

大規模クラウドソース配送システムの 効率的マッチング・メカニズム： 需給の価格弾力性を考慮した一般化

武田 翼¹・赤松 隆²

¹非会員 東北大学 大学院情報科学研究科人間社会情報科学専攻 (〒 980-8579 仙台市青葉区荒巻字青葉 6-6-06-408)
E-mail: tsubasa.takeda.t5@dc.tohoku.ac.jp

²正会員 東北大学教授 大学院情報科学研究科人間社会情報科学専攻 (〒 980-8579 仙台市青葉区荒巻字青葉 6-6-06-408)
E-mail: akamatsu@plan.civil.tohoku.ac.jp

クラウドソース配送システム (CSD) とは、個人ドライバーがトリップ時に回り道をして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、ドライバーと配送タスクのマッチングに膨大な計算を要する。また、最適マッチング実現のためには荷主の希望価格およびドライバーの希望報酬額を管理者が正確に把握する必要がある。本研究では、これらの課題を解決したマッチング手法を提案する。まず、オークション市場を導入することで、荷主とドライバーの選好をマッチングに反映する。続いて、このオークションに適用する計算効率のメカニズムを提案する。最後に、数値実験により提案メカニズムが計算効率を大幅に改善することを示した。

Key Words: crowdsourced delivery, last-mile delivery, auction market, accelerated gradient method

1. はじめに

クラウドソース配送システム (CSD) は、都市内をトリップする予定のドライバーが、依頼された配送をトリップ時に回り道をして行う配送システムである。多数の車両で少量ずつ荷物を運ぶため、CSD は近年の小口・多頻度な配送需要との親和性が高い。また、CSD では既存のトリップを利用することから、従来の手法より効率的に配送を実行することができる。実際に、既往研究¹⁾²⁾³⁾により、その効率性が証明されている。

しかし、CSD では、配送タスクを委託する荷主と配送を実行するドライバーを適切にマッチングする必要があり、その際に二つの課題が生じる：(課題 1) 荷主の希望価格およびドライバーの希望報酬額を管理者が得られない。(課題 2) 参加者の多いシステムにおいては、マッチング決定問題が非常に大規模になる。

CSD でのマッチング手法に関する既往研究のほとんどはこれらの課題を解決できていない。CSD の効率性を検証した上記の研究¹⁾²⁾³⁾はいずれも、比較的小規模なシステムを想定し、マッチングは管理者が独裁的に決定する。従って、これらの研究で扱ったマッチング手法では二つの課題を解決することができない。これに対し、Wang et al.(2016)⁴⁾は大規模なシステムでも適用可能な効率的なマッチング手法を開発したが、他の研究と同様に、そのマッチングに参加者の選好を反映することができておらず、課題 1 の解決には至ってい

ない。

この二つの課題の解決を目指した研究は、渡邊 (2020)⁵⁾のみである。しかし、渡邊の研究は、委託されるタスク数と配送を実行するドライバー数が一定であると仮定した特殊なケースのみを扱っている。実際には、タスクの価格によってこれらの数は変動すると考えるのが自然である。従って、本研究では、委託されるタスク数および配送を実行するドライバー数が価格によって変動する、より現実的な CSD モデルにおいて、二つの課題を解決するマッチング・メカニズムを開発する。具体的には、課題 1 をオークション市場の導入によって解決し、課題 2 を計算効率的なメカニズムの開発により解決する。

2. モデル・定式化

(1) 想定するシステム

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、(1) 荷主、(2) ドライバー、(3) 管理者の三つである。

荷主は労賃を支払い、タスクを委託する主体である。荷主全員の集合を \mathcal{A} とし、OD ペアが rs のタスクを所有する荷主 (以降 rs 荷主と呼ぶ) の集合を \mathcal{A}_{rs} と表す。 rs 荷主は、以下で定義される自らの費用を最小化する

よう、タスクを委託するかを選択する：

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

ここで、 c_a^0, c_a^1 は荷主 $a \in \mathcal{A}_{rs}$ の (タスク価格を除いた) 知覚費用を表し、 u_{rs} は rs タスクの価格である。以降では、荷主のタスク委託状況を表す二値変数 $k \in \{0, 1\}$ により、 c_a^0, c_a^1 をまとめて c_a^k と表記する。

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 \mathcal{B} とし、 OD ペアが od のドライバー (以降 od ドライバーと呼ぶ) の集合を \mathcal{B}_{od} と表す。ドライバは、以下で定義される自らの費用をを最小化するように、実行するタスクを最大一つ選択する：

