

タブーサーチを用いた灯油配送計画の最適化

大江 弘峻^{1,a)} 横山 想一郎² 山下 倫央² 川村 秀憲² 多田 満朗³

概要：灯油配送とは暖房用の灯油を顧客に定期的に配送するサービスである。顧客への給油順序を決定する灯油配送計画の最適化は、灯油タンクが空になる直前で給油しながら、配送経路をできるだけ短くしなくてはならない点が難しく、現実的な時間で求解可能な手法が求められている。本研究では灯油配送計画を在庫配送計画問題として定式化し、解法としてタブーサーチ、リペアリングを使用するヒューリスティクスを提案した。実験より提案手法が最適解の評価値の上界に匹敵する解を求められること、実問題に適用可能な解が求められる可能性を確認した。

1. はじめに

灯油配送は主に寒冷地で行われている業務である。寒冷地の住宅には冬場の暖房やロードヒーティングに使用する灯油を貯蔵するタンクが設置されており、定期的に配送を受けることで燃料を切らさずに暖房などを使用し続けることができる。通常、灯油配送業者は顧客と定期配送契約を結び、灯油配送のタイミングや配送量は配送業者側が灯油タンクを空にしないよう決めていく。こうした、どのタイミングでどの顧客に配送するかをまとめたスケジュールを配送計画と呼ぶ。灯油配送業務については、配送担当者、あるいは配送に使用するタンクローリーごとに配送地区が決められているケースが存在し、その配送地区についての配送計画を各担当者が作成していく。

灯油配送業務における問題を解決するための研究として、1つは灯油タンクの残量推定および灯油消費量予測 [1] があり、もう1つに灯油配送計画の最適化が挙げられる。本論文では、後者に関する研究を扱い、灯油配送計画問題を混合整数計画問題 (MIP) として定式化し、タンクローリーを1台に制限した設定での灯油配送計画問題について解を求める。最終的には、灯油配送計画の自動化が可能なシステムへ手法を応用することを考えて、12時間以内での配送計画の最適化を考える。具体的には、システムは1週間分の灯油配送計画最適化を12時間以内に実行し、翌日

分の配送計画を導出し、ドライバーへ提供することを想定する。この問題に対して、本論文では規模感として札幌市西区程度、タンクローリー1台を想定し、タブーサーチ、リペアリングによる近似解法を提案し、実データを用いて有効性を検証する。まずは、手法と最適解を比較するために、灯油配送計画問題の基礎インスタンスを作成し、MIPソルバを使用し厳密解、提案手法によって近似解を求め、実行時間及び解の評価値、実行可能解が得られたかどうかを比較する。続いて、より配送件数、計画期間が大きい実規模配送件数インスタンスを作成し、提案手法によって近似解を求める。そして、実行時間、実行可能解が得られたかどうかを評価し、最適解の評価値の上界と得られた評価値を比較する。最後に、札幌市西区に対して、1タンクローリーで行われている実際の配送履歴データから作成した、実データインスタンスを使用し、提案手法によって近似解を求める。そして、実行時間、得られた評価値について評価する。

以下に本論文の構成を示す。第2章では灯油配送最適化問題に関連した研究を解説する。第3章では灯油配送を最適化問題として考えた際の定式化について、第4章では提案手法であるタブーサーチ、リペアリングを解説する。第5章では手法を検証するための実験について解説し、第6章にて結果及び考察する。最後に、第7章にて結論を述べる。

2. 関連研究

灯油配送計画問題は、在庫配送問題 (IRP: Inventory Routing Problem)[2], [3] の一種として定式化ができる。在庫配送問題は、1期間内の各ノードに対する配送計画を考える車両配送計画問題 (VRP: Vehicle Routing Problem)

¹ 北海道大学 大学院情報科学院
Graduate School of Information Science and Technology,
Hokkaido University, Sapporo, Hokkaido, 060-0814, Japan

² 北海道大学 大学院情報科学研究院
Faculty of Information Science and Technology, Hokkaido
University, Sapporo, Hokkaido, 060-0814, Japan

³ ゼロスベック株式会社
ZeroSpec Inc., 060-0004, Japan

a) h.ooe@ist.hokudai.ac.jp

に対して期間と在庫管理の概念を追加し、複数期間を対象として配送コストと在庫コストの両面を最適化する配送計画を求める問題である。灯油配送計画問題と類似している問題は、Markov らが行うゴミ集に関する配送計画問題 [4], [5] である。ゴミ収集計画問題はゴミの収集によって収集車の積載量が増加するが、灯油配送計画問題では灯油の給油によってタンクローリーの積載量が減少するという点での違いが存在する。

灯油配送計画問題に最も類似する研究としては、Prescott らが行う暖房用ディーゼルオイルの配送計画問題 [6], [7] が存在する。Prescott らが行う研究 [6] では、非定期で発生するスポットと呼ばれる注文が実際の業務では頻発するために、短期間の最適化が好ましいと考え 1 日と 7 日の比較的短期間での最適化を行った。近似最適解を求める手法として、タブーサーチ [8] と巨大近傍探索 (LNS)[9] を使用したヒューリスティックが提案されており、それぞれの手法がディーゼルオイル配送計画問題において有効であることを示されている。同じく、暖房用ディーゼルオイルの配送計画問題に取り組んだ Bertsimas らの研究 [7] では、現実的な場面を想定した大規模な配送業務に関する効率的な配送計画の導出に関する研究を行った。巡回セールスマン問題 (TSP) のベンチマークとして知られている TSPLIB をベースに作成した大規模な疑似インスタンスに対して、クラスタリングと事前ルート生成を行うことで解空間の圧縮を行う手法を提案し、商用ソルバの一つである Gurobi を使用し現実的な時間での解の導出に成功した。本論文で取り扱う灯油と、上記の研究で扱った暖房用ディーゼルオイルについての違いとしては、単に配送する暖房用オイルの種類のみであり、配送計画問題の枠組みの中では同じ問題と扱ってよい。

