

Swarmを用いたデマンドバスのシミュレーション環境の構築

Development of a Simulation Environment for Dial-a-ride Systems Using Swarm

坪山 幸*¹ 水野 一徳*¹ 佐々木 整*¹ 西原 清一*²
 Tsukasa Tsuboyama Kazunori Mizuno Hitoshi Sasaki Seiichi Nishihara

*¹ 拓殖大学大学院電子情報工学専攻 *² 筑波大学大学院コンピュータサイエンス専攻
 Department of Computer Science, Takushoku University Department of Computer Science, University of Tsukuba

Dial-a-ride bus systems can be significant for vulnerable road users who depend on public traffic systems. However, It is necessary to sufficiently discuss usability in advance according to the distribution of the demands, or users for dial-a-ride buses. We have developed a simulation environment for evaluation of dial-a-ride bus systems on Swarm developed by Santa Fe Institute. We also conduct some tentative simulations, showing evaluation of usability when demands are increased.

1. はじめに

近年、地方都市及び周辺部では路線バスなどの公共交通機関の利用者が減少し、廃止または運行回数が減少されている。しかし、高齢者をはじめとする車を運転できない方々の交通手段として公共交通機関は非常に重要なものになっている。この問題の対策として、固定路線バスと比べて時間と経路を利用者の需要に応じて変更させることができ、利便性が高いデマンドバスが注目されているが、デマンドバスは移動経路が定まっていなため、利用者数によって移動距離が変わる。そのため利用者数が一定数を超えると利便性、採算性が低下する場合がある。また、デマンドバスの利便性、採算性を保つためには、デマンド(利用者)の分布に応じた事前の十分な検討が必要である。

本研究では、サンタフェ研究所が開発したマルチエージェントシミュレータである Swarm を用いてデマンドバスのシミュレーションをできる環境を構築する。また、デマンド数に応じたデマンドバスの利便性に関するシミュレーション結果を示す。

2. 研究の概要

2.1 デマンドバス

デマンドバスとは、通常一定の路線の決められた停留所、スケジュールでお客を乗降させる固定路線バスに対し、決められた停留所、スケジュールがなくお客の要望に応じて乗降場所やバス経路を自由に変更できるバスの運行形態である。利用する際、利用者は電話やインターネットが利用可能なパソコンを通じて、家庭やオフィスなどから事前の予約を行ない、予約状況に応じてバスが配車される。従来のバスと比べ、柔軟な運行ができる特徴がある。

主な利点として、交通弱者の「移動の足」となることで、モビリティを向上させることができる。また、通常の固定路線バスは利用者の有無に関わらず運行されるのに対し、デマンドバスは利用者がいない場合には運行されず、利用者が希望する乗車時刻を設定でき、行動スケジュールに応じたバスの利用が可能となり、利用者は長時間のバス待ちから解放される。事業者側も利用状況に応じては配車が可能となり、乗客がいない空バスの運行を防止でき、運行コストを下げる事ができる。さらに環境面

においても無駄に走ることがなくなるため CO2 を削減することができる。

デマンドバスには、巡回せずに電話等による(複数の)利用者の要望を受けて、希望乗車地点へ迎えに行く形態であるフルデマンドバス、路線バスの運行形態の一種で、条件に応じて基本路線外に迂回経路する形態であるオンデマンドバスの2種類がある。本研究では、フルデマンドバスを対象とする。

近年、デマンドバスに関する研究は巡回セールスマン問題や車輻配送問題から派生した問題である Dial-Ride Problem のスケジューリングアルゴリズムに関するものが多く、さまざまな条件での最適化アルゴリズムが行なわれている。しかし、デマンドバスが具体的にどのようなメリット、デメリットなどのどのような条件で発揮されるなどの視点での評価はほとんどされていない。

2.2 Swarm

Swarm とは、サンタフェ研究所が開発したマルチエージェントシミュレータで、日常の複雑現象をセルオートマトンのような形で表現することができるという特徴をもつ。

Java または Objective-C から使用でき、現象をオブジェクト指向的にモデル化しようとする場合に利用しやすく、具体的には次のような GUI の機能がある。

- シミュレーション内のオブジェクトのフィールドやメソッドに対話的にアクセス可能(プローブ:図3参照)。
- シミュレーションの様子(二次元平面上でのエージェントの分布)や統計的指標(線グラフやヒストグラム)をグラフィカルに表示。
- シミュレーションの各階層に独自の時計(スケジューラ)を持つ。

2.3 ダイクストラ法

本システムに用いる探索方法としてダイクストラ法を使用する。ダイクストラ法とは 1959 年にエドガー・ダイクストラによって考案されたグラフ理論における最短経路問題を解くためのアルゴリズムである。2 頂点間の最短経路を効率的に求めることができ、最小コストとなる経路から順次コスト計算を行なっていくことにより、効率的に最適解を得ることができる。

連絡先: 坪山 幸 拓殖大学大学院 電子情報工学専攻
 y3m311@st.takushoku-u.ac.jp

ダイクストラ法の具体的な特徴として以下の内容がある。

- 負のコスト(距離, または時間)を持つエッジは扱えない.
- 特定のノードからの最短距離およびその経路全てのノードに対して求める.

