

2025 理論談話会 #02

5月8日

Online algorithms for the multi-vehicle inventory-routing problem with real-time demands

Luca Bertazzi, Guilherme O. Chagas, Leandro C. Coelho, Demetrio Laganà, Francesca Vocaturo. (2025).
Transportation Research Part C: Emerging Technologies

交通・都市・国土学研究室 / Behavior in Networks Lab.

M1 薬師神晴悟 / Seigo Yakushijin

概要

- 時間とともに需要が判明する多車両在庫ルーティング問題（IRP）に対し、整数計画問題を逐次的に解くオンラインアルゴリズムを提案する。
- 各時点で解かれる部分問題には、改良型のBranch-and-cutまたはマスヒューリスティックを適用する。
- 提案手法の性能を、競合比にもとづく解析によって、対応するオフライン問題の最適解と比較して評価する。
- 計算実験を通じて、本手法が平均的に高い性能を示し、大規模問題にも有効であることを示した。

新規性、有用性、信頼性

新規性

- 需要がリアルタイムに明らかになるO-IRPを初めて競争分析と計算機実験で評価。
- O-IRPを解くための将来情報量を調整可能なオンラインアルゴリズムのクラスを提案。

有用性

- 現実世界のサイズのインスタンスでも効果的に機能する。
- マスヒューリスティックを用いると高品質な解を大幅に短い時間で得られる。

信頼性

- 競合比解析により理論的にワーストケースにおける性能を保証。
- 様々なサイズにインスタンスを用いた計算実験により手法の有効性を経験的に検証。

1. Introduction

研究の背景

リアルタイム情報は緊急物資輸送やeコマース等、さまざまな場面で重要である。



緊急物資配送

Uber Eats
Demaecan

Eコマース即時配送



just-in-time

製造のJIT部品供給



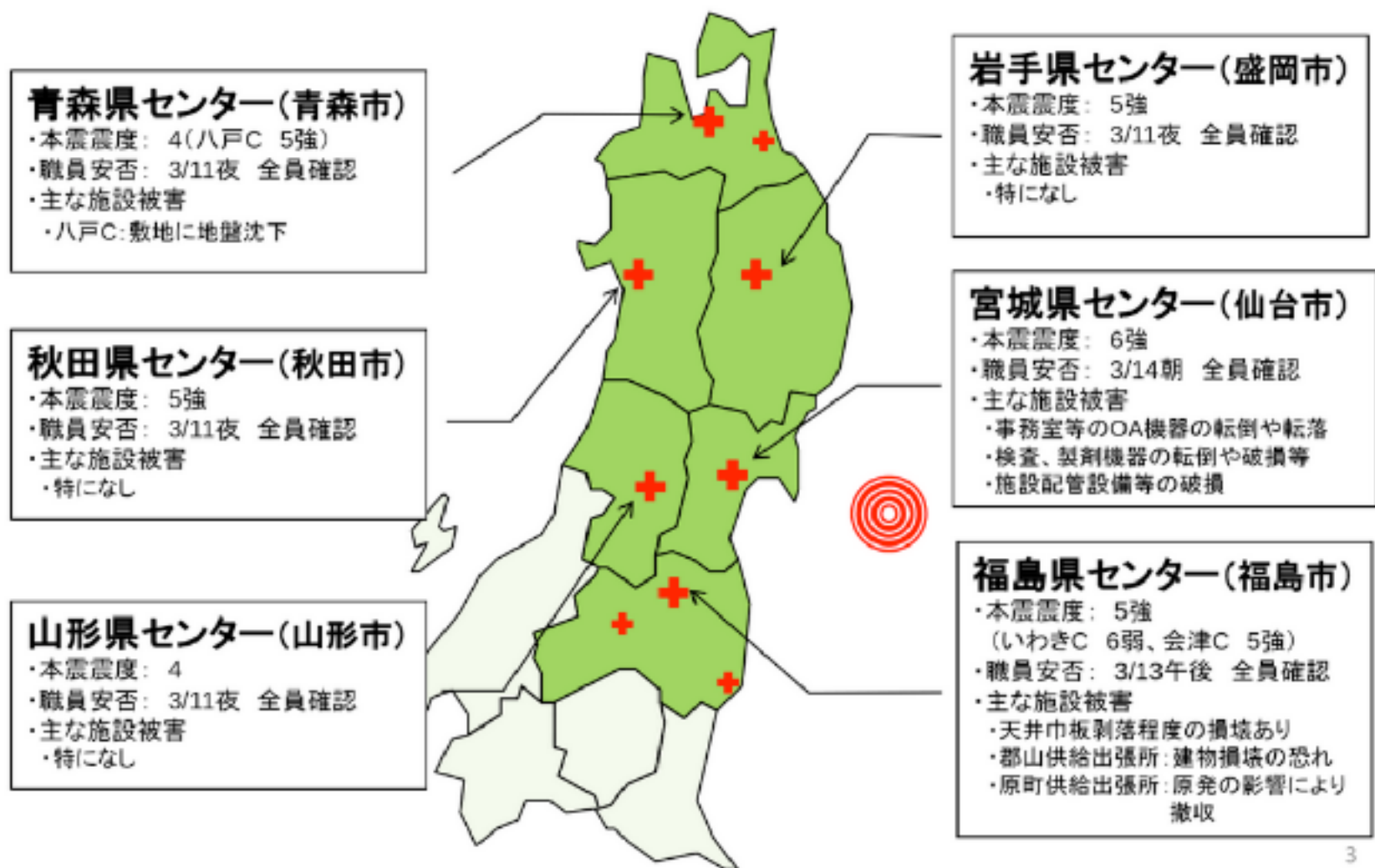
ラストワンマイル



緊急物資配送

東日本大震災における血液製剤の供給について

血液センターの被災状況



ライフラインの復旧と各業務の再開日

	ライフラインの復旧			献血受入再開日		検査・製剤再開日	
	電気	ガス	水道	固定施設	移動採血	検査	製剤
青森C	3/12	—	—	3/13	3/14	宮城C再開まで 東京Cで実施	3/13
八戸C	3/12	—	—	3/14			
岩手C	3/12	—	—	4/20	4/18	4/18	4/18
宮城C	3/12	3/27	3/22	4/18	5/1 (受入可能な会場 から再開)	4/13	4/13
秋田C	3/12	—	—	3/13	3/14	宮城C再開まで 東京Cで実施	3/13
山形C	3/12	—	—	3/13	3/14	宮城C再開まで 埼玉Cで実施	宮城C再開まで 新潟Cで実施
福島C	—	—	3/18	4/18	5/1 (受入可能な会場 から再開)	4/18	4/18
会津C	—	—	—	4/25			
いわきC	—	3/14	4/8	4/26			

検査: 東北各県の検査業務は宮城Cに集約
製剤: 岩手C、山形Cの製剤業務は宮城Cに集約

【被災地域における安定供給への影響】

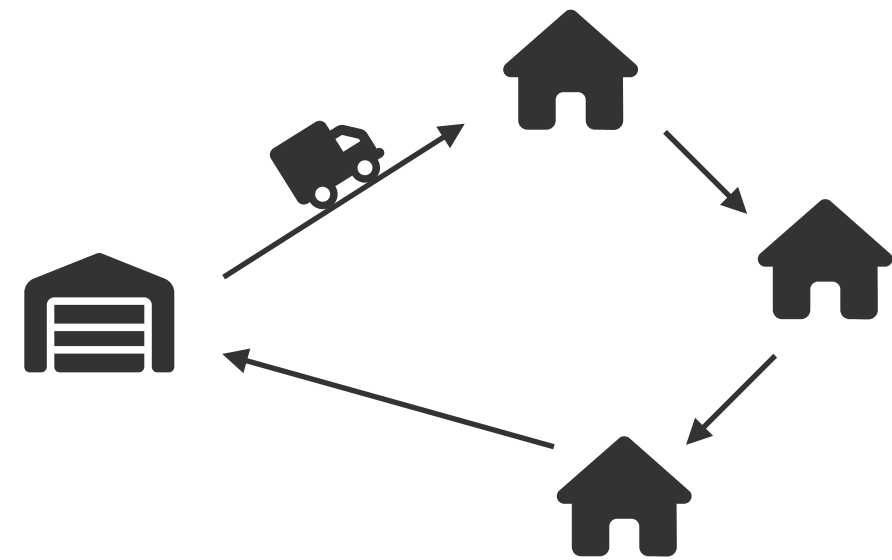
- 医療機関と連絡がとれない (有線電話・携帯電話ともに不通)
→ 医療機関に直接訪問し血液製剤を供給 (巡回供給)
- 燃料の不足 (ガソリン、軽油)
・ 緊急車両の燃料補給にも苦慮
・ 公共交通機関の不通、一般車両の燃料補給が困難なことから職員の通勤に支障 (基幹センターである宮城県センター供給業務を中心として、全国の血液センターから緊急車両 8 台、延べ100人の支援)
- 停電により自家発電装置のない供給施設等の血液製剤等は母体の血液センターへ搬送
- 被災地域での献血受入は困難