配送計画問題において、車両が保有する在庫を補給地点で一度補給するという条件を追加した問題が存在する。Schneider らの研究 [4], [10] では、車両配送計画問題における部分巡回路除去制約として用いられる MTZ 制約を応用し、顧客 i に訪問した時点での合計配送量を定義することで、補給地点に関する制約を表現している。

3. 定式化

本章では、灯油配送計画問題の定式化を述べる。まずは、定式化に必要な変数や制約式の係数の列挙及び説明を行い、目的関数、制約式の順番で定式化する。

灯油配送計画問題にはタンクローリーの出発地点と帰還地点を兼ねている拠点が 1 つ存在する。そこから 1 日最大 1 台のタンクローリーが顧客の灯油タンクを訪問し満タンまで給油する。タンクローリーには灯油積載上限があり、その値を超えない範囲で灯油が積載される。顧客への給油によってタンクローリーの灯油は減少する。タンクローリーが中間補給地点として定義される灯油の補給場所へ訪

表 1: 使用する制約式の係数

記号	内容
\mathcal{N}	顧客, 補給地点, デポを含むすべてのノード集合
\mathcal{N}'	顧客, 補給地点のノード集合
\mathcal{P}	顧客のノード集合
\mathcal{F}	補給地点のノード集合
s	出発地点を示すノード
e	帰還地点を示すノード
\mathcal{D}	配送計画日数の集合
C_i	顧客 i の灯油タンク容量
Q	タンクローリーの灯油積載上限
H	合計作業時間の上限
L_i	顧客 i の灯油タンク残量の限界値
α^d	日付 d が営業日かを示す 0-1 整数
β	単位給油量当たりの給油時間
δ_i	ノード i における固定作業時間
τ_{ij}	ノード i から j への移動時間
c_i^d	日付 d での顧客 i の灯油消費量

表 2: 使用する変数

記号	内容
I_i^d	日付 d における顧客 i の灯油タンク残量
q_i^d	日付 d における顧客 i への給油量
f_i^d	日付 d におけるノード i でのタンクローリーの空き容量
T_i^d	日付 d においてノード i に到達した際の合計作業時間
x_{ij}^d	日付 d において i から j までの移動が存在するかを示す 0-1 変数
y_i^d	日付 d において顧客 i に配送が行われたかどうかを示す 0-1 変数
z^d	日付 d において車両が使用されたかどうかを示す 0-1 変数

れることで、タンクローリーの灯油を満タンまで補充される。給油先の灯油タンクについては、顧客 1 人当たりに対して決まった容量の灯油タンクが備え付けられており、既知としてあらかじめ把握されている灯油消費量に基づいてタンク残量が減少する。ここで、灯油配送計画問題は、いかに顧客の灯油タンク切れを起さずに、1 日当たりの作業時間上限時間が決まっている中で効率よく、つまり、少ない作業時間で給油を行うかという問題になる。

3.1 使用する変数と制約式の係数

灯油配送計画問題の定式化に必要な変数と制約式の係数を表 1, 表 2 に示す。

定式化にあたり顧客, 中間補給地点, 出発地点, 帰還地点からなるノード集合 \mathcal{N} を定義する。続いて、日数に関する内容としては、配送計画日数分の各日数を示す \mathcal{D} , 日付 d が営業可能かを示す α^d を定義する。灯油タンクに関する内容としては、顧客 i の日付 d における灯油消費量 c_i^d , 灯油タンク残量 I_i^d , 及び灯油タンク容量 C_i , 灯油タンク残量の許容しきい値 L_i を定義する。タンクローリーに関する内容としては、タンクローリーの灯油積載上限量 Q , 日付 d におけるノード i 訪問時のタンクローリーの残量 f_i^d ,

日付 d における顧客 i へ給油量 q_i^d を定義する. 作業時間に関する内容としては, ノード i における固定作業時間 δ_i , ノード i から j への移動時間 τ_{ij} , 単位給油量当たりの給油時間 β , 合計作業時間の上限 H , 日付 d における顧客 i に到着時の合計作業時間 T_i^d を定義する. 最後に, 日付 d におけるノード i から j への移動を示す x_{ij}^d , 顧客 i への訪問を示す y_i^d , タンクローリーの使用を示す z^d を定義する.

3.2 目的関数

本論文で扱う灯油配送計画問題の目的関数は式 (1) で表すように, 帰還地点に到達した際の合計作業時間, つまり, 1 日の合計作業時間の総和の最小化として定義する.

$$\min \sum_{d \in \mathcal{D}} T_e^d \quad (1)$$

3.3 制約式

灯油配送計画問題における制約式は, 表現される内容別に以下の 5 つの項目に分けることができる.

