

時間制約付き車両配送問題における顧客満足度に基づいた CVTPR-Tree 手法

CVTPR-Tree Approach for Customer Satisfaction on Temporal Vehicle Routing Problem with Time Windows

向 直人^{*} 馮 鈞^{*}
渡邊 豊英^{*}

Naoto MUKAI Jun FENG
Toyohide WATANABE

公共交通手段としてのデマンドバスが注目されている。将来の大規模な運営のためには、シミュレーションによる利便性と採算性の評価が必要である。我々が対象とする問題は Temporal Vehicle Routing Problem with Time Windows と呼ばれ、ある地域内における顧客の乗降時間の制約を含む配送依頼に対し、車両（バス）を的確に配備し、顧客配送を効果的に解決することを目的としている。本論文で提案する CVTPR-Tree は、車両を木構造で索引付けすることにより、効率的に短時間で顧客の配車を可能にする。顧客の配送順序は車両の走行距離または顧客の時間制約に基づき決定する。

Demand-bus system is focused as a new transportation system. The profitability and usability of the system should be analyzed for adopting a wide community. In Temporal Vehicle Routing Problem with Time Windows (TVRPTW) we address, demands with time constraints of customers arise enduringly. Share-ride vehicles transport customers to their destinations. We propose CVTPR-Tree that indexes vehicles in consideration of time constraints. The CVTPR-Tree assigns customers to a vehicle effectively in a short time. Delivery orders for customers depend on riding distance of vehicles or time constraints of customers.

1. はじめに

利便性の高い交通システムとしてデマンドバスが注目されている。デマンドバスとは乗客の要求に応じて乗降車場所を自由に変更できる乗合いバスのことである。通常の固定路線バスに対し、少ない台数で運営可能なデマンドバスを今後の公共交通手段として検討する自治体が増えてきている。我々の対象とする Temporal Vehicle Routing Problem with Time Windows (TVRPTW) はデマンドバス問題をシミュレーション・レベルに単純化したものである。ある地域内にお

る顧客の乗降車時間の制約を含む配送依頼に対し、配備されている車両（バス）による相乗り方式のサービスを行う。この時間制約とは乗車時間制限を意味し、顧客によってその値は異なる。TVRPTW において、解決すべき問題は顧客の車両割当て、顧客の配送順序の決定である。提案する CVTPR-Tree はリアルタイムに収集された移動オブジェクトに関する位置データを用いて移動オブジェクト（車両）のデータベースを構成する。時々刻々と変化する移動オブジェクトの位置管理において、位置情報の更新負荷が問題となる。そこで、車両の位置・速度と顧客の乗降位置に基づき時間をパラメータとした包含矩形(Bounding Rectangle)によってオブジェクトの将来の位置を近似表現する。顧客は索引に従って効率的に短時間で車両に割り当られる。また、車両割当て後の顧客の配送順序は車両の走行距離または顧客の時間制約に基づき決定される。

2 章では TVRPTW の形式化を述べる。3 章では CVTPR-Tree を定義し、4 章では CVTPR-Tree に基づく顧客の割当て、5 章では配送順序のアルゴリズムを示す。6 章で実験・考察し、7 章でまとめる。

2. TVRPTW

複数台の車両で地域内の全ての要求発生地点を回るためにどのようなルートを走らせるかという問題は Vehicle Routing Problem (VRP) と呼ばれる。また、要求に時間制約が附加された問題は VRP with Time Windows (VRPTW)[1][2] と呼ばれる。我々の対象とする Temporal VRPTW (TVRPTW) では、要求が初期段階で与えられるのではなく、時間経過と共に継続的に発生する。これらの問題は NP 困難な問題であり、最適解を求めることは非常に困難である。従って、精度のよい近似解を求めることが現実的な解法となる。TVRPTW を以下のように形式化する。

