

# タブーサーチを用いた整備作業ダイヤ自動作成手法の開発

小久保 達也\* 加藤 怜\* 中東 太一\*

Development of Automatic Calculation Method for Maintenance Worker Scheduling Using Tabu Search

Tatsuya KOKUBO Satoshi KATO Taichi NAKAHIGASHI

Maintenance worker scheduling is a daily schedule for each worker group to perform inspection and maintenance work such as cleaning of rolling stock during a turnaround operation of a superior trains at a terminal station. Since it varies from day to day according to changes in daily train timetables and rolling stock operations, it is required to be created for each day. We have proposed a method for automatic creation of maintenance worker schedules with tabu search, one of meta-heuristics. In this paper, to confirm the effectiveness of the proposed method, we compared an actual schedule with schedules by proposed method. The results indicated that the proposed method could generate a stable solution within a practical computation time of 3 minutes, regardless of the size of the train schedule.

キーワード：整備作業ダイヤ、車両運用、スケジューリング、メタヒューリスティクス、タブーサーチ、混合整数計画法

## 1. はじめに

ターミナル駅において、特急列車等の優等列車を折返し運用する際には、清掃や点検等の整備作業を実施する必要がある。整備作業は、複数グループ（以下、組）に分かれて実施されるが、各組が1日に担当する列車の整備作業計画は整備作業ダイヤにより示される。整備作業ダイヤは、日々異なる列車ダイヤ・車両運用に対応するため、各日について作成される。現在、整備作業ダイヤは、熟練した計画作成担当者の手作業により作成されているが、整備作業員の労働負荷の観点での制約条件が複数存在するため、1日分の計画作成に数時間かけており、多大な労力を要している。したがって、整備作業ダイヤの自動作成による省力化が求められている。

整備作業ダイヤの自動作成に関する研究は、著者らの知る限り報告されていない。整備作業ダイヤの作成では、割り当てる整備作業の開始時刻や作業時間が整備作業ごとに異なること、組数および各組の始終業時刻、休憩時間や固定業務時間が所与等の特有の条件を考慮する必要がある。整備作業ダイヤ作成問題に類似している問題として、人員割当問題、生産スケジューリング問題、乗務員スケジューリング問題等が挙げられる。人員割当問題は、割り当てる作業の時間が均一であり、各人員の勤務時間も一律に固定で与えられている<sup>1)2)</sup>。生産スケジューリング問題は、各作業が処理される機械の順序が決まっており、各機械における作業の時間が異なるのみであり、目的関数は一般に作業の完了時刻の最小化となる<sup>3)</sup>。乗

務員スケジューリング問題は、作業員（乗務員）の始終業時刻が定められておらず、勤務に自由度がある<sup>4)</sup>。以上より、既存の研究成果を本問題へ適用するのは困難と考えられる。

本問題は、実用性の観点から、計算時間が3分程度以内であること、自動作成のために設定しなければならないパラメータの数が少ないこと、同じ入力であれば同じ整備作業ダイヤを出力すること等の要件がある。また、入力データによっては、すべての制約条件を満たす実行可能解が存在しない可能性が考えられる。さらに、実際に運用されている整備作業ダイヤ（以下、現行運用）を分析した結果、制約条件は、緩和不可能な制約充足の優先度が高い制約条件と緩和可能な制約充足の優先度が低い制約条件とに分かれていることが明らかになった。

そこで、前述した要件に基づき、メタヒューリスティック手法の1つであるタブーサーチ（以下、TS）を適用し、実規模の問題に対して、制約条件の優先度を考慮した整備作業ダイヤを短時間で自動作成可能な手法（以下、提案法<sup>5)6)7)</sup>を開発した。提案法の有効性を確認するため、現行運用および混合整数計画法（以下、MIP）によって作成した整備作業ダイヤとの比較を行った<sup>7)</sup>。

## 2. 整備作業ダイヤの概要

折返し列車のうち、整備作業を実施する必要のある列車はあらかじめ定められており、整備作業の開始時刻は列車の到着時刻により決まる。各整備作業にかかる時間は、整備作業の内容に応じて決められている。整備作業を行う作業員は組ごとに分かれて、それぞれ別の列車の