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

ここで、 c_b^0, c_b^s はドライバー $b \in \mathcal{B}$ の (タスク報酬額を除いた) 知覚費用を表す。

管理者は、参加者 (荷主とドライバー) の総費用を最小化する事を目指し、マッチングを決定する主体である。

また、本研究では都市をネットワーク $G \equiv (\mathcal{N}, \mathcal{L})$ で表現する。 \mathcal{N} は都市ネットワークのノード集合を表し、 \mathcal{L} はリンク集合を表す。タスクの起終点 rs とドライバーの起終点 od は、いずれも都市ネットワーク G 上のノードペアで表現される。

(2) 最適マッチング

本研究では、最適マッチングを以下の総費用最小化問題 [SO] として定義する：

[SO]

$$\min_{\mathbf{y}, \mathbf{z}} TC(\mathbf{y}, \mathbf{z}) = \mathbf{c}_c^\top \mathbf{y} + \mathbf{c}_p^\top \mathbf{z} \quad (3)$$

$$\text{s.t.} \quad \sum_{a \in \mathcal{A}_{rs}} y_a^1 = \sum_{b \in \mathcal{B}} z_b^{rs} \quad \forall rs \in \mathcal{RS} \quad (4)$$

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

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

ここで、 $\mathbf{c}_c \equiv \{c_a^k\}, \mathbf{c}_p \equiv \{c_b^{rs}\}$ であり、 $\mathbf{y} \equiv \{y_a^k\}, \mathbf{z} \equiv \{z_b^{rs}\}$ はそれぞれ荷主とドライバーのマッチングを示す 0-1 変数のベクトルである。 y_a^k は荷主 a が k 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 タスクを実行する際に 1 となり、それ以外で 0 となる。 \mathcal{RS} はタスク起終点の集合を表し、 \mathcal{RS}' は \mathcal{RS} にタスクを実行しない事を表すダミー OD ペア 0 を追加した集合である。

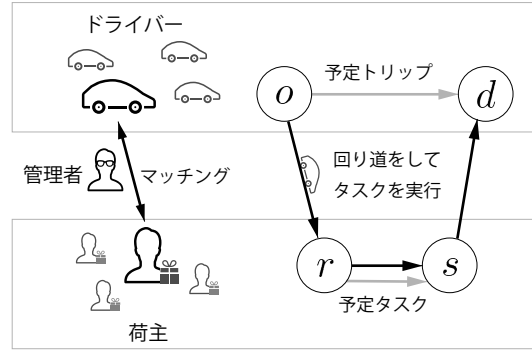


図-1 想定するシステム

[SO] の目的関数 (3) はタスク価格 (報酬額) u を除いた参加者の総費用である。これは、荷主とドライバー間における金銭の授受が単なる所得移転にすぎないためである。また、式 (4) は委託されたタスクと配送を実行するドライバー数が一致するという需給の条件である。式 (5) は荷主はタスクを委託するかしないかのどちらかを選択する事を表し、式 (6) はドライバーが最大で一つのタスクを実行する事を表している。

3. 最適マッチング問題の階層分解

[SO] は変数が膨大な組合せ最適化問題であり、このままでは現実的な時間での計算が不可能である。そこで、本研究では、離散的な変数 \mathbf{y}, \mathbf{z} を集計化した新たな変数の導入により別の定式化を行うことで、この課題の解決を目指す。具体的には、荷主およびドライバーをそれぞれ OD ペアが同じ小集合 $\mathcal{A}_{rs}, \mathcal{B}_{od}$ ごとに集計化する。、荷主集合 \mathcal{A}_{rs} 内で y_a^k を集計化した変数を $\mathbf{n}_{rs} \equiv \{n_{rs}^k\}$ 、ドライバー集合 \mathcal{B}_{od} 内で z_b^{rs} を集計化した変数を $\mathbf{f}_{od} \equiv \{f_{od}^{rs}\}$ とおく。以降では、 \mathbf{n}, \mathbf{f} をまとめて「タスク配分」と呼ぶ。また、荷主集合 \mathcal{A}_{rs} の人数を \bar{n}_{rs} 、ドライバー集合 \mathcal{B}_{od} の人数を q_{od} と表す。