- ノードに関する制約式
- 灯油タンク残量と給油量に関する制約式
- タンクローリーの残量に関する制約式
- 合計作業時間に関する制約式
- 使用変数の定義域に関する制約式

本節では, グループごとに制約式を分けて説明する.

3.3.1 ノードに関する制約式

ノードに関する制約式を, 式 (2) ~ (5) で定義する. 式 (2), (3) は出発地点及び帰還地点の制約式であり, 営業可能な日程でなおかつ, タンクローリーが使用される場合のみみ出発地点, 帰還地点からの移動が存在することを示す. 式 (4) は移動を示す x_{ij}^d と訪問を示す y_i^d をつなぐ制約式であり, 式 (5) は顧客及び補給地点の流入ノード本数と流出ノード本数を一致させる制約式である.

$$\sum_{j \in \mathcal{N}'} x_{sj}^d = \alpha^d z^d \quad d \in \mathcal{D}, \quad (2)$$

$$\sum_{i \in \mathcal{N}'} x_{ie}^d = \alpha^d z^d \quad d \in \mathcal{D}, \quad (3)$$

$$y_i^d = \sum_{j \in \mathcal{N} \setminus \{s\}} x_{ij}^d \quad i \in \mathcal{P} \quad d \in \mathcal{D}, \quad (4)$$

$$\sum_{j \in \mathcal{N} \setminus \{s\}} x_{ij}^d = \sum_{j \in \mathcal{N} \setminus \{e\}} x_{ji}^d \quad i \in \mathcal{N}' \quad d \in \mathcal{D}, \quad (5)$$

3.3.2 灯油タンク残量と給油量に関する制約式

続いて, 式 (6)~(11) では灯油タンク残量と給油量に関する制約式を定義する. 式 (6) では灯油タンク残量が定義され, 式 (7) ではタンク残量の上界が定義される. 式 (8)~(10) については, 顧客への給油量を定義しており, 給油が行われる場合には灯油タンクを満タンまで給油するように給油量が決定され, 給油が行われない場合は給油量は 0 と

なる. 式 (11) では顧客以外のノードである補給地点, 出発地点, 帰還地点での給油量は 0 となるように制約を定義している.

$$I_i^d = I_i^{d-1} + q_i^d - c_i^d \quad i \in \mathcal{P} \quad d \in \mathcal{D}, \quad (6)$$

$$I_i^d \leq C_i \quad i \in \mathcal{P} \quad d \in \mathcal{D}, \quad (7)$$

$$q_i^d \leq C_i - I_i^{d-1} \quad i \in \mathcal{P} \quad d \in \mathcal{D}, \quad (8)$$

$$q_i^d \geq C_i y_i^d - I_i^{d-1} \quad i \in \mathcal{P} \quad d \in \mathcal{D}, \quad (9)$$

$$q_i^d \leq C_i y_i^d \quad i \in \mathcal{P} \quad d \in \mathcal{D}, \quad (10)$$

$$q_i^d = 0 \quad i \in \mathcal{N} \setminus \mathcal{P} \quad d \in \mathcal{D}, \quad (11)$$

3.3.3 タンクローリーの残量に関する制約式

式 (12)~(15) ではタンクローリーの残量に関する制約式を定義する. 式 (12) では, タンクローリーの残量がタンクローリーの積載上限を超えないように上界を定義している. 式 (13) では, 顧客への給油時のタンクローリー残量を定義しており, 式 (14) では補給地点でのタンクローリー残量をを定義している. 式 (15) では帰還地点でのタンクローリー残量は次の日付に引き継がれることを定義している. 式 (16) では顧客への給油が存在しない日付において, タンクローリー残量を変わずに保持する制約を定義している.

$$f_i^d \leq Q \quad i \in \mathcal{N} \setminus \mathcal{F} \quad d \in \mathcal{D}, \quad (12)$$

$$f_i^d + q_i^d \leq f_j^d + Q(1 - x_{ij}^d) \quad i \in \mathcal{N} \setminus \{e\} \quad j \in \mathcal{P} \quad d \in \mathcal{D}, \quad (13)$$

$$f_i^d = 0 \quad i \in \mathcal{F} \quad d \in \mathcal{D}, \quad (14)$$

$$f_s^d = f_e^{d-1} \quad d \in \mathcal{D}, \quad (15)$$

$$f_e^d \geq f_s^d - Q \sum_{j \in \mathcal{N}'} x_{sj}^d \quad d \in \mathcal{D}, \quad (16)$$

3.3.4 合計作業時間に関する制約式

式 (17)~(19) では合計作業時間に関する制約式を定義する. 式 (17) では, 各ノードにおける到達時までの合計作業時間を定義している. 式 (18)~(19) では 1 日の合計作業可能時間を定義しており, 作業開始時刻は 0 として固定する.

$$\begin{aligned} T_i^d + \tau_{ij} + \delta_i + \beta q_i^d \\ \leq T_j^d + (H + \tau_{ij} + \delta_i + \beta C_i)(1 - x_{ij}^d) \end{aligned} \quad d \in \mathcal{D} \quad i \in \mathcal{N} \setminus \{e\} \quad j \in \mathcal{N} \setminus \{s\}, \quad (17)$$

$$T_e^d \leq H \quad d \in \mathcal{D}, \quad (18)$$

$$T_s^d = 0 \quad d \in \mathcal{D}, \quad (19)$$

3.3.5 使用変数の定義域に関する制約式

最後に, 使用する変数の定義域を式 (20)~(26) で定義する.

