲田大学大学院 情報生産システム研究科 村田智洋 教授に心より感謝申し上げます。そして、本論文をまとめるにあたり有益なご助言とご討論を賜りました、早稲田大学大学院 情報生産システム研究科 吉江修 教授、李義頡 教授に謹んで深く感謝申し上げます。

本研究の機会を与えて与えて頂くと共に、暖かいご指導とご鞭撻を賜りました、(株)日立製作所 システム開発研究所 元所長 小坂満隆 博士(現 北陸先端科学技術大学院大学 知識科学研究科 教授)、元所長 前田章 博士(現 (株)日立製作所 インフラシステム社 CTO)、(株)日立製作所 横浜研究所 所長 堀田多加志 博士に心より御礼申し上げます。同じく本研究の機会を与えて頂くと共に、本研究を推進するにあたり技術の実用化の観点から高所よりご指導を頂き、また格別のご配慮を賜りました、(株)日立製作所 情報制御システム事業部 交通第二システム部 元部長 田原辰彦氏(現 (株)日立情報制御ソリューションズ 取締役)に厚く御礼申し上げます。元上司として本研究と本論文をまとめる動機付けと機会を与えて頂くと共に、高所からご指導とご鞭撻を賜りました、(株)日立製作所 システム開発研究所 船橋誠壽 博士(現 横断型基幹科学技術研究団体連合 理事)、佐々木敏郎氏、森田豊久氏(現 (株)日立システムズ 経営戦略統括本部 主管研究員)、(株)日立製作所 横浜研究所 本間弘一 博士、栗栖宏充 博士(現 (株)日立製作所 水環境ソリューション事業 統括本部 主任技師)、角本喜紀 博士(現 (株)日立製作所 インフラシステム社 経営戦略室 担当部長)に心から感謝申し上げます。

第4章に関して、研究の機会を与您と共、技術の実用化の観点から格別のご指導とご助言を頂きました、(株)日立製作所 インフラシステム社 交通第二システム部 部長 黒川幸市郎氏に深く感謝申し上げます。鉄道輸送現場の現状やニーズをご教示頂くと共に、スケジューリング方式検討におけるご討論や評価実験へのご協力など、研究推進にあたり多大なるご支援を賜りました、(株)日立製作所 インフラシステム社 交通第二システム部 主任技師 上木暢高氏、技師 永井聡氏に厚く御礼申し上げます。

第5章に関して、車両基地における業務実態の調査の機会を与您と共、スケジューリング方式検討におけるご討論や評価実験へのご協力など、研究推進にあたり多大なるご支援を賜りました、(株)日立製作所 インフラシステム社 交通第二システム部 主任技師 岡田健一郎氏に心より御礼申し上げます。研究成果の実フィールドでの実証と実システムへの展開を強力に推進して頂いた、(株)日立製作所 インフラシステム社 交通第二システム部 主任技師 松尾雅一氏、同 交通第一システム部 技師 岡崎 貴則氏に厚く御礼申し上げます。

上司として本研究と本論文をまとめる動機付けと機会を与您と共、本論文の執筆にあたり格別のご配慮と多大なるご支援を賜りました、(株)日立製作所 横浜研究所 情報サービス研究センター センタ長 赤津雅晴 博士、同 主管研究員 河野克己 博士、同 社会インフラシステム研究部 部長 鮫嶋茂稔 博士に心より感謝申し上げます。

本研究全般を推進するにあたり、鉄道計画システム研究グループの一員として、日々様々なご討論、ご助言、ご支援を頂きました、(株)日立製作所 システム開発研究所 第一部 研究員 崎川修一郎氏(現(株)日立製作所 インフラシステム社 交通システム設計部 技師)、(株)日立製作所 横浜研究所 社会インフラシステム研究部 研究員 富山友恵氏、研究員 加藤祐子氏に心より感謝申し上げます。

最後に、長期にわたり本研究並びに本論文の執筆を応援してくださいました、(株)日立製作所 横浜研究所 社会インフラシステム研究部 SS5 研究ユニットの皆様心から御礼申し上げます。

