The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

異種車両連携を考慮する経路計画における整数計画法

Integer Programming for Cooperative Route Optimization

大滝啓介 *1 Keisuke Otaki　早川敬一郎 *1 Keiichiro Hayakawa　小出智士 *1 Satoshi Koide　大社綾乃 *1 Ayano Otaki　西 智樹 *1 Tomoki Nishi

*1豊田中央研究所
Toyota Central R&D Labs., Inc.

Cooperation among vehicles is essential for MaaS. In previous work, we developed a new cooperation concept, named delegation, using different types of vehicles. Further, we formalized a new route optimization problem enabling delegation. A fundamental approach to solve the problem is to define IP instances representing the optimization problem, and to solve the instances using a modern IP solver (e.g., Gurobi). However, the performance of solver-based methods highly depends on the instances and a subset of decision variables is not fully-used in optimization. In this paper, we develop a new IP instance of the optimization problem, evaluate them through numerical experiments, and discuss the instances and their features.

1. はじめに

交通や物流における社会問題は，都市化に伴って悪化することが予測されている *1．これらの社会問題に対処するため，既存の交通システムを有効活用する必要があり MaaS*2 が注目されている．MaaS では様々な意思決定の最適化が必要となり，例えばライドシェアでは，車両の待機場所や顧客への訪問経路，相乗り方法等を最適化することで，車両移動コストや顧客の待機時間を削減できると期待されている [Agatz 12, Ma 15, Furuhata 13]．同時に様々な機能を持った車両 (**異種車両**) が連携することで，新しいサービスが創発されることが期待されている [Hietanen 14]．我々は特に車両の経路計画に着目し，車両が取るべき経路の最適化について議論する．本稿では配送サービスを例として，サービス供給車両が出発地から目的地へ移動する際に利用する経路を最適化する問題を扱う．

図 1 にコンセプト図を示す．個別のモビリティが独立して配送サービスを行う場合，図 1(1) に示すように，最短経路によって移動コストを最小化できる．一方で図 1(2) に示すように**隊列** (platoon) を形成して移動経路を共有する場合，燃料消費が削減されることが知られている [Bonnet 00]．このような燃料消費削減効果を想定した経路計画は**車群経路計画** (vehicle platooning problem; VPP) と呼ばれ，整数計画法 (Integer programming; IP) を用いる解法や近似解法が研究されている [Larsson 15]．また似た目的関数を想定したグラフ上の最適化問題に対する手法を転用することもできる [Takise 16]．VPP では単一種の車両の経路を最適化するため，MaaS のように異種車両が存在する状況のモデルとして表現力が十分ではない．例えば図 1(3) に図示するように，ドローンがトラックに格納されて移動する配送などの異種車両連携を扱うためには，VPP よりも一般的なモデルが必要となる．

我々はこれまで，隊列を一般化させた**委譲** (delegation) という概念を定義し，委譲によって異種車両連携を考慮する最適化問題として，*Two-modal palette platooning problem* (2MP^3) と呼ばれる最適化問題のクラスについて議論してきた [Otaki 19]．この最適化問題は NP 困難であるため，大規模な問題を得ためには発見的解法が重要となる．しかし発見的解法自体を評価するために，厳密解が必要になる場面も多くある．そのため既存の研究では整数計画ソルバーを利用し，小規模な問題について厳密解を求め，発見的解法による解と比較することで，手法の性能を議論してきた．しかし既存の IP インスタンスは

(a) スケーラビリティが低く，小規模な問題であっても解けない場合がある

(b) 目的関数が限定的であり，決定変数に無駄がある

(c) 解に冗長な閉路を含むことがあり，最適値を求められない場合がある

という 3 つの問題があった．本稿では，2MP^3 に関する発見的解法を議論するための厳密解をより高速に，大規模な問題に対して与えることを目的として，VPP における既存研究 [Larsson 15] に基づいて，新たな IP インスタンスやその拡張の例を新しく提案し，数値実験を行って性質や性能を評価する．

