

タブーサーチによる中間補給を考慮した灯油配送計画問題の最適化

A Tabu Search Heuristic for the Kerosene Delivery Inventory Routing Problem with Intermediate Replenishment Facilities

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

Hiroataka OOE¹, Soichiro YOKOYAMA², Tomohisa YAMASHITA², Hidenori KAWAMURA² and Mitsuo TADA³

¹ 北海道大学 大学院情報科学院

¹ Graduate School of Information Science and Technology, Hokkaido University

² 北海道大学 大学院情報科学研究院

² Faculty of Information Science and Technology, Hokkaido University

³ ゼロスペック株式会社

³ ZeroSpec Inc.

Abstract: A kerosene delivery is the business that delivers kerosene for heating, which is mainly used in cold regions. In recent years, the workload of a kerosene delivery driver has increased due to the decrease in the number of kerosene delivery companies, thus running out of kerosene in a customer's tank caused by the delivery delays could become a serious problem. We formulate the kerosene delivery inventory routing problem to address the above real-world problem. We show that our tabu search heuristic can solve small-sized instances in a reasonable time while a MIP solver is hard to solve the instances in a reasonable time.

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に決めていく。こういった、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おり、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。

灯油配送業を取り巻く問題としては、近年人手不足によって、一人当たりの労働負荷が上昇していることが挙げられている。この労働負荷、および、労働時間の増加がさらなる呼び水となり、離職率の増加や、灯油配送遅れによる灯油切れが生じてしまう危険性があ

る。灯油切れにより、冬場に暖房が使用できない状況は当然、室温が下がり、生活を行う上での支障を来す可能性があるということはもちろん、ロードヒーティングが行えなくなることで、道路上で発生する事故の増加も考えられる。さらに、一度灯油を切らしてしまうと、灯油用のホース内に空気が詰まってしまう、それを取り除くために30分程度の時間を要するため、灯油を切らさないための配送計画の作成が重要となる。

本論文では、上記の灯油配送業務における問題を解決する目的で、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、灯油配送計画問題の求解を行う。具体的に、灯油配送計画問題は中間補給を考慮した在庫配送問題(IRP-IRF)として定式化することができる。この問題に対して、タブーサーチによる近似解法を提案し、その有効性を検証する。あらかじめ最適値の上界が算出可能なインスタンスを作成して、MIPソルバによる厳密解の求解、及びタブーサーチによる近似解の求解を行い、実行時間及び解の評価値を比較する。

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の解説を行う。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

*連絡先：北海道大学 大学院情報科学院
〒060-0814 札幌市北区北14条西9丁目
ooe@ist.hokudai.ac.jp

の解説を行い、第4章では提案手法であるタブーサーチの解説を行う。第5章では手法を検証するための実験についての解説を行い、第6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)[1, 2] 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) 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 [3, 4] 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 [5, 6] が存在する。Prescott らが行う研究 [5] 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 [7] と巨大近傍探索 (LNS)[8] 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 [6] 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 [9, 3] 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 灯油配送計画問題の定式化

本章では、灯油配送計画問題の定式化について考える。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していく。

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に給油を行っていく。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ている。そして、顧客への給油によってタンクローリーの灯油は減っていく。タンクローリーの灯油残量が少なくなってきた際には適宜、拠点ではなく中間補給地点として定義される灯油の補給場所へ訪れ、タンクローリーの灯油を補充する。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っていく。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こさずに、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っていく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 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、

表 1: 使用する変数と制約式の係数

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

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, 表現したい内容別に以下の 5 つのグループ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

そのため, 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

まず, ノードに関する制約式については, 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かつ, タンクローリーが使用される場合のみ出発地点, 帰還地点からの移動が存在することを示している. 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の定義を行っており、式(7)ではタンク残量の上界を定義している。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。

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

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

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

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 タブーサーチによる灯油配送計画問題の最適化

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としてタブーサーチ [7] 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その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 [5] 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。