\* 信号技術研究部 運転システム研究室

整備作業を担当する。1日に整備作業を行う組数は所与であるが、各組は勤務時間帯（AM・PM）ごとに分類されており、この分類を「組群」と呼ぶ。

整備作業ダイヤを作成する際に使用する入力データを図1に示す。図1中の上を示したテーブルは、当該日に整備作業が必要な列車を表している。「2M」等は列車番号と対応した列車の整備作業であり、番線と作業時間帯は所与である。図1中の下を示したテーブルは、各組が作業可能な時間帯を表している。「休」「所」は休憩や会議等の所定外業務（以下、固定作業）であり、その時間帯は所与である。固定作業に必要な時間を「固定作業時間」と呼ぶ。また、「始」は始業後準備時間、「終」は終業前準備時間、「中」は中断準備時間（休憩・所定外業務前に必要な準備時間）、「再」は再開準備時間（休憩・所定外業務後に必要な準備時間）であり、それぞれ必要な時間は所与である。始業後準備時間と終業前準備時間を「始終業準備時間」、中断準備時間と再開準備時間を「固定作業準備時間」と呼ぶ。

図1の入力データをもとに作成された整備作業ダイヤを図2に示す。図1は、組群が2つ、組が4つ、整備作業が2M～28Mの14列車の例であり、図2のように、所与の組に対して、全ての整備作業を割り当てる必要がある。作成担当者は、図1のうち、準備時間と固定作業時間を除いた時間帯で、以下に述べる各種勤務条件を満たしながら、各組のいずれかに整備作業を割り当てる。まず、始業時刻・固定作業の終了時刻が早い組順に最初の整備作業、終業時刻・固定作業の開始時刻が遅い

組順に最後の整備作業を割り当てる。また、ある整備作業から次の整備作業へ移行する際に作業番線が変更となる場合には、「番」で記載した番線移動時間が発生する。さらに、整備作業時間または番線移動時間の終了時刻から、次の整備作業時間の開始時刻までに一定時間を確保できず、連続で整備作業をしたとみなせる場合（以下、連続作業）には、連続作業後から次の整備作業への番線移動時間後に、トイレ休憩等に該当する「加」で記載した加算時間を確保する必要がある。

### 3. 整備作業ダイヤ作成問題

#### 3.1 整備作業ダイヤのグラフ表現

本研究では、整備作業ダイヤをノードとアークを用いたグラフ表現を応用して表現することとした。整備作業ダイヤをグラフ表現した例を図3に示す。ノードは、各整備作業に対応して設定する「整備作業ノード」、各組に対応して設定し、勤務開始を示す「始業ノード」、勤務終了を示す「終業ノード」、各組の各固定作業に対応して設定する「固定作業ノード」の4種類を設ける。また、各ノードには、「開始時刻」および「終了時刻」を設定する。整備作業ノードは対応する整備作業の開始時刻および終了時刻、始業ノードは開始時刻および終了時刻ともに対応する組の始業時刻、終業ノードは開始時刻および終了時刻ともに対応する組の終業時刻、固定作業ノードは対応する固定作業の開始時刻および終了時刻を設定する。アークは任意の2つのノードにおいて、一方

整備作業が必要な列車

番線	07:00	10:00	13:00	16:00	19:00
1		8M			26M
2	4M	6M 12M 16M		22M	
3			14M	20M	24M
4	2M	10M	18M		28M

2M等：整備作業

各組が作業可能な時間帯

組群	組	07:00	10:00	13:00	16:00	19:00
1	1	始		中 休 再	終	
	2	始		中 所 再	終	
2	3		始		中 休 再	終
	4		始		中 所 再	終

↑：出退勤時刻  
 休|所：休憩・所定外業務(固定作業)  
 始|終：始終業前後の準備  
 中|再：固定作業前後の準備  
 □：整備作業が可能な時間帯

図1 整備作業ダイヤの作成に必要な入力データの例

組群	組	07:00	10:00	13:00	16:00	19:00
1	1	始 2M 番	6M 12M 中 休 再 22M 終			
	2	始 4M 番	8M 中 所 再 24M 終			
2	3		始 10M 番 16M 番 加 20M 中 休 再 26M 終			
	4		始 14M 18M 中 所 再 28M 終			

