A chance-constrained dial-a-ride problem with utility-maximizing demand and multiple pricing structures

Dong, X., Chow, J.Y.J., Waller, S.T. & Rey, D. (2022). Transportation Research Part E, Vol.158, 102601

2024/5/23
BinN Lab M1
Shogo Hiramatsu

Abstract

- リクエスト受諾/拒否の機構を導入したChance Constrained DARP(CC-DARP)を提案
  Presenting a mechanism for accepting/rejecting user requests called Chance constrained DARP
- 受諾/拒否の決定においてロジットモデルで記述された利用者効用を考慮
  considering of users’ preference written by a Logit Model
- 局所探索をベースとしたヒューリスティクスによる求解
  Solution Method based on a local search based heuristics
- CC-DARPモデルが収益マネジメントにおける新たな意思決定ツールとなることを示唆
  Indicating that the CC-DARP model can be a new decision-support tool to inform on revenue management

Novelty, Usefulness, Reliability

Novelty

- 利用者受諾/拒否の決定に利用者選好を導入したこと
  Introducing users’ preference for accepting/rejecting mechanism

Usefulness

- 最適な運賃を具体的に示唆する結果が得られている点で有用性が高い
  Useful in that results can infer the optimal fare

Reliability

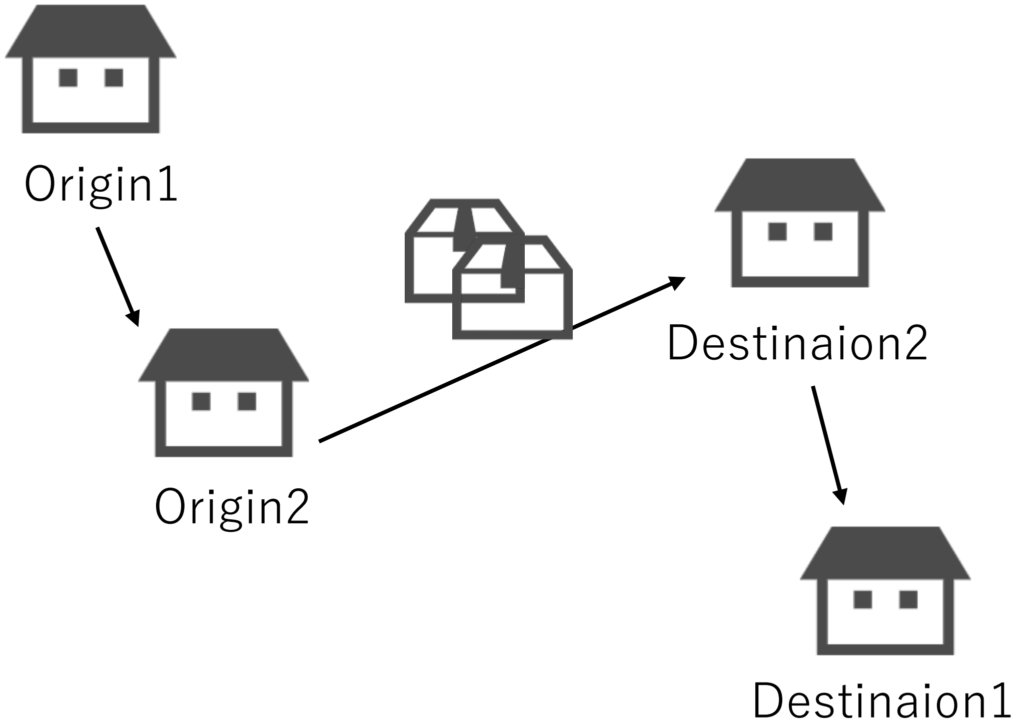
- ヒューリスティクスで求めた近似解を商用MILPソルバーを用いて求めた厳密解と比較している
  Comparing approximate solution with exact solution, which is solved by commercial MILP solver
- 理論上の数値実験と実データを用いた数値実験とで同様の結果が得られている
  Similar results were obtained from numerical examples both in theoretical experiment and in real data experiment

1. Introduction

Dial-a-ride problem (DARP)

輸送計画問題 / Vehicle Routing Problem (VRP)

- 複数の車両を用いて複数の顧客に荷物を集配送する問題 (梅谷 2020)
- Carrying **baggage** with designated pair of origin and destination



Dial-a-ride problem (DARP)

輸送計画問題 / Vehicle Routing Problem (VRP)

複数の車両を用いて複数の顧客に**荷物**を集配送する問題 (梅谷 2020)

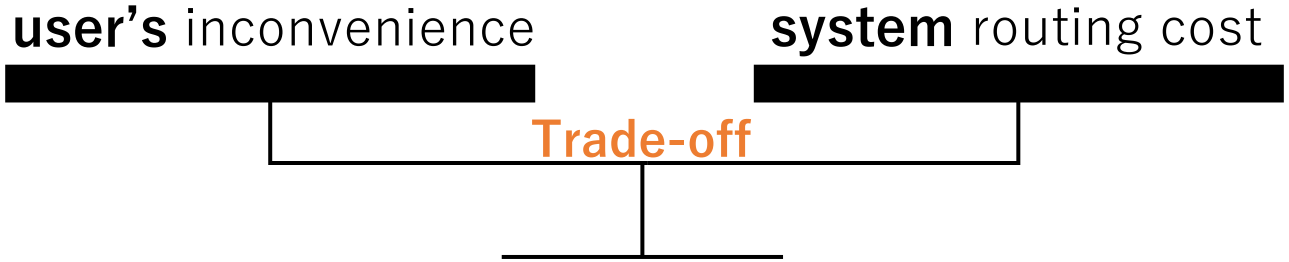
Carrying **baggage** with designated pair of origin and destination

Need to consider **service quality**

Dial-a-Ride Problem (DARP)

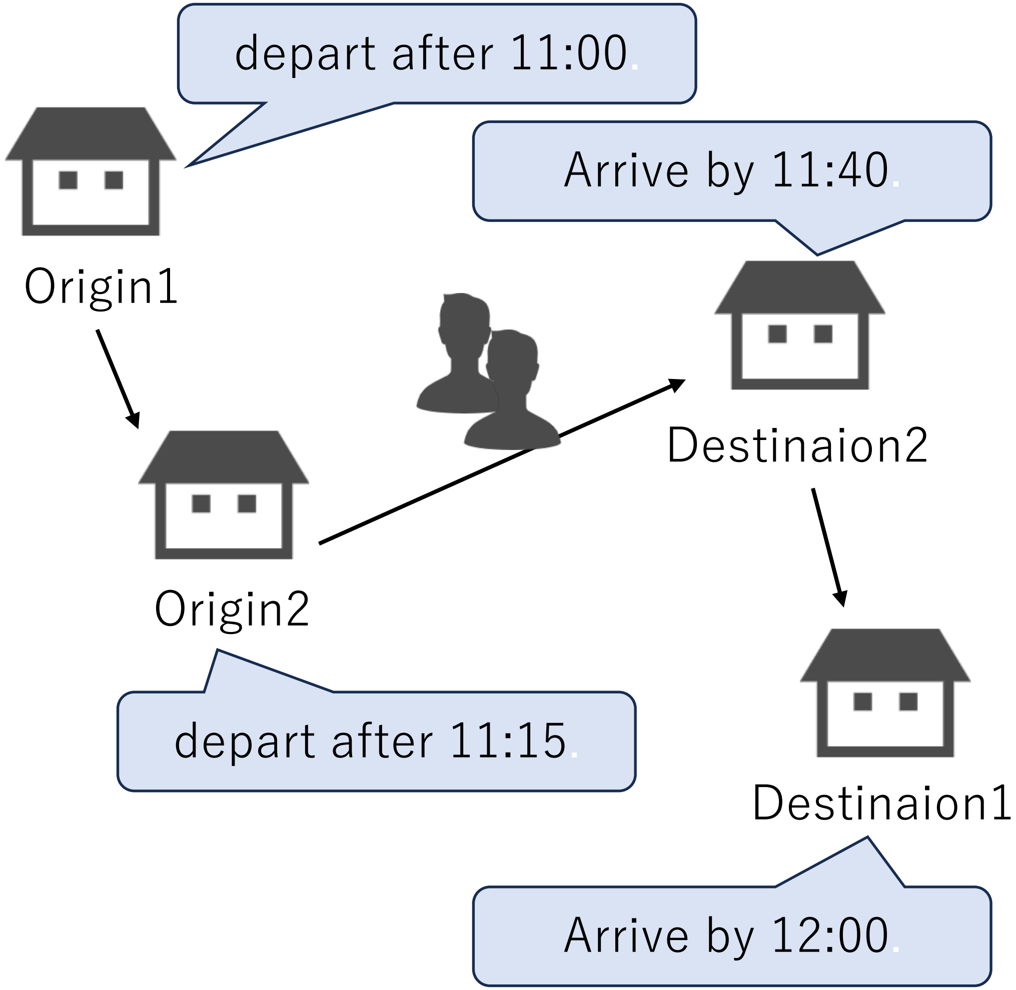
Door-to-door transportation for **people**

