

経路近似による配送車両の時空間索引の形成

Spatio-Temporal Indexing of Transport Vehicles Using Approximate Routes

向 直人[▼] 渡邊 豊英[◆]

Naoto MUKAI Toyohide WATANABE

近年の位置情報システムの進歩に伴い、デマンド・バスと呼ばれる新しい交通サービスが注目されている。顧客は配送車両を乗合いで利用し、戸口から戸口までの送迎サービスをオン・デマンドに受けることができる。このような実時間の配送サービスにおいては、配送車両の予定経路を効率よく管理し、位置の問合せに対して迅速に回答する必要がある。本稿は、配送車両の予定経路に基づいた R-Tree 型索引の形成手法について提案する。配送車両は予定経路に変更が発生すると、索引サーバに自身の近似予定経路を送信する。この近似経路に基づいて配送車両を包含する矩形が形成され、配送車両が R-Tree 型に索引付けられる。最後に、従来手法と提案手法を計算機シミュレーションにより比較し、その結果を報告する。

Dynamic transport services using the contemporary real-time location technologies which include fleet management, traffic assignment, and others are a broad domain with a variety of problems. In such real-time problems, indexing of the current and anticipated future positions of vehicles is needed to allocate customers to vehicles effectively. Therefore, we propose an indexing based on R-Tree using the table of anticipated future routes of vehicles. The key problem to index moving vehicles is how to bound them by rectangles. The forming way of bounding rectangle in our approach, which is called Approximate Routing Rectangle (ARR), requires client-server transactions whenever the anticipated future routes are changed. Finally, our simulation results show that our approach outperforms traditional approach.

1. はじめに

近年、GPS(Global Positioning System)や無線通信に代表される位置情報システムが急速に発展し、移動オブジェクトの位置・速度等の定期的な取得が可能になった。これにより、デマンド・バスと呼ばれる新しい交通サービスの導入も実現可能になった。顧客は配送車両を乗合いで利用し、戸口から戸口までの送迎サービスをオン・デマンドに受けることができる。このような実時間の配送サービスにおいて、顧客の車両割当や配送順序に対する迅速な問題解決が必要である。

本稿は、配送車両の予定経路に基づいた R-Tree 型索引[1]の形成手法を提案する。一般的に、従来の R-Tree 型索引は

[▼] 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.ss.is.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

車両の位置・速度を定期的を取得するために高い通信・計算コストを必要とした。さらに、車両の予測位置は更新時から推測されたため正確性に乏しい。そこで、提案手法は割り当てられた顧客の乗降車位置に基づき車両の予定経路を決定し、それを直線近似されたベクトル表現にコード化する。全車両の予定経路はサーバに集められ一括管理される。この際、車両位置の問合せを高速に処理するために、車両を包含する矩形 Approximated Routing Rectangle (ARR) を予定経路に基づき形成し、車両を R-Tree 型に索引付ける。

本稿の構成は以下である。2章で対象とする配送問題を形式化する。代表的な R-Tree 型の移動オブジェクト索引手法である Time Parameterized R-Tree (TPR-Tree)[2][3]を3章で紹介する。4章で提案する索引手法の詳細を述べ、5章で TPR-Tree との比較実験の結果を報告する。最後に、まとめと今後の課題を述べる。

2. 形式化

複数台の車両が顧客を最短コストで訪問するという問題は Vehicle Routing Problem (VRP)[4]と呼ばれる。また、配送要求のタイプに基づき、静的配送と動的配送の2種類に分類できる。静的配送では、配送要求が予約の形であらかじめ与えられ、それらを一括処理して最適解を導出する。一方、動的配送では、配送要求が発生した時点で、既に運行している車両からその配送要求を処理するために最も適した車両を決定する。このため、動的配送の解は実時間で導出されなくてはならない。我々の対象とする問題は後者であり、以下のように形式化する。

