

貨客提携ネットワーク上の動的システム最適 配分とそのシャープレイ評価

黛 風雅¹・羽藤 英二²

¹学生会員 東京大学大学院 工学系研究科社会基盤学専攻 (〒 113-8656 東京都文京区本郷 7-3-1)
E-mail: mayuzumi@bin.t.u-tokyo.ac.jp

²正会員 東京大学大学院教授 工学系研究科社会基盤学専攻 (〒 113-8656 東京都文京区本郷 7-3-1)
E-mail: hato@bin.t.u-tokyo.ac.jp

本研究では、貨客混載車両の経路決定問題を組合せ最適化問題として記述し、ゼロサプレス型二分決定グラフを用いた経路列挙索引による、動的な高速マッチング処理手法をリアルタイム解法として提案した。先行リクエストを優先するか否かに基づき、マッチング方策をシステム最適(SO)方策とFirst-Come-First-Serve(FCFS)方策で区別し、より時間を要するSO方策について数十秒のオーダーでリアルタイム処理が可能な例を数値計算で示した。また、FCFS方策で成立する貨客の提携に対し、特性関数で与えられる提携値を計算し、マッチングへの限界貢献度をShapley valueで評価する枠組みを提案し、旅客の往復利用が車両の効率的な運行を妨げる可能性を示すとともに、貨客混載車両への利用者配分方法を議論した。

Key Words: *matching, ZDD, Shapley value, passenger-and-freight coalition, On-line Dial-a-ride problem*

1. はじめに

デマンド交通は、利用者のリクエストをリアルタイムに処理しつつ、動的にその経路を決定することが、サービスとしての利便性向上のために必要とされてきている。近年のライドヘイリングサービスでは、利用者がスマートフォン等を通じて利用のリクエストを行い、オペレータが車両の経路を決定するようなアルゴリズムが一般的である。ライドヘイリングのオンデマンド性に加え、ライドシェアでは個別に利用希望を表明する複数の利用者が同乗する状況を想定する。このような利用は車両の効率的な利用の観点で重要である。

デマンド型・シェア型の交通は公共交通としての性質が強く、車両の利用者として移動目的のみに限らず、配達目的の利用者も受け入れるような試みもある。Ghilas et al.(2013)¹⁾やLi et al.(2014)²⁾では、需要やネットワークの設定として明示的に貨客を区別したシナリオで、車両のスケジューリングアルゴリズムを評価している。このような貨客混載の車両運行のシステムは、都市部よりも生産者が点在している地方部・郊外部で特に期待されている。

DARP(Dial-a-ride Problem)において、車両の運行経路を決定する際に全てのリクエストに最初から参照するバッチ処理ができず、逐次的なリアルタイム処理が要求される問題はOn-line DARPと称される。On-line DARPでは、車両の運行開始後に発生する顧客のリクエスト(En-route リクエスト)の存在を仮定する場合と、車両の

運行開始前に順次発生する予約(Pre-trip リクエスト)のみを扱う場合がある。問題の性質として、En-route リクエストを処理する際には、過去の運行経路を踏まえる必要があり、これによって経路を決定する際の解空間が限定される点が、Pre-trip のみの場合と異なる。また、実装を想定する場合には、各時点での逐次的な経路再計算に求められる計算性能も異なる。En-route リクエストの受入可能可否の判定や最適経路探索に際して、スケジューリングによる遅延を発生させないことがアルゴリズムの要件であり、数秒から数分程度のオーダーの計算時間がマッチング成立率の向上のために要求されることになる。

リクエストに前後関係が存在する場合、最適な貨客の組合せとその経路決定において順序による影響も評価する必要がある。システム最適の観点で望ましい解:SO(System Optimum)方策が、先行するリクエストを優先する場合:FCFS(First-Come-First-Serve)方策では達成されないこともある。車両が共有財産である場合にはSO方策による公平な運行が求められるが、先行する利用者の観点に立てば、サービスの便益ポテンシャルが十分に発揮されているとは言い難い。また、方策決定は採算性にも影響されるため、サービス主体の公共的性質によって便益の考え方自体が異なることもある。

以上を踏まえ、本研究では、需要が疎なネットワークを想定し、貨客が提携する場合の経路決定問題を扱う。解法として、逐次的な需要の発生にリアルタイムに対応するOn-line アルゴリズムを提案し、その計算性

能を評価する。また、提案する On-line アルゴリズムを用い、貨客の提携を評価する Shapley value に基づくフレームワークを提案し、貨客混載車両への利用者配分の方策について検討する。

2. 既往研究の整理

スケジューリング問題は組み合わせ最適化問題であり、NP 困難問題と知られているため、ネットワークサイズとそれに伴う計算時間が問題となる。リアルタイム解法として適用する場合には、ヒューリスティクスが採用されることが多い。(1) 節では、逐次挿入法に基づくヒューリスティックな動的アルゴリズムをデマンド交通に適用した例を挙げ、求められる計算性能や最適化問題としての条件について整理する。

貨客の組合せは、それぞれ貨客混載車両によって達成される経路によって評価する必要がある。プレイヤー間に異質性がある場合には、その組合せに対して一対一対応で特性関数としてその価値が定義されることになる。(2) 節の提携構造生成問題はプレイヤーの適当な分割によってシステム最適を試みる問題であり、特性関数の簡略表記法が重要である。ゼロサプレス型に部決定グラフ (ZDD) は、特性関数の簡略表記法として用いられているほか、高速な経路列挙索引手法として交通計画への応用例も複数見られる。(3) 節では、ZDD のデータ構造としての特徴と、それをういた経路列挙索引手法について整理した。

(1) 逐次挿入法による On-line DARP のリアルタイム解法

Jaw et al.(1986)³⁾ は、複数の Pre-trip リクエストを想定し、複数車両を割り当てる、多対多の、時間制約付き DARP (Dial-a-ride Problem) を扱った。解法アルゴリズムとしてヒューリスティックな手法を用い、逐次挿入法によってリクエスト時間の早い順に顧客を車両に割り当てるリアルタイム処理を行うことで、解の存在範囲を限定している。車両の運行スケジュールの決定問題は、各リクエストの pick-up / delivery の順序決定問題として扱われており、逐次挿入されるリクエストは、すでにマッチングが成立したリクエストのサービス水準に関する制約を悪化させない範囲で、最小のコストで達成可能なようにスケジューリングされる。サービス水準に関する制約とは、利用者と車両オペレータの双方の観点から定義されている。利用者については希望出発 / 到着時間からの乖離と総旅行時間から定義され、車両オペレータについては、新たな利用者による、既にその車両に割り当てられた他の利用者の旅行時間の増分の総和から定義される。希望出発 / 到着時間か

らの乖離をどの程度受け入れるかどうかは、利用者によって異質性があるはずであるが、ここでは許容可能な乖離時間や、その量に対する感度を一律に設定していると言う点で、分析上の課題がある。また、En-route リクエストへのリアルタイムな対応については想定できていないと筆者によって言及されている。

