

クラウド・ソーシング配送問題における 効率的マッチング

渡邊 大樹¹・赤松 隆²

¹非会員 東北大学 工学部建築・社会環境工学科 (〒 980-8579 仙台市青葉区荒巻字青葉 6-6-06-408)

E-mail: taiki.watanabe.p3@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

都市内を日々通行する車両を、宅配など”ラストワンマイル”の配送に利用するシステムである”クラウドソーシング・デリバリー”は、E-Commerceの普及による配送需要の増加に対処する手段の一つとして期待される。本研究では、クラウドソーシング配送システムにおいて、クラウドソースであるドライバーと配送タスクのマッチングをオークション市場を通して決定するシステムを考える。まず、市場均衡により実現するマッチングが、社会的に最適であることを示す。次に、オークション市場において解くべき落札者決定問題が非常に大規模になることを述べ、その問題の緩和・効率的求解方法の方向性を示す。最後に、その最適化問題の効率的解法を提案する。

Key Words: クラウドソーシング・デリバリー, 最適輸送問題, ラストワンマイル

1. はじめに

近年、E-Commerceの急速な発展に伴い、都市内の集配所から注文者への”ラストワンマイル”の配送需要が急増している。これにより、配送量の増加に加えて、小口・多頻度化という需要の”分散”が急速に進んでおり、これに現在の配送システムは対応しきれなくなりつつある。このため、この増加・分散する配送需要に対応できる、新たな配送システムの必要性が高まっている。

このような分散する配送業務を効率よく遂行する新たなシステムとして、”クラウドソーシングデリバリー(CSD)”が挙げられる。これは、都市内を通行している一般車両の余剰スペースに貨物を積載して配送し、ドライバーに対して賃金を支払うものである。このシステムは、トラックに大量の貨物を積載して一度に配送する従来の配送システムと比較して、一度に配送する貨物は少量であり、配送の回数が多い”分散した”配送システムであるから、分散する配送需要に対応できるシステムであると言える。また、配送がなくても行われるトリップを利用して配送を行うので、配送のためだけにトラックを走らせる従来の配送に比べて効率的に配送を行える可能性を持っている。

ただし、このシステムを効率的に運用するためには、配送意思をもつドライバーと配送業務を適切に(i.e. ドライバーの回り道が少なくなるように)マッチングするメカニズムを構築することが不可欠である。CSDシステ

ムにおけるマッチングに関する研究としては、Archetti et al.¹⁾, Arslan et al.²⁾などが挙げられるが、これらは管理者がドライバーに個人情報(e.g. 目的地, 希望到着時刻など)を開示させ、それを元に管理者が中央集権的にマッチングを決定することを想定している。このようなシステムは、ドライバーの(観測不可能な)私的選好を十分に考慮できず、社会的に非効率なマッチングが実現しうるという欠点がある。そうでないマッチングシステムとして、Allahviranloo³⁾はオークションによってマッチングを決定するシステムを提案したが、そのシステムで実現するマッチングが効率的なものかどうかは議論されていない。以上の議論から、(a)ドライバーの私的選好をマッチングに反映でき、かつ(b)社会的に最も効率的なマッチングを実現できるCSDシステムは、未だ提案されていない。

そこで本研究では、(a),(b)を同時に実現するCSDシステムの設計を目的とする。具体的には、マッチングと配送タスクの価格(ドライバーが受け取る額)をオークション市場で決定するCSDシステムを考える。マッチングを市場で決定すれば、ドライバーの私的効用は配送タスクの均衡価格に反映され、(a)が達成される。

その上で、本研究は以下の3つの段階に分けられる：(1)このシステムで実現するマッチング(i.e. 市場均衡状態)が社会的に最も効率的であることを示す、(2)オークションシステムを運用するための効率的解法を開発する、(3)オークションシステムの設計を行う。(1)は、(b)を達成できることを示すために必要である。(2)が

必要である理由は、CSD システムの特性から、オークションシステムの中で解くべき問題の規模が非常に大きいためである。(3)が必要である理由は、オークションシステムを正しく設計しなければ、オークションの参加者が戦略的な行動をとり、(1)で効率的であることを示した市場均衡状態が実現しない可能性があるためである。

本稿では、これらの目的のうち(1)、(2)を達成し、(3)の方向性を示した。具体的にはまず、市場均衡状態を定式化し、社会的に最適なマッチングとの比較を通して、均衡状態が社会的に最適なマッチングを実現できることを示し(1)を達成した。次に、オークションシステムとしてVCGメカニズムを利用することと、その中で解くべき問題(前述の通り、非常に大規模)を解く方法を提案し、(3)の方向性を示した。最後に、その方法に基づくアルゴリズムを開発し、(2)を達成した。

2. モデル

(1) 基本設定

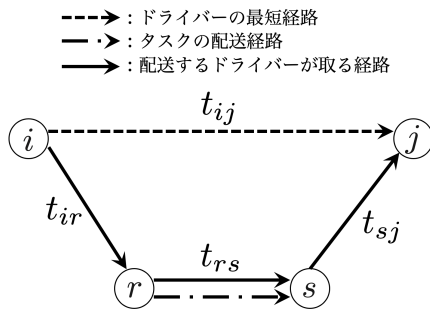


図-1 ドライバーとタスクの移動

ある都市内の交通ネットワークを考える。ある荷物の配送業務配送元のノード r と配送先のノード s の組 (r, s) で区別される。以下では、配送元が r 、配送先が s である配送業務を”タスク rs ”と呼ぶ。また、配送タスクはその OD ペア (配送元と配送先) ごとに、必要とするドライバーの数と価格が決まっている。タスク rs の必要人数と価格を、それぞれ n_{rs}, u_{rs} と表記する。