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。

続いて、近傍は以下の3つを用いて定義する。

- 挿入: 日付 d において給油されていない顧客、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 におけるルート内のノードを1つ選択し削除する
- 削除+挿入: 日付 d におけるルート内のノードを1つ選択し削除したのち、まだ給油されていない顧客を1つ選択しその位置に挿入する

メタヒューリスティクスには整数計画問題のような制約が存在しないため、そのままでは制約違反を問題に取り込むことができない。Cordeau らによるメタヒューリスティクスを車両配送計画問題、及びその派生問題に適用した研究 [10, 9, 4, 11] では、車両配送計画問題における制約違反を様々な形でペナルティとして定義している。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 を超過した日付の合計であり以下の数式で表現する

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

$$J^T(d) = \begin{cases} 1 & \text{if } T_e^d > H, \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

- 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を下回っているケースの合計であり、以下の数式で表現する

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

$$J^f(i, d) = \begin{cases} 1 & \text{if } f_i^d < 0, \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

- 灯油タンク残量不足ペナルティ V^l : 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 を下回るケースの合計であり、以下の数式で表現する

表 2: タブーサーチのパラメータ

| パラメータ | 設定値 |
|-------------------------|-------|
| タブー長 | 10 |
| 最大試行回数 | 1,000 |
| 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 |

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

$$J^l(i, d) = \begin{cases} 1 & \text{if } I_i^d < L_i, \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

これらのペナルティを用いたタブーサーチにおける評価値は、各ペナルティについて、どの程度評価値に影響させるかを決定するパラメータを w とすると以下の F より算出できる。

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

5 実験

本論文で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実験により検証した。タブーサーチにより得られた解の評価値の分析を可能とするためにあらかじめ最適値の上界の算出が可能な灯油配送計画問題のインスタンスを作成した。同一のインスタンスに対して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値の上界が妥当であることを示すとともに、タブーサーチによる近似解法と実行時間を比較した。

タブーサーチに関するパラメータは表2のように設定したものを使用し、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 [3] 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。MIPソルバの実行時間については、実験可能時間の都合から1時間に制限した。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、タブーサーチを使用した実験については、乱数シード値を変更した10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。

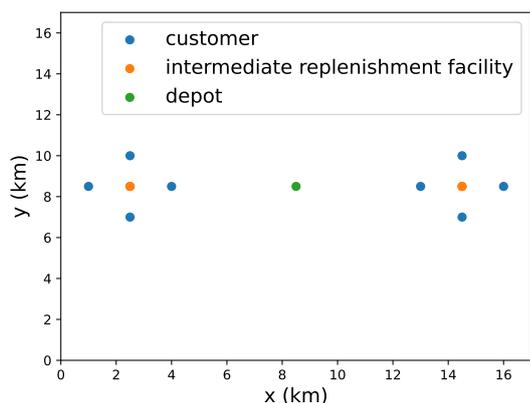


図 1: インスタンスの例

5.1 インスタンス作成

本実験ではあらかじめ最適値の上界を算出可能にすること目的に、インスタンスを作成した。

左右に顧客が密集するクラスタを配置する形でインスタンスを作成し、顧客は円状となるように配置させ、中心には補給地点を配置した。作成インスタンスの例を図 1 に示す。そのほかのパラメータについては、ある日付における給油がどちらか片方のクラスタのみとなるルートが最適となるように設定した。具体的には、ある日付 d において灯油の利用可能日数が小さい顧客がどちらかのクラスタにのみ存在するように消費量 D_i^d 及び初期タンク残量 I_i^0 を作成した。ここで、1 日のうちにクラスタをまたいだルートが生成される可能性を防ぐために、合計作業時間上限 H をクラスタ内のすべての顧客を給油した場合にかかる作業時間の中での最大のものを設定した。

本実験で使用した具体的なインスタンスは、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。

5.2 実験結果および考察

全体の実験結果を表 3 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、— で示す部分については 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 * が付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、タブーサーチの評価値と最適値の上界 Obj* との間のギャップを表す列となる。最後に、GAP^{Time} については、MIP ソルバとタブーサーチの実行時間のギャップを表す列となる。

