

オークションを用いたクラウド・ソーシング配送システム

渡邊 大樹^{*1} 武田 翼^{*2} 赤松 隆^{*1}

東北大学 大学院情報科学研究科人間社会情報科学専攻^{*1}

東北大学 工学部建築・社会環境工学科^{*2}

クラウドソーシング・デリバリー (CSD) とは、個人ドライバーが予定していたトリップと同時に配送を行う配送システムである。CSD は配送を効率化できる一方、多数のドライバーと配送業務をマッチングする膨大な計算を要する。また、マッチングを最適化するためにはドライバーの選好を管理者が知る必要があるが、これは一般的には難しい。本研究では、これらの課題を解決した CSD マッチングシステムを提案する。まず、オークションをマッチングに導入することで、ドライバーの選好を反映した最適マッチングを実現する。続いて、このオークションに適用する計算効率的メカニズムを提案する。数値実験により、提案メカニズムは計算効率性を大幅に改善することを示した。

An Auction-based Matching System for Crowdsourced Delivery

Taiki Watanabe^{*1} Tsubasa Takeda^{*2} Takashi Akamatsu^{*1}

Graduate School of Information Sciences, Tohoku University^{*1}

School of Engineering, Tohoku University^{*2}

Abstract

Crowdsourced delivery (CSD) is a last mile delivery system in which drivers of private cars make deliveries along the way of their trips. While the CSD system reduces delivery cost, it is required to solve the unsolvable large-scale optimization problem of matching many drivers with many delivery tasks. On the other hand, the service provider needs preference information of drivers to determine matching, which is generally unavailable. We propose a matching system for CSD which address these problems. In the proposed system, drivers and delivery tasks are matched by an auction, which achieves the socially optimal matching reflecting preferences of drivers. Furthermore, we propose an computationally efficient mechanism of this auction. Numerical experiments have shown that the proposed mechanism improves the efficiency of calculations in the order of several digits.

Keyword: Crowdsourced delivery, last-mile delivery, auction

1. はじめに

“クラウドソーシングデリバリー (CSD)” は、都市内をトリップする予定の車両ドライバーが、依頼された配送をトリップ時に回り道をして行うシステムである。CSD は、多数の車両で少量ずつ荷物を運ぶことができるため、小口・多頻度な需要との親和性が高い。現在、EC の発達により小口・多頻化する配送需要に配送システムが対応しきれないことが問題となっているが、CSD はこのような性質からこの問題を解決できる。

CSD では、ドライバーとタスク所有者を適切にマッチングする必要がある。しかしその際に、2つの問題点が生じる。一つ目は参加主体の選好情報を管理者が得られず、これをマッチングに反映できないという点である。本研究ではマッチングにオークションを導入することで、入札額として参加者に選好情報を提示させ、この課題を解決する。二つ目はドライバー及びタスク所有者が共に非常に多く、このマッチング問題が非常に大規模になるという点である。本研究では、こ

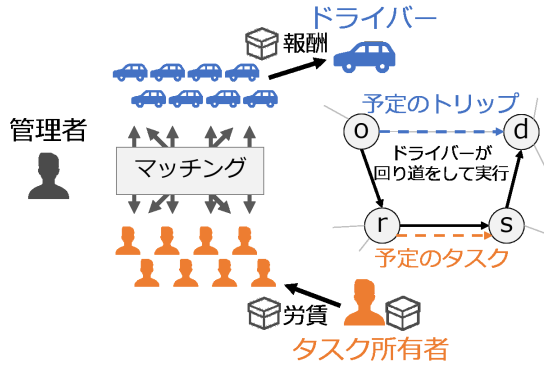


Fig.1 想定する CSD システム

のオークションに対する計算効率的メカニズムを開発することで、この課題を解決する。

以上の2点を考慮したCSDに関する研究は、渡辺・赤松(2020)¹⁾のみである。しかし、この研究ではドライバー数、タスク数は与件であるとし、これらの価格弾力性は考慮していない。実際のシステムにおいては、ドライバーはCSD管理者に提示される報酬額によって配送タスクを実行するかどうかを選択し、タスクの依頼者も管理者に提示される料金によって依頼するかどうかを選択する。すなわち、ドライバー数およびタスク数はともに価格弾力性がある。この価格弾力性を考慮しないマッチングシステムでは、ドライバーやタスクに無理なマッチング(i.e., 報酬や料金が満足なものでなくても配送や依頼を強いられる)を強いる可能性があり、望ましいシステムとは言えない。これらの問題を解決するため本研究では、ドライバー数、タスク数の価格弾力性を考慮できるマッチング方法を提案する。