- Time window constraints(制約)
- Maximum ride time constraint


user's inconvenience

system routing cost

Trade-off



Purpose and proposal

一般的な原則 / classic principle

全てのリクエストを運ぶ
Serve **all** requests

↓

全員を運ぶようにルートやスケジュールを最適化
Optimise the routing and scheduling **to serve all**

利用者選好の不考慮
Users’ preference not considered

↓

System routing cost **down**

Overall service quality **down**

↓

将来的に利用者数が減少するおそれ

Unstable future ridership

提案 / proposal

利用者選好を統合したDARPモデル
DARP model **integrated with users’ preferences** from DRT operators’ perspective

Relative utility for DRT is **low** → requests **rejected**

2. Literature Review

DARP

standard definition of DARP (Cordeau & Laporte 2003)

↓

Challenge: **trade-off** between **operational cost** and **user inconvenience**

1 objective fundtion: **Users**' inconvenience not emphasized

Minimise system cost

- Minimise total distance travelled
- Maximise vehicle occupancy rate
- minimize fleet(車両) size

Revenue and Profit

→ **requests** are allowed to be **left behind**

- Maximise total system profit
- **Constraints** regarding **customers' inconvenience** could be transferred into **penalty terms in the objective function**
- **One weighted-sum objective** function

DARP with profits

Traveling Salesman Problems (巡回セールスマン問題), VRPs with profit have studied many

↕

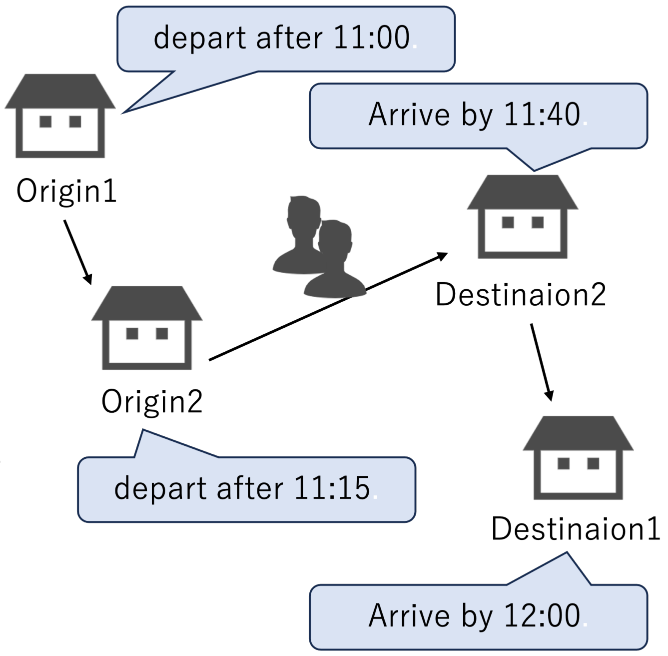
DARP with profit has **hardly ever explored**

- Pickup and Delivery Selection Problem (Schönberger et al., 2003)
  - Pickup and Delivery Problem with Time Window(PDPTW) + **accept decision** of the requests
- DARP with Split Requests and Profits (Parragh et al., 2015)
  - **Requests can be rejected** by DRT(デマンド交通) operators

↓

リクエスト受諾/拒否に**供給側の都合しか考慮されていない**

Requests accept/reject decisions **are based on supply-side features**



PDPTW

↔

Our work: Chance-Constrained DARP (CC-DARP)

accept/reject mechanism for **user selection (利用者選択)**

← **preferences of utility-maximizing users** in the long run

解法 / Solution methods

DARP:（混合）整数計画問題 / (Mixed) Integer Linear Problem (MILP)

→ 多項式時間での求解が困難な**NP-hard**

厳密解法 / Exact solution method

- 枝刈法 / A branch-and-cut algorithm (Cordeau, 2006)
- 列生成法の利用 / column generation technique

近似解法 / Approximate solution method

最適に近い解を少ない計算コストで求める

- Time-sequential insertion heuristic (Jaw et al., 1986)
- To better escape local optima(局所解),
    - **Tabu Search** framework for DARP (Cordeau & Laporte, 2003)
    - Neighbourhood Search (近傍探索)

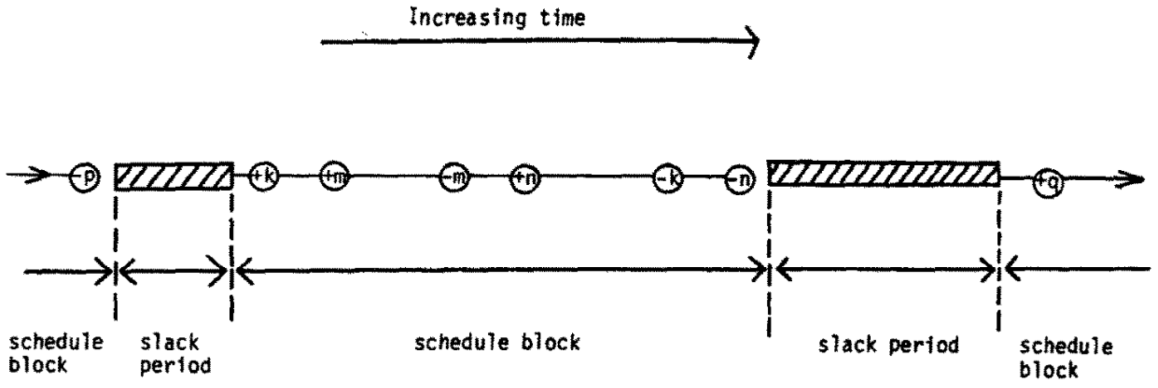


Fig. 1. Part of the work schedule of a vehicle *j*.

Jaw et al. (1986)

↔

Our work: CC-DARP model

- MILP formulation
- Customised **local search(局所探索)** based heuristic and a matheuristic

3. Mathematical formulation

Utility functions

2 travel modes choice: DRT service & private travel(自家用車など)

→ introduce **a logit-based model** to capture users’ utility

Utility for private travel (request i)

$$\hat{V}_i = -\beta_T \underline{t_{i,n+i}} - \beta_F \underline{\hat{c}_i}, \qquad \forall i \in \mathcal{P}.$$

Travel cost of private travel

Travel time between pick-up node and drop-off node

Utility functions

2 travel modes choice: DRT service & private travel(自家用車など)

→ introduce **a logit-based model** to capture users’ utility

Utility for DRT service (request i)

→**復路と往路**を区別 / distinguish between **inbound and outbound**

Inbound trip: 出発時刻の都合が大きい / have preferred departure time

Outbound trip: 到着時刻の都合が大きい / have desired arrival time

Inbound trip

$$V_i = -\beta_F \underbrace{F}_{\text{fare}} - \beta_T \underline{L_i} - \beta_S (\underline{B_i - e_i}),$$

Waiting time at pickup locations

Travel time (ride time)

Outbound trip

$$V_i = -\beta_F F - \beta_T L_i - \beta_S (\underline{l_{n+i} - B_{n+i}}),$$

早く着きすぎの分 / Amount of time that user arrives earlier than expected

Chance constraints

Difference in utility$\Delta U_i = \left(\widehat{V}_i - V_i\right) + \varepsilon = \overline{\Delta U_i} + \varepsilon$

→ random variable(変数) following a logistic distribution（ロジスティック分布）$\Delta U_i \sim (\overline{\Delta U_i}, s)$

「信頼度pで$\Delta U_i \le 0$」を意味する機会制約を導入

Introduce the chance constraints to ensure that $\Delta U_i \le 0$ holds with a confidence level of p

$$\Pr(\Delta U_i \le 0) \ge p, \qquad \forall i \in \mathcal{P}.$$

$$\Leftrightarrow \qquad 0 \ge Q_{\Delta U_i}(p),$$

$Q_{\Delta U_i}(p)$ is the inverse cumulative distribution function（累積分布関数の逆関数）of ΔU_i

$$\Leftrightarrow \qquad 0 \ge \overline{\Delta U_i} + s \ln\left(\frac{p}{1-p}\right), \quad (*)$$