そして、これまでの道のりを暖かく見守り、時に励まし、時に陰ながら支えてくれた、妻 尚美、長男 優斗、母ならびに亡き父に感謝致します。

参考文献

- [1] 独立行政法人 鉄道建設・運輸施設整備支援機構（鉄道・運輸機構）：「鉄道の特性と役割について」, <http://www.jrft.go.jp/02Business/Construction/const-yakuwari.html>
- [2] the Association of the European Rail Industry (UNIFE): “UNIFE World Rail Market Study -Status quo and outlook 2020-” (2010).
- [3] J. C. Laprie: “Dependable Computing and Fault Tolerance: Concepts and Terminology”, Proc. of Fault-Tolerant Computing Symposium (FTCS-15), Vol.III, pp.2-11 (1985).
- [4] 独立行政法人 科学技術振興機構 研究開発戦略センター：「情報化社会の安全と信頼を担保する情報技術体系の構築 ニュー・ディペンダビリティを求めて」, <http://crds.jst.go.jp/singh/wp-content/uploads/06sp071.pdf> (2007).
- [5] 岩野 和生: 「社会サービスのディペンダビリティ」, ディペンダビリティワークショップ報告書, CRDS-FY2006-WR-07, pp.56-68, 独立行政法人 科学技術振興機構 研究開発戦略センター (CRDS) (2007).
- [6] 増田 昭彦, 渡邊 均: 「サービスのディペンダビリティの標準化」, 日本信頼性学会誌: 信頼性, Vol.32, No.4, pp.220-226 (2010).
- [7] 列車ダイヤ研究会: 「列車ダイヤと運行管理」, 成山堂書店 (2008).
- [8] 電気学会・鉄道における運行計画 運行管理業務高度化に関する調査専門委員会 編: 「鉄道ダイヤ回復の技術」, オーム社 (2010).

- [9] 富井 規雄 : 「デマンド輸送実現のための課題」, 鉄道総研報告, Vol.17, No.12, pp.1-7 (2003).
- [10] 富井規雄 : 「鉄道のスケジューリングアルゴリズム - 現状と今後の課題 - 」, オペレーションズ・リサーチ, Vol.49, No.1 pp.33-39 (2004).
- [11] 富井規雄 : 「輸送計画・運行管理システムの高度化に向けて」, 電気学会誌, Vol.124, No.5 pp.288-291 (2004).
- [12] 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, pp.830-835 (2003).
- [13] 富井 規雄 : 「運転整理システムの高度化にむけての課題」, 鉄道総研報告, Vol.18, No.12, pp.1-5 (2004).
- [14] 富井 規雄 : 「鉄道のスケジューリング問題 : 難しさ面白さ」, オペレーションズ・リサーチ, Vol.53, No.8, pp.427-432 (2008).
- [15] 富井規雄 : 「鉄道の運行計画作成アルゴリズムの高度化」, 平成 23 年電気学会産業応用部門大会, 電気学会, pp.III:75-78 (2011).
- [16] 今泉 淳 : 「鉄道のスケジューリングに対する数理技術の現状 国内外における研究動向 」, 平成 23 年電気学会産業応用部門大会, 電気学会, pp.III:91-96 (2011).
- [17] D. A. Waterman: "A guide to expert systems", Addison-Wesley (1986).
- [18] 鶴田 節夫, 江口 俊宏 : 「時変目的協調推論技術の開発と列車運転整理 AI システムへの適用」, 情報処理学会研究報告, vol.1993, No.103. pp.65-74 (1993).
- [19] 荒木 大, 櫻井 茂明, 小川 真一郎, 篠原 正憲 : 「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 C, Vol.115, No.5, pp.672-680 (1995).
- [20] 平井 力, 富井 規雄, 田代 善昭, 近藤 繁樹, 藤森 淳 : 「運転整理パターン記述言語 R による列車運転整理案作成アルゴリズム」, 情報科学技術レターズ, 4 (2005).

