Joint location and cost planning in maximum capture facility location under random utilities

Duong, N. H., Dam, T. T., Ta, T. A., & Mai, T. (2023). Joint location and cost planning in maximum capture facility location under random utilities. Computers & Operations Research, 159, 106336.

理論談話会#16(2024/7/20)

M1 佐野

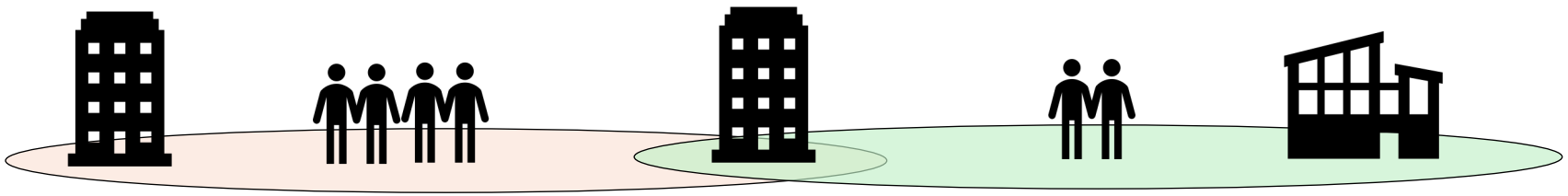
Summary

ランダム効用理論に基づく施設立地計画withコスト最適化

顧客の効用は、施設の配置だけでなく施設にかける予算にも影響を受ける
➡施設立地最適化とコスト最適化の統合問題を考える

施設立地とコストの統合問題であり、既存の解き方だと非凸
➡効用を確定項と誤差項の積として定式化

乗法型ランダム効用理論を用いる解法のアプローチを3つ提示
①CONIC reformulation
②Multicut outer-approximation
③Local search heuristic
➡数値実験でこれらの解法の性能を比較



1. Introduction
2. Problem formulation
3. Relation between MRUM and ARUM models
4. Solution methods
5. Numerical experiments
6. Conclusion

1. Introduction

2. Problem formulation

3. Relation between MRUM and ARUM models

4. Solution methods

5. Numerical experiments

6. Conclusion

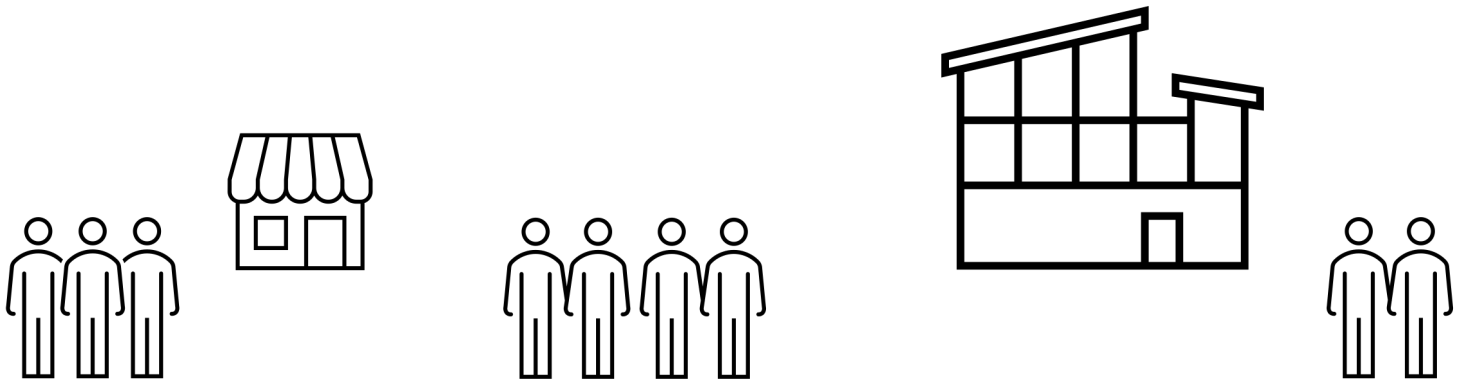
1. Introduction

顧客の需要がランダム効用最大化（RUM）に従うと仮定した施設立地問題

既存のRUMベースの施設立地問題は**立地のみ**に焦点を当てている
➡顧客の需要が施設の開設/運営コストなどの他要因によって左右される事実を考慮していない
➡本研究では、施設**立地**と施設にかける**コスト**を統合した最適化問題に取り組む

施設とコストの統合最適化問題では、施設の開設場所の選択と施設に利用可能な予算の配分を決定することによって、顧客の需要獲得を最大化する
(需要)獲得最大化問題は、**Maximum Capture Problem (MCP)** とも呼ばれる

ここでは、**施設に高い予算を費やすと**、顧客にとって施設が魅力的になり、**選択される可能性が高くなる**という考え方に基づいて、施設コストを顧客の効用に組み込んでいる（事業者目線）



1. Introduction

新規施設の場所の選択と予算の配分決定によって顧客需要を最大化
➡ **Maximum Capture Problem** (MCP, 最大補足問題)

バイナリ変数(場所の選択)と連続変数(コスト)を持つ最適化問題として定式化
二項変数と連続変数の両方を含むため、既存の厳密なアルゴリズムは直接適用できない

そこで、**MRUM（乗法型ランダム効用最大化）**に着目する

ARU（加法型ランダム効用）　MRU（乗法型ランダム効用）

$$U_{ni} = V_{ni} + \varepsilon_{ni} \qquad U_{ni} = V_{ni}\varepsilon_{ni}$$

Table 1
Solution methods.

ARUMとMRUMの解法の比較

Problem	MCP under ARUM	**MCP under MRUM**
Cost optimization	Highly non-convex, intractable to solve	**Exact methods:** - Convex optimization solver - Conic reformulation
Joint location and cost optimization	Intractable	**Exact methods:** - Conic reformulation - Outer Approximation **Heuristic method:** - Local search

コスト最適化

・ARUMでのコスト最適化問題は非凸であり複数の局所最適解が存在する可能性がある

・MRUMでは凸最適化として解くことができる

立地＆コスト最適化

厳密解法
①CONIC reformulation
②Multicut outer-approximation
ヒューリスティック解法
③Local search heuristic

1. Introduction

2. Problem formulation

3. Relation between MRUM and ARUM models

4. Solution methods

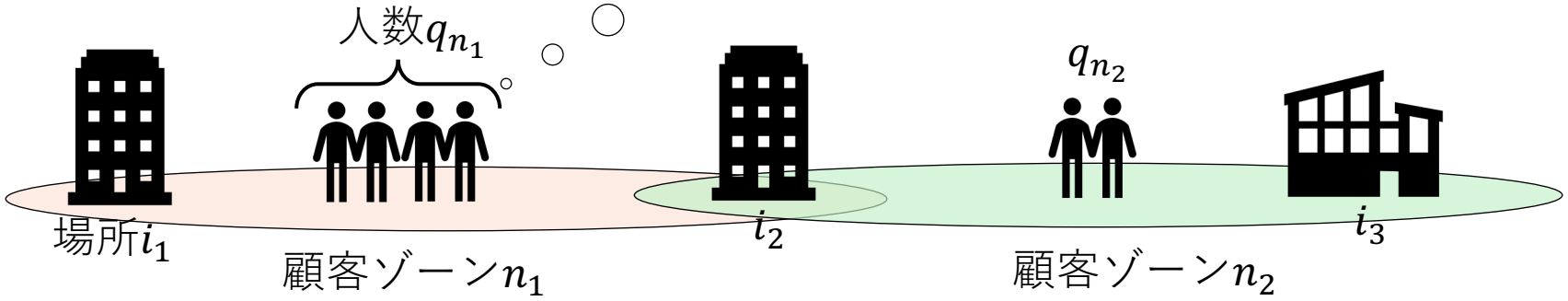
5. Numerical experiments

6. Conclusion

2. Problem formulation

問題設定

$$P(i|S_n) = \frac{e^{\ln V_{ni}}}{\sum_{j \in S_n} e^{\ln V_{nj}}} = \frac{V_{ni}}{\sum_{j \in S_n} V_{nj}}$$



ARUMとMRUMの基本式

ARUM

効用関数

$$U_{ni} = V_{ni} + \varepsilon_{ni}$$

MNLの下で誤差項がガンベル分布に従うと仮定

選択確率

$$P(i|S_n) = \frac{e^{V_{ni}}}{\sum_{j \in S_n} e^{V_{nj}}}$$

notation
n：個人
i：選択肢
S_n：選択肢集合

MRUM

効用関数

$$U_{ni} = V_{ni}\varepsilon_{ni}$$

$$P(V_{ni}\epsilon_{ni} \geq V_{nj}\epsilon_{nj}\ \forall j \in S_n) = P(\ln V_{ni} + \ln \epsilon_{ni} \geq \ln V_{nj} + \ln \epsilon_{nj},\ \forall j \in S_n)$$

対数をとることで加法型と似た構造の式を書ける

選択確率

$$P(i|S_n) = \frac{e^{\ln V_{ni}}}{\sum_{j \in S_n} e^{\ln V_{nj}}} = \frac{V_{ni}}{\sum_{j \in S_n} V_{nj}}$$

2. Problem formulation

立地＆コスト統合の最適化（ARUM）の立式

顧客ゾーン$n \in I$での個人の選択確率

$$P^{\mathrm{ARUM}}(i|S,\mathbf{V}^n) = \frac{e^{V_{ni}}}{U_n^c + \sum_{i' \in S} e^{V_{ni'}}}$$

