

# 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

坪山 幸<sup>1</sup> 水野 一徳<sup>1</sup> 佐々木 整<sup>1</sup> 西原 清一<sup>2</sup>  
 拓殖大学工学部情報工学科<sup>1</sup> 筑波大学大学院コンピュータサイエンス専攻<sup>2</sup>

## 1 はじめに

近年、少子高齢化の進展により、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。しかし、デマンドバスは移動経路が定まっていないため、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ります[1]。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。

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、希望乗車地点へ迎へ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。

### 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

Development of a Simlation Enviroment for Dial-a-ride Systems Using Swarm

Tsukasa Tsuboyama<sup>1</sup>, Kazunori Mizuno<sup>1</sup>, Hitoshi Sasaki<sup>1</sup>, and Seiichi Nishihara<sup>2</sup>

<sup>1</sup>Department of Computer Science, Takushoku University

<sup>2</sup>Department of Computer Science, University of Tsukuba

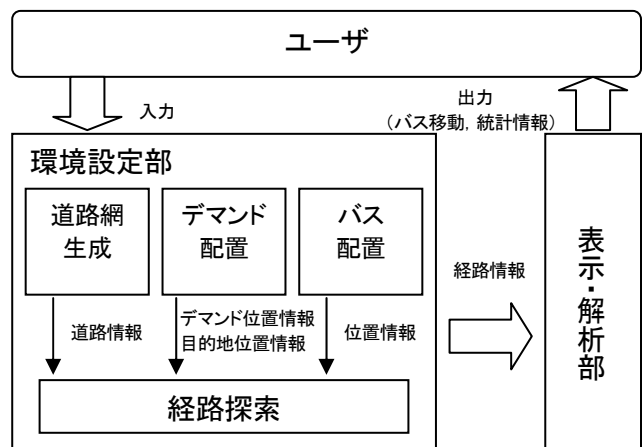


図1 システムの概要図

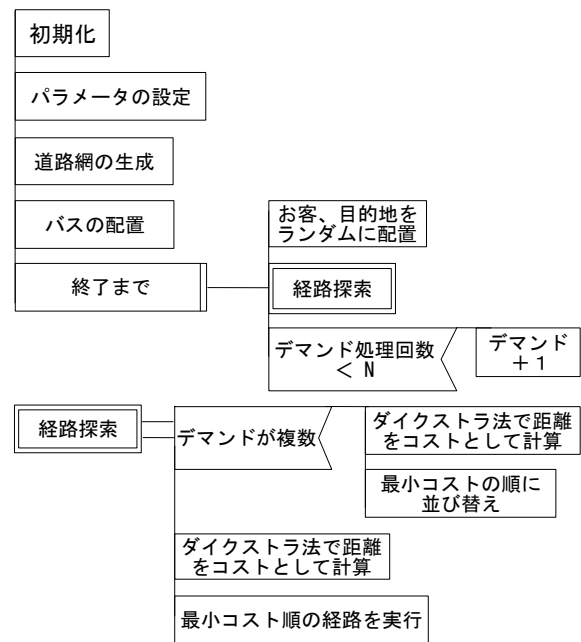


図2 本手法のアルゴリズム

特徴をもつ [2].

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。

### 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。本システムの概要図を図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。

#### 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する。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。
- ・ バスには何人でも乗ることができ、乗降時間は無視できるものとする。
- ・ デマンドの出発地、目的地は一樣にランダムに決定されるものとする。
- ・ デマンドは一カ所に一人とする。
- ・ 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。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 実行例

図3は、街のサイズを11×11とした場合の本システムの実行画面を表している。図2の手順に従って、初期のデマンド数は2、N=100とし、デマンドを徐々に増やし、デマンド数が10になるまでのデマンドバスのシミュレーションを行なった。

図4は各デマンド数に対する平均処理時間を表している。図4より、デマンド数が増えるに従い、処理時間がほぼ線形的に増大している結果が得られていることが分かる。

### 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。

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

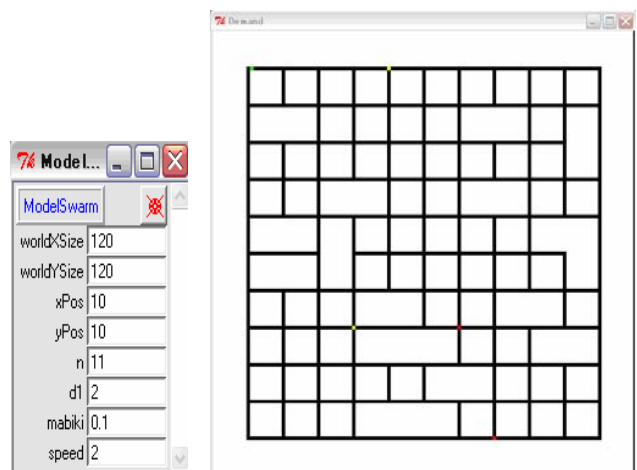


図3 実行画面  
(左はプローブ、右はシミュレート画面)

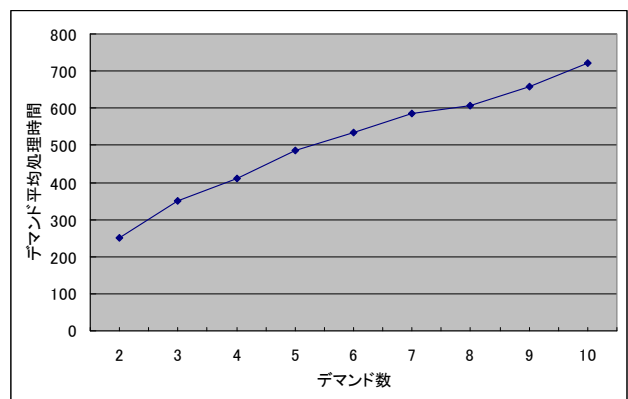


図4 実行結果

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。

### 参考文献

- [1] 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)
- [2] 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