

中継可能な協調型輸送モデルの提案

藤井 貴彬[†] 瀧瀬 和樹^{††} 浅野 泰仁^{††} 吉川 正俊^{††}

[†] 京都大学工学部情報学科 〒606-8501 京都府京都市左京区吉田本町 36-1

^{††} 京都大学情報学研究科 〒606-8501 京都府京都市左京区吉田本町 36-1

E-mail: [†]{tfujii,takise}@db.soc.i.kyoto-u.ac.jp, ^{††}{asano,yoshikawa}@i.kyoto-u.ac.jp

あらまし インターネットの普及によりネットショッピングが需要が増え、宅配荷物が近年増加している。しかし、荷物の増加に対してそれらを輸送する宅配業者の従業員の増員が十分に行われておらず、一人当たりの負担も増加している。この問題を解決するため、宅配業者以外も輸送に参加する協調型輸送を実施しているが、一部の人しか参加することができないため、宅配業者の負担を軽減しきれない。本研究では、さらに多くの人が参加できるような輸送システムとして、一つ荷物を複数の輸送者によって分担し輸送する、“中継可能な協調型輸送モデル”を提案する。この輸送モデルを設定し、厳密解を求める手法を考案・実験をすることにより、このモデルにおける性質や時間計算量の問題を特定し、より効率的な手法の検討を行っている。この研究により現在問題となっている宅配業者の過剰な負担を多くの人に分散することができ、宅配業者の負担を軽減することが期待される。

キーワード グラフ, 配送, 協調型輸送, 厳密解, 最適化, 最適経路, シェアリングエコノミー

1. はじめに

現在、スマートフォンやPCの普及によりインターネットがより身近な存在となっており、Amazon.com^(注1)などのネットショッピングの需要が大きくなっている。それに伴い、ネットショッピングで注文した商品を届けるための輸送形態も変化し、従来は注文してから数日後に届くものであったが、近年 Amazon では当日輸送といったサービスも実施し、ますます便利なものとなっている。

しかし、このようなサービスが充実すればするほど宅配物も増加し、宅配業者の負担も増加することが問題となっている。実際に、ヤマト運送は当日輸送サービスによる負担が大きすぎると発表している^(注2)。

宅配業者の負担増加は解決すべき問題として既に認識されており、負担を軽減しようと様々な試みが行われている。その方法として、宅配業者のみによって輸送を行うのではなく、様々な人が参入することで輸送を行う協調型輸送が挙げられる。例えば、Amazon はが既存の配達業者ではなく一般の人が荷物を配達する新しいオンデマンドのサービス、Amazon Flex^(注3)を開始した。また、様々な配送・輸送サービスを提供している Uber が Uber Freight^(注4)という協調型輸送サービスを開始した。Uber Freight は有資格の運送業者や個人トラック運転手が輸送する貨物を手軽に探して受注できるようにしたオンデマンドのサービスである。

このような、個人が参入してより多くの人が携わる協調型輸送によって個々の負担を軽減しようとする試みが行われている。しかし、これは宅配に長時間従事できる一部の人しか参加する

ことができず、急激に増えた宅配業者の負担を十分に軽減できるようなものではない。

そこで本研究ではさらに多くの人が参加できるような配送システムとして、日常生活のついでのようなわずかな時間でも輸送に参加できるモデルを提案したいと考えた。そして、わずかな時間でも参加可能にするためには、荷物の輸送に中継を可能にしたモデルが必要であると考えた。具体的には、複数の輸送者が単一の荷物を運ぶことも考慮し、ある輸送者が荷物を途中まで輸送しそこから別の輸送者が荷物を引き継いで輸送するという、中継を挟んで人と人とが協調して輸送するシステムである。(以降、中継可能な協調型輸送と呼ぶ)

例えば、ある人が通学するとき、自分の家の近くの施設 A (例

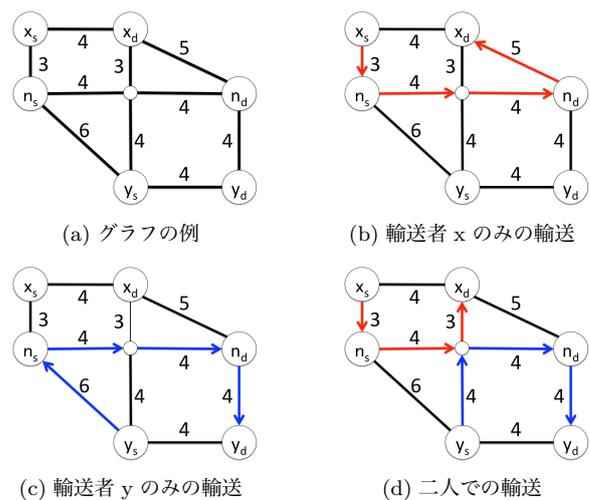


図1 協調型輸送が有用となる例

(注1) : <https://www.amazon.co.jp>

(注2) : <https://mainichi.jp/articles/20170407/k00/00e/020/198000c>

(注3) : <https://flex.amazon.com>

(注4) : <https://freight.uber.com>

えばコンビニエンスストア)から学校の近くにある施設 B に荷物を持って寄り道をして運ぶ。次に、施設 B の近くに住んでいる別の人が通勤のついでに荷物を施設 B で受け取り勤務近くの

施設 C に寄り道して届ける。このように中継可能な協調型輸送であれば輸送する個人にあまり負担をかけずに荷物を最終目的地まで運ぶことが可能である。

図 1 の (a) から (d) は中継可能な協調型輸送が有用である具体的なグラフの例である。(a) のようなグラフについて考え、各円はグラフの節点を表している。節点にある記号は、二人の輸送者 x と y と一つの荷物 n の出発地点を表す x_s, y_s, n_s と目的地を表す x_d, y_d, n_d である。また各辺の値は節点間の距離を示している。

このグラフにおいて、輸送者 x と y は両者共にただ目的地に到達するだけならば距離 4 だけで移動できる。次に (b) のように輸送者 x が一人で荷物を輸送するならば合計距離が 16 となり、12 増加することとなる。一方 (c) のように輸送者 y が一人で荷物を輸送するならば合計距離が 18 となり、14 増加することとなる。しかし、(d) のように二人で輸送した場合ならば、輸送者 x の合計距離は 10、輸送者 y の合計距離は 12 となり、それぞれの増加した距離は 6 と 8 になる。このように、中継可能な協調型輸送により二人で輸送すれば、一人一人の迂回する負担は一人で輸送するより小さくすることができる。