2. モデル・定式化

本章では想定するシステムを設定し、システムに参加する各主体の知覚費用を定式化する。その後、社会的最適なマッチングを定式化し、市場を活用した取引がこのマッチングを実現することを示す。

2.1 システムの設定

本研究で想定するシステムを図1に示す。このシステムにおける行動主体はタスク所有者、ドライバー、管理者の3つである。

タスク所有者は労賃を支払い、タスクを委託する主体である。タスク所有者全員の集合を A とし、個人を a で表す。また、OD ペアが rs のタスク (以降 rs タスクと呼ぶ) を所有するタスク所有者の集合は A_{rs} と表す。 rs タスクの所有者は、以下で定義される自らの知覚費用を最小化するよう、タスクを委託するかどうかを選択する：

$$\begin{cases} c_a^0 & \text{タスクを委託しない} \\ c_a^1 + u_{rs} & \text{タスクを委託する} \end{cases} \quad (1)$$

c_a^0 はタスク所有者 a がタスクを委託しない時の知覚費用を表し、 c_a^1 はタスク所有者 a がタスクを委託する時の (タスク価格を除いた) 知覚費用を表している。 $k \in \{0,1\}$ はタスクの委託状況を表し、 $k=0$ の時はタスクを委託しない、 $k=1$ の時はタスクを委託する状態を表している。 また、 u_{rs} は rs タスクの価格である。

ドライバーは報酬を受け取りタスクを実行する主体である。ドライバー全員の集合を B とし、個人を b で表す。また、OD ペアが od のトリップを予定するドライバー (以降 od ドライバーと呼ぶ) の集合は B_{od} と表す。 od ドライバーは、以下で定義される自らの知覚費用を最小化するよう、実行するタスクを選択する：

$$\begin{cases} c_b^0 & \text{タスクを実行しない} \\ c_b^{rs} - u_{rs} & rs \text{ タスクを実行する} \end{cases} \quad (2)$$

c_b^0 はドライバー b がタスクを実行しない時の知覚費用を表し、 c_b^{rs} はドライバー b が rs のタスクを実行する時の (タスク報酬額を除いた) 知覚費用を表している。

管理者は、社会的総費用が最小となるようにタスク所有者とドライバーのマッチングを決定する主体である。

また、システムを適用する都市はノード集合 \mathcal{N} 、リンク集合 \mathcal{L} で構成されるネットワーク $G(\mathcal{N}, \mathcal{L})$ で表現する。

2.2 社会的最適マッチング

提案システムでの実現目標である社会的最適状態を達成するマッチング問題は、以下の総費用最小化問題 [SO-P] となる：

[SO-P]

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{z}} TC(\mathbf{y}, \mathbf{z}) &= \sum_a \sum_{k \in \{0,1\}} c_a^k y_a^k + \sum_b \sum_{rs \in RS'} c_b^{rs} z_b^{rs} \quad (3) \end{aligned}$$

$$\text{s.t.} \quad \sum_{a \in A_{rs}} y_a^1 \leq \sum_b z_b^{rs} \quad \forall rs \in RS \quad (4)$$

$$\sum_{k \in \{0,1\}} y_a^k = 1 \quad \forall a \in A \quad (5)$$

$$\sum_{rs \in RS'} z_b^{rs} = 1 \quad \forall b \in B \quad (6)$$

$$y_a^k \in \{0,1\} \quad \forall a \in A, k \in \{0,1\} \quad (7)$$

$$z_b^{rs} \in \{0,1\} \quad \forall b \in B, rs \in RS \quad (8)$$

ここで、 y_a^k はタスク所有者 a が k の状態の時に1、それ以外で0となる変数である。同様に、 z_b^{rs} はドライバー b が rs のタスクを実行する場合に1、それ以外で0となる変数である。また、 RS は全てのタスク OD

ペアの集合を表し, RS' は RS にダミー OD ペア 0 を 1 つ追加した集合である. 目的関数にタスク価格 (報酬額) \mathbf{u} が含まれないのは, 金銭の授受が各主体間での所得移転にすぎないためである.

[SO-P] は前述のとおり, 管理者が直接解くことはできない. なぜなら, 参加主体の知覚費用 \mathbf{c} が私的情報であるため管理者が直接観測できないためである. 本稿では, オークション市場を導入することでこの問題を解決する.