一方、この道路ネットワーク上では日々車両が通行しており、ここに CSD システムを導入した時、通行する車両の一部がそのシステムへの参加 (配送を行う) 意思を持つ。車両はその OD ペア (i, j) で区別される。以下では、配送の意思を持つ OD ペア ij のドライバーを”ドライバー ij ”と呼び、ドライバー ij の総数 (ij の OD 交通量のうち配送意思を持つ人数) を q_{ij} と表記する。また、本稿全体を通して、以下の仮定を置く。

仮定 1. 都市道路ネットワークの各リンクの所要時間

表-1 記号の定義

記号	定義
i	ドライバーの起点ノード
j	ドライバーの終点ノード
r	タスクの起点ノード
s	タスクの終点ノード
q_{ij}	OD ペア ij のドライバーの数
n_{rs}	OD ペア rs のタスクの必要人数
u_{rs}	タスクの価格
x_{ij}^{rs}	ドライバー ij のうちタスク rs を行う人数
t_{ab}	ノード a, b 間の最短経路費用
c_{ij}^{rs}	ドライバー ij がタスク rs を実行することで増加する交通費用

は、配送意思を持つドライバーの数に依存しない定数である。

仮定 2. ドライバーの総数は、配送タスクの総必要人数に一致する。

仮定 1 は、配送意思を持つドライバーの数が全交通量に対して十分に小さいと考えれば妥当な仮定である。また仮定 2 も、ドライバーが足りない場合はシステム管理主体が代わりに配送ドライバーを提供し、過剰な場合は参加者を適切な基準 (e.g. 先着順、過去の業務履歴等) によって選抜すれば、実現することができる。

CSD システムは、ドライバーとタスクのマッチング (どのドライバーがどのタスクを行うか) を決めるシステムであると言える。マッチングを表現するため、ドライバー ij のうち、タスク rs を行う人数を x_{ij}^{rs} と表記する。この値を全ての ij, rs に対して決定することが、マッチングの決定に相当する。また、ドライバー ij がタスク rs を実行した時にとる回り道に費やされる費用を c_{ij}^{rs} と表記する。この費用は、ドライバーがタスクを行う場合の交通費用から、行わない場合の費用を引くことで得られる。すなわち、ノード ab 間の最短経路費用を t_{ab} として

$$c_{ij}^{rs} = t_{ir} + t_{rs} + t_{sj} - t_{ij} \quad (1)$$

と表される。

(2) 市場による価格調整

本研究で取り扱うモデルでは、各タスクの価格、およびドライバーとタスクのマッチングを、ドライバーと管理者の間の市場取引により決定する。その結果、あるタスクを行う意思を持つドライバーの数と、そのタスクの必要人数が一致するように、価格が調整される。ドライバーは、自らの得られる効用が最大になるタス

クを選択する。

この価格調整により市場均衡状態が実現したとき、以下の条件が成立する。

1. ドライバーのタスク選択に関する均衡条件

均衡状態では、”どのドライバーも自分だけがタスクを変更しても効用を改善できない”均衡状態が実現する。ドライバー ij がタスク rs を実行するときの効用は、価格から交通費用を引いた値 $u_{rs} - c_{ij}^{rs}$ で表されるから、この均衡状態は以下のように定式化される：

$$\begin{cases} u_{rs} - c_{ij}^{rs} = v_{ij} & \text{if } x_{ij}^{rs} > 0 \\ u_{rs} - c_{ij}^{rs} \leq v_{ij} & \text{if } x_{ij}^{rs} = 0 \end{cases} \quad \forall i, j, r, s \quad (2)$$

where $v_{ij} = \max_{r,s} \{u_{rs} - c_{ij}^{rs}\}$

2. ドライバーの保存則

$$\sum_r \sum_s x_{ij}^{rs} = q_{ij} \quad (3)$$

3. 需給均衡条件

均衡状態において、タスクに正の価格が付いていれば需給が均衡しており、そうでなければドライバーの供給過剰である：

$$\begin{cases} \sum_i \sum_j x_{ij}^{rs} = n_{rs} & \text{if } u_{rs} > 0 \\ \sum_i \sum_j x_{ij}^{rs} \geq n_{rs} & \text{if } u_{rs} = 0 \end{cases} \quad \forall r, s \quad (4)$$

ただし、本研究で設定している仮定 1 より、ドライバーは全てタスクに割り当てられ、供給過剰は発生しない。したがって、この条件は実際には $\sum_i \sum_j x_{ij}^{rs} = n_{rs}$ と等価である。

3. 市場均衡状態の効率性

市場均衡状態の社会的な効率性を調べるため、比較対象として、社会的に最も効率的な配分を求める問題を定式化する。CSD システムの効率性は、全ての配送を行うのに必要な社会的費用の総和で測ることができる。ドライバー ij がタスク rs を行う際に生じる社会的費用は、ドライバーがタスク実行のために取る回り道の距離の増加量 $c_{ij}^{rs} = t_{ir} + t_{rs} + t_{sj} - t_{ij}$ である。したがって、総社会的費用を最小化するマッチングは、以下の最適化問題 [SO] の解となる：