この集計変数 \mathbf{n}, \mathbf{f} を導入することで、[SO] と等価な以下の階層的問題を得ることができる：

[SO/Master]

$$\min_{\mathbf{n}, \mathbf{f} \geq 0} \sum_{rs \in \mathcal{RS}} CC_{rs}^*(\mathbf{n}_{rs}) + \sum_{od \in \mathcal{OD}} PC_{od}^*(\mathbf{f}_{od}) \quad (7)$$

$$\text{s.t.} \quad n_{rs}^1 = \sum_{od \in \mathcal{OD}} f_{od}^{rs} \quad \forall rs \in \mathcal{RS} \quad (8)$$

$$\sum_{k \in \{0, 1\}} n_{rs}^k = \bar{n}_{rs} \quad \forall rs \in \mathcal{RS} \quad (9)$$

$$\sum_{rs \in \mathcal{RS}'} f_{od}^{rs} = q_{od} \quad \forall od \in \mathcal{OD} \quad (10)$$

[SO/Sub(rs)]

$$CC_{rs}^*(\mathbf{n}_{rs}) \equiv \min_{\mathbf{y}(rs)} \mathbf{c}_c^\top \mathbf{y}(rs) \quad (11)$$

$$\text{s.t.} \quad \sum_{k \in \{0,1\}} y_a^k = 1 \quad \forall a \in \mathcal{A}_{rs} \quad (12)$$

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

[SO/Sub(od)]

$$PC_{od}^*(\mathbf{f}_{od}) \equiv \min_{\mathbf{z}(od)} \mathbf{c}_p^\top \mathbf{z}(od) \quad (14)$$

$$\text{s.t.} \quad \sum_{rs \in \mathcal{RS}'} z_b^{rs} = 1 \quad \forall b \in \mathcal{B}_{od} \quad (15)$$

$$\sum_{b \in \mathcal{B}_{od}} z_b^{rs} = f_{od}^{rs} \quad \forall rs \in \mathcal{RS}' \quad (16)$$

マスター問題は参加者の総費用を最小化するタスク配分 \mathbf{n}, \mathbf{f} を決定する問題であり、サブ問題はタスク配分を与件として、各小集合 $\mathcal{A}_{rs}, \mathcal{B}_{od}$ ごとに総費用を最小化するマッチング \mathbf{y}, \mathbf{z} を決定する問題である。また、マスター問題中の \mathcal{OD} はドライバーの起終点集合を表す。

ここで、サブ問題を解くことは、各集合 $\mathcal{A}_{rs}, \mathcal{B}_{od}$ 内で VCG オークションを行うことと等価である。VCG オークションは、Vickrey⁶⁾, Clarke⁷⁾, Groves⁸⁾ によって提案された封印入札型¹のオークション・メカニズムであり、「最適なマッチングを実現でき、入札者が虚偽の入札を行う動機がない」という望ましい性質を有している。従って、管理者は参加者の知覚費用 \mathbf{c} を入札額として得ることができ、課題 1 が解決する。また、サブ問題は [SO] と同様の組合せ最適化問題であるが、各集合が十分に小さいことから、容易に計算することができる。

しかし、マスター問題はサブ問題の最適値関数を用いているため、素朴な解き方では計算に膨大な時間を要してしまう。従って、課題 2 を解決するために、次章ではマスター問題 [SO/Master] の効率的解法を開発する。

4. マスター問題の効率的解法

(1) [SO/Master] の縮約

[SO/Master] は、サブ問題の最適値関数を用いていることから、結果的に参加者の知覚費用 \mathbf{c} を離散的に考慮した問題となっている。しかし、知覚費用を各集合 $\mathcal{A}_{rs}, \mathcal{B}_{od}$ ごとに集計化した分布を考えることで、[SO/Master] は \mathbf{n}, \mathbf{f} のみを変数とした比較的規模の小さな問題へと変形することができる。実際には、知覚費用分布は日々のオークションで得られる参加者の入

¹ 他者の入札額を確認できず、自身の入札は一度しか行えないオークション形式。

札額を基にして推定するが、本研究では Gumbel 分布に従うものと仮定する：

仮定 1 参加者の知覚費用は以下の式で表される：

$$c_a^k = C_{rs}^k + \varepsilon_a^k \quad \forall a \in \mathcal{A}_{rs} \quad (17)$$

$$c_b^{rs} = C_{od}^{rs} + \varepsilon_b^{rs} \quad \forall b \in \mathcal{B}_{od} \quad (18)$$

ただし、 C_{rs}^k, C_{od}^{rs} は管理者が観測可能な荷主およびドライバーの交通費用であり、 ε_a^k はパラメータ θ_c の Gumbel 分布に、 ε_b^{rs} はパラメータ θ_p の Gumbel 分布に従う私的費用である。