↑：出退勤時刻  
 2M等：整備作業  
 休|所：休憩・所定外業務(固定作業)  
 始|終：始終業前後の準備  
 中|再：固定作業前後の準備  
 番：整備作業間に必要な番線移動  
 加：連続で作業した場合に必要なトイレ休憩等

図2 図1の入力データに基づいて手作業で作成された整備作業ダイヤの例

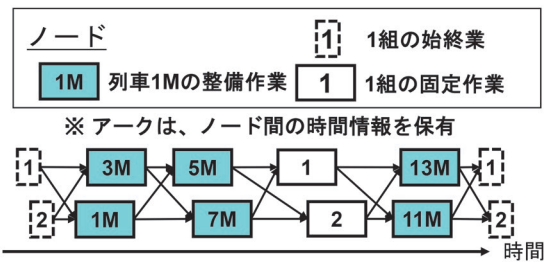


図3 整備作業ダイヤのグラフ表現の例

の開始時刻が、もう一方の終了時刻である場合に、後者のノードから前者のノードに向けて張る。ただし、いずれかのノードから始業ノードに向けたアーク、終業ノードからいずれかのノードへ向けたアーク、始業ノード同士のアーク、終業ノード同士のアーク、異なる組の固定作業ノード同士のアークは張らない。アークは、ある作業の次に実施可能な作業を意味し、実施可能な場合にアークが張られることになる。また、各アークは、「作業間の時間」の情報を保有する。これは、接続した2つのノード間の間合い時間を意味し、「接続先ノードの開始時刻—接続元ノードの終了時刻」で計算する。本研究では、番線移動時間、各準備時間、加算時間の必要時間が不足している場合においても、アークを設定することで、後述する優先度の高い制約条件を充足するため、優先度の低い時間に関する制約条件の違反を許容可能なグラフとした。

### 3.2 制約条件

制約条件は、「割り当てに関する条件」、「作業本数に関する条件」、「時間に関する条件」の大きく3種類に分けられる。以下に、制約条件を示す。

#### (1) 割り当てに関する条件

- ・制約条件①：作業割り当て制約  
対象の全作業にいずれかの組を1組のみ割り当てる。
- ・制約条件②：作業割り当て順序制約  
始業時刻・固定作業終了時刻が早い組の順に、直後の最初の作業を割り当てる。また、終業時刻・固定作業開始時刻が遅い組の順に、直前の最後の作業を割り当てる。なお、この制約が適用される固定作業と、適用されない固定作業がある。

#### (2) 作業本数に関する条件

- ・制約条件③：作業本数上限制約  
各組が1日で作業可能な本数は設定値以内とする。
- ・制約条件④：組群内作業本数差上限制約  
ある組群内における各組の作業本数の差は設定値以内とする。
- ・制約条件⑤：組群間作業本数差上限制約

ある組群間の各組の作業本数の差は設定値以内とする。

#### (3) 時間に関する条件

- ・制約条件⑥：番線移動時間制約  
作業間で番線移動が発生する場合は、移動のために必要な時間を確保する。
- ・制約条件⑦：始業・終業準備時間制約  
各組の始業後および終業前に必要な準備時間を確保する。
- ・制約条件⑧：固定作業準備時間制約  
各組の固定作業時には、前後に中断準備時間および再開準備時間を確保する。
- ・制約条件⑨：連続作業時間制約  
作業間で十分な間合い時間がなく一定回数連続して作業した場合には、番線移動時間に加え加算時間を確保する必要がある。具体的には「番線移動が終了した時刻から次の作業の開始時刻までの間合い時間が〇分以下で、連続して作業を△回行った場合、次の作業後に□分確保する」といった条件の形で定義される。また、条件は複数種類ある場合がある。

これらのうち、緩和不可能な制約充足の優先度が高い制約条件は、割り当てと作業本数に関する各種制約条件①～⑤であり、緩和可能な制約充足の優先度が低い制約条件は、時間に関する制約条件⑥～⑨である。現行運用を分析した結果、優先度が高い制約条件の充足が、実用面で重要であることが明らかとなっている。