N 人の顧客を $C = \{c_1, c_2, \dots, c_N\}$ とする。顧客 c_j の乗降車要求を $DEMAND_j = (r_j, d_j, T_j)$ と表す。 r_j と d_j はそれぞれ乗降車位置であり、 T_j は顧客の乗車制限時間である。顧客満足度 CS_j を式(1)のように定義する。 tr_j と td_j はそれぞれ乗車時刻と降車時刻である。 σ は顧客満足度の曲線の傾きを決定するパラメータであり、値が小さいほど、顧客は満足を得やすいことになる。顧客満足度は図 1 に示すように、制限時間以内に顧客の配送を完了できれば最大値 1 となり、乗車時間が増加するにつれ値は 0 に近づく。

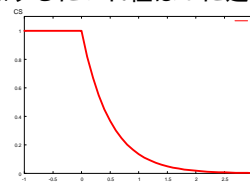


図 1 顧客満足度の値域
Fig.1 Range of Customer Satisfaction

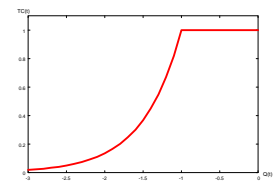


図 2 時間制約の値域
Fig.2 Range of Time Constraint

$$P_j = ((td_j - tr_j) - T_j) / T_j$$

$$CS_j = \begin{cases} 1 & P_j \leq 0 \\ \exp\left(\frac{-P_j}{\sigma}\right) & P_j > 0 \end{cases} \quad (1)$$

^{*} 学生会員 名古屋大学大学院情報科学研究科博士後期課程 naoto@watanabe.nuie.nagoya-u.ac.jp

^{*} 学生会員 名古屋大学大学院工学研究科博士後期課程 feng@watanabe.nuie.nagoya-u.ac.jp

^{*} 正会員 名古屋大学大学院情報科学研究科 watanabe@is.nagoya-u.ac.jp

K 台の車両を $B = \{b_1, b_2, \dots, b_k\}$ とする。車両 b_i の時刻 t の位置を $\bar{b}_i(t) = (x_i(t), y_i(t))$, 速度ベクトルを $\vec{b}_i(t) = (vx_i(t), vy_i(t))$ と表す。割り当てられた顧客の配送順序は図 3 に示すような配列 $q_i(t)$ で与えられる。配列には顧客の乗車・降車位置が格納され、配列の先頭から要求が処理される。配列は時刻 t をパラメータに持ち時間経過と共にその内容が変化する。

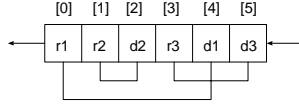


図 3 顧客の配送順序配列

Fig.3 Array of Delivery Order

位置 p_1, p_2 間の経路距離を $d(p_1, p_2)$ とする。長さ L の配列を持つ車両 b_i の時刻 t の走行コスト $|q_i(t)|$ を式(2)で定義する。

$$|q_i(t)| = \sum_{l=0}^{L-1} d(q_i(t)[l], q_i(t)[l+1]) \quad (2)$$

TVRPTW に対する研究視点は、車両の走行コスト（採算性）を抑えながら顧客満足度（採算性）を最大化することにある。目的関数は式(3)によって与えられる。

$$\max \left(\frac{\sum_{j=1}^N CS_j}{\sum_{i=1}^K |q_i(t)|} \right) \quad (3)$$

3. CVTPR-Tree

移動オブジェクトの位置・速度を管理するための手法として静的オブジェクトの索引手法である R*-Tree[3] を拡張した TPR-Tree[4,5] が提案されている。TPR-Tree は時間 t をパラメータとした包含矩形(Bounding Rectangle)を形成することにより移動オブジェクトを索引付ける。我々の提案する Constraint Vehicle TPR-Tree(CVTPR-Tree)は TPR-Tree の拡張版であり、顧客の乗降車位置と時間制約が考慮される。以下に CVTPR-Tree の構造を述べる。

3.1 葉ノード

葉ノードは車両 b_i へのポインタ、包含矩形 $R(t)$ 、時間制約 $TC(t)$ から構成される。

$$(b_i, R(t), TC(t))$$

包含矩形 $R(t)$ を x 座標・ y 座標の間隔として定義する。

$$R(t) = ((R_x^{\rightarrow}(t), R_x^{\leftarrow}(t)), (R_y^{\rightarrow}(t), R_y^{\leftarrow}(t)))$$