Tsubouchi et al.(2009)⁴⁾ は、オンデマンドバスのスケジューリングアルゴリズムにおいて新規のリクエストに対する Jaw et al.(1986)³⁾ のリアルタイム処理方策を採用し、容量制約付き・複数車両・多対多の時間制約付き On-line DARP (Capacitated On-line DARP) を扱っている。各リクエストに対して、車両割当問題を一段階目、スケジューリング問題 (各リクエストの pick-up / delivery の順序決定問題) を二段階目とする最適化問題として、スケジューリング問題のスケールが大きくならないように車両ごとの問題に分解することで、計算規模の大規模化を図った。車両数が 1 台から 6 台の場合について扱っており、計算時間が最も長くなる 6 台の場合について新たなリクエストの処理に要する時間は、それが実行可能な場合には 4.51 秒、実行不可能な場合で探索に時間を要する場合であっても 17.4 秒と、高速に求解された。しかし、そのようなアルゴリズムが交通システムとして総コストをどの程度減少させることができているか、また、個別のリクエストがシステムに与える影響について検討されていない。Tsubouchi and Yamato(2012)⁵⁾ では、スケジューリング問題において総旅行距離と乗合人数の二つを目的関数に含む最適化問題として発展させているが、車両の割当の段階で乗合人数を最大化するような割当を考えられていない。

システム最適状態を志向する公共交通への実装を考えると、リアルタイム性を有するアルゴリズムが逐次的にリクエストを現実的な計算時間で処理すると同時に、車両・利用者間の効率的なマッチングを生成する必要がある。このとき、既往の研究においては、個別のリクエストがどの程度の希望出発/到着時間からの乖離を受け入れることができるかという異質性がシステムに与える影響について検討されてきていない。

(2) 協力ゲームにおける提携構造表現

協力ゲーム理論では、提携値の総和が最大になるようにエージェントの集合を分割する問題が提携構造形成 (Coalition Structure Generation: CSG) 問題として定義されている。これは、完全集合分割問題 (1986)¹¹⁾ と等価な問題であり、分散車両の経路決定問題にも用いられている (Sandholm and Lesser 1997¹²⁾)。CSG 問題では各提携で得られる提携値を計算するため、提携値の計算法が重要となるが、ゲームによる利得が提携構造ごとに定められる特性関数に基づく場合は計算不可能に

なる傾向にある。

ブラックボックス的であるために扱いが困難とされてきた提携構造形成問題における特性関数に対し、近年では特性関数に対する簡潔な表現方法が提案されてきた。たとえば、Marginal Contribution networks: MC-nets (Jeong and Shoham 2005¹³⁾) と呼ばれる簡略表記法では、特性関数に存在する特徴を利用し、特性関数を論理的なルール集合として簡略に表現している。また、Synergy coalition group: SCG (Conitzer and Sandholm 2006¹⁴⁾) と呼ばれる表記法では、提携に属するエージェント間に正の相乗効果がある提携の利得のみ記述する特性関数の簡略化が提案されている。

特性関数の表現方法として Sakurai et al.(2011)¹⁵⁾ は、SCG による提携構造生成問題に ZDD を用いるスキームを提案しており、その有用性が示されている。

(3) ゼロサプレス型二分決定グラフ (ZDD)

二分決定グラフ (Binary Decision Diagram: BDD) は、ブール関数の有向グラフ表現であり、大規模な組合せ論理データを操作することができる。Minato(1993)⁶⁾ が提案した ZDD(Zero-suppressed BDD) も、この表現の一種であるが、グラフノードの縮約方法が異なる。縮約化は等価な接点の共有と、冗長節点の削除の 2 つの操作によるが、BDD と ZDD は、後者の冗長節点の削除規則のみが異なる。等価な接点の共有とは、0-枝でも 1-枝でも分岐先が同じ場合は分岐節点を削除して直結することである。BDD では、共通の行き先をもつ分岐節点が 2 個以上あれば 1 個にまとめて他を削除する、という削除規則を適用する。一方 ZDD では、1-枝が 0-末端節点を直接指している場合に、この節点を削除するという削除規則を適用する。組合せ集合に無関係な接点を削除する規則は、疎な組合せ集合に対して特に効果的である (湊 2015⁷⁾)。

BDD / ZDD は元来は VLSI(大規模集積回路) 設計のための論理関数の照合や論理回路の単純化に用いられてきたものであり、ZDD も同様の動機で開発されたものであった。組合せ集合はしかし、VLSI 論理設計以外にもさまざまな分野で現れる基礎的な概念であるため、知能情報処理の局面に現れるグラフの問題や組合せ問題にも効果的に利用できることが示されていった。

二分木による組合せ表現では、その生成規則として重複する同一の選択枝が発生させないことが必要となる。したがって、ZDD を現実のネットワークを想定した経路列挙のために採用する場合、経路内で同一のリンク上を移動するというサイクリックフローの存在が問題となる。二分木生成におけるそのような課題は、BDD や ZDD がデータマイニングのなかでも頻出アイテム集合マイニングの手法としての可能性を見出されるよう

になったことで検討されることとなった。

Kurai et al.(2006)⁸⁾ は重複する組合せの要素を、その要素が出現する順番によって符号化することで、重複する要素を表現するという手法を提案した。しかし、文字の種類と出現位置の 2 つを組にして符号化したものを ZDD のアイテム変数として割り当てるという手法では、組合せ全列挙を試みる際に組合せ数が指数関数的に増加するため、計算負荷が高いものである。そこで Loekito et al.(2010)⁹⁾ は要素の重複を扱うために、seqBDD を開発し提案している。seqZDD は ZDD の縮約規則を 0-枝側にしか適用せず、1-枝には順序付けの制約を外すというものである。データマイニングのための工夫として用いられた、符号化による重複要素の表現を参考に、Hayakawa and Hato(2018)¹⁰⁾ では、時間構造化ネットワークの概念を応用して、離散時間タイムステップごとに発生する全ての選択可能なリンクを表現するように、タイムステップによる符号化をネットワーク上の各リンク選択に対して行った。このようなネットワーク表現によって、ノード間の移動のみならず、同一ノード上での滞在行動も表現可能である。

(4) 本研究の目的

以上を踏まえ、本研究では交通・物流の異質性のある需要パターンに対応する貨客混載車両の User・Driver 間のマッチングを想定し、ZDD の経路列挙索引を生かした逐次的なリクエスト処理による、動的なマッチングアルゴリズムを提案する。また、マッチングによって達成される経路 (協力ゲームにおいては提携) の評価指標として、Shapley value を用いた公平な利用者の車両への配分フレームワークを提案する。

3. 逐次列挙索引による貨客マッチングアルゴリズム

本節では、Zero-suppressed Binary Decision Diagram (ZDD) を用いた経路列挙索引手法について示し、貨客混載車両の User と Driver を逐次的にマッチングさせる動的なアルゴリズムについて説明する。