立地＆コスト統合のMCPにおける確定項

$$V_{ni} = a_{ni}\, x_i + b_{ni}$$

顧客ゾーンnの施設iに費やされたコストに対する感度を表すパラメータ
➡施設にかけられた費用が顧客の効用に影響するという仮定

立地＆コスト統合のMCP

ゾーンnでの顧客人数

$$\max_{S \in \mathcal{S},\ \mathbf{x}} \left\{ f^{\mathrm{ARUM}}(S,\mathbf{x}) = \sum_{n \in I} q_n \sum_{i \in S} \frac{e^{a_{ni} x_i + b_{ni}}}{U_n^c + \sum_{i' \in S} e^{a_{ni'} x_{i'} + b_{ni'}}} \right\}$$

$$\text{subject to} \quad \sum_{i \in S} x_i \le C, \quad \text{予算制約}$$

$$|S| \le K$$

$$x_i \in [L_i, U_i],\ \forall i \in S \quad (1)$$

$$\mathbf{x} \in \mathbb{R}_+^{|S|},$$

notation

n ：顧客ゾーン
i ：新施設
U_{ni}：個人nのiでの効用
m ：新施設を配置可能な場所
$[m]$：全ての配置可能場所
I ：すべての顧客ゾーンの集合（**顧客の特徴による分類と地理分布による分類の可能性のどちらもある**）
q_n：それぞれのゾーンnにいる顧客の数
V_{ni}：顧客$n \in I$の施設$i \in [m]$における確定項
V^n： V_{ni}でのサイズmのベクトル
S ：立地可能場所の部分集合, $S \subset [m]$
U_n^c：顧客ゾーンnにおける競合他社の施設の選択効用

a_{ni}：顧客ゾーンnの施設iに費やされたコストに対する感度を表すパラメータ
x_i：施設iに費やすコスト
b_{ni}：距離や駐車の可能性などの顧客の選択に影響する他の要素

C ：新施設への予算の最大値
K ：開設される施設数の最大値
L_i, U_i：施設iのコストの下限と上限
$\mathcal{S}$ ：部分集合[m]の全て

2. Problem formulation

立地＆コスト統合の最適化（ARUM）の立式

立地＆コスト統合のMCP-ARUM

$$V_{ni} = a_{ni}\, x_i + b_{ni}$$

$$\max_{\mathbf{y},\mathbf{x}} \left\{ f^{\mathrm{ARUM}}(\mathbf{y},\mathbf{x}) = \sum_{n\in I} q_n \frac{\sum_{i\in[m]} y_i e^{a_{ni}x_i+b_{ni}}}{U_n^c + \sum_{i\in[m]} y_i e^{a_{ni}x_i+b_{ni}}} \right\}$$

$$\text{subject to} \quad \sum_{i\in[m]} x_i \le C \quad \text{予算制約} \qquad (2)$$

$$\sum_{i\in[m]} y_i \le K \quad \leftarrow \text{施設数制約} \qquad (3)$$

$$x_i \le y_i U_i,\ \forall i \in [m] \qquad (4)$$

$$x_i \ge y_i L_i,\ \forall i \in [m] \qquad (5)$$

（(4)(5)：施設費用上限下限制約）

$$\mathbf{x} \in \mathbb{R}_+^m,\ \mathbf{y} \in \{0,1\}^m.$$

$\mathbf{x}$：コスト　$\mathbf{y}$：立地

このARUMの式を踏まえてMRUMバージョンを定式化

notation

n ：顧客ゾーン
i ：新施設
U_{ni}：個人nのiでの効用
m ：新施設を配置可能な場所
$[m]$：全ての配置可能場所
I：すべての顧客ゾーンの集合（顧客の特徴による分類と地理分布による分類の可能性のどちらもある）
q_n：それぞれのゾーンnにいる顧客の数
S：立地可能場所の部分集合, $S \subset [m]$
U_n^c：顧客ゾーンnにおける競合他社の施設の選択効用
a_{ni}：顧客ゾーンnの施設iに費やされたコストに対する感度を表すパラメータ
x_i：施設iに費やすコスト
b_{ni}：距離や駐車の可能性などの顧客の選択に影響する他の要素
C ：新施設への予算の最大値
K ：開設される施設数の最大値
L_i, U_i：施設iにかける予算の下限と上限
S ：部分集合[m]の全て
y_i ：**部分集合Sが場所iを含むなら$y_i=1$, そうでない場合は$y_i=0$の二項変数でSを表すことができる**

2. Problem formulation

MRUM ver.の立式

顧客ゾーン$n \in I$での個人の選択確率(ARUM)

$$P^{\mathrm{ARUM}}(i|S,\mathbf{V}^n) = \frac{e^{V_{ni}}}{U_n^c + \sum_{i' \in S} e^{V_{ni'}}}$$

$$P(V_{ni}\epsilon_{ni} \geq V_{nj}\epsilon_{nj}\ \forall j \in S_n) = P(\ln V_{ni} + \ln \epsilon_{ni} \geq \ln V_{nj} + \ln \epsilon_{nj},\ \forall j \in S_n)$$

対数をとることで加法型と似た構造の式を書ける

MRUMでの選択確率

$$P^{\mathrm{MRUM}}(i|S,\mathbf{V}^n) = \frac{V_{ni}}{1+\sum_{i' \in S} V_{ni'}} = \frac{a_{ni}x_i + b_{ni}}{U_n^c + \sum_{i' \in S} a_{ni'}x_{i'} + b_{ni'}}$$

立地＆コスト統合のMCP-MRUM

$$V_{ni} = a_{ni}\, x_i + b_{ni}$$

$$\max_{\mathbf{y},\mathbf{x}} \left\{ f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}) = \sum_{n \in I} q_n \frac{\sum_{i \in [m]} y_i (a_{ni}x_i + b_{ni})}{U_n^c + \sum_{i \in [m]} y_i (a_{ni}x_i + b_{ni})} \right\}$$

$$\text{subject to} \quad \sum_{i \in [m]} x_i \leq C \quad \text{予算制約}$$

$$\sum_{i \in [m]} y_i \leq K \quad \leftarrow \text{施設数制約}$$

$$x_i \leq y_i U_i,\ \forall i \in [m] \quad \leftarrow \text{施設費用上限下限制約}$$

$$x_i \geq y_i L_i,\ \forall i \in [m]$$

$$\mathbf{x} \in \mathbb{R}_+^m,\ \mathbf{y} \in \{0,1\}^m.$$

$\mathbf{x}$：コスト　$\mathbf{y}$：立地

ARUMと比較

$$\left\{ \frac{\sum_{i \in [m]} y_i e^{a_{ni}x_i + b_{ni}}}{U_n^c + \sum_{i \in [m]} y_i e^{a_{ni}x_i + b_{ni}}} \right\}$$

指数関数が線形関数に置き換えられたため、
MRUM での最適化問題はより単純な構造となる

1. Introduction

2. Problem formulation

3. Relation between MRUM and ARUM models

4. Solution methods

5. Numerical experiments

6. Conclusion

3. Relation between MRUM and ARUM models

MRUMとARUMの関係について4つの方法で分析し、
ARUMとMRUMの選択確率は近似関係にあることを示す

ARUMでの選択肢jの選択確率

$$\left(\exp\left(a_j x_j + b_i\right)\right) / \left(\exp\left(a_j x_j + b_j\right) + c\right)$$

コスト：$a_j x_j$　選択パラメータ：b_i　他の選択肢の効用：c

MRUMでの選択肢jの選択確率

$$\left(\left(a'_j x_j + b'_i\right)\right) / \left(\left(a'_j x_j + b'_j\right) + c'\right)$$

コスト：$a'_j x_j$　選択パラメータ：b'_i　他の選択肢の効用：c'