本研究では、中継可能な協調型輸送のモデルの作成を行い、そのモデル内における厳密解のアルゴリズムの考案・実験を行うことでこのモデルにおける性質・時間計算量の問題を特定するという貢献をした。具体的なモデルの性質としては、荷物の輸送経路が決定した後に荷物がどの辺をどの順序で輸送されるか、及び輸送者が複数の荷物を輸送する場合にどの辺でどの荷物をどの順番で輸送するか、これら二つの順序が重要となることがわかった。

2. 関連研究

複数人が合流するときにそれぞれ最短距離の合計が最小となる地点を求める研究に、Yan らの研究 [1] がある。更にこの研究を発展させたものとして、複数人が好きな経路を持ち、それぞれの経路からの迂回した最短距離の合計が最小となる地点を求める研究に、Ahmadi らの研究 [2] がある。

本研究でも、一つの荷物を複数の輸送者によって輸送される場合には荷物を引き継ぐ中継地点が存在するため、輸送者と輸送者がどこで合流するべきかを決定しなければならない。しかし、[1] [2] の研究と異なり、合流地点と輸送者の位置だけではなく荷物の出発地点及び目的地も考慮しなければならない。また、本研究では輸送者及び荷物に出発時刻と到着時刻が存在する。そのため、輸送者の移動距離の合計を最小にするだけでは出発時刻及び到着時刻を満たさない場合が存在する。よって、それら時間制約も考慮して移動距離の合計を最小にする必要がある。

次に、特定の施設にある輸送車が顧客の荷物を輸送・回収する問題に配送計画問題がある。配送計画問題には条件設定によって様々な問題に分類される。特定の施設が一つではなく複数の場合に拡張した研究に Yu らの研究 [3] があり、そこに輸送車に複数の種類（積載量の違いなど）も拡張した研究に Salhi らの研究 [4] がある。また、顧客の荷物の回収に時間制約を与えた研究が Abel らの研究と [5] や Vidal らの研究 [6] であり、特に [6] で

は複数の施設かつ荷物の回収に時間制約を与えた場合でも対応できることが言及されている。

本研究は配送計画問題を大幅に拡張したのと考えられる。配送計画問題では特定の施設に全ての荷物を回収する問題であるため輸送車と荷物の目的地は特定の施設に絞られる。しかし、本研究では輸送者と荷物の目的地が全ての地点でも可能であるという点で異なっている。また、時間制約については荷物だけではなく輸送する側にも存在している点でも異なっている。

また、合流を考慮した経路最適化問題の研究に Zhang の研究 [7] や Takise らの研究 [8] がある。合流を考慮した経路最適化問題とは、複数のユーザが各々の出発地点から目的地へ向かうとき経路の途中で合流して移動することに利益があると考え、その合流の利益も考慮した最短経路を決定する問題である。さらに、合流を考慮した問題を派生させたライドシェアリングの研究として Cao らの研究 [9] や Drews らの研究 [10] がある。合流を考慮した問題と異なる点は、合流を考慮した問題はどちらも移動可能であることに対し、ライドシェアリングでは輸送を行う一方のみが移動可能であるという点である。

本研究も合流を考慮した経路最適化問題やライドシェアリングの問題の一種と捉えることができる。例えば、本研究における荷物を客、輸送者をタクシーに変換すればライドシェアリングのように扱うことができる。しかし、一般的なライドシェアリングでは一台のタクシーが客を目的地まで運びきるのに対し、本研究では複数の輸送者が中継して輸送することも可能としている点で異なっており、[7] [8] [9] [10] よりも計算量が大きくなると予想される。

3. 中継可能な協調型輸送モデル

3.1 問題設定

本論文では、中継可能な協調型輸送を考慮した輸送システムに道路情報、輸送者、荷物の情報が与えられたときに、システムにとって最大の利益となるような荷物の輸送経路及び輸送者の輸送経路を決定することを目標としている。次節にて、入力となる道路情報、輸送者、荷物の情報及びシステム側の利益を決定する関数について定義を行う。ただし、輸送者及び荷物のデータはある期間に区切って前以て用意されているとし、また荷物は輸送することでシステム側に不利益を与えるものは存在せず、全て輸送するものとする。

3.2 問題の定式化のための定義・制約

本研究では中継可能な協調型輸送のモデルの提案、及びそのモデルの問題について扱う。以下、このモデル及び問題を定義するための用語・記号を定義する。

[定義 3.1](グラフ) 道路情報として無向グラフ $G(V, E)$ を用いる。 V は荷物の積み込み・積み下ろしを行う節点 v の集合であり、 E は道路の経路を表す辺 e の集合である。各辺 e には重み $w(e)$ を持ち、辺の始点を $v_s(e) \in V$ 、終点を $v_d(e) \in V$ で表す。2 節点 v と v' を最短距離で移動したときにかかる時間を $W(v, v') > 0$ で表す。

[定義 3.2](荷物) 各荷物は b_i で表し、全ての荷物の集合を B と表す。各荷物 b の出発地点は $s(b_i) \in V$ 、目的地は $d(b_i) \in V$ 、

荷物が移動可能となる時刻は $t_s(b_i)$, 目的地に到着しなければならない上限時刻は $t_d(b_i)$ で表す。

[定義 3.3](輸送者) 各輸送者は u_j で表し, 全ての輸送者の集合を U と表す. 各輸送者 u_j の出発地点は $s(u_j) \in V$, 目的地は $d(u_j) \in V$, 輸送者が移動可能となる時刻は $t_s(u_j)$, 目的地に到着しなければならない上限時刻は $t_d(u_j)$, 積載荷物数は $m(u_j) \geq 1$ で表す.

[定義 3.4](輸送経路) 荷物が出発地点から目的地までに通る経路を **荷物の輸送経路** と呼び, 輸送者が目的地から到着地点までに通る経路を **輸送者の輸送経路** と呼ぶ.

[定義 3.5](時刻制約) 各荷物 b_i の出発地点から移動を開始する時刻を $T_s(b_i)$, 目的地に到達した時刻を $T_d(b_i)$ と表し, 各輸送者 u_j の出発地点から移動を開始する時刻を $T_s(u_j)$, 目的地に到達した時刻を $T_d(u_j)$ と表す. 出発地点から移動を開始する時刻 $T_s(b_i)$ は荷物が移動可能となる時刻 $t_s(b_i)$ よりも後になるため,

$$T_s(b_i) \geq t_s(b_i) \quad (1)$$

となり, 目的地に到達した時刻 $T_d(b_i)$ は目的地に到着しなければならない上限時刻 $t_d(b_i)$ より前になるため,

$$T_d(b_i) \leq t_d(b_i) \quad (2)$$

となる. 輸送者についても同様に, 出発地点から移動を開始する時刻 $T_s(u_j)$ は輸送者が移動可能となる時刻 $t_s(u_j)$ より後になるため,

$$T_s(u_j) \geq t_s(u_j) \quad (3)$$

となり, 目的地に到達した時刻 $T_d(u_j)$ は目的地に到着しなければならない上限時刻 $t_d(u_j)$ より前になるため,

$$T_d(u_j) \leq t_d(u_j) \quad (4)$$

となる.