2.3 社会的最適マッチングの市場を活用した実現方法

本節では, オークションで達成される市場均衡状態が, 社会的最適状態と等価であることを示す.

市場均衡状態では, 以下の 3 条件が同時に成立する:

$$\begin{cases} y_a^1 = 1 & \text{if } c_a^1 + u_{rs} = \rho_a \\ y_a^1 = 0 & \text{if } c_a^1 + u_{rs} \geq \rho_a \end{cases} \quad \forall a \in A_{rs}, rs \in RS \quad (9)$$

$$\begin{cases} z_b^{rs} = 1 & \text{if } c_b^{rs} - u_{rs} = \rho_b \\ z_b^{rs} = 0 & \text{if } c_b^{rs} - u_{rs} \geq \rho_b \end{cases} \quad \forall b \in B, rs \in RS \quad (10)$$

$$\begin{cases} u_{rs} \geq 0 & \text{if } \sum_{a \in A_{rs}} y_a^1 = \sum_b z_b^{rs} \\ u_{rs} = 0 & \text{if } \sum_{a \in A_{rs}} y_a^1 \leq \sum_b z_b^{rs} \end{cases} \quad \forall rs \in RS \quad (11)$$

ρ_a, ρ_b は, 各々, タスク所有者 a の均衡状態における (タスク価格を含む) 知覚費用, ドライバー b の均衡状態における (タスク報酬額を含む) 知覚費用を表している. (9), (10) はそれぞれ, タスク所有者とドライバーが自身の費用を最小化しているという条件であり, (11) は, 需給均衡条件である.

この市場均衡状態は, [SO-P] の最適性条件と一致する (証明は省略). 従って, 以下の命題が成立する.

命題 1 提案システムで実現するマッチングは, 社会的総費用を最小化する.

3. 提案メカニズム

本章では, 提案するメカニズムの概要を示し, 提案メカニズムで解く必要のある最適化問題を定式化する.

提案メカニズムでは, タスク所有者とドライバーの集合をそれぞれ OD ペア rs, od ごとに分解し, 2 段階のプロセスでマッチングを決定する. 第 1 段階では, rs タスクの実行数, および od ドライバー集合へのタスク配分数を決定する. これらは過去のオークション履歴から, 各主体の知覚費用の分布を統計的に推定することで決定する. 第 2 段階では, オークションにより, 個々のタスク所有者のタスク委託有無, および od ドライバー集合内でのマッチングを決定する. この分解により, 各オークションはごく小規模なものとなる.

この 2 段階のプロセスは, [SO-P] を 2 段階の階層的問題に分解して解くことに相当する. 従って, 提案メカ

ニズムで得られるマッチングは, [SO-P] の解と一致する. 上位問題では, 実行しない rs タスク数 n_{rs}^0 と実行する rs タスク数 n_{rs}^1 , およびタスクを実行しない od ドライバー数 f_{od}^0 と rs タスクを実行する od ドライバー数 f_{od}^{rs} を決定する. この問題は, 提案メカニズムの第 1 段階に相当する. 下位問題では, 上位問題で決定した \mathbf{n}, \mathbf{f} を与件として, 個々のタスク所有者のタスク委託有無 \mathbf{y} および, ドライバー集合内でのマッチング \mathbf{z} を決定する. 以降, この [SO-P] の階層分解を行い, メカニズムの各プロセスの手続きをより具体的に示す.

まず, [SO-P] を \mathbf{n}, \mathbf{f} を含んだ形式に書き換える:

$$\min_{\mathbf{y}, \mathbf{z}, \mathbf{n}, \mathbf{f}} TC(\mathbf{y}, \mathbf{z}) = \sum_{rs \in RS} \left(\sum_{a \in A_{rs}} \sum_{k \in \{0,1\}} c_a^k y_a^k \right) + \sum_{od \in OD} \left(\sum_{b \in B_{od}} \sum_{rs \in RS'} c_b^{rs} z_b^{rs} \right) \quad (12)$$

s.t. (5), (6), (7), (8),

$$n_{rs}^k = \sum_{a \in A_{rs}} y_a^k \quad \forall rs \in RS, k \in \{0,1\} \quad (13)$$

$$f_{od}^{rs} = \sum_{b \in B_{od}} z_b^{rs} \quad \forall od \in OD, rs \in RS' \quad (14)$$

$$n_{rs}^1 \leq \sum_{od \in OD} f_{od}^{rs} \quad \forall rs \in RS \quad (15)$$

$$\sum_{k \in \{0,1\}} n_{rs}^k = \bar{n}_{rs} \quad \forall rs \in RS \quad (16)$$

$$\sum_{rs \in RS'} f_{od}^{rs} = q_{od} \quad \forall od \in OD \quad (17)$$

ここで, $\bar{n}_{rs} = |A_{rs}|, q_{od} = |B_{od}|$ である. また, OD はドライバーのトリップ OD ペアの集合を表す.

