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

小久保　達也*　　加藤　怜　　中東　太一（鉄道総合技術研究所）

Comparison with Tabu Search and Mixed Integer Programming in the Development of Automatic Calculation Method for Maintenance Worker Schedules

Tatsuya Kokubo*, Satoshi Kato, Taichi Nakahigashi(Railway Technical Research Institute)

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. In this paper, to confirm the effectiveness of the proposed method, we compared an actual schedule with schedules by proposed method, a mathematical programming method that guarantees theoretically optimal solution. The results indicated that the proposed method can 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，折返し運用，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
(Maintenance Worker Diagram, turn-back operation, scheduling, metaheuristic, tabu search, mixed integer programming)

1. はじめに

終着駅などのターミナル駅における優等列車の折返し運用時には，車両の清掃等の点検整備作業（以下，整備作業）が必要である。整備作業は複数のグループに分かれて実施しており，各グループが 1 日に担当する列車の整備作業計画を「整備作業ダイヤ」と呼ぶ。整備作業ダイヤは，日々の列車ダイヤ・車両運用の変化に応じて，各日について作成する必要がある。現在，整備作業ダイヤは，熟練した計画担当者の手作業により作成されているが，1 日分の計画作成に数時間程度の多大な労力を費やしている。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。問題構造として本問題に類似している問題として，人員割当問題[1][2]，生産スケジューリング問題[3]，乗務員スケジューリング問題[4]などが挙げられる。しかし，組数や休憩時間が所与である点，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点など，整備作業ダイヤ作成特有の条件を考慮すると，既存の手法を本問題に適用することは困難であると考えられる。

さらに本問題は，実用の観点を踏まえると，3 分程度の計算時間での求解が求められる。また，本問題は入力データによっては全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とともに，短時間で実行可能解の有無を判定することは容易ではない。そこで著者らは，必ず実行可能な最適解が得られる保証はないものの，短時間での求解が期待できる，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を採用し，実際の列車ダイヤに対して約 60 秒程度で実用的な整備作業ダイヤを自動的に出力可能な手法を開発した[5][6]（以下，提案法）。しかしながら，上述の通り，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はない。

そこで本論文では，過去の列車ダイヤを用いて，理論的に最適解を保証する混合整数計画法によって作成された整備作業ダイヤと，実際に運用されている整備作業ダイヤ（以下，現行運用）との目的関数値と各評価値の比較を示すことで，提案法の有効性を確認する。

2. 整備作業ダイヤの概要

整備作業を行う担当は複数人のグループ（以下，組）に分かれて，それぞれ別の列車の整備作業を担当する。1 日に整備作業を行う組数はあらかじめ定められている。また，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。整備作業ダイヤの例を図 1 に示す。図 1 の「休」「所」は休憩や会議等の所定外業務（以下，固定作業）であり，時間帯は所与である。なお，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，「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），「終」は終業前準備時間であり，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」とし，中断準備時間と再開準備時間を「固定作業準備時間」とする。これらの準備時間と固定作業時間を除いた時間帯で，「2M」等の列車番号と対応した列車の整備作業を行う。加えて，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。

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。また，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，「加」で記載した加算時間を確保しなければならない。

3. タブーサーチを用いた整備作業ダイヤ自動作成手法

本論文で提案するタブーサーチによる自動作成手法の概要を述べる。基本的な考え方は文献(6)と同様であるが，より実用的な整備作業ダイヤを作成するため，制約条件を追加した。

〈3･1〉整備作業ダイヤのグラフ表現　整備作業ダイヤをノードとアークのグラフを用いて表現する。ノードは整備作業，始終業または固定作業を表し，アークは整備作業と固定作業の順序を表し，作業間の時間情報を持つ。

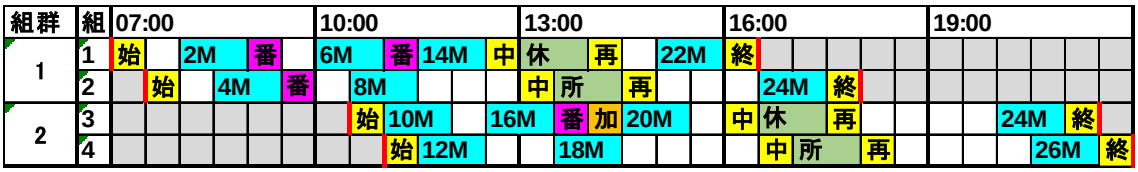
〈3･2〉制約条件　以下に制約条件を示す。連続作業制約は複数の条件が設定されることもある。また，本論文では，現行運用において考慮されている「作業割り当て順序制約」を制約条件(B)として追加した。