この仮定の下では、[SO/Master] は以下の [SO-A] に帰着する：

[SO-A]

$$\min_{\mathbf{n}, \mathbf{f} \geq 0} \sum_{rs \in \mathcal{RS}} \{ \mathbf{C}_{rs}^\top \mathbf{n}_{rs} - H_{rs}^c(\mathbf{n}_{rs}) \} + \sum_{od \in \mathcal{OD}} \{ \mathbf{C}_{od}^\top \mathbf{f}_{od} - H_{od}^p(\mathbf{f}_{od}) \} \quad (19)$$

s.t. 式 (8), (9), (10)

ここで、 $H_{rs}^c(\mathbf{n}_{rs}), H_{od}^p(\mathbf{f}_{od})$ はエントロピー関数であり、 rs 荷主の総数 \bar{n}_{rs} および od ドライバーの総数 q_{od} を用いて、以下の式で表される：

$$H_{rs}^c(\mathbf{n}_{rs}) = -\frac{1}{\theta_c} \sum_k n_{rs}^k \ln \frac{n_{rs}^k}{\bar{n}_{rs}} \quad \forall rs \in \mathcal{RS} \quad (20)$$

$$H_{od}^p(\mathbf{f}_{od}) = -\frac{1}{\theta_p} \sum_{rs \in \mathcal{RS}'} f_{od}^{rs} \ln \frac{f_{od}^{rs}}{q_{od}} \quad \forall od \in \mathcal{OD} \quad (21)$$

(2) [SO-A] の縮約

[SO-A] の変数 $\mathbf{f} \equiv \{f_{od}^{rs}\}$ は 4 ノード (o, d, r, s) 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することができ、問題規模を更に縮小することができる。これを具体的に示すために、図-2 のような仮想ネットワーク $G_v \equiv (\mathcal{N}_v, \mathcal{L}_v)$ を構築する。その特徴は以下の通りである：

- ノードは 4 つの集合 $\mathcal{O}, \mathcal{D}, \mathcal{R}, \mathcal{S}$ で構成される。 \mathcal{O}, \mathcal{D} はドライバーのトリップ起終点を、 \mathcal{R}, \mathcal{S} は配送タスクの起終点を表現する集合である。
- リンクは 4 つのノード集合を $\mathcal{O} \rightarrow \mathcal{R} \rightarrow \mathcal{S} \rightarrow \mathcal{D}$ と繋いでおり、ドライバーがタスクを実行する際の経路を表現する。また、 $\mathcal{O} \rightarrow \mathcal{D}$ を繋ぐリンクは、タスクを実行しないドライバーの経路を表す。
- リンク ij のリンクコスト t_{ij} は、対応する実ネットワーク上の二つのノード $i, j \in \mathcal{N}$ 間の最短経路費用とする。

この仮想ネットワーク G_v を構築する事により、ドライバーの交通費用 C_{od}^{rs} は以下のように表現することができる。

$$C_{od}^{rs} = t_{or} + t_{rs} + t_{sd} \quad (22)$$

したがって、ドライバーの知覚費用 c_b^{rs} は仮想ネットワーク上の各経路に対する知覚費用と捉えることができ、タスク配分 \mathbf{f} を決定する問題は、 G_v の「経路」への交通量配分問題と捉えることができる。特に、知覚費用分布が Gumbel 分布に従う仮定の下では、この交通量配分問題は LOGIT 配分問題となる。

LOGIT 配分問題は、DP 原理により、 G_v の「経路」ではなく「リンク」のみを扱う問題へと変形する事ができる。具体的には、Akamatsu(1997)⁹⁾ が示した変形により、以下の [SO-L] へと変形することができる。

[SO-L]

$$\min_{\mathbf{n}, \mathbf{x} \geq 0} \sum_{rs \in \mathcal{RS}} \{ \mathbf{C}_{rs}^\top \mathbf{n}_{rs} - H_{rs}^c(\mathbf{n}_{rs}) \} + \sum_{o \in \mathcal{O}} \left\{ \sum_{ij \in \mathcal{L}_v} t^\top \mathbf{x}^o - H^o(\mathbf{x}^o) \right\} \quad (23)$$

$$\text{s.t.} \quad n_{rs}^1 = \sum_{o \in \mathcal{O}} x_{rs}^o \quad \forall rs \in \mathcal{RS} \quad (24)$$

$$\sum_{i \in \mathcal{NI}_l} x_{il}^o - \sum_{j \in \mathcal{NO}_l} x_{lj}^o = \begin{cases} q_{od} & \text{if } l \in \mathcal{D} \\ 0 & \text{otherwise} \end{cases} \quad \forall l \in \{ \mathcal{R}, \mathcal{S}, \mathcal{D} \} \quad (25)$$