User は、Driver の運行開始前と開始後の両方で旅程を要求することができ、それぞれを Pre-trip リクエストと En-route リクエストと呼ぶ。本節で提案するアルゴリズムでは、それらのリクエストを受理し、Driver の経路、またそれに伴う User の経路を決定するオペレータの存在を想定する。オペレータは User からの要求があるごとに逐次的に実現可能な経路を再列挙し、総旅行距離や総移動コストを最小化する経路の組合せを探る。

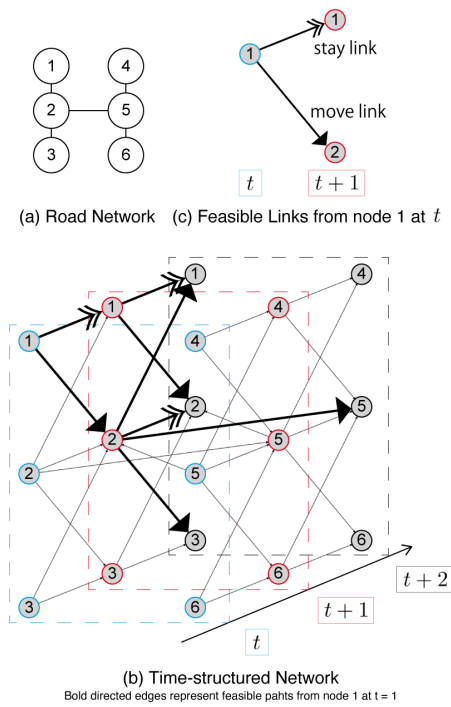


図-1 経路の時間構造化ネットワーク表現: (a) 道路ネットワークを (b) 時間軸方向に拡張し, (c) move link: 移動リンクと stay link: 滞在リンクを導入する.

(1) 組合せ最適問題のための時間構造化ネットワーク表現

ネットワークが図-1(a)のように表現されるとき, リンク間の移動を離散時間によって符号化すると, 図-1(b)のように表される. これを, $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ と定義する.

時間構造化されたネットワーク上のリンク $e \in \mathcal{E}$ は図-1(c)のように, 移動リンクと滞在リンクの2つに分類することができる. それぞれの移動リンクにはネットワーク上の距離に相当する, τ_e が紐づけられ, 滞在リンクについてもその滞在コストが紐づけられる. 時間方向に拡張したネットワークは, $t \in T = \{0, \dots, T_{max} - 1\}$ の離散タイムステップを持つとする.

ノード $n \in \mathcal{N}$ へ流入するリンク, あるいはノード $n \in \mathcal{N}$ から流出するリンクは, それぞれ $I(n) \subset \mathcal{E}$, $O(n) \subset \mathcal{E}$ と表す. 図-1(b),(c) 中の $e_{(1,2)}^t$ は, 図-1(a)で示されたノード1からノード2へのエッジ上を, ある離散タイムステップ $t \in T$ において移動する, 移動リンクを示す例である. 表-1に, 上記の時間構造化ネットワーク表現に関する変数をまとめる.

(2) 実行可能経路の列挙索引化手法: ZDD

User と Driver の各時刻における実現可能な経路を評価するために, システムは実現可能な経路の全ての組合せを列挙し, 決定木によって構造化することを考える. ここで, 組合せ集合を論理関数によって表現する列挙

表-1 時間構造化ネットワークに関する記号一覧

Time horizon	
T	Discretized time horizon
<hr/>	
Spatial Network	
$\mathcal{G} = (\mathcal{N}, \mathcal{E})$	Time-structured network
\mathcal{N}	Set of nodes in \mathcal{G}
\mathcal{E}	Set of links in \mathcal{G}
$I(n) \subset \mathcal{E}$	Set of inflow links to node $n \in \mathcal{N}$
$O(n) \subset \mathcal{E}$	Set of outflow links from node $n \in \mathcal{N}$
\mathcal{N}	Set of nodes in the spatial network
\mathcal{E}	Set of links in the spatial network

索引手法である, ZDD: Zero-suppressed Binary Decision Diagrams (Minato 1993⁶⁾)を導入する. また, 本研究でのネットワーク表現では, Hayakawa and Hato (2018)¹⁰⁾の手法を採用し, 時間構造化ネットワーク上のグラフにおけるリンク選択を decision variable として, ZDD のノードとして設定する. これによって, 実ネットワークで発生するループ構造を排し, それと同時に滞在行動の表現も可能なネットワーク表現を実現する.

図-2では, 実行可能経路の二分決定木による表現方法と, 二分決定木から BDD / ZDD への縮約化を示している. 図-2(a)は, Node 1 にいる User が Node 2 に2タイムステップ以内に到着するという需要を満たすときの実行可能経路を表している. タイムステップ t における Node 1 と 2 本のリンクで結ばれるノードを考慮し, 図-2(b)のように二分決定木で条件を満たす経路を全て列挙すると, 時間構造化ネットワーク上の選択可能なリンク数である 4 を指数とする 2^4 通りの組み合わせを列挙しなければならない. 図-2(b)の二分決定木で表現される経路を縮約することで得られる ZDD を図-2(c), 対して BDD を図-2(d)で示した. ZDD は BDD の列挙法に比べ, 実行可能経路の列挙索引化に優れた性能を発揮することが分かる.

(3) User と Driver の定義

貨客混載システムは, 2種類の User の存在を仮定する. 一つは, 旅客 (Passenger) の移動に伴う交通需要, もう一つは貨物 (Freight) の移動に伴う配送需要である. それぞれの行動論的な性質・財の種類に基づく性質が需要の時間的・空間的な分布に影響することとなる. User / Driver に関する記号を表-2にまとめる.

User の集合を I^u , その車両の Driver の集合を I^d , また, それらをあわせたマッチングの系に属する個人の集合を $I = I^u \cup I^d$ と定義する. 各個人 $i \in I$ の経路は, 出