- [21] 平井 力 :「列車ダイヤ乱れ時の再スケジューリングアルゴリズム」, オペレーションズ・リサーチ, Vol.53, No.8, pp.446-452 (2008).
- [22] 中村 達也, 平井 力 :「指令の考え方を再現する運転整理案作成アルゴリズムの開発」, 交通・電気鉄道研究会, 電気学会, Vol.TER-10, No.51, pp.17-22 (2010).
- [23] 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」, 情報処理学会論文誌 : 数理モデル化と応用, Vol.46, No.SIG 2, pp.26-38 (2005).
- [24] 國松 武俊, 平井 力 :「利用者デマンドを考慮した運転整理案作成アルゴリズムの開発」, 鉄道総研報告, Vol.23, No.8, pp.17-22 (2009).
- [25] 清水 宏之, 野末 尚次 :「新しい計画技術と運行管理 制約プログラミングを用いた新幹線運転整理システム」, 第 16 回電気学会産業応用部門大会, 電気学会, pp.831-836 (2002).
- [26] 崎川 修一郎, 佐藤 達広, 森田 豊久, 大田健二 :「運転整理知識を活用した制約プログラミングによる整理案作成方式」, 電気学会論文誌 C, Vol.130 , No.2 , pp.332-342 (2010).
- [27] J. Tořnquist: “Computer-based decision support for railway traffic scheduling and dispatching : A review of models and algorithms”, ATMOS 2005-5th Workshop on Algorithmic Methods and Models for Optimization of Railways (2005).
- [28] S. Wegele, R. Slovak and E. Schnieder: “Real-time decision support for optimal dispatching of train operation”, RailHannover 2007 : 2nd International Seminar on Railway Operations Modeling and Analysis, Hannover, Germany (2007).
- [29] A. D’Ariano and M. Pranzo: “An advanced real-time dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances”, RailHannover

-
- 2007 : 2nd International Seminar on Railway Operations Modeling and Analysis, Hannover, Germany (2007).
- [30] J. Rodrigues: “A constraint programming model for real-time train scheduling at junctions”, *Transportation Research Part B*, 41, pp.231-245 (2007).
- [31] S. Wegele and E. Schnieder: “Automated dispatching of train operations using genetic algorithms”, *Computers in Railways IX*, pp.775-784, WIT Press (2008).
- [32] C. J. Goodman and R. Takagi: “Dynamic re-scheduling of trains after disruption”, *Computers in Railways IX*, pp.765-774, WIT Press (2008).
- [33] 高橋理, 片岡健司, 小島央士, 浅見雅之 : 「ダイヤ乱れ時における列車乗務員運用整理案の自動作成」, *電気学会論文誌 D*, Vol.128, No.11, pp.1291-1297 (2008).
- [34] R. Freling, J. M. P. Paixão, and A. P.M. Wagelmans, “Models and algorithms for vehicle scheduling,” Report 9562/A, Econometric Institute, Erasmus University Rotterdam (1999).
- [35] T. Erlebach, M. Gantenbein, D. Hürlimann, and G. Neyer, “On the Complexity of Train Assignment Problems,” *Proc. 12th International Symposium on Algorithm and Computation (ISAAC 2001)*, *Lecture Notes in Computer Science* 2223, pp. 390-402 (2001).
- [36] V. Cacchiani, A. Caprara, and P. Toth, “Solving a Real-World Train Unit Assignment,” *Proc. 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATOMS 2007)*, pp .79-95 (2007).
- [37] A. Caprara, M. Fischetti, P. L. Guida, P. Toth, and D. Vigo, “Solution of Large-Scale Railway Crew Planning Problems: the Italian Experience,” *Lecture Notes in Economics and Mathematical Systems* 471, pp. 1-18 (1999).