O-IRPの導入

IRP

輸送と在庫管理という通常は個別に解決される2つの領域を統合することを目指す問題。

「輸送コストと在庫コストの合計」を最小化する。



いつ、どの顧客に、どれだけの量を、どういうルートで届けるか

VMI

ベンダー側がバイヤーと事前に適切な在庫レベルや在庫ポリシーを取り決め、バイヤーに代わり主導して在庫を補給すること。

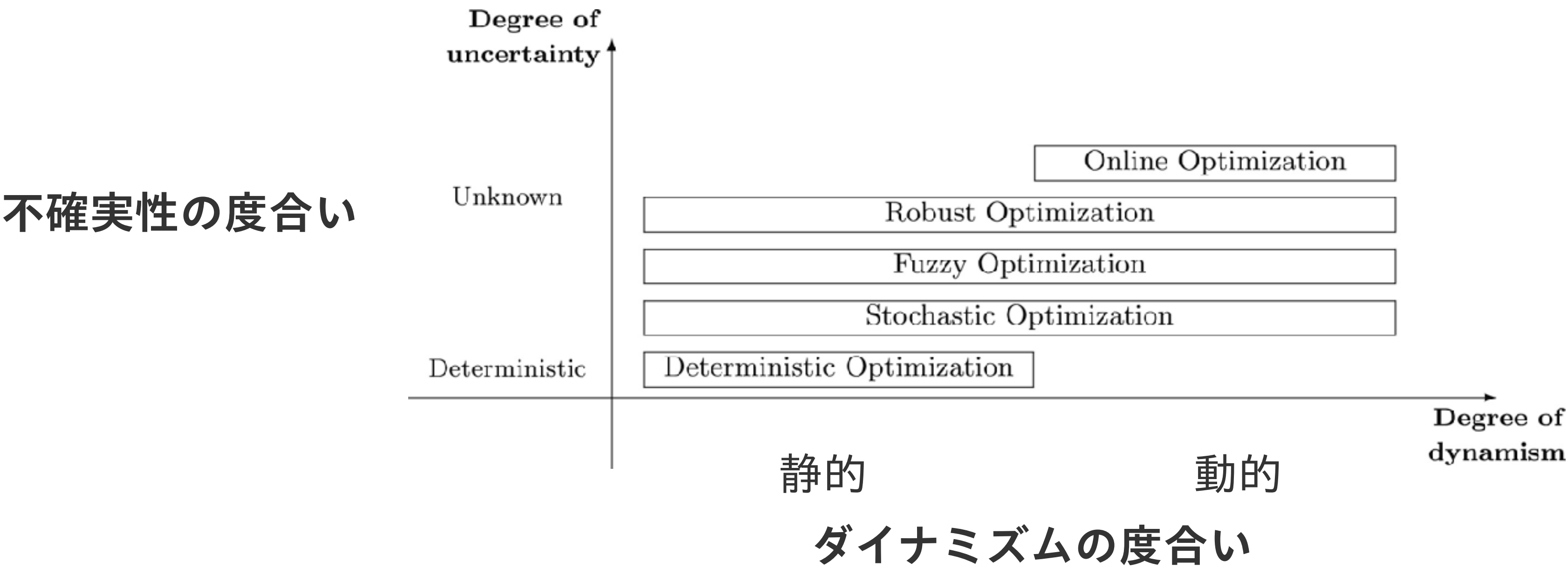
Eコマースの文脈で、時間的に明らかになる需要に対して順次対応することを考えたい → 「オンライン」

Online-Inventory Routing Problem

👉 複数車両により在庫ルーティング問題のオンライン版 (O-IRP)

2. Literature Reviews

組合せ最適化問題の分類



- 決定論的最適化 … 入力に関する完全な知識を持つという静的な条件で最適解を求める。
- 確率的最適化 … 確率分布が既知であるか、または正確に推定できる条件下で最適解を求める。
- ファジィ最適化 … 人間の「あいまいさ」を数理的に扱い、最適解を求める。
- ロバスト最適化 … 将来の変動に備え、どんな状況でも性能が落ちにくい解を求める。
- **オンライン最適化** … 逐次的な入力に対して逐次的に意思決定する状況下で最適解を求める。

オンライン最適化問題の関連研究

Sleator and Tarjan (1985)

オフライン問題の厳密解法との比較を行った。

Karlin et al. (1988)

競合比解析の概念を提案した。

競合比

オンラインアルゴリズムの最悪時の性能を、

同じ問題を完全な情報で解く最適オフラインアルゴリズムと比較した比率。

任意の問題のインスタンス I に対して、 $\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq c$ を満たすような c を**競合比**と言う。

$\text{ALG}(I)$... オンラインアルゴリズム ALG が与える解のコスト

$\text{OPT}(I)$... 入力を全て事前に知っていた場合、最適オフラインアルゴリズム OPT が与える解のコスト

IRPの背景と重要性

IRP

輸送と在庫管理という通常は個別に解決される2つの領域を統合することを目指す問題。
「輸送コストと在庫コストの合計」を最小化する。

Bell et al. (1983) … 最初のIRPの研究。

→ 液化ガスの在庫管理と車両のルーティング・スケジューリングを統合した。

在庫管理やルーティング単体の問題は古くから多くあるが、それを統合したIRPの研究は新しい。

- **海運輸送** (Christiansen et al., 2011) : セメント製品の沿岸配送。
- **小規模店舗向けVMI** (Bertazzi et al., 2019) : P&Gによる東南アジア都市の小規模店舗補充。
- **軍事補給作戦** (McKenna et al., 2020) : 無人航空機を用いた米軍の補給。
- **スーパーマーケット** (Gaur and Fisher, 2004 ; Laganà et al., 2016) 。
- **ガソリンスタンドの燃料配送** (Popović et al., 2012) 。
- **災害支援物資の配送** (Çankaya et al., 2019) 。
- **廃油の回収** (Cárdenas-Barrón and Melo, 2021) 。

IRPの関連研究 | 決定論的なIRP

入力が予め知られている静的な状況下での問題。

Archetti et al. (2007) : 混合整数計画問題と分枝限定法による最適解。

Solyalı and Süral (2011) : order-up-to levelポリシーに基づく定式化とヒューリスティック。

Skålnes et al. (2023) : マスヒューリスティックと分枝限定法の統合。

IRPの関連研究 | 不確実性を考慮したIRP

- **ファジーIRP (Niakan and Rahimi, 2015)**
 - 医薬品の配送問題。
 - 需要・輸送・不足コストをファジィ数で表現。
 - 多目的確率線形計画と対話型ファジィ解法を提案。
- **ロバストIRP (Solyalı et al., 2012)**
 - 多期間IRPに対して区間需要を仮定。
 - ロバスト混合整数計画によるモデル化と解法。

IRPの関連研究 | 不確実性を考慮したIRP

確率的IRP (Stochastic IRP)

- Hvattum et al. (2009) :
 - 離散分布に基づく確率的需要。
 - マルコフ決定過程 (MDP) として定式化。
- Huang and Lin (2010) :
 - 自動販売機の補充問題。
 - アントコロニー最適化を用いた手法。
- Markov et al. (2018) :
 - 予測誤差を考慮した需要予測モデル。
 - 現実のデータに基づく適用可能性が高い。
- Yu et al. (2012) :
 - スプリットデリバリーと在庫レベル・倉庫レベルのサービス水準を考慮。
 - 非線形制約の線形化やラグランジュ緩和を用いたハイブリッド解法を提案。
- Alvarez et al. (2021) :
 - 商品供給と需要がともに不確実。
 - 2段階確率計画 + プログレッシブヘッジアルゴリズムによるヒューリスティック解法。

IRPの関連研究 | その他

- Bertazzi et al. (2013) :
 - order-up-to level ポリシー + 需要の過不足にペナルティコストを課す。
 - ロールアウトアルゴリズムを提案。
- Coelho et al. (2014) :
 - 反応的ポリシー vs. 予測型ポリシーの比較。
 - 確率情報の利用は計算時間とトレードオフ。
- Bertazzi et al. (2015) :
 - 生産制約下の確率的需要 + 外部輸送。
 - 平均需要を用いた単純化が非現実的な結果を招く。
 - スモールサイズでは動的計画法、大規模にはマスヒューリスティックス。
- Bertazzi et al. (2016) :
 - 外部委託車両 + 確率的需要。
 - 解法の詳細は途中で省略されているが、実装可能性が高いアプローチ。

3. Problem description

表記

時間

- 計画対象期間集合 $\mathcal{T} = \{1, 2, \dots, T\}$
- 各 $t \in \mathcal{T}$ に対し、
$$\mathcal{T}_t(\gamma) = \{t, t + 1, \dots, \min[t + \gamma, T]\} \subseteq \mathcal{T}$$