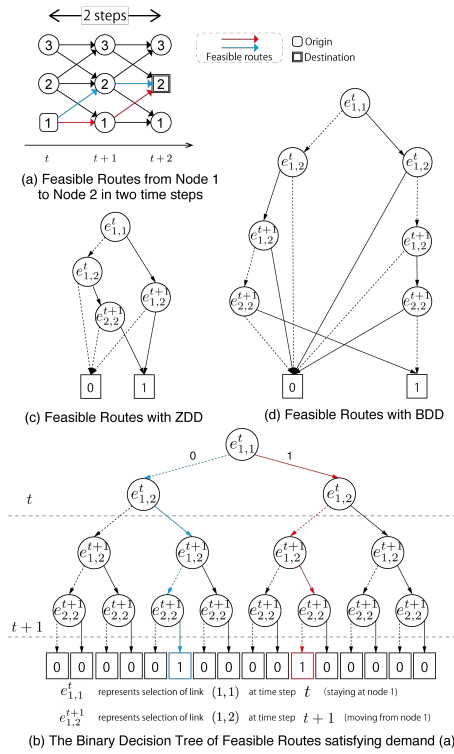


図-2 ZDD を用いた時間構造化ネットワーク上のリンク選択の組み合わせの列挙索引化手法

発地 n_i^b , 目的地 n_i^e , タイムウィンドウ T_i , 必須活動場所 $n_i^{a,j}$, 必須活動場所への滞在時間 $t_{a_i,j \in J_i}^{min}$ (最小滞在時間の存在を仮定する), 目的地への必須到着時間 $t_i^{n_e}$ によって定義される. タイムウィンドウ $T_i = [t_i^b, t_i^e]$ のそれぞれ開始時間と終了時間は, 必ずしも出発時間と到着時間を意味せず, このタイムウィンドウ内で User の経路に関する制約条件を満たす全ての出発時間・到着時間が選択される. ここで, 必須活動場所は必須活動数: J_i 個の要素を含む必須活動リスト $A_i = \{a_{i,1}, \dots, a_{i,J_i}\}$ によって定義されている. また Driver については, 車両容量 L_{max}^i が定義され, 貨客のマッチングにおいては積載量 L^i はこれを超えないことが制約条件となる.

リンク上を移動する User について単位時間あたりに発生する移動コストは C_i^M で表され, 各移動リンクの移動コストは C_i^M とリンク移動時間 t の積で, $\tau_e = C_i^M t$ と表される.

滞在コストは $C_{T_i,n}^S$ で表され, 同じノードに滞在する User の単位時間あたりのコストである. CoS は滞在场所によっては負になることがある. 移動目的の User にとって非目的地における滞在は負の効用をもつ待ち時間であり, CoS は同一ノードでの滞在時間に対して単調増加関数となる. Driver や貨物の配送目的の User にとっては, CoS は単位時間あたり一定のコストとする. したがって, 以下の条件を満たす閾値 $T_{i,n} = T_{i,n}^{th}$ が存

表-2 User / Driver の実行可能経路に関する記号

I	Set of individuals in the network
I^u	Set of users of shared vehicle
I^d	Set of drivers of shared vehicles
n_i^b	Origin node of each individual
n_i^e	Destination node of each individual
$t_i^{n_e}$	Mandatory arrival time at the destination
$n_i^{a_i,j}$	Mandatory activity node for each activity
J_i	The number of mandatory activities
A_i	Set of Mandatory activities
T_i	Active time-window
t_i^b	Beginning time of Active time-window
t_i^e	Ending time of Active time-window
$t_{a_i,j}^{min}$	Minimum visit time at each activity
$T_{i,n}$	Total staying time at each node
$T_{i,n}^{th}$	Threshold time of efficient successive staying time at each node
β_i	Time discounted rate
L^i	Shared vehicle's capacity
L_{max}^i	Shared vehicle's maximum capacity
$C_{T_i,n}^S$	Cost of Staying at each node
C_i^M	Link travel cost per unit time
G_i	Set of selectable links $e \in \mathcal{E}$
G^t	Optimum routes for all individual at t
Z_i	Set of individual feasible routes enumerated and indexed with ZDD
Z^t	Set of integrated feasible routes enumerated and indexed with ZDD at t

在することになる.

$$C_{T_{i,n}}^S > 0 \wedge C_{T_{i,n}-1}^S \leq 0. \quad (1)$$

活動から得られる効用の飽和効果は satiation parameter (Bhat 2005¹⁶) と称され, 逐次的な配分において特定の活動への配分の切り上げを可能としている.

実行可能経路集合 Z_i は時間構造化ネットワークにおけるリンク集合 G_i から抽出される部分集合を元として持つ集合族である. 実行可能経路集合 Z_i が複数個人間で集合演算によってタイムステップ t において統合される時, これを Z^t と表す. Z^t は, Z_i に含まれる複数の実行可能経路の和集合によって組合せ全列挙することができる. ここでは, Z^t を実行可能経路集合族と呼び, Z_i と区別するが, 両者はいずれも集合族である. また, こうして得られた Z^t に含まれる最適経路集合を G^t と表す.

(4) ZDD を用いたマッチングアルゴリズム

本項では、User と Driver をマッチングさせ、User のリクエストを動的に処理するアルゴリズムについて説明する。表 (4) では、あるタイムステップ t において発生した新規の User リクエストを受け付け、実行可能な経路を再計算するプロセスを示している。また、直前のリクエストが t^- に発生しているものとする。

(STEP 4) の最適化問題は、移動に加えて滞在によるコストも想定し、目的関数として将来効用を割り引く時間割引率を外生的に与える場合: 式 (2) と、総走行距離のみ想定する場合: 式 (3) を想定し、以下の式 (2)-(8) の形で表す。決定変数 $\delta_i, \eta_{i,e \in E,t}, \theta_{i,n \in N,t}$ はそれぞれ、個人 i のリクエストが受け入れられること、個人 i がタイムステップ t にリンク e に流入すること、個人 i がタイムステップ t にノード n に留まることを表している。

$$\min_{\delta_t, \eta_t, \theta_t} TTC^t = \min_{\delta_{>t}, \eta_{>t}, \theta_{>t}} TTC^{t+1} \quad (2)$$

$$+ \sum_{i \in I} \delta_i \beta_i \left(\sum_{e \in \mathcal{E}} \eta_{i,e,t} \tau_e C_i^M + \sum_{n \in \mathcal{N}} \theta_{i,n,t} C_{T_i,n}^S \right)$$

or

$$\min_{\delta_t, \eta_t, \theta_t} TTC^t = \min_{\delta_{>t}, \eta_{>t}, \theta_{>t}} TTC^{t+1} \quad (3)$$

$$+ \sum_{i \in I} \delta_i \sum_{e \in \mathcal{E}} \eta_{i,e,t} \tau_e$$

subject to

$$\forall n \in \mathcal{N}, \forall i \in I, \forall t \in T: \quad (4)$$

$$\left(\theta_{i,n,t} + \sum_{e \in O(n)} \eta_{i,e,t} \right) - \left(\theta_{i,n,t-1} + \sum_{e \in I(n)} \eta_{i,e,t-\tau_e} \right)$$

$$= \begin{cases} \delta_i & t = t_i^b \\ -\delta_i & t = t_i^e \\ 0 & \text{otherwise} \end{cases}$$

$$\forall e \in \mathcal{E}, \forall t \in T: \sum_{i \in I^u} \eta_{i,e,t} \leq \sum_{i \in I^d} \eta_{i,e,t} L_i^{max}, \quad (5)$$

$$\forall n \in \mathcal{N}, \forall i \in I: \theta_{i,n,T_{max}} = 0, \quad (6)$$

$$\forall e \in \mathcal{E}, \forall i \in I: \eta_{i,e,T_{max}} = 0, \quad (7)$$