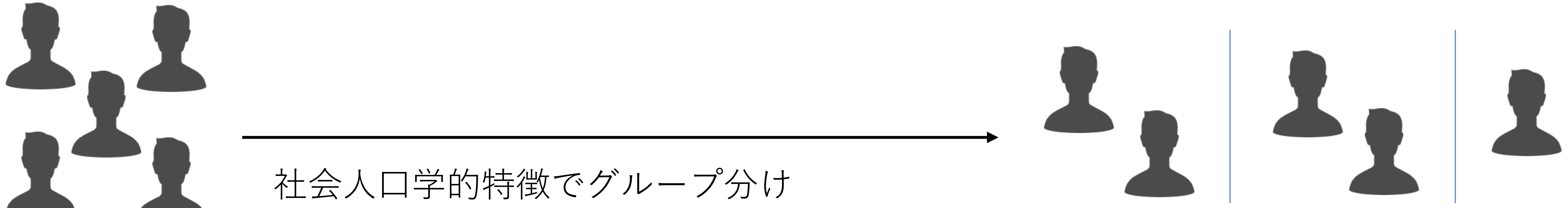
Chance constraints hold(機会制約が成り立つ) when DRT is selected

→ with binary variable(二値変数) y_i which equals to 1 when DRT is selected, very large const. W

$$\overline{\Delta U_i} - (1 - y_i)\mathrm{W} \le -s \log\left(\frac{p}{1-p}\right)$$

Only when $y_i = 1$, (*) holds

Class based fare structures



社会人口学的特徴でグループ分け
Group by socio-demographic characteristics

users

various classes

Parameters: defined for **each class**

Utility for private travel (request i, class m):

$$\hat{V}_i = -\beta_T^m t_{i,n+i} - \beta_F^m \hat{c}_i,$$

Utility for DRT service (request i, class m):

$$V_i = -\beta_F^m f_{im} - \beta_T^m L_i - \beta_S^m (B_i - e_i), \qquad \forall m \in \mathcal{M}, \forall i \in \mathcal{IB} \cap \mathcal{P}_m,$$

$$V_i = -\beta_F^m f_{im} - \beta_T^m L_i - \beta_S^m (l_{n+i} - B_{n+i}), \ \forall m \in \mathcal{M}, \forall i \in \mathcal{OB} \cap \mathcal{P}_m.$$

2 fare structures
- Distance based
- Zone based

Chance constraints (request i, class m):

$$\overline{\Delta U_i} - (1 - y_i)\mathrm{W} \leq -s_m \log(\frac{p_m}{1 - p_m})$$

Chance-constrained DARP formulation

Based on the classic DARP model by **Cordeau (2006)**

Decision Variable (決定変数)

$$x_{ij}^k = \begin{cases} 1 \text{ if vehicle k moves from node i to j} \\ 0 \ \text{ otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 \text{ if request i is accepted} \\ 0 \ \text{ otherwise} \end{cases}$$

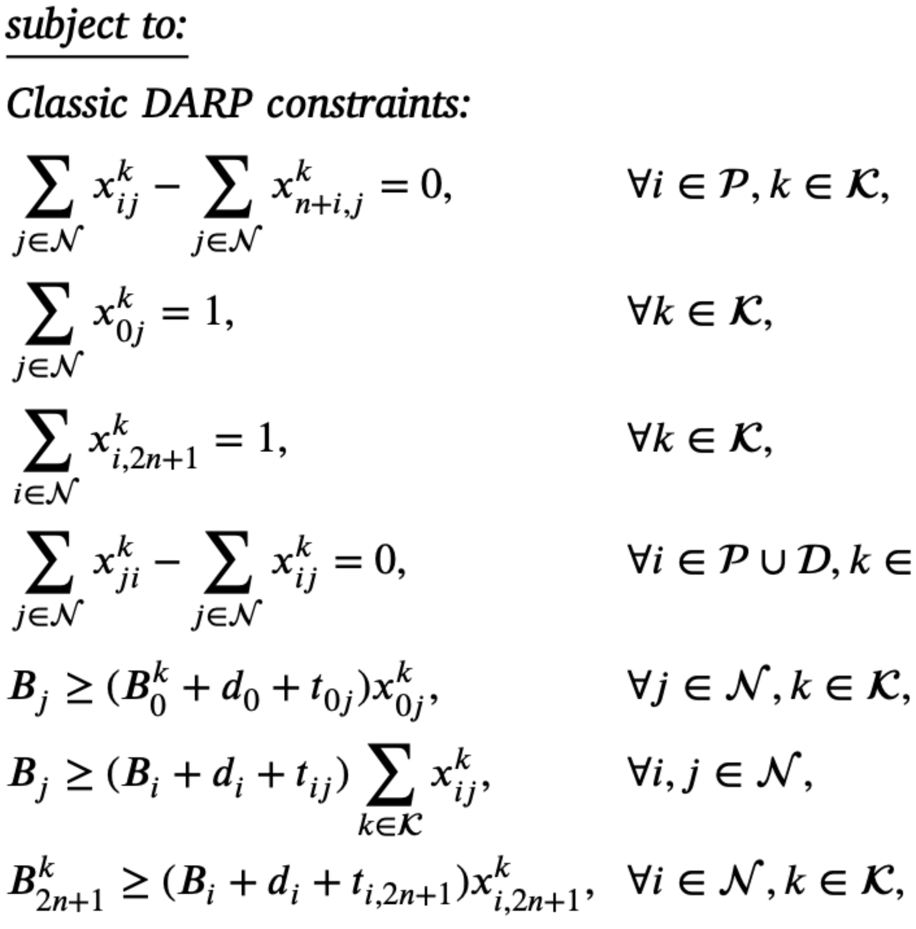
※モデルやアルゴリズムをより理解したい方は
Cordeau & Laporte (2007)によるレビュー論文を
見ると，他のモデルとの比較もできて良いです.
※ For more study, please read the review paper
by Cordeau & Laporte (2007)

Objective Function (目的関数): **Profit maximisation (利益最大化)**

$$\max z = \underbrace{\sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{P}_m} f_{im}\underbrace{q_{im}}_{\substack{\text{load}\\ \text{人数}}} y_i}_{\text{総収入}} - \underbrace{\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} c_{ij}x_{ij}^k}_{\text{総コスト}}$$

Chance-constrained DARP formulation

Constraints



subject to:

Classic DARP constraints:

$$\sum_{j\in\mathcal{N}} x_{ij}^k - \sum_{j\in\mathcal{N}} x_{n+i,j}^k = 0, \qquad \forall i \in \mathcal{P}, k \in \mathcal{K},$$

Pick-up and drop-off nodes for each request are visited by the same vehicle.

$$\sum_{j\in\mathcal{N}} x_{0j}^k = 1, \qquad \forall k \in \mathcal{K},$$

Each route starts at the origin depot

$$\sum_{i\in\mathcal{N}} x_{i,2n+1}^k = 1, \qquad \forall k \in \mathcal{K},$$

Each route ends at the destination depot

$$\sum_{j\in\mathcal{N}} x_{ji}^k - \sum_{j\in\mathcal{N}} x_{ij}^k = 0, \qquad \forall i \in \mathcal{P} \cup \mathcal{D}, k \in \mathcal{K},$$

流量保存則 / Flow conservation

$$\left.\begin{aligned}
&B_j \geq (B_0^k + d_0 + t_{0j})x_{0j}^k, && \forall j \in \mathcal{N}, k \in \mathcal{K},\\
&B_j \geq (B_i + d_i + t_{ij}) \sum_{k\in\mathcal{K}} x_{ij}^k, && \forall i, j \in \mathcal{N},\\
&B_{2n+1}^k \geq (B_i + d_i + t_{i,2n+1})x_{i,2n+1}^k, && \forall i \in \mathcal{N}, k \in \mathcal{K},
\end{aligned}\right\}$$

時間一貫性 / Time consistency

Chance-constrained DARP formulation

Constraints

$$\left.\begin{aligned} & Q_j \geq (Q_0^k + q_{jm}) x_{0j}^k, && \forall m \in \mathcal{M}, j \in \mathcal{N}_m, k \in \mathcal{K}, \\ & Q_j \geq (Q_i + q_{jm}) \sum_{k \in \mathcal{K}} x_{ij}^k, && \forall i \in \mathcal{N}, m \in \mathcal{M}, j \in \mathcal{N}_m, \\ & Q_{2n+1}^k \geq (Q_i + q_{2n+1,m}) x_{i,2n+1}^k, && \forall i \in \mathcal{N}, k \in \mathcal{K}, \end{aligned}\right\}$$

