ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
ADDIS ABABA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Decentralized Motion Coordination Method Design using CO-FIELD Approach of SWARM AI metaheuristics for Improving the Reliability of Bus Transit System

By Sisay Mitiku

<u>Advisor</u>

Dr. Getachew Alemu

A Thesis Proposal Submited to the School of Graduate Studies of Addis Ababa Institeut of technology in Partial Fulfilment for the Degree of Master of Science in Computer Engineering

November 28, 2014
ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
ADDIS ABABA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Decentralized Motion Coordination Method Design using CO-FIELD Approach of SWARM AI metaheuristics for Improving the Reliability of Bus Transit System

By Sisay Mitiku

Advisor

Dr. Getachew Alemu

APPROVED BY BOARD OF EXAMINERS

Chairman, Dept. of Graduate committee ___________________________ Signature ___________________________

Dr. Getachew Alemu _
Thesis Advisor / Chair ___________________________

____________________ ___________________________
Internal Examiner

____________________ ___________________________
External Examiner

Declaration

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this and any universities, and all sources of materials used for the thesis work have been fully acknowledged.

Name: Sisay Mitiku Desta signature: ____________________-

Place: Addis Ababa

Date of submission August 2013

This thesis has been submitted for examination with my approval as a university advisor.

Dr. Getachew Alemu Signiture: __________________
Advisor name

Acknowledgement

Firstly, I would like to thank the almighty God for his unconditional support both in the good time and the bad time of my life. Then I would also like to thank those people who have, in one way or another, helped me to make this work a reality. Especial I would like to thank my advisor Dr. Getachew Alemu for his tolerance, patience and encouragements. Then I would like to thank my families, for their priceless material and non-material support.

Abstract

Bus transit system plays a major role in combating both air pollution and road congestion and is one of the most important modes of transportation. In spite all this however, it is not considered to be reliable mode of transportation by its customers. The complex nature of the transportation system in general and bus transit operation in particular makes it difficult for the application of traditional mathematical model. In this thesis, the problem of regulating and monitoring the reliability of a bus transit system using a SWARM artificial intelligence solution is addressed. The increasing availability of near-real time data from intelligent bus transit system makes the applicability of such solution more attractive. As the bus transit system is distributed and stochastically dynamic because of uncertain inter-stop trip time and uneven passenger distribution, the application of interaction based and emergent self-organized solution such as swarm ai solution is highly recommended. The problem is formulated as a distributed motion coordination problem. A gradient field (co-field) coordination model of swarm artificial intelligence which is inspired by the nature of naturally found fields such as electro statistic and electromagnetic fields is used to solve the proposed model. Multi-agent simulation model is used both to model the bus transit system and to iteratively design the SWARM artificial intelligence metaheuristics. The simulation is implemented with NetLogo integrated development environment so that the desired emergent phenomena is designed and evaluated. Line 31 of Ambessa Awtobis organization, Addis Ababa, Ethiopia, is taken as a case study to improve the reliability of the developed multi-agent simulation. Different simulation experiment is carried out and different measure of effectiveness of the system is collected. The result from the multi-agent simulation experiment shows that the proposed method is adaptive to wider passenger density scenarios. From the result, we can conclude that decentralized metaheuristics of control methods without any sort of formal mathematical model can be a viable solution for improving the bus transit system reliability problem. More over this method also helps to solve the problem of how effectively to utilize the increasingly available huge near-time data from intelligent transit system. Our recommendation is that a research on design support system of swarm Artificial intelligence solution, such as reducing a programming overhead for rapid prototyping of emergent phenomena is worth doing in the future.

Key words: Bus transit reliability problem, Bus holding, Computational field, Multi-agent simulation, Case study,

Table of Content

List of Abbreviation

AVL……………………………………………………..Automatic Vehicle Location

APC…………………………………………………...Automatic Passenger Counter

ITS…………………………………………………….Intelligent Transit System

KISS…………………………………………………..Keep It Simple Stupid

KIDS………………………………………………….Keep It Descriptive Stupid

TAPAS………………………………………………..Take a Previous model, Add Something

SWARM AI…………………………………………..SWARM Artificial Intelligence

Co-fields………………………………………………Computational Fields

MOE…………………………………………………..Measure of Effectiveness

List of Figures

List of table

CHAPTER ONE

INTRODUCTION

1.1 Overview and Motivation

Public city bus (Bus transit system) plays a vital role in fulfilling the transportation requirement of both developed and developing countries. Because of their low road space and fuel consumption per passenger capacity, they play their part in alleviating the problem of both congestion and atmospheric pollution. It is for this reasons that governments both from developed and developing countries usually encourage the use of this mode of transportation.

In spite of all these advantage, however, this mode of transportation is not perceived to be very reliable means of transportation system by the general public [1-3]. This problem is aggravated by the fact that the public use the reliability of this system as the most important factor in measuring the value of this system[2-3]. This fact indicate that, before a pronounceable increase of the ridership share of public transport system is observed, as most governments want to see, the reliability problem of this system should be addressed.

The difficulty with ensuring the reliability of this mode of transport is that, as part of the general transportation system, it is subjected to many complex factors (non-deterministic and very dynamic in nature) such as traffic light, congestion, uneven passenger demand distribution and so on. For this reason theoretically solving the reliability of this system using conventional mathematical methods and practically testing and implementing different proposed solution has been a difficult challenge. Moreover the manual nature of data measurements and collection methods makes the whole process less reliable and more costly [4].

To address the difficulty in data collection and measurement methods, bus transit system operators especially from developed countries started to equip their vehicles with modern electronic apparatus such as Automatic vehicle locating system (AVL), Automatic Passenger counter (APC), and so on. And operators that utilize these technologies are usually called Intelligent transit System (ITS) [5][6] and it helps them for collecting and resourcefully utilizing near real-time data so that they improve their reliability. Example of such system is shown in figure 1.1.

As a result of this, researches that deal about how effectively and resourcefully harness the data which is collected from ITS, emerged (e.g.[7-12]). These research's' problems are usually formulated as a coordination problem in which different entities (agents) of the bus transit system such as bus, bus stops, passengers, etc are made to cooperate with each other in near real-time using these ITS technologies so that they collectively address the bus transit reliability problem as a whole [13-21].

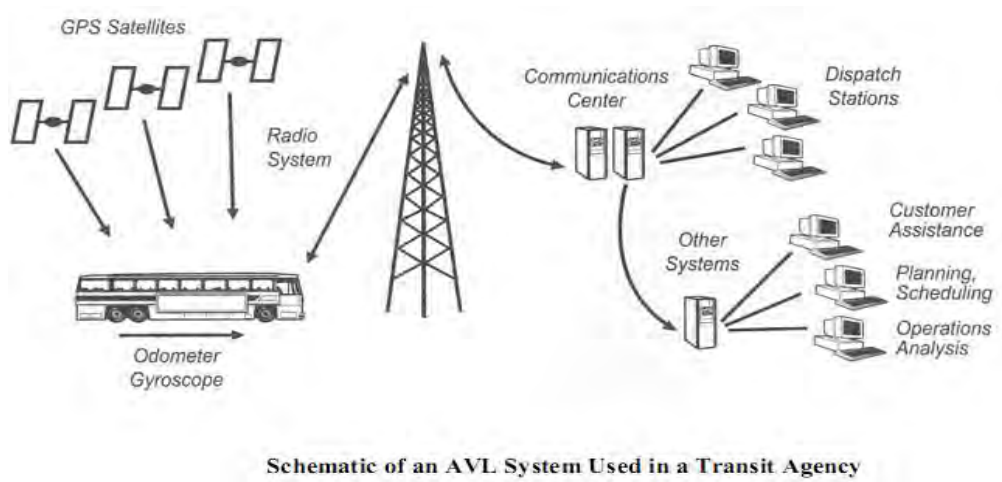


Figure 1.1 ITS Fleet management system (source [5])

This research is the extension of this trend in which the coordination problem is addressed with a nature-inspired metaheuristics called SWARM Artificial intelligence (SWARM AI) which is an interaction based solution that imitate its problem solving methods from naturally occurring phenomena such as social insets (e.g. ants, bees, termites) and physical phenomena like gravitational and electrostatic phenomena.