ただし、 γ は 0 と $T - 1$ の間の整数。また τ は $\mathcal{T}_t(\gamma)$ の終わり。

各頂点 $i \in V$

- 単位量あたりの在庫コスト h_i
- 最大在庫量 U_i
- 初期在庫量 $I_i^0 \leq U_i$
- 在庫量 I_i^τ
- 点 i が時間 τ に車両 $k \in \mathcal{K}$ により訪問されるか $y_i^{k\tau}$
- エッジ (i, j) を車両 $k \in \mathcal{K}$ が通った回数
- 点 i に車両 $k \in \mathcal{K}$ が配送した輸送量

車 $\mathcal{K} = \{1, \dots, K\}$

- 同質
- 輸送容量 Q

各辺 $(i, j) \in E$

- 輸送コスト c_{ij}

表記 | 時間に関連する事柄

時間

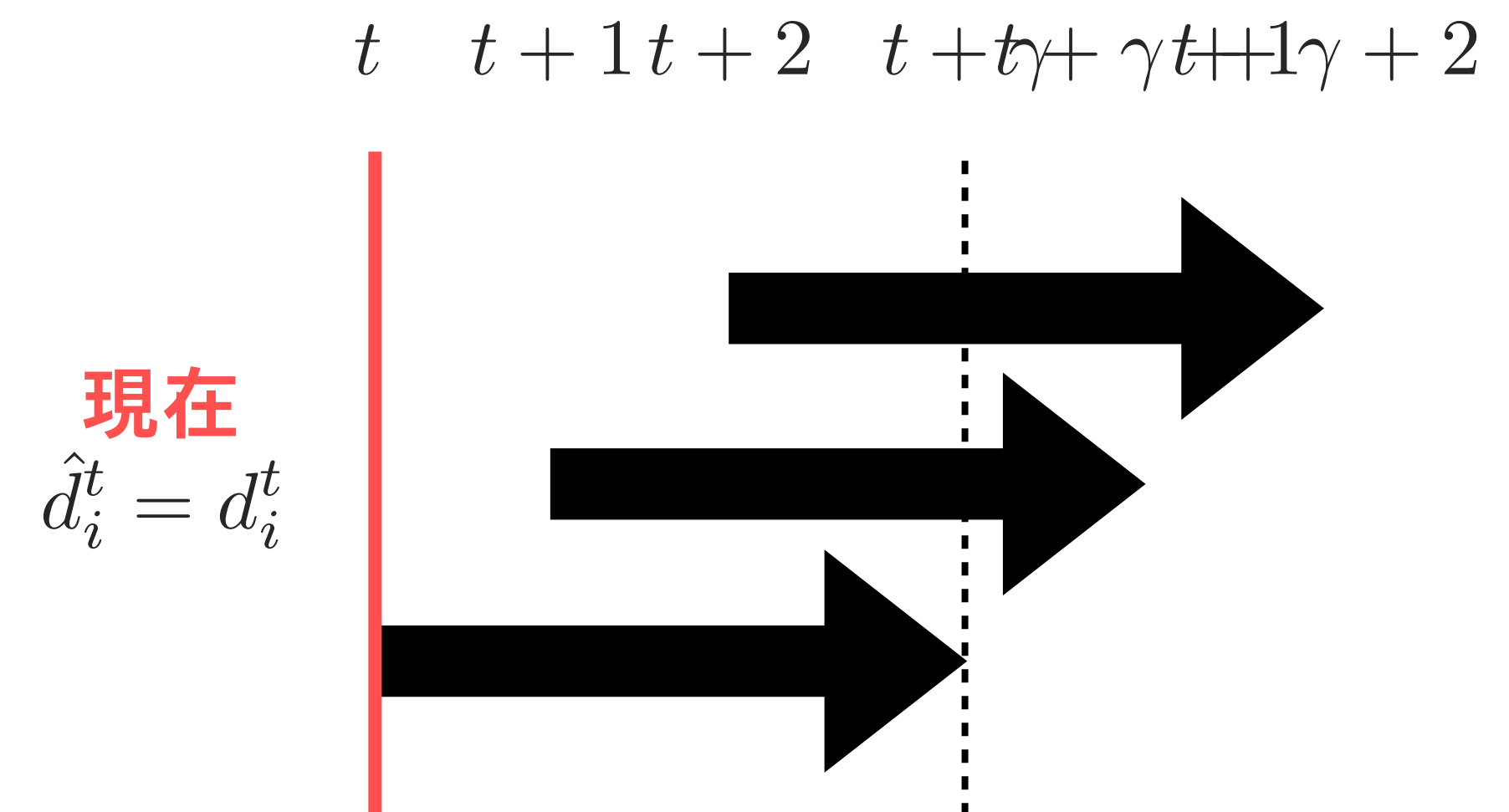
- 計画対象期間集合 $\mathcal{T} = \{1, 2, \dots, T\}$
- 各 $t \in \mathcal{T}$ に対し、
 $\mathcal{T}_t(\gamma) = \{t, t+1, \dots, \min[t+\gamma, T]\} \subseteq \mathcal{T}$

ただし、 γ は 0 と $T-1$ の間の整数。また τ は $\mathcal{T}_t(\gamma)$ の終わり。

どれだけ先の情報まで考慮して逐次行動を決定するか ($\gamma=0$ なら現在のみを考慮)

$\tau \in \mathcal{T}$ において、

- サプライヤーの保有在庫 r_τ
- 各顧客の需要 d_i^τ (不完全情報 $\hat{d}_i^\tau$)

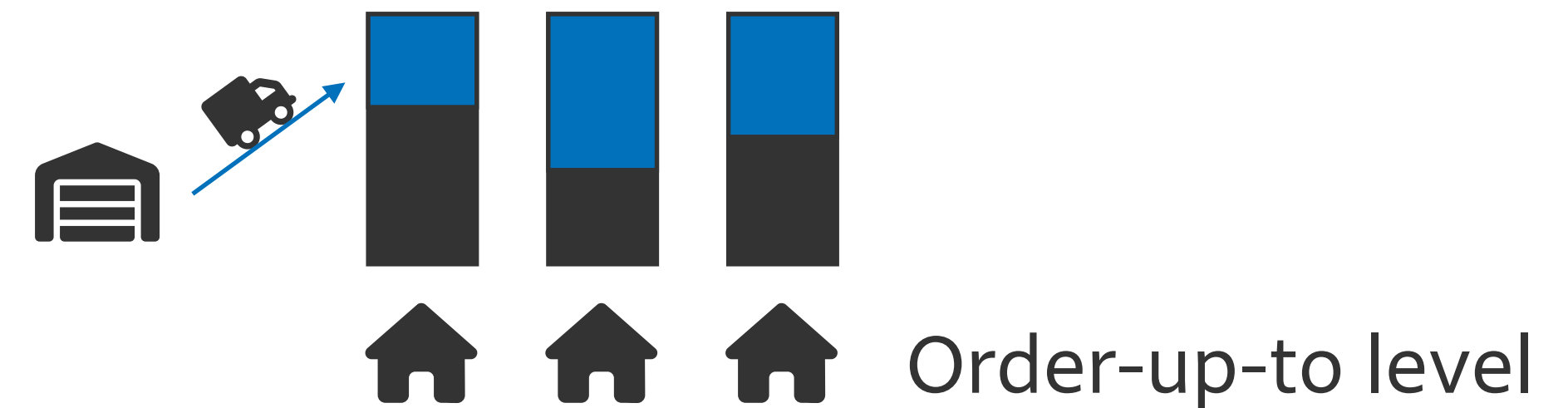


現在の γ 先までの情報がわかっている。

問題設定

時間 τ において顧客 i に配送されるべき総量

$$\hat{D}_i^t = \max \left\{ \sum_{\tau \in \mathcal{T}_t(\gamma)} \hat{d}_i^\tau - I_i^{t-1}, 0 \right\}$$



代表的なIRPのポリシー (Bertazzi and Speranza, 2012)

- **Order-up-to level** … 顧客に配送するたびに顧客の在庫が最大在庫量に一致するようにする。
- **Maximum level** … 顧客に配送するたびに顧客の在庫が最大在庫量を上回らないようにする。👉採用

• サプライヤー・顧客における在庫コスト

• ルーティングコスト

の和を最小化するような配送量とルートを決めたい。



4. Online algorithms

アルゴリズムの概要

$\text{ALG}(\gamma)$ というクラスのオンラインアルゴリズムを提案する。

👉 各 $t \in \mathcal{T}$ の始まりにおいて、混合整数計画問題 $\text{O-IRP}_t(\gamma)$ を解く $\rightarrow$ その時点における計算結果を保存。