- 割り当てに関する制約条件
  - (A) 割り当て制約
    全ての整備作業にいずれかの組を 1 組のみ割り当て
  - (B) 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順に始めの整備作業を割り当て，終業時刻・固定作業開始時刻が遅い組順に最後の整備作業を割り当て
- 整備作業本数に関する制約条件
  - (C) 作業本数上限制約
    各組が 1 日で整備作業可能な列車本数は設定値以内
  - (D) 組群内作業本数差制約
    組群内の各組の整備作業可能な列車本数の差は設定値以内
  - (E) 組群間作業本数差制約
    組群間の整備作業可能な列車本数の差は設定値以内
- 時間に関する制約条件
  - (F) 番線移動時間制約
    番線移動の際には一定の時間を確保
  - (G) 始終業準備時間制約
    始業後，終業前に一定の準備時間を確保
  - (H) 固定作業準備時間制約
    固定作業前後に一定の準備時間を確保
  - (I) 連続作業時間制約
    一定時間空かずに連続で整備作業をした場合，追加時間を加算

〈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，整備作業員の負荷均等・低減を考慮し，効率的に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作業本数分散，負荷の低減と効率的な割り当てを目的とした番線移動時間・加算時間の重み付きの最小化を目的関数とする。また，制約条件(A)は未割当整備作業本数，制約条件(B)は作業順序を違反する組合せ数，制約条件(C)～(H)は設定値からの逸脱量の二乗和を重み付きペナルティとして加算する。これにより，ペナルティの重みによって，制約条件(A)を充足する実用的な解を算出するなど，優先して充足する必要がある制約条件を選択可能となる。以下に目的関数を示す。

$$min\left\{w_1t_1+w_2t_2+w_3t_3+\sum_{n=1}^{N}p_ne_n\right\}\cdots\cdots(1)$$

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 $(w_1+w_2+w_3=1)$



N：ペナルティとして加算する制約条件数

e_n：条件nの違反量$(n = 1, \dots N)$

p_n：各ペナルティに対する重み係数

〈3･4〉整備作業ダイヤ自動作成へのタブーサーチの適用

（１）初期解生成法　以下に，初期解生成法のフローを示す。

Step.1　各組に「作業開始可能時刻」を設定し，始業後の準備時間の固定作業ノードの終了時刻を元に，作業開始可能時刻を更新する。

Step.2　作業開始可能時刻が早い組順に始めの整備作業ノードを割り当て，作業開始可能時刻を更新する。

Step.3　終業前の準備時間の固定作業ノードの開始時刻が遅い順に最後の整備作業ノードを割り当てる。

Step.4　作業開始可能時刻が早い組順に整備作業を割り当て，連続作業が発生した場合には加算時間を加算して作業開始可能時刻を更新する。

Step.5　作業開始可能時刻が固定作業ノードの開始時刻を過ぎた場合は，バックトラックし前の整備作業ノードから固定作業へ接続し，作業開始可能時刻を更新する。

Step.6　作業開始可能時刻が Step.3 で割り当てた最後の整備作業ノードの開始時刻を過ぎた場合には，バックトラックし前の整備作業ノードから最後の整備作業ノードへ接続する。

Step.7　全ての組が Step.6 で最後の整備作業ノードへ接続後，初期解を出力する。

（２）近傍解生成法　以下に，3 つの近傍解生成法を示す。

①1-1 交換　ある組の 1 つの整備作業ノードと，他の組の同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。

②2-1 交換　ある組の連続した 2 つの整備作業ノードと，他の組の同時間帯の 1 つの整備作業ノードと入れ替える。

③1-0 挿入　ある組の 1 つの整備作業ノードを，他の組の同時間帯の作業間に挿入する。

4. 混合整数計画法による整備作業ダイヤ自動作成手法

〈4･1〉混合整数計画法によるモデル化　混合整数計画法（Mixed-Integer Programming，以下「MIP」）とは，対象とする問題を，数式を用いて組合せ最適化問題としてモデル化（定式化）する手法である。定式化ができれば，汎用の数理最適化ソルバーに入力することで，理論的な最適解を得ることができる。

以下では，本問題の定式化の概略を述べる。基本的な考え方として，本問題を割当問題としてモデル化する。そこで，組iを作業jに割り当てる場合に 1，さもなければ 0 とする 0-1 変数x_{ij}を導入する。この変数を用いて，3.2 節で示した各制約条件および(１)式の目的関数を表現する。

〈4･2〉制約条件　各制約条件の定式化上での表現方法について，以下に概略を示す。

- 制約条件(A)は，割当問題と同様に，各作業に対し，1 つの組のみx_{ij}が 1 となる条件を設けて対応する。
- 制約条件(B)は，変数x_{ij}のみで条件を記述するのが困難であるため，論理条件で記述し，追加の 0-1 変数を導入することで対応する。
- 制約条件(C)は，各組のx_{ij}の合計に対し上限を設けることで対応する。制約条件(D)および(E)も各組のx_{ij}の合計を算出し，各組ごとに比較することで対応する。
- 制約条件(F)は，事前に番線移動時間を考慮し，同一組が割当不可能な 2 つの作業の組合せを列挙し，その組合せでは同一の組のx_{ij}がいずれも 1 にならない条件を設けることで対応する。
- 制約条件(G)および(H)は，事前に各組の始終業準備時間，固定作業準備時間を考慮し，割当不可能な作業を列挙し，そのx_{ij}は 0 に固定することで対応する。
- 制約条件(I)は，事前に連続作業とみなせる作業集合を列挙し，各集合に含まれるすべての作業について同一組のx_{ij}が 1 にならないようにすることで対応する。