道路ネットワークは、式(1)のように、ノード集合とエッジ集合から構成される無向グラフとして与える。ノードは交差点、エッジはノード間の道路セグメントを表す。また、エッジを図1のように単位距離 d で正規化する。この単位距離 d は単位時間で車両が移動する平均距離とする。つまり、車両は常にいずれかのノード上に存在していると仮定する。

$$G = (N, E)$$

$$N = \{n_1, n_2, \dots\} \quad (1)$$

$$E = \{e_i = (n - n') : n, n' \in N\}$$

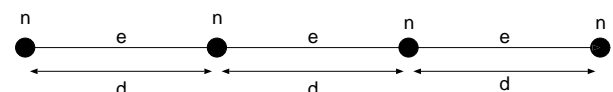


図1 エッジ長の正規化

Fig. 1 Normalization for Edge Length

顧客を式(2)で与える。また、顧客の時刻 t の位置を $c(t)$ と表す。顧客の歩行速度は時刻 t に依存せず一定速度 $|c|$ を保つとする。配送要求 $d(c)$ は要求発生時刻 t_s 、道路ネットワーク上の乗車ノード n_r 、降車ノード n_d の組で与えられる。

$$C = \{c_1, c_2, \dots\}$$

$$c(t) = (c_x(t), c_y(t)) \quad (2)$$

$$d(c) = (t_s, n_r, n_d) : n_r, n_d \in N$$

k 台の車両を式(3)で与える。また、車両の時刻 t の位置を $v(t)$ 、速度ベクトルを $\vec{v}(t)$ と表す。ただし、速度ベクトルは車両の進行方向を表し、その大きさは時刻 t に依存せず車両が単位時間に移動する平均距離 d とする。以降は、車両の

位置・速度ベクトルの組合せを車両ベクトルと呼び、 $\hat{v}(t)$ と表記する。

$$\begin{aligned} V &= \{v_1, v_2, \dots, v_k\} \\ v(t) &= (v_x(t), v_y(t)) \\ \vec{v}(t) &= (\vec{v}_x(t), \vec{v}_y(t)) \\ \hat{v}(t) &= (v(t), \vec{v}(t)) \end{aligned} \quad (3)$$

3. 関連研究

これまでに、提案されている移動オブジェクトのための R-Tree 型の索引手法は以下の 2 種類に分類される。

- 過去・現在位置の索引手法
- 現在位置とそこから予測される未来位置の索引手法
より詳細な分類は文献[4]で述べられている。提案手法は後者に分類され、その代表的な手法である Time-Parameterized R-Tree (TPR-Tree) の拡張形である。TPR-Tree は、定期的に移動オブジェクトから位置と速度ベクトルを取得し、そこから予測される線形の軌跡を包含する矩形を形成する。索引への問合せは特定領域内に存在する全ての移動オブジェクトを検索するものであり、以下の 3 種類に分類される。

- **Time-slice Query** 時刻 t に矩形 R に重複する移動オブジェクトを探索する(式(4))

$$Q_{\text{timeslice}} = (R, t) \quad (4)$$

- **Window Query** 時刻 t^{\rightarrow} から t^{\leftarrow} の間に矩形 R に重複する移動オブジェクトを探索する(式(5))

$$Q_{\text{window}} = (R, t^{\rightarrow}, t^{\leftarrow}) \quad (5)$$

- **Moving Query** 時刻 t^{\rightarrow} から t^{\leftarrow} の間に速度ベクトル $\vec{R}(t)$ で変形する矩形 R に重複する移動オブジェクトを探索する(式(6))

$$Q_{\text{moving}} = (R, \vec{R}(t), t^{\rightarrow}, t^{\leftarrow}) \quad (6)$$

次節に述べる提案手法は TPR-Tree の構造を受け継いでおり、これらの問合せを処理可能である。また、顧客の車両割当は Moving Query を拡張することによって実現する。