$$\forall i \in I^d: \delta_i = 1, \quad (8)$$

式 (4) は、リクエストが受け入れられた User についての時間構造化ネットワーク上の流量保存則を表しており、リクエストが受け入れられなかった場合は常に流量が 0 となる。式 (5) は、個人 i がタイムステップ t においてリンク e への流入量が、そのリンクを通過する車両の容量以下になることを表す制約条件である。式 (6)、式 (7) は、最終タイムステップ T_{max} 以降には移動・滞在活動が発生しないことを表しており、式 (8) は、全ての Driver が必ずマッチングに加わることを表している。

t において発生する新たなリクエストを考慮して新た

表-3 逐次列挙索引を用いたマッチングアルゴリズム

(STEP 1) Individual Network Definition G_i

- 出発地 n_i^b , 目的地 n_i^e , タイムウィンドウ T_i , 必須活動場所 $n_i^{a_i,j}$ を input とする。
- 時間構造化ネットワーク上で選択可能なリンク $e \in \mathcal{E}$ の集合 G_i を列挙する。

(STEP 2) Enumeration-Indexing of Individual Feasible Paths Z_i

- ZDD を用いて各個人の実行可能経路集合 Z_i を列挙索引化。ZDD node は $\eta_{i,e \in E,t}$ に対応。
- 必須活動で限定操作を行う。最小滞在時間 $t_{a_i,j \in J_i}^{min}$ を満たさない経路を削除。

(STEP 3) Integrating New Request and Re-calculate Z^t

- 集合族の演算によって新たな User の経路 Z_i を先行する User / Driver の経路 Z^{t^-} と統合*: $Z^{t^-} \sqcup Z_i$
- 先行する User がいない場合 (最初の User) については、Driver の経路のみからなる Z_i と統合する。
- 容量制約で限定操作を行う。Driver の移動リンク上と同時・同一リンクに発生しない User の移動リンクを含む経路は削除。
- $Z^t \neq \emptyset$ の場合はリクエストを受け入れ、 Z^{t^-} を Z^t に更新。
 - (SO 方策) Z^t として全ての実行可能経路集合族を保存
 - (FCFS 方策) Z^t として User i の最小コストを達成する実行可能経路集合族のみ保存
- $Z^t = \emptyset$ の場合はリクエストを受け入れず、 Z^{t^-} を更新せずにプロセスを停止。

*JOIN 演算では、それぞれに含まれる要素どうしの組合せ集合が導かれる。経路を表す集合を要素としてもつ集合属 A(要素 a, b) と集合属 B(要素 b, c) の存在を仮定とき、 $a-b, a-c, b-b, b-c$ の 4 通りを要素にもつ $A \sqcup B = \{a \cup b | a \in A \cap b \in B\}$ が生成される。

(STEP 4) Search for optimal paths G^t from Z^t

$Z^t \neq \emptyset$ の場合、総コスト (総走行距離または総旅行コスト) を目的関数とする最適化問題: 式 (2)-(8) を解き、User / Driver の最適経路 G^t を更新。

に経路を再計算する際に、式 (2)、式 (3) の定式化ではいずれも総旅行コストを目的関数にしているが、先行するリクエストを優先するか否かで最適化方策を SO 方策と FCFS 方策で区別する。User のリクエスト順を考

慮せず、最大マッチングを実現しつつ全体最適解を求める方策を SO 方策とし、User のリクエスト順を考慮して、先行するリクエストの最適経路を保存したうえで新たな User のリクエストを受け入れる方策を、FCFS 方策とする。FCFS 方策では、新たな User のリクエストの受け入れが不可能になる可能性が高く、最大マッチングを達成しようとする観点とは対立する。

(5) En-route リクエストに対する適応的列挙索引

SO 方策と FCFS 方策は、実現される解が異なるという点に加えて、求解時に参照する組合せ集合のサイズに大きな差が見られる。SO 方策では、先行するリクエストについて実現可能な全ての経路を保存したまま表 (4) の (STEP3) における JOIN 演算が実行されるためである。

ネットワークサイズが大きいことで、求解がそもそも不可能になる場合もあるほか、最適経路の求解性能も著しく減少する。特に En-route リクエストに対する即時性が損なわれることで、User の急なリクエストに対応できない可能性がある。

したがって、En-route リクエストに関する Z^i と Z^t の JOIN 演算において、集合演算の工夫によってネットワークサイズを落とす適応的な逐次列挙索引手法 (図 (5)(a)) を採用した。Pre-trip リクエストによって $t = 0$ に決定された経路で運行している場合に、 t で En-route リクエストが発生する場合を考える。 G^0 に従う経路を通過したというログを参照して、動的に解候補を Z^0 から $Z^0[t]$ に縮減する手続きを実行することで、JOIN 演算の前に Z^t のサイズを落とすような手続きである。一方、図 (5)(b) では、そのような動的な手続きを行わない場合である。

(6) 提携構造の Shapley value による評価

列挙索引手法である ZDD は、貨客と車両による協力ゲームにおける提携を評価する上で有効なデータ構造を有しているといつて良いだろう。本節では、FCFS 方策によって形成される提携を、ZDD を用いて Shapley value に基づき評価するフレームワークについて説明する。

有限集合 $N = 1, 2, \dots, n$ でプレイヤーの集合を表し、 N の部分集合 $S \subseteq N$ を提携 (coalition) と呼ぶ。協力ゲームでは、通常どの提携も形成可能であるとし、任意の提携 S について、各プレイヤー $i \in N$ が S のメンバーか否かであるため、 2^n 個の可能な提携が存在すると仮定する。任意の提携 S について、 S が形成されたとき S が独立で達成できる利得の最大値を考え、このときに利得を提携値と呼ぶ。また、提携 S と提携値 $v(S)$ を対応させる実数値関数を特性関数という。