$$I_i^d \geq L_i \quad i \in \mathcal{P} \quad d \in \mathcal{D}, \quad (20)$$

$$q_i^d \geq 0 \quad i \in \mathcal{N} \quad d \in \mathcal{D}, \quad (21)$$

$$f_i^d \geq 0 \quad i \in \mathcal{N} \quad d \in \mathcal{D}, \quad (22)$$

$$T_i^d \geq 0 \quad i \in \mathcal{N} \quad d \in \mathcal{D}, \quad (23)$$

$$x_{ij}^d \in \{0, 1\} \quad i, j \in \mathcal{N} \quad d \in \mathcal{D}, \quad (24)$$

$$y_i^d \in \{0, 1\} \quad i \in \mathcal{P} \quad d \in \mathcal{D}, \quad (25)$$

$$z^d \in \{0, 1\} \quad d \in \mathcal{D} \quad (26)$$

4. タブーサーチを使用したヒューリスティクス

本論文では、灯油配送計画問題の近似最適解を導出する手法の1つとしてタブーサーチ [8] を提案する。タブーサーチはメタヒューリスティクスの1種であり、近傍探索による解の改善（移動操作）をメモリに記憶し、記憶された移動操作を禁止することによって、局所解への停滞を回避する手法である。タブーサーチは様々な配送計画問題にて有効性が確認されており、Prescott らによるディーゼルオイルの配送計画問題を扱った研究 [6] においても、タブーサーチを使用したヒューリスティックが問題に対して有効であることが確認されている。

タブーサーチを始めとするメタヒューリスティクスでは、基本となる手法のほかに初期解の用意あるいは初期解生成アルゴリズムの定義、及び近傍の定義が必要である。本論文ではタブーサーチの初期解生成アルゴリズムとして、貪欲法に基づく手法を提案し、近傍として「挿入」と「削除」及びこれら2つを組み合わせた「削除+挿入」の3種類を定義する。

4.1 初期解生成アルゴリズム

貪欲法に基づく初期解生成アルゴリズムについては、日付 d においてその日に給油を行わなければ灯油残量が灯油タンク残量の限界値 L_i を下回ってしまうような給油必須顧客をあらかじめ計算し、その顧客を解として加えていく。解として加える際には、現在の地点から最も移動時間及び作業時間が小さくなる顧客を選択していく。この時、タンクローリーの残量が灯油タンクの満タンまでの給油量に満たない場合が発生するため、そうなった場合にはランダムに補給地点1つを選び解に追加し、直後に給油予定であった顧客を解に追加する。以上の操作を、給油必須顧客がなくなるまで続けたものが初期解として出力される。

4.2 近傍

続いて、近傍は以下の3つを用いて定義する。

- **挿入**：日付 d において配送順に加わっていない顧客、または補給地点を1つ選択しルートに追加する
- **削除**：日付 d における配送順のノードを1つ選択し削

除する

- **削除+挿入**：日付 d における配送順のノードを1つ選択し削除したのち、まだ配送順に加わっていない顧客を1つ選択しその位置に挿入する

4.3 特別選択基準

本研究では、タブーサーチの性能改善のために特別選択基準 [8] を用いる。特別選択基準とは、タブーリストに格納されている禁止されている移動についても評価を行い、これまでの解よりも良い結果が出たものについては、例外的に解として認める手法である。

4.4 ペナルティ

本論文では灯油配送計画問題に存在する3つの制約に関するペナルティを導入する。

- **作業時間超過ペナルティ V^T** ：合計作業時間 T_e^d が合計作業可能時間 H を超過した値の合計であり以下の数式で表現する。

$$V^T = \sum_{d \in \mathcal{D}} E^T(d), \quad (27)$$

$$E^T(d) = \begin{cases} T_e^d - H & \text{if } T_e^d > H, \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

- **給油量不足ペナルティ V^f** ：ある日付 d におけるノード i 訪問時のタンクローリー残量 f_i^d が0を下回っている値の合計であり、以下の数式で表現する。

$$V^f = \sum_{d \in \mathcal{D}} \sum_{i \in \mathcal{P}} E^f(i, d), \quad (29)$$

$$E^f(i, d) = \begin{cases} -f_i^d & \text{if } f_i^d < 0, \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

- **灯油タンク残量不足ペナルティ V^I** ：日付 d における顧客 i の灯油タンク残量 I_i^d が、設定した灯油タンク残量限界値 L_i を下回った値の合計であり、以下の数式で表現する。

$$V^I = \sum_{d \in \mathcal{D}} \sum_{i \in \mathcal{P}} E^I(i, d), \quad (31)$$

$$E^I(i, d) = \begin{cases} L_i - I_i^d & \text{if } I_i^d < L_i, \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

これらのペナルティを用いたタブーサーチにおける制約に違反する解の評価値は、各ペナルティにおける係数を w とすると以下の F より算出する。

$$F = \sum_{d \in \mathcal{D}} T_e^d + w^T V^T + w^f V^f + w^I V^I \quad (33)$$

4.5 リペアリング

リペアリングは、メタヒューリスティクスによる近傍の

候補を探索する際に、ペナルティを取り除く処理である。本論文で扱う灯油配送計画問題では、4.4節で示したように3種類のペナルティが存在しており、そのうち灯油残量切れペナルティ、タンクローリー残量不足ペナルティに対するリペアリングを導入する。2つだけのリペアリングするのかの理由については、それぞれのペナルティを取り除くための操作が、労働時間ペナルティでは「削除」、灯油残量不足ペナルティ、タンクローリー残量不足ペナルティでは「挿入」となっており、単純な処理によりすべてのペナルティを取り除くことが難しいためである。