Algorithm 1: Class of online algorithms $\text{ALG}^{(\gamma)}$

```
1 foreach  $t \in \mathcal{T}$  do  
2   |   Solve model  $\text{O-IRP}_t(\gamma)$   
3   |   Store the values of the variables at time  $t$   
4 end  
5 Return the corresponding value of the objective function
```

オンライン問題ならではの不確実性を考慮するため、 $\gamma = 0, 1, 2$ の場合のみを考えることにする。

$\gamma = 0$

定式化

$$\min \sum_{i \in V} \sum_{\tau \in \mathcal{T}_t(\gamma)} h_i I_i^\tau + \sum_{(i,j) \in E} \sum_{k \in \mathcal{K}} \sum_{\tau \in \mathcal{T}_t(\gamma)} c_{ij} x_{ij}^{k\tau}$$

在庫コスト ルーティングコスト

s.t.

$$I_{0\tau} = I_0^{\tau-1} + r_\tau - \sum_{i \in V'} \sum_{k \in \mathcal{K}} q_i^{k\tau}, \quad \forall \tau \in \mathcal{T}_t(\gamma)$$

サプライヤーにおける在庫量保存

$$I_i^\tau = I_i^{\tau-1} + \sum_{k \in \mathcal{K}} q_i^{k\tau} - \hat{d}_i^\tau, \quad \forall i \in V', \forall \tau \in \mathcal{T}_t(\gamma)$$

顧客における在庫量保存

$$\sum_{k \in \mathcal{K}} q_i^{k\tau} \leq U_i - I_i^{\tau-1}, \quad \forall i \in V', \forall \tau \in \mathcal{T}_t(\gamma)$$

顧客に配送して良い量の上限

$$q_i^{k\tau} \leq U_i y_i^{k\tau}, \quad \forall i \in V', \forall k \in \mathcal{K}, \forall \tau \in \mathcal{T}_t(\gamma)$$

車両が訪問しない点には一切配送しない

$$\sum_{\tau \in \mathcal{T}_t(\gamma)} \sum_{k \in \mathcal{K}} q_i^{k\tau} = \hat{D}_i^t, \quad \forall i \in V'$$

配送量は顧客の総需要を充足する

$$\sum_{i \in V'} q_i^{k\tau} \leq Q y_0^{k\tau}, \quad \forall k \in \mathcal{K}, \forall \tau \in \mathcal{T}_t(\gamma)$$

配送量は車両容量を超えない

$$\sum_{j \in V, i < j} x_{ij}^{k\tau} + \sum_{j \in V, j < i} x_{ji}^{k\tau} = 2y_i^{k\tau}, \quad \forall i \in V, \forall k \in \mathcal{K}, \forall \tau \in \mathcal{T}_t(\gamma)$$

次数制約 (完全グラフ)

定式化 (続き)

$$\sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{T}, i < j} x_{ij}^{k\tau} \leq \sum_{i \in \mathcal{T}} y_i^{k\tau} - y_m^{k\tau}, \quad \forall \mathcal{T} \subseteq V', \forall k \in \mathcal{K}, \forall \tau \in \mathcal{T}_t(\gamma), \forall m \in \mathcal{T} \quad \text{部分巡回路を排除}$$

$$\sum_{k \in \mathcal{K}} y_i^{k\tau} \leq 1, \quad \forall i \in V', \forall \tau \in \mathcal{K}_t(\gamma) \quad \text{各時間に同じ点を訪問する車両は多くて1台}$$

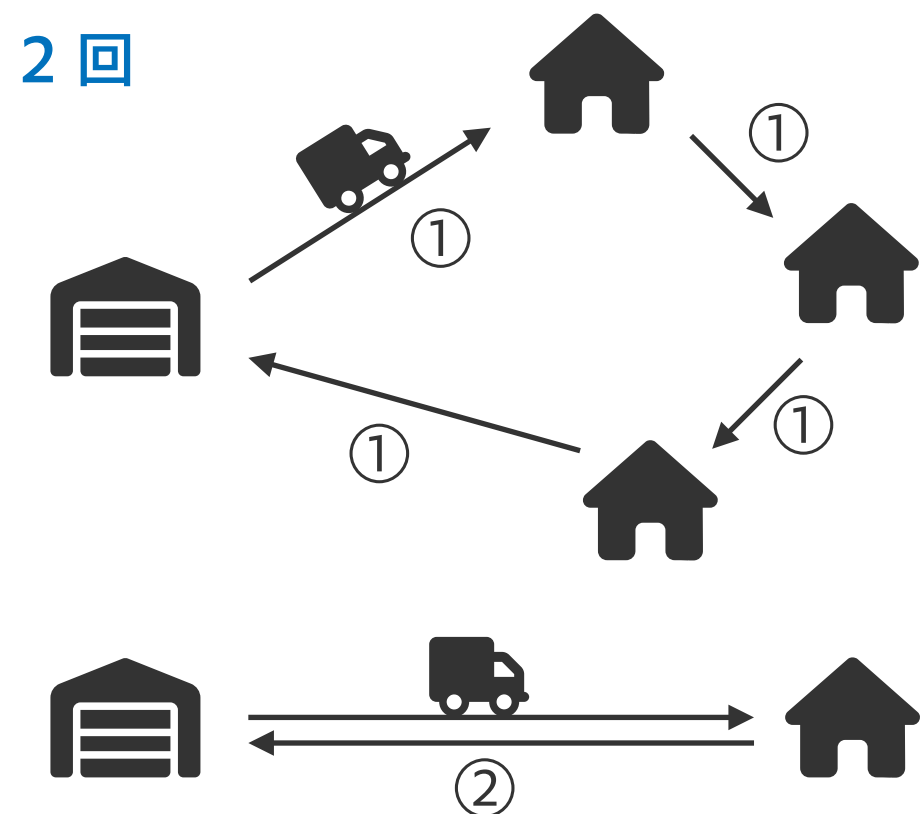
$$I_i^\tau \in \mathbb{R}^+, \quad \forall i \in V, \forall \tau \in \mathcal{T}_t(\gamma) \quad \text{在庫量は非負}$$

$$q_i^{k\tau} \in \mathbb{R}^+, \quad \forall i \in V', \forall \tau \in \mathcal{T}_t(\gamma) \quad \text{配送量は非負}$$

$$x_{ij}^{k\tau} \in \{0, 1, 2\}, \quad \forall j \in V', \forall k \in \mathcal{K}, \forall \tau \in \mathcal{T}_t(\gamma) \quad \text{サプライヤーと顧客間の辺を通るのは 0, 1, 2 回}$$

$$x_{ij}^{k\tau} \in \{0, 1\}, \quad \forall i \in V', \forall j \in V', \forall k \in \mathcal{K}, \forall \tau \in \mathcal{T}_t(\gamma) \quad \text{顧客間の辺を通るのは 0, 1 回}$$

$$y_i^{k\tau} \in \{0, 1\}, \quad \forall i \in V, \forall k \in \mathcal{K}, \forall \tau \in \mathcal{T}_t(\gamma) \quad \text{ある点を訪問するかどうか (バイナリ変数である)}$$



5. Competitive Analysis of $ALG^{(\gamma)}$

競合比解析とは

競合比

オンラインアルゴリズムの最悪時の性能を、同じ問題を完全な情報で解く最適オフラインアルゴリズムと比較した比率。

$$c = \frac{\text{ALG}(I)}{\text{OPT}(I)}$$

理論的な性能保証を与える。

Split-delivery vehicle routing problem

↓ 今回はSDVRPを解く

オンラインアルゴリズムのワーストケースにおける解をオフライン問題の最適解と比較することで、オンラインアルゴリズムの性能の下限（コストの上限）を理論的に示すことができる。

競合比解析 | 定理1

$$\frac{\text{ALG}^{(\gamma)}}{\text{OPT}} \leq \max \left\{ \frac{h_{\max}}{h_{\min}}, T \right\}$$

↑ 在庫を保持する場所によるコストのばらつき

証明（の雰囲気）

全期間の総在庫量を B とすると、これは意思決定変数に依存しない。

オフライン最適解の在庫コスト、オンライン解の在庫コストはともに $h_{\min} B$ 以上で $h_{\max} B$ 以下。

顧客 i に届ける総量を D_i とすると、

- オフライン最適解の配送コスト $\geq$ SDVRP の最適解
- オンライン解の配送コストは T の各期間に同じベースルートを使って部分配送することにより SDVRP の最適解の高々 T 倍以下になる。

$$\text{よって、} \frac{\text{ALG}(\gamma)}{\text{OPT}} \leq \frac{h_{\max} B + T z_{\text{SDVRP}}(D)}{h_{\min} B + z_{\text{SDVRP}}(D)} \leq \max \left\{ \frac{h_{\max}}{h_{\min}}, T \right\}$$

$z_{\text{SDVRP}}(D) \cdots$ SDVRP の最適解

競合比解析の解釈

- 競合比は1以上の範囲で、小さければ小さいほど良い。
 - 競合比解析はワーストケースにおける性能を示すもの。
- ただし、現実的な適用においてはワーストケースに陥ることはあまりなく、経験的には競合比が定理1から導かれる値よりもずっと小さくなる。