ここで、 $\mathbf{x}^o \equiv \{x_{ij}^o\}$ は起点別リンク交通量のベクトルである。具体的には、 x_{ij}^o は G_v 上で $o \in \mathcal{O}$ を出発し、リンク $ij \in \mathcal{L}_v$ を通行するドライバー数を表す。また、 $\mathcal{NI}_l(\mathcal{NO}_l)$ 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。 $H^o(\mathbf{x}^o)$ はエントロピー関数であり、以下の式で定義される：

$$H^o(\mathbf{x}^o) = -\frac{1}{\theta_p} \sum_{ij \in \mathcal{L}_v} x_{ij}^o \ln \frac{x_{ij}^o}{\sum_{i \in \mathcal{NI}_j} x_{ij}^o} \quad (26)$$

都市ネットワークのノード数 $|\mathcal{N}|$ を用いると、経路変数は最大で $|\mathcal{N}|^4$ のオーダーになるのに対し、起点別リンク変数は最大でも $|\mathcal{N}|^3$ のオーダーである。すなわち、問題規模が大幅 $(1/|\mathcal{N}|)$ に縮小される。

[SO-L] の双対問題は、より効率的なアルゴリズムの適用が可能な、以下の [SO-D] に帰着する：

[SO-D]

$$\max_{\mathbf{u}} TS(\mathbf{u}) = \bar{\mathbf{n}}^\top \mathbf{V}(\mathbf{u}) + \mathbf{q}^\top \boldsymbol{\mu}(\mathbf{u}) \quad (27)$$

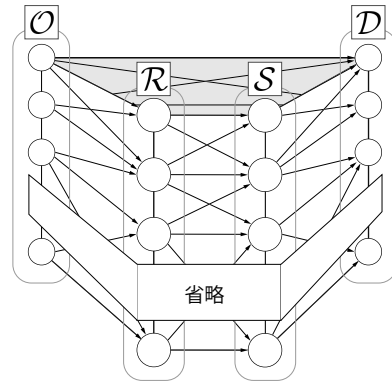


図-2 仮想ネットワーク G_v

ここで、 $\bar{\mathbf{n}} \equiv \{\bar{n}_{rs}\}$, $\mathbf{q} \equiv \{q_{od}\}$ は rs 荷主の総数および od ドライバーの総数を指し、変数 u_{rs} は rs タスクの価格(報酬額)である。また、 $\mathbf{V}(\mathbf{u}) \equiv \{V_{rs}(u_{rs})\}$, $\boldsymbol{\mu}(\mathbf{u}) \equiv \{\mu_{od}(\mathbf{u})\}$ はそれぞれ、タスク価格が \mathbf{u} の時の rs 荷主と od ドライバー期待最小費用であり、以下の式で定式化される：

$$V_{rs}(u_{rs}) = -\frac{1}{\theta_c} \ln \sum_k \exp\{-\theta_c(C_{rs}^k + k u_{rs})\} \quad (28)$$

$$\mu_{os}(\mathbf{u}) = -\frac{1}{\theta_p} \ln \sum_{r \in \mathcal{R}} \exp\{-\theta_p(t_{or} + t_{rs} - u_{rs})\} \quad (29)$$

$$\mu_{od}(\mathbf{u}) = -\frac{1}{\theta_p} \ln \left[\sum_{s \in \mathcal{S}} \exp[-\theta_p\{\mu_{os}(\mathbf{u}) + t_{sd}\}] + \exp(-\theta_p t_{od}) \right] \quad (30)$$

ここで、目的関数の第一項および第二項はそれぞれ配送業務に対する需要関数と供給関数を価格 \mathbf{u} で積分したものに对应する。すなわち、目的関数値は参加者の総余剰と捉えることができ、[SO-D] は総余剰最大化問題となる。

(3) タスク配分決定アルゴリズム

提案アルゴリズムでは、[SO-D] の解を利用し、タスク配分 \mathbf{n} , \mathbf{f} を決定する。これは、[SO-D] が無制約問題であり、効率的に解を導出できるからである。