- [38] M. Fischetti, A. Lodi, S. Martello, and P. Toth, “A Polyhedral Approach to Simplified Crew Scheduling and Vehicle Scheduling Problems,” *Management Science*, Vol. 47, Issue 6, pp. 833-850 (2001).
- [39] K. Haase, G. Desaulniers, and J. Desrosiers, “Simultaneous Vehicle and Crew Scheduling in Urban Mass Transit Systems,” *Transportation Science*, Vol. 35, No. 3, pp. 286-303 (2001).
- [40] 片岡 健司, 駒谷 喜代登: 「 Simulated Evolution 法を用いた乗務員運用計画作成手法」, 電気学会論文誌 D, Vol.117, No.7, pp.808-814 (1997).
- [41] 福村 直登, 坂口 隆, 富井 規雄, 中村 達也, 西森 進矢, 登坂 安彦: 「鉄道における車両運用計画作成問題」, 2004 年春季研究発表会, 日本オペレーションズリサーチ学会, pp.134-135 (2004).
- [42] 小川 健一, 今泉 淳, 森戸 晋, 福村 直登: 「鉄道の多拠点乗務員運用問題に対する列生成アプローチ」, 2004 年春季研究発表会, 日本オペレーションズリサーチ学会, pp.136-137 (2004).
- [43] 大野 明良, 西 竜志, 乾口 雅弘, 高橋 理, 上田 健詞: 「定期検査制約を有する車両運用計画問題への列生成法の適用」, 電気学会論文誌 C, Vol.132 No.1, pp.151-159 (2012).
- [44] R. Freling, R. M. Lentink, L. G. Kroon and D. Huisman: “Shunting of Passenger Train Units in a Railway Station”, *Econometric Institute Report 277*, Erasmus University Rotterdam, Econometric Institute (2002).
- [45] S. He, R. Song and S. S. Chaudhry: “Fuzzy Dispatching Model and Genetic Algorithms for Railyards Operations”, *European Journal of Operational Research*, Vol. 124, No. 2, pp.307-331 (2000-7).
- [46] G. D. Stefano and M. L. Koči: “A Graph Theoretical Approach To the Shunting Problem”, *Proc. Workshop on Algorithmic Methods and Models for Optimization of Railways*,

- Electronic Notes in Theoretical Computer Science, Vol. 92, pp.16-33, Budapest, Hungary (2003).
- [47] N. Tomii, L. J. Zhou and N. Fukumura: “An Algorithm for Station Shunting Scheduling Problems Combining Probabilistic Local Search and PERT”, Proc. Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-99), Springer LNCS 1611, pp.788-797, Cairo, Egypt (1999).
- [48] N. Tomii and L. J. Zhou: “Depot Shunting Scheduling using Combined Genetic Algorithm and PERT”, Proc. Int. Conf. on Computers in Railways (COMPRAIL 2000), WIT Press Computers in Railways VII, pp.437-446, Bologna, Italy (2000).
- [49] 富井 規雄, 周 利劍, 福村 直登: 「確率的局所探索と PERT を組み合わせた駅構内入換計画作成アルゴリズム」, 電気学会論文誌 C, Vol.119, No.3, pp.289-295 (1999)
- [50] W. Herroelen and R. Leus: “Robust and reactive project scheduling: a review and classification of procedures”, International Journal of Production Research, Vol.42, No.8, pp.1599-1620 (2004).
- [51] D. Ouelhadj and S. Petrovic: “A survey of dynamic scheduling in manufacturing systems”, Journal of Scheduling, Vol.12, N.4, pp.417-431 (2009).
- [52] 野々部 宏司, 茨木 俊秀: 「資源制約付きスケジューリング問題の定式化と近似解法」, 数理解析研究所講究録, Vol.1068, pp.216-224 (1998).
- [53] H. E. Sakkout and M. Wallace: “Probe backtrack search for minimal perturbation in dynamic scheduling”, Constraints, Vol.5, No.4, pp.359-388 (2000).
- [54] O. Alagož and M. Azizoglu: “Rescheduling of identical parallel machines under machine eligibility constraints”, European Journal of Operational Research, Vol.149, Issue 3, pp.523-632 (2003).

