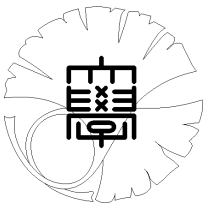
数理科学実践研究レター 2020–4 July 15, 2020

レイアウト変更を伴う配車サービスのモデル化

by

竹内 大智



UNIVERSITY OF TOKYO

GRADUATE SCHOOL OF MATHEMATICAL SCIENCES

KOMABA, TOKYO, JAPAN

レイアウト変更を伴う配車サービスのモデル化

竹内大智[1]（東京大学大学院数理科学研究科）
Daichi Takeuchi (Graduate School of Mathematical Sciences, The University of Tokyo)

概 要

ドライバーのない配車サービスを運用するに当たり、多様な利用客とそのニーズに適した車両とを対応させるアルゴリズムを開発することは重要である。本稿では、単一の車両をレイアウト変更しながら運用する場合について考察する。

1 はじめに

自動操縦車（robot vehicle、以下 RV と略す）を用いた配送サービスを展開する上で、どの利用客にどの車両を対応させるかを決める効率的なアルゴリズムを構築することは必要不可欠である。利用客の種類や利用開始場所は時間とともに変化するため、アルゴリズムも時間変化に対応できるものでなくてはならない。[1] では、そのようなアルゴリズムが考案されており、効率についても計算されている。本稿では彼らの結果を改良し、より我々の目的に即したアルゴリズムを構築することを目標とする。具体的には、変更点は次の通りである。

1) [1] では 1 種類の車両のみを扱っているが、複数の種類の車両を運用することで多様な利用客のニーズ（通勤客、観光客、宅配等）に対応できるようにする。

2) 適宜車両のレイアウト（種類）を変更をして、時間変化に伴い増減する利用客のニーズに対応できるようにする。

3) [1] では単独客のみを扱っているが、団体客も扱えるようにする。

2 アルゴリズムの構成（cf. [1]）

基本的には、アルゴリズムの構築は [1] で示された方法に従う。従って、変更点について特に詳しく説明する。
函数 $\tau\colon \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ を固定する。2 地点 $x, y \in \mathbb{R}^2$ に対して、$\tau(x, y)$ は x から y へ移動するのに必要な時間を表している。実用上は τ は三角不等式 $\tau(x, z) \leq \tau(x, y) + \tau(y, z)$ を満たすであろうが、本稿ではその仮定は不要である。
まず利用客（request）と、車両（vehicle）を次のように定義する。

定義 1 (cf. [1])

1) 有限集合 X を固定する。X の元 x を *type* と呼ぶ。各 type $x \in X$ に対して、正の整数 m_x を指定しておく。m_x は type の利用客を乗せる車両の最大収容可能人数を表す。

2) *request* とは *6* つ組 $(o_r, d_r, t_r, t_r^d, n_r, x_r)$ のことである。ここで、$o_r \in \mathbb{R}^2$ は request の待ち始める場所を表し、$d_r \in \mathbb{R}^2$ は目的地、t_r は待ち始める時間、t_r^d は希望の到着時間を表している。また、n_r は正の整数で、request の人数を表す。$x_r \in X$ であり、これを request r の *type* と呼ぶ。request r の人数 n_r は $n_r \leq m_{x_r}$ を満たすとする。

3) *vehicle* とは *3* つ組 $v := (q_v, P_v, x_v)$. ここで $q_v \in \mathbb{R}^2$ は vehicle v の現在地、P_v は v が乗せている request の有限集合である。$x_v \in X$ であり、これを vehicle v の *type* と呼ぶ。P_v の元の数は m_{x_v} 以下とする。

[1] daichi@ms.u-tokyo.ac.jp

上の定義について補足する。type とは利用客の種類（通勤客、観光客、宅配等）を表したものである。request の中に type と人数の情報を入れたことが主だった [1] から変更した点である。
今、request からなる有限集合 R と vehicle からなる有限集合 V が与えられたとする。V に含まれる vehicle v がどの request を運ぶのかを決定し、且つそのような対応の中でコストが最小のものを見つけたい。コスト関数 $\mathcal{C}$ としては、例えば次のような物が考えられる。

$$\mathcal{C} := \Sigma_v \Sigma_{r \in P_v} \delta_r + \Sigma_r \delta_r + \Sigma_{r'} c \tag{1}$$

記号について説明する。request r に対して、δ_r とは r を運ぶ際に実際に起こった遅延のことである。希望到着時刻が t_r^d なので、これは $\delta_r = t - t_r^d$ となる。また、r' は運ばれない request を走り，c はペナルティを表す、予め決めておく（大きな）定数である。
上では同時に request を運ぶ場合のみを考えており、運ばれない request r' が出てくるのはそのためである。同時ではなく順番に request を乗せ、降ろしていくことにすれば全ての request を運ぶことができるが、それでは以下で説明するアルゴリズムが適用できず、計算量が膨大になる可能性がある。