3. Relation between MRUM and ARUM models

分析1

ARUMとMRUMの選択確率について、
無作為にパラメータ$(a_j, b_j, c, a_j', b_j', c')$を選定し、コスト$x_j$を変化させる

⬇その結果をプロットした図

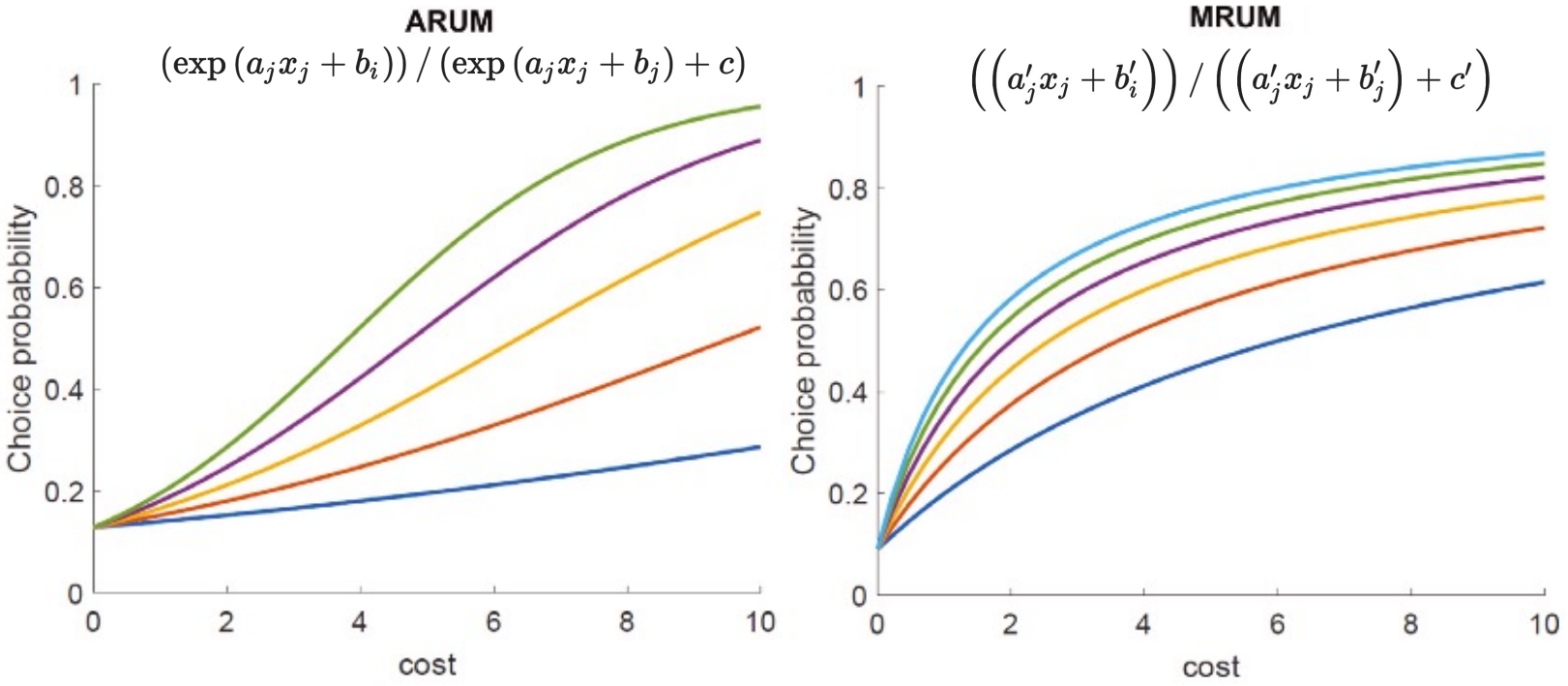


Fig. 1. Choice probability functions under ARUM and MRUM.

MRUMの選択確率関数は凹形状であるが、ARUMの選択確率関数は凹形状ではないことがわかる
MRUMでは、ARUMでの選択確率関数と比較して、x_jが小さいときには速く増加するが、x_jが大きくなると遅くなる

3. Relation between MRUM and ARUM models

分析2 $\left(\exp\left(a_j x_j + b_i\right)\right) / \left(\exp\left(a_j x_j + b_j\right) + c\right)$

⬆の式でx_jを変化させ、コストとそのときの確率のペアを記録する $\left\{\left(x_j^1, p_j^1\right), \left(x_j^2, p_j^2\right) \ldots\right\}$

それらのデータポイントを最小二乗法を用いてMRUM選択確率関数に適合させる（下図左側）
MRUMによって生成されたデータポイントをARUM選択確率関数に適合させる場合も同様である（下図右側）

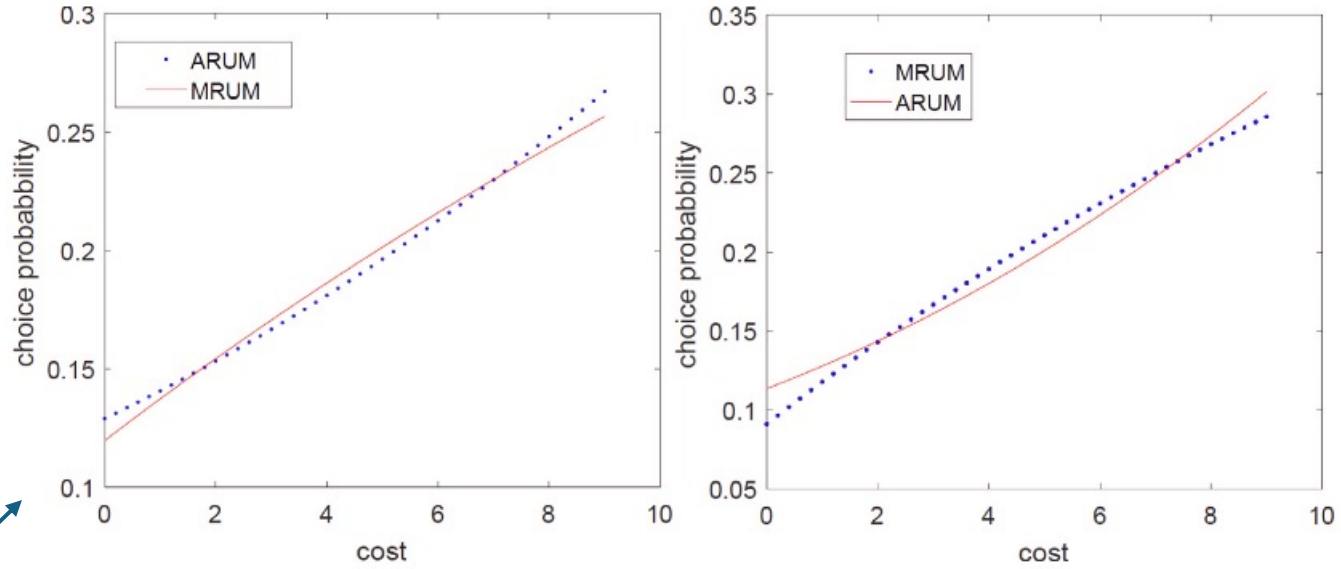


Fig. 2. Fitting choice probability functions yielded by the two frameworks in two-dimensional space.

関数$(\exp(0.1x_j - 0.3))/(\exp(0.1x_j - 0.3) + 5)$を使ってデータ点を生成
適合後に$((0.14x_j + 0.82))/((0.14x_j + 0.82) + 6.02)$を得た
RMSE(Root Mean Square Error)は0.0049(約2%)だった

関数$(x_j + 3))/(((x_j + 3)) + 30)$を使用してデータ点を生成
適合後に$(\exp(0.14x_j + 0.1))/(\exp(0.14x_j + 0.1) + 8.67)$を得た
RMSEは0.0092(約4%)となった

3. Relation between MRUM and ARUM models

分析3

分析2を3D空間に拡張し、MCP目的関数の高次元での近似について確認する
データポイント生成のために、x_i, x_jの2つのコスト変数に応じてMCP目的関数を選択

データポイント（**青い点**）とフィッティングされた関数（**表面**）を図に示す
この実験では、ARUM と MRUM での MCP 目的関数が 3D 空間で互いによく近似できることを示している

ARUM→MRUM

点：$\frac{\exp(0.1x_i-0.3)+\exp(0.2x_j-0.3)}{\exp(0.1x_i-0.3)+\exp(0.2x_j-0.3)+5}$

表面：$\frac{\left((0.15x_i-0.24)+(0.25x_j-0.24)\right)}{\left((0.15x_i-0.24)+(0.25x_j-0.24)+6.25\right)}$

RMSE：0.0044

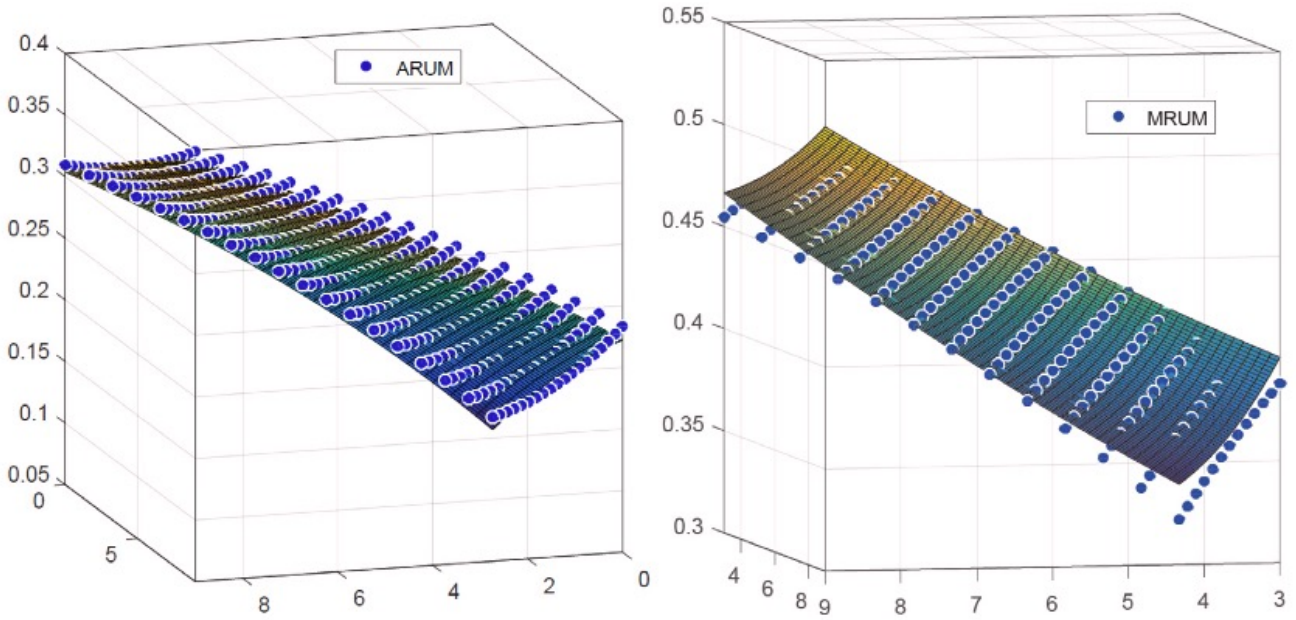


Fig. 3. Fitting choice probability functions yielded by the two frameworks in 3D space.