まず、灯油残量不足ペナルティを解消するためのリペアリングについては、事前に灯油残量不足が発生している顧客と日付を取り出し、その日付の配送順にその顧客を挿入することでペナルティの解消を行う。続いて、給油量不足ペナルティのリペアリングについては、事前に給油量不足が起こった位置と日付を取り出し、その日付、位置について補給地点を直前に挿入することでペナルティの解消を行う。灯油タンク残量不足ペナルティの場合は、挿入する位置、給油量不足ペナルティの場合はどの補給地点に挿入するかについての候補が複数あるため、節約値と呼ばれる地点の挿入によってどの程度距離が増加したかを示す値を計算し、節約値が最小となる候補を選択し、リペアリングを実行する。

5. 実験

本論文では、提案した灯油配送計画問題に対するタブーサーチの有効性を検証する目的で実験を行った。具体的には、札幌市西区、タンクローリー1台程度の規模となる灯油配送計画問題に対して、提案手法によって解が求められるかどうかを検証する目的で実験を行った。実験には、あらかじめ最適解の評価値の上界の算出が可能な灯油配送計画インスタンスおよび、実データを基に作成したインスタンスを使用した。基礎インスタンスではMIPソルバにより厳密解を求め、あらかじめ算出された最適解の評価値の上界が妥当であることを示すとともに、提案手法が算出した近似解法と実行時間を比較した。実規模配送件数インスタンスでは、最適解の評価値の上界と提案手法との比較を行った。実データを基に作成したインスタンスでは、得られた評価値や配送計画に関する妥当性の検証を行った。

5.1 インスタンス作成

本節では、実験で使用した灯油配送計画インスタンスの作成について述べる。本論文では、事前に最適解の評価値の上界の計算が可能なインスタンスと、実データを基に作成したインスタンスの2種類を作成した。

5.1.1 最適解の評価値の上界が計算可能なインスタンス

あらかじめ最適解の評価値の上界を算出可能なインスタンスについては、1日に配送すべき顧客をクラスタとして

密集させ、ある1日はある顧客クラスタへの配送が最適解に近づくように作成した。これを可能とするために、各クラスタと各配送する日程を対応付け、そのクラスタ内の顧客はその日程以降に残量不足を起こすように初期タンク残量と消費量を調整した。また、1クラスタ以外への配送が最適な配送条件とならないように、十分クラスタ間の距離を離して配置した。この上界の計算が可能なインスタンスについては、基礎インスタンスと実規模配送件数インスタンスの2種類としてさらに細分化したものを作成し実験に使用した。

5.1.1.1 基礎インスタンス

基礎インスタンスについては、顧客クラスタの数を2に設定し、顧客の数、計画日数、クラスタ内での各顧客の消費量にばらつきがあるかの3項目を変化させ合計10個用意した。顧客の数は3件から8件、計画日数は2日から4日の間で定義し、クラスタの個数はすべてのインスタンスで2つとして定義した。

5.1.1.2 実規模配送件数インスタンス

実規模配送件数インスタンスについては、実際の配送状況と1日の配送件数を揃えたインスタンスとなっており、顧客クラスタの個数を7に設定し、7日間の計画問題を作成した。顧客クラスタ内の顧客数は10から30件まで変化させたものを用意した。

5.1.2 実データを基に作成したインスタンス

実データを基に作成したインスタンスについては、実際の配送データを基にインスタンスを作成しており、札幌市西区、1タンクローリーの配送履歴データから2020年12月6日～12月12日、2021年1月18日～1月24日の1週間を切り取り作成した。作成に当たっては、配送対象となる顧客を取り出す基準として、灯油残量がタンクサイズの1割、3割を計画期間中に下回った顧客を対象として選択した。2020年12月6日～12日のインスタンスについては、1割の場合は10件、3割の場合は22件の顧客が配送対象として選択され、2021年1月18日～24日のインスタンスについては、1割の場合は33件、3割の場合は66件の顧客が配送対象として選択された。実データを基に作成したインスタンスについては、1日の作業時間が8時間を超えず、期間中に顧客が設定した残量しきい値を下回らず、給油量がタンクローリーの残量を超えない場合に実行可能解となる。

5.2 実験設定

タブーサーチ、リペアリングを使用したヒューリスティクスに関するパラメータは事前実験の結果をもとに設定した。設定パラメータについては表3に掲載する。MIPソルバについてはMarkovらによるごみ収集配送計画問題に関する研究[4]においても使用されているCPLEXのバージョン20.1.0.0をデフォルトの設定で使用した。各手法の

表 3: タブーサーチのパラメータ

パラメータ	設定値
タブー長	100
最大試行回数	500
作業時間超過ペナルティ係数 (w^T)	1,000
給油量不足ペナルティ係数 (w^f)	1,000
灯油残量不足ペナルティ係数 (w^l)	1,000

実行時間は、実際の支援システムでの利用を想定し、12時間を制限時間として実行した。MIP ソルバを使用した実験については、顧客の並び順をランダムに変更した3試行の実行時間の平均値を結果として使用し、ヒューリスティクスを使用した実験については、乱数シード値を変更した3試行の実行時間および各試行で得られた最良解の評価値の平均値を使用した。