本研究では、[SO-D] を解くアルゴリズムとして、Nesterov(1983)¹⁰⁾ が提案した加速勾配法を用いた。加速勾配法は、最悪収束率の観点から最適な 1 次法アルゴリズム² である。通常、最適化問題を解くアルゴリズムとしては、1 次法より 2 次法³ のほうが効率的であるが、[SO-D] は次元が大きく、ヘッセ行列の計算量およびメモリ使用量が膨大になるため、2 次法の適用は非効率的である。以上の理由により、加速勾配法を採用した。実際には、加速勾配法に近接勾配法を組み合わせた FISTA¹¹⁾

² 目的関数の勾配を用いる最適化アルゴリズム。

³ 目的関数のヘッセ行列を用いる最適化アルゴリズム。

を適用し、さらに、最適解付近での収束を早める効果のあるリスタート・メソッド¹²⁾を用いた。具体的なアルゴリズムを Algorithm1 に示す。なお、本研究では各種パラメータの値を、 $\epsilon = 1.0, \eta = 1.1, L_0 = 1.0$ と設定した。

加速勾配法では、目的関数 TS およびその勾配 ∇TS の評価速度がその性能の決め手となる。目的関数値は式 (27) によって計算でき、勾配 $\nabla TS(\mathbf{u})$ は以下の式で計算できる：

$$\frac{\partial TS(\mathbf{u})}{\partial u_{rs}} = -n_{rs}^1(u_{rs}) + \sum_{o \in \mathcal{O}} x_{rs}^o(\mathbf{u}) \quad (31)$$

$$n_{rs}^1(\mathbf{u}) = \bar{n}_{rs} \frac{\exp\{-\theta_c(C_{rs}^1 + u_{rs})\}}{\sum_k \exp\{-\theta_c(C_{rs}^k + ku_{rs})\}} \quad (32)$$

ここで、 $x_{rs}^o(\mathbf{u})$ は、リンク rs にコスト $-u_{rs}$ を追加した仮想ネットワーク G_v において LOGIT 配分を行った際の、 o を出発し、リンク $rs \in \mathcal{RS}$ を通行する人数である。したがって、 ∇TS は LOGIT 配分アルゴリズムによって計算できるが、前章で行った問題規模の縮小を活かすような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アルゴリズムが望ましい。したがって、本研究では、この要請を満たす Markov 連鎖配分¹³⁾を適用した。具体的なアルゴリズムを Algorithm2 に示す。ここで、Algorithm2 中の p_{rs}^o, p_{sd}^o はそれぞれ、 o を出発するドライバーの内、リンク rs, sd を通行する割合を示し、以下の式で与えられる：

$$p_{rs}^o(\mathbf{u}) = \exp[-\theta_p\{t_{or} + t_{rs} - \mu_{os}(\mathbf{u})\}] \quad (33)$$

$$p_{sd}^o(\mathbf{u}) = \exp[-\theta_p\{\mu_{os}(\mathbf{u}) + t_{sd} - \mu_{od}(\mathbf{u})\}] \quad (34)$$

以上のアルゴリズムを用いることにより、マスター問題は効率的に計算することができる。具体的には、[SO-D] の最適解 \mathbf{u}^* から以下の式によってマスター問題の最適配分 \mathbf{n}, \mathbf{f} を決定することができる：

$$n_{rs}^k = \bar{n}_{rs} \frac{\exp\{-\theta_c(C_{rs}^k + ku_{rs}^*)\}}{\sum_{k \in \{0,1\}} \exp\{-\theta_c(C_{rs}^k + ku_{rs}^*)\}} \quad \forall rs \in \mathcal{RS}, k \in \{0,1\} \quad (35)$$

$$f_{od}^{rs} = q_{od} \frac{\exp\{-\theta_p(C_{od}^{rs} - u_{rs}^*)\}}{\sum_{rs \in \mathcal{RS}'} \exp\{-\theta_p(C_{od}^{rs} - u_{rs}^*)\}} \quad \forall od \in \mathcal{OD}, rs \in \mathcal{RS}' \quad (36)$$

従って、課題 2 が解決される

5. 数値実験

本研究では、提案メカニズムの計算効率性を検証するため、ランダムに作成したネットワーク上で数値実験を行った。具体的には、以下の三つの値を変化させ、提案手法における求解の所要時間を評価した：

- 実験 1：平均 OD ドライバー数 ($|\mathcal{B}|/|\mathcal{N}|^2$)