4. ARR-Tree

配送サービスにおける車両の索引手法 ARR-Tree を提案する。提案手法では、割り当てられた顧客の乗降車位置から車両の予定経路を決定し、それを直線近似された車両ベクトル表現にコード化する。全車両の予定経路はサーバに集められ一括管理される。サーバはクエリを高速に処理するために車両の索引構造を形成する。

4.1 トランザクション

本節では、車両(クライアント)・サーバ間のトランザクションについて述べる。各車両はシステムが顧客を割り当てることによって予定経路が決まる。ここでは、車両は顧客の乗降位置を最短経路で経由すると仮定する。また、車両の予定経路はノードを要素とした待ち行列で表現される。図 2 は、乗車位置 n_r 、降車位置 n_d が割り当てられたときの車両の予定経路表現である。

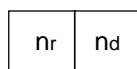


図 2 乗降位置による経路表現

Fig. 2 Route Expression with Ride-on and Drop-off Nodes

前述したように、車両は常に一定の時間間隔で道路ネットワークのいずれかのノード上に存在すると仮定する。そこで、車両の予定経路は図 3 のように、時刻 t をパラメータに持つ車両ベクトルの待ち行列で表現することができる。本稿では、待ち行列の各要素をトランザクションにおける転送パケットとみなす。

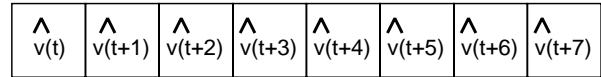


図 3 車両ベクトルによる経路表現

Fig. 3 Route Expression with Time-Parameterized Vector

ここで、通信コスト(転送パケット数)と位置精度のトレード・オフ関係が生じる。つまり、エッジの正規化距離 d を短くすれば、より正確な車両の位置をより短い時間間隔で把握可能である。しかし、転送パケット数は増大してしまい、サーバの負荷は大きくなる。よって、このトレード・オフのバランスを決定するために、経路の直線近似アルゴリズムを導入する。直線近似アルゴリズムは、経路の始点ノードと終点ノードを結ぶ直線の距離 L と、その直線から経路上の他のノードとの距離 l を計算し、その距離の比を求める。もし、その比が閾値 ω 以下であれば直線に近似する。このプロセスを部分経路に再帰的に適用し、近似候補のノードがなくなるまで繰り返す。図 4 は、図 3 の経路表現における距離比 l/L を示している。図 5 は近似後の経路を表している。結果的に、直線近似アルゴリズムを適用することで、経路表現は図 6 のように圧縮することができる。当然、近似の度合は閾値 ω に依存する。よって、閾値 ω は、配送車両の台数、通信帯域、処理速度等の通信インフラを考慮して決定しなくてはならない。

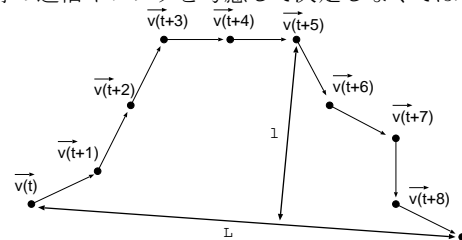


図 4 予定経路の直線近似

Fig. 4 Straight Line Approximation of Transport Route

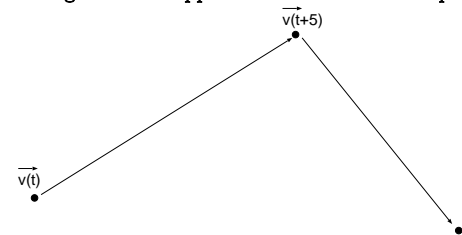


図 5 直線近似後の経路表現

Fig. 5 Transport Route Approximated by Straight Lines

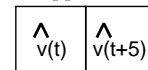


図 6 近似車両ベクトルによる経路表現

Fig. 6 Route Expression with Approximated Vector

また、予定経路が修正・更新されるタイミングは以下の二つのケースが存在する。

- **新規顧客の割当** 新規の顧客が車両に割り当てられることにより、車両の予定経路が変更される。
- **経路誤差率の増加** 予定経路はあくまで現在の状態から見込まれる経路であり、必ずしも、その時刻にその位