以下で包含矩形と時間制約を具体的に定義する。また、 x 座標についてのみ考え、 y 座標については省略する。

3.1.1 包含矩形

まず、顧客乗降車位置を頂点に持つ四角形を要求矩形(Demand Rectangle)と定義する。顧客 c_j の要求矩形 DR_j を式(4)で定義する。

$$DR_j = ((DR_{jx}^{\rightarrow}, DR_{jx}^{\leftarrow}), (DR_{jy}^{\rightarrow}, DR_{jy}^{\leftarrow}))$$

$$DR_{jx}^{\rightarrow} = \min(r_{jx}, d_{jx}) \quad (4)$$

$$DR_{jx}^{\leftarrow} = \max(r_{jx}, d_{jx})$$

車両 b_i の要求矩形 DR_i を割り当てられた全顧客 $c_j (j = 0, \dots, k)$ の要求矩形 DR_j を包含するように定義する(式(5))。

$$DR_i = ((DR_{ix}^{\rightarrow}, DR_{ix}^{\leftarrow}), (DR_{iy}^{\rightarrow}, DR_{iy}^{\leftarrow}))$$

$$DR_{ix}^{\rightarrow} = \min_{j=0}^k (DR_{jx}^{\rightarrow}) \quad (5)$$

$$DR_{ix}^{\leftarrow} = \max_{j=0}^k (DR_{jx}^{\leftarrow})$$

CVTPR-Tree は更新間隔 I で周期的に更新される。更新時刻 t_{upd} の包含矩形は車両 b_i の位置と車両の要求矩形によって決まる(式(6))。

$$R_x^{\rightarrow}(t_{upd}) = \min(x_i(t_{upd}), DR_{ix}^{\rightarrow})$$

$$R_x^{\leftarrow}(t_{upd}) = \max(x_i(t_{upd}), DR_{ix}^{\leftarrow}) \quad (6)$$

包含矩形の両端の速度を車両 b_i の速度とする(式(7))。

$$R_{vx}^{\rightarrow} = vx_i(t_{upd})$$

$$R_{vx}^{\leftarrow} = vx_i(t_{upd}) \quad (7)$$

時刻 $t > t_{upd}$ の包含矩形を式(8)のように定義する。

$$R_x^{\rightarrow}(t) = R_x^{\rightarrow}(t_{upd}) + R_{vx}^{\rightarrow} \times (t - t_{upd})$$

$$R_x^{\leftarrow}(t) = R_x^{\leftarrow}(t_{upd}) + R_{vx}^{\leftarrow} \times (t - t_{upd}) \quad (8)$$

3.1.2 時間制約

顧客 c_j の時刻 t の時間制約 $TC_j(t)$ を式(9)によって定義する。

ζ は制約値の最小値である。図 2 は時間制約値($\zeta = 0$)の時間経過に伴う変化を示している。車両が制限時間内に顧客の配送完了が十分に可能ならば値は小さいが、時間の経過と共に増加し、制限時間内に配送不可能になると、最大値 1 となる。

$$\begin{cases} Q_j(t) = \frac{-T_j}{(d(\bar{b}_i, r_j) + d(r_j, d_j)) / \bar{b}_i} & (\text{乗車前}) \\ Q_j(t) = \frac{(t - tr_j) - T_j}{d(\bar{b}_i, d_j) / \bar{b}_i} & (\text{乗車後}) \end{cases} \quad (9)$$

$$TC_j(t) = \begin{cases} \exp\left(\frac{Q_j(t)+1}{\sigma}\right) + \zeta & Q_j(t) \leq -1 \\ 1 + \zeta & Q_j(t) > -1 \end{cases}$$