### 3.3 目的関数

本問題では、組数、各組の始終業時刻は所与であるため、定められた組数と時間内で、整備作業員の労働負荷の均等や低減を考慮して、作業を割り当てる必要がある。そこで、負荷の均等を目的とした各組の作業本数分散と、負荷の低減を目的とした組全体の番線移動時間、加算時間の重み付き和を目的関数の評価値とし、最小化する。

また、先述のように、入力データによっては、すべての制約条件を満たす実行可能解が存在しない可能性が考えられる。そこで、各制約条件の設定値からの逸脱量をもとにペナルティを設定し、重み係数をかけた値を目的関数値に加算する。提案法における各ペナルティは、制約条件①は未割当整備作業本数、制約条件②は作業順序を違反する組合せ数とする。制約条件③～⑤、⑦～⑨は、各制約条件の設定値から大きく違反することを抑制するため、設定値からの逸脱量の二乗和とする。また、番線移動時間の制約条件⑥は、最低限必要な番線移動時間を確保するため、設定値からの逸脱量の四乗和とする。さらに、制約条件には必ず充足すべき項目と緩和可能な項目があるため、各ペナルティに重み係数をかけ、制約

条件の優先度を考慮可能とする。具体的には、優先度の高い制約条件①～⑤の重みを非常に大きな値となるように設定する。以下に目的関数を示す。

$$\min \left\{ w_1 t_1 + w_2 t_2 + w_3 t_3 + \sum_{n=1}^9 p_n e_n \right\} \quad (1)$$

$t_1$  : 作業本数分散

$t_2$  : 合計番線移動時間

$t_3$  : 合計加算時間

$w_1, w_2, w_3$  : 各項に対する重み係数 ( $w_1 + w_2 + w_3 = 1$ )

$e_n$  : 制約条件  $n$  のペナルティ ( $n = 1, \dots, 9$ )

$p_n$  : 各ペナルティに対する重み係数

## 4. 整備作業ダイヤ自動作成問題へのタブーサーチの適用

### 4.1 タブーサーチの概要

提案法で適用した TS は、Fred Glover が組合せ最適化問題を解くために提案したメタヒューリスティック手法の一種である<sup>8)</sup>。TS は、直近で探索した解を記録するタブーリストを用いて探索を行う。記録する解の個数は、事前に設定したタブーリスト長によって決まり、タブーリストに含まれる解に遷移することはできない。TS は、制約を満たす近傍解の中でタブーリストに含まれない目的関数値が最も良い解に、現在まで探索した中で最も目的関数値が良い解（以下、最良解）より改悪でも移動することで、多峰性のある目的関数でも良質な解を探索することが可能である。また、メタヒューリスティック手法の性質上、必ずしも最適解が得られる保証はないが、一般的に、大規模問題に対して短時間に良質な解を生成可能であることが知られているため、多くの分野で幅広く使われている手法である。

提案法では、一般に効率的な探索が可能と知られているアスピレーション基準を適用する<sup>8) 9)</sup>。アスピレーション基準とは次時点の探索点の候補の中で、より目的関数値が良い点は、タブーであっても採用するというものである。

本研究では、タブーリストに入れる要素は 4.3 節で述べる近傍解生成法によって変更される整備作業のみとする。加えて、最良解より目的関数値が良い近傍解は、変更される整備作業がタブーリストに含まれていても採用する。

提案法における TS の手順を以下に示す。

- Step.1 初期解を生成する (4.2 節参照)。生成した初期解を暫定解とする。探索回数  $iter = 1$  とする。
- Step.2 暫定解から近傍解を生成する (4.3 節参照)。ただし、近傍解生成によって変更される整備作業がタブーリストに含まれる場合、最良解よりも

目的関数値が高い場合は、近傍解として生成しない。

- Step.3 近傍解の中で、目的関数値が最も良い解を選択し、暫定解とする。近傍解生成によって変更された整備作業をタブーリストに入れる。
- Step.4  $iter$  が事前に設定した  $iter_{max}$  に達したならば最良解を出力し終了する。そうでなければ、 $iter = iter + 1$  とし、Step.2 へ戻る。