実験には AMD Ryzen 7 3700X CPU, 16GB RAM を搭載した計算機を使用し、オペレーティングシステムは Ubuntu 20.04 を使用した。

6. 実験結果および考察

6.1 基礎インスタンス実験の結果および考察

基礎インスタンスでの実験結果を表 4 に示す。各表の列について、 $n()$ は集合の要素数を返す関数であり、左から $n(P)$ が顧客数、 $n(D)$ が計画日数を表す列となっている。続いて、消費量のばらつきについては、作成した各クラスタ内における顧客の消費量が一定ではなく、ばらつきが存在するかどうかを示す列である。Obj* については、インスタンス生成時にあらかじめ計算した最適解の評価値の上界を示す列となっている。Time に関しては MIP ソルバの場合は最適解が求まるまでの実行時間 (秒)、タブーサーチの場合は最大試行回数までの実行時間 (秒) を示しており、数値が記入されていない部分については MIP ソルバが制限時間内に最適解を求められなかったことを示している。Obj については、それぞれの手法が算出した最も良い評価値である。ここで、MIP ソルバに関する評価値について、評価値の後ろに * がついている行に関しては、MIP ソルバが最適解の保証に成功したことを示している。なお、最適解の導出に成功した * のつく結果については、3 試行すべての結果で最適解が得られており、実行時間は 3 試行の平均値となっている。また、顧客数が 6 件、計画期間日数が 3 件と 4 件のインスタンスについては、どちらも、3 試行中 2 回のみ実行可能解が得られたので、実行可能解の中で平均値を計算している。GAP^{Obj} については、タブーサーチ及び ALNS の評価値と最適解の評価値の上界 Obj* との間のギャップを表す列となる。

6.1.1 MIP ソルバの実験結果および考察

表 4 より、MIP ソルバを使用した実験結果について、特に顧客数が 8 件で計画期間が 4 日のインスタンスでは制限

時間内に実行可能解を導出することができなかった。また、実行時間については、顧客数が 4 件、計画日数が 2 日、クラスタ内の消費量にばらつきがないインスタンスでは 0.30 秒であったのに対して、顧客数を 2 件増やした顧客数が 6 件のインスタンスでは、23.23 秒、さらに 2 件増やした顧客数が 8 件のインスタンスでは 14102.60 秒 (約 2 時間 55 分) となっている。この結果から、顧客の数に応じて実行時間が急激に増大することが推測される。実務規模の灯油配送計画問題は 1 台の車両あたり約 600 件の顧客、計画期間は 1 週間あるいはそれ以上となるため、今回の実験結果を踏まえると、MIP ソルバを用いて実務規模の問題を直接解くことは難しいと考えられる。

続いて、算出した最適解の評価値の上界と得られた評価値の比較を行うと、解が得られたすべてのインスタンスにおいては算出した最適解の評価値の上界と値が一致しているため、求めた上界が最適値となっていることが分かる。ただし、実行可能解のみが求まっているインスタンスや実行可能解が求まっていないインスタンスについては、最適解の評価値の上界が最適値と一致するものであるという保証はされない。しかし、ほかのインスタンスでは正しく最適値が求まっていたことを踏まえると、タブーサーチ、ALNS の有効性の検討をあらかじめ算出された最適解の評価値の上界との比較により行うことは妥当と考えられる。

6.1.2 タブーサーチの実験結果および考察

表 4 より、タブーサーチにより求められた実行可能解の評価値は最適解の評価値の上界からのギャップがすべて 5% 以内に収まっていることがわかる。今回のインスタンスは、顧客数が少ないことや顧客の位置座標分布が恣意的であるという点で内容が現実のものとは異なるが、最適値からのギャップが 5% 以内に収まっていることは、提案されたタブーサーチが灯油配送計画問題に対して十分な性能を発揮することを示唆している。

タブーサーチの実行時間については、最短のものが顧客数 4 件、計画日数 2 日、消費量のばらつきがクラスタ内で一致するインスタンスで 1.99 秒、最長のものが顧客数 8 件、計画日数 4 日、消費量のばらつきがクラスタ内で一致するインスタンスで 18.45 秒となった。今回得られた実行可能解の評価値と実行時間の関係を MIP ソルバで得られた結果と比較すると、インスタンスの規模が小さいものでは MIP ソルバの方が少ない時間で最適解を求められているのに対し、規模が大きくなるケースではタブーサーチは実行時間を抑えながら算出した最適解の評価値の上界に近い評価値の導出に成功していることが分かる。つまり、タブーサーチの場合はより大きな実務規模のインスタンスに対しても高速により実行可能解が得られる可能性が高いと考えられる。

表 4: MIP ソルバ及びタブーサーチの実験結果

インスタンス				MIP ソルバ		タブーサーチ			Gap ^{Time} [s]
$n(P)$	$n(D)$	消費量 ばらつき	Obj*	Time [s]	Obj	Time [s]	Obj	Gap ^{Obj} [%]	
4	2	無し	86.66	0.30	86.66*	1.99	86.66	0.0	-1.69
6	2	無し	110.91	23.23	110.91*	4.98	110.91	0.0	18.25
8	2	無し	134.08	—	134.08	5.70	135.85	1.3	—
3	3	無し	116.83	0.37	116.83*	3.21	120.88	3.3	-2.84
6	3	無し	171.97	—	171.97	8.02	174.34	1.3	—
4	4	無し	178.33	1.16	178.33*	5.17	180.33	1.1	-4.01
8	4	無し	275.14	—	—	18.45	282.73	2.6	—
4	4	有り	86.66	1.32	86.66*	5.12	86.66	0.0	-3.80
6	4	有り	110.08	—	110.08	10.12	110.08	0.0	—
8	4	有り	134.08	—	—	15.56	134.08	0.0	—