MRUM→ARUM

点：$\frac{\left((x_i+3)+(2x_j+1)\right)}{\left((x_i+3)+(2x_j+1)+30\right)}$

表面：$\frac{\exp(0.16x_i-0.36)+\exp(0.13x_j+0.71)}{\exp(0.16x_i-0.36)+\exp(0.13x_j+0.71)+5}$

RMSE：0.0054

3. Relation between MRUM and ARUM models

分析4　MRUMがMCPの文脈でARUMを近似できるかどうかをさらに調べる

ARUMモデル　$u_j = \beta_C C_j + \beta_D D_j$

j：選択場所
C_j：コスト
D_j：距離

◎選択場所集合の数を$m \in \{20,40,\ldots,200\}$で変化させる

◎それぞれの選択について ランダムかつ均一に100セットのコストと距離の要素を生成する

$$\{(D_j, C_j)\}\ where\ C_j \in [3,9]\ and\ D_j \in [2,7]$$

◎実データの選択パラメータ➡ $\beta^*{}_C = 0.2\ and\ \beta^*{}_D = -0.3$

各選択の要素$\{(C_j, D_j), j \in [m]\}$と実パラメータ$(\beta^*{}_C, \beta^*{}_D)$から1000個の観測値を生成

次に、同じ効用仕様のMRUMモデルを指定し、
生成された選択観測値を使用して最大尤度推定によってそのパラメータを推定する

ARUMモデルと推定されたMRUMモデルを比較するために、1000個コストベクトル量xのサンプルをランダムかつ均一に生成し、 2つの選択モデルにおけるMCP目的関数間のパーセンテージギャップを計算する

$$\frac{\left|f^{\mathrm{ARUM}}(\mathbf{x}) - f^{\mathrm{MRUM}}(\mathbf{x})\right|}{f^{\mathrm{ARUM}}(\mathbf{x})} \times 100\%$$

3. Relation between MRUM and ARUM models

分析4

$$\max_{\mathbf{y},\mathbf{x}}\left\{ f^{\mathrm{ARUM}}(\mathbf{y},\mathbf{x}) = \sum_{n\in I} q_n \frac{\sum_{i\in[m]} y_i e^{a_{ni}x_i+b_{ni}}}{U_n^c+\sum_{i\in[m]} y_i e^{a_{ni}x_i+b_{ni}}} \right\}$$

$$\max_{\mathbf{y},\mathbf{x}}\left\{ f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}) = \sum_{n\in I} q_n \frac{\sum_{i\in[m]} y_i (a_{ni}x_i+b_{ni})}{U_n^c+\sum_{i\in[m]} y_i (a_{ni}x_i+b_{ni})} \right\}$$

yを除外した形

$$\frac{\left| f^{\mathrm{ARUM}}(\mathbf{x}) - f^{\mathrm{MRUM}}(\mathbf{x}) \right|}{f^{\mathrm{ARUM}}(\mathbf{x})} \times 100\%$$

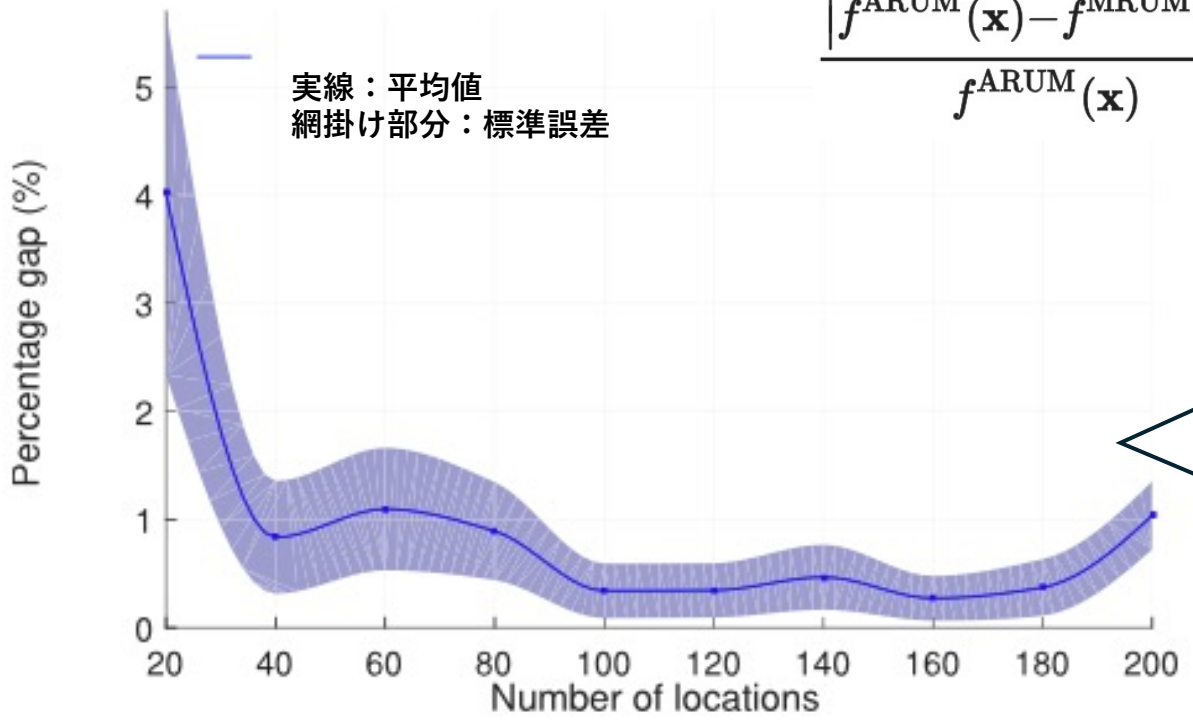


Fig. 4. Percentage gaps between the objective values of the MCP under the MRUM and fitted ARUM models.

パーセンテージの差はm=20のときに4%
その他の選択肢では約1%であることがわかる

観測値の選択モデルがARUMの場合でも、
MRUMモデルで十分に近似できることを示している

1. Introduction

2. Problem formulation

3. Relation between MRUM and ARUM models

4. Solution methods

5. Numerical experiments

6. Conclusion

4. Solution methods

MCP-MRUMを3つの方法で解く

①CONIC reformulation（円錐再定式化）
②Multicut outer-approximation（マルチカット外部近似）
③Local search heuristic（ローカルヒューリスティック）

MCP-MRUM

$$\max_{\mathbf{y},\mathbf{x}}\left\{ f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}) = \sum_{n\in I} q_n \frac{\sum_{i\in[m]} y_i(a_{ni}x_i+b_{ni})}{U_n^c+\sum_{i\in[m]} y_i(a_{ni}x_i+b_{ni})} \right\}$$

どの$i \in [m]$についても、y_i =0のとき$x_i = y_i x_i$ =0, $y_i = 1$のとき$x_i y_i = x_i$
つまり、いずれの場合も $x_i y_i = x_i$であるとして、双線形項$y_i x_i$をx_iに置き換えて簡略化する

MCP-MRUM-2

$$\max_{\mathbf{y},\mathbf{x}}\left\{ f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}) = \sum_{n\in I} q_n \frac{\sum_{i\in[m]} a_{ni}x_i+b_{ni}y_i}{U_n^c+\sum_{i\in[m]} a_{ni}x_i+b_{ni}y_i} \right\}$$

4. Solution methods

①CONIC reformulation（円錐再定式化）

円錐最適化（二次錐計画問題）とは、次の形式の円錐二次不等式上の線形関数の最適化を指す

$$\|\mathbf{Az} - \mathbf{b}\| \leq \mathbf{cx} + \mathbf{d}$$

A, b, c, dは適切なサイズの行列/ベクトルを表す
円錐二次最適化問題は、CPLEXやGUROBIなどの市販の最適化ツールで使用できる

このような回転した2次円錐/双曲不等式が
$z_1^2 \leq z_2 z_3 \quad z_1, z_2, z_3 \geq 0$ のとき、
円錐二次不等式$\|(2z_1, z_2 - z_3)\| \leq z_2 + z_3$として再定式化できる

円錐再定式化のための設定

$$\theta_n = \frac{1}{(U_n^c + \sum_{i \in S} b_{ni}) + \sum_{i \in S} a_{ni} x_i}$$

$$w_n = \left(U_n^c + \sum_{i \in S} b_{ni}\right) + \sum_{i \in S} a_{ni} x_i$$

MRUMでのMCPの円錐再定式化

$$\begin{aligned}
\max_{\mathbf{x},\mathbf{w},\boldsymbol{\theta}} \quad & \textstyle\sum_{n \in I} q_n - \sum_{n \in I} q_n U_n^c \theta_n \\
\text{subject to} \quad & w_n = \left(U_n^c + \textstyle\sum_{i \in S} b_{ni}\right) + \sum_{i \in S} a_{ni} x_i, \ \forall n \in I \\
& \theta_n w_n \geq 1, \ \forall n \in I \\
& \textstyle\sum_{i \in S} x_i \leq C \\
& x_i \in [L_i, U_i], \ \forall i \in S \\
& \mathbf{x} \in \mathbb{R}_+^{|S|}, \ \mathbf{w}, \boldsymbol{\theta} \in \mathbb{R}_+^{|I|}.
\end{aligned}$$

4. Solution methods

①CONIC reformulation（円錐再定式化）

MCP-MRUM-2を**CONICプログラム**として表すと

$$\theta_n = \frac{1}{U_n^c + \sum_{i\in[m]} b_{ni} y_i + \sum_{i\in[m]} a_{ni} x_i}$$

$$w_n = U_n^c + \sum_{i\in[m]} b_{ni} y_i + \sum_{i\in[m]} a_{ni} x_i$$

目的関数を表すと、

$$f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}) = \sum_{n\in I} q_n \frac{\sum_{i\in[m]} a_{ni}x_i + b_{ni}y_i}{U_n^c + \sum_{i\in[m]} a_{ni}x_i + b_{ni}y_i}$$

$$= \sum_{n\in I} q_n \left(1 - \frac{U_n^c}{U_n^c + \sum_{i\in[m]} a_{ni}x_i + b_{ni}y_i}\right)$$

$$= \sum_{n\in I} q_n - \sum_{n\in I}(q_n U_n^c \theta_n).$$

f^{MRUM}の最大化➡この部分の最小化

$$\sum_{n\in I}(q_n U_n^c \theta_n).$$

MCP-MRUM-2

$$\max_{\mathbf{y},\mathbf{x}} \left\{ f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}) = \sum_{n\in I} q_n \frac{\sum_{i\in[m]} a_{ni}x_i + b_{ni}y_i}{U_n^c + \sum_{i\in[m]} a_{ni}x_i + b_{ni}y_i} \right\}$$