[SO]

$$\min_{\mathbf{x}} z_{SO}(\mathbf{x}) = \sum_i \sum_j \sum_r \sum_s c_{ij}^{rs} x_{ij}^{rs} \quad (5)$$

$$\text{s.t. } \sum_r \sum_s x_{ij}^{rs} = q_{ij} \quad \forall i, j \quad (6)$$

$$\sum_i \sum_j x_{ij}^{rs} = n_{rs} \quad \forall r, s \quad (7)$$

$$x_{ij}^{rs} \geq 0 \quad \forall i, j, r, s \quad (8)$$

この問題の最適解が満たすべき最適性条件 (KKT 条件) は、以下の 3 つの条件を満たす \mathbf{u}, \mathbf{v} が存在することで

ある。

$$\begin{cases} u_{rs} - c_{ij}^{rs} = v_{ij} & \text{if } x_{ij}^{rs} > 0 \\ u_{rs} - c_{ij}^{rs} \leq v_{ij} & \text{if } x_{ij}^{rs} = 0 \end{cases} \quad \forall i, j, r, s \quad (9)$$

$$\sum_r \sum_s x_{ij}^{rs} = q_{ij} \quad \forall i, j \quad (10)$$

$$\sum_i \sum_j x_{ij}^{rs} = n_{rs} \quad \forall r, s \quad (11)$$

\mathbf{u}, \mathbf{v} はそれぞれ制約条件 (6),(7) である。ここで、 u_{rs} をタスク rs の市場均衡価格、 v_{ij} をドライバー ij の市場均衡効用 $\max_{r,s} \{u_{rs} - c_{ij}^{rs}\}$ とすると、これらの 3 条件は全て満たされ、さらに条件 (9),(10),(11) が、市場均衡条件 (2),(3),(4) と等価であることがわかる。したがって、以下の命題が成立する。

命題 1. 市場均衡状態において実現するマッチングは、社会的費用を最小にするものである。

4. オークション・メカニズム

前章で市場均衡状態が社会的に最適なマッチングであることが示された。ただし、もしオークション市場においてドライバーが戦略的な行動¹を取るインセンティブが働く場合、市場均衡状態は必ずしも実現しない。そのため、効率的なマッチングを市場によって実現するためには、このようなことが起こらないオークション・メカニズムを設計する必要がある。本稿ではオークション・メカニズムの設計は行わないが、次章以降の議論のため、その方向性のみ示しておく。

市場均衡状態を実現させる望ましいメカニズムとして、VCG メカニズムが挙げられる。これは Vickrey⁴, Clarke⁵, Groves⁶ によって提案されたオークション・メカニズムであり、以下の望ましい性質を持つことが知られている：(1) 効率的な資源配分を実現できる。(2) 参加者に戦略的な行動をとるインセンティブが存在しない。

このメカニズムは、以下のような手続きでマッチングとタスクの価格決定を行う。

1. 入札：ドライバーは全てのタスクに対して入札を行う
2. マッチング決定：オークション管理者は、自らの支払う賃金が最小になるようにマッチングを決定する
3. 価格決定：タスクの価格は、その落札者が入札することによって生じる他者の社会的余剰の減少分とする。

したがって、落札者決定フェイズにおいて、以下の問題 [Match] をとく必要がある。

¹ 不当に高い希望報酬額を申告することで利得を増加させるなど

[Match]

$$\min \sum_a \sum_{r,s} \hat{c}_a^{rs} y_a^{rs} \quad (12)$$

$$\text{s.t.} \sum_a y_a^{rs} = n_{rs} \quad \forall r, s \quad (13)$$

$$\sum_{rs} y_a^{rs} = 1 \quad \forall a \quad (14)$$

$$y_a^{rs} \in \{0, 1\} \quad \forall r, s, a \quad (15)$$

ここで、 a は個々のドライバーを表し、 \hat{c}_a^{rs} はドライバー a のタスク rs への入札額、 y_a^{rs} はドライバー a がタスク rs を落札する場合に 1、そうでない場合に 0 となる変数である。しかし、CSD は、各タスク (オークションで取引される”財”) が、ノード二つのペアで定義され、タスクの総種類数は最悪の場合、ノード総数の二乗に比例して増加していくという性質がある。したがってこの問題は、CSD のこの性質から、(ドライバーの人数) \times (タスクの OD ペア数) という膨大な数の変数を持つ組み合わせ最適化問題となり、現実的な時間で解くことは難しい。そこで、この問題に代わる、現実的な時間で解ける勝者決定問題が必要になる。

問題を解きやすくするため、ドライバーの OD ペアが分かっていると仮定し、ドライバーを OD ペアごとに集計する。

ドライバー a 全ての集合を \mathcal{A} とし、そのうち OD ペアが ij のもののみを含む部分集合を \mathcal{A}_{ij} とする。次に、ドライバーの入札額の集合 $\{\hat{c}_a^{rs} | a \in \mathcal{A}_{ij}\}$ から、何らかの代表値 (e.g. 平均など) \hat{c}_{ij}^{rs} を得る。そして、ドライバーの入札額を OD ペア ij ごとの代表値に揃える。すなわち、

$$\hat{c}_a^{rs} = \hat{c}_{ij}^{rs} \quad \forall a \in \mathcal{A}_{ij} \quad \forall i, j \quad (16)$$

であるとする。さらに、ドライバーの落札を表す変数 y_a^{rs} も OD ペアごとに集約して $x_{ij}^{rs} = \sum_{a \in \mathcal{A}_{ij}} y_a^{rs}$ とし、さらにこの整数変数 x_{ij}^{rs} を実数変数とみなす。以上の操作により、[Match] は [SO] に帰着される。[SO] は [Match] より変数の少ない実数計画問題であるから、より解きやすい問題である。したがって、[SO] を [Match] の代わりに解き、その最適解 (OD ペアごとの落札者数) を各 OD ペアのドライバーに分配するという方法は、現実的な落札者決定方法であると考えられる。

ただし、[SO] も CSD の性質上、変数 x_{ij}^{rs} がノードを表す 4 種類の添字 i, j, r, s によって区別されるため、依然として変数の数が多く、解き難い問題である。²[SO] は落札者決定に利用できるだけでなく、市場均衡状態でのマッチングの性質を分析する際にも必要になるた