- [55] K. M. Calhoun, R. F. Deckro, J. T. Moore, J. W. Chrissis and J. C. V. Hove: “Planning and re-planning in project and production scheduling”, *Omega*, Vol.30, pp.155-170 (2002).
- [56] S. D. Wu: “One-machine rescheduling heuristics with efficiency and stability as criteria”, *Computers and Operations Research*, Vol.20, No.1, pp.1-14 (1993).
- [57] F. Deblaere: “Reactive scheduling in the multi-mode RCPSPP”, *Computers & Operations Research*, Vol.38, Issue 1, pp.63-74 (2011).
- [58] J. C. Bean, J. Mittenthal and C. E. Noon: “Match up scheduling with multiple resources release dates and disruptions”, *Journal of Operations Research*, Vol.39, No.3, pp.471-483 (1991).
- [59] 谷水 義隆, 小松 祐介, 小澤 知里, 岩村 幸治 : 「共進化遺伝的アルゴリズムを用いた統合型リアクティブスケジューリングに関する研究」生産システム部門講演会講演論文集 2008, pp.29-30 (2008).
- [60] 藤井 信忠, 貝原 俊也, 大塚 太朗: 「分散型遺伝的アルゴリズムによるリアクティブ・スケジューリング手法に関する研究」生産システム部門講演会講演論文集 2009, pp.21-22 (2009).
- [61] L.H. Wu, X.D. Chen, X. Chen, and Q.X. Chen: A genetic algorithm for reactive scheduling based on real-time manufacturing information, 5th International Conference on Responsive Manufacturing - Green Manufacturing (ICRM 2010), pp.375-381 (2010).
- [62] P. I. Cowling, D. Ouelhadj and S. Petrovic: “Dynamic scheduling of steel casting and milling using multi-agents”, *Journal of Production Planning and Control*, Vol.15, pp.1-11 (2004).
- [63] G. P. Henning and J. Cerda: “Knowledge-based predictive and reactive scheduling in industrial environments”, *Computers and Chemical Engineering*, Vol.24, No.9, pp.2315-2338 (2000).

- [64] 長谷川 豊：「鉄道における運行管理システム」, 情報処理, Vol.25, No.3, pp.222-231 (1984).
- [65] 五十嵐 昭夫, 正司 重男, 竹村 喜市, 江田 和孝, 松丸 宏, 解良 和郎：「21世紀を目指した新しい新幹線総合システム (COSMOS) の開発」, 平成8年電気学会産業応用部門全国大会, 電気学会, pp.439-444 (1996).
- [66] 宮島 弘：「運行管理システムの変革 (安全・安定輸送の確保をめざして)」, JR EAST Technical Review, No.5, pp.12-20 (2003).
- [67] 片岡 健司, 明日香 昌, 駒谷 喜代登：「安全・安定に寄与する鉄道情報処理技術 (後編)」, 情報処理, Vol.48, No.9, pp.1008-1017 (2007).
- [68] 井上 健造, 有澤 太一, 相馬 眞, 辺田 文彦：「運転整理システムの開発」, JR EAST Technical Review, No.20, pp.59-61 (2007).
- [69] 安斎 信雄, 香西 恵介：「入換トータルシステム」, 鉄道と電気技術, Vol.6, No.2, pp.19-23 (1995).
- [70] 原 啓太：「運用トータル管理システムの開発」, JR EAST Technical Review, No.5, pp.43-54 (2003).
- [71] 辺田 文彦：「輸送システムの開発概要」, JR EAST Technical Review, No.20, pp.19-23 (2007).
- [72] 小島 央士, 浅見 雅之, 相馬 眞, 辺田 文彦：「乗務員運用整理支援システムの実運用検証試験」, JR EAST Technical Review, No.20, pp.55-61 (2007).
- [73] 辺田 文彦, 相馬 眞, 若井 雅宏, 小島 央士, 片岡 健司, 高橋 理：「列車ダイヤ乱れ時の乗務員運用整理支援」, オペレーションズ・リサーチ, Vol.53, No.8, pp.453-457 (2008).
- [74] 若井 雅宏, 相馬 眞, 辺田 文彦：「乗務員運用整理支援システムの開発」, JR EAST Technical Review, No.28, pp.61-66 (2009).