〈4･3〉目的関数　基本的には，(１)式をそのまま目的関数として設定する。ただし，MIP によるモデル化の制約上，以下の点が異なる。

- 作業本数分散t_1は扱うことが困難であるため，「作業本数の最大値」－「作業本数の最小値」を評価することで代替する。
- 本数に関する制約（制約(C)～(E)）に対しては，MIP では本数に関して 2 乗のペナルティを付与することが困難であるため，違反本数に対し比例のペナルティを付与することで代替する。

5. ケーススタディ

〈5･1〉試算条件　本論文では，3 日分の実際の列車ダイヤを用いて検証を行った。今回試算する列車ダイヤにおいて整備作業が必要な列車本数と組数は小規模の Case.1 が 99 本で 7 組，中規模の Case.2 が 120 本で 8 組，大規模の Case.3 が 147 本で 9 組である。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：0~3 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

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
Table.1 Weighting coefficients for the objective function for case study.

w_1	w_2 [min.]	w_3 [min.]	p_1 ※(A)	p_2 (B)	p_3 (C)	p_4 (D)	p_5 (E)	p_6 (F)	p_7 (G)	p_8 (H)	p_9 (I)
0.9	0.05	0.05	100,000	100,000	100,000	100,000	100,000	100	100	100	100

※ ペナルティは対応する制約条件を示す

表 2　現行運用，タブーサーチ，MIP による試算結果の目的関数値と各評価値
Table.2 Total objective function values and values of every term in the objective function of the actual maintenance worker diagram, tabu search method, and mixed integer programming method.

Case. ※	目的関数値	t_1	t_2 [min.]	t_3 [min.]	p_1 (A)	p_2 (B)	p_3 (C)	p_4 (D)	p_5 (E)	p_6 (F)	p_7 (G)	p_8 (H)	p_9 (I)
1-I	708.67	0.69	131	30	0	0	0	0	0	0	1	0	6
1-II	506.86	0.12	100	35	0	0	0	0	0	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	1.00	168	75	0	0	0	0	0	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**	0.67	164	65	0	0	0	0	0	5	20	29	85
3-III	916517.35	0.67	200	135	1	2	1	2	3	7	1	76	81

※先頭の数字は Case 番号，末尾 I：現行運用，II：タブーサーチ，III：MIP を示す

1 に示す。現行運用の目的関数値と各評価値を複数分析した結果を踏まえ，優先度を考慮しながら目的関数の各項の重み係数を設定した。また，タブーサーチのパラメータは，タブーリスト長 50，探索回数 5,000 回（計算時間 180 秒で打切り）とし，プログラムは，PC（Intel Core i7-8700 (3.20GHz)）上の OS（Windows 10），C 言語（Visual studio 2019）を利用して構築した。MIP は，Intel Core i7-8700K(3.70GHz)，64GB RAM の PC 上で，Gurobi Optimizer 9.5.1 を用いて計算し，180 秒で打切りとした。

〈5･2〉試算結果　各 Case における現行運用(I)，提案法であるタブーサーチ手法(II)，MIP(III)による試算結果として，各目的関数値を表 2 に示す。

小規模の Case.1 では，MIP を用いた自動作成結果が最も良い目的関数値であった。また，タブーサーチを用いた自動作成手法は，現行運用よりも良い目的関数値となった。中規模の Case.2 では，MIP を用いた自動作成結果が最も良い目的関数値であった。また，タブーサーチを用いた自動作成手法は，現行運用よりも劣った目的関数値となった。大規模の Case.3 では，タブーサーチを用いた自動作成手法による結果が最も良い目的関数値であった。また，MIP を用いた自動作成手法は，現行運用よりも劣った目的関数値となった。

MIP は Case.1，2 では現行運用，タブーサーチと比べ目的関数値が下回る良質な解を得ているものの，Case.3 を含め 180 秒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示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がわかった。特に，優先度の高い制約条件(A)～(E)を違反しており，実用的ではない。

すなわち，MIP は最適解を得られる保証があるものの，整備作業数が多い日に対しては，実用上求められる計算時間では実用的な解を算出ことができない可能性がある。一方で，タブーサーチは MIP と比べ目的関数値が劣るケースもあるものの，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(A)～(E)を充足した求解ができている。以上より，本研究が提案するタブーサーチ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

本論文では，整備作業ダイヤ自動作成のために開発したタブーサーチ手法を用いた提案法の有効性を確認するため，理論的に最適解を保証する数理計画法および現行運用との比較を行った。ケーススタディにより，提案法は，列車ダイヤの規模に関わらず，3 分の実用的な計算時間内で安定的な解を出力することが可能であることを確認した。

今後は，タブーサーチ手法における適切なパラメータ設定の構築や，より目的関数値が改善される改良手法の検討を行い，実用化を目指す。

文　献

（1） B.Korte, 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)