次に, この問題を, 上位問題 [SO-P/Master] とタスク所有者に関する下位問題 [SO-P/Sub(rs)], ドライバーに関する下位問題 [SO-P/Sub(od)] に分解する.

下位問題 [SO-P/Sub(rs)] では, 実行されるタスク数を与件として, 各タスク所有者集合ごとに誰のタスクを実行するかを決定する:

[SO-P/Sub(rs)]

$$\begin{aligned} CC_{rs}^*(\mathbf{n}_{rs}) &= \min_{\mathbf{y}_{(rs)}} CC_{rs}(\mathbf{y}_{(rs)} | \mathbf{n}_{rs}) \\ &= \sum_{a \in A_{rs}} \sum_{k \in \{0,1\}} c_a^k y_a^k \end{aligned} \quad (18)$$

$$\text{s.t. } \sum_{k \in \{0,1\}} y_a^k = 1 \quad \forall a \in A_{rs} \quad (19)$$

$$\sum_{a \in A_{rs}} y_a^k = n_{rs}^k \quad \forall k \in \{0,1\} \quad (20)$$

$$y_a^k \in \{0,1\} \quad \forall a \in A_{rs}, k \in \{0,1\} \quad (21)$$

下位問題 [SO-P/Sub(*od*)] では、配分されたタスク数を与件として、各ドライバー集合ごとに、マッチングを決定する：

[SO-P/Sub(*od*)]

$$\begin{aligned} PC_{od}^*(\mathbf{f}_{od}) &= \min_{\mathbf{z}_{(od)}} PC_{od}(\mathbf{z}_{(od)} | \mathbf{f}_{od}) \\ &= \sum_{b \in B_{od}} \sum_{rs \in RS'} c_b^{rs} z_b^{rs} \end{aligned} \quad (22)$$

$$\text{s.t.} \quad \sum_{rs \in RS'} z_b^{rs} = 1 \quad \forall b \in B_{od} \quad (23)$$

$$\sum_{b \in B_{od}} z_b^{rs} = f_{od}^{rs} \quad \forall rs \in RS' \quad (24)$$

$$z_b^{rs} \in \{0, 1\} \quad \forall b \in B_{od}, rs \in RS' \quad (25)$$

一方上位問題 [SO-P/Master] は、各集合において最適なマッチングが行われるという前提の下、最適な配分 \mathbf{n}, \mathbf{f} を決定する：

[SO-P/Master]

$$\begin{aligned} \min_{\mathbf{n}, \mathbf{f}} TC(\mathbf{n}, \mathbf{f}) \\ = \sum_{rs \in RS} CC_{rs}^*(\mathbf{n}_{rs}) + \sum_{od \in OD} PC_{od}^*(\mathbf{f}_{od}) \end{aligned} \quad (26)$$

$$\text{s.t.} \quad (15), (16), (17)$$

$$n_{rs}^k \geq 0 \quad \forall k \in \{0, 1\}, rs \in RS \quad (27)$$

$$f_{od}^{rs} \geq 0 \quad \forall rs \in RS', od \in OD \quad (28)$$

さらに、上位問題 [SO-P/Master] の目的関数における最適値関数 CC^*, PC^* を、過去のマッチング結果から得られる c の分布を用いて統計的に推定する。

本研究では、 rs タスク所有者の知覚費用 c_a^k を以下のように仮定する：

$$c_a^k = C_{rs}^k - \varepsilon_a^k \quad \forall k \in \{0, 1\} \quad (29)$$

C_{rs}^0, C_{rs}^1 は各々、タスク所有者が配送を自力で行う際とタスクを委託する際の交通費用である。また、 $\varepsilon_a^0, \varepsilon_a^1$ は私的効用値であり、パラメータ θ_C の Gumbel 分布に従う確率項である。同様に、 od ドライバーの知覚費用を以下のように仮定する：

$$c_b^{rs} = C_{od}^{rs} - \varepsilon_b^{rs} \quad \forall rs \in RS' \quad (30)$$