表 5: 実規模配送件数インスタンスの実験結果

クラスター内 顧客数	N_{fs}	$N_{fs}^{R_a}$	$N_{fs}^{R_o}$	Gap ^{Obj} [%]	Gap ^{R_a} [%]	Gap ^{R_o} [%]
10	0	2	1	—	-1.3	0.5
20	0	1	3	—	-3.0	-4.9
30	0	0	0	—	—	—

6.2 実規模配送件数インスタンスの実験結果および考察

表について、 N_{fs} はリペアリングを適用しない設定で実行した際に求まった実行可能解の数、 $N_{fs}^{R_a}$ はリペアリングを解に適用する設定で実行した際に求まった実行可能解の数、 $N_{fs}^{R_o}$ はリペアリングを解に適用せず、評価の際にだけ適用する設定で実行した際に求まった実行可能解の数、 Gap_{Obj} はリペアリングを適用しない設定で実行した際に得られた最良解の評価値と最適解の評価値の上界とのギャップの平均、 $Gap_{Obj}^{R_a}$ はリペアリングを解に適用する設定で実行した際に得られた最適解の評価値の上界とのギャップの平均、 $Gap_{Obj}^{R_o}$ はリペアリングを解に適用せず、評価の際にだけ適用する設定で実行した際に得られた最適解の評価値の上界とのギャップの平均である。

表 5 より、求まった実行可能解の数を確認すると、1 日の配送件数が 10 件のインスタンスにおいて、リペアリングを解に適用した設定で 2 回、リペアリングを評価にのみ使用した設定で 1 回、1 日の配送件数が 20 件のインスタンスにおいて、リペアリングを解に適用した設定で 1 回、リペアリングを評価にのみ使用した設定で 3 回実行可能解が求まった。また、実行可能解が得られた場合において、最良解の評価値と最適解の評価値の上界とのギャップを確認すると、すべてが 0.5% 以内に収まっていることが確認できる。つまり、実規模配送件数インスタンスでは、1 日 20 件配送までの設定においては、リペアリングを適用したタブーサーチによって最適解の評価値の上界に近い値となる評価値を持つ最良解が求められた結果となった。

以上より、1 日 20 件程度の配送件数を扱う実問題に対して、リペアリングを適用したタブーサーチが制限時間内に最適解の評価値の上界に近い値を導出できる可能性が高い

表 6: 実データを基に作成したインスタンスの実験結果

計画開始日	残量 しきい値	Obj	Obj ^{R_a}	Obj ^{R_o}
2020 年 12 月 6 日	0.1	155.8	155.9	103.1
	0.3	243.9	348.0	240.7
2021 年 1 月 18 日	0.1	448.0	444.8	353.0
	0.3	753.0	840.1	654.2

といえる。

6.3 実データを基に作成したインスタンスの実験結果および考察

実データを基に作成したインスタンスの実験結果を表 6 に示す。表 6 について、 Obj はリペアリングを適用しない条件での最良の評価値 (分)、 Obj^{R_a} はリペアリングを解に適用する条件での最良の評価値 (分)、 Obj^{R_o} はリペアリングを評価の際にのみ適用する条件での最良の評価値 (分) を示す。なお、求まった最良解はすべて実行可能解となった。

まず、表 6 より得られた最良の評価値について確認すると、最小のもので 103.1、最大のもので 840.1 となった。これはつまり、最小の場合は 1 週間の配送に 103.1 分 (約 1 時間 43 分)、最大の場合は 1 週間に 840.1 分 (約 14 時間) かかるということになる。この結果について考察すると、まず、現実での配送では 1 日平均で 40 件の配送が行われており、タンクサイズの約 4 割が給油されている。つまり、残量しきい値が 6 割を下回った顧客に対して 40 件の配送がおこなわれているといえる。一方、今回設定した状況では、灯油タンク残量のしきい値が 1 割、あるいは 3 割を下回った場合に配送対象となるような状況となっているため、配送対象となる顧客が現実の配送状況と異なり、全体的に少なくなったといえる。理想的な配送状況としては、今回の実験設定のようなできるだけ残量が少ない状況での給油が望ましいことから、今回実験で行った残量が 1 割、または 3 割を下回った顧客を配送対象とする配送計画が実際の灯油配送においてもより効率的な配送が行えるものだと考え

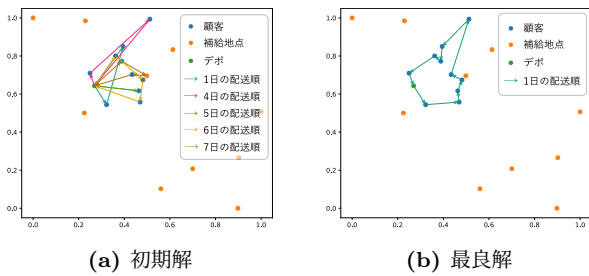


図 1: 得られた配送計画 (2020 年 12 月 6 日、しきい値 1 割)