### 4.2 初期解生成法

以下に、初期解生成法の手順を示す。なお、事前に、各組に「現在ノード」を設定する。現在ノードとは、各組に割り当てられたノードの中で最も終了時刻が遅いノードである。

- Step.1.1 全ての組に対し対応する始業ノードを割り当て、全ての組の現在ノードを初期設定する。
- Step.1.2 全ての組の中から、現在ノードの終了時刻が最も早い組を選択し（ただし、現在ノードが終業ノードの組は選択しない）、現在ノードからのアークが接続している整備作業ノードの中で、いずれの組にも割り当てられていない、かつ最も作業間の時間が短い整備作業ノードを抽出する。抽出できない場合は、Step.1.5 へ。
- Step.1.3 抽出した整備作業ノードの終了時刻が、Step.1.2 で選択した組に対応するいずれか、かつ未割当の固定作業ノードの開始時刻を過ぎる場合は、Step.1.4 へ。Step.1.2 で選択した組に対応する終業ノードの開始時刻を超過する場合は、Step.1.5 へ。それ以外の場合、抽出した整備作業ノードを割り当て、現在ノードを更新し、Step.1.7 へ。
- Step.1.4 固定作業ノードを割り当て、現在ノードを更新し、Step.1.6 へ。
- Step.1.5 終業ノードを割り当て、現在ノードを更新する。
- Step.1.6 全ての組の現在ノードが終業ノードとなった場合、現在の割り当てを初期解とし、式 (1) に基づく目的関数値を算出し、処理を終了する。さもなければ、Step.1.2 へ戻る。

### 4.3 近傍解生成法

提案法では、全ての評価値の改善と割り当て制約を除く全ての制約条件の解消を目的として、3つの近傍解生成法を構築した。近傍解生成法①は、時間に関する評価値と時間に関する制約条件の違反の解消に寄与する。一方で、改善作業本数は変更しないため作業本数分散の評価値の改善と作業本数に関する制約条件の違反の解消には寄与しない。また、近傍解生成法②と③は全ての評価値の改善と割り当て制約を除く全ての制約条件の解消に

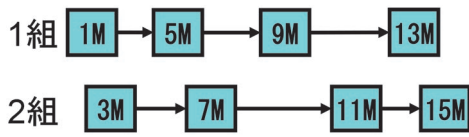


図4 暫定解の例

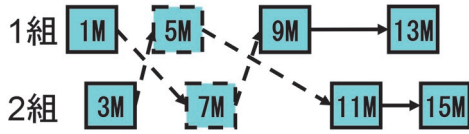


図5 1-1 交換の例

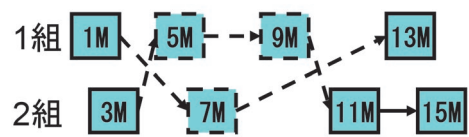


図6 2-1 交換の例

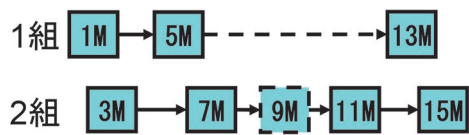


図7 1-0 挿入の例

寄与する。複数の近傍解生成法を用いて、広範囲に探索することで、最良解を得る可能性を向上させる。近傍解生成のもととなる暫定解の例を図4に示す。

近傍解生成法①：1-1 交換

ある組の1つの整備作業ノードと、他の組の同時時間帯の交換可能な1つの整備作業ノードを入れ替える。図5にこの方法で作成した近傍解の例を示す。

近傍解生成法②：2-1 交換

ある組の連続した2つの整備作業ノードと、他の組の同時時間帯の1つの整備作業ノードと入れ替える。図6にこの方法で作成した近傍解の例を示す。

近傍解生成法③：1-0 挿入

ある組の1つの整備作業ノードを、他の組の同時時間帯の作業間に挿入する。図7にこの方法で作成した近傍解の例を示す。

## 5. ケーススタディ

### 5.1 試算条件

提案法の有効性を確認するため、3日分の実際の列車ダイヤと車両運用を用いて検証を行った。具体的には、同日の現行運用とMIPによって作成した整備作業ダイヤとの比較を行った。なお、MIPの定式化の詳細については、文献7を参照されたい。