乗車人数の一貫性 / Load consistency

$$L_i = B_{n+i} - (B_i + d_i), \qquad \forall i \in \mathcal{P},$$

Ride time

$$B_{2n+1}^k - B_0^k \leq T, \qquad \forall k \in \mathcal{K},$$

車庫を出てから戻るまでの時間の条件
Duration of routes

$$e_i \leq B_i \leq l_i, \qquad \forall i \in \mathcal{P} \cup \mathcal{D},$$

Time window constraints

$$t_{i,n+i} \leq L_i \leq L, \qquad \forall i \in \mathcal{P},$$

Ride time boundary (限界)

$$\max\left\{0, q_{im}\right\} \leq Q_i \leq \min\left\{Q, Q + q_{im}\right\}, \qquad \forall m \in \mathcal{M}, i \in \mathcal{N}_m, k \in \mathcal{K},$$

乗車人数の制約 / Capacity constraints

$$x_{ij}^k \in \{0, 1\}, \qquad \forall i \in \mathcal{N}, j \in \mathcal{N}, k \in \mathcal{K},$$

$$y_i \in \{0, 1\}, \qquad \forall i \in \mathcal{P},$$

$$\underline{V}_i \leq V_i \leq \overline{V}_i, \qquad \forall i \in \mathcal{P},$$

Chance-constrained DARP formulation

Constraints

Utility for DRT service (request i, class m):

$$V_i = -\beta_F^m f_{im} - \beta_T^m L_i - \beta_S^m (B_i - e_i), \qquad \forall m \in \mathcal{M}, \forall i \in \mathcal{IB} \cap \mathcal{P}_m,$$

$$V_i = -\beta_F^m f_{im} - \beta_T^m L_i - \beta_S^m (l_{n+i} - B_{n+i}), \ \forall m \in \mathcal{M}, \forall i \in \mathcal{OB} \cap \mathcal{P}_m.$$

Chance constraints (request i, class m):

$$\overline{\Delta U_i} - (1 - y_i)\mathrm{W} \leq -s_m \log(\frac{p_m}{1 - p_m})$$

Accept/Reject constraints

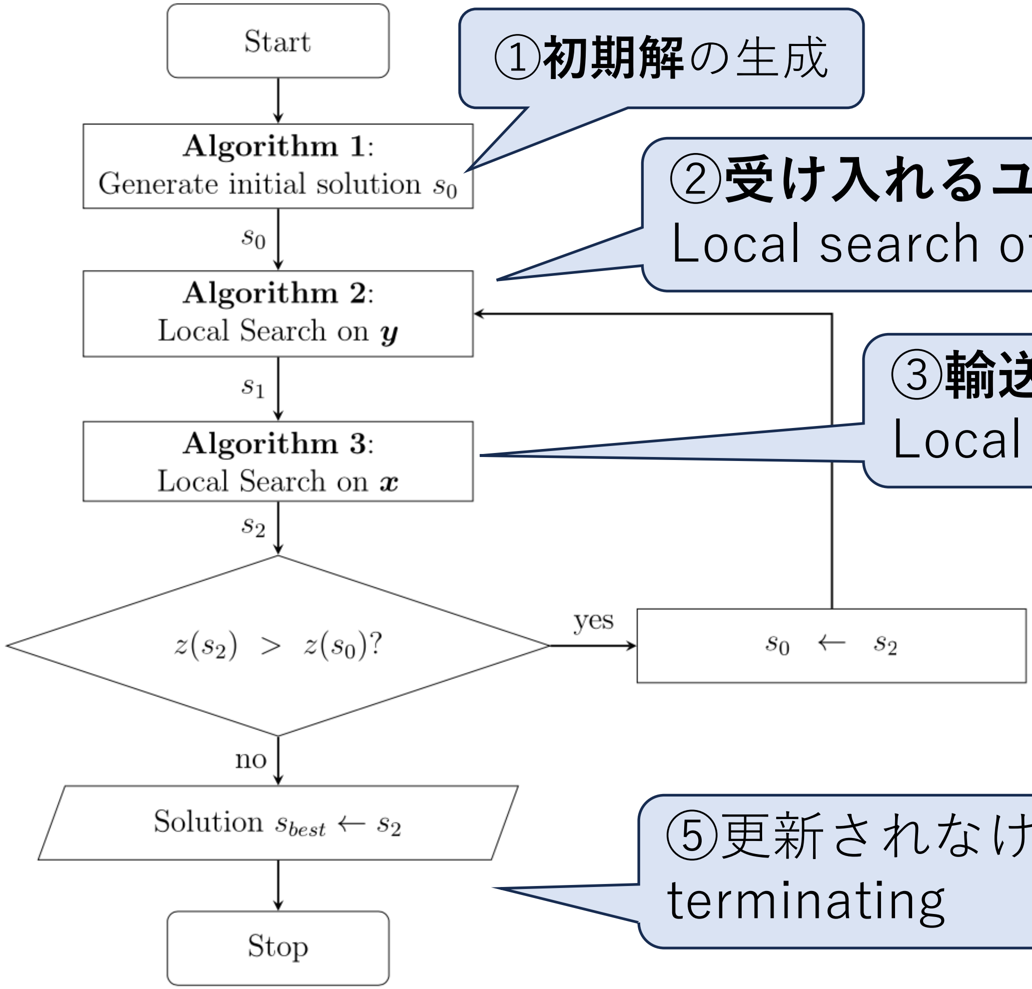
$$\sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{K}} x_{ij}^k = y_i, \qquad \forall i \in \mathcal{P}.$$

利用者iを受け入れるとき，ある車両がノードiに停車する

4. Solution methods

Overview of LS-H algorithm

Proposal: a **local search** based descent heuristic (LS-H) algorithm



①**初期解**の生成

②**受け入れるユーザー**の探索
Local search of user selection

③**輸送ルート**の探索
Local search of routing optimization

④解の更新
updating

⑤更新されなければ終了
terminating

Generation of initial solution

1. e_i が大きくなるようにリクエストを並び替えて挿入順リストInOrとする
   Sorting requests as e_i increases and set that list as Inserting Order list ‘InOr’
   ※ e_i is left side of time window

2.Time windowをより厳しくする / Tighten time window

往路

$$B_i \le \underline{l_{n+i} - t_{i,n+i} - d_i}, \quad \forall i \in \mathcal{OB},$$

$$B_i \ge \underline{e_{n+i} - L - d_i}, \quad \forall i \in \mathcal{OB}.$$

Dの到着最遅時刻
-直接移動した場合の移動時間
-Oでの滞在時間

Dの到着最早時刻
-乗車時間の上限
-Oでの滞在時間

復路

$$B_{n+i} \le l_i + d_i + L, \quad \forall i \in \mathcal{IB},$$

$$B_{n+i} \ge e_i + d_i + t_{i,n+i}, \quad \forall i \in \mathcal{IB}.$$

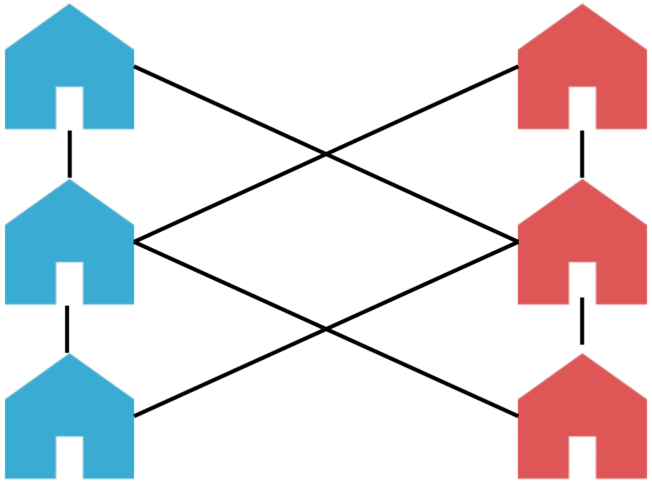
Generation of initial solution

3. Calculate decentralization index D_i and direct travel time index TT_i

$$D_i = \frac{\sum_{j\in\mathcal{N}\setminus\{i,n+i\}} t_{ij} + \sum_{j\in\mathcal{N}\setminus\{i,n+i\}} t_{n+i,j}}{\sum_{i'\in\mathcal{P}}\sum_{j\in\mathcal{N}\setminus\{i',n+i'\}} t_{i'j} + \sum_{i'\in\mathcal{P}}\sum_{j\in\mathcal{N}\setminus\{i',n+i'\}} t_{n+i',j}}, \qquad \forall i \in \mathcal{P}.$$

$$\frac{\text{Total distance of pick-up node of request i from all other nodes + that of drop-off node of request i}}{\text{分子の全リクエストについての総和 (sum of numerator for all requests)}}$$