られる。以上より、提案手法のヒューリスティクスは残量しきい値を低く設定した状況であれば、札幌市の西区、1 タンクローリー程度の規模において、上界に近い解が得られると考えられる。

続いて、得られた配送計画の評価を行うために、2020 年 12 月 6 日、灯油残量しきい値が 1 割のインスタンスにおいて、最も良い評価値となった評価の際にのみリペアリングを適用したタブーサーチが算出した配送計画、および貪欲法によって求めた初期解の配送計画を図 1 に示す。図 1 より、配送計画を比較すると、最良解の配送計画は 1 日にまとまった配送をしており、また、配送経路間で交差するような配送計画とはなっていないことが確認できる。一方で、初期解による配送計画については、複数の日付にまたがって、配送経路が交差するようなものとなっている。配送計画は、労働時間を超えなければ、まとめて 1 日で配送してしまっただけが出発、帰還に関する経路が節約でき、また、交差の発生しない計画については、同じ地点への出戻りのような経路が起こっていないことを示すため、良い計画であるといえる。そのため、手法が求めた最良解は無駄のない配送計画であるといえる。

7. おわりに

本論文では、灯油配送計画問題を定式化し、MIP ソルバ、タブーサーチによって求解を行った。タブーサーチについては、実行可能解を求めるための工夫としてリペアリングを導入した。結果として、ヒューリスティクスによる実行結果については、1 日の配送件数が 20 件、計画期間が 7 日のインスタンスまでの条件で最適解の評価値の上界に近い実行可能解が得られた。実データを基に作成したインスタンスについては、ヒューリスティクスによって札幌市西区、1 タンクローリー程度の規模において実行可能解の導出が可能であることを確認した。以上より、本論文で提案したヒューリスティクスは、使用した実データの業務規模となる、タンクローリーが 1 台、札幌市西区程度の灯油配送業務において、灯油残量しきい値を実情よりも低く設定し、1 日の配送件数が 20 件以内、計画期間が 7 日以内となる配送計画であれば、上界に近い解を導出できる可能性を示した。

今後は、実行可能解の導出ができなかった 1 日の配送件

数が 30 件以上の実規模配送件数インスタンスにおいて、実行可能解が求まるような手法を検討していきたい。また、インスタンス自体においても、配送件数を実問題に近づけるだけでなく、顧客の位置分布や灯油タンクのサイズといった内容も実問題に近づけていくことを検討したい。現実の配送業務で発生しているが取り込めていない問題として、配送が可能な時刻が決まっている顧客への対応としての時間窓の追加、移動時間や灯油消費量など誤差を含むパラメータへの対応としてのロバスト最適化やパラメータの確率分布化といった内容を検討していく。

参考文献

- [1] 椿康平, 横山想一郎, 山下倫央, 川村秀憲: 灯油配送の効率化に向けた状態空間モデルを用いた灯油残量の推定, 情報処理北海道シンポジウム 2020 (2020).
- [2] Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., Mack, R. G. and Prutzman, P. J.: Improving the Distribution of Industrial Gases with an ON-LINE Computerized Routing and Scheduling Optimizer, *Interfaces*, Vol. 13, No. 6, pp. 4–23 (online), available from (<http://www.jstor.org/stable/25060491>) (1983).
- [3] Coelho, L. C., Cordeau, J.-F. and Laporte, G.: Thirty Years of Inventory Routing, *Transportation Science*, Vol. 48, No. 1, pp. 1–19 (online), DOI: 10.1287/trsc.2013.0472 (2014).
- [4] Markov, I., Varone, S. and Bierlaire, M.: Integrating a Heterogeneous Fixed Fleet and a Flexible Assignment of Destination Depots in the Waste Collection VRP With Intermediate Facilities, *Transportation Research Part B: Methodological*, Vol. 84, pp. 256–273 (online), DOI: <https://doi.org/10.1016/j.trb.2015.12.004> (2016).
- [5] Markov, I., Bierlaire, M., Cordeau, J.-F., Makhnoon, Y. and Varone, S.: Waste Collection Inventory Routing With Non-Stationary Stochastic Demands, *Computers & Operations Research*, Vol. 113, p. 104798 (online), DOI: <https://doi.org/10.1016/j.cor.2019.104798> (2020).
- [6] Prescott-Gagnon, E., Desaulniers, G. and Rousseau, L. M.: Heuristics for an Oil Delivery Vehicle Routing Problem, *Flexible Services and Manufacturing Journal*, Vol. 26, No. 4, pp. 516–539 (online), DOI: 10.1007/s10696-012-9169-9 (2014).
- [7] Bertsimas, D. J., Gupta, S. and Tay, J.: Scalable Robust and Adaptive Inventory Routing, *Optimization Online*, pp. 1–29 (2016).
- [8] Glover, F. and Laguna, M.: *Tabu Search*, Kluwer, MA (1997).
- [9] Shaw, P.: Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems, *Principles and Practice of Constraint Programming — CP98* (Maher, M. and Puget, J.-F., eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 417–431 (1998).
- [10] Schneider, M., Stenger, A. and Hof, J.: An Adaptive VNS Algorithm for Vehicle Routing Problems With Intermediate Stops, *Operations Research Spectrum*, Vol. 37, pp. 353–387 (online), DOI: 10.1007/s00291-014-0376-5 (2015).