1.2The Bus Reliability Problem: Bus Bunching

Bus bunching is a well known city bus reliability problem which was identified almost 60 years ago and whose solution is so difficult to drive using conventional mathematical methods that it is still an active research area both in academy and professional research institution.

As part of the general transportation system, bus transit system is subjected to different non-deterministic factors such as non-deterministic inter-stop trip time (because of traffic light, weather condition, accident, and so on) and non-deterministic dwell time at different bus stop (because of uneven passenger distribution for boarding and alighting). For these reasons, the service from bus transit system is inherently unreliable.

And as one phenomenon, which can be taken as a sign of bus transit unreliability, the phenomena of bus bunching occurs as follows: as the time a bus spends at a bus stop increases with the number of users that need to board and alight the bus, and as the expected number of users waiting to board at any bus stop generally increases with the time between successive bus arrivals time increases, the two properties in tandem cause a positive feedback effect, that once a tendency for bus bunching is created, it will always leads for more bunching until the buses start to go together.

This means if a bus is delayed or slowed a small amount (because of random trip time for example) so that the number of passengers waiting at the next bus stop is larger than expected, the bus will have to dwell longer, slowing it down further. Similarly, if a bus that is momentarily speed up has fewer passengers to board in the next stop, and then this will speed it up further. This effect grows exponentially over time and if the space between two buses starts decreasing, they will eventually have little or no space between them and start moving together [3]. Assuming that the number of buses on a single route and the length of a route is fixed, for every group of bunched buses, there will be locations along the route not served by buses for long periods of time.

Bus bunching affects both the operator side and the passenger side unfavorably. It costs the passenger for example by increasing the mean and the standard deviation of waiting time at a bus stop. It also affects the service providers by making the number of passenger served by each bus uneven [3]. This means when two buses are bunched, the bus in front will become overloaded, while the bus behind will become almost empty. If vehicle overtaking is not allowed, this phenomenon will continue until both buses make to the end of the route.

The difficulties with solving the bus bunching problem is that, like vehicle congestion problem, the bus bunching problem is the result of the unprogrammed and non-deterministic interaction between the different distributed entities (agents) of the transportation system in which the bus transit system operate. This means, it is difficult to trance the cause of these problems to a single source. Rather the cause of the problem is the sum total result of a complex and unprogrammed interaction of the different factors and actors (e.g. other buses, user and non-user passengers, other cars, traffic light, weather condition, accidents and so on) in which the bus transit system operates. As their interaction is so unprogrammed and only known probabilistically, it is difficult to model the causal link between the behavior of the individual agents and the aggregate macroscopic behaviors (e.g. congestion, bus bunching and so on) of the system as a whole. It is for this reason that an interdisciplinary approach can be a viable means of addressing this problem as this research advocates [13][18-21].

1.3 Swarm Artificial Intelligence (SWARM AI)

Social insects like ants, bees, termites have lived on earth and survived for millions years. Although they have small size with crude sensing and actuating capabilities, considered as a group, it is observed by so many scientists that as a social creatures, they can do many amazing complex activities. These activities includes finding the shortest path to their food source, building nests which is bigger than thousands of their size, transporting large sized objects cooperatively and so on. The way of their problem solving is characterized by distributedness, cooperative, indirect communication through the environment, etc. Because of these characteristics, scientists especially from computer science and artificial intelligence started to model and imitate the way these insects approach their problem (they take inspiration from them). The result of such effort is a toolkit of bio-inspired (nature-inspired) algorithms and metaheuristics among which SWARM AI is one of them.

Although SWARM AI has been applied for many problems for the last 30 years that include network routing, travel sales man problem, vehicle routing problem, network routing and so on [23-29], its application for transportation problem in general and bus transit system in particular is very limited [22] . In this research we feel that a transportation problem, in our case public bus transit reliability monitoring, has important characteristics that makes it attractive for the application of SWARM AI. These characteristics includes the environment in which the public bus operate is inherently distributed with many independent entities with their own objectives that can easily mapped to multi-agent system. Moreover the environment is not only constantly changing (dynamic) but also it is changing in non deterministic way.

This approach has many advantage compared to the previous methods with deterministic mathematical modeling that treat the problem holistically. The first advantage is scalability, because if there is only a single controller, it will be computational and communicational bottleneck. But if the architecture and framework of the problem modeling is distributed like in this case, every agent in the system share the computational and communicational burden equally, making its scalability easy. Some of the other advantage of our system is its robustness, self organizing, less costly, and so on as described in chapter 3.

1.4 Problem Statement

As finding a way of monitoring the reliability of bus transit system for avoiding bus bunching problem centrally and using analytical model is difficult, an interdisciplinary method that use nature inspired metaheuristics using multi-agent framework as a simulation paradigm can be a viable solution. This means that given the different entities (agents) of the system and their interaction, the problem is how one can design a coordination mechanism between these agent in a multi-agent simulation framework that use SWARM AI metaheuristics and problem specific heuristics (i.e. bus holding) so that the collective result of their un-programmed interaction (the emergent behavior of such interaction) result in avoidance of bus bunching (improvement of their reliability in general).

For our purpose we use bus holding strategy as a problem domain specific heuristic to be used in combination with the SWARM AI metaheuristics. This is because bus holding heuristics is the most commonly used method of countering bus operation irregularity resulted from stochastic nature of inter-stop trip time and uneven passenger demand distribution at different bus stops.

Our argument for using this approach instead of the traditional analytical, centralized deterministic mathematical methods can be summarized as follow:

- The bus reliability problem like congestion problem is emergent phenomena resulted from the un-programmed and stochastic interaction of different agents of the system such as bus agent, traffic light agent, other cars agents, passenger agents and so on. So we believe that for emergent problem type such as congestion and bus transit unreliability, emergent solution like SWARM AI solution can an effective solution.
- Moreover the bus transit system are not only dynamic but also change in a stochastic way which makes the applicability of SWARM AI solution whose problem solving power resides in its agents stochastic decision making and probabilistic based coordination between agents can be a viable problem-solution pairs
- And finally we believe that in order to design such coordination methods for addressing the reliability of bus transit system using SWARM AI, we use a multi-agent simulation modeling as it is mostly used in this context.