[定義 3.6](利益関数) システム側が得る利益を導く関数を定義するために, まずシステム側の収入関数と支出関数を定義する.

収入関数 Rwd_{in} は, 荷物輸送の依頼者は各荷物ごとの大きさと最短経路における輸送経路に依存した金額と設定するため, 荷物の輸送の最短距離を x , 荷物の大きさを y , 定数を α とすると, $Rwd_{in}(b) = \alpha \cdot x \cdot y (b \in B)$ となる.

支出関数 Rwd_{out} は, 荷物の輸送距離と輸送するために要した迂回距離の合計に比例した金額とするため, 荷物の輸送距離を x' , 輸送するのに要した迂回距離を z , 定数を β_1, β_2 とすると, $Rwd_{out}(u) = \beta_1 \cdot x' + \beta_2 \cdot z (u \in U, \beta_1 > \beta_2)$ となる.

ここで, $(\beta_1 > \beta_2)$ と定義しているのは, 荷物を持って移動する方が荷物持たずに移動するより輸送として貢献が高いと考えたため, 報酬の値を大きくとるようにしている.

以上よりシステム側の利益関数 Rwd は以下のように定義できる.

$$Rwd(b, u) = \sum_{b \in B} Rwd_{in}(b) - \sum_{u \in U} Rwd_{out}(u) \quad (5)$$

3.3 問題

前節より, 本論文における問題は以下のように定式化される.

入力 道路情報 $G(V, E)$, 各荷物データ $b \in B$, 各輸送者データ $u \in U$

出力 各荷物と各輸送者の時刻制約 (1)~(4) を満たすような輸送経路

目的関数 (最大化)

$$\operatorname{argmax}(Rwd(b, u)) \quad (6)$$

問題を単純にするため, 本論文では以下 2 点を条件設定として加える.

I 荷物の輸送経路は出発地点と目的地を結ぶ最短経路 (複数ある場合は予め定めた最短経路) に限定する

II 各輸送者 u が一度に輸送できる積載荷物数 $m(u)$ は $m(u) = 1$ に限定する

4. 提案手法

4.1 利益関数の単純化

まず問題設定で与えられた荷物は全て輸送するため, 次の命題が成り立つ.

[命題 4.1] 荷物と輸送者の輸送経路が決定すれば, 各輸送経路の辺における輸送時刻割当てによって目的関数の値は変化しない.

【証明】

これは, 与えられた荷物は全て輸送するため収入関数の値が固定される. したがって, 目的関数が以下のように表せる.

$$\operatorname{argmax}(Rwd(b, u)) = \min\left(\sum_{u \in U} Rwd_{out}(u)\right) \quad (7)$$

支出関数 Rwd_{out} は輸送者が通る輸送経路のみ依存するため, 荷物と輸送者の輸送経路が決定すれば, 目的関数の値は輸送時刻割当てに依らず決定される. ■

4.2 輸送経路の決定

命題 4.1 より, 荷物と輸送者の輸送経路が決定すれば目的関数の値を一意に決まることがわかった. したがって, 厳密解を求めるためには全ての荷物と輸送者の輸送経路を列挙することが重要となる. しかし, 前節の条件設定より荷物の輸送経路は出発地点と目的地を結ぶ最短経路に限定するため, 輸送者の輸送経路のみ決定すればよい. 輸送者の輸送経路を決定するには輸送者がどの荷物のどの部分の輸送を担当するかを先に決める必要がある. そのため, 次の輸送組み合わせと輸送順序を定義する.

[定義 4.2](荷物の輸送順序) 各荷物を通る輸送経路の辺の順序を **荷物の輸送順序** と呼ぶ. 荷物 b_i の輸送順序を $seq(b_i) = (e_{i1}, e_{i2}, \dots)$ とし, 輸送順序の k 番目の辺を e_{ik} とする. 更に, $seq(b_i)$ の大きさ, つまり b_i の輸送経路の長さを $L(b_i)$ とする.

荷物の輸送順序の例として, 図 2 の (a)(b) を挙げる. ただし, 実線は節点間の辺を表し, 点線は節点間の最短経路を表している. まず b_1 について考えると, 出発地点から e_{11}, e_{12} の順に辺を

通って目的地点へ向かうため、 $e_{11}^b = e_1, e_{12}^b = e_2$ となる。次に b_2 について考えると、出発地点から e_3, e_4 の順に辺を通って目的地点へ向かうため、 $e_{21}^b = e_3, e_{22}^b = e_4$ となる。

[定義 4.3](輸送組合せ) どの輸送者がどの荷物の輸送経路のどの辺を輸送するかの組合せを**輸送組合せ**と呼び、輸送者 u_j が荷物 b_i の輸送順序で k 番目の辺 e_{ik}^b を輸送する組み合わせを $c(i, k, j)$ と表す。また、輸送組合せの集合を C と表す。

輸送組合せを表す例として図 2 の (a) を挙げる。荷物の輸送経路のうち、赤色の辺は u_1 が輸送する辺で、青色の辺は u_2 が輸送する辺とする。このとき、 u_1 の担当である赤色の辺が e_1 と e_4 で、これらは荷物の輸送順序を考慮すると e_{11}^b と e_{22}^b となるため、輸送組合せが $c(1, 1, 1), c(2, 2, 1)$ となる。同様に u_2 の担当である青色の辺が e_2 と e_3 で、これらは荷物の輸送順序を考慮すると e_{12}^b と e_{21}^b となるため、輸送組合せが $c(1, 2, 2), c(2, 1, 2)$ となる。

[定義 4.4](輸送者の輸送順序) 輸送組合せが決定されたなら

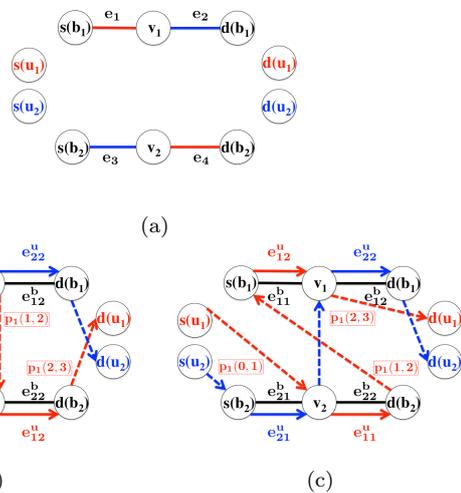


図 2 輸送順序を表す例

ば、輸送者ごとに担当する各辺をどのような順序で輸送するか決定する必要がある。この順序を**輸送者の輸送順序**と呼ぶ。輸送者 u_j の輸送順序を $seq(u_j) = (e_{j1}, e_{j2}, \dots)$ とし、輸送順序の l 番目の辺を e_{jl}^u とする。更に、 $seq(u_j)$ の大きさ、つまり u_j の輸送する辺の数を $L(u_j)$ とする。