置に存在するとは限らない。例えば、もし道路が混雑していれば、車両の到着は遅くなると考えられる。そこで、予定位置と現在位置との誤差距離が一定値を越えたら、予定経路を再計算しサーバに送信する。

サーバは車両から送信された予定経路を表形式で管理する(図7)。また、受信した予定経路は展開され、各時刻の車両予測位置として置き換えられる。表中の各行は各車両の予定経路を表している。

time	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7
id								
1	$v_1(t)$	$v_1(t+1)$	$v_1(t+2)$	$v_1(t+3)$	$v_1(t+4)$	$v_1(t+5)$	$v_1(t+6)$	
2	$v_2(t)$	$v_2(t+1)$	$v_2(t+2)$	$v_2(t+3)$	$v_2(t+4)$	$v_2(t+5)$		
3	$v_3(t)$	$v_3(t+1)$	$v_3(t+2)$	$v_3(t+3)$	$v_3(t+4)$	$v_3(t+5)$	$v_3(t+6)$	$v_3(t+7)$
4	$v_4(t)$	$v_4(t+1)$	$v_4(t+2)$					

図7 予定経路の表管理
Fig. 7 Table of All Anticipated Routes

4.2 Approximated Routing Rectangle

R-Tree における、葉ノードのエントリ E は、式(7)で示すように、空間オブジェクト集合へのポイント $V_e \in V$ とその空間オブジェクト集合を最小面積で包含する矩形 MBR で与えられる。

$$E = (V_e, MBR) \quad (7)$$

しかし、R-Tree で車両のような時間経過と共にその位置を変化させる移動オブジェクトを索引付けるには、移動オブジェクトの位置・速度を頻繁に取得し、それに伴って索引構造の更新が必要になる。図8は索引木の更新のタイミングを示しており、 t_u は更新時刻、 I は更新間隔である。更新間隔 I を極端に短くしなければ、車両は包含から頻繁に逸脱してしまうため、R-Tree では正確な車両の索引を得ることは困難である。

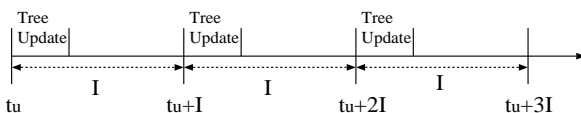


図8 索引木の更新
Fig. 8 Updates of Indexed Tree

TPR-Tree は、式(8)で示すように、包含する車両の更新時刻の最高・最低速度で矩形 (Conservative Bounding Rectangle: CBR) を膨張・収縮させることによって移動オブジェクトを索引付ける。

$$E = (V_e, CBR(t))$$

$$\overline{CBR}_{\max} = \max(\vec{v}(t_u)) : v \in V_e \quad (8)$$

$$\overline{CBR}_{\min} = \min(\vec{v}(t_u)) : v \in V_e$$

図9 (a) は4台の車両の軌跡を包含する CBR を表している。CBR は更新時刻のみの速度を用いるため、矩形が冗長に膨張してしまうことが多い。矩形面積の増加は矩形同士の重複の原因となり、探索効率の低下を招く。また、車両が進行方向を変えてしまうと矩形から大きく逸脱することも多

い。矩形はその面積を可能な限り小さく保ち、かつ、車両が逸脱することなく包含することが望ましい。

一方、提案する矩形 (Approximated Routing Rectangle: ARR) の形状は車両の近似予定経路に基づく。まず、表から索引に必要な更新時刻 t_u から $t_u + I$ までの列を選択する。もし、選択された時刻のバケットが存在しなければ、車両は前時刻の位置に静止していると仮定し、欠損部分を補間する。例として、更新時刻 $t_u = t$ 、更新間隔 $I = 4$ とすると、図7中の時刻 t から $t+4$ までのバケットが選択される。このとき、 $v_4(t+2)$ のバケットは $v_4(t+3)$, $v_4(t+4)$ にコピーされる。次に、式(9)により、各時刻の車両の最大・最小位置を計算し、それを包含矩形とする。図9(b)は4台の車両の直線近似された軌跡を包含する ARR を表している。面積は CBR に比べ冗長な部分が少なく、さらに逸脱は直線近似幅内に抑えることができる。