2.1 RV-グラフ

[1] では、アルゴリズムの構成のためにグラフ（RV-グラフ、RTV-グラフ）を構成している。本稿でも彼らの方法が適用できる。
request、及び vehicle の有限集合 R、V を固定する。RV-グラフとは R と V の元を頂点とし、$r \in R$ と、r を運ぶことのできる $v \in V$ とを結び、これを辺としたものである。[1] では、更に同時に運ぶことのできる request r_1, r_2 も結んでいるが、これは次節で行うのが適当と思われる。
request r と vehicle v が辺で結ばれる時、特にそれらの type は同じでなければならない。従って、最初に request と vehicle を type 毎に分けておいて、type 毎に構成しても同じことである。

2.2 RTV-グラフ

trip T とは、ある vehicle v で同時に運ぶことのできる R の部分集合のことである。trip は RV-グラフを元に、次のように帰納的に構成できる。
まず、元の数が一つの trip とは、ある v と結ばれている request r のことである。これは RV-グラフから構成できる。元の数が n の trip T' が構成されたとして、そこに別の request r を付け加えられるかを計算する。付け加えられれば元の数が $n+1$ の trip が得られる。全ての trip はこのようにして得ることができる。
trip が構成された後、RTV-グラフは次のように構成される。RTV-グラフとは、request r と trip T と vehicle v を頂点に持つ。$r \in T$ である時 r と T を辺 $e(r, T)$ で結び、T を vehicle v で運べる時 T と v を辺 $e(T, v)$ で結ぶ。これらを辺とするのが RTV-グラフである。RV-グラフの時と同様、これは、request 及び vehicle を type 毎に分けてから構成しても同じである。

2.3 レイアウト変更の導入

レイアウト変更も加味するときは次のようにすればよい。例えば、type x_1 の車両を別の type x_2 に変更する時、次のような request $r_{x_1 \to x_2}$ を導入する。この request は目的地がレイアウト変更の工場で、人数 n_r は x_1 を運ぶ車両の最大収容可能人数とする。このようにすることで、上の構成を全く形式的に適用することができる。n_r を最大としているのは、通常の request を乗せた状態でレイアウト変更の工場に向かわないためである。

3　最適化及び効率について

最適化についても [1] の方法が適用可能である。
まずは $e(T, v)$ を T のサイズが大きく、運ぶコストが小さい順に並べ、その順番で v と T を対応させることにする。これは [1] では greedy assignment と呼ばれている。この対応を整数線形計画法により最適化していく。
各辺 $e(T_i, v_j)$ に対し変数 $\epsilon_{i,j} \in \{0, 1\}$ を用意する。各 request r_k に対し変数 $\chi_k \in \{0, 1\}$ を用意する。これらは $\epsilon_{i,j} = 1$ が T_i を v_j で運ぶことを意味し、$\chi_k = 1$ が r_k はどの v でも運ばれないことを意味する変数である。$\epsilon_{i,j}$ と χ_k の choice $\Phi = \{\epsilon_{i,j}, \chi_k\}$ に対し、そのコスト $\mathcal{C}(\Phi)$ を例えば (1) で定義する。これら変数は次の制約条件を持つ。

1) 全ての j に対して $\Sigma_i \epsilon_{i,j} \leq 1$。

2) 全ての k に対して $\Sigma \epsilon_{i,j} + \chi_k = 1$。ここで左辺の Σ は辺 $e(T_i, v_j)$ が存在し、$r_k \in T_i$ となっているような (i, j) を渡っている。

$\epsilon_{i,j}, \chi_k$ がこれらの条件を満たし、且つコスト $\mathcal{C}(\Phi)$ を最小にするものを探索する。
RTV-グラフが構成された後、上の方法の計算量はおよそ次のようになる。まず type の数が 1 の場合は $n^{m\nu}$ のオーダーとなる。ここで ν は vehicle の数、n は request の数、m は vehicle の最大収容可能人数である。これは、各 vehicle v に対し、v で運ぶ request の選択は $\frac{n!}{m!(n-m)!}$ 通りであることから従う。type が複数あるときは、type 毎の計算量を全て掛け合わせたものとなる。

4　終わりに

本稿では、複数の種類の車両を、レイアウト変更しながら運用するにはどうすべきかについて考察し、基本的には単一の車両を、レイアウト変更せずに運用する場合と同じようにできることが分かった。しかしながら、コスト函数 (1) をどのように決めるか、或いはレイアウト変更用の request（2.3）をどこに何人配置すべきかまでは考察できておらず、そのためには実際にプログラムを組んで実験するべきである。

5　謝辞

本課題を提供してくださった日産自動車株式会社の高松敦さま、原加代子さまに感謝申し上げます。また、セミナーの時間調節や議論の総括をしていただいた間瀬崇史さま、多くのアドバイスをいただいた金井雅彦さま、鮑園園さま、共に課題に取り組んだ FMSP コース生の柴田翔さま、長谷川隆祥さまに感謝いたします。本研究は FMSP リーディング大学院の助成を受けております。

参考文献

[1] Alonso-Mora, J., Samaranayake, S., Waller, A., Frazzoli, E., Rus, D.: *On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment*, PNAS January 17, 2017 114 (3) 462-467.