車両 b_i の時間制約 $TC(t)$ を割り当てられた全顧客 $c_j (j = 0, \dots, k)$ の時間制約値の最大値とする(式(10))。その理由は、現実問題において、時間制約の高い乗車客が一人でもいれば、車両はその顧客を優先して送り届けなくてはならないからである。

$$TC(t) = \max_{j=0}^k (TC_j(t)) \quad (10)$$

3.2 中間ノード

中間ノードは最大 M の子ノードへのポインタを持つ。従って、木の高さは最も低くて $\log_M(K)$ となる。

3.2.1 包含矩形

子ノード集合の包含矩形を $R_i(t), (i = 0, \dots, m < M)$ とする。中間ノードの時刻 t_{upd} の包含矩形は全ての子ノードの時刻 t_{upd} の包含矩形をさらに包含する矩形として定められる(式(11))。

$$R_x^{\rightarrow}(t_{upd}) = \min_{i=0}^m (R_{ix}^{\rightarrow}(t_{upd})) \quad (11)$$

$$R_x^{\leftarrow}(t_{upd}) = \max_{i=0}^m (R_{ix}^{\leftarrow}(t_{upd}))$$

包含矩形の両端の速度を子ノード集合の最大・最小の速度と定める(式(12))。

$$R_{vx}^{\rightarrow} = \min_{i=0}^m (R_{ivx}^{\rightarrow}) \quad (12)$$

$$R_{vx}^{\leftarrow} = \max_{i=0}^m (R_{ivx}^{\leftarrow})$$

時刻 t の包含矩形は葉ノードと同じ式(8)で定義される。

3.2.2 時間制約

子ノード集合の時間制約を $TC_i(t), (i = 0, \dots, m < M)$ とする。時間制約 $TC(t)$ を式(13)のように子ノード集合の時間制約の平均値として定義する。その理由は、次節で提案する車両割当アルゴリズムにおいて、車両負荷の偏りを表現する必要があるからである。

$$TC(t) = avg_{i=0}^m (TC_i(t)) \quad (13)$$

3.3 例

要求 $DR1 \cdot DR2$ を担う車両 b_1 と要求 $DR3 \cdot DR4$ を担う車両 b_2 の包含矩形を図4に示す。これらの車両は図5に示す木構造で索引付けされる。

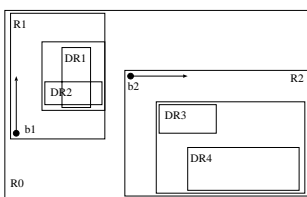


図4 包含矩形

Fig.4 Bounding Rectangles

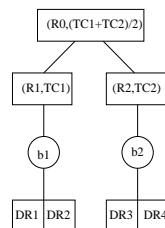


図5 木構造

Fig.5 Tree Structure

4. 車両割当

CVTPR-Tree への問合せ処理により車両割当を行う。包含矩形 R の面積を $A(R)$ と表す。顧客の要求 DR_j が発生したとすると、木の根から式(14)または式(15)を評価しながら、評価値の小さいノードを降下していき、最終的に到達した葉ノードの車両に要求を割り当てる。最短で $M \cdot \log_M(K)$ のオーダーで車両割当が可能である。式(14)は配送領域の小さな車両を優先する。この手法を VTPR と呼ぶ。

$$A(R(t) \cup DR_j) \quad (14)$$

式(15)は、時間制約が小さくかつ配送領域の小さい車両を優先する。この手法を CVTPR と呼ぶ。CVTPR は時間制約の最小値 ζ が小さければ時間制約を優先し、逆に大きければ配送領域を優先する。

$$TC(t) \times A(R(t) \cup DR_j) \quad (15)$$

5. 顧客配送順序

顧客割当の決定後、顧客の配送順序を決定する。つまり、配送順序を表す配列 $q(t)$ に顧客の乗車・降車位置を挿入する。以下に2通りの挿入方法を提案する。

5.1 距離優先

走行コスト $|q(t)|$ が最小になるように顧客配送の順序を決定する。図6のような顧客要求を考える。図中の $(r1, 1.0)$ は乗車位置 $r1$ と時間制約値 1.0 を意味する。配送順序配列 $q(t)$ は図7のようになる。走行コストを優先することにより、顧客の乗車時間が増加してしまい、満足度の低下を引き起こす可能性がある。