また, 以下にダイクストラ法のアルゴリズムを示す.

- (1) 始点を決める.
- (2) すべての点において, 始点からの距離として最大値を設定する.
- (3) 始点に隣接している全ての点において, 始点からの距離を求める.
- (4) 始点に隣接している全ての点の中で, 最短の点を確立する.
- (5) 確立した点に接続されている全ての点において, 始点からの距離を求める.
 - (5.1) 新しく計算された距離が現在の距離より短い場合, 新しく計算された距離を始点からの距離とする. 以前の経路は使用しないものとする.
 - (5.2) そうでない場合, 現在の距離を始点からの距離にする. さらにその経路を使用しないものとして切断する.
- (6) 現時点で始点からの最短の点を確定とする.
- (7) 全ての点が確立するまで, (5)~(6)を繰り返す.

3. デマンドバスのシステム設計

3.1 システム概要

本研究におけるシステム概要について説明する. 本システムの概要を図1に示す.

まず, ユーザがプローブでパラメータの入力を行ない(表1参照), 道路情報, デマンド位置情報, 目的地位置情報, 位置情報が取得され, 経路探索を行なう. 探索された経路情報を表示・解析部に情報を取得し, バス移動・統計情報を出力する. 従って本システムは以下の部分により構成する.

- プローブからパラメータを入力.
- 道路情報取得.
- デマンド位置情報, 目的位置情報を取得.
- 現在位置取得.
- 取得した情報から経路探索.
- バス移動, 統計情報取得.

3.1.1 道路情報

交差点や道路の節目をノードとし, ノードとノードを結ぶ道路をリンクで表現する. 各ノードにはノード番号を与える. 与えられたノード番号から座標として情報を返し, 経路計算や経路案内に使用する.

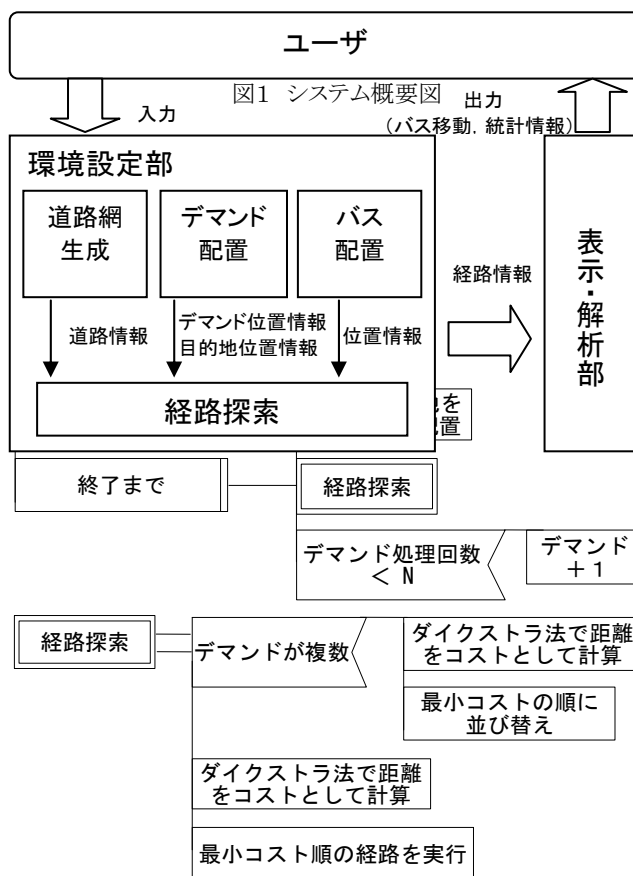


図2 本手法のアルゴリズム

3.1.2 デマンド情報

デマンド情報では, デマンド出発地, 目的地位置の情報と ID を持つ. 実行したとき, 最初にデマンドの出発地点のデータを保持させる. デマンドの出発地点に到着したとき, そのデマンドの ID をもとに目的地位置情報を取得する.

3.2 基本環境

本システムでは, 以下のような環境を基本とする.

- 都市全体は正方形とし, 街路は基盤目状であるが, 任意の割合で道路(リンク)が削除されている.