輸送順序を表す例として図 2 の (b) を挙げる。 u_1 は 1 番目に e_1 で輸送を行い、2 番目に e_4 で輸送を行うため、 $e_{11}^u = e_1, e_{12}^u = e_4$ となる。一方、 u_2 が 1 番目に e_3 で輸送を行い、2 番目に e_2 で輸送を行うため、 $e_{21}^u = e_3, e_{22}^u = e_2$ となる。

ここで、与えられた一つの輸送組合せに対して各輸送者の輸送順序が複数ありうることに注意されたい。その例として図 2 の (b) と (c) に挙げる。(b) と (c) の輸送組合せは C は共に $\{c(1, 1, 1), c(1, 2, 2), c(2, 1, 2), c(2, 2, 1)\}$ である。しかし、輸送者 u_1 に注目すると、(b) では $e_1 \rightarrow e_4$ の順に輸送していることに対し、(c) では $e_4 \rightarrow e_1$ の順に輸送している。そのため、 e_{11}^u が (b) では e_1 であるが、(c) では e_4 となり、輸送順序が異なる。また、ある荷物 b_i における荷物の輸送順序について、 k 番目

の終点と $k + 1$ 番目を始点は同一の節点となる。すなわち、 $v_d(e_{ik}^b) = v_s(e_{i(k+1)}^b)$ となる。

一方で、ある輸送者 u_j における輸送者の輸送順序について、 l 番目に輸送する荷物と $l + 1$ 番目に輸送する荷物が異なる場合、 l 番目の辺の終点と $l + 1$ 番目の辺の始点が異なる可能性がある。そのため、異なる場合の節点間の最短経路を $p_i(l, l + 1)$ と表す。以上より、各輸送者の輸送経路を決定するためには全ての輸送組合せを列挙し、更に各輸送者の輸送順序を全て列挙する必要がある。

まず、全ての輸送組合せを列挙することを考える。これは荷物の輸送経路の全ての辺を輸送者全員で分ける場合の数を考えれば良い。例として荷物の輸送経路が e_1 と e_2 の二つあり、輸送者が u_1 と u_2 の二人いる場合を考える。まず e_1 で誰が輸送するか考え、輸送者が二人いるので u_1 が輸送する場合と u_2 が輸送する場合の二通りがある。次に e_2 で誰が輸送するか考える。これは、 e_1 を u_1 が担当した場合と u_2 が担当した場合それぞれにおいて e_2 をどちらが担当するかの二通り存在する。よって、この場合は計四通りの輸送組合せが存在する。したがって、荷物の輸送経路の辺の数を $|e|$ 、輸送者の数を $|U|$ とすると、輸送組合せは合計 $|U|^{|e|}$ 通り存在する。

次に、全ての輸送順序を列挙することを考える。これは与えられた輸送組合せにおいて、各輸送者ごとに担当する辺の順序を考えれば良い。例えば、輸送者 u_1 が担当する辺が e_1, e_2, e_3 の三つあった場合、 u_1 の輸送順序は $(e_1 \rightarrow e_2 \rightarrow e_3), (e_1 \rightarrow e_3 \rightarrow e_2), (e_2 \rightarrow e_1 \rightarrow e_3), (e_2 \rightarrow e_3 \rightarrow e_1), (e_3 \rightarrow e_1 \rightarrow e_2), (e_3 \rightarrow e_2 \rightarrow e_1)$ の合計 $3! = 6$ 通りが存在する。この場合分けが各輸送者ごとに発生するので、 u_j の担当する辺の数を $|e_{u_j}|$ とすると、合計 $|e_{u_1}|! * |e_{u_2}|! * \dots * |e_{u_{|U|}}|!$ 通り存在する。

上記のように各輸送者の輸送順序を全て列挙することで、荷物と輸送者の輸送経路を全て列挙できる。しかし、その中には輸送不可能である場合が存在する。その判定のため、荷物と輸送者の輸送順序をより詳細に表す辺順序割当てと輸送前後関係を定義する。

[定義 4.5](辺順序割当て) ある荷物 b_i をある輸送者 u_j がある辺で輸送し、その辺が b_i の荷物の輸送順序においては k 番目かつ u_j の輸送者の輸送順序においては l 番目であるとする。この割当てを**辺順序割当て** $a(i, k, j, l)$ と呼ぶ。全ての辺順序割当て a の集合を A と表す。

ここで、辺順序割当て $a(i, k, j, l)$ に対応する辺を e_a と表すとすると、 $e_a = e_{ik}^b = e_{jl}^u$ となる。

また、ある辺 e_a において荷物が輸送される場合、輸送が開始される時刻を $t_s(e_a)$ 、輸送が終了する時刻を $t_d(e_a)$ で表し、

$$t_s(e_a) < t_d(e_a) \quad (8)$$

となる。

辺順序割当ての例として図 2 の (b) を用いて考える。辺 e_1 は荷物の輸送順序だと e_{11}^b となり、輸送者の輸送順序だと e_{11}^u となるため、辺順序割当てとして表すと $a_1(1, 1, 1, 1)$ となる。他の辺についても同様に考え、 e_2 は $a_2(1, 2, 2, 2), e_3$ は $a_3(2, 1, 2, 1), e_4$ は $a_4(2, 2, 1, 2)$ となる。

[定義 4.6] (輸送前後関係) 2 組の辺順序割当て $a(i, k, j, l), a'(i', k', j', l')$ のそれぞれに対応する $e_a, e_{a'}$ 2 辺において,

(A) e_a と $e_{a'}$ で輸送されている荷物が同じで、その荷物の輸送順序において e_a の次に $e_{a'}$ で輸送される場合、すなわち $i = i'$ かつ $k + 1 = k'$ を満たすか、

(B) e_a と $e_{a'}$ で輸送している輸送者が同じで、その輸送者の輸送順序において e_a の次に $e_{a'}$ で輸送する場合、すなわち $j = j'$ かつ $l + 1 = l'$ を満たすとき、 e_a と $e_{a'}$ は **輸送前後関係** を持つと言い、 $r(a, a')$ と表す。また、全ての輸送前後関係 r の集合を R と表す。