1.5 Objective/Aim of the Study

The main contributions of this research is to mitigate the problem of bus bunching (improving reliability) using SWARM AI metaheuristic and bus holding heuristic and test its feasibility using multi-agent simulation modeling. For the solution to be feasible, real time data from ITS is assumed to be available. More specifically, the objective of this research can be summarized as:

- To model the bus transit operation with multi-agent simulation that can be used as a test bed for measure of effectiveness of the proposed methods.
- To apply a SWARM AI (co-field based [46-54]) coordination mechanism for improving the reliability of bus transit system equipped with ITS.
- To implement and demonstrate the performance of the proposed method using NetLogo multi-agent programming IDE using real world data from number 31 of Ambessa city bus, Addis Ababa, Ethiopia.

1.6 Value of the Research

The value of this research can be seen from different angle such as

- For harnessing the huge real-time data that ITS makes available effectively and innovatively for improving the reliability of the bus transit system
- The multi-agent simulation can be used as a platform for studying the feasibility of equipping those which are not already equipped with ITS before making the large investment needed to deploy it.
- As SWARM AI, which is a branch of artificial intelligence, is a very young field, its field of application (class of problem on which it can be applied) is still growing and its potentiality is still in the process of discovery. More specifically its potential for solving a problem which is distributed, dynamic, and stochastic still a hot subject of recent researches area. Most transportation problem in general and transit system reliability control and regulation in particular belong to this class of problem and so that this research can be considered to be timely.
- Although simulating transportation process, including transit system, using multi-agent framework for modeling purpose is not new, there are few if any research that utilize this multi-agent simulation as an evaluation tools for testing the performance of different algorithms (metaheuristics). Specifically there are few literatures that utilize SWARM AI for improving transit system service reliability such as ant colony metaheuristics as control and optimization algorithms and multi-agent system as performance evaluation tools.

1.7 The Outline of the Thesis

The general outline of this thesis is shown in fig. 1.2. As already shown in the figure, in chapter one, we have discussed about the motivation, objective of the research. In this chapter we introduced the bus transit reliability problem and bus bunching problem which is an instance of bus transit unreliability problem. We have also introduced the methodology that we are going to use (i.e. SWARM AI). Finally we have discussed the value of the research, why it is so timely to do it now.

In the second chapter, we are going to review some researches which have something to do with our objectives. In this literature review chapter, we see past researches which are done on bus transit reliability problem and use similar methodology as we do. The third chapter gives general background about the SWARM AI methodology. The information in this chapter will gives us more or less enough information that we need for designing the proposed coordination mechanism in the chapter that follows it.

In the fourth chapter we are going to design the multi-agent simulation that we are going to use for iteratively design the proposed SWARM AI coordination mechanism and for testing and comparing the performance of the proposed metaheuristics with other methods. In the fifth chapter we are going to design the proposed methodology using the multi-agent simulation which is designed in chapter four. Here iterative method of software and metaheuristic implementation models such as TAPAS model is used. In chapter six we will evaluate the performance of the designed metaheuristics and compare it with other comparable methods. The simulation experiment that will be carried out in this chapter will be calibrated with relevant data from other researches done on a bus route line 31(from legehar to shiromeda) of Ambessa Awtobis organization found in Addis Ababa , Ethiopia. Finally in chapter 7 we will conclude and give future recommendation about our topic.

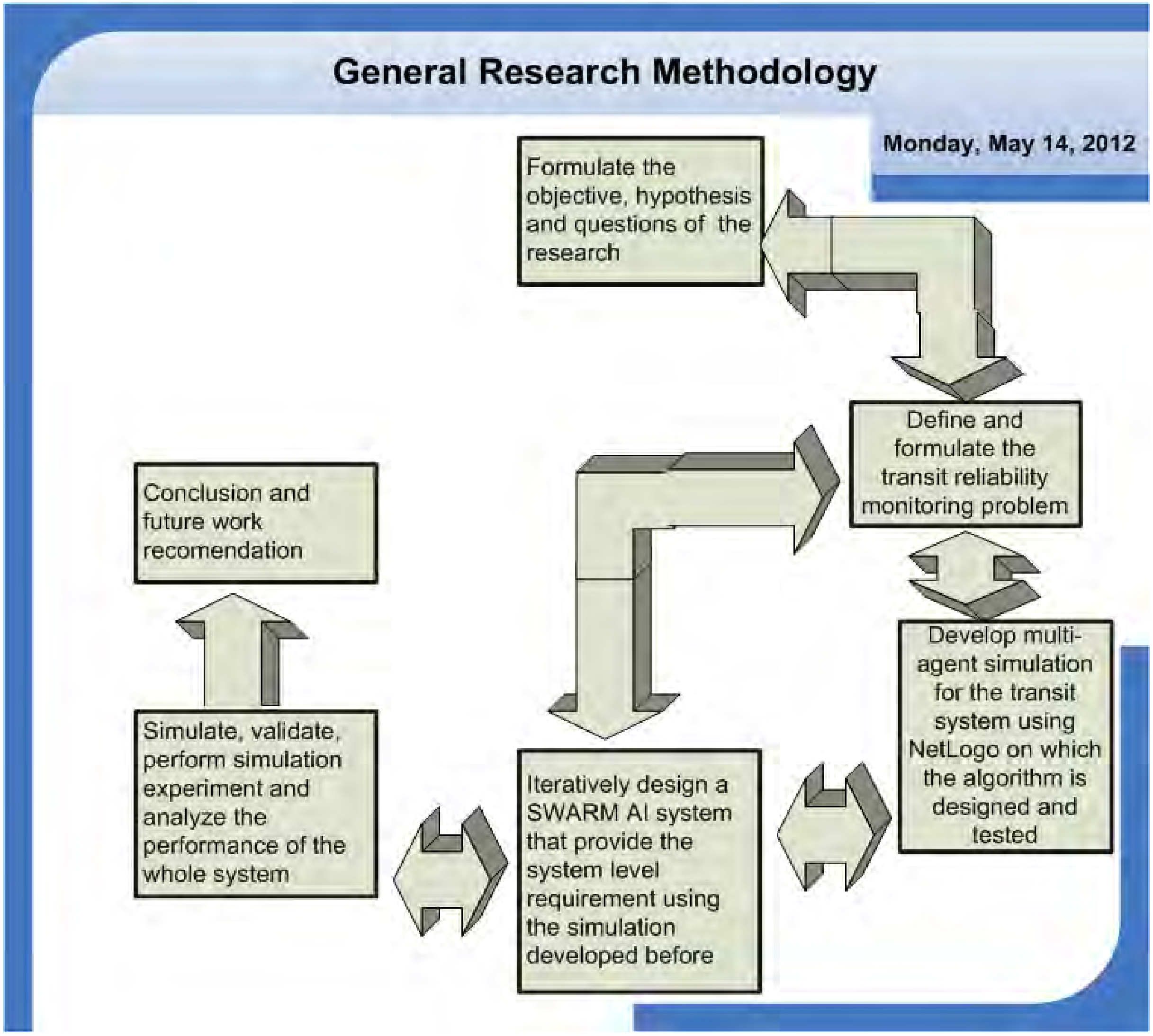


Figure 1.2 The general research methods

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the last chapter we have introduced the motivation and objective of this research which is to model and find an alternative way of improving the reliability of bus transit system operation problem using an interdisciplinary approach called SWARM AI metaheuristics. In this chapter we will review past research works that are related to our objectives.

2.2 Bus Transit Reliability Problem

Researches dealing on bus transit service reliability problem in general and bus bunching problem in particular have been done for almost 50 years e.g. [30]. But because of the nature of its difficulty however, it is still an active area of research. In this section we will see some of the past researches done on this topic.