今回試算する事例における整備作業が必要な列車本数と組数は、Case.1が99本で7組（小規模）、Case.2が120本で8組（中規模）、Case.3が147本で9組（大規模）である。また、各日共通で以下のような条件を設定した。

- 作業本数上限：18本
- 組群内作業本数差上限：2本
- 組群間作業本数差上限：2本
- 各準備時間：10分～20分
- 番線移動時間：0分～3分
- 連続作業制約1：5分空かずに2連続で整備作業した場合、5分加算
- 連続作業制約2：8分空かずに5連続で整備作業した場合、10分加算

ケーススタディにおける目的関数の各項の重み係数を表1に示す。目的関数における各ペナルティに対する重み係数は、制約充足の優先度が低い制約条件⑥～⑨よりも、制約充足の優先度が高い制約条件①～⑤を優先的に満たすような整備作業ダイヤを作成することを指向して設定した。また、TSのパラメータは、タブーリスト長50、探索回数5,000回で計算終了とした。また、実用上の要件として、180秒以内での自動作成が求められているため、全ての試算において、計算時間180秒で打ち切りとした。計算には、PC（CPUがIntel Core i7-8700（3.20GHz）、メモリが64GB）を用いて、TSを用いた提案法の開発にはC言語（Microsoft Visual Studio 2019）を使用し、MIPの求解には、数理最適化ソルバーGurobi Optimizer 9.5.1<sup>10)</sup>を使用した。

### 5.2 試算結果

各Caseにおける現行運用（Ⅰ）と、提案法（Ⅱ）、MIP（Ⅲ）による試算結果の目的関数値、各評価値、各ペナルティを表2に示す。表2において、提案法の目的関数の各評価値と制約充足の優先度が高い制約条件の

表1 ケーススタディにおける目的関数の各項の重み係数<sup>7)</sup>

$w_1$	$w_2$ [min.]	$w_3$ [min.]	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$p_7$	$p_8$	$p_9$
0.9	0.05	0.05	100,000	100,000	100,000	100,000	100,000	100	100	100	100

表2 各 Case の現行運用 (I) と、提案法 (II), MIP(III) による試算結果の目的関数値, 各評価値, 各ペナルティ<sup>7)</sup>

Case.	目的関数値	$t_1$	$t_2$ [min.]	$t_3$ [min.]	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$	$e_9$
1-I	708.67	0.69	131	30	0	0	0	0	0	0	1	0	6
1-II	506.86	<b>0.12</b>	<b>100</b>	<b>35</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	0	1	0	4
1-III	506.66	0.12	96	35	0	0	0	0	0	0	1	0	4
2-I	5613.65	1.00	165	90	0	0	0	0	0	2	1	19	34
2-II	17313.05	<b>1.00</b>	<b>168</b>	<b>75</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	5	104	0	64
2-III	3711.43	0.75	135	80	0	0	0	0	0	3	5	0	29
3-I	14917.35	0.67	200	135	0	0	0	0	0	0	1	92	56
3-II	13912.05	<b>0.67</b>	<b>164</b>	<b>65</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	5	20	29	85
3-III	916517.35	0.67	200	135	1	2	1	2	3	7	1	76	81

各ペナルティを赤字で記載する。

現行運用と提案法により生成した整備作業ダイヤを比較すると、Case.1 では、提案法により合計番線移動時間は31分短縮し、合計加算時間は5分増加した。Case.2 では、提案法により合計番線移動時間は3分増加したが、合計加算時間は15分短縮した。Case.3 では、提案法により合計番線移動時間は36分、合計加算時間は70分短縮した。以上より、提案法は現行運用と比べて、番線移動時間、加算時間のいずれもほぼ同等か低減できていることから、提案法により整備作業員の負荷を低減した整備作業ダイヤが作成できることを確認した。