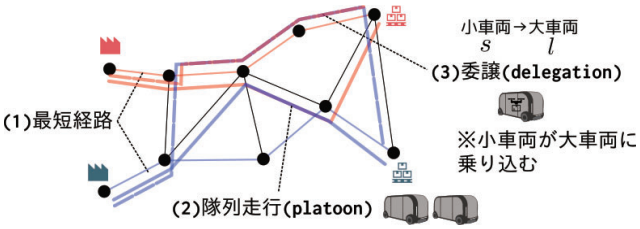


図 1: 経路計画のコンセプト比較: (1) は最短経路 (2) は隊列走行 (3) はコスト委譲による乗り込み型の連携走行

2. 準備

自然数 $n \in \mathbb{N}$ に対して $[n] = \{1, \cdots, n\}$ とし，記号 $\langle \cdot \rangle$ はリストを，演算子 $+$ でリストの連結演算を表す．

2.1 グラフと異種車両リクエスト

本稿では $G = (V, E, w)$ を単純な重み付き有向グラフとする．辺 $(u, v) \in E$ の重み $w(u, v)$ は単純に $w_{u,v}$ と記す．

連絡先: 大滝啓介，豊田中央研究所，愛知県長久手市横道 41-1，otaki@mosk.tytlabs.co.jp

*1　都市化について，例えば World Urbanization Prospects: The 2014 Revision Population Database が参考になる．

*2　Mobility as a Service

2 頂点 $s,t \in V$ 間の経路 $p_{s,t}$ を頂点の列として $p_{s,t} = \langle v_1, v_2, \ldots, v_{k+1}\rangle, v_1 = s, v_{k+1} = t, (v_i, v_{i+1}) \in E$ $(1 \leq i \leq k)$ と表し，i 番目の頂点 v_i を $p_{s,t}(i)$ で表す．また 2 頂点 s,t 間の全ての経路の集合を $\Pi(s,t)$ で表し，特に最短経路を $\pi(s,t) \in \Pi(s,t)$ です．なお最短経路 $\pi(s,t)$ とは，経路に沿った辺重みの総和で定義される距離 $d(s,t) = \sum_{1\leq i\leq k} w(v_i, v_{i+1})$ が最小となる経路である．

本稿ではトラックなどを想定した「大車両」と，小型の配送車両やドローンを想定した「小車両」の二種類の車両を扱う．これらの車両はいずれも移動サービスを提供し，図 1 で例示したように**小車両は大車両に乗り込む**ことができるといった，特殊な連携を可能とする．大車両と小車両は，それぞれ N_L 台と N_S 台存在し，識別子 $l \in [N_L]$ と $s \in [N_S]$ で表す．個別の車両は，移動サービスを表現する単一の**リクエスト** $r = (o,d) \in V \times V$ を持つ．ある経路 p が $p \in \Pi(o,d)$ であるとき，p は r を達成する充足経路と呼ぶ．本稿では，与えられた全ての車両に対する充足経路の集合を，問題の解と呼ぶ．以上の定義より連携を考慮しない場合には，最短経路を移動することで解のコストを最小化することができる．

2.2 単一車種を想定した経路計画 VPP

車両連携を考慮する場合の基本的な考え方は，最短経路を利用する場合と比較し，複数の車両が連携を行うことが移動コストに対して利益を生む（つまり**移動コストを削減する**）ような目的関数を用いて経路計画を行うことである．単一車両を想定する VPP においては，以下の式に応じて移動コストを低減するような目的関数が利用されている [Larsson 15]．解 $\mathcal{P}$ のコスト $c(\mathcal{P})$ を，パラメータ η $(0 < \eta < 1)$ を用いて

$$\begin{aligned} c(\mathcal{P}) &= \sum_{(u,v)\in E} w_{u,v} g_{u,v} \\ g_{u,v} &= (\mathcal{P} \text{ 中の車群の先頭車両台数}) \\ &\quad + \eta \times (\mathcal{P} \text{ 中の車群の後続車両台数}) \end{aligned} \tag{1}$$

と定義する．式 (1) によれば，辺 $(u,v) \in E$ を可能な限り車群を構成して通過することで，本来 $w_{u,v}$ が必要だった移動コストが $\eta w_{u,v}$ に削減される．