C_{od}^0, C_{od}^{rs} は各々、ドライバーが rs タスクを行わない際とタスクを実行する際の交通費用である。また、 $\varepsilon_b^0, \varepsilon_b^{rs}$ は私的効用値であり、パラメータ θ_P の Gumbel 分布に従う確率項である。

これらの仮定の下では、最適値関数 CC^*, PC^* は以下の式で表すことができる：

$$CC_{rs}^*(\mathbf{n}_{rs}) = \sum_{k \in \{0, 1\}} n_{rs}^k C_{rs}^k - H_{rs}^C(\mathbf{n}_{rs}) \quad (31)$$

$$PC_{od}^*(\mathbf{f}_{od}) = \sum_{rs \in RS'} C_{od}^{rs} f_{od}^{rs} - H_{od}^P(\mathbf{f}_{od}) \quad (32)$$

$$\text{where} \quad H_{rs}^C(\mathbf{n}_{rs}) = -\frac{1}{\theta_C} \sum_{k \in \{0, 1\}} n_{rs}^k \ln \frac{n_{rs}^k}{\bar{n}_{rs}} \quad (33)$$

$$H_{od}^P(\mathbf{f}_{od}) = -\frac{1}{\theta_P} \sum_{rs \in RS'} f_{od}^{rs} \ln \frac{f_{od}^{rs}}{q_{od}} \quad (34)$$

従って、上位問題 [SO-P/Master] は \mathbf{n}, \mathbf{f} を変数とする Closed-Form の問題 [SO-A-P] となる：

[SO-A-P]

$$\begin{aligned} \min_{\mathbf{n}, \mathbf{f}} \sum_{rs \in RS} \left\{ \sum_{k \in \{0, 1\}} C_{rs}^k n_{rs}^k - H_{rs}^C(\mathbf{n}_{rs}) \right\} \\ + \sum_{od \in OD} \left\{ \sum_{rs \in RS'} C_{od}^{rs} f_{od}^{rs} - H_{od}^P(\mathbf{f}_{od}) \right\} \end{aligned} \quad (35)$$

$$\text{s.t.} \quad (15), (16), (17), (27), (28)$$

また、上位問題 [SO-A-P] の双対問題 [SO-A-D] は以下のように定式化できる：

[SO-A-D]

$$\max_{\mathbf{u} \geq \mathbf{0}} \sum_{rs \in RS} \bar{n}_{rs} V_{rs}^C(u_{rs}) + \sum_{od \in OD} q_{od} V_{od}^P(\mathbf{u}) \quad (36)$$

$V_{rs}^C(u_{rs}), V_{od}^P(\mathbf{u})$ はそれぞれ、タスク価格が \mathbf{u} の時のタスク所有者、ドライバーの認知する期待最小費用を表しており、以下の式で定義される：

$$V_{rs}^C(u_{rs}) = -\frac{1}{\theta_C} \ln \sum_k \exp \{-\theta_C (C_{rs}^k + u_{rs}^k)\} \quad (37)$$

$$V_{od}^P(\mathbf{u}) = -\frac{1}{\theta_P} \ln \sum_{rs \in RS'} \exp \{-\theta_P (C_{od}^{rs} - u_{rs})\} \quad (38)$$

ただし、 $u_{rs}^0 = u_{rs}, u_{rs}^1 = 0, u_0 = 0$ である。

以上のように、提案システムでは [SO-A-P] および [SO-P/Sub] を続けて解くことでマッチングを決定する。しかし、[SO-A-P] は問題規模が $|\mathcal{N}|^4$ の大規模な問題であり、汎用的アルゴリズムで解くことは難しい。そこで、次章から [SO-A-P] の効率的な解法を提案する。

4. 上位問題の効率的解法

[SO-A-P] は 4 ノード (o, d, r, s) の組み合わせを列挙しているが、問題の DP 原理的性質を利用すれば、この列挙を回避して問題規模を大幅に縮小することができる。これを具体的に示すために、本章では [SO-A-P] を交通量配分問題へと変形し、リンク変数のみを用いた形式に再定式化する。この再定式化により、列挙が回避され問題が縮約される。