6. Solving the O – IRP_t(γ)

Algorithm 1: Class of online algorithms ALG^(γ)

```
1 foreach  $t \in \mathcal{T}$  do  
2   | Solve model O-IRPt( $\gamma$ )     
3   | Store the values of the variables at time  $t$   
4 end  
5 Return the corresponding value of the objective function
```

O-IRPの解法

2通りの手法が用いられている。

1. Branch-and-cut アルゴリズム

- （本研究の核ではない。）
- 問題のサイズが大きくなると性能が落ちる。

2. マスヒューリスティックアルゴリズム

- 解の質を高く保ちながら計算時間を短縮できる。

※ 2つを併用しているわけではないことに注意。

① Branch-and-cut アルゴリズム

Branch-and-cut アルゴリズム

1. 線形緩和した問題を単体法によって解く。
2. 得られた最適解が整数条件を満たしていなければ切除平面法により**線形不等式制約を追加する**。
3. 問題を複数の部分問題に分割し、それぞれにおいて線形緩和問題を解く。

Archetti et al. (2007), Coelho et al. (2012)

$x_{0i}^{k\tau} \leq 2y_i^{k\tau}, \forall i \in V', \forall k \in \mathcal{K}, \forall \tau \in \mathcal{T}_t(\gamma)$ サプライヤー→顧客 i (往路) または顧客 $i \rightarrow$ サプライヤー (復路) で辺 $(0, i)$ を使っているなら、必ず顧客 i を訪問したとみなす。

$x_{ij}^{k\tau} \leq y_i^{k\tau}, \forall i, j \in V', \forall k \in \mathcal{K}, \forall \tau \in \mathcal{T}_t(\gamma)$ 顧客間の辺 (i, j) を通過するなら、必ず終点である顧客 j を訪問したとみなす。

$y_i^{k\tau} \leq y_0^{k\tau}, \forall i \in V', \forall k \in \mathcal{K}, \tau \in \mathcal{T}_t(\gamma)$ 車両 k が期間 τ に顧客 i を訪問するならば、その車両 k は期間 τ にサプライヤーを含むルートを巡回しなければならない

① Branch-and-cut アルゴリズム

線形不等式制約 (続き)

Coelho and Laporte (2013) / 対称性に着目

$$y_0^{k\tau} \leq y_0^{(k-1)\tau}, \forall k \in \mathcal{K} \setminus 1, \forall \tau \in \mathcal{T}_t(\gamma)$$

車両は若い番号から使用する。

$$y_i^{k\tau} \leq \sum_{j < i} y_j^{(k-1)\tau}, \forall i \in V', \forall k \in \mathcal{K} \setminus 1, \forall \tau \in \mathcal{T}_t(\gamma)$$

インデックスが小さい顧客から順に車両に割り当てる。

Lefever (2018) / 在庫レベルと配送量の上限・下限に着目

$$I_i^\tau \geq I_i^{0\tau}, \forall i \in V', \forall \tau \in \mathcal{T}_t(\gamma)$$

残在庫量は期間中配送が行われない場合にの在庫量を下回らない。

$$q_i^{k\tau} \leq U_i - I_i^{0\tau}, \forall i \in V', \forall k \in \mathcal{K}, \forall \tau \in \mathcal{T}_t(\gamma)$$

配送量は最大在庫量から $\uparrow$ を差し引いた値を超えない。

② マスヒューリスティックアルゴリズム

O-IRPの高品質な解を高速に得るアルゴリズムとして、マスヒューリスティックを提案している。

1. 巨大なツアーの構築

- 需要が判明している全ての頂点に対してTSPを解く。

2. 顧客の割り当て

- ビンパッキング問題（与えられた「荷物（重さや個数がついている）」をつめる「箱（ビンやコンテナなど）」の最小数を見つけるもの）を解く。

3. 簡略化された O-IRP を解く

- 複雑な巡回制約を 1., 2. で得られた結果に基づく訪問順序制約に置き換えて問題を単純化して解く。

4. 解の改善

- 各時間について、3. で訪問した顧客のみを対象としてCVRPを解く。

複雑なO-IRP モデルを、TSP、BPP、CVRPといったより単純な問題に分解し、段階的に解く。

Algorithm 2: Matheuristic for the O-IRP_t(γ)

1 $C_I \leftarrow 0, C_R \leftarrow 0$

2 **foreach** $i \in \mathcal{T}$ **do**

3 $S_i \leftarrow LKH2(V_i^r)$

4 $P_i \leftarrow solveBPP(S_i)$

5 $C_I \leftarrow C_I + solveOIRP(i, P_i)$

6 **foreach** $\tau \in T_i(\gamma)$ **do**

7 $C_R \leftarrow C_R + solve2CFCVRP(\tau)$

8 **end**

9 **end**

10 **return** $C_I + C_R$

// initialize inventory and routing costs

// solve a TSP using the LKH-2 heuristic

// solve a BPP and obtain the set of tours P_i

// solve a simplified O-IRP_t(γ)

// solve a CVRP model (Baldacci et al., 2004) using the O-IRP solution

7. Computational results

計算実験の設定

第1群 (Archetti et al. 2007)

- 顧客数5~50、期間 $T=3, 6$
- 車両台数 $K=1, \dots, 5$
- 在庫コストクラス: Low Cost ($h_i \in [0.01, 0.05], h_0 = 0.03$), High Cost ($h_i \in [0.1, 0.5], h_0 = 0.3$)
- タイムリミット：2時間

第2群 (Skålnes et al. 2024)

- 顧客数：10, 25, 50, 100、期間 T ：最大12
- 実現可能性を保証するために Bin Packing による車両数調整後の実例的データ
- タイムリミット：6時間。

計算実験の設定

評価指標

- 上限値 (UB: Upper Bound)
- 下限値 (LB: Lower Bound)
- ギャップ $((UB - LB)/UB)$ … 最適性ギャップ。最適解を直接評価することは困難なこともある。
- 計算時間 (Time)
- 経験的競合比 (Z) … 競合比を実際に計算して得た結果に基づいて求める。

$$Z = \frac{UB \left(\text{ALG}^{(\gamma)} \right)}{LB \left(O - \text{IRP}_1(T - 1) \right)}$$

第1群 - Branch-and-cut ①

Table 1. Average computational results of each algorithm for the first group of instances with $T = 6$ and $O-IRP_t(\gamma)$ solved through branch-and-cut.

Algorithm	Low cost					High cost							
	UB	LB	Gap (%)	Time (s)	Z	UB	LB	Gap (%)	Time (s)	Z			
O-IRP ₁ ($T - 1$)	8081.13	7186.91	7.53	1	3390.56	–	14152.68	13305.72	4.19	1	3383.81	–	
ALG ⁽⁰⁾	11377.62	11368.32	0.06		323.08	1.58	2	17509.02	17501.05	0.03		328.68	1.32
ALG ⁽¹⁾	9776.50	9714.60	0.42		935.01	1.35		15822.78	15772.89	0.21		870.37	1.18
ALG ⁽²⁾	8749.48	8550.34	1.45		1741.24	1.20	3	14856.84	14643.82	0.93		1624.64	1.11
					4							4	