2.3 複数車種を想定した経路計画 2MP^3

最適化問題 2MP^3 は，図 1(3) で想定した用に，大車両が小車両に乗り込むことで，移動コストを 0 と見なす連携に対する経路計画である．式 (1) を参考にして定義された式 (2) を目的関数に用いている [Otaki 19].

$$\begin{aligned} c^{(\mathrm{del})}(\mathcal{P}) &= \sum_{(u,v)\in E} w_{u,v} g_{u,v}^{(\mathrm{del}),L} \\ g_{u,v}^{(\mathrm{del}),L} &= (\text{車種 } L \text{ の走行台数}) \\ &+ (\text{どの } L \text{ にも搭乗していない } S \text{ の台数}) \end{aligned} \tag{2}$$

以下に形式的な経路最適化問題を定義する．

問題 1 (2MP^3 [Otaki 19])．小車両と大車両の移動リクエストがそれぞれ $\mathcal{R}_S = \{r_1^{(S)}, \cdots, r_{N_S}^{(S)}\}$ と $\mathcal{R}_L = \{r_1^{(L)}, \cdots, r_{N_L}^{(L)}\}$ として所与とする．2MP^3 とは容量制約 $Q_l^{(LS)}$ の下で，式 (2) を最小化する解 $\mathcal{P}_T = \{P_i \mid i \in [N_T], P_i \in \Pi(o_i^{(T)}, d_i^{(T)})\}$ $(T \in \{S, L\})$ と，乗合いの組合せ $\mu = \{(s,l,u,v) \mid s \in [N_S], l \in [N_L], u, v \in V\}$ を求める問題である．集合 μ は車両間の連携を表す **(コスト委譲) 割当**と呼ぶ．もし $(s,l,u,v) \in \mu$ であるとき，(u,v) の移動の際に s は l に積載されていると解釈する．

3. 2MP^3 のための整数計画法

本章では既存の IP インスタンスを解説し，次にこれを改良した IP を提案・議論する．

3.1 既存の IP インスタンス `IP-NA` [Otaki 19]

既存の IP インスタンスとして以下の式が提案されている．本稿ではこの既存インスタンスを `IP-NA` と呼ぶ．インスタンス `IP-NA` では決定変数として，大/小車両が辺 $(u,v) \in E$ を通過するかどうかを表す変数 $f^{(L)}, f^{(S)}$ と，2MP^3 に定義される割当 μ を表す変数を最適化する．目的関数は移動コストの最小化である．以下の式では，添字は $s \in [N_S], l \in [N_L], T \in \{S, L\}$, $(u,v) \in E$ である:

$$\min_{f^{(L)}, f^{(S)}, \mu} \sum_{(u,v)\in E} w_{u,v} g_{u,v}^{(\mathrm{del}),L} \tag{3}$$

subject to

$$g_{u,v}^{(\mathrm{del}),L} = \sum_{l\in[N_L]} f_{u,v,l}^{(L)} + \sum_{s\in[N_S]} \phi_{u,v}^{s} \tag{4a}$$

$$\phi_{u,v}^{s} = \prod_{l\in[N_L]} (1 - \mu_{u,v}^{s,l}) \times f_{u,v,s}^{(S)} \tag{4b}$$

$$2\mu_{u,v}^{s,l} \leq f_{u,v,s}^{(S)} + f_{u,v,l}^{(L)} \tag{4c}$$

$$\sum_{l\in[N_L]} \mu_{u,v}^{s,l} \leq 1, \sum_{s\in[N_S]} \mu_{u,v}^{s,l} \leq Q_l^{(LS)} \tag{4d}$$

$$\sum_{v} f_{u,v,i}^{(T)} - \sum_{v} f_{v,u,i}^{(T)} = \begin{cases} 1 & \text{if } u = o_i^{(T)} \\ -1 & \text{if } u = d_i^{(T)} \\ 0 & \text{o/w} \end{cases} \tag{4e}$$