輸送前後関係の例として図 2 の (b) を用いる。この図での e_{a_1} と e_{a_2} に輸送前後関係があるかどうか考える。両辺はともに b_1 を輸送する辺であり、 e_1 の次に e_2 で輸送をする。これは条件 (A) を満たしているため、この二つの辺は輸送前後関係 $r_1(a_1, a_2)$ となる。同様に e_{a_3} と e_{a_4} もこの条件 (A) を満たしているため、輸送前後関係 $r_2(a_3, a_4)$ である。次に、 e_{a_1} と e_{a_4} について考える。両辺はともに u_1 が輸送する辺であり、 e_1 の次に e_2 で輸送する。これは条件 (B) を満たしているため、この二つの辺は輸送前後関係 $r_3(a_1, a_4)$ となる。同様に e_{a_2} と e_{a_3} もこの条件 (B) を満たしているため、輸送前後関係 $r_4(a_2, a_3)$ である。つまり、このグラフには輸送前後関係が以上の四つ存在する。

全ての辺順序割当ての輸送前後関係を表すために次のグラフを定義する。

[定義 4.7] (輸送順序関係グラフ) 与えられた全ての辺順序割当ての集合を A 、それに対する輸送前後関係を R とするとき、全ての辺の前後関係を表すグラフとして有向グラフ $H(A, R)$ を用い、**輸送順序関係グラフ** と呼ぶ。有向辺の向きは輸送前後関係 $r(a, a')$ において $a \rightarrow a'$ の向きとする。

輸送順序関係グラフの例として図 3 を挙げる。図 3 は図 2 の (b) に対応した輸送順序関係グラフである。ただし、 e_k に対応した辺順序割当てを a_k とし、 $e_k = e_{a_k}$ とする。各ノードはそれぞれの辺順序割当てに対応するため、 a_1 から a_4 の四つがある。輸送前後関係は $r_1(a_1, a_2), r_2(a_3, a_4), r_3(a_1, a_4), r_4(a_3, a_2)$ である。そのため、 $a_1 \rightarrow a_2, a_3 \rightarrow a_4, a_1 \rightarrow a_4, a_3 \rightarrow a_2$ の四つ有向辺が存在する。輸送順序関係グラフを用い、与えられた辺順

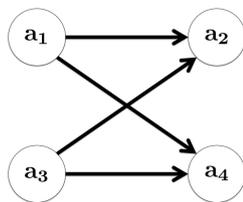


図 3 輸送順序関係グラフの例

序割当て集合が輸送可能であるか調べるために、整合した輸送時刻割当てを定義する。

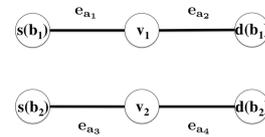
[定義 4.8] (整合した輸送時刻割当て) 荷物と輸送者の輸送経路が決定され、各輸送経路の辺 e における輸送開始時刻及び輸

送終了時刻も決定したとする。このとき、全ての輸送前後関係 $r(a, a')$ において輸送時刻に矛盾が生じない、すなわち

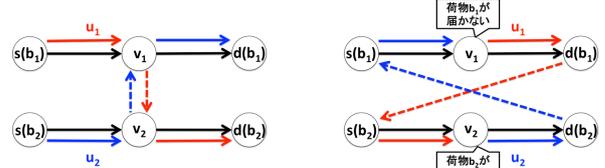
$$t_d(e_a) \leq t_s(e_{a'}) \quad (9)$$

を満たす輸送時刻を **整合した輸送時刻割当て (Consistent Time Assignment)** と呼び、CTA と表す。このとき、 $T_s(b_i)$ は最初の輸送順序である辺で輸送を開始した時刻と等しいため、 $T_s(b_i) = t_s(e_{i1}^b)$ となり、 $T_d(b_i)$ は最後の輸送順序である辺で輸送を終了した時刻と等しいため、 $T_d(b_i) = t_d(e_{ik}^b)$ ($k = L(b_i)$) となる。一方、 $T_s(u_j)$ は輸送順序の最初の辺で輸送を開始した時刻に、出発地点からその辺へ行くまでにかかる時間を引いた時刻と等しいため、 $T_s(u_j) = t_s(e_{j1}^u) - W(s(u_j), v_s(e_{j1}^u))$ となり、 $T_d(u_j)$ は輸送順序の最後の辺で輸送を終了した時刻に、そこから目的地点へ行くまでにかかる時間を加えた時刻と等しいため、 $T_d(u_j) = t_d(e_{jl}^u) + W(v_d(e_{jl}^u), d(u_j))$ ($l = L(u_j)$) となる。

ここで、CTA が存在しない A があることに注意されたい。そ



(a) グラフの例



(b) 輸送可能な例

(c) 輸送不可能な例

図 4 矛盾した輸送順序の例

のような例を図 4 の (a) から (c) で例に挙げる。(a) は荷物 b_1 と b_2 の出発地点と目的地点及び輸送経路が表している。(b) と (c) の赤矢印と青矢印は輸送者 u_1, u_2 の輸送経路を表している。ただし、実線は節点間の辺を表し、点線は節点間の最短経路を表している。

(b) の例で荷物の輸送が可能かどうかを考える。例えば、 u_1 は自身にとって輸送順序の 1 番目の輸送、つまり $s(b_1)$ から v_1 へ b_1 の輸送を行い、 u_2 も自身にとって輸送順序の 1 番目の輸送、つまり $s(b_2)$ から v_2 へ b_2 の輸送を行うことができる。次に両者にとって輸送順序の 2 番目の輸送を行うため、 u_1 は v_2 へ、 u_2 は v_1 へ向かう。そうすれば最後に u_1 が b_2 を v_2 から $d(b_2)$ へ輸送し、 u_2 が b_1 を v_1 から $d(b_1)$ へ輸送することができるので (b) の例は輸送完了となる。

次に (c) の例をみると、(b) の例のようにはいかないことがわかる。まず、 u_1 について考える。 u_1 は最初に e_{a_2} で輸送を行うため、(8) より $t_s(e_{a_2}) < t_d(e_{a_2})$ である。次に、輸送前後関係 $r(e_{a_2}, e_{a_3})$ が存在するため、(9) より $t_d(e_{a_2}) \leq t_s(e_{a_3})$ である。更に、 e_{a_3} で輸送を行うため、(8) より $t_s(e_{a_3}) < t_d(e_{a_3})$