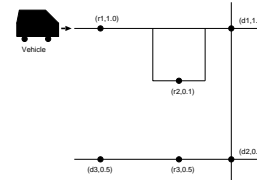


図6 顧客要求

Fig.6 Demands of Customers

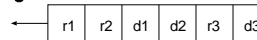


図7 距離優先の配送順序

Fig.7 Traveling Distance Measure

5.2 満足度優先

顧客の満足度が最大になるように顧客配送の順序を決定する。つまり、顧客の時間制約値が高い順に配送する。配送順序配列 $q(t)$ は図8のようになる。時間制約を優先することにより、顧客の満足度は高くなるが、車両の走行コストも高くなることが考えられる。

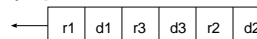


図8 満足度優先の配送順序

Fig.8 Customer Satisfaction Measure

6. 実験・考察

提案手法(表1)を評価実験した。評価項目は、平均顧客満足度(図9)、平均顧客乗車時間(図10)、平均走行コスト(図11)、配送完了顧客数(図12)、平均顧客待ち時間(図13)の5つであり、図中の横軸は車両数を表している。道路ネットワークとして 21×21 のグリッド(キャンパスサイズは 1000×1000 ピクセル)を用いた。終了時刻は $t=10000$ とし、各時刻に10%の確率で新たな顧客の要求を発生させる。顧客の乗降車位置と車両の初期位置は道路ネットワークからランダムに選択される。顧客の乗車時間制限 T_j は最短到着時間の1~5倍の間の値でランダムに決定される。また、車両速度 $|\vec{b}|=5$, 更新間隔 $I=10$, 最大ノード数 $M=5$, 時間制約最小値 $\zeta=0.1$, 顧客満足度パラメータ $\sigma=0.5$ を実験で用いた。

平均顧客満足度に関して、CS-CVTPR は車両数に依存す

ることなく全体的に高い満足度を得ることができた。その理由は満足度優先の配送順序を用いており、一人ずつ着実に配送を完了していくからである。一方、CVTPRとVTPRにおいては車両数の影響が大きく、車両数に比例して満足度が増加した。また、CVTPRの方がVTPRよりも良い値を示した。時間制約を車両割当の判断の一つに組み込んだことにより、車両への負荷が分散されたからであると考えられる。また、顧客満足度は平均顧客乗車時間に直結しており、同様の傾向を示した。配送完了顧客数、平均車両走行距離、平均顧客待ち時間に関しては、VTPRとCVTPRは同じような曲線を描き、共に良い値を示した。一方、CS-CVTPRはいずれの項目においても下回る結果となった。その理由は、配送順序において満足度を優先することにより、一人一人の乗車時間は減らせるものの、走行距離が長くなってしまい、配送完了顧客数や顧客待ち時間が増大してしまうからである。以上の結果から、顧客満足度(利便性)を高めることを重視するならばCS-CVTPRが良いが、さらに車両コスト(採算性)も考えるならばCVTPRが相応しい。

表1 比較手法

Table 1 Comparative Patterns

表記	割当	順序
VTPR	VTPR	距離優先
CVTPR	CVTPR	距離優先
CS-CVTPR	CVTPR	満足度優先

7. まとめと今後の課題

本稿ではデマンドバス・システムをシミュレーションし、その利便性・採算性を評価した。提案したCVTPR-Treeは動的に車両の索引を形成し、短時間かつ効率的な車両割当を可能とした。また、走行距離と顧客満足度を基準とした顧客配送順序を提案した。最後に評価実験を行い、提案手法が有効であることを示した。最適化アルゴリズムを用いた顧客交換によるパフォーマンスの向上が今後の課題である。

【謝辞】

本研究の一部は、名古屋大学情報系 COE：「社会情報基盤のための音声・映像の知的統合」の支援による。ここに記して感謝の意を表す。また、日頃からご指導頂いた愛知工業大学・石井直宏教授に感謝します。