$$f_{u,v,l}^{(L)}, f_{u,v,s}^{(S)}, \mu_{u,v}^{s,l} \in \{0,1\} \tag{4f}$$

なお i は $T = L$ の場合には $[N_L]$ を，$T = S$ の場合には $[N_S]$ を動く車両の識別子である．制約 (4a) は式 (2) に対応する．制約 (4b) は単独で移動する小車両を規定する．制約 (4c) では相乗り $\mu_{u,v}^{s,l} = 1$ は l と s が共に (u,v) を移動している場合に限ることを意味する．制約 (4d) は乗り込む割当の排他性と容量制約を示す．制約 (4e) は解となる経路の条件を示す．制約 (4f) は決定変数の値域を制約している．

3.2 改良 IP インスタンス `IP-A`

前節の `IP-NA` は，車両同士の組合せに関する決定変数 μ を導入しているため表現力が高いが，一方で目的関数に μ が出現しないため冗長であり，結果として大規模な問題に対してパフォーマンスが低下する．そこで式 (2) を直接表現するような改良型の IP インスタンスとして `IP-A` を提案する．インスタンス `IP-A` では，`IP-NA` と同様に経路 $f^{(L)}, f^{(S)}$ を決定変数として持ち，新しい整数変数として $z_{u,v}$ を導入し辺 (u,v) を通過する際に乗り込む小車両の台数として最適化する．以下に追加される制約を示す．

$$0 \leq z_{u,v} \leq \min\left(\sum_{l\in[N_L]} Q_l^{(LS)} f_{u,v,l}^{(L)}, \sum_{s\in[N_S]} f_{u,v,s}^{(S)} \right)$$

以上の制約と決定変数により，換算量 $g_{u,v}^{(\mathrm{del}),L}$ を

$$g_{u,v}^{(\mathrm{del}),L} = \sum_{l\in[N_L]} f_{u,v,l}^{(L)} + \sum_{s\in[N_S]} f_{u,v,s}^{(S)} - z_{u,v} \tag{5}$$

として再定義する．式 (5) は，式 (2) に定義された車両の台数を直接計算するような IP になっている．

表 1: IP インスタンスの比較

name	種類	変数	特徴	変数/制約数
IP-NA	01-IP	経路 $f^{(L)}, f^{(S)}$ 組合せ μ	非匿名	多い
IP-A	IP	経路 $f^{(L)}, f^{(S)}$ 乗込台数 z	匿名	少ない

3.3 冗長な閉路の除去

車両が乗り込むことで移動コストが 0 と見なされる性質から，インスタンス IP-NA および IP-A は，いずれも冗長な閉路を含むことがあった．このような巡回路を避けるには，例えば MTZ 制約などを利用することができる [Miller 60].

3.4 IP の特徴比較

提案した手法の比較結果を表 1 に示す．

定義より，IP-NA における全ての決定変数は二値変数であるため，これは 01-IP である．一方で，IP-A においては，辺 $(u, v) \in E$ を移動する際に大車両に乗り込む小車両の台数 z を用いて容量制約を考慮しているため，一般の IP である．

次に IP インスタンスの決定変数から区分けされる特徴について述べる．インスタンス IP-NA では，割当 μ によって,「ある辺 $(u, v) \in E$ において，どの小車両 s が，どの大車両 l に乗車」に関する情報を全て得ることができる．我々はこのような状況を (連携が) **非匿名**であると呼んでいる．一方で IP-A では，式 2 に基づいて,「どの大車両にも乗車していない小車両の台数」を，変数 z を用いて直接最適化している．つまり，どの大車両が小車両を運搬しているかに関する情報を得ることができない．そのため我々はこのような状況を（連携が）**匿名**であると呼んでいる．VPP に対する既存の IP インスタンスは匿名なインスタンスとなっている [Larsson 15].