D_iは他のリクエストから離れている度合いを示す
→ D_iが高いほど後々にルートに入れるのが大変
→ D_iが高いほど先に解に入れるべき



Pick-up node of i

Drop-off node of i

Numerator: sum of the link

Generation of initial solution

3. Calculate decentralization index D_i and direct travel time index TT_i

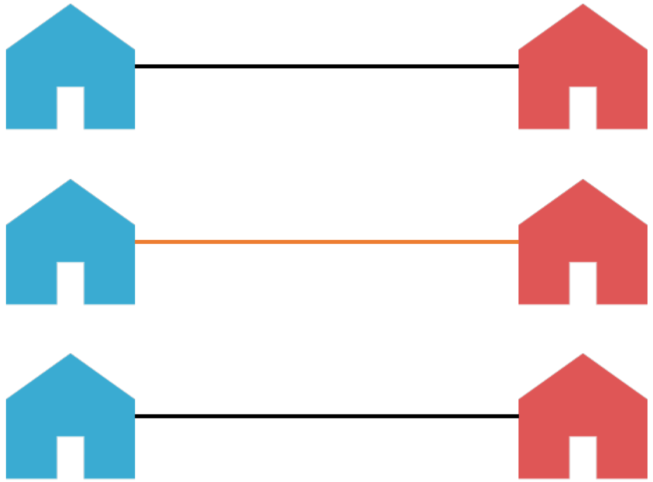
$$TT_i = \frac{t_{i,n+i}}{\sum_{i' \in \mathcal{P}} t_{i',n+i'}}, \qquad \forall i \in \mathcal{P}.$$

$$\frac{\text{Total distance between pick-up and drop-off node of request}}{\text{分子の全リクエストについての総和 (sum of numerator for all requests)}}$$

TT_iは他のOD距離の短さの度合いを示す

→ TT_iが短いほど後々にルートに入れると
リクエストiの乗車時間が冗長になる

→ TT_iが短いほど先に解に入れるべき



Numerator: length of orange link
Denominator: sum of all links

Generation of initial solution

4. Calculate indexG_i

$$G_i = \omega(1 - D_i) + (1 - \omega)TT_i.$$

The shorter G_i is, the more priority to be inserted request i has

5. Replace the first K requests in InOr with the smallest G_i

K: the number of vehicles

Adjust the remaining InOr by swapping i-1 and i, if the following inequation holds

$$G_{i-1} - G_i \geq \delta.$$

優先順位があまりにひっくり返っているものについては，優先順位が高い順になるように並び替える

Generation of initial solution

6. Do following procedure for each request in InOr

① Try to insert into every vehicle

② **If all insertion is infeasible**, add that request into the unvisited request pool v
もしどの挿入も実行不可能なら，このリクエストを「未訪問リクエストv」に入れる
Else, **insert** that request to vehicle **where additional cost is the lowest** and add that request into visited requests pool v
そうでなければ，最小のコストで挿入できる車両にこの車両を入れ，このリクエストを「訪問済みリクエストv」に入れる

初期解の完成!
Initial solution is generated!

Local search on user selection

3つの操作 / 3 operator

- ADD
  - **Add a request into vehicle** work-schedules and **the best insertion is adopted**
- REM
  - **Remove a request** from vehicle work-schedule and return the remaining work schedule
  - The rest of the routing sequence does not change
- SWAP
  - **Removing one** existing request and **adding** one request from the unserved requests pool v
  - ADD and REM executed **simultaneously(同時に)**

Local search on user selection

Algorithm

① for every request in unserved requests pool:

Apply **ADD**

Record obtained vehicle work-schedules

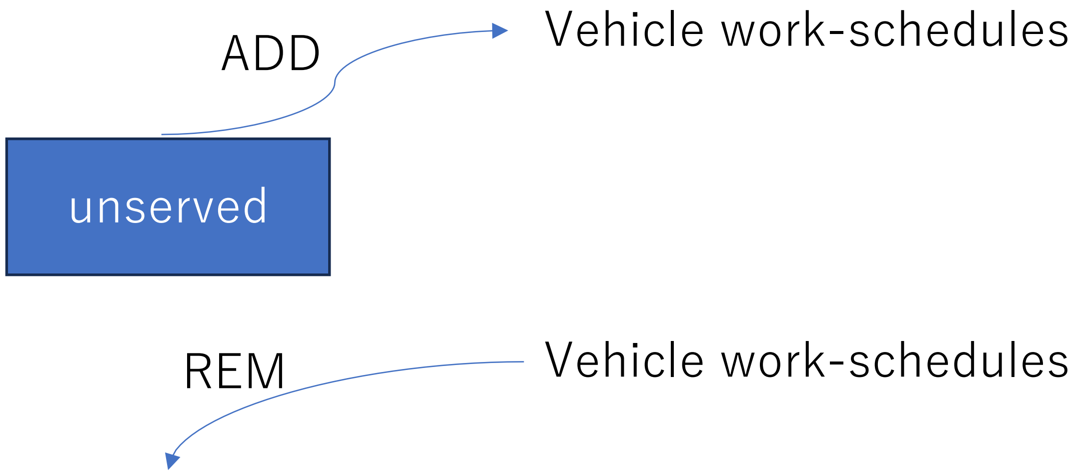
② for every request in visited requests pool

Apply **REM**

Record obtained vehicle work-schedules

③ find **best vehicle work-schedules obtained in ① and ②**

update unserved requests pool and visited requests pool



ADD

Vehicle work-schedules

unserved

REM

Vehicle work-schedules

Local search on user selection

Algorithm

④ for every combination of i in visited requests pool and j in unserved requests pool

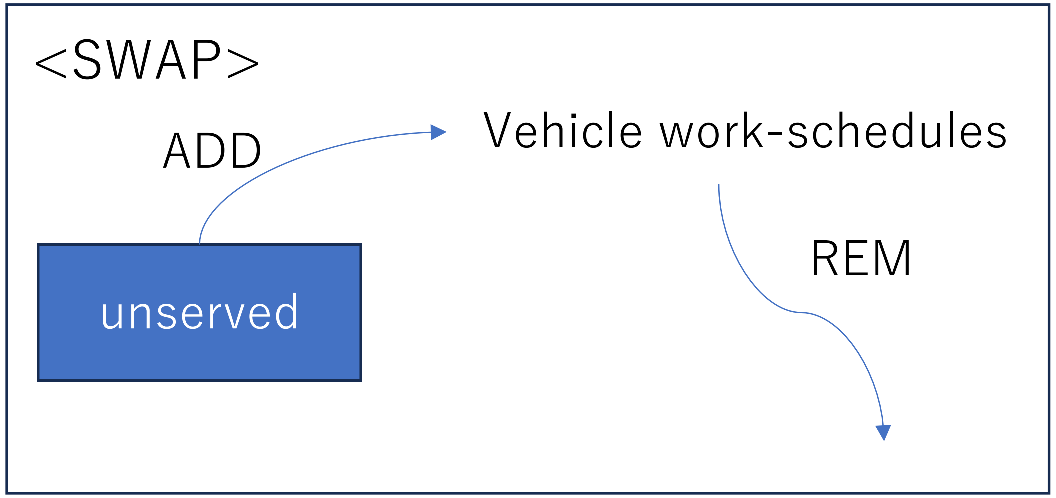
  Apply **SWAP**

  Record obtained vehicle work-schedules

⑤ find **best vehicle work-schedules obtained in ④**

update unserved requests pool and visited requests pool

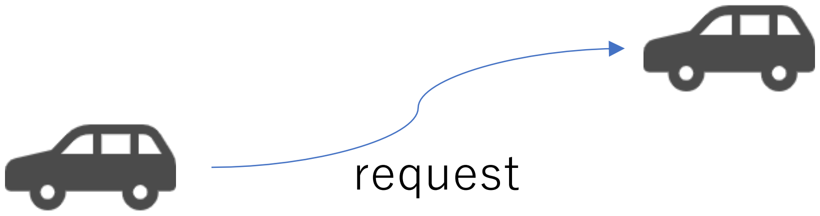
⑥ repeat until no update



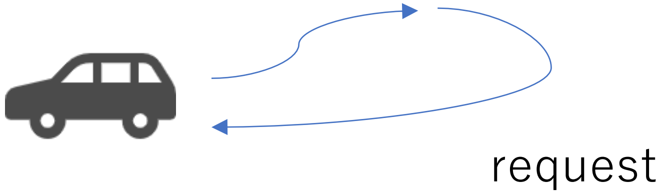
Local search on routing optimization

2つの操作 / 2 operator

- RA
  - **Inter-tour exchange**
  - **Re-Assigns** request to **another vehicle**