-
- [75] 安保 徳央 :「ダイヤ乱れ時の乗務員運転整理支援機能の向上に向けた方策」, 第 29 回 情報システム研究会, 電気学会, pp.23-28 (2007).
- [76] 2005 年度版ものづくり白書 (製造基盤白書), 厚生労働省, 経済産業省, 文部科学省 (2005).
- [77] M. L. Fisher, “The Lagrangian Relaxation Method for Solving Integer Programming Problems,” *Management Science*, Vol. 27, No. 1, pp. 1-18 (1981).
- [78] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Inurische Mathematik*, Vol. 1 pp. 269-271 (1959).
- [79] R. Bellman, “On a Routing Problem,” *Quarterly of Applied Mathematics*, Vol. 16, No. 1, pp. 87-90 (1958).
- [80] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Introduction to Algorithms, Second Edition,” MIT Press (2001).
- [81] N. Z. Shor, “Minimization Methods for Non-differentiable Functions,” *Springer Series in Computational Mathematics*, Springer (1985)
- [82] M. E. Lübbecke, J. Desrosiers : “Selected Topics in Column Generation”, *Operation Research*, Vol.53, No.6, pp.1007-1023 (2005).
- [83] 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*, Vol.46, pp.316-329 (1998).
- [84] E. Aarts, and J. K. Lenstra : “Local Search in Combinatorial Optimization”, John Wiley & Sons, Chichester (1997).
- [85] P. Brucker and R. Schlie: “Job-shop Scheduling with Multi-purpose Machines”, *Computing*, Vol. 45, No. 4, pp.369-375 (1990).

-
- [86] K. Jansen, M. Mastrolilli and R. Solis-Oba: “Approximation Algorithms for Flexible Job Shop Problems”, Proc. Latin American Theoretical Informatics (LATIN 2000), Springer LNCS 1776, pp.68-77, Punta del Este, Uruguay (2000).
- [87] N. Zribi, I. Kacem, A.Elkamel and P. Borne: “Optimization by Phases for the Flexible Job-shop Scheduling Problem”, Proc. Asian Control Conference, Vol.3, pp.1889-1895, Melbourne, Australia (2004).
- [88] K. Marriott and P.J. Stuckey: Programming with Constraints: an Introduction, MIT Press (1998).
- [89] I.J. Lustig and J. Puget: “Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming”, Interfaces, Vol.31, No.6, 29-53 (2001).
- [90] R. Dechter: “Constraint Processing”, Morgan Kaufmann (2003).
- [91] R. Bartak: “Constraint propagation and backtracking-based search”, First International Summer School on CP (2005).
- [92] 増井 忠幸, 百合本 茂, 片山 直登: 「ロジスティクスのOR」, pp.186-202, 槇書店 (1998).
- [93] 鍋島一郎: 「スケジューリング理論」, 森北出版 (1974).
- [94] 笠井 琢美, 戸田 誠之助: 「計算の理論」, 情報数学講座, 共立出版 (1993).
- [95] 佐藤 達広: 「計画とスケジューリングシステム」, 人工知能学事典, 15-3, 人工知能学会編, 共立出版 (2005).
- [96] 柳浦 睦憲, 茨木 俊秀: 「組合せ最適化 -メタ戦略を中心として-」, 朝倉書店 (2001).
- [97] M. Dorigo and T. Stuetzle: “Ant Colony Optimization”, Bradford Book (2004).

- [98] A. Frenández, A. P. Cucala and J. I. Sanz: “An integrated information model for traffic planning, operation and management of railway lines”, Computers in Railways IX, pp.743-752, WIT Press (2008).
- [99] 椎名 孝之 : 「確率計画法」, 「応用数理計画ハンドブック」(久保幹雄, 田村明久, 松井知己編), 理論編 第 13 章, pp.710 - 769, 朝倉書店 (2002).
- [100] D.H. Wolpert and W.G. Macready: “No free lunch theorems for optimization”, Evolutionary Computation, IEEE Transactions on , vol.1, no.1, pp.67-82 (1997).
- [101] A. Gorod, B. Sauser and J. Boardman: “System-of-Systems Engineering Management: A Review of Modern History and a Path Forward”, IEEE Systems Journal, Vol.2, No.4, pp.484-499 (2008).
- [102] 飯島 光一郎, 福本 恭, 志村 明俊, 加藤 博光, 平澤 茂樹 : 「社会インフラを支えるシステム技術」, 日立評論, Vol.93, No.12, pp.52-57 (2011).
- [103] 佐藤 達広, 富山 友恵, 松尾 雅一 : 「鉄道車両基地における構内入換計画支援システムの開発」, 平成 23 年電気学会産業応用部門大会, 電気学会, pp.III:83-86 (2011).
- [104] 電気学会・鉄道における運行計画高度化と評価に関する調査専門委員会 編 : 「鉄道における運行計画高度化と評価」, 電気学会技術報告 (2011).
- [105] the United Nations: World Population Prospects (The 2010 Revision), <http://www.un.org/esa/population/unpop.htm> (2010).
- [106] 河野 克己, 平澤 茂樹 : 「社会インフラの高度化に向けたシステム技術開発の動向」, 電気学会論文誌 C, Vol.1332, No.2, pp.183-186 (2012).