円錐再定式化のための設定

$$\theta_n = \frac{1}{(U_n^c + \sum_{i\in S} b_{ni}) + \sum_{i\in S} a_{ni}x_i}$$

$$w_n = \left(U_n^c + \sum_{i\in S} b_{ni}\right) + \sum_{i\in S} a_{ni}x_i$$

MRUMでのMCPの円錐再定式化

$$\max_{\mathbf{x},\mathbf{w},\boldsymbol{\theta}} \quad \sum_{n\in I} q_n - \sum_{n\in I} q_n U_n^c \theta_n$$

$$\text{subject to} \quad w_n = \left(U_n^c + \sum_{i\in S} b_{ni}\right) + \sum_{i\in S} a_{ni}x_i, \ \forall n \in I$$

$$\theta_n w_n \geq 1, \ \forall n \in I$$

$$\sum_{i\in S} x_i \leq C$$

$$x_i \in [L_i, U_i], \ \forall i \in S$$

$$\mathbf{x} \in \mathbb{R}_+^{|S|}, \ \mathbf{w},\boldsymbol{\theta} \in \mathbb{R}_+^{|I|}.$$

最小化問題に置き換える

$$\min_{\mathbf{x},\mathbf{y},\mathbf{w},\boldsymbol{\theta}} \quad \sum_{n\in I} q_n U_n^c \theta_n$$

$$\text{subject to} \quad \theta_n w_n = 1, \ \forall n \in I$$

$$w_n = U_n^c + \sum_{i\in[m]} b_{ni} y_i + \sum_{i\in[m]} a_{ni} x_i, \ \forall n \in I$$

$$\text{Constraints } (2)\text{–}(3)\text{–}(4)\text{–}(5),$$

$$\mathbf{x} \in \mathbb{R}_+^m, \ \mathbf{y} \in \{0,1\}^m, \quad \mathbf{w},\boldsymbol{\theta} \in \mathbb{R}_+^{|I|}.$$

4. Solution methods

①CONIC reformulation

最小化問題

$$\min_{\mathbf{x},\mathbf{y},\mathbf{w},\boldsymbol{\theta}} \quad \sum_{n\in I} q_n U_n^c \theta_n$$

$$\text{subject to} \quad \theta_n w_n = 1,\ \forall n \in I$$

$$w_n = U_n^c + \sum_{i\in[m]} b_{ni} y_i + \sum_{i\in[m]} a_{ni} x_i,\ \forall n \in I$$

θ_nとω_nは非負であり
$\theta_n\omega_n = 1$を$\theta_n\omega_n \geq 1$に置き換えることができる

CONIC reformulation

$$\min_{\mathbf{x},\mathbf{y},\mathbf{w},\boldsymbol{\theta}} \quad \sum_{n\in I} q_n U_n^c \theta_n$$

$$\text{subject to} \quad \theta_n w_n \geq 1,\ \forall n \in I$$

$$w_n = 1 + \sum_{i\in[m]} b_{ni} y_i + \sum_{i\in[m]} a_{ni} x_i,\ \forall n \in I$$

$$\text{Constraints}\ (2)\text{–}(3)\text{–}(4)\text{–}(5),$$

$$\mathbf{x} \in \mathbb{R}_+^m, \mathbf{y} \in \{0,1\}^m, \mathbf{w}, \boldsymbol{\theta} \in \mathbb{R}_+^{|I|}.$$

制約条件(再掲)

$$\text{subject to} \quad \sum_{i\in[m]} x_i \leq C$$

予算制約

$$\sum_{i\in[m]} y_i \leq K$$

施設数制約

$$x_i \leq y_i U_i,\ \forall i \in [m]$$

$$x_i \geq y_i L_i,\ \forall i \in [m]$$

施設費用
上限下限制約

$$\mathbf{x} \in \mathbb{R}_+^m,\ \mathbf{y} \in \{0,1\}^m.$$

コスト　立地

4. Solution methods

②Multicut outer-approximation（MOA, マルチカット外部近似）

MOA,マルチカット外部近似

非線形の目的関数を多数のサブ関数に分割（カット）して線形部を作る

MOAスキーム（Duran and Grossmann、1986、Mai and Lodi、2020）を適用するために、
顧客ゾーンの集合Iを$\mathrm{U}_{l\in\mathcal{S}}\mathcal{D}_l = I$であるような$\mathcal{S}$（グループ$\mathcal{D}_1,\ldots,\mathcal{D}_{\mathcal{S}}$）に分割する

それを踏まえてMCP-MRUM-2を以下のように書く

$$f(\mathbf{x},\mathbf{y}) = \sum_{l\in[\mathscr{L}]} \phi_l(\mathbf{y},\mathbf{x})$$

$$\phi_l(\mathbf{y},\mathbf{x}) = \sum_{n\in\mathscr{D}_l}\left(q_n - \frac{q_n U_n^c}{U_n^c+\sum_{i\in[m]} a_{ni}x_i+y_i b_{ni}}\right)$$

MCP-MRUM-2

$$\max_{\mathbf{y},\mathbf{x}}\left\{f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}) = \sum_{n\in I} q_n \frac{\sum_{i\in[m]} a_{ni}x_i+b_{ni}y_i}{U_n^c+\sum_{i\in[m]} a_{ni}x_i+b_{ni}y_i}\right\}$$

各凹関数$\phi_l(y,x)$を外部近似するアルゴリズムは、以下のサブ勾配カットを作成することで構築できる

$$\theta_l \le \nabla_{\mathbf{x}}\phi_l(\mathbf{y},\mathbf{x})(\mathbf{x}-\bar{\mathbf{x}}) + \nabla_{\mathbf{y}}\phi_l(\mathbf{y},\mathbf{x})(\mathbf{y}-\bar{\mathbf{y}}) + \phi_l(\bar{\mathbf{y}},\bar{\mathbf{x}})$$

$(\bar{y},\bar{x})$：試行した時点での解の候補
$\nabla_{\mathrm{y}}\phi_l(y,x)$と$\nabla_{\mathrm{x}}\phi_l(y,x)$：$y,x$に関しての$\phi_l(\cdot)$の勾配

MOAは、各ステップで次のマスター問題にサブ勾配カットを追加する反復手順を実行する

4. Solution methods

Multicut outer-approximation

$$\max_{\mathbf{x},\mathbf{y},\mathbf{w},\boldsymbol{\theta}} \quad \sum_{l\in[\mathscr{L}]} \theta_l$$

$$\text{subject to} \quad \theta_l \leq \nabla_{\mathbf{x}}\phi_l\left(\mathbf{y},\mathbf{x}\right)\left(\mathbf{x}-\overline{\mathbf{x}}^t\right) + \nabla_{\mathbf{y}}\phi_l\left(\mathbf{y},\mathbf{x}\right)\left(\mathbf{y}-\overline{\mathbf{y}}^t\right)$$

$$+\phi_l\left(\overline{\mathbf{y}}^t,\overline{\mathbf{x}}^t\right), \qquad l\in[\mathscr{L}], t=1,2\ldots$$

$$\text{Constraints (2)–(3)–(4)–(5)}, \quad \mathbf{x}\in\mathbb{R}_+^m, \mathbf{y}\in\{0,1\}^m, \boldsymbol{\theta}\in\mathbb{R}_+^{\mathscr{L}},$$

$(\bar{y}^t, \bar{x}^t)$：t回目の計算時点での解の候補

制約条件(再掲)

$$\text{subject to} \quad \sum_{i\in[m]} x_i \leq C$$

$$\sum_{i\in[m]} y_i \leq K$$

$$x_i \leq y_i U_i, \ \forall i\in[m]$$

$$x_i \geq y_i L_i, \ \forall i\in[m]$$

$$\mathbf{x}\in\mathbb{R}_+^m, \ \mathbf{y}\in\{0,1\}^m.$$

このアルゴリズムは、

$$\sum_{l\in[\mathscr{L}]}\theta_l^* \leq \sum_{l\in[\mathscr{L}]}\phi_l\left(\mathbf{y}^*,\mathbf{x}^*\right)+\tau, \text{ where } \tau>0$$

停止閾値

となるような解(y^*, x^*, θ^*)を得るまで続けられる

$\phi_l(y,x), \forall l\in[\mathcal{S}]$の凹性はMOAが最終的に最適解に収束することを保証する

4. Solution methods

③Local search heuristic

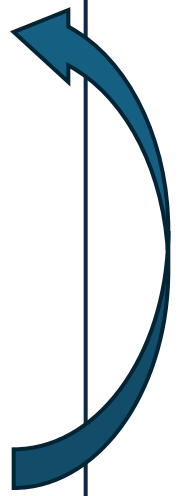
ARUMの下での施設配置MCPにおいて、局所探索がパフォーマンスを発揮することが示されている
MRUMの下での施設とコストの統合最適化問題を解くための局所探索手順を開発する

局所探索アルゴリズム

Step1
空の解から始めて、目的関数が増加する場所を取りながら、現在の解に場所を追加していく
解の候補が容量Kに達したときorより良い解を見つけることができない場合に停止