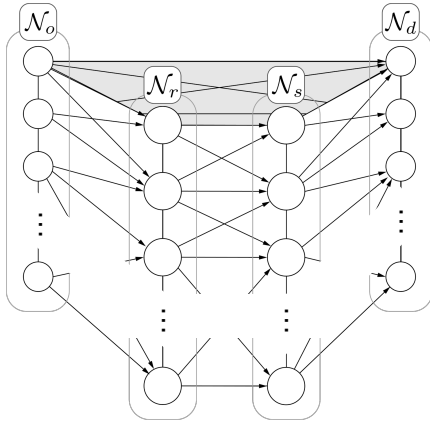


Fig.2 仮想ネットワーク

4.1 仮想ネットワークの構築

まず、配分対象とするネットワークとして、[SO-A-P]のDP原理的構造を表現する仮想的なネットワーク $G_v(\mathcal{N}_v, \mathcal{L}_v)$ を構築する。より具体的には、 G_v は図2で表すネットワークであり、以下のような構造を持つ：

- ノード集合 \mathcal{N}_v は \mathcal{N} のコピー4つ($\mathcal{N}_o, \mathcal{N}_r, \mathcal{N}_s, \mathcal{N}_d$)の和集合である。各ノードの番号は自身のコピー元のノード番号と同じとする。
- 各リンク $ij \in \mathcal{L}_v$ のリンクコスト τ_{ij} は、 $i, j \in \mathcal{N}_v$ のコピー元であるノード $i', j' \in \mathcal{N}$ 間の最短経路費用 $t_{i'j'}$ とする：

$$\tau_{ij} = t_{i'j'} \quad \forall ij \in \mathcal{L}_v \quad (39)$$

G_v の経路への交通配分がドライバーとタスクのマッチングを表していることは、以下のようにして確かめられる： G_v は経路 $o-r-s-d$ の観測交通費用は、 od ドライバーが rs タスクを実行する時の観測交通費用 C_{od}^{rs} に一致する。同様に、経路 $o-d$ の観測交通費用は、 od ドライバーがタスクを実行しない時の観測交通費用 C_{od}^0 に一致する。すなわち、ドライバー1人を G_v の経路を通行する車両1台として表現すれば、ある od ドライバー b が rs タスクを実行する際に費やす費用 $c_b^\beta = C_{od}^\beta - \varepsilon_b^\beta$ は、ドライバー b を表す車両1台を G_v 上の経路 $o-r-s-d$ に配分する際の認知経路費用と解釈できる。従って、費用の観点から見れば、 G_v の経路に車両1台を配分することは、1人のドライバーとタスクをマッチングさせることと等価である。

4.2 交通量配分問題への変形および縮約

前節で示した G_v への交通配分とCSDシステムにおけるマッチングの対応を利用すれば、[SO-A-P]は G_v 上の交通量配分問題とみなすことができる。ここで、 G_v 上での各車両の認知経路費用は、観測経路費用 C_{od}^{rs} とGumbel分布に従う誤差項の和で表現される状況を考えているため、この交通量配分問題はLOGIT配分問題となる。

さらにAkamatsu(1997)²⁾によって示された通り、LOGIT配分問題は起点別リンクフローのみを用いた

[SO-L-P]に等価変形することができる：

[SO-L-P]

$$\min_{\mathbf{n}, \mathbf{x} \geq \mathbf{0}} \sum_{rs \in RS} \sum_{k \in \{0,1\}} n_{rs}^k C_{rs}^k - \sum_{rs \in RS} H_{rs}^C(\mathbf{n}_{rs}) + \sum_{o \in \mathcal{N}_o} \sum_{ij \in \mathcal{L}_v} \tau_{ij} x_{ij}^o - \sum_{o \in \mathcal{N}_o} H^o(\mathbf{x}^o) \quad (40)$$

$$\text{s.t. } n_{rs}^1 \leq \sum_{o \in \mathcal{N}_o} x_{rs}^o \quad \forall rs \in RS \quad (41)$$

$$\sum_{i \in NI_l} x_{il}^o - \sum_{j \in NO_l} x_{lj}^o = \begin{cases} q_{od} & l = d \in \mathcal{N}_d \\ 0 & \text{otherwise} \end{cases} \quad \forall l \in \{\mathcal{N}_r, \mathcal{N}_s, \mathcal{N}_d\} \quad (42)$$

$$\text{where } H^o(\mathbf{x}^o) = -\frac{1}{\theta_P} \left(\sum_{ij \in \mathcal{L}_v} x_{ij}^o \ln \frac{x_{ij}^o}{\sum_{i \in NI_j} x_{ij}^o} \right) \quad \forall o \in \mathcal{N}_o \quad (43)$$