² 実際、もしそれぞれの添字が N の要素数 N だけあるとすれば変数 N^4 個の LP となる。LP に対する最も速いとされている主双対内点法アルゴリズム⁷⁾ は変数の数 n 、LP 全体に必要な記憶領域 L を用いて、計算量が $O(n^3 L)$ であり、問題 [SO] に対して適用するとその計算量は $O(N^{12} L)$ となってしまう。

め、今後の研究展開において重要な問題である。そのため、次章以降では [SO] の効率的解法を提案する。

5. 効率的解法開発の方針

(1) 多種最小費用流問題への変形

本章では、最適化問題 [SO] の目的関数を、多種最小費用流問題の形式に変換する。これにより変数の数を減らすことができ、かつ単一種最小費用流問題の重ね合わせに分解することで効率的に解くことができる。

そのために、次のような仮想ネットワーク [VN] を構築する：まず、オリジナルのネットワークを 4 つコピーし、それぞれをドライバーの起終点 i, j 、タスクの起終点 r, s に対応させる。次に、全ての $i-r$ 間、 $s-j$ 間と、タスクの存在する $r-s$ 間をこの向きの有向リンクでつなぐ。例えば、オリジナルネットワークが図 2 だった場合、[VN] は図 3 のようなものである。また、 i, j, r, s に対応させたネットワークのノード集合をそれぞれ $\mathcal{I}, \mathcal{J}, \mathcal{R}, \mathcal{S}$ としている。

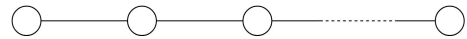


図-2 オリジナルネットワーク

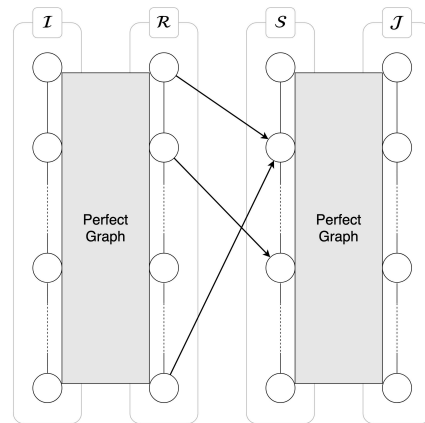


図-3 仮想ネットワーク [VN]

次に、[SO] が [VN] 上の多種最小費用流問題とみなせることを示す。[VN] における $i \in \mathcal{I}$ から $j \in \mathcal{J}$ への経路は $i-r-s-j \forall r, s$ のみである。したがって、[SO] の制約条件 (6) は、 x_{ij}^{rs} を経路 $i-r-s-j$ の流量、 q_{ij} をフローの総量と解釈すれば、[VN] 上のフロー保存則とみなせる。

一方、制約条件 (7) は、リンク rs の容量制約とみなせる。

最後に, [SO] の目的関数を変形すると,

$$\sum_{i,j} \sum_{r,s} c_{ij}^{rs} x_{ij}^{rs} \quad (17)$$

$$= \sum_{i,j} \sum_{r,s} [t_{ir} + t_{sj}] x_{ij}^{rs} + \sum_{r,s} t_{rs} \left(\sum_{i,j} x_{ij}^{rs} \right) - \sum_{i,j} t_{ij} \left(\sum_{r,s} x_{ij}^{rs} \right) \quad (18)$$

となる. このうち第二項と第三項は制約条件 (7),(6) より定数となるため, 目的関数から除外できる. したがって, (18) はリンク $i-r, r-s, s-j$ の交通費用をそれぞれ $t_{ir}, 0, t_{sj}$ と設定すると, [VN] 上のフローの総費用を表す.

以上より, [SO] は [VN] 上の最小費用流問題と解釈できる.

さて, この最小費用流問題は, 起点ごとに区別されたリンクフローを変数として等価に表現できる. 具体的には, 経路変数ベースのフロー保存則 (6) を, 起点別リンク変数ベースの等価な表現に変更することで, 以下の問題 [SO-Trans] に変形される.

[SO-Trans]

$$\min_{\mathbf{x}} z_T(\mathbf{x}) = \sum_i \sum_r t_{ir} x_{ir} + \sum_i \sum_s \sum_j t_{sj} x_{sj}^i \quad (19)$$

$$\text{s.t. } x_{ir} - \sum_s x_{rs}^i = 0 \quad \forall r, i \quad (20)$$

$$\sum_r x_{rs}^i - \sum_j x_{sj}^i = 0 \quad \forall s, i \quad (21)$$

$$\sum_s x_{sj}^i = q_{ij} \quad \forall j, i \quad (22)$$

$$\sum_i x_{rs}^i = n_{rs} \quad \forall r, s \quad (23)$$

以降の記述を簡潔にするため, この問題をベクトル-行列形式で書き直しておく.

$$\min_{\mathbf{x}} \sum_i \mathbf{t}^\top \mathbf{x}^i \quad (24)$$

$$\text{s.t. } \mathbf{A} \mathbf{x}^i = \mathbf{b}^i \quad \forall i \quad (25)$$

$$\sum_i \mathbf{x}_{RS}^i = \mathbf{n} \quad (26)$$

$$\mathbf{x}^i \geq \mathbf{0} \quad \forall i \quad (27)$$

ここで, \mathbf{A} は [VN] のノード-リンク接続行列, \mathbf{x}_{RS}^i はノード集合 \mathcal{R} から \mathcal{S} へのリンクフローのみを含むベクトルである. フロー保存則 (20),(21),(22) が (25) に, 容量制約 (23) が (26) に, それぞれ対応している.