研究業績一覧

学術論文誌論文

- [A1] A Lagrangian Relaxation Method for Crew and Vehicle Rescheduling of Railway Passenger Transportation and its Application
Tatsuhiro Sato, Tomoe Tomiyama, Toyohisa Morita and Tomohiro Murata
電気学会論文誌, Vol.132-C, No.2, pp.260-268 (2012).
- [A2] 運転整理知識を活用した制約プログラミングによる整理案作成方式
崎川 修一郎, 佐藤 達広, 森田 豊久, 大田健二
電気学会論文誌, Vol.130-C, No.2, pp.332-342 (2010).
- [A3] ネットワーク表現を用いた作業計画向け割当条件分析手法の提案
富山 友恵, 佐藤 達広, 森田 豊久, 佐々木 敏郎
電気学会論文誌, Vol.129-C, No.11, pp.2083-2090 (2009).
- [A4] Crew and Vehicle Rescheduling Based on a Network Flow Model and Its Application to a Railway Train Operation
Tatsuhiro Sato, Shuichiro Sakikawa, Toyohisa Morita, Nobutaka Ueki and Tomohiro Murata
IAENG International Journal of Applied Mathematics,
Volume 39 Issue 3, pp.142-150 (2009).
- [A5] 条件変化に伴う再計画を考慮した鉄道車両基地構内入換スケジューリング方式

佐藤 達広, 角本 喜紀, 村田 智洋

電気学会論文誌, Vol.127-C , No.2 , pp.274-283 (2007).

[A6] ラグランジュ緩和法を用いた空港スポット割当問題の解法

佐藤 達広, 佐々木 敏郎, 瀬古沢 照治

電気学会論文誌, Vol.123-C , No.7 , pp.1341-1349 (2003).

国際会議・シンポジウム（査読付）

[A7] Rescheduling Algorithm Using Operational Patterns for Rolling Stock Operation at Train Depots

Tomoe Tomiyama, Tatsuhiko Sato, Kenichiro Okada and Takashi Wakamiya

Proc. The 13th International Conference on Design and Operation in Railway Engineering (CompRail2012), pp555-566 (2012).

[A8] An Analysis Method of Service Process Considering Resource Operation Rules

Tomoe Tomiyama, and Tatsuhiko Sato

Proc. The 2nd IEEE International Conference on Social Computing (SocialCom2010), Symposium on Social Computing Applications (SCA-10), pp.595-600 (2010).

[A9] Lagrangian Relaxation Method for Network Flow Modeled Crew and Vehicle Rescheduling

Tatsuhiko Sato, Tomoe Tomiyama, Toyohisa Morita and Tomohiro Murata

Proc. The 2nd IEEE International Conference on Advanced Computer Control (ICACC 2010), Vol.1, pp.403-408 (2010).

[A10] Model and Solution Method for Railroad Crew and Vehicle Rescheduling

Tatsuhiko Sato, Shuichiro Sakikawa, Toyohisa Morita, Nobutaka Ueki and Tomohiro Murata

Proc. International MultiConference of Engineers and Computer Scientists 2009
(IMECS 2009), pp.1112-1117 (2009).