4. 数値実験による IP インスタンスの検証

提案した IP インスタンスを人工グラフ上のランダムなリクエスト群に対して適用し，評価と考察を行う．実験は Intel Core i7-3770K (3.50GHz) の CPU，32GB のメモリ，Gurobi 7.5.2, Python 3.5.2，Julia 0.6 を備えた PC を用いて実施した．

実験では $(N_L, Q_l^{(LS)})$ のラベルを利用して問題例を表現する．例えば $(N_L, Q_l^{(LS)}) = (2, 3)$ は，大車両が 2 台，それぞれの容量が 3 台分であり，小車両は 6 台 $(= 2 \times 3)$ 用意する問題を意味する．

4.1 IP インスタンス同士の計算時間比較

我々が提案した IP インスタンスを，人工データを用いて評価する．まず頂点位置にノイズを付与した 10×10 の格子グラフを作成し，グラフ頂点をランダムに選択してリクエストと見なして実験を行う．問題例として $(N_L, Q_l^{(LS)}) \in [2] \times [4]$ となる例をそれぞれ 30 問作成して，解を得るために必要だった計算時間の平均を求める．図 2 に結果を図示する．横軸は問題ラベル，縦軸は計算時間の対数である．

今回のランダムなリクエストに対しては，2 つの IP インスタンスを比較しても，計算時間に大きな差は見られなかった．一つの理由として，IP ソルバー Gurobi [Gurobi Optimization, LLC 18] が IP インスタンスを事前処理する際に，IP-NA において一見冗長に見える変数や制約を除去し，結果として似たような問題を解くことになっていると予測できる．

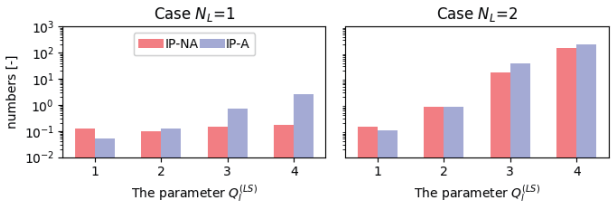
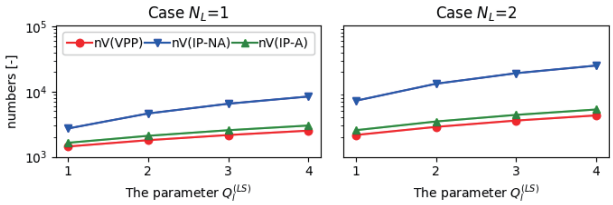
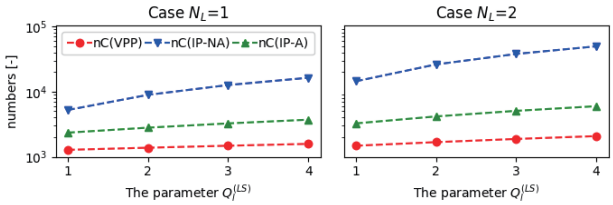


図 2: IP-NA と IP-A の計算時間比較



(a) 変数数 nV



(b) 制約数 nC

図 3: VPP，IP-NA，および IP-A 変数数と制約数の比較

4.2 IP インスタンス同士の変数・制約数比較

次に IP で問題例を表現した際の，IP インスタンス自体を比較し，表 1 で述べた変数/制約数の比較が正しいかどうかを確認する．インスタンスの比較を行うため，前節と同じく 10×10 のランダムな格子グラフを作成した．また $(N_L, Q_l^{(LS)}) \in [2] \times [4]$ の問題例を利用し，それぞれを IP インスタンスで表現した際の変数数と制約数を比較した．比較する際に，参考として元となった VPP の IP インスタンス [Larsson 15] を同時に計測した．図 3 に結果を示す．結果から分かる通り，図 3a に示す変数数や図 3b に示す制約の数は非匿名の IP インスタンス IP-NA は 10^1 個程度のオーダで多いことが分かる．一方で匿名の IP インスタンス IP-A や既存の VPP インスタンスはそれらの数が小さく，増加も緩やかである．