(2) 解法の基本的方針

詳細は後述するが, [SO-Trans] は, 単一起点 (single-commodity flow) 問題である場合は少ない計算量で解くことができる. そこで, [SO-Trans] を起点 i ごとに分

解された単一起点問題の重ね合わせにすることで, 効率的に解くことを考える. [SO-Trans] は, 容量制約条件 (26) 以外はもともと起点別に分解できる形になっているから, (26) を起点別に分解しさえすれば良い.

分解の方法としては以下の 2 通りが考えられる. ここでは, 分解の考え方のみを示し, 具体的な解法は 7 章で述べる.

a) Price-directive Decomposition

まず, 制約 (23) を Lagrange 緩和する. この制約の Lagrange 乗数を $\mathbf{u} = [\dots, u_{rs}, \dots]^\top$ として, [SO-Trans] は以下の [SO-P] と等価である.

[SO-P]

$$\max_{\mathbf{u}} \min_{\mathbf{x}} z_P(\mathbf{x}) = \sum_i \mathbf{t}_T^\top \mathbf{x}^i + \mathbf{u}^\top \left(\mathbf{n} - \sum_i \mathbf{x}_{RS}^i \right) \quad (28)$$

$$\text{s.t. } \mathbf{A} \mathbf{x}^i = \mathbf{b}^i \quad \forall i \quad (29)$$

$$\mathbf{x}^i \geq \mathbf{0} \quad \forall i \quad (30)$$

[SO-P] は, \mathbf{u} を定数と見て問題を解く下位問題と, 下位問題の最適値関数を最大化する \mathbf{u} を求める上位問題の二段階に階層化できる. そして, 下位問題は起点別に分割が可能である. 具体的には, 起点別に分解された下位問題 [SO-P-L] と上位問題 [SO-P-U] は, それぞれ以下の通りである.

[SO-P-L]

$$\min_{\mathbf{x}} z_{PL}^i(\mathbf{x}^i) = \mathbf{t}_P^\top \mathbf{x}^i \quad (31)$$

$$\text{s.t. } (29), (30) \quad (32)$$

[SO-P-U]

$$\max_{\mathbf{u}} z_P U(\mathbf{u}) = \sum_r \sum_s u_{rs} n_{rs} + \sum_i z_{PL}^{i*}(\mathbf{u}) \quad (33)$$

$$\text{where } z_{PL}^{i*}(\mathbf{u}) = \min_{\mathbf{x}} \{ z_{PL}^i(\mathbf{x}^i) \mid (29), (30) \} \quad (34)$$

ここで, \mathbf{t}_P は, \mathbf{t} の $r \in \mathcal{R}$ から $s \in \mathcal{S}$ へのリンクの費用を 0 から $-u_{rs}$ に変更したベクトルである. 下位問題は, $-\mathbf{u}$ をリンク rs の費用と考えれば, 容量制約のない単一種最小費用流問題であるから, よく知られた解法が存在する. 具体的な下位問題, 上位問題の解法は 7 章で詳説する.

b) Resource-directive Decomposition

まず, 制約条件 (26) の右辺 \mathbf{n} を, 起点 i ごとに分割することを考える. 具体的には, (26) を,

$$\mathbf{x}_{RS}^i = \mathbf{n}^i \quad \forall i \quad (35)$$

$$\sum_i \mathbf{n}^i = \mathbf{n} \quad (36)$$

$$\mathbf{1}^\top \mathbf{n}^i = \mathbf{1}^\top \mathbf{q}^i \quad \forall i \quad (37)$$

$$\mathbf{n}^i \geq \mathbf{0} \quad \forall i \quad (38)$$

に換えた問題を考える。ここで、条件 (37) は、起点 i から発生するドライバーが起点別に与えられた全てのタスクを完了できるための条件である。

これにより、問題 [SO-Trans] は以下の [SO-R] と等価である。

[SO-R]

$$\min_{\mathbf{n}^{(i)}} \min_{\mathbf{x}} \sum_i \mathbf{t}^\top \mathbf{x}^i \quad (39)$$

$$\text{s.t. } \mathbf{A}\mathbf{x}^i = \mathbf{b}^i \quad \forall i \quad (40)$$

$$\mathbf{x}_{RS}^i = \mathbf{n}^i \quad \forall i \quad (41)$$

$$\mathbf{x}^i \geq \mathbf{0} \quad \forall i \quad (42)$$

$$(36), (37), (38)$$

$$\text{where } \mathbf{n}^{(i)} = [\dots, \mathbf{n}^{i^\top}, \dots]^\top \quad (43)$$

さらに、この問題は、起点別容量 $\mathbf{n}^{(i)}$ を固定した下での最適解を求める下位問題と、その最適値を最小化する $\mathbf{n}^{(i)}$ を求める上位問題に階層化でき、下位問題は起点別に分解される。起点別に分解された下位問題 [SO-R-L] と上位問題 [SO-R-U] は、それぞれ以下の通りである。

[SO-R-L]

$$\min_{\mathbf{x}^i} z_{RL}^i(\mathbf{x}) = \mathbf{t}^\top \mathbf{x}^i \quad (44)$$

$$\text{s.t. } (40), (41), (42) \quad (45)$$

[SO-R-U]

$$\min_{\mathbf{n}^{(i)}} z_{RU}^i(\mathbf{n}^{(i)}) = \sum_i z_{RL}^{i*}(\mathbf{n}^{(i)}) \quad (46)$$

$$\text{s.t. } (36), (37), (38) \quad (47)$$

$$\text{where } z_{RL}^{i*}(\mathbf{n}^{(i)}) = \min_{\mathbf{x}} \{z_{RL}^i(\mathbf{x}) \mid (40), (41), (42)\} \quad (48)$$