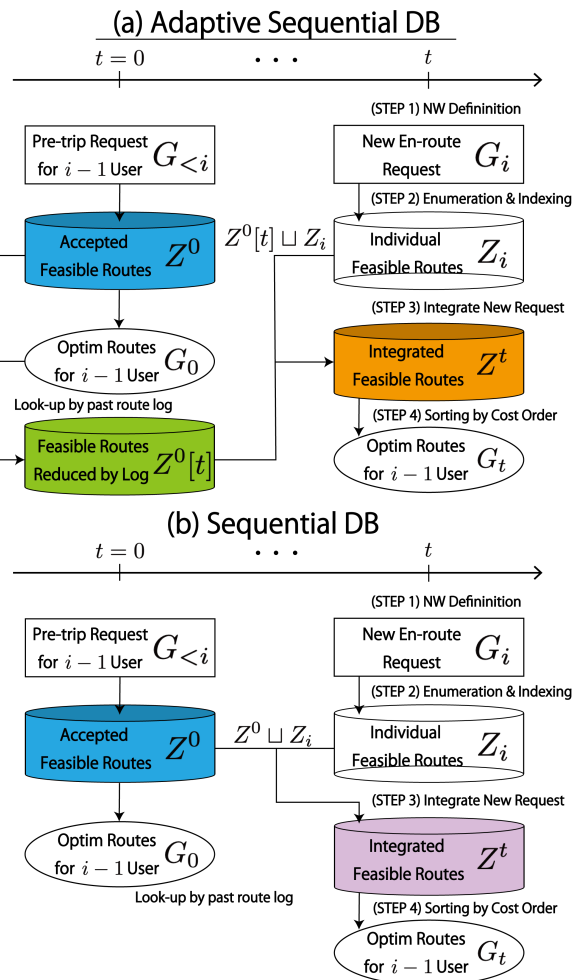


図-3 En-route リクエストに対する適応的逐次列挙索引手法

プレイヤー $i \in N$ が利得 x_i を得ることを表すベクトル $\mathbf{x} = (x_1, \dots, x_n)$ を利得ベクトルという。ゲーム (N, v) の Shapley value は、各 $i \in N$ の利得が

$$\phi(v)_i = \frac{1}{n!} \sum_{S \subseteq N \setminus \{i\}} |S|!(n - |S| - 1)! [v(S \cup \{i\}) - v(S)] \tag{9}$$

で与えられる利得ベクトル $\phi(v)$ である。

全プレイヤーが一列に整列して、先頭から $|S|$ 人が提携 S を形成しているとする。Shapley value では、提携 S を形成するとき、プレイヤーが 1 人ずつ加わっていくという提携形成を考える。プレイヤー i が最後の加入者になって、提携 $S \cup \{i\}$ が形成され、限界提携値 $v(S \cup \{i\}) - v(S)$ の貢献をするという事象を想定する。このような事象を記述する N の順列の個数は $|S|!(n - |S| - 1)!$ であり、この順列が提携する確率が、 $\frac{1}{n!} |S|!(n - |S| - 1)!$ であるから、値 $\phi(v)$ は、各提携がこうして形成されたとした場合の、プレイヤー i による限界提携値の期待値を与えることとなる。

本研究では協力ゲームの性質を鑑みて、移動にかかるコストをプレイヤーによって分担する状況を考える。

つまり、プレイヤー集合から新たにプレイヤーが参加することで、一人当たりが負担するコストを軽減することになると考える。したがってマッチングに参加することによる限界貢献度は、提携 S の総旅行距離を TTD_S とすると、

$$v(S \cup \{i\}) - v(S) = \frac{TTD_S}{|S|} - \frac{TTD_{S \cup \{i\}}}{S \cup \{i\}} \quad (10)$$

と表される。特性関数 $v(x)$ は、プレイヤー集合 x から一人当たりの総旅行距離分担への写像としており、提携構造の評価が可能になるといえる。

4. 数値実験

本章では、3.において提案したマッチングアルゴリズムの計算性能を、En-route リクエストへの応答性から検討し、マッチング方策 (SO/FCFS) の違いによる解の変化について考察する。また、需要パターンとして貨客の異質性が実行可能解や最適解に与える影響についても分析する。

(1) 実験設定

数値計算では、6 ノードからなる図-1 の空間ネットワークを $T_{max} = 15$ ステップに時拡張した時間構造化ネットワーク上の User/Driver を想定する。また、User 数を最大で 4 として、1 台の車両によって担われるものとする。User は 2 種類の旅客 (ヒト-a / ヒト-b) と 2 種類の貨物 (モノ-a / モノ-b or モノ-c) に分けられ、旅客は round trip を、貨物は one-way trip をリクエストするものとする。また、目的地以外の必須活動数は最大で 1 とした。Table -4 に User と Driver の需要・供給に関する設定を示した。モノ-c はモノ-b と Origin のみが異なるとしている。移動コストは貨物について考慮せず、必須活動や最小滞在時間も存在しないものとする。必須到着時間はモノ-a のみについて想定した。

(2) System Optimum

ヒト-a, ヒト-b, モノ-a, モノ-b(or モノ-c) による SO 方策での En-route マッチングの結果を示す。ここでは、モノ-b(モノ-c) を En-route リクエストとして扱う。モノ-b とモノ-c の単位時間当たりの滞在コスト C_i^M はモノ-a より 3 倍大きい値に設定しており、これは急な利用が必要となる財の時間価値が高いことを反映している。また Threshold Time defined in 式 1 は 3 としており、目的地での活動性質によって長期の滞在による効用減衰効果の解釈は異なるが、ここでは購買活動を想定して 1 ステップを 10 分という解釈を与えることとする。

表-5 に、図 (5)(a)(b) のそれぞれの En-route リクエスト処理手法によるマッチングでの経路列挙数とその計算

性能について整理した。 $t = 6$ において発生するモノ-b / モノ-c の En-route リクエストが 1 タイムステップ以内に処理されると仮定して、モノ-b / モノ-c が $t = 7$ から受け入れ可能である場合について示している。また、Optimum Path Search Time の列は、より時間のかかる式 2 の目的関数についての結果である。本解析は Intel(R) Xeon(R) CPU @2.00GHz, シングルコア, 12.69GB RAM 搭載のマシンで行い、数値実験での ZDD の構築には Python 3 の Graphillion library (Inoue et al. 2016¹⁷⁾) を用いた。

Z_a^t, Z_b^t は、それぞれ (5)(a)(b) の方策による結果を表している。(Pre), (En) はそれぞれ Pre-trip, En-route リクエストであることを示しており、 Z_a^t (モノ-b), Z_a^t (モノ-c) は En-route リクエストがモノ-b の場合と、モノ-c の場合についての結果である。(Pre) は Pre-trip リクエスト, (En) は En-route リクエストであることをそれぞれ示している。

式 (2) を目的関数とする場合の求解性能は、JOIN Calculation Time と Optimum Paths Search Time の和によって求められる。この場合の Optimum Paths Search Time は、個人の経路コストにおいて各リンクでの滞在コストが過去の滞在履歴に依存するため、全てのパターンについて探索する必要がある。Num of Paths に比例する。 $Z_a^{t=7}$ の方法では、 $Z_b^{t=7}$ に比べて経路列挙数が 10% 程度におさえられているため、Optimum Paths Search Time も同様に抑えられている。また、JOIN Calculation Time も 30% 以下に抑えられていることが分かる。このとき、 $Z_a^{t=7}$ では En-route リクエストの処理は 60[sec] 以内で実行可能であることが分かった。

ここで、式 (3) で総走行距離の最小化を目的関数とする時、Optimum Paths Search は ZDD の SORT 演算によって非常に高速に求解可能であり、最も経路数が多い $Z_b^{t=7}$ (モノ-b) の場合であっても 0.4261[sec] と式 (2) の場合に比べて著しく短い時間で計算可能である。これは経路コストがリンクの距離に依存する単純和であることによる。