ここで、 x_{ij}^o は、 G_v において起点が $o \in \mathcal{N}_o$ であるリンク $ij \in \mathcal{L}_v$ のフローを表す。また、 $NI_l(NO_l)$ は、ノード l に流入(から流出)するリンクの上流(下流)側ノードの集合である。

一方、[SO-A-D]は以下の[SO-L-D]に再定式化できる：

[SO-L-D]

$$\max_{\mathbf{u} \geq \mathbf{0}} TC_{LD}(\mathbf{u}) = \sum_{rs \in RS} \bar{n}_{rs} V_{rs}^C(u_{rs}) - \sum_{od \in OD} q_{od} \mu_d^o(\mathbf{u}) \quad (44)$$

変数 u_{rs} は rs リンクに賦課される通行料金と解釈できる。 μ_i^o は、 G_v 上で、 rs リンクに費用 $-u_{rs}$ が足されている状態における、起点 $o \in \mathcal{N}_o$ からノード $l \in \mathcal{N}_v$ までの期待最小費用を表す。具体的には、 μ_k^o は以下で定義される：

$$\mu_r^o = \tau_{or} \quad \forall r \in \mathcal{N}_r \quad (45)$$

$$\mu_s^o(\mathbf{u}) = -\frac{1}{\theta_P} \ln \sum_{r \in \mathcal{N}_r} \exp[-\theta_P \{\mu_r^o + \tau_{rs} - u_{rs}\}] \quad \forall s \in \mathcal{N}_s \quad (46)$$

$$\mu_d^o(\mathbf{u}) = -\frac{1}{\theta_P} \ln \sum_{s \in \mathcal{N}_s} \exp[-\theta_P \{\mu_s^o(\mathbf{u}) + \tau_{sd}\}] \quad \forall d \in \mathcal{N}_d \quad (47)$$

Algorithm 1 Main Algorithm

Require: $\epsilon > 0, L_0 > 0, \eta > 1$ **Ensure:** \mathbf{u}^*

```
 $\mathbf{u}_0 \leftarrow \mathbf{0}, \hat{\mathbf{u}}_0 \leftarrow \mathbf{0}, l = 0, t_0 = 1$ 
while  $\max_{rs} \left\{ \left| \frac{\partial TC_{LD}}{\partial u_{rs}}(\mathbf{u}_{l+1}) \right| \right\} > \epsilon$  do
   $TC_{LD}(\mathbf{u}_l), \nabla TC_{LD}(\mathbf{u}_l)$ 
   $L_{l+1} \leftarrow \min_{l \in \mathbb{Z}_{\geq 0}} \{l | P(\mathbf{u}_l, L_l, l) \leq Q(\mathbf{u}_l, L_l, l)\}$ 
  if  $\nabla TC_{LD}(\mathbf{u}_l) \cdot (\mathbf{u}_{l+1} - \mathbf{u}_l) > 0$  then
     $t_{l+1} = 1$ 
  else
     $t_{l+1} = \frac{1 + \sqrt{1 + 4t_l^2}}{2}$ 
  end if
   $\hat{\mathbf{u}}_l = \mathbf{u}_l - \frac{1}{L_{l+1}} \nabla z_{LD}(\mathbf{u}_l)$ 
   $\mathbf{u}_{l+1} = \hat{\mathbf{u}}_l + \frac{t_l - 1}{t_{l+1}} (\hat{\mathbf{u}}_l - \hat{\mathbf{u}}_{l-1})$ 
   $l \leftarrow l + 1$ 
end while
return  $\mathbf{u}^* = \mathbf{u}^l$ 
```

このように、問題を起点別リンク変数で表現することで、問題規模を $|\mathcal{N}|^4$ から $3|\mathcal{N}|^3$ に縮小することができる。

5. アルゴリズム

本章では、前章で変形された [SO-L-P] を解くアルゴリズムを提案する。提案アルゴリズムでは双対問題 [SO-L-D] を解き、その解を利用して上位問題の解を得る。これは、[SO-L-D] が無制約最適化問題であり、[SO-L-P] に比べて効率的に解くことができるからである。