It is shown by different researches that bus transit systems are not serving the people reliably although most users rate the reliability of this system as the most important factor [2-3]. In a UK practical-transit guide reports that, passengers' perception of local bus services buts service reliability as the number one factor in their importance even twice as importance as the next factor i.e service frequency. In order to cope up with this problem, service providers may employ a variety of operations control techniques to make sure that it operates properly. Among these, traffic signal priority [31] and bus holding [7-8][14] are the most used control techniques.

In the case of bus holding, a vehicle ready to be dispatched before it's planned scheduled or headway departure time may be held at a stop so that service regularity will be maintained. In case of traffic signal priorities, buses will get preferential treatment in timing of the traffic signal so that once the traffic signal is informed that a bus is approaching it, it will keep the green light until the bus pass it. But as traffic signal priority is out of this research scope, we will limit our discussion to only research works that use only the bus holding heuristic as a way of maintaining service regularity and improve service reliability as we will see below.

Put simply, bus holding is the process of intentionally delaying a vehicle at a station after passengers have alighted and boarded. Early works improving bus transit reliability problem using bus holding techniques mostly used analytical methods of modeling the system e.g. [32-33]. However as the environment in which the bus transit system is so complex with so many variable which are stochastic, distributed, dynamic and so on in nature, it is difficult to find mathematical model that take into consideration all these characteristics without committing either to complex mathematical expression which is difficult to understand and to use or too simple model with a lot of simplifying assumption which render the model useless for practical purpose. As a result, researches that use simulation method of modeling the system started to emerge [34-36].

However in order to do analytical and simulation modeling of the bus transit system operation, cost effective and reliable means of collecting performance measure of the system should be available. This has been the major problem as earlier research uses manual data-collection methods which is not only unreliable means of collecting data but also very costly to collect and process it.

In order to cope up with this problem, service providers start to equip themselves with automated data collection and communication technologies [5-6]. These service providers are collectively called intelligent transit system (ITS) and use apparatus such as APC, AVL, on-board computer and so on. As a result of all these efforts, serviced providers are able to harness huge amount of performance data that manual data collection method users cannot even dreamed of.

The problem with ITS is that, although this system enable the access of huge amount of performance data, the problem of how to effectively and efficiently utilize these data was not given proper attention [2]. This is because collecting large amount of data is not a solution by itself unless we use the data resourcefully. This problem has been beautifully expressed by [2] as "Overflow of data is as bad as underflow of data".

This is a theoretical problem in which many researches has been done and is being done. Some researches in this line of research has the objective of utilizing the huge data harnessed from ITS for developing more reliable simulation model that can be used for different purpose [34][37]. [34] for example used the automatically collect data from ITS to develop a simulation model that can be used for simulating different control measure for improving the reliability of the transit system such as bus holding. He used an event-based simulation paradigm and used Matlab as implementation toolkits.

The other use of the collected data from ITS which is related to our research is the use of these data for finding different ways of improving the reliability of bus transit system. Here the availability of this data rich ITS system means that we can afford to use simulation models that are data intensive such as multi-agent simulation, and use innovative problem solving methods from artificial intelligence which do not need traditional mathematical model. This is because multi-agent simulation is the major simulation paradigm on which different innovative problem solving methods from artificial intelligence are designed and tested.

In his call for research, [22] for example recommended the application SWARM artificial intelligence (SWARM AI) introduced in the last chapter, as suitable methods for approaching transportation problem in general and public city bus system in particular. His reason for this call is that these transportation systems do not lend themselves for conventional mathematical modeling as they are inherently complex system with a lot of entities (agents) distributed in stochastically dynamic environment. Other reason that we may add for using this method for bus transit system includes the availability of simulation paradigm suitable for designing and testing SWARM AI (i.e. multi-agent simulation modeling), availability of powerful personal computer for implementing and running the simulation and finally availability of programming IDE such as Netlogo [59], dedicated to multi-agent simulation and so on. As a result of such and similar effort, researches has been done that use methods which do not use formal mathematical models and that gives good enough solution with relatively short time but without any guarantee of existence and optimality of the solution which are called heuristic and metaheuristics.

2.3 Heuristics and Metaheuristics for Bus Transit Reliability Problem

In response to different research call that invite the application of SWARM AI for bus reliability problem, different researches that use different heuristic and metaheuristics for improving the bus transit reliability problem have emerged [13-22]. For example [13] use the real-time data from ITS to improve the reliability of the bus transit system. Their objective is to design a distributed coordination mechanism for solving the bus holding problem using a multi-agent negotiation policy and measure its performance using simulation. They perform different simulation experiment with four control strategies with the result that their method outperforms other comparable strategy such as even headway, schedule based and the default method without any holding control.

Other example of this type include [42] in which the bus holding problem is formulated as an optimization problem with passenger and operator cost has been used as a function to optimize. They use ant colony optimization (ACO) which is a brunch of SWARM AI. Their conclusion is that heuristic and metaheuristics is suitable for approaching the holding problem and recommend other researcher to try other related heuristics model. [43] Used genetic and greedy heuristic to solve the optimum number and location of holding points using simulation as a methodology. He claimed to have 11% of improvement from the base scenario but his method is an off-line optimization.

Finally [19] use a self-organizing method of regulating the operation of a single route bus transit system. They use an idea of anti-pheromone which is a sort of Stigmergic principles of ant colony optimization metaheuristics. The advantage to use this method is that as the demand of bus transit system change constantly throughout the day, this methods try to adapt to the change in demand by itself (is adaptive). Using multi-agent simulation, he showed that his methods outperform other comparable methods such as min-max method from his previous work (i.e. [20]). The drawback of this method is that no methodological issue is addressed as to how he came up with his solution. Moreover the method use a global variable (total number of passenger waiting in the system) which difficult to access practically and which is against the designing principle of locality of interaction in SWARM AI (including ant colony optimization).

2.4 Summary and Research Gap

From the literature review that we have done in this chapter; we identify that the application of innovative problem solving methods such as bio-inspired (nature-inspired) metaheuristics which have been being applied for data rich application such as internet application is not widely applied to ITS. Moreover although multi-agent simulation have been used as a modeling the operation of the bus transit system (ITS), its application for designing different optimization for improving the reliability using SWARM AI metaheuristics has not been widely done. In the next chapter we will see the general theoretical background that we need in the following chapters.

CHAPTER 3

METHODLOGY

3.1 Introduction

In the last chapter we have seen some past researches which are related to our objectives i.e. using the huge data that we get from ITS for finding way of improving the reliability of the bus transit system using SWARM AI coordination mechanism. This in turn contributes in combating the problem of vehicle congestion and atmospheric pollution. In this chapter we will give a theoretical background that we will need in the subsequent chapters.

3.1.1 Metaheuristics

Many optimization and control found in application are so difficult to find their solution using conventional mathematical methods within a reasonable time and computing resource that we usually resort to approximate solution that gives a good enough solution within a reasonable time. We call this method a heuristics (rule of tab) approach to problem solving. The word heuristic has its origin in the old Greek word heuriskein, which means the art of discovering new strategies (rules) to solve problems. They give us a near optimal solution within a reasonable computational time and resource. They often use problem specific knowledge to either build or improve the solution. Greedy and local search algorithms are example of commonly used heuristics algorithms.