下位問題は容量制約つき単一種最小費用流問題である。そして、[VN] におけるこの問題は、非常に簡単に解くことができる。その解法は 6 章で詳説する。

6. [SO-R-L] の解法

本章では、[SO-R-L] として登場した、[VN] における容量制約つき単一種最小費用流問題の解法を解説する。

問題を簡潔にするため、起点別容量制約 (41) をフロー保存則 (40) の一部をなす (20), (21) に代入する。これにより目的関数の $\sum_i \sum_r t_{ir} x_{ir}$ の部分は定数になるため目的関数から除かれ、また x_{ir} に関わるフロー保存則 (20) も不要になる。これにより [SO-R-L] は、以下の最適化問題 [SO-R-L'] に帰着する。

[SO-R-L']

$$\min_{\mathbf{x}} \sum_s \sum_j t_{sj} x_{sj}^i \quad (49)$$

$$\text{s.t. } \sum_j x_{sj}^i = \Delta_s^i \equiv \sum_r n_{rs}^i \quad \forall s \quad (50)$$

$$\sum_s x_{sj}^i = q_{ij} \quad \forall j \quad (51)$$

$$x_{sj}^i \geq 0 \quad \forall s, j \quad (52)$$

この問題は Hitchcock 輸送問題の形式であり、最適輸送問題の分野でこの形式の問題の解法に関して多くの研究蓄積がある。例えば、目的関数の係数 t_{sj} が Monge 性を有する、すなわち任意の添字の組 $s, s' \in \mathcal{S}, j, j' \in \mathcal{J}, s < s', j < j'$ に対して

$$t_{sj} + t_{s'j'} \leq t_{s'j} + t_{sj'} \quad (53)$$

が成立する場合、この問題は北西隅法と呼ばれるアルゴリズムで解くことができる。北西隅法は n 変数の問題を計算量 $O(n)$ で解くアルゴリズムであり、この問題の場合は変数が s, j に依存するため変数の数は N^2 であるから、[SO-Trans] は $O(N^2)$ で解けることになる。係数が Monge 性を有するかどうかはネットワークの形状に依存するが、例えば図 2 に示したような一次元ネットワークはこの性質を有する。また、係数が Monge 性を満たさない場合でも、各係数を Monge sequence と呼ばれる配列に並べ替えることができれば、同じように計算量 $O(N^2)$ で解くことができることも知られている。

それ以外の場合も、二重制約重力モデルの利用などで解を得られる。

7. [SO] の効率的アルゴリズム

この節では、5 章で示した起点別問題分解を利用した効率的なアルゴリズムを提案する。

(1) Price-directive decomposition

a) 下位問題

[SO-P-L] は LP であり、その最適値関数が一部微分不可能であるため、上位問題に対して Newton 法などの勾配法が利用できない。そこで、この目的関数に以下のようにエントロピー項を加えて目的関数を微分可能にする。以下で用いる θ は適当なパラメータである。

$$z_{PL}(\mathbf{x}) = \sum_i z_{PL}^i(\mathbf{x}) \quad (54)$$

$$z_{PL}^i(\mathbf{x}) = \mathbf{t}_P^\top \mathbf{x}^i - \frac{1}{\theta} H^i(\mathbf{x}) \quad (55)$$

ここで,

$$H^i(\mathbf{x}) = \sum_s \left(\sum_r x_{rs}^i \right) \ln \left(\sum_r x_{rs}^i \right) - \sum_r \sum_s x_{rs}^i \ln x_{rs}^i + \sum_j \left(\sum_s x_{sj}^i \right) \ln \left(\sum_s x_{sj}^i \right) - \sum_s \sum_j x_{sj}^i \ln x_{sj}^i \quad (56)$$

エントロピー項を導入する前の問題は All-or-Nothing 配分問題の等価最適化問題であるから, 導入後の問題は [VN] 上の LOGIT モデルに基づく確率的配分問題の等価最適化問題である. したがって, その目的関数の最適値は

$$z_L^i(\mathbf{x}^*) = \sum_j q_{ij} S_{ij}(\mathbf{u})$$

where

$$S_{ij}(\mathbf{u}) = -\frac{1}{\theta} \ln \left\{ \sum_r \sum_s \exp[-\theta(t_{ir} - u_{rs} + t_{sj})] \right\} \quad (57)$$

となる.

b) 上位問題

以上の議論から, 結局, 上位問題は以下の問題に帰着される.

$$\max_{\mathbf{u}} z_{PU}(\mathbf{u}) = \sum_r \sum_s u_{rs} n_{rs} + \sum_i \sum_j q_{ij} S_{ij}(\mathbf{u}) \quad (58)$$

この問題は, 目的関数が微分可能である制約なし凸計画問題であるから, Newton 法によって効率的に解くことができる. そのアルゴリズムは以下のようにまとめられる.

Step 0 : 初期設定

収束判定に使う定数 ϵ , 初期解 \mathbf{u}^0 を設定. $l = 0$ とする.

Step 1 : 移動方向決定

線形連立方程式 $\nabla^2 z_U(\mathbf{u}^l) \mathbf{d}^l = -\nabla z_U(\mathbf{u}^l)$ の解 \mathbf{d}^l を移動方向に設定.

Step 2 : ステップサイズ決定

\mathbf{d}^l に沿う直線探索によりステップサイズ α^l を決定.

Step 3 : 更新

$\mathbf{u}^{l+1} = \mathbf{u}^l + \alpha^l \mathbf{d}^l$ とする.

Step 4 : 収束判定

$z_U(\mathbf{u}^{l+1}) - z_U(\mathbf{u}^l) < \epsilon z_U(\mathbf{u}^l)$ ならば終了. そうでなければ $l := l + 1$ として Step 1 へ.