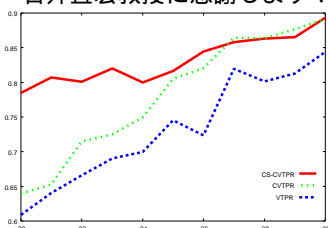


図9 平均顧客満足度

Fig.9 Customer Satisfaction

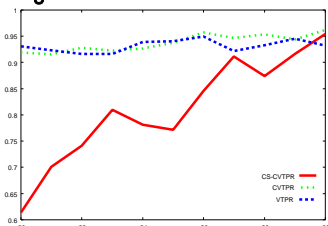


図12 配送完了顧客数

Fig.12 Transportable Customers

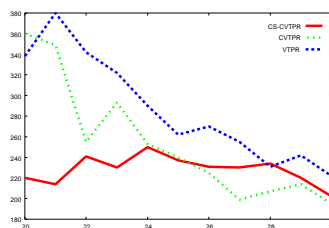


図10 平均顧客乗車時間

Fig.10 Average of Riding Time

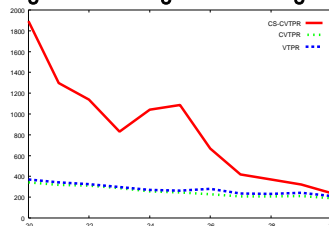


図13 平均顧客待ち時間

Fig.13 Average of Waiting Time

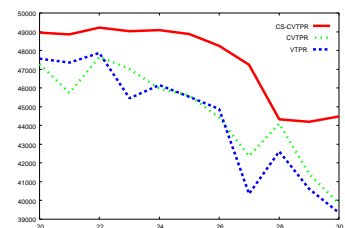


図11 平均車両走行コスト

Fig.11 Average of Traveling Cost

【文献】

[1] M. Desrochers, J.K. Lenstra, M.W.P. Savelsbergh, and F. Soumis: "Vehicle routing with time windows: Optimization and approximation", Methods and Studies, pp.65-84(1988).
 [2] M.M. Solomon and J. Desrosiers: "Time window constrained routing and scheduling problems", Transportations Science, pp.1-13(1988).
 [3] N. Bechmann, H. P. Kriegel, R. Schneider, and B.Seeger: "The R*-Tree : An efficient and robust access method for points and rectangles", Proceedings of ACM SIGMOD 1990, pp.322-331(1990).
 [4] Simonas Saltenis, Christian S. Jensen, Scott T. Leutenegger, and Mario A. Lopez: "Indexing the positions of continuously moving objects", Proceedings of ACM SIGMOD 2000, pp.331-342 (2000).
 [5] Sionas Saltenis, and Christian S. Jensen: "Indexing of moving objects for location-based services, Proceedings of ICDE 2002, pp.463-473(2002).

向 直人 Naoto MUKAI

名古屋大学大学院情報科学研究科博士後期課程在学中。2003 名古屋工業大学大学院工学研究科博士前期課程修了。地理情報システムと高度交通情報システムの研究・開発に従事。情報処理学会学生会員。日本データベース学会学生会員。

馮 鈞 Jun FENG

名古屋大学大学院工学研究科博士後期課程在学中。1994 中国河海大学コンピュータ学院修士課程修了。地理情報システム、空間データベースシステムと空間検索の研究・開発に従事。情報処理学会学生会員。日本データベース学会学生会員。

渡邊 豊英 Toyohide WATANABE

名古屋大学大学院情報科学研究科教授。1974 京都大学大学院工学研究科修士課程修了。1975 同大学工学研究科博士課程中退。工学博士。統合化環境、分散協調環境、データベース環境、教育支援システム、文書理解に興味を持つ。情報処理学会、電子情報通信学会、日本ソフトウェア科学会、人工知能学会、システム制御情報学会、教育システム情報学会、日本データベース学会会員、ACM、IEEE-CS、AAAI、AAACE 各会員。