Step2
勾配ベースの局所探索
$\Phi(y)$の勾配情報は、非線形目的関数f^{MRUM}を近似し、次の解の候補を見つけるために使用される

Step3
現在の選択肢集合内の地点を、Step2で見つかった解を改善する選択肢集合外の地点と交換する



ヒューリスティック解法のための派生関数

$$\Phi(\mathbf{y}) = \max_{\mathbf{x}\in\mathscr{X}(\mathbf{y})} \left\{ f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}) \right\},$$

$$\mathscr{X}(\mathbf{y}) = \left\{ \mathbf{x}\in\mathbb{R}_+^m \middle|\ \textstyle\sum_{i\in[m]} x_i \le C,\ x_i \le y_i U_i,\ x_i \ge y_i L_i,\ \forall i \in [m] \right\}$$

4. Solution methods

③Local search heuristic

派生関数を定義

$$\Phi(\mathbf{y}) = \max_{\mathbf{x}\in\mathscr{X}(\mathbf{y})} \left\{ f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}) \right\},$$

$$\mathscr{X}(\mathbf{y}) = \left\{ \mathbf{x}\in\mathbb{R}_+^m \,\middle|\, \textstyle\sum_{i\in[m]} x_i \le C,\ x_i \le y_i U_i,\ x_i \ge y_i L_i,\ \forall i\in[m] \right\}$$

派生関数の双対問題

$$L\left(\mathbf{x},\lambda,\boldsymbol{\gamma}^U,\boldsymbol{\gamma}^L\middle|\mathbf{y}\right) = f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}) - \lambda\left(\textstyle\sum_{i\in[m]} x_i - C\right) - \sum_{i\in[m]} \gamma_i^U (x_i - y_i U_i) + \sum_{i\in[m]} \gamma_i^L (x_i - y_i L_i).$$

$$\mathbf{y}\in\mathscr{Y} \qquad \mathscr{Y} = \left\{ \mathbf{y}\in\{0,1\}^m \,\middle|\, \textstyle\sum_{i\in[m]} y_i \le K \right\}$$ ← 容量K

鞍点$L(x,\lambda,\gamma^U,\gamma^L|y)$➡ $x^*(y),\lambda^*(y),\gamma_i^{U*}(y),\gamma_i^{L*}(y),\forall i\in[m]$

$\Phi(y)$の勾配情報

$$\frac{\partial\Phi(\mathbf{y})}{\partial y_i} = \frac{\partial f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}^*)}{\partial y_i} + \gamma_i^{U*}(\mathbf{y})\,U_i - \gamma_i^{L*}(\mathbf{y})\,L_i,\ \forall i\in[m]$$

4. Solution methods

③Local search heuristic

ラグランジュ乗数 $\lambda^*, \gamma^{U*}, \gamma^{L*}$ は

ラグランジュ双対問題 $\min_{\lambda,\gamma^U,\gamma^L} \max_{\mathbf{x}} L\left(\mathbf{x}, \lambda, \boldsymbol{\gamma}^U, \boldsymbol{\gamma}^L \middle| \mathbf{y}\right)$ を解くことで求まる

既存の最適化ソルバーで計算し、得られた最適解からラグランジュ乗数を推定することで

凸問題 $\max_{\mathbf{x}\in\mathscr{X}(\mathbf{y})} f^{\mathrm{MRUM}}(\mathbf{y}, \mathbf{x})$ を直接解くこともできる

つまり、x^*が唯一の最適解となる $\frac{\partial f^{\mathrm{MRUM}}(\mathbf{y},\mathbf{x}^*)}{\partial x_i} - \lambda^* - \gamma_i^{U*} + \gamma_i^{L*} = 0,\ \forall i \in [m]$

Step1
空の解から始めて、目的関数が増加する場所を取りながら、現在の解に場所を追加していく
解の候補が容量Kに達したときorより良い解を見つけることができない場合に停止

Step2
勾配ベースの局所探索
$\Phi(y)$の勾配情報は、非線形目的関数f^{MRUM}を近似し、次の解の候補を見つけるために使用される

Step3
現在の選択肢集合内の地点を、Step2で見つかった解を改善する選択肢集合外の地点と交換する

本研究においてこの局所探索手順は高価であり、
円錐再定式化と外部近似アルゴリズムほど性能が高くない

1. Introduction

2. Problem formulation

3. Relation between MRUM and ARUM models

4. Solution methods

5. Numerical experiments

6. Conclusion

5. Numerical experiments

Experimental settings

提案したアルゴリズムを評価するために、各問題（コストMCP、統合MCP）に対してデータセットをランダムに生成
コスト最適化問題のデータセットには合計240のインスタンスがあり、各ペアに対して24のインスタンスが生成される

$(|I|, m)$ （顧客ゾーンの集合、利用可能場所）
{(100, 100), (100, 1000), (100, 3000), (200, 2000), (400, 1000), (800, 400), (800, 800), (1000, 1000), (2000, 2000), (5000, 1000)}

※立地＆コスト最適化問題のデータセットでは、ペア (2000, 2000) は除外
　このサイズのインスタンスはコストが高すぎて、どのアプローチでも時間予算内で解決できないため

数値設定

$C \in \{0.2m, 0.5m, 0.7m\}$　　$n \in I$
$K \in \{0.2m, 0.5m, 0.7m\}$　　$i \in [m]$
a_{ni} $from$ $[0.5,1.5]$の間からランダムかつ均一に生成
b_{ni} $from$ $[1,10]$の間からランダムかつ均一に生成
q_n:$[1,10][90,100]$の間からランダムかつ均一に生成
$L_i, U_i, \forall i \in [m]: \sum_{i\in[m]} L_i \leq C$を満たすように均一に生成

notation
a_{ni}：顧客ゾーンnの施設iに費やされたコストに対する感度
b_{ni}：顧客の選択に影響する他の要素
C：新施設への予算の最大値
K：開設される施設数の最大値
L_i, U_i：施設iにかける予算の下限と上限
q_n：それぞれのゾーンnにいる顧客の数
m：新施設を配置可能な場所の合計

実験環境
プロセッサ AMD Ryzen 3-3100 CPU @ 3.60 GHz、RAM 24 GB、Windows 10 を搭載した PC

各インスタンスの CPU 時間制限は 600 秒
アルゴリズムが時間予算を超えると停止し、最適解を記録する

5. Numerical experiments

Comparison results for the cost optimization MCP

CONICプログラムを CPLEXで解くことで、生成インスタンスを解く
2つのアプローチを CONIC(CPLEX)と CONVEX(fmincon)と表記

CONICは、30のグループすべてで全インスタンスに対して最適目的関数を提供
CONVEXは、30ののうち24で最適目的関数を提供
一方、CONICに必要な平均CPU時間は、CONVEXにかかる時間よりも大幅に短い

中規模および大規模のインスタンス（m>400)のとき、
CONICアプローチはCONVEXよりも約5～30倍高速

$(|I|, m) = (100,1000)$のとき、
CONICはCONVEXより約15倍、m＝3000のとき約30倍高速

Table 2
Comparison results for the cost optimization MCP instances, 8 instances per row.

$\|I\|$	m	C	# Instances with best objective		Average CPU time (s)	
			CONIC	CONVEX	CONIC	CONVEX
100	100	20	**8**	**8**	0.05	0.11
100	100	50	**8**	**8**	0.05	0.09
100	100	70	**8**	**8**	0.05	0.10
100	1000	200	**8**	**8**	0.41	6.10
100	1000	500	**8**	**8**	0.42	7.36
100	1000	700	**8**	**8**	0.43	6.96
800	400	80	**8**	**8**	1.84	5.48
800	400	200	**8**	**8**	1.85	6.10
800	400	280	**8**	**8**	1.79	6.32
800	800	160	**8**	**8**	3.77	50.48
800	800	400	**8**	**8**	3.76	52.97
800	800	560	**8**	**8**	3.77	57.25
400	1000	200	**8**	**8**	1.55	22.72
400	1000	500	**8**	**8**	1.53	23.69
400	1000	700	**8**	**8**	1.56	24.31
1000	1000	200	**8**	**8**	7.30	112.53
1000	1000	500	**8**	**8**	7.18	121.48
1000	1000	700	**8**	**8**	7.44	129.71
200	2000	400	**8**	**8**	1.85	55.35
200	2000	1000	**8**	**8**	1.89	55.54
200	2000	1400	**8**	**8**	1.85	50.73
100	3000	600	**8**	**8**	1.84	60.93
100	3000	1500	**8**	**8**	1.88	63.72
100	3000	2100	**8**	**8**	1.90	49.79
5000	1000	200	**8**	4	136.08	579.13
5000	1000	500	**8**	2	116.72	588.30
5000	1000	700	**8**	0	114.81	600.00
2000	2000	400	**8**	0	48.40	600.00
2000	2000	1000	**8**	0	46.44	600.00
2000	2000	1400	**8**	0	47.55	600.00
Average			**8**	7.1		

5. Numerical experiments

Joint location and cost optimization MCP

生成インスタンスは216個

円錐再定式化（CN）、マルチカット外部近似アルゴリズム（M1,M5）、および局所探索（LS）を比較

MOAで重要なパラメータ
➡各計算で加えるカットの数 $T \in \{1,5\}$

最適解（solved to optimality）に着目すると
CONIC は約 57.0% で最適解を提供
MOAは 99.7%で最適解を提供

最適目的関数(best objectives)に着目すると
平均してCONIC は約 98.6% のインスタンスで最適な値を返すという点で優れたパフォーマンスを発揮

これは、CONIC が時間予算を超えており、導いた最適解が最適であることを確認できないため

LSは、他のアプローチと比較して性能が悪く、25.25%でのみ最適解を示す

Table 3
Comparison results for the joint location and cost optimization problem, for small-sized and medium-sized instances grouped by $(m, |I|, C, K)$ (8 instances per row); CN stands for the CONIC approach and M1, M5 stand for the MOA algorithms with $T = 1, 5$, respectively.