またこのとき、 $t = 7$ 以降においては、 $Z^{t>7}$ (モノ-c) $\neq \emptyset$ となり、モノ-c のマッチングを成功させるためには必ず 1 タイムステップ以内の高速求解が必要となることが分かった。

(3) First-Come-First-Serve

4.(2) 節では、成立するマッチングにヒト-a, ヒト-b, モノ-a の順序は関係しない場合を扱っているが、一般的なサービスでは先行するリクエストが優先されることが多い。本節ではヒト-a, ヒト-b, モノ-a, モノ-b による FCFS 方策での Pre-trip リクエストのマッチングを考える。プレイヤー集合 $N =$ ヒト-a, ヒト-b, モノ-a, モノ-b の各 User の順序を考慮する場合、形成される経路は異

表-4 User / Driver の需要・供給情報

	User				Driver
	ヒト-a	ヒト-b	モノ-a	モノ-b(/モノ-c)	
Origin	Node 1	Node 4	Node 2	Node 4(/6)	Node 5
Destination	Node 1	Node 4	Node 5	Node 5	Node 5
Active time-window	[0,14]	[3,14]	[0,14]	[6,14]	[0,14]
Mandatory activity	Node 5	Node 2	-	-	-
Mandatory arrival time	-	-	4	-	-
Minimum visit time	2	3	-	-	-
Link Travel Cost	0.3	0.3	0	0	1
Threshold time of successive staying time	3	3	-	-	-
Initial CoS at Node 2	-10	-10	0	0	1
Initial CoS at Node 5	-10	-10	-1	-1	1
CoS at other Nodes	0.5	0.5	0.5	1.5	1
Time-discount rate	0.98	0.98	1	1	1
Vehicle Capacity	-	-	-	-	4

表-5 En-route リクエストに対する実行可能経路集合族 Z^t の列挙数と計算性能

	Num of Paths	JOIN Calculation	Optimum Paths
		Time [sec]	Search Time[sec]
ヒト-a(Pre)	90,134	0.6541	
ヒト-b(Pre)	1,922	1.7601	
モノ-a(Pre)	476,496	0.1793	
モノ-b(En)	1,136	0.1001	
モノ-c(En)	1,136	0.0936	
X	1,597,054	1.0432	
$Z_a^{t=7}$ (モノ-b)	1,445	31.672 + 0.007 (0.007 for Look-up)	20.5798
$Z_a^{t=7}$ (モノ-c)	348	17.7785 + 0.006 (0.006 for Look-up)	4.5404
$Z_b^{t=7}$ (モノ-b)	14,815	103.7040	281.9732
$Z_b^{t=7}$ (モノ-c)	2,052	89.7169	26.5443

なる。3.11 に従い、これを提携としてその提携値を計算し、 N からなる提携構造を評価することを考える。

表-6 は、FCFS 方策によってプレイヤー集合のリクエスト順を考慮した際に成立する提携と、そのときの最適経路数・最小旅行距離を示している。異なるリクエスト順によって形成される ${}_4P_4 = 4! = 24$ 通りのパターン数のうち、提携が成立したリクエスト順が表-6 に示した 20 通りであった。提携への出現頻度をカウントすると、モノ-a、モノ-b が 16 回で同率で最も多かった。次いでヒト-b は 10 回、ヒト-a は 8 回で最も少なかった。

また、4 つのリクエスト全てに応じることができる提携は見られなかった。特に、ヒト-a、ヒト-b のいずれ

もが含まれている提携は無く、ヒト-a、ヒト-b は目的地ノードにおける最短訪問時間が設定されていることと、往復であるため車両の拘束時間が長いことが、両者のリクエストが同時に満たされる可能性を落とすことが示された。加えて、モノ-a、モノ-b のリクエストが最初に行われるという条件下では、全てのパターンについて 2User 以上からなる提携が成立することが示された。

限界貢献度は提携に参加する順序を変えたすべての順列について、式 (10) の値をとって平均したものであるから、表-6 において成立している全ての提携と、その提携が形成されるまでの全ての提携について考えればよいことになる。したがって、プレイヤー集合 N に

表-6 ヒト-a, ヒト-b, モノ-a, モノ-b から成立する提携と, その最適経路数・最小旅行距離

Order of Request			Min Travel	Optim
1	2	3	Distance	Path Num
ヒト-a	モノ-a		13	1
ヒト-a	モノ-a	モノ-b	16	14
ヒト-a	モノ-b		15	14
ヒト-a	モノ-b	モノ-a	16	2
ヒト-b	モノ-a		13	1
ヒト-b	モノ-a	モノ-b	14	1
ヒト-b	モノ-b		11	3
ヒト-b	モノ-b	モノ-a	14	1
モノ-a	ヒト-a		15	45
モノ-a	ヒト-a	モノ-b	18	6
モノ-a	ヒト-b		13	210
モノ-a	ヒト-b	モノ-b	14	1
モノ-a	モノ-b		6	60
モノ-a	モノ-b	ヒト-b	14	20
モノ-b	ヒト-a		15	210
モノ-b	ヒト-a	モノ-a	16	2
モノ-b	ヒト-b		11	30
モノ-b	ヒト-b	モノ-a	14	16
モノ-b	モノ-a		6	16
モノ-b	モノ-a	ヒト-b	14	20

ついて, 表-6 に示した提携とその提携が形成されるまでの全提携による実行可能経路集合族を提携候補として, Shapley value は

$$\phi(v) = (-4.5, -2.1, 2.6, 2.4) \quad (11)$$

となり, ヒト-a, ヒト-b について負の値をとっており, モノ-a, モノ-b については正の値をとる. つまり, 旅客による往復利用は車両の効率的な運行を悪化させるとともに, 貨物による片道の利用は車両をより効率的に使うことを可能にしていると解釈可能である. また, Shapley value の結果は, 表-6 において示される提携の形成傾向と整合的であることも分かった.

また, 負の Shapley value を持つ旅客の要求は, そのタイミングで受け入れ可能であっても拒否し, 代替輸送手段や他の共有車両を促進したほうが効率的になる可能性があることを示唆しており, これは CSG(Coalition Structure Generation) 問題における利用者の車両割当問題への自然な拡張と言える. あるいは, Shapley value は特定のネットワークに対する運賃構造の指標を示すこととなるため, 負の限界貢献度を有する利用者に対してより高い値を課すなど, オークション設計問題へと拡張することができるだろう.

5. 本研究の成果と今後の課題

本論文では, 貨客の移動 / 配送需要から発生するリクエストに基づいて運行する車両を想定し, 時間制約付きの On-line DARP として定式化を行った. 利用者の En-route リクエストに対して, 逐次挿入により最適経路を逐次計算する, 列挙索引を用いたアルゴリズムを提案し, その計算性能を評価した. また, 提案する On-line アルゴリズムを用い, 貨客の提携を評価する Shapley value に基づくフレームワークを提案し, 貨客混載車両への利用者配分の方策を, SO(System Optimum) 方策と FCFS(First-Come-First-Serve) 方策でのマッチングによって形成されるマッチング結果から議論した.