Recently researcher from diverse fields started a new class of algorithms call metaheuristic which are a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. The suffix Meta, also a Greek word, means “upper level methodology”. The use of metaheuristics has significantly increased the ability of finding very high-quality solutions for hard problems. Unlike exact optimization algorithms, in metaheuristics, we are not guaranteed neither to find the global optimum solution nor its speed of convergence. And also unlike approximation algorithms, metaheuristics do not define how close the obtained solutions is from the optimal ones are, as we do not know the value of the global optimal solution in advance. Example of metaheuristics that are common includes Tabu search, simulated annealing, genetic algorithms, and Bio-inspired metaheuristics which is the subject of this research and which is discussed in the next section.

3.1.2 Bio-Inspired Metaheuristics

Social living creatures such as social insects (ants, bees, termites and so on), birds, fish and the like have managed to survived here on earth for millions years. Although individually considered they are small in size with primitive sensing and actuating capabilities, functioning as a group, it is observed that they can do so many complicated activities which are beyond the capability of each individual member.

For examples some species of foraging ants are known by their managing of finding the shortest path to their food source in spite of complexity of the environment in which they operate. Some species of termites are also known by their capability of building huge nests which are thousands fold bigger than their individual member size with a functionality of temperature and air flow regulation. Other social animal such as some species of fish and birds are also known by their complicated collective spatial pattern formation for the purpose of collectively avoiding predators and even for collectively search for food (hunting) functionality as illustrated in the following figures. The following figure and table illustrate some example of social animals which are known by their collective problem solving.

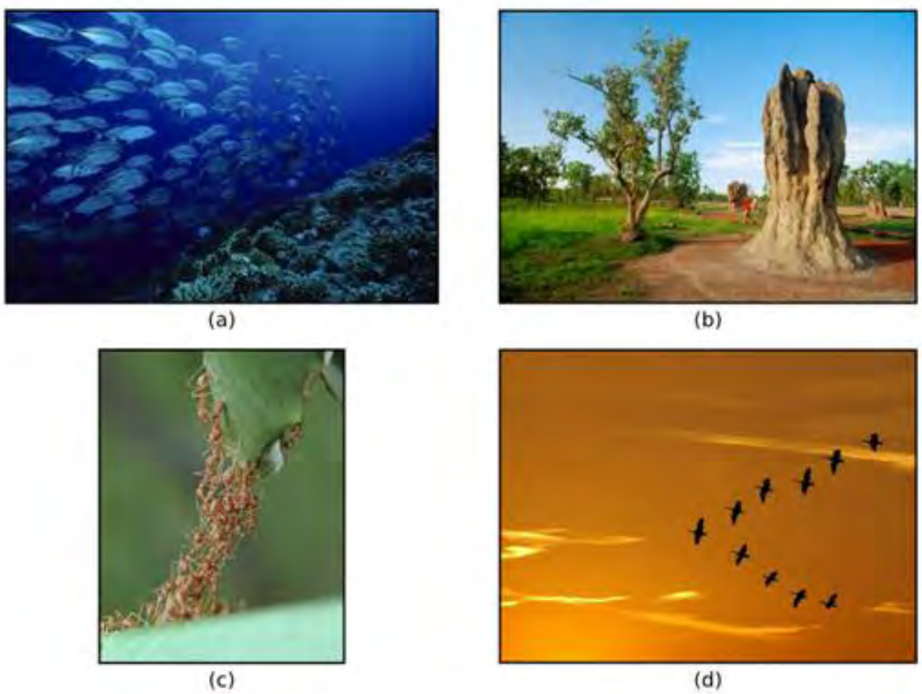


Figure 3.1. Example of social behavior in nature that are commonly studied by SWARM AI. a) fish b) termite nest c) ant collective transport d) bird flocking together All images taken from iStockphoto .com

Table 3.1 Some example of SWARMING behavior of living things

Swarming Behavior	Entities
Pattern Generation	Bacteria, Slime Mold
Path Formation	Ants
Nest Sorting	Ants
Cooperative Transport	Ants
Food Source Selection	Ants, Bees
Synchronization	Fireflies
Schooling	Fish
Flocking	Birds

The way these social animal solve their problem has special characteristics in that scientists specially from biology, computer science and artificial intelligence started to model and imitate them and start to apply to the problem that are difficult to solve by traditional mathematical methods. As a result of such effort, a collection of interaction based (coordination based) problem solving toolkits which are called SWARM artificial intelligence (SWARM AI) is born. SWARM AI is also know by different names such as Bio-inspired collective intelligence, Emergent self-organizing system, nature-inspired metaheuristic and so on. In this research the term SWARM AI, can refer for all terms that we mentioned before. And in the next section some concepts of SWARM AI metaheuristics will be discussed.

3.2 SWARM AI

SWARM AI is a branch of AI that has gained increasing popularity in recent years. The main idea of SWARM AI is to model and imitate the successful problem solving way of social living things (especially social insects) and apply the underlying principle to the solution of difficult problems which our society faces and which is difficult to solve using traditional mathematical methods. The sheer fact that these social insects have managed to survive here on earth for millions of year using these problem solving methods makes it worth imitating.

SWARM AI has formally been defined in many ways. [43] for example defines it as:

"Any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies."

As can be seen from the above definition, SWARM AI is a problem solving design tools that is taken from the social insects and other social living things, for the purpose of applying their principle to our problem which are similar in nature to that of these insects.

For our purpose, we can use the definition of SWARM AI adapted from [44]:

Definition: - SWARM AI is a field of AI that studies about the design of useful self-organizing system resulted from the emergence of system component through multiple local interaction of simple agent or individual system components.

3.2.1 Concepts of SWARM AI

As we have said before, SWARM AI is a collection of bio-inspired problem solving metaheuristics which takes its inspiration from different social life creatures such as ants, termites, fish, and birds and even from non-living natural phenomena such as from gas particle movements. To illustrate the concept of this SWARM AI, let as see ant foraging behavior [57] which is among the first phenomena to be modeled and used in the bio-inspired problem solving toolkit. This ant foraging example is the most cited illustrative example in SWARM AI literatures.

As shown in the fig.3.2, ants which are in the process of foraging their food face the following scenario. At the outset, every foraging ant moves randomly in search of food because it has no global information about the food location. While it is searching randomly it deposit a pheromone along its path. If any of the explorer ants come across the food sources, it returns to its nest using the same path that it used before depositing pheromone. As a result ants that use the shortest path become the first to reach the nest. What this in turn means is that, the path that this ant uses has the most concentrated pheromone which intern invite other ants to use it.

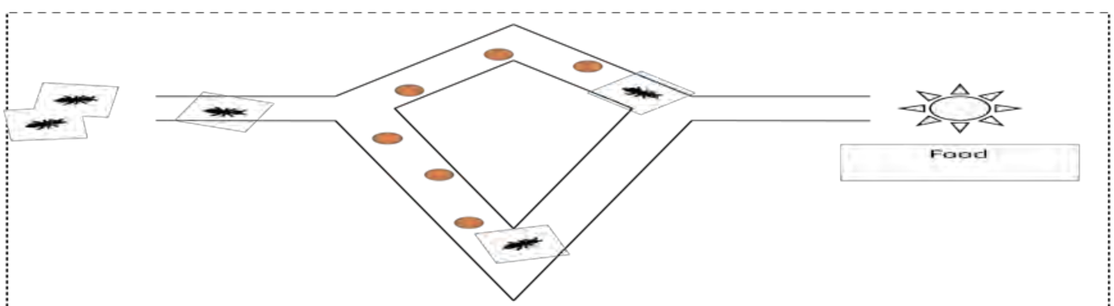


Figure 3.2 Ant foraging behavior

What this means is that the ant that randomly selected the path to the food source that has shorter distance will return very soon to the nest and the amount of pheromone deposited along its path will be greater than the longer path. This high concentration of pheromone in turn attracts other many ants in its way, increasing the pheromone even further. As a result in the next few iterations almost all the ants use this path and the colony as a whole is said to solve the shortest distance problem without centralized controller, without global information only using the environment as a means and medium of communication and coordination.