Algorithm 1 メインアルゴリズム : FISTA

Require: $\epsilon > 0, \eta > 1, L_0 > 0, TS, \nabla TS$

Ensure: \mathbf{u}^*

```

1:  $\mathbf{v}_1 \leftarrow \mathbf{u}_0, t_1 \leftarrow 1$ 
2: while  $k = 1, 2, \dots$  do
3:   while  $\iota = 0, 1, \dots$  do
4:      $F = TS(\mathbf{v}_k + \frac{1}{\eta^\iota L_{k-1}} \nabla TS(\mathbf{v}_k))$ 
5:      $Q = TS(\mathbf{v}_k) + \frac{1}{2\eta^\iota L_{k-1}} \|\nabla TS(\mathbf{v}_k)\|^2$ 
6:     if  $F \geq Q$  then
7:       break.
8:     end if
9:   end while
10:   $L_k = \eta^\iota L_{k-1}$ 
11:   $\mathbf{u}_k = \mathbf{v}_k + \frac{1}{L_k} \nabla TS(\mathbf{v}_k)$ 
12:  if  $\|\nabla TS(\mathbf{u}_k)\| < \epsilon$  then
13:    break.
14:  end if
15:  if  $\nabla TS(\mathbf{x}_k) \cdot (\mathbf{x}_k - \mathbf{x}_{k-1}) < 0$  then
16:     $t_k = 1$ 
17:  end if
18:   $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$ 
19:   $\mathbf{v}_{k+1} = \mathbf{u}_k + \frac{t_k - 1}{t_{k+1}} (\mathbf{x}_k - \mathbf{x}_{k-1})$ 
20: end while
21: return  $\mathbf{u}^* = \mathbf{u}_k$ 

```

Algorithm 2 サブアルゴリズム : Markov 連鎖配分

Require: \mathbf{u}

Ensure: $TS(\mathbf{u}), \nabla TS(\mathbf{u})$

```

1:  $\mu_{os}, \mu_{od}, V_{rs}, TS$  を計算. By (29), (30), (28), (27)
2:  $p_{rs}^o, p_{sd}^o$  を計算. By (33), (34)
3: for  $os \in \mathcal{OS}$  do
4:    $X_s^o = \sum_{d \in \mathcal{D}} q_{od} p_{sd}^o$ 
5: end for
6: for  $rs \in \mathcal{RS}$  do
7:    $n_{rs}^1$  を計算. By (32)
8:    $G_{rs} = -n_{rs}^1 + \sum_{o \in \mathcal{O}} p_{rs}^o X_s^o$ 
9: end for
10: return  $TC(\mathbf{u}), \nabla TC(\mathbf{u}) = \{G_{rs}\}_{\forall rs}$ 

```

- 実験 2：都市ネットワークのノード数 ($|\mathcal{N}|$)
- 実験 3：タスクの起点数 ($|\mathcal{R}|$)

具体的な設定値を表-1 に示す。なお、簡単のため、荷主の総数はドライバーの総数と同じであるとした。また、荷主およびドライバーの LOGIT パラメータ (θ_c, θ_p) はいずれも 5.0 と設定した。実験は各設定ごとに 10 回ずつ行った。

図-3 は数値実験結果である。図中のエラーバーはそ

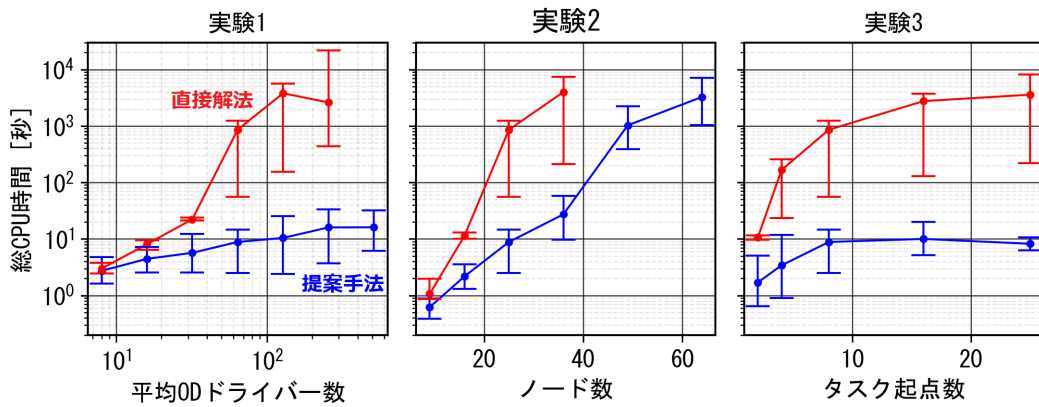


図-3 数値実験結果（求解時間）

表-1 実験の設定

	$ \mathcal{B} / \mathcal{N} ^2$	$ \mathcal{N} $	$ \mathcal{R} $
実験 1	操作	25	8
実験 2	64	操作	8
実験 3	64	25	操作