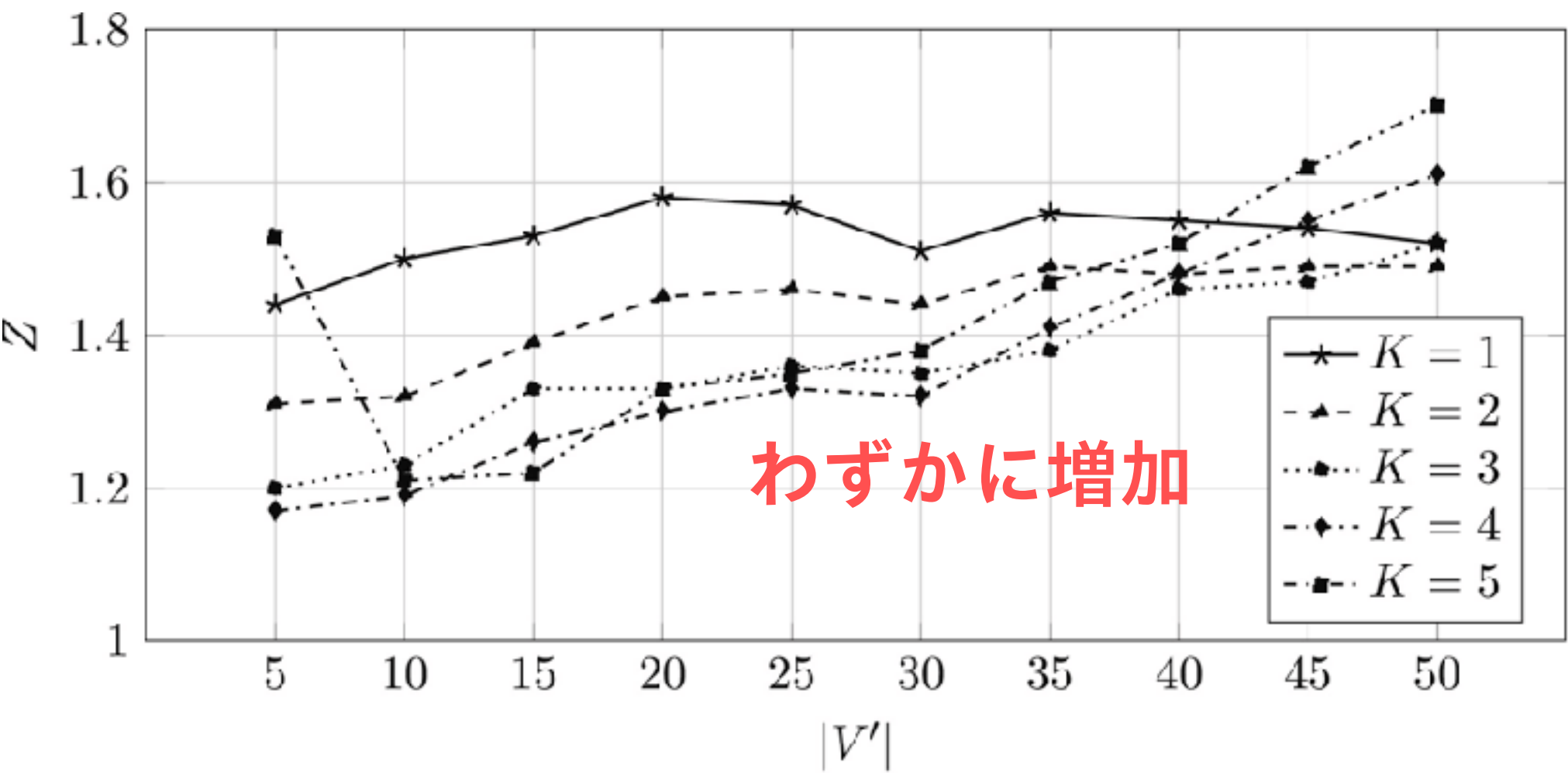
Table 2. Average computational results of each algorithm for the first group of instances with $T = 3$ and $O-IRP_t(\gamma)$ solved through branch-and-cut.

Algorithm	Low cost					High cost						
	UB	LB	Gap (%)	Time (s)	Z	UB	LB	Gap (%)	Time (s)	Z		
O-IRP ₁ ($T - 1$)	3739.80	3447.98	4.68	1	2171.88	–	8454.90	8151.08	2.24	1	2147.44	–
ALG ⁽⁰⁾	4932.78	4885.26	0.63		903.94	1.42	2	9645.76	9597.75	0.32	891.84	1.19
ALG ⁽¹⁾	4406.00	4218.66	2.20		1534.78	1.26	3	9065.70	8932.47	0.93	1530.24	1.10
					4						4	

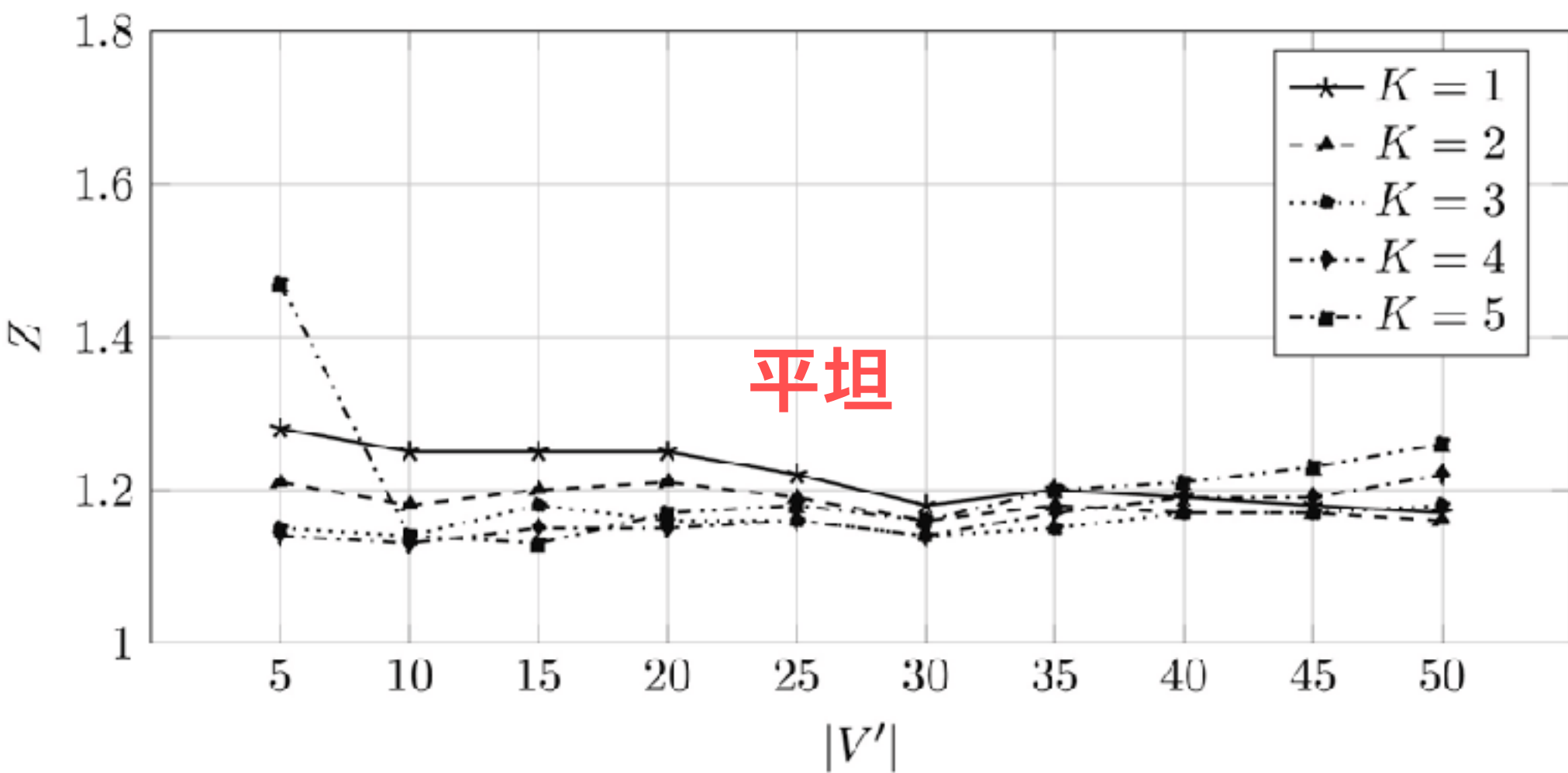
1. Branch-and-cut を用いても $T = 6$ のインスタンスでは制限時間内にオフライン問題を最適に解くことは難しい。 $T = 3$ のインスタンスでは Gap は小さくなるものの、時間制限内に解けないインスタンスも存在する。
2. ALG(0) は計算時間は大幅に短いが、経験的競合比は大幅に高くなる（解の品質が悪い）。
3. γ の値が増えるにつれて計算時間は増加するが、経験的競合比は低くなる。多くの情報を利用できると解の品質が上がる。
4. 理論上の競合比（ $T=6$ のとき 6, $T=3$ のとき 5）よりも経験的競合比ははるかに良い。

第1群 - Branch-and-cut ②

顧客数が ALG(0) の経験的競合比に与える影響を示す。



(a) Low Cost



(b) High Cost

→ 大規模になると IRP を解くのが難しくなるため。

第1群 - Branch-and-cut ③

車両数は経験的競合比に有意な影響を与えない。

Table 3. Average computational results obtained by combining instances with the same number of vehicles (instances of the first group with up to 30 customers for $T = 6$; O-IRP_t(γ) solved through branch-and-cut).