- RI
  - **Intra-tour exchange**
  - **Re-Insert** request into its **originally assigned vehicle**



- If RA fails, RI is applied

Local search on routing optimization

Algorithm

① for every request in visited requests pool

If **feasible**, Apply **RA**

Record obtained vehicle work-schedules

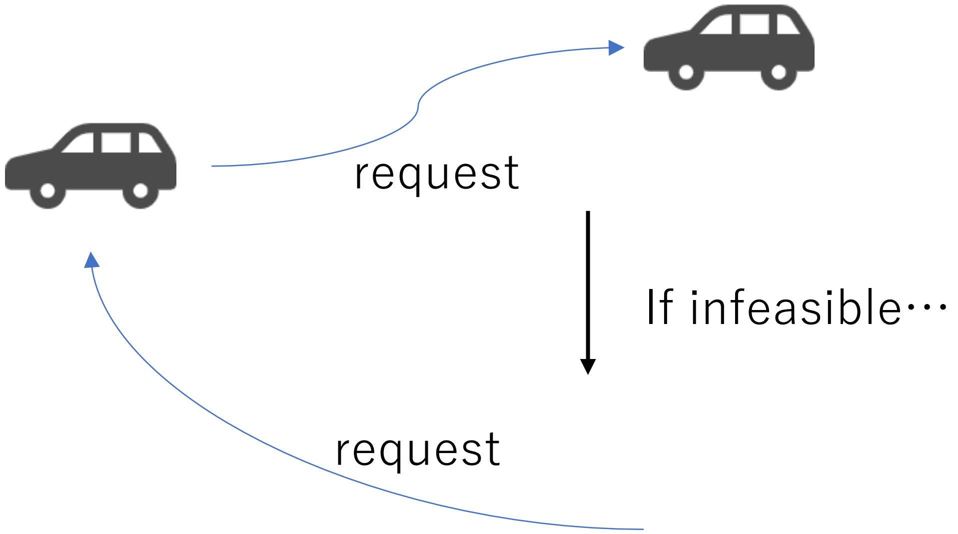
Else, Apply **RI**

Record obtained vehicle work-schedules

② find **best vehicle work-schedules obtained in ①**

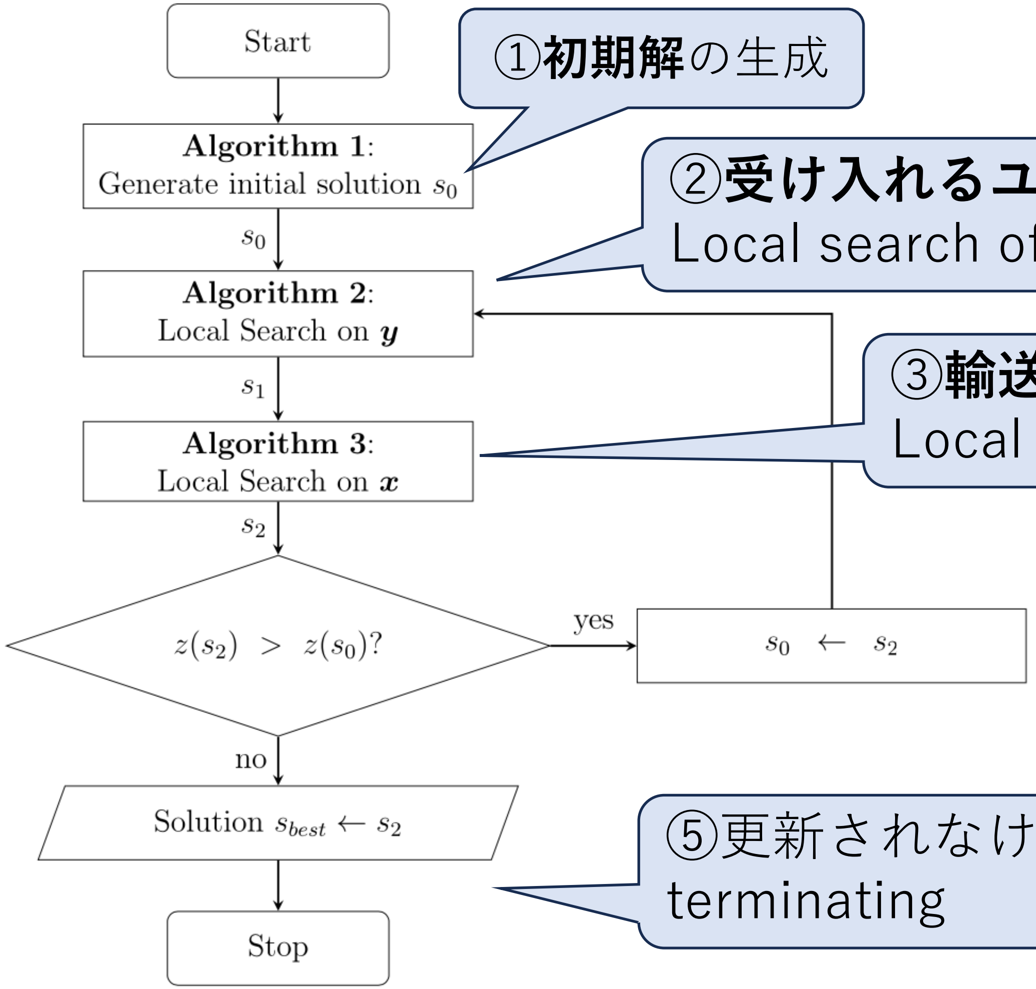
update

③ repeat until no update



Overview of LS-H algorithm

Proposal: a **local search** based descent heuristic (LS-H) algorithm



①**初期解**の生成

②**受け入れるユーザー**の探索
Local search of user selection

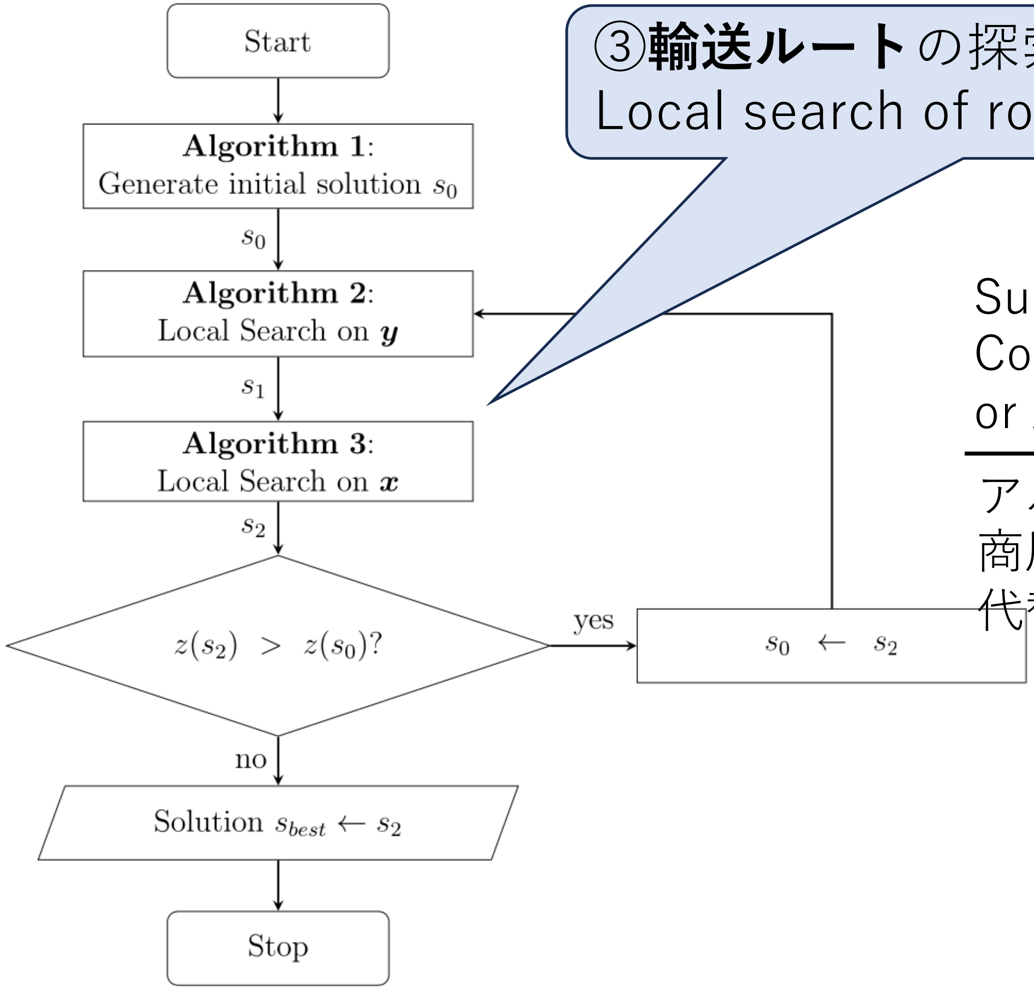
③**輸送ルート**の探索
Local search of routing optimization