- すべての交差点にバス停があるものとする。
- 交通渋滞は存在せず、バスは一定の速度で運行できるものとする。
- バスには何人で乗車することができ、乗降時間は無視できるものとする。
- デマンドの出発地、目的地は一様にランダムに決定されるものとする。
- デマンドは一人とする。
- バスは一台とする。
- バスの乗り換えは行なわないものとする。

3.3 処理手順

本手法によるアルゴリズムを図2に示す。

まず、プローブでパラメータの設定を行ない、設定された値により、道路網が生成され、バスの配置、デマンドの出発地、目的地がランダムで配置される。なお、デマンドの出発地と目的地の経由地点のデータを保持し、バスが各経由地点に到着した場合は、その地点から削除される。経路探索方法として、本手法では、ダイクストラ法で距離をコストとして計算を行ない、最小コスト順に実行させる。ここでは、デマンド全部の処理回数をN回繰り返し、処理が終了した場合、デマンドを1プラスし、デマンド数が終了の値まで実行したら、デマンドの平均達成処理時間を出力する。

4. 実験

4.1 実験条件

本研究では、図2の手順に従い、以下に示す2つの状況でシミュレーション実験を行なった。

- (1) 街のサイズ11×11
- (2) 街のサイズ20×20

なお、上記の2つの状況とも、間引き率を0.1、初期デマンド数=2とした。

表1の worldXSize と worldYSize は実行時の道路網のサイズ、xPos と yPos はバスの初期位置、nはn×nの道路網のノードの数、d1は初期デマンド数、mabiki は道路網のリンクを削除する割合、speed は実行したときのバスの速度を表している。

また、実験環境としては、Windows XP が搭載の PC (CPU: 2.83GHz Intel Core2, RAM: 4GB) を用いて、プログラムを Java で記述した。

4.2 実行結果

図3は初期の実行画面で、コントロール画面とプローブを表している。図4は、街のサイズを11×11とした場合の本システムの実行時の道路網の画面を表している。

実行画面の色として、黒は道路網、緑はバス、赤はデマンドの位置、黄はデマンドの目的地を表している。図5、図6はそれぞれ4.1節における状況(1)、(2)に対する実験結果を表している。図5、図6とも初期デマンド数から終了デマンド数10までのそれぞれを100回シミュレートした場合(つまり図2における N=100)のデマンドの平均処理ステップ数を表している。

表1 プローブの内容

パラメータ	意味
worldXSize	シミュレートの大きさ(X軸)
worldYSize	シミュレートの大きさ(Y軸)
xPos	バスの初期位置(X軸)
yPos	バスの初期位置(Y軸)
n	n×nの道路網
d1	デマンド数
mabiki	間引き率
speed	バスの速度

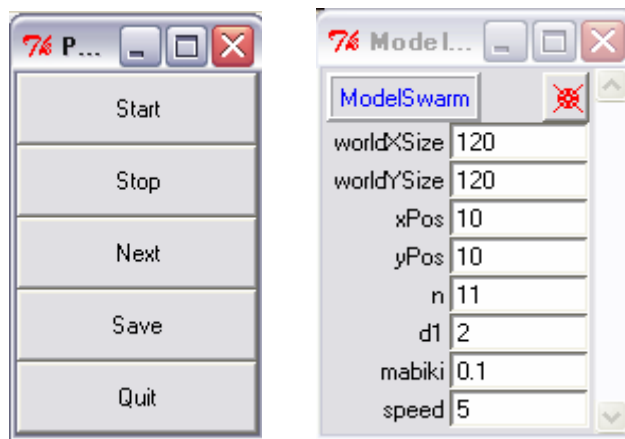


図3 実行画面

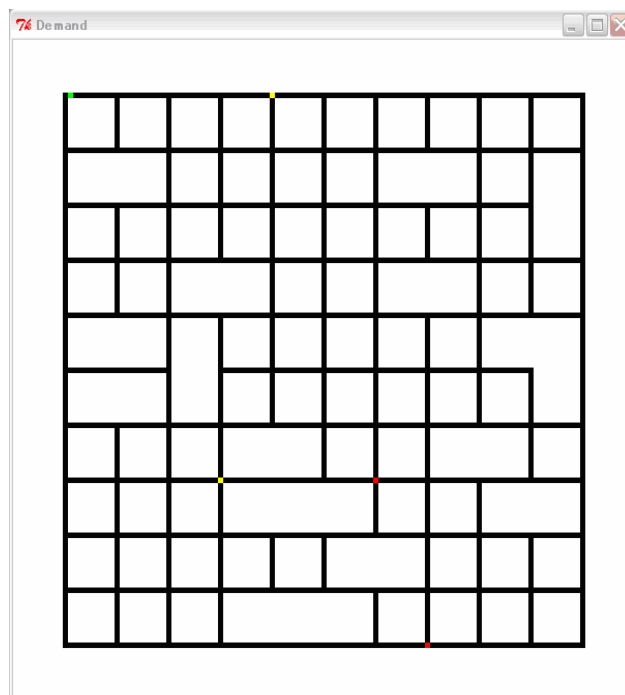


図4 実行時の道路網

図5よりデマンド数の増加に従って、処理ステップが線形的に増大している。またデマンド数が増加に従って、処理ステップ数の間隔が徐々に短くなっていることがわかる。

図6より図5と同じくデマンド数の増加に従って、処理ステップが線形的に増大している。また、結果より道路網を11×11から20×20に大きくすると、全体のデマンド平均処理ステップ数が約2倍ほど増えていることがわかる。