Algorithm	K	Low cost					High cost				
		UB	LB	Gap (%)	Time (s)	Z	UB	LB	Gap (%)	Time (s)	Z
ALG ⁽⁰⁾	1	9260.58	9260.58	0.00	1.92	1.63	15392.75	15392.75	0.00	1.67	1.32
	2	10207.48	10207.48	0.00	13.36	1.57	16339.65	16339.65	0.00	15.61	1.31
	3	11164.55	11164.55	0.00	78.71	1.53	17296.72	17296.72	0.00	81.86	1.29
	4	12354.68	12349.11	0.04	631.76	1.55	18486.85	18481.35	0.02	645.54	1.31
	5	13900.83	13859.90	0.25	889.65	1.59	20029.12	19994.78	0.13	898.70	1.34
ALG ⁽¹⁾	1	7991.09	7991.07	0.00	6.91	1.41	14045.20	14045.20	0.00	5.40	1.21
	2	8563.30	8563.30	0.00	74.48	1.32	14619.57	14619.57	0.00	84.18	1.17
	3	9361.47	9352.97	0.07	914.28	1.28	15439.79	15431.61	0.03	625.81	1.15
	4	10856.11	10775.35	0.60	1465.06	1.37	16838.20	16771.99	0.30	1489.60	1.19
	5	12110.53	11890.33	1.42	2214.31	1.39	18171.15	17996.07	0.72	2146.88	1.21
ALG ⁽²⁾	1	6585.30	6585.28	0.00	26.59	1.16	12662.08	12662.02	0.00	27.98	1.08
	2	7429.77	7424.37	0.06	739.96	1.14	13451.95	13451.87	0.00	594.59	1.07
	3	8512.85	8438.79	0.70	1684.67	1.17	14661.72	14570.51	0.42	1685.00	1.09
	4	9825.77	9514.17	2.42	2870.38	1.24	16009.77	15691.19	1.48	2702.25	1.13
	5	11393.69	10789.11	4.02	3384.58	1.31	17498.67	16843.48	2.75	3113.37	1.16

Cost	Algorithm	K	T = 3					T = 6				
			UB	LB	Gap (%)	Time (s)	Z	UB	LB	Gap (%)	Time (s)	Z
Low	ALG ^{(T-t),1}	1	3516.82	3516.30	0.01	442.83	1.24	6357.23	6357.23	0.00	380.91	1.12
		2	3867.66	3865.84	0.03	628.70	1.22	7311.01	7311.01	0.00	658.32	1.12
		3	4301.41	4298.97	0.04	1509.31	1.22	8504.41	8504.41	0.00	1458.51	1.17
		4	4747.76	4742.64	0.07	2240.00	1.23	9783.74	9783.71	0.00	2001.09	1.23
		5	5135.91	5127.65	0.08	2454.28	1.25	11302.49	11302.49	0.00	2586.14	1.30
	ALG ^{(T-t),2}	1	4283.79	4283.79	0.00	109.88	1.52	9214.09	9214.06	0.00	3.35	1.62
		2	4481.62	4480.42	0.02	256.23	1.41	10119.23	10119.19	0.00	16.72	1.55
		3	4794.27	4787.45	0.10	559.52	1.36	11114.59	11114.55	0.00	228.35	1.53
		4	5238.88	5166.70	0.99	1673.90	1.36	12336.87	12333.40	0.02	772.91	1.55
		5	5705.21	5543.35	2.04	2276.01	1.39	13897.66	13871.83	0.15	1069.56	1.59
High	ALG ^{(T-t),1}	1	8282.07	8278.77	0.02	510.66	1.10	12442.44	12442.44	0.00	445.17	1.05
		2	8510.27	8506.59	0.03	672.81	1.08	13338.88	13338.85	0.00	810.39	1.05
		3	8989.88	8987.28	0.02	1495.07	1.09	14614.97	14614.90	0.00	1360.15	1.09
		4	9400.22	9397.56	0.02	2111.19	1.09	15966.51	15966.51	0.00	2124.73	1.13
		5	9887.66	9882.10	0.05	2422.12	1.11	17491.03	17491.03	0.00	2595.25	1.16
	ALG ^{(T-t),2}	1	8991.07	8991.05	0.00	109.01	1.21	15345.74	15345.70	0.00	3.63	1.32
		2	9187.94	9187.94	0.00	213.76	1.17	16250.87	16250.84	0.00	19.37	1.30
		3	9503.10	9493.72	0.06	640.72	1.16	17245.94	17245.91	0.00	200.11	1.29
		4	9945.72	9873.48	0.48	1674.86	1.16	18461.34	18453.97	0.03	793.90	1.31
		5	10403.75	10245.39	0.90	2225.40	1.17	20045.11	20014.67	0.12	977.40	1.34

第1群 - Branch-and-cut ④

全ての結果において、High Cost（在庫コストが高い）の方が Low Cost よりも経験的競合比が低い。

→ 在庫コストが高いと不要な配送が抑制され、オフラインの場合に近くなる。

Cost	Algorithm	K	T = 3					T = 6				
			UB	LB	Gap (%)	Time (s)	Z	UB	LB	Gap (%)	Time (s)	Z
Low	ALG ^{(T-t),1}	1	3516.82	3516.30	0.01	442.83	1.24	6357.23	6357.23	0.00	380.91	1.12
		2	3867.66	3865.84	0.03	628.70	1.22	7311.01	7311.01	0.00	658.32	1.12
		3	4301.41	4298.97	0.04	1509.31	1.22	8504.41	8504.41	0.00	1458.51	1.17
		4	4747.76	4742.64	0.07	2240.00	1.23	9783.74	9783.71	0.00	2001.09	1.23
		5	5135.91	5127.65	0.08	2454.28	1.25	11302.49	11302.49	0.00	2586.14	1.30
	ALG ^{(T-t),2}	1	4283.79	4283.79	0.00	109.88	1.52	9214.09	9214.06	0.00	3.35	1.62
		2	4481.62	4480.42	0.02	256.23	1.41	10119.23	10119.19	0.00	16.72	1.55
		3	4794.27	4787.45	0.10	559.52	1.36	11114.59	11114.55	0.00	228.35	1.53
		4	5238.88	5166.70	0.99	1673.90	1.36	12336.87	12333.40	0.02	772.91	1.55
		5	5705.21	5543.35	2.04	2276.01	1.39	13897.66	13871.83	0.15	1069.56	1.59
High	ALG ^{(T-t),1}	1	8282.07	8278.77	0.02	510.66	1.10	12442.44	12442.44	0.00	445.17	1.05
		2	8510.27	8506.59	0.03	672.81	1.08	13338.88	13338.85	0.00	810.39	1.05
		3	8989.88	8987.28	0.02	1495.07	1.09	14614.97	14614.90	0.00	1360.15	1.09
		4	9400.22	9397.56	0.02	2111.19	1.09	15966.51	15966.51	0.00	2124.73	1.13
		5	9887.66	9882.10	0.05	2422.12	1.11	17491.03	17491.03	0.00	2595.25	1.16
	ALG ^{(T-t),2}	1	8991.07	8991.05	0.00	109.01	1.21	15345.74	15345.70	0.00	3.63	1.32
		2	9187.94	9187.94	0.00	213.76	1.17	16250.87	16250.84	0.00	19.37	1.30
		3	9503.10	9493.72	0.06	640.72	1.16	17245.94	17245.91	0.00	200.11	1.29
		4	9945.72	9873.48	0.48	1674.86	1.16	18461.34	18453.97	0.03	793.90	1.31
		5	10403.75	10245.39	0.90	2225.40	1.17	20045.11	20014.67	0.12	977.40	1.34

第1群 - Branch-and-cut | 不現実性のシナリオ

これまでの実験は「ローリングホライゾン方式」

複数日におよぶ配送計画の内、初日のみを採用し、翌日以降の計算に関しては、計算期間を1日スライドして、同様に計画の作成と初日の配送計画の採用を繰り返すというもの。https://www.azbil.com/jp/corporate/pr/library/review/2020/tech202004_05.html



一般化し、将来の需要の一部だけが徐々に知られるようなケースを考える。

- シナリオ1 … **ランダム性あり**

- 現在の期間 (t) : 全ての顧客 (100%) の需要が完全に知られている。
- 将来の期間 ($t' > t$) : 顧客の割合に基づいて需要情報がわかる顧客が減っていく。

(例 : $t+1$ で 50%, $t+2$ で 25%, $t+3$ 以降で 10%)

- シナリオ2 … **ランダム性なし**

- 現在の期間 (t) : 全ての顧客 (100%) の需要が完全に知られている。
- 将来の期間 ($t' > t$) : 平均需要より大きい需要を持つ全ての顧客 (100%) の需要が完全にわかる。

第1群 - Branch-and-cut | 不現実性のシナリオ

Table 5. Average computational results obtained for the additional scenarios by combining all instances with the same number of vehicles (instances of the first group with up to 50 customers for $T = 3$ and 30 customers for $T = 6$).