である。よって、 $t_s(e_{a_2}) < t_d(e_{a_2}) \leq t_s(e_{a_3}) < t_d(e_{a_3})$ が成り立つ。同様に、 u_2 について考えると、 $t_s(e_{a_4}) < t_d(e_{a_4}) \leq t_s(e_{a_1}) < t_d(e_{a_1})$ が成り立つ。ここで、 b_1 と b_2 に関する輸送前後関係を考えると $r'(e_{a_1}, e_{a_2})$ と $r''(e_{a_3}, e_{a_4})$ が存在するため、 $t_d(e_{a_1}) \leq t_s(e_{a_2})$ と $t_d(e_{a_3}) \leq t_s(e_{a_4})$ の二つが成り立つ。以上より、 $t_d(e_{a_1}) < t_s(e_{a_1})$ が成り立つことになるが、このような時刻は存在しない。したがって、(c) では輸送することが不可能である。

CTA が存在しない A を **矛盾した辺順序割当て** と呼ぶ。これはいわゆる計算機科学におけるデッドロックの状態に対応するものである。我々は DAG を用いることによって矛盾した辺順序割当てを検知できることを発見した。DAG とは有向非巡回グラフ (Directed acyclic graph) である。この DAG を用いることで次の補題が成り立つ。

[補題 4.9] 与えられた辺順序割当て集合 A とそれに対応した輸送前後関係 R により定まった輸送順序関係グラフを $H(A, R)$ とする。 H が DAG でないならば、 A は矛盾した辺順序割当てである。

証明を行う前にトポロジカルソートについて説明する。トポロジカルソートとは有向グラフの各有向エッジの順序を守るようにノードを並べる操作のことである。DAG ならばトポロジカルソートが可能であり、Kahn のアルゴリズム [11] や Tarjan のアルゴリズム [12] によって並べることができる。

【証明】

DAG でないならば矛盾した辺順序割当てであることを背理法によって示す。つまり、CTA が存在すれば、DAG ではない輸送順序関係グラフが存在すると仮定し矛盾点を示す。まず、DAG でないならば、少なくとも一つ閉路が存在する。その中のうち一つの閉路、 $a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_k \rightarrow a_1$ について考える。CTA が存在すれば (8) と (9) が成り立つため、

$$\begin{aligned} t_s(e_{a_1}) < t_d(e_{a_1}) \leq t_s(e_{a_2}) < t_d(e_{a_2}) \leq \dots \\ &\leq t_s(e_{a_n}) < t_d(e_{a_n}) \leq t_s(e_{a_1}) \\ \implies t_s(e_{a_1}) < t_s(e_{a_1}) \end{aligned} \quad (10)$$

となり、矛盾する。■

補題 4.9 より、DAG だけが矛盾した辺順序割当てではない可能性がある。そのため、列挙した辺順序割当てのうち DAG であるもののみ次節の時刻割当てについて考える。

4.3 実行可能な輸送時刻割当て

補題 4.9 により CTA が存在する、矛盾のない辺順序割当てを検知することができる。しかし、CTA は荷物と輸送者の移動が可能となる時刻割及び到着しなければならない時刻を考慮したものではない。そのため、次にそれらの時刻を考慮した実行可能な輸送時刻割当てを定義する。[定義 4.10] (**実行可能な輸送時刻割当て**) CTA のうち、輸送者及び荷物の出発時刻と到着時刻の時間制約を満たしたものを **実行可能な輸送時刻割当て (Feasible Time Assignment)** と呼び、FTA と表す。つまり、ある CTA が決定した全ての荷物 $b_i \in B$ と全ての輸送者 $u_j \in U$ において、(1) から (4) までの時刻制約を満たすものが

FTA となる。

また、ある輸送者と荷物の輸送経路に割り当てた二つの実行可能な輸送時刻割当て FTA と FTA' があるとき、FTA で決定された輸送経路の各辺 e の輸送開始時刻を $t_s(e)$ と表し、FTA' で決定された輸送経路の各辺 e の輸送開始時刻を $t'_s(e)$ と表すとする。ここで、全ての辺 e において、 $t'_s(e) \leq t_s(e)$ を満たすならば FTA と FTA' の関係を $\text{FTA}' \ll \text{FTA}$ と表すとする。更に、ある辺順序割当て集合の FTA のうち各々の辺の輸送時刻が最も早い輸送時刻に割り当てられたものを **最早な FTA** と呼び、FTA* と表す。つまり FTA* とは、ある辺順序割当て集合の任意の FTA において、 $\text{FTA}' \ll \text{FTA}$ となる時刻割当てである。

ここで、与えられた辺順序割当て集合 A に FTA が存在すれば命題 2.1 より目的関数の値は定まる。そこで我々は FTA を求める貪欲アルゴリズムを提案する。全体の方針は輸送が可能となった荷物から順に輸送を行い、全ての荷物の輸送が完了したときに荷物と輸送者が目的地点に制限時刻までに到着しているか判定し、FTA を求めるアルゴリズムである。

まず荷物の輸送が可能となった状態について考える。この状態では輸送したい辺の始点に輸送者と荷物の両者が到着している。そのため、輸送前後関係 $r(a, a')$ において a' での輸送は a での輸送が完了していなければ輸送が不可能である。逆に、ある辺順序割当て a' における $r(a, a')$ の関係を満たす任意の a について、全ての a で輸送が完了していれば a' での輸送が可能である。この輸送が可能となった辺順序割当ての順番はグラフ $H(A, R)$ のトポロジカルソートによって求めることができる。

次に荷物の輸送が可能となった輸送時刻割当てについて考える。これは輸送者と荷物のそれぞれが輸送する辺の始点に一番早く到着する時刻を求め、遅い方の時刻が輸送開始時刻となる。輸送にかかる時間は輸送を行う辺の重みに等しいため、輸送終了時刻は輸送開始時刻に輸送する辺の重みを加えたものとなる。

輸送する辺の輸送終了時刻が求めれば、その時刻から出発して荷物と輸送者のそれぞれが目的地点へ目的地点に到着しなければならない上限時刻までに辿りつけるかを判定する。間に合うのであれば、荷物と輸送者の出発地点を輸送する辺の終点に書き換え、荷物と輸送者の移動可能となる時刻を輸送終了時刻に書き換える。こうすることにより、次の輸送する辺での輸送時刻を求めるとき、荷物と輸送者のそれぞれの出発地点・移動可能となる時刻から求めることが可能となる。

本手法は以下の手順からなる。ただし、 a_x とは輸送順序関係グラフ $H(A, R)$ にトポロジカルソートを行なった x 番目の辺順序割当てであり、初期値は $x = 1$ である。

- (1) $a_x(i, k, j, l)$ で輸送する荷物 b_i と輸送者 u_j が対応する辺 e_a の始点に到着する時刻を求める。
- (2) b_i と u_j とで遅い方を e_a での輸送開始時刻として、この辺の輸送終了時刻を求める。
- (3) b_i と u_j の出発地点を e_a の終点に更新し、 b_i と u_j の移動可能となる時刻を (2) で求めた輸送終了時刻に更新する。
- (4) b_i と u_j のそれぞれが目的地点に辿り着く下界時刻を求め、制約時刻を満たしているか判定し、満たしていれば x に 1 を加える。満たしていない場合はアルゴリズムを停止する。

(5) $x \leq |A|$ ならば 1. に戻る. そうでなければ, 全ての辺順序割当てについて輸送時刻割当てが完了しているので, 利益関数の値を求める.