5.2.1 MIP ソルバの実験結果および考察

表 3 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解が求まっていたことを踏まえると、タブーサーチの有効性の検討をあらかじめ算出された最適値の上界との比較により行うことは妥当と考えられる。

5.2.2 タブーサーチの実験結果および考察

表 3 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

表 3: 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] |
|------------------|------------------|-------------|--------|-------------|---------|-------------|--------|---------------------------|----------------------------|
| $n(\mathcal{P})$ | $n(\mathcal{D})$ | 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 | — |

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。

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。

タブーサーチについては、最大試行回数を増やすことで実行時間を増やすことができるため、今後は試行回数を増やした場合にどこまで実行時間に対して解の性能を高められるのかを検証する必要があると考えられる。

6 おわりに

本論文では、灯油配送計画問題を定式化し、MIPソルバ及びタブーサーチによって求解を行った。結果として、小規模でかつ恣意的に顧客を配置して作成したインスタンスであっても、MIPソルバは最適解を導出するまでに大きな時間を要することが分かった。一方、タブーサーチによる実行結果については、インスタン

スの規模が大きい場合であっても10秒程度で算出した最適値の上界に近い実行可能解が得られた。以上より、灯油配送計画問題は整数計画問題としては複雑で、MIPソルバによる厳密解法を本論文のように直接使用するだけでは大規模なインスタンスに対して妥当な解を現実的な時間で算出することは難しいと考えられる。タブーサーチを使用した近似解法については、小規模なインスタンスではあるものの、顧客数や計画期間数を増加させた場合においても、実行時間を同程度に抑えながら算出した最適値の上界とのギャップも低く抑えられていることが確認でき、より大規模なインスタンスにおいてもよい近似解が得られる見込みが高いと考えられる。

今後は、より大規模で実務規模のインスタンスに対して提案手法であるタブーサーチの有効性を検証したい。また、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、当日に非定期で発生する電話注文による緊急配送についての定式化といった内容を検討していく。

参考文献

- [1] Walter J. Bell, Louis M. Dalberto, Marshall L. Fisher, Arnold J. Greenfield, R. Jaikumar, Pradeep Kedia, Robert G. Mack, and Paul J. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, Vol. 13, No. 6, pp. 4–23, 1983.

- [2] Leandro C. Coelho, Jean-François Cordeau, and Gilbert Laporte. Thirty years of inventory routing. *Transportation Science*, Vol. 48, No. 1, pp. 1–19, 2014.
- [3] Iliya Markov, Sacha Varone, and Michel Bierlaire. 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, 2016.
- [4] Iliya Markov, Michel Bierlaire, Jean-François Cordeau, Yousef Maknoon, and Sacha Varone. Waste collection inventory routing with non-stationary stochastic demands. *Computers & Operations Research*, Vol. 113, p. 104798, 2020.
- [5] Eric Prescott-Gagnon, Guy Desaulniers, and Louis Martin Rousseau. Heuristics for an oil delivery vehicle routing problem. *Flexible Services and Manufacturing Journal*, Vol. 26, No. 4, pp. 516–539, 2014.
- [6] Dimitris J Bertsimas, Swati Gupta, and Joel Tay. Scalable Robust and Adaptive Inventory Routing. *Optimization Online*, pp. 1–29, 2016.
- [7] Fred Glover and Manuel Laguna. *Tabu Search*. Kluwer, MA, 1997.
- [8] Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In Michael Maher and Jean-François Puget, editors, *Principles and Practice of Constraint Programming — CP98*, pp. 417–431, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
- [9] Michael Schneider, Andreas Stenger, and Julian Hof. An adaptive VNS algorithm for vehicle routing problems with intermediate stops. *OR Spectrum*, Vol. 37, pp. 353–387, mar 2015.
- [10] Jean-François Cordeau, Michel Gendreau, and Gilbert Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, Vol. 30, No. 2, pp. 105–119, sep 1997.
- [11] Yuichi Nagata, Olli Bräysy, and Wout Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, Vol. 37, No. 4, pp. 724–737, 2010.