(2) Resource-directive Decomposition

a) 下位問題

解法は 6 章で示したものをを用いる. ここでは, 二重制約グラビティモデルによって近似的に解く方法を用いて解説する³. 具体的には, 目的関数にエントロピー

³ 北西隅法を用いる場合も, 下位問題の最適値関数の勾配を劣勾配に置き換えることで, 同じ手法を用いたアルゴリズムが

項を加え,

$$z_{RL}^i(\mathbf{x}^i) = \sum_s \sum_j t_{sj} x_{sj}^i + \frac{1}{\gamma} \sum_s \sum_j x_{sj}^i (\ln x_{sj}^i - 1) \quad (59)$$

とした問題を解く. この問題の解, および目的関数の最適値は, その最適性条件から

$$x_{sj}^{i*} = \exp[-\gamma(t_{sj} - \lambda_s^i - \mu_j^i)] \quad (60)$$

$$z_{RL}^{i*} = \sum_s \lambda_s^i \Delta_s^i + \sum_j \mu_j^i q_{ij} - \frac{\sum_j q_{ij}}{\gamma} \quad (61)$$

where

$$\lambda_s^i = \frac{1}{\gamma} \left(\ln \Delta_s^i - \ln \sum_j \exp[-\gamma(t_{sj} - \mu_j^i)] \right) \quad (62)$$

$$\mu_j^i = \frac{1}{\gamma} \left(\ln q_{ij} - \ln \sum_s \exp[-\gamma(t_{sj} - \lambda_s^i)] \right) \quad (63)$$

$$\Delta_s^i = \sum_r n_{rs}^i \quad (64)$$

である. (導出は付録を参照されたい.) ここで, λ^i, μ^i は, それぞれ制約条件 (50),(51) に対する Lagrange 乗数である. これらは, まず λ^i を適当な値に設定した上で, (63), (62) に λ^i, μ^i を繰り返し交互に代入することで得ることができる.

また, 上位問題の解の改訂に用いる, 最適値関数の \mathbf{n} に関する偏微分は, $\Delta_s^i = \sum_r n_{rs}^i$ より, 包絡線定理を用いて

$$\frac{\partial z_{RL}^{i*}}{\partial n_{rs}^i} = \lambda_s^i \quad \forall r, s, i \quad (65)$$

となる.

b) 上位問題

この問題は, 線形制約条件のみを持つ凸計画問題であるから, 射影勾配法によって解くことができる. そのアルゴリズムは以下の通りである.

Step 0-1 : 初期設定

収束判定に使う定数 ϵ , 最大繰り返し回数 L を設定. $l = 0$ とする.

Step 0-2 : 初期解設定

\mathbf{n} の初期値 $\mathbf{n}_0^{(i)}$ を (36),(37),(38) を満たすように定める.

Step 1-1 : 移動方向決定

下位問題の Lagrange 乗数を求める計算を行い, 最適値 z_{RL}^* を求める. また, \mathbf{d} を以下のように定める:

$$\mathbf{d} = -\nabla z_{RU}(\mathbf{n}^{(i)}) = \left[\dots, -\frac{\partial z_{RU}}{\partial n_{rs}^i}, \dots \right] \quad (66)$$

$$\frac{\partial z_{RU}}{\partial n_{rs}^i} = t_{ir} + \lambda_s^i \quad (67)$$

Step 1-2 : 移動方向の射影

\mathbf{d} を”等式制約および有効不等式制約の係数行列の核空間”に射影し, それをベクトル $\hat{\mathbf{d}}$ とする.

構築可能である. この問題に対する劣勾配法の詳細については kennington⁸⁾ を参照されたい.

Step 2 : 収束判定

上位問題の目的関数値 $z_{RU}(\mathbf{n}_l^{(i)})$ を計算する。 $z_{RU}(\mathbf{n}_l^{(i)}) - z_{RU}(\mathbf{n}_{l-1}^{(i)}) < \epsilon z_{RU}(\mathbf{n}_l^{(i)})$ ならば終了。そうでなければ Step 3 へ。

Step 3 : ステップサイズ決定

移動方向に沿う直線探索により、非負条件を侵害しないようにステップサイズを決定する。

Step 4 : 更新

$\mathbf{n}_{l+1}^{(i)} = \mathbf{n}_l^{(i)} + \hat{\mathbf{d}}$ とする。Step 1 へ戻る。

Step 1-2 における \mathbf{d} を射影する方法については付録を参照されたい。

8. おわりに

本稿ではまず、ドライバーと配送タスクを市場においてマッチングする CSD のモデルを設定し、そのモデルにおいて実現するマッチングが社会的に最適であることを示した。次に、オークション・メカニズムの設計やモデルの解の性質の分析に利用できる、効率的求解アルゴリズムを提案した。提案アルゴリズムの性能は、数値実験を通して確認し、発表時に示す予定である。本研究では、導入するオークション・メカニズムとして VCG メカニズムを想定し、その落札者決定問題 [Match](非集計整数計画問題) を、集約した実数変数問題 [SO] に緩和して解くアプローチをとった。本稿の残された課題は、示した [Match] の解法とそのアルゴリズムを用いた、詳細なオークション市場メカニズムの設計である。また、この解法は、問題が [Match] より簡単になる一方、得られる解は [Match] の近似解である。この近似によるマッチングの効率低下量の検証、またそれを小さく抑える方策の検討も今後の課題である。

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) および日本学術振興会・科学研究費補助金・挑戦的研究 (萌芽)(課題番号:18K18916) 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。