Cost	Algorithm	K	T = 3					T = 6				
			UB	LB	Gap (%)	Time (s)	Z	UB	LB	Gap (%)	Time (s)	Z
Low	ALG ^{(T-t),1}	1	3516.82	3516.30	0.01	442.83	1.24	6357.23	6357.23	0.00	380.91	1.12
		2	3867.66	3865.84	0.03	628.70	1.22	7311.01	7311.01	0.00	658.32	1.12
		3	4301.41	4298.97	0.04	1509.31	1.22	8504.41	8504.41	0.00	1458.51	1.17
		4	4747.76	4742.64	0.07	2240.00	1.23	9783.74	9783.71	0.00	2001.09	1.23
		5	5135.91	5127.65	0.08	2454.28	1.25	11302.49	11302.49	0.00	2586.14	1.30
	ALG ^{(T-t),2}	1	4283.79	4283.79	0.00	109.88	1.52	9214.09	9214.06	0.00	3.35	1.62
		2	4481.62	4480.42	0.02	256.23	1.41	10119.23	10119.19	0.00	16.72	1.55
		3	4794.27	4787.45	0.10	559.52	1.36	11114.59	11114.55	0.00	228.35	1.53
		4	5238.88	5166.70	0.99	1673.90	1.36	12336.87	12333.40	0.02	772.91	1.55
		5	5705.21	5543.35	2.04	2276.01	1.39	13897.66	13871.83	0.15	1069.56	1.59
High	ALG ^{(T-t),1}	1	8282.07	8278.77	0.02	510.66	1.10	12442.44	12442.44	0.00	445.17	1.05
		2	8510.27	8506.59	0.03	672.81	1.08	13338.88	13338.85	0.00	810.39	1.05
		3	8989.88	8987.28	0.02	1495.07	1.09	14614.97	14614.90	0.00	1360.15	1.09
		4	9400.22	9397.56	0.02	2111.19	1.09	15966.51	15966.51	0.00	2124.73	1.13
		5	9887.66	9882.10	0.05	2422.12	1.11	17491.03	17491.03	0.00	2595.25	1.16
	ALG ^{(T-t),2}	1	8991.07	8991.05	0.00	109.01	1.21	15345.74	15345.70	0.00	3.63	1.32
		2	9187.94	9187.94	0.00	213.76	1.17	16250.87	16250.84	0.00	19.37	1.30
		3	9503.10	9493.72	0.06	640.72	1.16	17245.94	17245.91	0.00	200.11	1.29
		4	9945.72	9873.48	0.48	1674.86	1.16	18461.34	18453.97	0.03	793.90	1.31
		5	10403.75	10245.39	0.90	2225.40	1.17	20045.11	20014.67	0.12	977.40	1.34

- オンラインアルゴリズムは理論値よりも低い経験的競合比を示す。
- 経験的競合比は High Cost の方が Low Cost よりも低い。

第1群 - Branch-and-cut vs. マスヒューリスティック

Table 6. Branch-and-cut vs. matheuristic: average computational results of each algorithm for the first group of instances.

Cost	T	Algorithm	Branch-and-cut				Matheuristic		
			UB	LB	Time (s)	Z	UB	Time (s)	Z
Low	3	O-IRP ₁ ($T - 1$)	3739.80	3447.98	2171.88	–	3646.49	52.05	–
		ALG ⁽⁰⁾	4932.78	4885.26	903.94	1.42	4920.22	18.82	1.42
		ALG ⁽¹⁾	4406.00	4218.66	1534.78	1.26	4366.32	23.64	1.26
	6	O-IRP ₁ ($T - 1$)	8081.13	7186.91	3390.56	–	7932.11	2329.95	–
		ALG ⁽⁰⁾	11377.62	11368.32	323.08	1.58	11375.30	15.66	1.58
		ALG ⁽¹⁾	9776.50	9714.60	935.01	1.35	9766.64	19.46	1.35
High	3	O-IRP ₁ ($T - 1$)	8454.90	8151.08	2147.44	–	8358.40	50.11	–
		ALG ⁽⁰⁾	9645.76	9597.75	891.84	1.19	9631.52	19.03	1.19
		ALG ⁽¹⁾	9065.70	8932.47	1530.24	1.10	9054.32	24.17	1.11
	6	O-IRP ₁ ($T - 1$)	14152.68	13305.72	3383.81	–	14036.42	2240.60	–
		ALG ⁽⁰⁾	17509.02	17501.05	328.68	1.32	17488.77	16.17	1.31
		ALG ⁽¹⁾	15822.78	15772.89	870.37	1.18	15879.71	20.83	1.19
		ALG ⁽²⁾	14856.84	14643.82	1624.64	1.11	14867.25	191.78	1.11
					2	1			

- 1. 2つの手法で解の品質に差異はない。
 - 2. マスヒューリスティックは計算時間を大幅に短縮する。
- マスヒューリスティックはBranch-and-cutと同等の解の質を維持しつつ、計算時間を大幅に短縮する有効な手法である。

第2群 - マスヒューリスティック

Table 7. Average computational results obtained by combining instances of the second group with the same number of customers.

Algorithm	$ V' $	UB_{MH}	LB_{BC}	Z	#	Time (s)	
						MH	BC
O-IRP ₁ ($T - 1$)	10	5236.25	4746.80	–	54	17581.65	18433.25
	25	13426.23	10756.38	–	47	21608.34	21622.11
	50	29550.13	20604.45	–	53	21632.44	21603.70
	100	51080.38	29755.32	–	49	22012.56	21600.41
ALG ⁽⁰⁾	10	6855.04	–	1.46	54	5.92	–
	25	16242.75	–	1.67	47	220.63	–
	50	33977.74	–	1.88	53	15076.28	–
	100	55235.01	–	2.17	49	18732.49	–
ALG ⁽¹⁾	10	5845.94	–	1.24	54	9.26	–
	25	14186.46	–	1.46	47	16196.19	–
	50	30599.06	–	1.71	53	21518.36	–
	100	52201.93	–	2.08	49	21595.53	–
ALG ⁽²⁾	10	5564.19	–	1.17	54	372.57	–
	25	13876.47	–	1.42	47	20689.57	–
	50	30089.35	–	1.67	53	21454.91	–
	100	52280.05	–	2.08	49	21672.96	–

↓ 制限時間内に求解した数

- Branch-and-cut が実行可能解を制限時間内に見つけれない問題でも求解可能。
- 大規模インスタンスに対しても比較的低い経験的競合比を示す。

計算実験のまとめ

- マスヒューリスティックを使ってO-IRPを解くアプローチが、現実的な規模のO-IRPを解くのに非常に有効である。
- マスヒューリスティックは解の質を維持しながら計算時間を大幅に削減できる。
- 提案したオンラインアルゴリズムは理論的に示されたワーストケースよりも経験的にはるかに良い性能を発揮する。

6. Conclusion

まとめ

- オンラインアルゴリズムは、情報が時間とともに明らかになる現実の問題に有効。
- 本研究は、需要が徐々に判明するIRPに対するオンラインアルゴリズムを提案。
- 整数計画を用いたアルゴリズムで、ワーストケースに対する性能保証（競合比）を理論的に証明。
- 実験では、提案手法が高精度かつ高速で現実的にも有効であることを確認。
- 今後は、さらに効率的で高性能なオンライン手法の開発が課題。

- 重村が卒論で扱った災害時に被害状況を把握して動的に経路を決定するという問題設定に使えるかなと思いつきながら読んだ。
- 理論的にはシンプルでありながら、計算実験によりその有用性が示されているという点でインパクトがあったと感じた。
- 卒論のときにレビューをしたVRP関連の研究がいくつか出てきて自分の中での積み重ねを若干感じた（全然まだまだだが）。

2-1. 研究フレームワーク | 動的なUAV4次元軌道最適制御

4

動的に協調と衝突回避を考慮しながら、道路調査カバレッジを最大化

UAVを用いた道路啓開調査 → 調査済み道路カバレッジを最大化するのが目的



① 動的に軌道を更新

古典的なVRPでは静的に解を計算していたが、被災/調査状況によって調査順序を変更

* VRP: 配車計画問題

② UAVごとの観測情報

すべての情報は観測できない前提で、部分観測マルコフ過程を用いて、各UAV内で計算を行う

③ 平均場で協調と衝突回避

平均場近似を用いて、計算量を削減しつつ、UAV同士の協調と衝突回避を行う

重村奏画, 部分観測マルコフ過程を用いたUAVの4次元軌道最適制御-災害時の道路啓開を対象に- 卒論発表資料, 2025

参考資料

- 内閣府. 南海トラフ巨大地震における医療対策の取組状況. 2023-10-02. https://www.bousai.go.jp/jishin/nankai/taisaku_wg_02/9/pdf/2.pdf (2025-05-06閲覧) .
- 三井倉庫. VMIとは何か？小売業から始まり、サステナブルな物流としても期待. <https://www.mitsui-soko.com/column/2022/04> (2025-05-07閲覧) .
- 一般社団法人富山県薬業連合会. 歴史と伝統. <https://www.toyama-kusuri.jp/ja/history/index.html> (2025-05-07閲覧) .
- ウィリアム・J・クック. 驚きの数学 巡回セールスマン問題. 2013. 青土社.
- 河瀬康志, オンライン最適化の競合比解析. 2021-08-04. <https://www.kurims.kyoto-u.ac.jp/coss/coss2021/kawase-lecture.pdf> (2025-05-07閲覧) .