m	$\|I\|$	C	K	# Instances solved to optimality			# Instances with best objectives			Average CPU time (s)		
				CN	M1	M5	CN	M1/M5	LS	CN	M1	M5
100	100	20	20	**8**	**8**	**8**	**8**	**8**	**8**	1.3	**0.2**	0.4
100	100	20	50	**8**	**8**	**8**	**8**	**8**	**8**	1.7	**0.1**	**0.1**
100	100	20	70	**8**	**8**	**8**	**8**	**8**	7	1.4	**0.0**	0.1
100	100	50	20	7	**8**	**8**	**8**	**8**	**8**	80.8	**1.0**	1.1
100	100	50	50	**8**	**8**	**8**	**8**	**8**	**8**	3.1	**0.3**	0.4
100	100	50	70	**8**	**8**	**8**	**8**	**8**	**8**	1.2	**0.0**	0.1
100	100	70	20	7	**8**	**8**	**8**	**8**	**8**	78.2	1.2	**0.8**
100	100	70	50	**8**	**8**	**8**	**8**	**8**	**8**	0.8	**0.1**	0.3
100	100	70	70	**8**	**8**	**8**	**8**	**8**	**8**	0.9	**0.0**	0.1
400	800	80	80	0	**8**	**8**	**8**	**8**	0	600.0	**36.1**	40.8
400	800	80	200	1	**8**	**8**	**8**	**8**	0	530.1	**2.5**	5.7
400	800	80	280	1	**8**	**8**	**8**	**8**	0	543.4	**0.4**	0.7
400	800	200	80	0	**8**	**8**	**8**	**8**	0	600.0	**98.5**	102.3
400	800	200	200	5	**8**	**8**	7	**8**	0	343.6	**8.4**	10.7
400	800	200	280	2	**8**	**8**	7	**8**	0	498.6	**0.6**	**0.6**
400	800	280	80	0	**8**	**8**	**8**	**8**	0	600.0	**99.8**	126.1
400	800	280	200	7	**8**	**8**	**8**	**8**	0	172.0	**13.3**	17.8
400	800	280	280	6	**8**	**8**	**8**	**8**	0	318.3	**0.7**	2.2
800	800	160	160	0	**8**	**8**	**8**	**8**	0	600.0	**123.6**	147.6
800	800	160	400	3	**8**	**8**	**8**	**8**	0	448.0	**8.9**	12.9
800	800	160	560	3	**8**	**8**	**8**	**8**	0	460.1	**1.6**	1.9
800	800	400	160	0	6	**8**	**8**	**8**	0	600.0	349.2	**272.8**
800	800	400	400	2	**8**	**8**	**8**	**8**	0	479.7	**22.4**	30.0
800	800	400	560	6	**8**	**8**	**8**	**8**	0	201.0	**4.8**	6.7
800	800	560	160	0	6	**8**	**8**	**8**	0	600.0	337.8	**288.0**
800	800	560	400	1	**8**	**8**	**8**	**8**	0	537.2	**21.4**	35.9
800	800	560	560	7	**8**	**8**	**8**	**8**	0	163.5	**13.9**	35.2
1000	100	200	200	1	**8**	**8**	**8**	**8**	0	525.5	9.3	**8.8**
1000	100	200	500	7	**8**	**8**	**8**	**8**	0	132.0	**1.2**	2.7
1000	100	200	700	**8**	**8**	**8**	**8**	**8**	0	3.9	**0.8**	1.5
1000	100	500	200	6	**8**	**8**	**8**	**8**	4	208.5	19.1	**15.1**
1000	100	500	500	6	**8**	**8**	6	**8**	0	153.2	**1.8**	5.6
1000	100	500	700	**8**	**8**	**8**	**8**	**8**	0	3.4	**1.0**	3.1
1000	100	700	200	7	**8**	**8**	**8**	**8**	4	100.9	20.2	**20.1**
1000	100	700	500	**8**	**8**	**8**	**8**	**8**	4	20.0	**2.4**	5.1
1000	100	700	700	**8**	**8**	**8**	**8**	**8**	0	6.2	2.0	**1.7**
1000	400	200	200	0	**8**	**8**	**8**	**8**	0	600.0	**45.6**	68.5
1000	400	200	500	4	**8**	**8**	**8**	**8**	0	338.2	**3.7**	6.9
1000	400	200	700	5	**8**	**8**	**8**	**8**	0	296.1	**1.3**	2.8
1000	400	500	200	0	**8**	**8**	**8**	**8**	4	600.0	123.9	**82.6**
1000	400	500	500	5	**8**	**8**	**8**	**8**	0	248.0	**6.7**	11.3
1000	400	500	700	4	**8**	**8**	7	**8**	0	327.6	**5.3**	6.1
1000	400	700	200	3	**8**	**8**	**8**	**8**	4	406.2	180.0	**152.6**
1000	400	700	500	5	**8**	**8**	**8**	**8**	0	357.3	**7.9**	15.3
1000	400	700	700	6	**8**	**8**	**8**	**8**	0	203.8	11.5	**3.0**
Average				4.56	7.91	8	7.89	8	2.02			

5. Numerical experiments

平均 CPU 時間に着目すると
すべてのインスタンスの平均 CPU 時間に関して、MOA が他のアプローチ より優れていることもわかる

各アプローチで必要な平均 CPU 時間が最も短いグループの数で、M1 (シングルカット外部近似) が他のアプローチを圧倒し、M5 がそれに続く

CONIC は一部に対してのみパフォーマンスを発揮し、LS はほとんどの場合で時間予算を超え、常に最も遅いアプローチになる

LSの性能は悪い
LS の各ステップでは、多数のコスト最適化問題を解決する必要があるため、立地選択のみを考慮する場合よりもはるかにコストがかかる

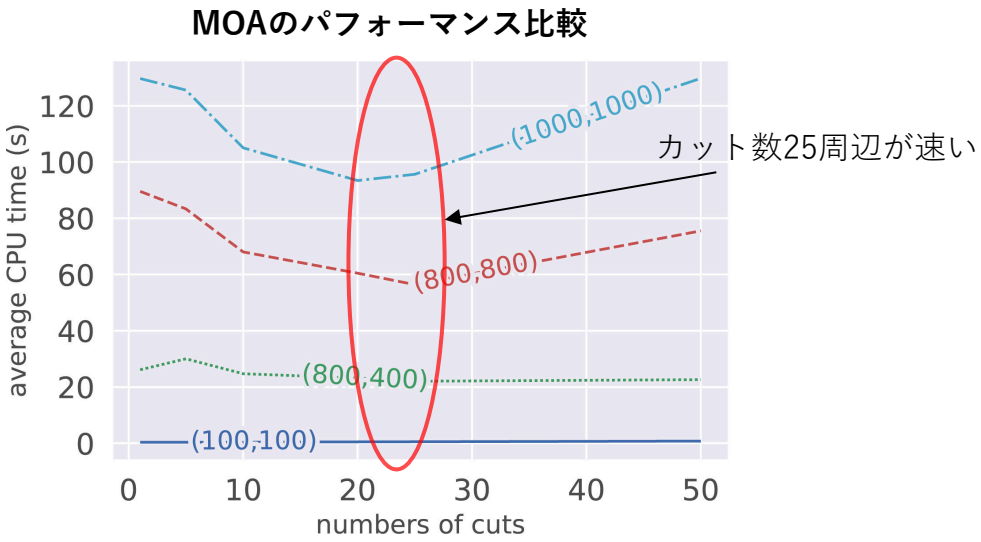
MOAアルゴリズムは、立地＆コスト最適化で最高のパフォーマンスを発揮し、そのパフォーマンスは各反復でマスター問題に追加されるカットの数（T）に依存するため、追加の検証でその特性を確認する

2024/7/20

Table 3
Comparison results for the joint location and cost optimization problem, for small-sized and medium-sized instances grouped by $(m, |I|, C, K)$ (8 instances per row); CN stands for the CONIC approach and M1, M5 stand for the MOA algorithms with $T = 1, 5$, respectively.