Although, from the above illustrating example, one can identify many concepts related with SWARM AI, the two most important concepts are Self-organization and Emergence which are described below.

Self-organization: Self-organization is a concept original found in physics and chemistry to describe the existence of macroscopic pattern resulted from process and interaction defined at micro specific levels [43]. But recently it is increasingly used to describe phenomena in SWARM AI the existence of high level properties (collective phenomena) resulting from the low level local interaction of the constituting agents such as insects, bees, and birds.

Emergence: SWARM AI systems typically have higher level properties that cannot be observed at the level of the individual elements, and that can be seen as a result of their interactions (more than the sum of their parts) [43]. These properties are said to be the mergence properties of the system. The source of these emergent properties can be traced back to the local interaction of the system components (agents). Emergence describes the appearance of structures at a higher level that are not explicitly represented in lower-level components. The reliance of swarming systems on locally available information makes it difficult for them to reason explicitly about higher-level structures, so emergence tends to be an important mechanism in swarming systems.

3.2.2 Advantage of SWARM AI

There are many advantages to this type of problem solving. The first advantage is that it is a distributed way of problem solving. This in turn means as in the centralized case, the time needed to collect global information by a single or few members of the system and solving the global solution if there is any and distributing the solution back to each member is minimized. The disadvantage of centralized control and optimization is that the central controller will be overloaded and become the communication and processing bottleneck of the whole system. As any system cannot be stronger than its weakest link, the performance of the whole system will only be as good as the central controller. This distribute problem solving is a useful frame work to apply to many problem that we are facing. Some systems and environment like public transit system is a distributed system by nature which constitutes many agents without a central controller (it is a multi-agent system).

The other advantage of this system is that it is robust. This means, for example, as all of ants use local information for their decision making, if any of them die, there is no much damage to the whole system, as any of them do not hold the information and computing resource that determine the fate of the whole system. This is unlike to the centralized way of problem solving in which if the central controller fails, the whole system will fail. But in the case of the distributed way of problem solving if any member of the system fail (die), other member will take over the process which means it is robust way of problem solving.

The third advantage of this distributed problem solving is that it is less costly. This means as each member of the system deals only local condition (local communication and computation) there is no need to have very expensive communication and computing power to the whole member of the system. This means the cost of building system component and maintaining them will significantly reduced.

3.2.3 General Design Principle of SWARM AI

Building self-organizing emergent solutions (another name for SWARM AI) implies achieving the required system level macroscopic behavior using multiple autonomous agents each able to make their own decisions. The agents cooperate using complex and often unpredictable (un programmed) interactions because at any moment each agent can autonomously decide and do any (inter)action with any other agent and with the environment.

And its problem-solving power mainly resides in the interactions and coordination, instead of in the internal reasoning of individual agents. Due to the characteristics of SWARM AI, designing microscopic activity based on macroscopic requirements or verifying macroscopic requirements from the microscopic system is hard and even impossible to derive analytically. This is because the problem solving power of SWARM AI is generally resides in the un-programmed and probabilistic local interaction of different agents which is not convenient to model it using mathematical methods.

For this reason, currently in the design of SWARM AI algorithms, there is no formal and definite step by step process that enables us to design the algorithms from start to end. The best thing that one can use to design interaction based solution such as SWARM AI and emergent self-organizing solution is to use patterns of successfully applied past solution for analogues problem (i.e. pattern oriented problem solving).

However there are some general guidelines to apply these solutions even though they are not specific enough for application of a given problem as shown in the fig. 3.3 and table 3.2 which is summarized from [44].