4.3 考察

4.2節の実験結果より、デマンド数の増加に従って、処理ステップが線形的に増大している結果が示された。これは、一般的にデマンドが増えると処理ステップ数も増えることは必然であることがわかるが、処理ステップ数の間隔が徐々に短くなっていることから、デマンドの数がある一定の数に増えることにより、処理ステップ数が変わらなくなるのではないかと考えられる。

また、街のサイズの規模を大きくした場合、その規模に比例して、デマンドの処理にも時間がかかっている。これは、街のサイズは約2倍であるが、街のノードや街のリンク自体は4倍である。しかし、図6の実験結果と図5の実験結果を比べて、処理ステップ数が約2倍であることから、街のサイズと処理ステップ数は比例していると考えられる。

今回、実験を行なった状況は非常に簡素な状況であり、現実的な状況に近いとはいえない。より多様な状況やシナリオに対応できるように本システムの拡張していくことが必要不可欠である。具体的には、以下のようなことが考えられる。

- バスの複数での実行。
- バス乗客の制限。
- デマンドを一か所に複数可。
- シミュレートする環境を具体的な都市を対象に評価。
- 渋滞や乗客の混雑率。
- ランダムで利用率の変化。
- 本手法で使用した経路探索のダイクストラ法ではなく、他の最適化アルゴリズムの導入。
- プローブの内容追加。
- 固定路線バスのシミュレーションを作成し、デマンドバスとの比較。

これらの機能を付与することで、より現実的なシミュレーションの実行環境にしていく予定である。

5. おわりに

本報告では、Swarmを用いたデマンドバスのシミュレーション環境の構築を行なった。また、デマンド数の増加に対してデマンド処理時間を計測するシミュレーションを行なった。

今後はより多様な環境やシナリオに対応できるように本システムの拡張を行なっていく予定である。

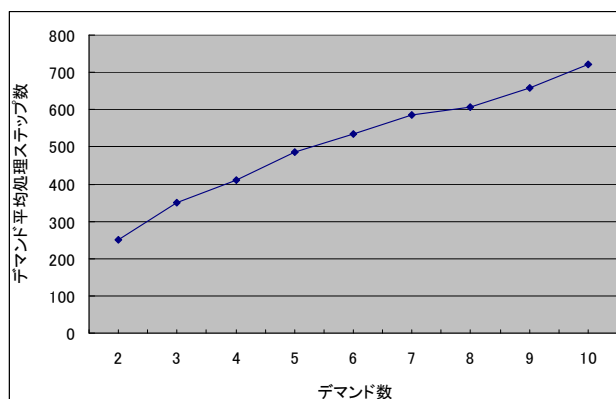


図5 実行結果

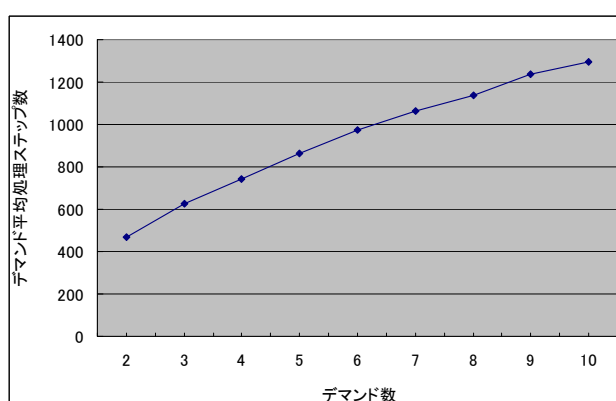


図6 実行結果

参考文献

[野田 五十樹 2008] 野田 五十樹, 篠田 孝裕, 太田 正幸, 中島 秀之: シミュレーションによるデマンドバス利便性の評価, 情報処理学会論文誌, Vol. 49, No. 1, pp. 242-252, 情報処理学会 (2008)

[伊庭 斉志 2007] 伊庭 斉志: 複雑系のシミュレーション-Swarmによるマルチエージェント・システム-, コロナ社

[小野 良太 2012] 小野良太, 川村秀憲, 鈴木恵二: オンデマンドシステム設計へ向けたシミュレーション評価, 第26回人工知能学会全国大会, 2012, 3F2- OS-10-5