m	$\|I\|$	C	K	# Instances solved to optimality			# Instances with best objectives			Average CPU time (s)		
				CN	M1	M5	CN	M1/M5	LS	CN	M1	M5
100	100	20	20	**8**	**8**	**8**	**8**	**8**	**8**	1.3	**0.2**	0.4
100	100	20	50	**8**	**8**	**8**	**8**	**8**	**8**	1.7	**0.1**	**0.1**
100	100	20	70	**8**	**8**	**8**	**8**	**8**	7	1.4	**0.0**	0.1
100	100	50	20	7	**8**	**8**	**8**	**8**	**8**	80.8	**1.0**	1.1
100	100	50	50	**8**	**8**	**8**	**8**	**8**	**8**	3.1	**0.3**	0.4
100	100	50	70	**8**	**8**	**8**	**8**	**8**	**8**	1.2	**0.0**	0.1
100	100	70	20	7	**8**	**8**	**8**	**8**	**8**	78.2	1.2	**0.8**
100	100	70	50	**8**	**8**	**8**	**8**	**8**	**8**	0.8	**0.1**	0.3
100	100	70	70	**8**	**8**	**8**	**8**	**8**	**8**	0.9	**0.0**	0.1
400	800	80	80	0	**8**	**8**	**8**	**8**	0	600.0	**36.1**	40.8
400	800	80	200	1	**8**	**8**	**8**	**8**	0	530.1	**2.5**	5.7
400	800	80	280	1	**8**	**8**	**8**	**8**	0	543.4	**0.4**	0.7
400	800	200	80	0	**8**	**8**	**8**	**8**	0	600.0	**98.5**	102.3
400	800	200	200	5	**8**	**8**	7	**8**	0	343.6	**8.4**	10.7
400	800	200	280	2	**8**	**8**	7	**8**	0	498.6	**0.6**	**0.6**
400	800	280	80	0	**8**	**8**	**8**	**8**	0	600.0	**99.8**	126.1
400	800	280	200	7	**8**	**8**	**8**	**8**	0	172.0	**13.3**	17.8
400	800	280	280	6	**8**	**8**	**8**	**8**	0	318.3	**0.7**	2.2
800	800	160	160	0	**8**	**8**	**8**	**8**	0	600.0	**123.6**	147.6
800	800	160	400	3	**8**	**8**	**8**	**8**	0	448.0	**8.9**	12.9
800	800	160	560	3	**8**	**8**	**8**	**8**	0	460.1	**1.6**	1.9
800	800	400	160	0	6	**8**	**8**	**8**	0	600.0	349.2	**272.8**
800	800	400	400	2	**8**	**8**	**8**	**8**	0	479.7	**22.4**	30.0
800	800	400	560	6	**8**	**8**	**8**	**8**	0	201.0	**4.8**	6.7
800	800	560	160	0	6	**8**	**8**	**8**	0	600.0	337.8	**288.0**
800	800	560	400	1	**8**	**8**	**8**	**8**	0	537.2	**21.4**	35.9
800	800	560	560	7	**8**	**8**	**8**	**8**	0	163.5	**13.9**	35.2
1000	100	200	200	1	**8**	**8**	**8**	**8**	0	525.5	9.3	**8.8**
1000	100	200	500	7	**8**	**8**	**8**	**8**	0	132.0	**1.2**	2.7
1000	100	200	700	**8**	**8**	**8**	**8**	**8**	0	3.9	**0.8**	1.5
1000	100	500	200	6	**8**	**8**	**8**	**8**	4	208.5	19.1	**15.1**
1000	100	500	500	6	**8**	**8**	6	**8**	0	153.2	**1.8**	5.6
1000	100	500	700	**8**	**8**	**8**	**8**	**8**	0	3.4	**1.0**	3.1
1000	100	700	200	7	**8**	**8**	**8**	**8**	4	100.9	20.2	**20.1**
1000	100	700	500	**8**	**8**	**8**	**8**	**8**	4	20.0	**2.4**	5.1
1000	100	700	700	**8**	**8**	**8**	**8**	**8**	0	6.2	2.0	**1.7**
1000	400	200	200	0	**8**	**8**	**8**	**8**	0	600.0	**45.6**	68.5
1000	400	200	500	4	**8**	**8**	**8**	**8**	0	338.2	**3.7**	6.9
1000	400	200	700	5	**8**	**8**	**8**	**8**	0	296.1	**1.3**	2.8
1000	400	500	200	0	**8**	**8**	**8**	**8**	4	600.0	123.9	**82.6**
1000	400	500	500	5	**8**	**8**	**8**	**8**	0	248.0	**6.7**	11.3
1000	400	500	700	4	**8**	**8**	7	**8**	0	327.6	**5.3**	6.1
1000	400	700	200	3	**8**	**8**	**8**	**8**	4	406.2	180.0	**152.6**
1000	400	700	500	5	**8**	**8**	**8**	**8**	0	357.3	**7.9**	15.3
1000	400	700	700	6	**8**	**8**	**8**	**8**	0	203.8	11.5	**3.0**
Average				4.56	7.91	8	7.89	8	2.02			

5. Numerical experiments

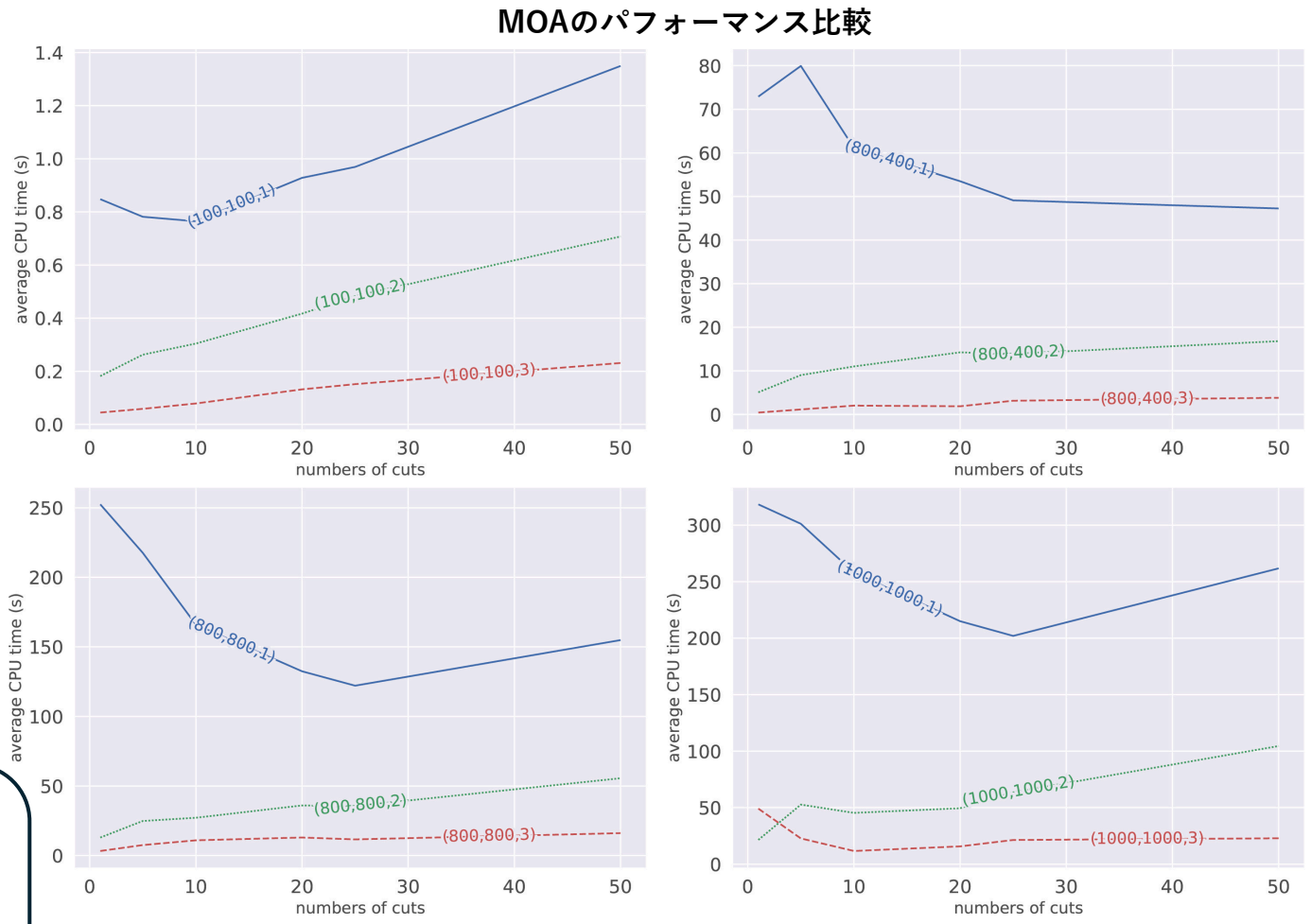


$(|I|, m) \in \{(100,100), (800,400), (800,800), (1000,1000)\}$
(顧客ゾーンの集合、利用可能場所)

$T\ vary\ in\ \{1,5,10,20,25,50\}$
MOAでのカット数

K=0.5m,0.7mのときはT=1~5で速く計算できる
K=0.2mのときT=10~25が最も実行時間が短い
Tが大きすぎると性能ダウン（メモリ不足）

施設数の上限(K)が小さいと時間がかかる（青線）



$(|I|, m) \in \{(100,100), (800,400), (800,800), (1000,1000)\}$

$T\ vary\ in\ \{1,5,10,20,25,50\}$

$K \in \{0.2m, 0.5m, 0.7m\}$　K：開設される施設数の最大値
1 , 2 , 3

1. Introduction
2. Problem formulation
3. Relation between MRUM and ARUM models
4. Solution methods
5. Numerical experiments
6. Conclusion

6. Conclusion

・ランダム効用モデルの下での施設立地とコスト最適化の結合問題

・MCPコスト最適化の文脈において、ARUMとMRUMが互いによく近似できることを示した

・最適化の観点からは、ARUMのコスト最適化問題は非凸であり、扱うのが困難であるのに対し、MRUMのコスト最適化問題は凸最適化ソルバーによって効率的かつ正確に解くことができる

・円錐再定式化とマルチカット外部近似に基づく2つの厳密解法と、局所探索ヒューリスティックを提案し、MRUMの下での位置とコストの合同最適化問題を扱った

・生成したインスタンスに基づく実験を行い、コスト最適化問題に対するCONICアプローチと、統合問題に対するMOAアルゴリズムの効率性を示した

・興味深い将来の方向性は、MRUMによらないGEVや混合ロジットモデルのような他のタイプの選択モデルを検討することであろう

・MRUMの利点に関する発見は、品揃えと価格の最適化（Wang, 2012）やセキュリティ・ゲーム問題（Yang et al., 2012）のような他の選択ベースの意思決定問題におけるMRUMフレームワークの使用を探求することが有望であることを示唆している

所感

・対象とする問題（立地だけでなく施設の質も含めたMCP）は興味深く、応用性もありそう。
　　・顧客ゾーンの分類など、深掘りできるところが多い。

・解法の近似アルゴリズムが難解で苦労した。

・実験を生成データで行うので、想定する顧客分布や立地点がイメージしにくかった。
　　・性能評価が目的なので仕方ない