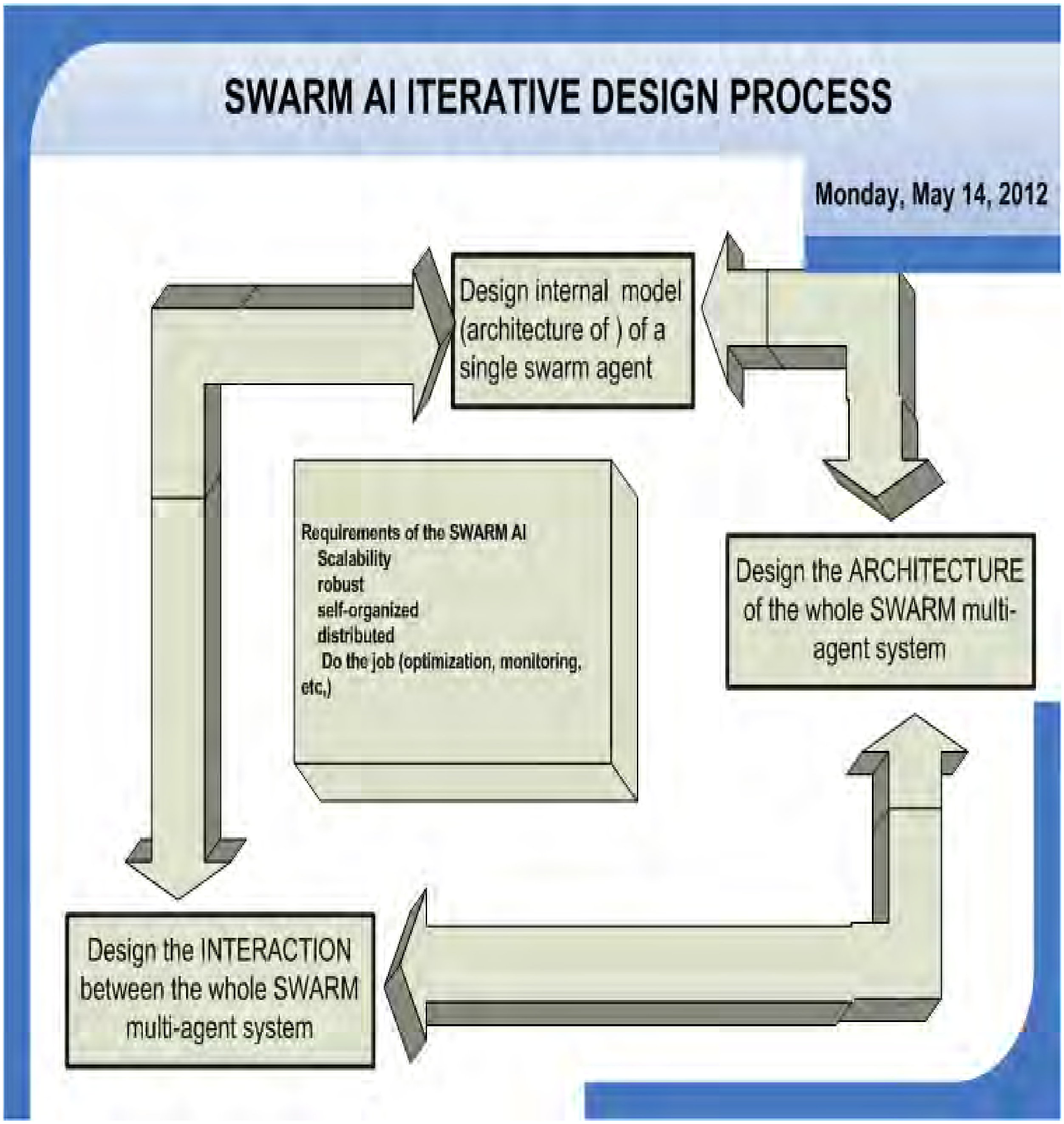


Figure 3.3 SWARM AI iterative design method

Table 3.2 General design principle of SWARM AI

	Coupling	Autocatalysis	Function
Agent architecture	• Keep agents small • Map agents to Entities, not Functions	• Diversify agents to keep flows going • Redundancy	• Generate behavioral diversity • Give agents access to a fitness measure • Provide a mechanism for selecting among alternative behaviors
System architecture	• Use a distributed environment • Use an active environment	• Think Flows rather than Transitions • Boost and Bound • Decentralization • Parallelism	• Modularity • Bottom Up Control
Interaction type	• Locality • Indirect Communication • Information Sharing	• Recursion, Self-Similarity • Feedback, Reinforcement • Information Sharing	• Randomization • Evolutionary Change • Forgetting

What these table and figure says is that, in the design of SWARM AI metaheuristics, one may start the process by designing the one of the components of the model (e.g. Agents, Architecture and interaction), then iteratively include other components as the need arises as shown in fig. 3.3 Then after a working prototype is done, we cross check weather some of the requirements of the model which is shown in table 3.2 is met. And if not met, we do the full cycle of the process shown in fig. 3.3 again.

3.2.4 Why Multi-Agent Simulation Modeling of Designing SWARM AI Solution

As Problem domains that require the application of SWARM AI solution is difficult to model using conventional mathematical model, other types of modeling paradigm is needed. Multi-agent simulation modeling is usually used for this purpose as it does not require a formal mathematical modeling. And also as we see in the last section, currently in the design of SWARM AI solution, there is only general guidelines which are not much help is we are going to go through each steps of designing SWARM AI solution.

Moreover as we have seen before, the problem solving power of SWARM AI solution resides in the un-programmed, probabilistic interaction of the constituent agents which is difficult to foresee it without trying to reproduce it using computer simulation. Multi-agent simulation is the simulation paradigm suitable for this purpose as it enables to model each interacting entities as a simulation agent.

The other logic for designing SWARM AI solution using multi-agent simulation is the idea of emergence. This is because designing emergent phenomena requires try and error procedure as the macroscopic desired system requirement is difficult to forecast from the behavior of the individual agents. Multi-agent simulation modeling fill this gap by enabling designers to model the solution using multi-agent computer simulation modeling and play with it until the desired micro-macro mapping is found.

As a result of these and other reasons, multi-agent simulation modeling is used as a main programming paradigm in the design of SWARM AI solution and that is why we use multi-agent simulation modeling for designing the coordination mechanism using SWARM AI methods.

3.2.5 Application of SWARM AI

SWARM AI is a suitable solution for a problem domain that has the characteristics of having a discreet component that can be mapped to agents which are resource constrained autonomous entities, with distributed, stochastically dynamic environment. There should also be a decentralization of information in the environment. Below we will see some of the characteristics of problem domains that can benefits from an application SWARM AI solution some of which are described in [45].

Discreetness: SWARM AI is suitable for application domain of which there are distributed entities (actors or agents) with their own individual set of objectives and capability (actuating or sensing capability). This is because SWARM AI is a population based metaheuristics where a lot of agents participate in the emergent construction of a solution resulted from their local interaction. More over the decision variables will be suitable for SWARM AI application if they are discrete (choose able from a discrete set of variables as for example choosing from taking different path a the case of graph structure of some sort).

Resource constrained: As SWARM AI is an approximation algorithm of some sort, it doesn't make sense to use it to a problem domain without resource constraint such as processing time, processing power (speed), memory and communication bandwidth. This is because if the problem domain is not constraint, it is justifiable to use brute force (exact method) method of finding solution if it has infinite time to do so.

Distributedness: It sensible to apply SWARM AI for application domain where the agents are distributed (scattered) on a distributed graph of some sort. This is because as SWARM AI uses a localized communication of some sort, we have to have some space to localize them (enabling agents to know who is local to it and who is not).

Decentralized: It doesn't make sense to apply SWARM AI for application that requires centralized controllers because in SWARM AI resource (communication bandwidth) is so constrained that the communication bandwidth between each agent and the central controller become a bottleneck for the system's performance. In SWARM AI the overall control of the whole system is assumed to be the emergent phenomena of distributed communication and coordination resulted from their interaction.

Dynamic: If the system of application does not change or change slowly or even rapidly in a determined way then application of centralized solution method of some sort is enough for its application. In this case it is not recommended to apply SWARM AI for these types of application. But If the system change rapidly, in obscured way (unpredictably) and the rate of change outpace the communication speed between each agent and a centralized control, the SWARM AI will be a recommended choice for its application. This means we localized and empower each agent to respond to change in its locality so that the system responds to a change not only to rapidly changing problem domain but also changing in a nondeterministic (stochastic) way.

3.3 Why SWARM AI for Reliability Problem of Public Bus

At this stage we can see why SWARM AI is appropriate for application in a transportation system like city bus transit reliability control and monitoring system. Firstly we can see that transit system is composed of multi-agent system (discrete system component) such as passenger, bus, other cars and so on [13]. We can also see entities such as bus stops, roads and the like as artificial agents with their own objectives and capability. We can see for example the road as an agent with the objective of transferring maximum number of cars per hours and with the capability of measuring its performance.

More over bus transit system as part of any transportation system is a distributed system (distributed along a two dimensional road structure) so that it can be safely can be modeled as a graph of some sort and is suitable for SWARM AI. More over as it is not an application of some hard real-time system; it is likely to be a resource constrained application. This means we cannot expect the operator to provide infinite computational and communication resource for using exact and brute force method to monitor its reliability.

Finally as bus transit system is changing not only rapidly but also stochastically, using centralized control system for its reliability control should not be an option. It is better to use some sort of distributed control system that explicitly model the uncoordinated nature of agents objective and capability and to empower each system agents like bus and bus system to respond to changes in the vicinity of their locality. In this case SWARM AI can be a good candidate for its application.

3.4 SUMMARY

In this chapter we have seen some background about SWARM AI methods that we need in the design of coordination mechanism for improving the reliability of the bus transit system. We have seen about the concept and challenge of designing SWARM AI, the advantage of using SWARM AI, the problem domain where it can be suitable for application and so on. Finally we have seen why bus transit reliability problem is one of the many problem domains suitable for the application of SWARM AI. In the next chapter we are going to design and implement the multi-agent simulation model of the bus transit system that we are going to use in the following chapters.

CHAPTER 4

Multi-Agent Simulation Model Development

4.1 Introduction

In the last chapter we have seen conceptual knowledge about the methodology we will use in this chapter (SWARM AI) and to the problem instance (bus transit reliability problem) so that it will help us what we will do in this and the following chapters. In this chapter we are going to design and implement the multi-agent simulation model of the bus transit system. We will use the model not only as a test bed platform to evaluate and test the performance of the designed model but also is used as tool to iteratively design the SWARM AI metaheuristics.