④解の更新
updating

⑤更新されなければ終了
terminating

MATH-H algorithm

LS-H algorithm



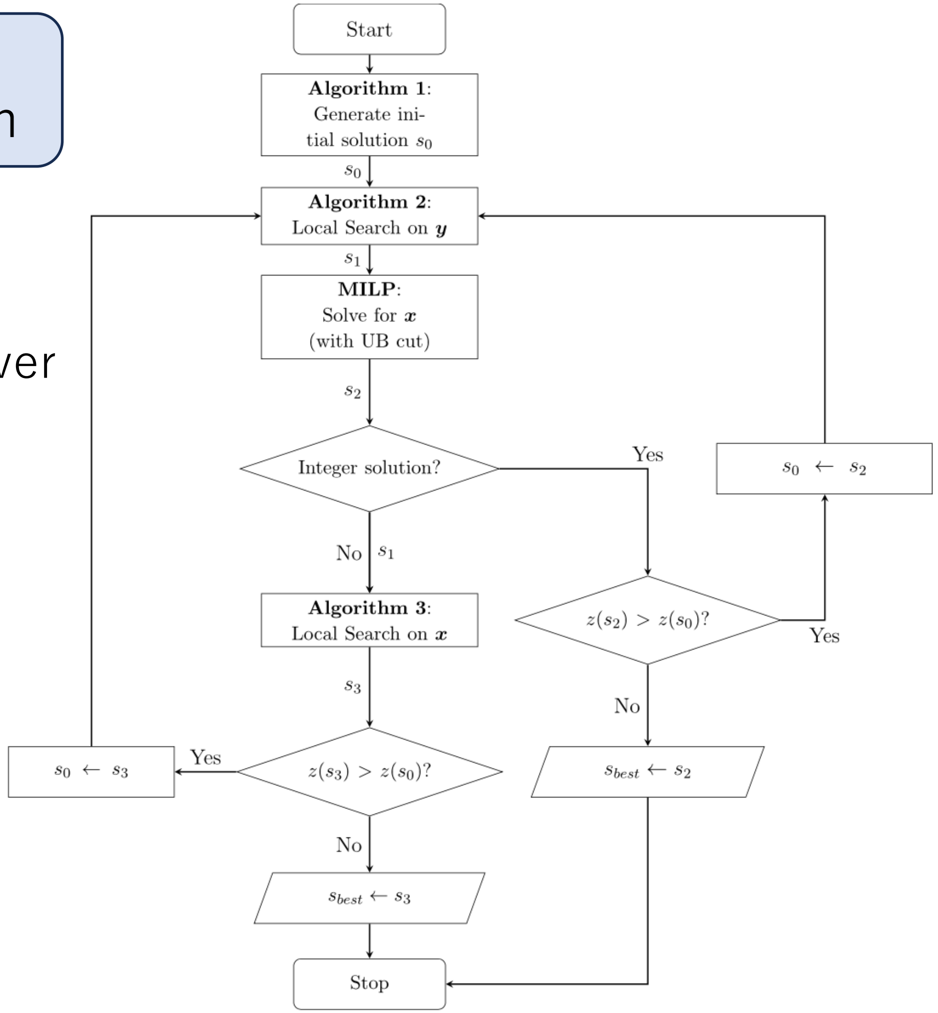
③**輸送ルート**の探索

Local search of routing optimization

Substitute
Commercial MILP solver
or Algorithm3

アルゴリズム3を
商用MILPソルバーで
代替する

MATH-H algorithm



5. Numerical experiments

Outline of Numerical Experiments(数値実験)

Experiments design

- Requests: 16-96
- Number of vehicles: 2-8
- Vehicle capacity: 3

Experimental Environment

- PC with 16GB of RAM, intel i7 processor at 2.9GHz
- CPLEX 12.10 is used as commercial solver

What experiment?

- Comparison between classic-DARP and CC-DARP
  従来のDARPと提案したCC-DARPの比較
- The sensitivity analysis of flat fare and comparing of solution method
  一律運賃fの感度分析と解法の評価
- Experiments with New York City taxi data

Comparison between classic-DARP and CC-DARP

CC-DARP: Users' utility is considered

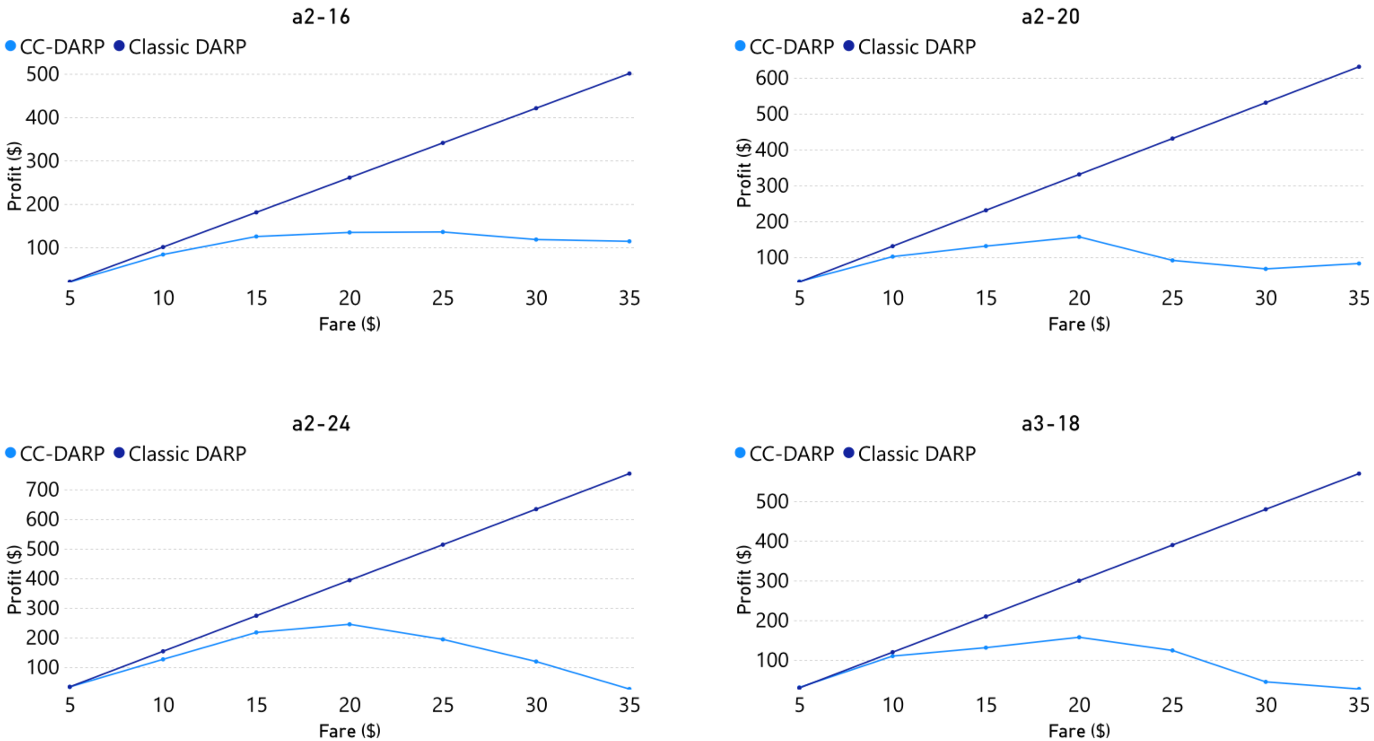


Fig. 5. Profit trends for CC-DARP and Classic DARP formulations.