5. 拡張

IP インスタンス IP-NA 及び IP-A はいずれも最低限の目的関数となっているため，これを元にして様々な目的関数や所望の性質を加え拡張することができる．

5.1 グラフ G のトポロジが変化する場合

大車両がグラフ $G_L = (V_L, E_L)$ を，小車両がグラフ $G_S = (V_S, E_S)$ を移動する場合，IP を適用するためのグラフ $G = (V, E)$ を次のように作成する．まず頂点集合 $V = V_L \cup V_S$ と辺集合 $E = E_L \cup E_S$ を設定する．制約式 (4e) における経路 $f^{(L)}$ と $f^{(S)}$ については，それぞれの辺集合 E_L と E_S 上で定義する．制約式 (4a) において換算量 $g_{u,v}$ を定義する際には，辺集合 E を $E_{L \cap S} \cup E_{L \setminus S} \cup E_{S \setminus L}$ として共通の辺と片方にのみ移動可能な辺を集合として分離し，$(u, v) \in E_{L \cap S}$ であれば

式 (4a) を，そうでなければ $f^{(L)}$ もしくは $f^{(S)}$ によって単に経路を計上すれば良い．

5.2 辺や頂点の通過時間を明示的に扱う場合

本稿では問題を簡単にするため，時間情報は扱わず，静的な問題として異種車両連携経路計画を定義している．例えば既存モデルである VPP では，通過時間に相当する量を IP インスタンスに導入することで，時間を考慮して同じ辺を移動する経路計画を可能にしている [Larsson 15]．また時拡大ネットワークを用いて時間情報を静的グラフに表現することで，間接的に時間情報を扱うことができる．

5.3 組合せ μ を利用して目的関数を拡張する場合

既存の IP インスタンスである `IP-NA` は，車両連携を組合せ μ として明示的に扱うことができるため，スケーラビリティに劣る分，モデルとしての表現力が高い．つまりある辺 $(u, v) \in E$ にといて，$\sum_{s \in [N_S]} \mu^{s,l}_{u,v} \geq 1$ である車両 $l \in [N_L]$ は，車両連携の委譲先になっている大車両であると判定される．これを利用することで，例えば「少なくとも k 回以上は，どの大車両もドローンを載せる」といった負荷を均等にする制約などを記述することが出来る．

6. 結論

本稿では，異種車両の経路を最適化する問題に利用される整数計画法の性質とスケーラビリティについて実験を通じて議論した．また目的関数や制約を拡張することで，基本となる IP インスタンスを拡張していくことができることを議論した．今後は具体的な MaaS アプリケーションを想定した検証を行い，整数計画法の知見を利用した高度な発見的解法の設計を行う．

参考文献

[Agatz 12] Agatz, N., Erera, A., Savelsbergh, M., and Xign, W.: Opimization for dynamic ride-sharing: A review, *EJOR*, Vol. 223, pp. 295–303 (2012)

[Bonnet 00] Bonnet, C. and Fritz, H.: Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing, Technical report, SAE Technical Paper (2000)

[Furuhata 13] Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., and Koenig, S.: Ridesharing: The state-of-the-art and future directions, *TRP-B*, Vol. 57, pp. 28–46 (2013)

[Gurobi Optimization, LLC 18] Gurobi Optimization, LLC, : Gurobi Optimizer Reference Manual (2018)

[Hietanen 14] Hietanen, S.: Mobility as a service, pp. 2–4 (2014)

[Larsson 15] Larsson, E., Sennton, G., and Larson, J.: The vehicle platooning problem: Computational Complexity and Heuristics, *TRP-C*, Vol. 60, pp. 258–277 (2015)

[Ma 15] Ma, S., Zheng, Y., and Wolfson, O.: Real-Time City-Scale Taxi Ridesharing, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, No. 7, pp. 1782–1795 (2015)

[Miller 60] Miller, C. E., Tucker, A. W., and Zemlin, R. A.: Integer programming formulation of traveling salesman problems, *Journal of the ACM (JACM)*, Vol. 7, No. 4, pp. 326–329 (1960)

[Otaki 19] Otaki, K., Koide, S., Okoso, A., and Nishi, T.: Cooperative Routing with Heterogeneous Vehicles, in *to appear in AAMAS2019 (Main Track Extended Abstract)* (2019)

[Takise 16] Takise, K., Asano, Y., and Yoshikawa, M.: Multi-user routing to single destination with confluence, in *Proc. of the 24th ACM SIGSPATIAL*, pp. 72:1–72:4 (2016)