$$E = (V_e, ARR(t))$$

$$ARR_{\max}(t) = \max(v(t)) : v \in V_e \quad (9)$$

$$ARR_{\min}(t) = \min(v(t)) : v \in V_e$$

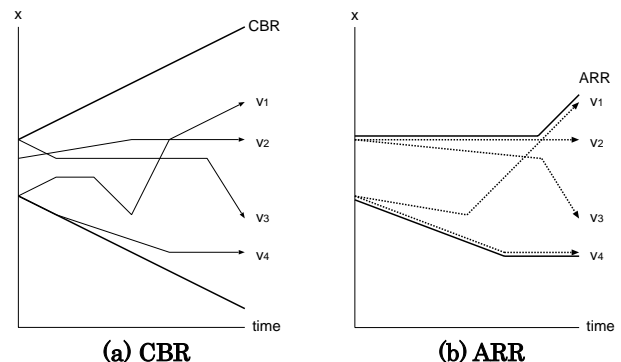


図9 包含矩形
Fig. 9 Bounding Rectangle

4.3 問合せ

顧客を割り当てる車両の決定は、式(10)で表すように、Moving Query の拡張として定義できる。顧客が要求発生時刻 t_s から速度 $|\vec{c}|$ で移動すると仮定し、顧客の移動可能領域に重複する包含矩形を選択する。もしくは、総面積が最小となる包含矩形を選択する。この操作を根から葉まで繰り返すことで、顧客を割り当てる車両の候補が決定される。結果的に、顧客を短時間で乗車させることが可能な車両の候補が得られるが、最終的に候補から割り当てる車両を決定するには、車両負荷や顧客満足度等の要因を考慮しなくてはならない。割当手法の改善は今後の課題である。

$$Qallocate = (c(t_s), \vec{c}, t_r, (I - t_s)) \quad (10)$$

5. 実験

提案手法 ARR と従来手法 CBR の比較実験を行い、探索コスト (矩形面積)、通信コスト (通信バケット数)、位置精度 (矩形からの逸脱距離) の3項目を評価した。実験環境を以下に述べる。3台の車両を名古屋市の1km四方の道路ネットワーク(1000×1000ピクセル)上でランダムに移動させた。

また、正規化距離 $d = 20$ ，車両速度 $|\vec{v}| = 40\text{km/h}$ ，更新間隔 $I = 5, \dots, 15$ ，近似閾値 $\omega = 0, \dots, 0.2$ とした。

図 10, 11, 12 は索引木の更新間隔 I を変化させた結果である。CBR の面積は更新間隔 I に依存して増加している。一方、ARR は更新間隔 I には依存せず小さい面積を保っている。同様の傾向が通信パケット数においても見られるが、ARR の方がより多くの通信パケット数を必要とすることを示している。逸脱距離は CBR の方が ARR と比べて低い値を示した。しかし、その性質は大きく異なる。つまり、CBR においては、車両が進行方向を変化させたとき大きく矩形から逸脱してしまう可能性があるのに対して、ARR は逸脱距離を直線近似幅内に抑えることができる。図 13, 14, 15 は直線近似の閾値 ω を変化させた結果である。閾値 ω は包含矩形の面積には影響しない。CBR の通信パケット数は常に一定であるのに対し、ARR は閾値 ω を大きくするほど、通信パケット数を減少させる。たとえ閾値 ω が小さい値でも、正規化されたエッジの直線近似が発生するためその効果は大きい。また、近似率が高いほど、車両の逸脱距離は増加してしまう。結果的に、ARR は CBR に比べて閾値 ω を変えることによって環境に柔軟適応可能であるといえる。また、包含面積を抑えることによって探索効率を向上させることが可能である。