[-10, 10]x[-10,10]の平面上で実験

classic DARP

- Profit is propotilnal to fare
  → unrealistic
- 運賃に比例して収益が増加しており非現実的

CC-DARP

- Nonlinear pattern / 非線形
- If fare is too high, profit decreases
- 運賃が高すぎると収益が低下する結果に → 現実的

利用者効用を用いた機会制約の導入で，収益マネジメントが可能に
Chance-constrained approach can inform on revenue management

The sensitivity of flat fare and comparing method

Comparison between LS-H and MATH-H

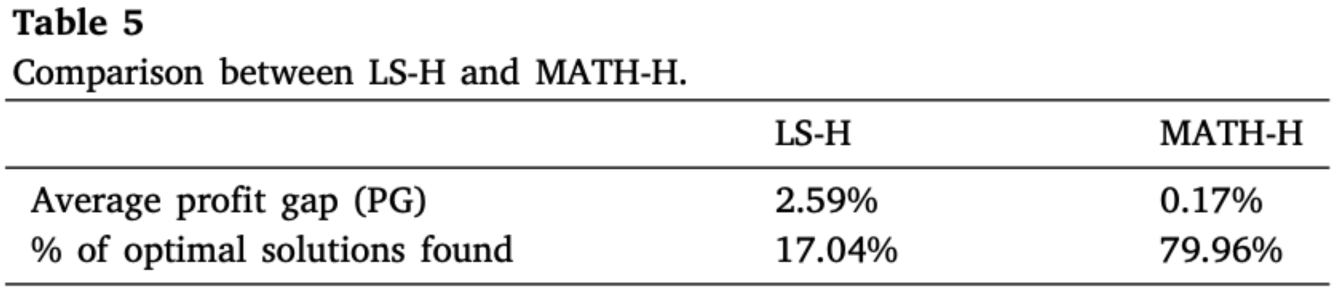


Table 5
Comparison between LS-H and MATH-H.

	LS-H	MATH-H
Average profit gap (PG)	2.59%	0.17%
% of optimal solutions found	17.04%	79.96%

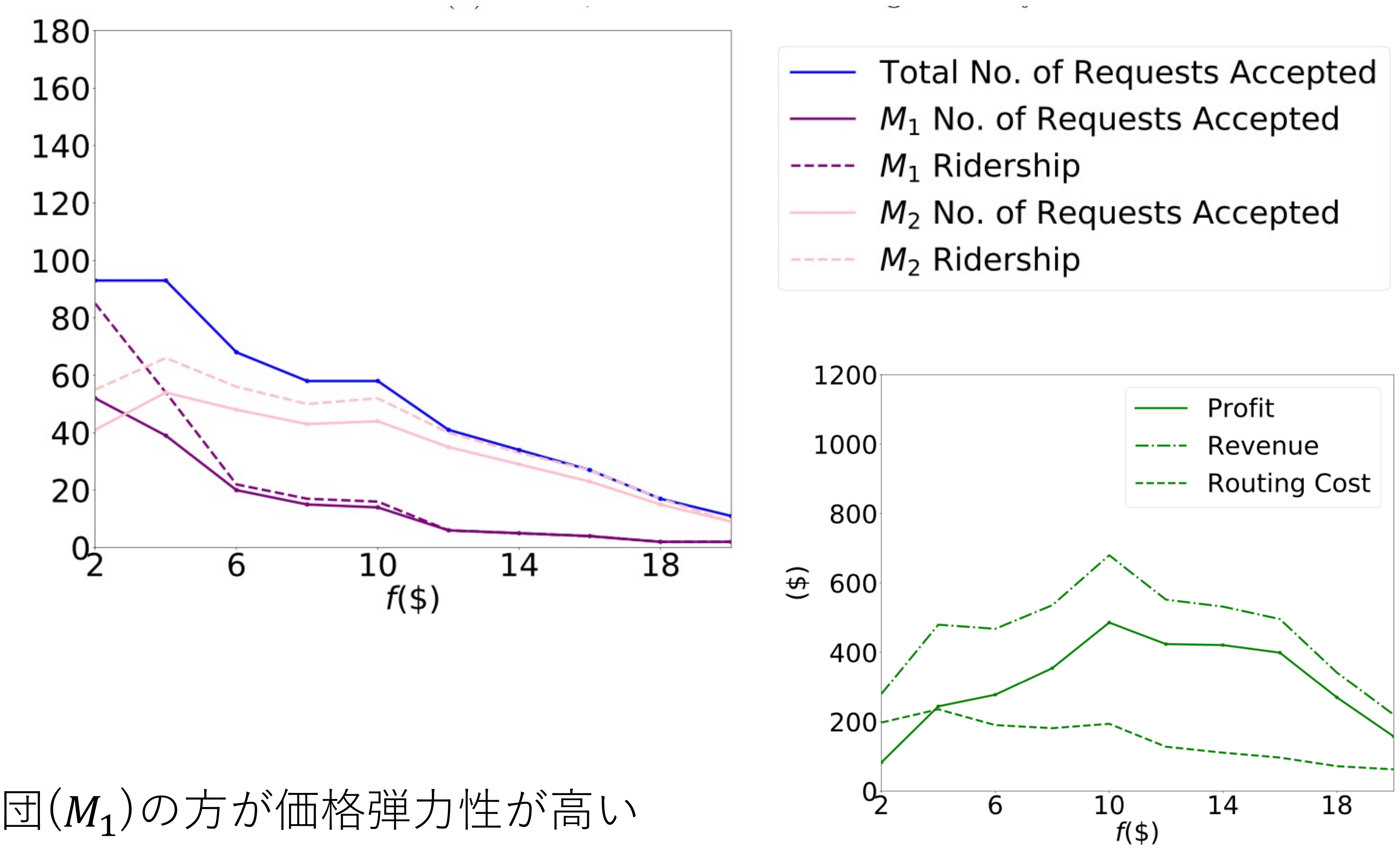
MATH-Hは80%近くのケースで最適解を
得られている
↓
利用者選択の局所探索アルゴリズムの有効性
が示された
proposed local search on user selection
is good enough to find the optimal selection

- 中規模のケース：MATH-Hは厳密解に近い解を得られている
  medium-sized instances: MATH-H is successful in reducing the profit gap
- 大規模のケース：MATH-HもLS-Hも厳密解に近い解を得られている
  large instances: Both LS-H and MATH-H problem well
- 運賃が小さいケースではLS-Hは収束するのに苦労した
  LS-H struggles to converge when a cheap fare is charged

Experiments with NYC taxi data (2006)

23,772トリップから，**最も頻繁に訪問されている出発地と目的地を含む143のリクエストを抽出**し，数値実験を行った．

結果
- 簡易平面上での数値実験と同様に運賃が一定値を超えると利益が低下する．
- 一律運賃，距離ベースの運賃，ゾーンベースの運賃を比較すると，**ゾーンベースの運賃が利益，利用者数を最大化する**
- 一律運賃では時間価値の高い集団(M_1)の方が価格弾力性が高い



(a) Profit, revenue and routing costs - f.

6. Conclusion

Conclusion

Formulation

- Introduction of chance-constrained DARP model that **captures users' preferences**
- Captures users' preferences via a Logit Model

Solution Method

- Local search based heuristic algorithm (LS-H) and matheuristic (MATH-H)

  → **performed well compared to the best integer solution** by an exact MILP solver

Result

The **proposed CC-DARP formulation** provides a new decision-support tool to inform on **revenue management** for DRT services.

Future Work

- Consideration of other alternative transportation modes
- Developing an efficient exact solution algorithm
- Improving the robustness of heuristics for low fare

所感

- DARPの定式化自体に行動モデルを組み込むのは面白いと思った.
  - 利用者自身の行動選択結果をリクエストとして定式化しているのではなく，利用者の効用関数をもとに**事業者が**受け入れ可否を決めている.
- 複数の解法による求解結果の比較がなされているので，説得力がある
- 一部に近似解法を用いた求解によって，80%程度の割合で厳密解を求められていることは意外だった．最適解に到達しやすい変数のみ近似解法を用いて求める，といった方法で計算時間を短縮できるのであれば，試してみても良さそう.
- ゾーンベースの運賃構造がベストな戦略と言っているにもかかわらず，運賃体系によるprofitの違いを比較したグラフがなかったのは残念に感じた.