提案法は、全ての Case において、実用的な計算時間約 60 秒で最良解を得た。また、提案法は、全ての Case において、優先度の高い制約条件①～⑤に対応するペナルティ項が全て 0 となり、実用的な解が得られた。本問題は、計算時間が3分程度であること、優先度の高い制約条件①～⑤を充足することが、実用面で最重要であるため、提案法は、短時間で実用的な整備作業ダイヤを作成可能であるといえる。

MIP は Case.1, 2 では現行運用、提案法と比べ目的関数値が下回る良質な解を得ているものの、すべての Case において、180 秒の計算時間では最適解を得ることはできなかった。特に、Case.3 では目的関数値が明らかに劣っているが、180 秒の段階では暫定解の精度を表す双対ギャップが極めて大きいことから、明らかに計算時間が不足していることを確認した。また、優先度の高い制約条件①～⑤を違反しており、実用的な解が得られていない。すなわち、MIP は最適解を得られる保証があるものの、特に整備作業数が多い日に対しては、実用上求められる数分程度の計算時間では、充足すべき制約条件を必ずしも全て満たすことができず、実用的な解を算出することができない可能性がある。

以上より、提案法は、整備作業数が多い日でも安定して短時間で、優先度の高い制約条件①～⑤を充足し、か

つ、合計番線移動時間と合計加算時間を短縮するような整備作業ダイヤを求めることができおり、本研究が提案する TS を用いた手法は実用面で有効な手法であるといえる。

## 6. まとめ

本研究では、メタヒューリスティック手法の1つである TS を適用し、実規模の問題に対して数分程度で整備作業員の負荷を低減した整備作業ダイヤを自動作成可能な手法を開発した。提案法の有効性を確認するため、現行運用の整備作業ダイヤとの目的関数値と各評価値を比較した。また、提案法はメタヒューリスティック手法の性質上、必ずしも最適解が得られる保証が無いことから、理論的に最適解を保証する MIP によって作成された整備作業ダイヤとの比較を行った。

ケーススタディにより、提案法は、作業が必要な列車本数に関わらず、安定して、3分程度以内の実用的な計算時間内で、優先度の高い制約条件を満たす解を出力し、かつ、合計番線移動時間と合計加算時間を短縮するような整備作業ダイヤを作成することが可能であることが明らかとなった。以上より、実用面で有効な手法であることを確認した。今後、パラメータや目的関数各項の重みの感度分析を行い、より目的関数値の良い解を出力可能なパラメータと目的関数各項の重みを決定し、実務への導入に向けて取り組んでいく。

## 文献

- 1) B. Korte and J. Vygen, "Combinatorial Optimization," Springer-Verlag Berlin Heidelberg, 2006.
- 2) 柳浦睦憲, 茨木俊秀: 組み合わせ最適化—メタ戦略を中心として—, 朝倉書店, 2001
- 3) 黒田充, 村松健児: 生産スケジューリング, 朝倉書店,

2002

- 4) (財) 鉄道総合技術研究所運転システム研究室：鉄道のスケジューリングアルゴリズム, NTS, 2005
- 5) 小久保達也, 加藤怜, 中東太一, 武内陽子, 田中峻一：整備作業ダイヤ自動作成へのタブーサーチの適用, 電気学会交通・電気鉄道研究会, TER-22-063, 2022
- 6) 小久保達也, 加藤怜, 中東太一：制約違反を許容したタブーサーチによる整備作業ダイヤ自動作成手法, 電気学会産業応用部門大会, R5-12, 2022
- 7) 小久保達也, 加藤怜, 中東太一：整備作業ダイヤ自動作成手法の開発におけるタブーサーチ手法と混合整数計画法の比較, 第29回鉄道技術連合シンポジウム (J-RAIL2022), S5-1-2, 2022
- 8) F. Glover and M. Laguna, "Tabu Search, Kluwer Academic Publishers," 1997.
- 9) 橋本英樹, 野々部宏司：入門タブー探索法, オペレーションズ・リサーチ, Vol.58, No.12, pp.703-707, 2013
- 10) Gurobi Optimizer, <https://www.octobersky.jp/products/gurobi> (参照日：2023年9月29日)