Algorithm1 に以上で説明した手法の詳細を示す. ただし, 辺順序割当て集合 A のあつまりを Λ と表している.

上記の貪欲法は次の補題 4.11 を満たしている.

Algorithm 1 貪欲アルゴリズム

```

1:  $Cost^* = \infty$ 
2: while  $\Lambda \neq \emptyset$  do
3:    $\forall A$  を  $\Lambda$  から取り出す
4:    $Cost = Greedy(B, U, A, 1)$ 
5:   if  $Cost < Cost^*$  then
6:      $Cost^*$  を  $Cost$  に更新
7:     FTA を更新
8:   end if
9: end while
10: FTA と  $Cost^*$  を出力
11: procedure GREEDY( $B, U, A, x$ )
12:   if  $x \leq |A|$  then
13:      $a_x(i, k, j, l) \in A$  ▷ 輸送を行う辺順序割当て
14:      $t^b = t_d(b_i)$ 
15:     ▷ 荷物の輸送する辺の始点に到着する時刻
16:      $t^u = t_d(u_j) + W(s(u_j), v_s(e_{a_x}))$ 
17:     ▷ 輸送者の輸送する辺の始点に到着する時刻
18:      $t_s(e_{a_x}) = \max(t^b, t^u)$  ▷ 輸送開始時刻
19:      $t_d(e_{a_x}) = t^s + w(e_{a_x})$  ▷ 輸送終了時刻
20:      $s(b_i) = v_d(e_{a_x})$ 
21:      $t_s(b_i) = t^b$  ▷ 荷物の更新
22:      $s(u_j) = v_d(e_{a_x})$ 
23:      $t_s(u_j) = t^u$  ▷ 輸送者の更新
24:      $t^b = t_s(b_i) + W(s(b_i), d(b_i))$ 
25:     ▷ 荷物が目的地点に最早で到着する時刻
26:      $t^u = t_s(u_j) + W(s(u_j), d(u_j))$ 
27:     ▷ 輸送者が目的地点に最早で到着する時刻
28:     if  $t^b > t_d(b_i)$  or  $t^u > t_d(u_j)$  then
29:       ▷ 目的地点に時刻制約までに到着できない場合
30:       return  $\infty$ 
31:     else
32:       return  $Greedy(B, U, A, x + 1)$ 
33:     end if
34:   else
35:     return  $Rwd(U)$  ▷ 利益関数の値を返す
36:   end if
37: end procedure

```

[補題 4.11] 与えられた辺順序割当て A に FTA となる時刻割当てが存在するならば, 提案した貪欲アルゴリズムで FTA を求めることができる. さらに提案した貪欲アルゴリズムによって導き出した FTA は FTA* である.

【証明】

A に対してある FTA が存在し, その FTA の輸送する各辺 e での輸送開始時刻 $t_s(e)$ と輸送終了時刻 $t_d(e)$ を, 貪欲アルゴリズムで求めた FTA' の輸送する各辺での輸送開始時刻と輸送終了

時刻に変換しても時刻制約 (1) から (4) を満たしたままであること示して証明する. ただし, A に対する輸送前後関係を R とし, 輸送順序関係グラフ $H(A, R)$ にトポロジカルソートを行なったものを, a_1, a_2, \dots, a_n とする. また, a_x におけるある FTA での輸送開始時刻を $\hat{t}_s(e_{a_x})$, 輸送終了時刻を $\hat{t}_d(e_{a_x})$ とする. まず, $a_1(i, k, j, l)$ における輸送を考え, 貪欲アルゴリズムでの輸送開始時刻と輸送終了時刻を求める. ここで, a_1 は $H(A, R)$ をトポロジカルソートによって並べられた 1 番目のノードであるため, a_1 に入ってくる辺は存在しない. つまり, $\forall r(a_y, a_x) \in R$ となる y は存在しない. よって, $k = 1, l = 1$ となる. これは, a_1 での輸送が荷物も輸送者も共に輸送順序が 1 番目の輸送であることを指している. そのため, 荷物と輸送者はそれぞれの初期地点にいる. 荷物 b_i と輸送者 u_j のそれぞれが e_{a_1} の始点に到着する時刻を t^b, t^u とすると, $t^b = t_s(b_i), t^u = t_s(u_j) + W(s(u_j), v_s(e_{a_1}))$ となる. よって貪欲アルゴリズムによって求められた a_1 での輸送開始時刻 $t'_s(e_{a_1})$ と輸送終了時刻 $t'_d(e_{a_1})$ は, $t'_s(e_{a_1}) = \max(t^b, t^u), t'_d(e_{a_1}) = t'_s(e_{a_1}) + w(e_{a_1})$ となる. $t'_s(e_{a_1})$ は b_i と u_j それぞれの初期地点から移動可能となった時刻に移動した時刻であるため, e_{a_1} の始点へに b_i と u_j が共に到着している最も早い時刻である. よって, $t'_s(e_{a_1}) \leq \hat{t}_s(e_{a_1})$ であり, $t'_d(e_{a_1})$ も同様に, $t'_d(e_{a_1}) \leq \hat{t}_d(e_{a_1})$ である. つまり貪欲アルゴリズムで求められた輸送開始時刻及び輸送終了時刻に変更したとき, b_i と u_j それぞれの目的地点に到着した時刻が変更する前より遅くなることはない. よって, (3) と (4) の時刻制約は守られたままである. さらに, 貪欲アルゴリズムによって求められた, b_i が出発地点から移動を開始する時刻 $T_s(b_i)$ と u_j が出発地点から移動を開始する時刻 $T_s(u_j)$ は, 共にそれぞれが移動が可能となった時刻であるため, $T_s(b_i) = t_s(b_i), T_s(u_j) = t_s(u_j)$ である. 故に (1) と (2) の時刻制約は守られたままである. したがって, a_1 の輸送開始時刻と輸送終了時刻を変更しても (1) から (4) の時刻制約を満たし, 最早の時刻での輸送である.