付録 I 移動方向ベクトルの射影方法

制約条件 (36),(37) は線形等式制約であるから、任意の制約式一つを省いて制約条件の冗長性を無くした上で、全ての行が一次独立な係数行列 \mathbf{M} と定数ベクトル \mathbf{p} を用いて

$$\mathbf{M}\mathbf{n}^{(i)} = \mathbf{p} \quad (\text{I.1})$$

と表現できる。また、非負条件 (38) は

$$-\mathbf{I}\mathbf{n}^{(i)} \leq \mathbf{0} \quad (\text{I.2})$$

と表現できる。ここで、 \mathbf{I} は単位行列である。

まず、アルゴリズム中のある暫定解 $\mathbf{n}_l^{(i)}$ における有効非負制約条件を特定し、それに対応する係数行列 \mathbf{I} の行のみを取り出して新たな行列 \mathbf{I}_l とし、

$$\mathbf{M}_l = \begin{bmatrix} \mathbf{M} \\ -\mathbf{I}_l \end{bmatrix} \quad (\text{I.3})$$

とする。この行列の核空間への射影行列は

$$\mathbf{P} = \mathbf{I} - \mathbf{M}_l^\top (\mathbf{M}_l \mathbf{M}_l^\top)^{-1} \mathbf{M}_l \quad (\text{I.4})$$

で与えられるから、射影後の移動方向ベクトル $\hat{\mathbf{d}}$ は、

$$\hat{\mathbf{d}} = \mathbf{P}\mathbf{d} \quad (\text{I.5})$$

となる。この結果、 $\hat{\mathbf{d}} \neq \mathbf{0}$ となる場合はこれを移動方向として用いれば良い。そうでない場合は、以下の操作を行う。

まず、ベクトル

$$\begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = -[\mathbf{M}_l \mathbf{M}_l^\top]^{-1} \mathbf{M}_l \mathbf{d} \quad \boldsymbol{\alpha} \in \mathbf{R}^M, \boldsymbol{\beta} \in \mathbf{R}^{N_l} \quad (\text{I.6})$$

を計算する。ここで、 M は等式制約の数、 N_l は暫定解 $\mathbf{n}_l^{(i)}$ における有効非負制約の数である。そして、 $\boldsymbol{\beta} \geq \mathbf{0}$ であれば、暫定解 $\mathbf{n}_l^{(i)}$ は最適性条件を満たしているから、終了する。そうでなければ、 $\boldsymbol{\beta}$ の負の成分に対応する \mathbf{M}_l の行を除いて新たに \mathbf{M}_l を定義し、それを用いて射影行列 \mathbf{P} を計算する。そして、改めて $\hat{\mathbf{d}} = \mathbf{P}\mathbf{d}$ を計算し、これを移動方向として用いる。

付録 II Resource-directive decomposition 下位問題の解法

下位問題は以下の通り。

$$\min_{\mathbf{x}^i} \sum_s \sum_j t_{sj} x_{sj}^i + \frac{1}{\gamma} \sum_s \sum_j x_{sj}^i (\ln x_{sj}^i - 1) \quad (\text{II.1})$$

$$\text{s.t.} \quad \sum_j x_{sj}^i = \Delta_s^i \quad \forall s \quad (\text{II.2})$$

$$\sum_s x_{sj} = q_{ij} \quad \forall j \quad (\text{II.3})$$

$$x_{sj} \geq 0 \quad \forall s, j \quad (\text{II.4})$$

この最適性条件は、以下のようになる。

$$\begin{cases} t_{sj} + \frac{1}{\gamma} \ln x_{sj}^{i*} - \lambda_s^i - \mu_j^i = 0 & \text{if } x_{sj}^i > 0 \\ t_{sj} + \frac{1}{\gamma} \ln x_{sj}^{i*} - \lambda_s^i - \mu_j^i \geq 0 & \text{if } x_{sj}^i = 0 \end{cases} \quad \forall i, s, j \quad (\text{II.5})$$

$$\sum_j x_{sj}^{i*} = \Delta_s^i \quad \forall s \quad (\text{II.6})$$

$$\sum_s x_{sj}^{i*} = q_{ij} \quad \forall j \quad (\text{II.7})$$

ここで、二重制約重力モデルにおいては常に $\mathbf{x}^i > \mathbf{0}$ が成立することから、(II.5) より

$$t_{sj} + \frac{1}{\gamma} \ln x_{sj}^{i*} - \lambda_s^i - \mu_j^i = 0 \quad (\text{II.8})$$

$$x_{sj}^{i*} = \exp[-\gamma(t_{sj} - \lambda_s^i - \mu_j^i)] \quad (\text{II.9})$$

となり、(60) を得る。これを最適性条件 (II.6),(II.7) および目的関数 (II.1) に代入することで、それぞれ (62),(63),(61) を得る。

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, 2018.
- 3) Allahviranloo, M. and Baghestani, A.: A dynamic crowdshipping model and daily travel behavior, *Transportation Research Part E: Logistics and Transportation Review*, Vol.128, pp.175–190, 2019.
- 4) Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders, *The Journal of finance*, Vol.16, No.1, pp.8–37, 1961.
- 5) Clarke, E. H.: Multipart pricing of public goods, *Public choice*, Vol.11, No.1, pp.17–33, 1971.
- 6) Demange, G., Gale, D., and Sotomayor, M.: Multi-item auctions, *Journal of Political Economy*, Vol.94, No.4, pp.863–872, 1986.
- 7) Gonzaga, C. C.: An algorithm for solving linear programming problems in $O(n^3)$ operations, *Progress in mathematical programming*, pp. 1–28, Springer, 1989.
- 8) Kennington, J. and Shalaby, M.: An effective subgradient procedure for minimal cost multicommodity flow problems, *Management Science*, Vol.23, No.9, pp.994–1004, 1977.

(2019. 10. 4 受付)

AN EFFICIENT MATCHING SYSTEM FOR CROWDSOURCED DELIVERY

Taiki WATANABE and Takashi AKAMATSU