提案アルゴリズムの計算性能について, リクエストに基づき実行可能経路集合族を JOIN 演算によって逐次的に構築することで, 単純な総走行距離の最小化問題であれば, 10^{-1} のオーダー程度で可能であることが分かった. しかし, 総旅行コストの評価においては, 旅客の移動についての効用関数に経路ログに依存するパラメータである satiation parameter を考慮しているため, その評価は経路数分の繰り返し計算を要することが課題として明らかになった. SO 方策では, 先行する User と Driver からなる経路集合の組合せ数が非常に大きくなるため, 経路ログに応じて適応的にデータベースサイズを縮減することで, 求解性能を向上させ, 小さな系においてはリクエストに対して即時性のあるシステムとして実装可能であることが示唆された.

FCFS 方策では, リクエスト順が提携の成立と最適経路に与える影響について, 列挙索引を用いて特性関数によって与えられる提携値を計算し, Shapley value として User の組合せを評価する手法を提案した. この手法を用いて, 貨客提携ネットワークを評価し, 旅客による往復利用は車両の効率運行を妨げるとともに, 貨物は車両をより効率的に使うために好ましい性質をもつことが示された.

貨客混載車両以外の手段で復路が担保されるように, 各ノードの定時定路線の公共交通との接続性を考慮したネットワークを記述することが, 現実的なネットワークに適用する場合に必要なと考えられる. また, 最大マッチングを優先する SO 方策と, 先行するリクエストの効用を優先する FCFS 方策が, 利用者・利用形態に与える長期的な影響の比較も必要である. リクエストの成功・棄却によって利用選好が更新される day-to-day の影響を考慮し, 利用者の確率的なリクエストを扱う際の車両の運行を考えることもサービスとして利用されるために重要な論点である.

提携構造を Shapley value によって評価するフレームワークは, 複数車両問題において最適な利用者割当を

可能とするアルゴリズムや料金負担を考慮したオークション設計に対して応用可能であり、本研究の今後の展望とする。

nology Transfer, Vol.18, No.1, pp.57–66, 2016.

(2022. 3. 6 受付)

参考文献

- 1) Ghilas, V., Demir, E., and Van Woensel, T.: Integrating passenger and freight transportation: Model formulation and insights, *BETA publicatie: Working papers*, Vol.441, 2013.
- 2) Li, B., Krushinsky, D., Reijers, H. A., and Van Woensel, T.: The share-a-ride problem: People and parcels sharing taxis, *European Journal of Operational Research*, Vol.238, No.1, pp.31–40, 2014.
- 3) Jaw, J.-J., Odoni, A. R., Psaraftis, H. N., and Wilson, N. H.: A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows, *Transportation Research Part B: Methodological*, Vol.20, No.3, pp.243–257, 1986.
- 4) Tsubouchi, K., Hiekata, K., and Yamato, H.: Scheduling algorithm for on-demand bus system, Vol. 10, pp. 189 – 194, 05 2009.
- 5) Tsubouchi, K. and Yamato, H.: Adapted scheduling algorithm for on-demand bus system, *International Journal of Intelligent Transportation Systems Research*, Vol.10, pp.82–92, 5 2012.
- 6) Minato, S.: Zero-suppressed bdds for set manipulation in combinatorial problems, pp. 272–277, 1993.
- 7) Minato, S.: Super-fast graph enumeration algorithm - new approach for combinatorial problem (in Japanese), 2015.
- 8) Kurai, R., Minato, S.-i., and Zeugmann, T.: N-gram analysis based on zero-suppressed bdds, *Annual Conference of the Japanese Society for Artificial Intelligence*, pp. 289–300, Springer, 2006.
- 9) Loekito, E., Bailey, J., and Pei, J.: A binary decision diagram based approach for mining frequent subsequences, *Knowledge and Information Systems*, Vol.24, No.2, pp.235–268, 2010.
- 10) Hayakawa, K. and Hato, E.: Dynamic traffic resources allocation under elastic demand of users with space-time prism constraints, *arXiv preprint arXiv:1806.10719*, 2018.
- 11) Yeh, D. Y.: A dynamic programming approach to the complete set partitioning problem, *BIT*, Vol.26, No.4, pp.467–474, dec 1986.
- 12) Sandholm, T. W. and Lesser, V. R.: Coalitions among computationally bounded agents, *Artificial intelligence*, Vol.94, No.1-2, pp.99–137, 1997.
- 13) Jeong, S. and Shoham, Y.: Marginal contribution nets: A compact representation scheme for coalitional games, *Proceedings of the 6th ACM Conference on Electronic Commerce*, pp. 193–202, 2005.
- 14) Conitzer, V. and Sandholm, T.: Complexity of constructing solutions in the core based on synergies among coalitions, *Artificial Intelligence*, Vol.170, No.6-7, pp.607–619, 2006.
- 15) Sakurai, Y., Ueda, S., Iwasaki, A., Minato, S.-I., and Yokoo, M.: A compact representation scheme of coalitional games based on multi-terminal zero-suppressed binary decision diagrams, *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 4–18, Springer, 2011.
- 16) Bhat, C. R.: A multiple discrete–continuous extreme value model: formulation and application to discretionary time-use decisions, *Transportation Research Part B: Methodological*, Vol.39, No.8, pp.679–707, 2005.
- 17) Inoue, T., Iwashita, H., Kawahara, J., and Minato, S.: Graphillion: software library for very large sets of labeled graphs, *International Journal on Software Tools for Tech-*

SYSTEM OPTIMUM DYNAMIC ASSIGNMENT PROBLEM IN PASSENGER-AND-FREIGHT COALITION AND ITS EVALUATION BASED ON SHAPLEY VALUE

Fuga MAYUZUMI, Eiji HATO

In this study, an optimization problem for the operation of passenger-and-freight vehicles is described as a combinatorial optimization problem. A fast matching algorithm for dynamic vehicle operations was proposed with employing a route enumeration and indexing method: Zero-suppressed Binary Decision Diagram. As for matching policies, System Optimum (SO) policy and First-Come-First-Serve (FCFS) policy were defined based on whether preceding requests should be prioritized or not. SO policy that takes longer time for calculation than FCFS policy performed real-time processing on the order of tens of seconds in the numerical example. We also proposed the framework to calculate the value of coalition in characteristics function form considering the matching with FCFS policy using ZDD. The marginal contribution to the matching is evaluated by Shapley value, utilizing the methodological feature of ZDD that can enumerate and index feasible patterns. Numerical experiments on the scenario assuming heterogeneity in the trip patterns of passengers and freight were conducted, focusing on their round-trip and one-way trips especially, which lead to showing that round-trip requests deteriorate the matching efficiency. We also discussed the appropriate assignment method of users to vehicles in passenger-and-freight coalition network.