6. まとめと今後の課題

本稿では、配送サービスにおける車両の予定経路に基づいた R-Tree 型の索引手法を提案した。車両は予定経路に変更が発生すると、索引サーバに自身の近似予定経路を送信する。この近似経路に基づいて配送車両を包含する矩形 ARR を形成し、車両を R-Tree 型に索引付ける。最後に、提案手法と従来手法の比較実験を行い、提案手法がより柔軟で効果的な索引を構成できることを示した。今後の課題は、提案手法を現実社会における車両配送問題に適用することである。

[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。

[文献]

- [1] A. Guttman: "R-trees: a dynamic index structure for spatial searching", Proc. Of ACM SIGMOD, pp. 47-57(1984).
- [2] 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).
- [3] Sionas Saltenis, and Christian S. Jensen: "Indexing of moving objects for location-based services, Proceedings of ICDE 2002, pp. 463-473(2002).
- [4] Lau, H.C., M. Sim and K.M. Teo: "Vehicle Routing Problem with Time Windows and a Limited Number of Vehicles", European Journal of Operational Research, 559-569(2003).
- [5] Y. Tao and D. Papadias: "Spatial Queries in dynamic environments", Data & Knowledge Engineering, pp. 75-101(2004).

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。

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学生会員、ACM、IEEE-CS、AAAI、AAACE 各会員。

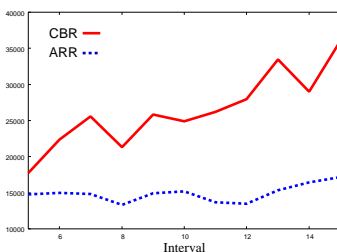


図 10 面積 (I)
Fig. 10 Area (I)

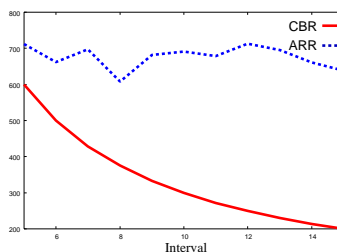


図 11 通信パケット数 (I)
Fig. 11 Number of Packets (I)

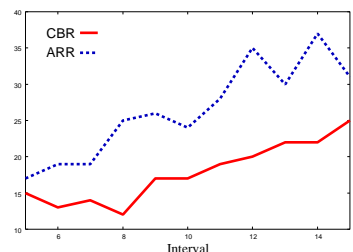


図 12 逸脱距離 (I)
Fig. 12 Deviation Distance (I)

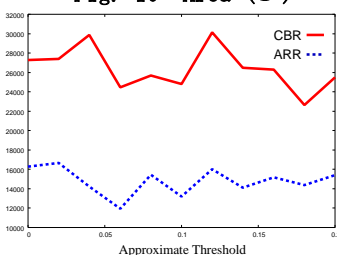


図 13 面積 (ω)
Fig. 13 Area (ω)

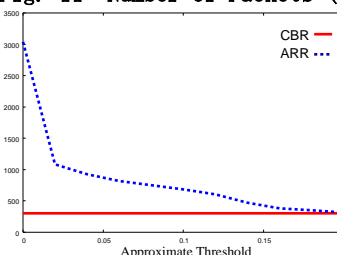


図 14 通信パケット数 (ω)
Fig. 14 Number of Packets (ω)

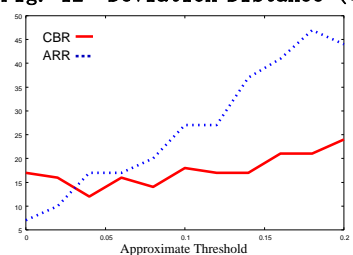


図 15 逸脱距離 (ω)
Fig. 15 Deviation Distance (ω)