講演

- [B1] 鉄道車両基地における構内入換計画支援システムの開発
佐藤 達広, 富山友恵, 松尾 雅一
平成 23 年電気学会産業応用部門大会, 電気学会, pp.III:83-86 (2011).
- [B2] エキスパートの知識を抽出・活用する運転整理支援システムの開発
加藤 祐子, 崎川 修一郎, 佐藤 達広, 吉田 弘毅
第 43 回情報システム研究会, 電気学会, Vol.IS-10, No.50-72, pp.7-12 (2010).
- [B3] 復旧プロセスを可視化する対話型運転整理シミュレーション方式の開発
崎川 修一郎, 佐藤 達広, 森田 豊久, 吉田 弘毅
第 34 回情報システム研究会, 電気学会, Vol.IS-08, No.13-22, pp.41-45 (2008).
- [B4] グラフ表現を用いた作業計画向け割当て条件分析手法の提案
富山友恵, 佐藤 達広, 森田 豊久, 佐々木 敏郎
平成 20 年電気学会電子・情報・システム部門大会, 電気学会, pp.987-992 (2008).
- [B5] 鉄道車両・乗務員運用整理アルゴリズムの基本検討
佐藤 達広, 崎川 修一郎, 森田 豊久, 上木 暢高
平成 19 年電気学会全国大会, 電気学会, pp.III:109-110 (2007).
- [B6] 鉄道運行管理における目的指向型運転整理シミュレーション方式の開発
崎川 修一郎, 佐藤 達広, 森田 豊久, 大田 健二
第 27 回情報システム研究会, 電気学会, pp.83-88 (2006).
- [B7] 鉄道車両基地における構内入換計画スケジューリング方式の開発

佐藤 達広, 角本 喜紀, 奥田 和章

平成 17 年電気学会産業応用部門大会, 電気学会, pp.III:91-94 (2005).

- [B8] ハミルトン閉路の近傍操作に基づく遺伝的アルゴリズムと鉄道車両運用計画問題への適用

佐藤 達広, 佐々木 敏郎

第 59 回情報処理学会全国大会, 情報処理学会, pp.2:165-166 (1999).

- [B9] GA・ヒューリスティック融合スケジューリング方式— 車両転線計画の作成 —

佐藤 達広, 江口 俊宏, 村田 智洋

人工生命とその応用シンポジウム, 情報処理学会, pp.57-62 (1997).

- [B10] GA・ヒューリスティック融合スケジューリング方式：車両転線計画の作成

佐藤 達広, 江口 俊宏, 村田 智洋

第 55 回情報処理学会全国大会, 情報処理学会, pp.475-476 (1997).

- [B11] GA 応用車両基地トータル運用支援システム：配車計画機能

江口 俊宏, 佐藤 達広, 村田 智洋

第 53 回情報処理学会全国大会, 情報処理学会, pp.167-168 (1996).

- [B12] GA 応用車両基地トータル運用支援システム：車両運用計画機能

佐藤 達広, 江口 俊宏, 村田 智洋

第 53 回情報処理学会全国大会, 情報処理学会, pp.165-166 (1996).

著書

- [C1] (共著)「鉄道における運行計画高度化と評価」, 電気学会技術報告 1240 号,

電気学会・鉄道における運行計画高度化と評価に関する調査専門委員会 / 編,

電気学会 (2011).

- [C2] (共著)「鉄道ダイヤ回復の技術」
電気学会・鉄道における運行計画・運行管理業務高度化
に関する調査専門委員会 / 編, オーム社 (2010).
- [C3] (寄稿)「人工知能学事典」15-3 計画とスケジューリングシステム
人工知能学会編, 共立出版 (2005).

その他

登録特許

- [C4] 運転整理支援システムおよびその方法, 特許第 5044021 号, 日本
- [C5] 運転整理支援システム、及びその方法とプログラム, 特許第 5041952 号, 日本
- [C6] 運行乱れ回復支援装置、方法及びプログラム, 特許第 4786280 号, 日本
- [C7] 図形データ管理装置、特許第 4460427 号, 日本
- [C8] 航空機スポット管理システム, 特許第 4406160 号, 日本
- [C9] 車両基地構内入換計画作成装置、方法及びプログラム, 特許第 4241584 号, 日本
- [C10] 座席指定システム, 特許第 4228422 号, 日本
- [C11] 交通ダイヤ管理装置, 特許第 4090981 号, 日本
- [C12] 運行機材の運用計画作成方法及びシステム, 特許第 3928268 号, 日本