次に, 貪欲アルゴリズムに従い b_i と u_j の出発地点を e_{a_1} の終点に更新し, b_i と u_j の移動可能となる時刻を $t'_d(e_{a_1})$ に更新し, $H(A, R)$ から a_1 とそれに隣接した前後輸送関係を除いた $H'(A', R')$ について考えると, a_2 での輸送を荷物も輸送者も共に輸送順序が 1 番目の輸送と見なすことができる. よって, a_2 についても a_1 と同様の議論が可能となり, a_2 の輸送開始時刻と輸送終了時刻を変更しても (1) から (4) の時刻制約を満たし, 最早の時刻での輸送である. a_3 以降も帰納的に同様の議論が可能であるため, 全ての辺順序割当ての輸送開始時刻と輸送終了時刻を変更しても (1) から (4) の時刻制約を満たし, 全ての辺順序割当てでの輸送が最早の時刻での輸送である. したがって, 貪欲アルゴリズムで FTA を求めることが可能であり, その FTA は FTA* となる. ■

5. 実験

5.1 データセット

本論文の提案手法は計算量が大きくなるものと予想されたため, 非常に規模の小さいデータセットで行った. データセットは以下のように設定した.

- $V = 20, E = 35$ であるグラフ
- 輸送者の出発地点と目的地はランダム, 出発時刻と到着時刻は 0~1440 分の範囲で 15 分刻みでランダム.
- 荷物の出発地点と目的地はランダム, 出発時刻は 0~720 分の範囲で 60 分刻みでランダム, 到着時刻は 1080~1440 分の範囲で 60 分刻みでランダム
- 利益関数のパラメータ β_1, β_2 は $\beta_1 = 1.5, \beta_2 = 1$ に設定

5.2 実験結果

本研究の問題にはどのような性質でありどの程度計算量が大きくなるかを見るため, 四つのデータセットをサンプルとして作成して実験した. それぞれのサンプルにおける結果は表の通りである.

	サンプル 1	サンプル 2	サンプル 3	サンプル 4
荷物数	2	2	2	2
輸送者数	2	2	2	3
全荷物経路辺数	7	8	9	8
輸送組合せ数	128	256	512	6,561
辺順序割当て集合数	40,320	362,880	3,628,800	1,814,400
DAG 数	1,536	5,136	16,032	71,766
DAG 判定時間 (sec)	62.059	573.774	5650.181	2818.151
QTA 判定時間 (sec)	0.021	0.051	0.139	0.719
累計時間 (sec)	62.080	573.825	5650.32	2818.87

表 提案手法の実験結果

表を見ると輸送者数や全荷物経路辺数が増えると辺順序割当て集合数が大きく増加していることがわかる. 具体的には, 全荷物経路辺数を $|E|$, 輸送者数を $|U|$ とすると, 輸送組合せ数は $|U|^{|E|}$ となり, 辺順序割当て集合数は最低でも $|U|^{|E|}(|E|/|U|)^{|U|}$, 最高は $|U|^{|E|}(|E|)^{|U|}$ である. 一方, DAG 数は辺順序割当て集合数に比べ増加量は少ない. 補題 4.9 は DAG が関係しているという問題の性質を明らかにしただけではなく, 厳密アルゴリズムの計算時間を大幅に短くするのも役立つ. つまり, DAG である辺順序割当て集合を直接列挙することができれば, 本論文の手法より計算量を抑えた手法を考案できると期待される.

6. おわりに

本研究では宅配業者の負担軽減に繋がる新しい輸送モデルとして, 一つの荷物を複数人によって中継して輸送するというのも考慮した中継可能な協調型輸送モデルを提案し, このモデルの最適化問題の厳密解アルゴリズムを考案した. さらに, この問題には DAG が密接に関係していることも発見し, 計算量をより短くする手がかりにもなることがわかった.

本研究では, 荷物という物を輸送することについて取り組んだが, ヒッチハイク等の人を輸送する場合にも応用できると考えており, その場合には人の待ち時間も考慮する必要がある. また, 将来的に予想されるロボットによる無人輸送にも, ロボット間が連携して輸送するという点で貢献できると考えている.

しかし, 提案手法の問題設定には荷物の輸送経路の固定した点や一度に輸送できる荷物が一つのみにした点, 入力された荷物を全て運ぶという点で非現実的な輸送システムとなってしまった. 今後の課題として, これらの問題設定を取り除いた, 特に荷

物を全て運ぶという点を取り除き, 入力された荷物の中にはシステム側にとって不利益となる荷物が存在する場合でも対応できる手法について取り組みたい. また, DAG を利用や近似により大きくなりすぎている計算量を短くする手法の考案にも取り組みたい.

謝 辞

本研究は JSPS 科研費 JP17H06099, JP15K00423 の助成を受けたものです. ここで心より感謝申し上げます.

文 献

- [1] Da Yan, Zhou Zhao and Wilfred Ng, "Efficient Algorithms for Finding Optimal Meeting Point on Road Networks," PVLDB, vol. 4, no. 11, pp. 968979, 2011.
- [2] Elham Ahadi, Mario A. Nascimento, "k-Optimal Meeting Points Based on Preferred Paths," ACM SIGSPATIAL 2016. Poster Pater, October 31 November 3, 2016
- [3] B Yu, Z-Z Yang and J-X Xie, "A parallel improved ant colony optimization for multi-depot vehicle routing problem," Journal of the Operational Research Society, 62, 183-188, 2011
- [4] Said Salhi, Arif Imran and Niaz A. Wassan, "The Multi-Depot Vehicle Routing Problem with Heterogeneous Vehicle Fleet: Formulation and A Variable Neighborhood Search Implementation," Computers & Operations Research 52, 315325, 2013
- [5] Abel Garcia-Najera, John A. Bullinaria, "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows," Computers & Operations Research 38, 287-300, 2011
- [6] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, Christian Prins, "A Hybrid Genetic Algorithm with Adaptive Diversity Management for a Large Class of Vehicle Routing Problems with Time Windows," Computers & Operations Research 40, 475489, 2013
- [7] Zhang X, Asano Y, Yoshikawa M, "Mutually beneficial Confluent Routing", IEEE Transactions on Knowledge and Data Engineering, Volume: 28, No. 10, 2681-2696, 2016.
- [8] Kazuki Takise, Yasuhito Asano and Masatoshi Yoshikawa, "Multi-user Routing to Single Destination with Confluence", ACM SIGSPATIAL 2016. Poster Pater, October 31 November 3, 2016
- [9] Cao B, Alarabi L, Mokbel M. F, Basalamah A, "A scalable dynamic ride sharing system", Proceedings of 16th MDM, 413, 2015.
- [10] Drews F, Luxen D, "Multi-hop ride sharing", Proceedings of 6th SoCS, 7179, 2013.
- [11] Kahn, A. B, "Topological sorting of large networks", Commun. ACM, Vol. 5, 558562, 1962.
- [12] Tarjan, R. E, "Edge-disjoint spanning trees and depth-first search", Acta Informatica, Vol. 6, 171185, 1976, 10.1007/BF00268499.