本研究では、加速勾配法と Markov 配分を組み合わせたアルゴリズムにより [SO-L-D] を解く。

加速勾配法は、Nesterov(1983)³⁾ によって提案された 1 次法アルゴリズムであり、最悪収束率の観点から 1 次法として最適なアルゴリズムである。通常、小規模な問題では 1 次法より 2 次法の方が効率的だが、[SO-L-D] は次元が大きく Hessian の計算量およびメモリ使用量が膨大になるため 2 次法 (e.g., Newton 法) の適用は非効率的である。以上の理由から、本研究では加速勾配法を採用した。具体的なアルゴリズムは Algorithm1 に示す。Algorithm1 中の $P(\mathbf{u}_l, L_l, l), Q(\mathbf{u}_l, L_l, l)$ は以下の式で与えられる：

$$P(\mathbf{u}_l, L_l, l) \equiv TC_{LD} \left(\mathbf{u}_l - \frac{1}{\eta^l L_l} \nabla TC_{LD}(\mathbf{u}_l) \right) - TC_{LD}(\mathbf{u}_l) \quad (48)$$

$$Q(\mathbf{u}_l, L_l, l) \equiv -\frac{1}{2\eta^l L_l} \|\nabla TC_{LD}(\mathbf{u}_l)\|^2 \quad (49)$$

Markov 連鎖配分は、加速勾配法で必要となる目的関数値 TC_{LD} および勾配 ∇TC_{LD} の計算に利用する。 TC_{LD} は式 (44)、 ∇TC_{LD} は以下の式で計算すること

ができる：

$$\frac{\partial TC_{LD}(\mathbf{u})}{\partial u_{rs}} = -n_{rs}^1(u_{rs}) + \sum_o x_{rs}^o(\mathbf{u}) \quad \forall rs \in RS \quad (50)$$

$$n_{rs}^1(u_{rs}) = \bar{n}_{rs} \frac{\exp\{-\theta_C (C_{rs}^1 + u_{rs}^1)\}}{\sum_k \exp\{-\theta_C (C_{rs}^k + u_{rs}^k)\}} \quad \forall rs \in RS \quad (51)$$

$x_{rs}^o(\mathbf{u})$ は、リンク rs のコストを $\tau_{rs} - u_{rs}$ とした G_v において LOGIT 配分を行った際の、起点が o であるリンク rs のフローである。従って、 $TC_{LD}, \nabla TC_{LD}$ は LOGIT 配分アルゴリズムによって計算できるが、前章で行った問題の縮約を活かす効率的な計算を行うためには、適用する LOGIT 配分アルゴリズムは経路列挙を必要としない LOGIT 配分アルゴリズムでなければならない。そのため本研究では、この要請を満たす Markov 連鎖配分を適用した。

6. 終わりに

本研究では、CSD における、大規模システムの場合でも適用可能なオークションを用いたマッチング方法を提案した。提案メカニズムでは、まずドライバーを OD ペアごとのグループに分け、マッチングは 2 段階で決定する。第 1 段階では、実行するタスク数とドライバーグループごとへのタスクの配分を決定する。第 2 段階では、各グループごとにオークションを行うことによって、タスク所有者のタスク委託有無、およびドライバーグループ内でのマッチングを決定する。各小市場の規模が十分に小さいことから、第 2 段階のオークションは容易に行うことができる。一方、第 1 段階で解く問題は直接解くと計算負荷が大きいため、マッチングを経路への交通量配分問題として表現できる仮想ネットワーク上の交通量配分問題に変換し、経路変数をリンク変数に置き換えることで問題規模を大幅に縮小した。

本研究では、委託されるタスクは全て同質であるとしているが、実際のタスクはそのサイズが異なる等の異質性があると考えるのが自然である。従って、タスクの異質性を考慮したシステムの設計は今後の課題となる。

謝辞 本研究は、日本学術振興会・科学研究費補助金・基盤研究 (B) (課題番号:18H01551) の助成を受けた研究の一部です。ここに記し、感謝を表します。

参考文献

- 1) 渡邊大樹・赤松隆: クラウド・ソーシング配送システムにおける効率的なマッチング, 交通工学研究発表会 **40**, 116, 2020.
- 2) Akamatsu, T.: Decomposition of path choice entropy in general transport networks, *Transportation Science*, Vol.31, No.4, pp. 349-362, 1997.
- 3) Nesterov, Y. E.: A method for solving the convex programming problem with convergence rate $O(1/k^2)$, *Dokl. akad. nauk Sssr*, Vol.269, pp. 543-547, 1983.