それぞれ 10 回の実験における最大値と最小値を示している。この結果から、大規模なシステムにおいては、提案手法によって計算速度が大幅 (100 倍以上) に改善されたことが分かる。また、直接解法で結果が表示されていない点は、メモリ不足により直接解法では計算できなかったことを表している。従って、提案手法では大幅なメモリ削減効果があることが示された。すなわち、提案手法は、時間およびメモリ使用量の観点から、直接解法より優位であることが示された。

6. おわりに

本研究では、大規模な CSD システムでも適用可能であり、需給の価格弾力性を考慮できるマッチング手法を開発した。通常、大規模な CSD システムでは最適なマッチングを決定する問題が非常に大規模となる課題が生じる。そこで、本研究で提案するメカニズムでは、参加者を OD ペアごとの小集合 $\mathcal{A}_{rs}, \mathcal{B}_{od}$ へ分割し、二段階でマッチングを決定する：第一段階では、実行するタスク数 n と各ドライバー集合へのタスク配分 \mathbf{f} を決定し、第二段階では、各小集合内で VCG オークションを行うことによって、荷主のタスク委託有無およびドライバーの実行タスクを決定する。この分解により、第二段階でのマッチング決定問題は非常に小規模なものとする事ができる。さらに、第一段階のタスク配分決定問題に対しても効率的な解法を開発することに

より、計算時間を大幅に改善する事に成功した。

さらに、数値実験では、提案するマッチング決定手法について求解の所要時間の観点から分析を行い、直接解法に対する優位性を明らかにした。具体的には、大規模なシステムにおいては 1/100 以下の時間で解を求めることが確認された。また、提案手法では大幅なメモリ削減効果があることも示された。

本研究で想定したモデルは基本的なものであり、様々な拡張が考えられる。例えば、時間に関する制約を含むモデルである。実際のタスクは配送時間に制約がある場合がほとんどであるが、今回のモデルではこれを考慮していない。また、サービス水準を導入したモデルも考えられる。本研究では、全てのタスクは同質であるとしているが、実際のタスクは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。以上のような拡張モデルを考え、より現実的なシステムを構築して行くことが今後の課題となる。

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) Archetti, C., Savelsbergh, M., and Speranza, M. G.: The vehicle routing problem with occasional drivers, *European Journal of Operational Research*, Vol.254, No.2, pp.472–480, 2016.
- 2) Arslan, A. M., Agatz, N., Kroon, L., and Zuidwijk, R.: Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers, *Transportation Science*, Vol.53, No.1, pp.222–235, 2019.
- 3) Chen, P. and Chankov, S.: Crowdsourced delivery for last-mile distribution: An agent-based modelling and simulation approach, *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 1271–1275, IEEE, 2017.
- 4) Wang, Y., Zhang, D., Liu, Q., Shen, F., and Lee, L. H.: Towards enhancing the last-mile delivery: An

- effective crowd-tasking model with scalable solutions, *Transportation Research Part E: Logistics and Transportation Review*, Vol.93, pp.279–293, 2016.
- 5) 渡邊大樹: クラウドソーシング配送システムにおける効率的マッチング, 卒業論文, 東北大学, 工学部建築・社会環境工学科, 2020.
 - 6) Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders, *The Journal of finance*, Vol.16, No.1, pp.8–37, 1961.
 - 7) Clarke, E. H.: Multipart pricing of public goods, *Public choice*, pp. 17–33, 1971.
 - 8) Groves, T.: Incentives in teams, *Econometrica: Journal of the Econometric Society*, pp. 617–631, 1973.
 - 9) Akamatsu, T.: Decomposition of path choice entropy in general transport networks, *Transportation Science*, Vol.31, No.4, pp.349–362, 1997.
 - 10) 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.
 - 11) Beck, A. and Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM journal on imaging sciences*, Vol.2, No.1, pp.183–202, 2009.
 - 12) O’donoghue, B. and Candes, E.: Adaptive restart for accelerated gradient schemes, *Foundations of computational mathematics*, Vol.15, No.3, pp.715–732, 2015.
 - 13) Akamatsu, T.: Cyclic flows, markov process and stochastic traffic assignment, *Transportation Research Part B: Methodological*, Vol.30, No.5, pp.369–386, 1996.