4.2 Multi-Agent Simulation Modeling of the Bus Transit Operation

Problem Formulation for multi-agent simulation

Driven by the ever increasing consciousness of air pollution that result in global warming and increase in city vehicle congestion which result in wastage of fuel and personal time, governments are giving big attention for the better provision of transit system such as bus transit and train transit. However in order to attract and entice private vehicle users to use transit system such as bus transit system, the reliability of the bus transit system should be improved.

In order to address this problem, bus transit operators are starting to equip their vehicles with modern communication electronics technology that enable them to improve their service reliability. However unless a method of harnessing these technology effectively is developed, the large amount of real-time data will be wasted and may cause the operators additional maintenance cost.

One approach of formulating the effective utilization of the real-time data increasingly available from ITS is as a coordination problem in which different agents of the bus transit system cooperatively solve the reliability problem using the real-time data. And one approach to the solution of the coordination problem is to use one instance of bio-inspired metaheuristics called SWARM AI. And the purpose of the following multi-agent simulation model design is to use it as a design platform for iteratively design the SWARM AI metaheuristics to solve the coordination problem of bus transit system using real-time data available from ITS so that the reliability of the system will improve. The multi-agent simulation model is also used as a performance evaluation of the designed metaheuristics and compared them with other comparable methods of improving the reliability (headway regulation) of bus transit system of single route.

Therefore from the description of the above problem definition, the objective of the following multi-agent simulation modeling is to develop an agent based model that can be used to design SWARM AI metaheuristics iteratively to solve the cooperation problem bus transit system using real-time data from available from ITS so that the reliability of the bus transit system will improve. Moreover the model can also be used to evaluate and compare the performance the designed metaheuristic and compare them with other comparable methods of improving the reliability of the bus transit system.

4.3 Multi-Agent Simulation Model Design

Although there is no universally defined method of developing multi-agent simulation models, there are some suggestions as how to go developing it methodologically. One widely accepted method of approaching this problem is to start the design selecting the one of the components of multi-agent simulation model (Agents, Architecture and interaction), then iteratively include other components as the need arises as in the next figure.

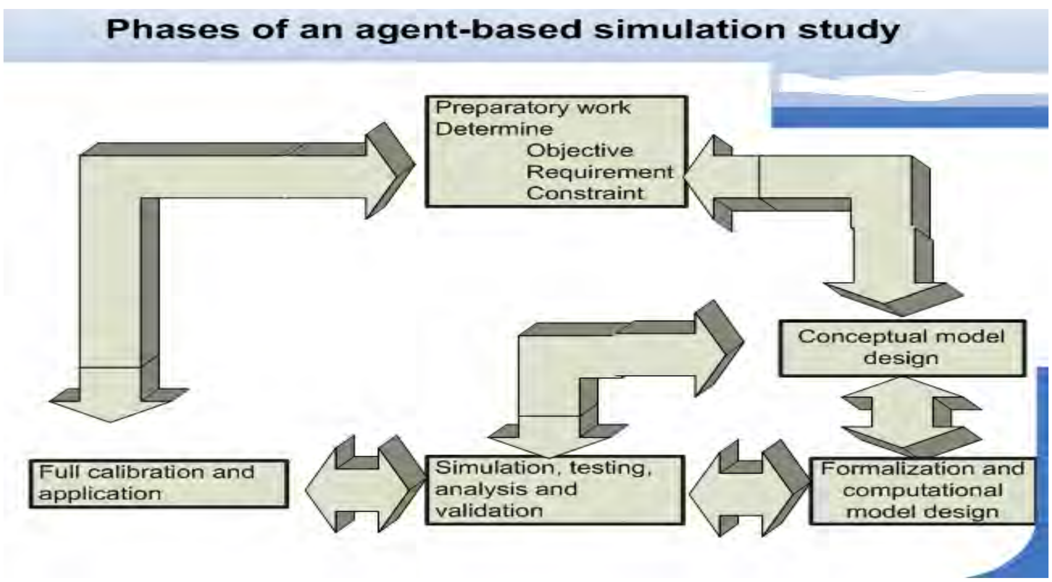


Figure 4.1.Multi-agent simulation model design

What this means is that, first we start the design process first by defining for example the agents of the model and then iteratively add interaction and architectural (environmental) components as the need arises. In the next section we are going to start the multi-agent simulation model using the agent driven approach then add other components in the following iteration.

4.4 Agent-Driven Model Design

Observing the bus transit system operation one can identify the following as agents of the model.

- Bus agent
- Bus stop agents
- Passenger agents
- Road-segment agents and so on

But the kind and number of agents can only limited by our imagination and the objective of the modeling as there is no rule that limit it. Then based on the objective of the modeling, the following agents and their variables are identified which is shown in the next table.

Table 4.1 Agents and their associatedactions and variable in the bus transit system

Agent	Associated variables	Associated actions
Bus agent	• Capacity • Speed limit • Boarding and alighting time per passenger • Etc	• Moving • Boarding passenger • Alighting passenger • Hold bus
Passenger agent	• Distribution of arrival rate per stop • Arrival time • Alighting time • Trip time • Holding time	• Arrival to bus stop • board on bus • Alight from bus

Bus stop	• Passenger arrival rate • Headway mean and variance at stop • Arrival time of current bus • Arrival of load • Arrival time of current bus • Departing time of current bus • Holding time of current bus • Dwell time	• Simulation modeling actions (generating passengers, taking performance tatistics) • Metaheuristics (co-fields) specific actions (taking care of co-field dynamics generating, propagating and terminatin fields)
Bus route	• Number and location of stops • Distribution of passenger arrival rate • etc	• Taking performance evaluation of the route, • Take care of scheduling of the bus transit system
Road segment agent	• Road segment trip time (distribution type) • Mean and deviation of trip time	• Connecting bus stop across the bus route

4.5 Architecture Design (Conceptual Design)

The architectural component of the multi-agent simulation model answers the question of "who talk with whom, when and where" [45]. The other component of the multi-agent model (interaction) on which the problem solving power of the model of SWARM AI resides answer the question of "Why, how and what they talk about" [45]. As one can design as many possible architecture as once imagination allow and there is no correct way of designing it, we develop the following interaction table which is one is shown in the next table as a starting brainstorming idea for finding the possible multi-agent architecture of the bus transit system operation that suit our objective.

Table 4.2. Interaction table for conceptual design bus transit system

Agents	BUS	PASSENGER	BUS STOP	ROAD SEGMENT	BUS ROUTE
BUS	-----	Passenger	Bus stop at	Bus move	Bus serve

		ride bus	bus stop	along the road	route
PASSENGER	Passenger ride bus	-------	Passenger wait, board and alight at bus stop	---	Passenger select a rout to use
BUS STOP	Bus stop at bus stop	Passenger wait, board and alight at bus stop	------	Bus stop spaced along the road	Bus stop serve different route
ROAD SEGMENT	Bus move along the road	-----		-----	
BUS ROUTE	Bus serve route	Passenger select a rout to use	Bus stop serve different route	Bus route is a collection of road segment	----

And from the interaction table of the above table, the conceptual domain model of the bus transit system is derived.

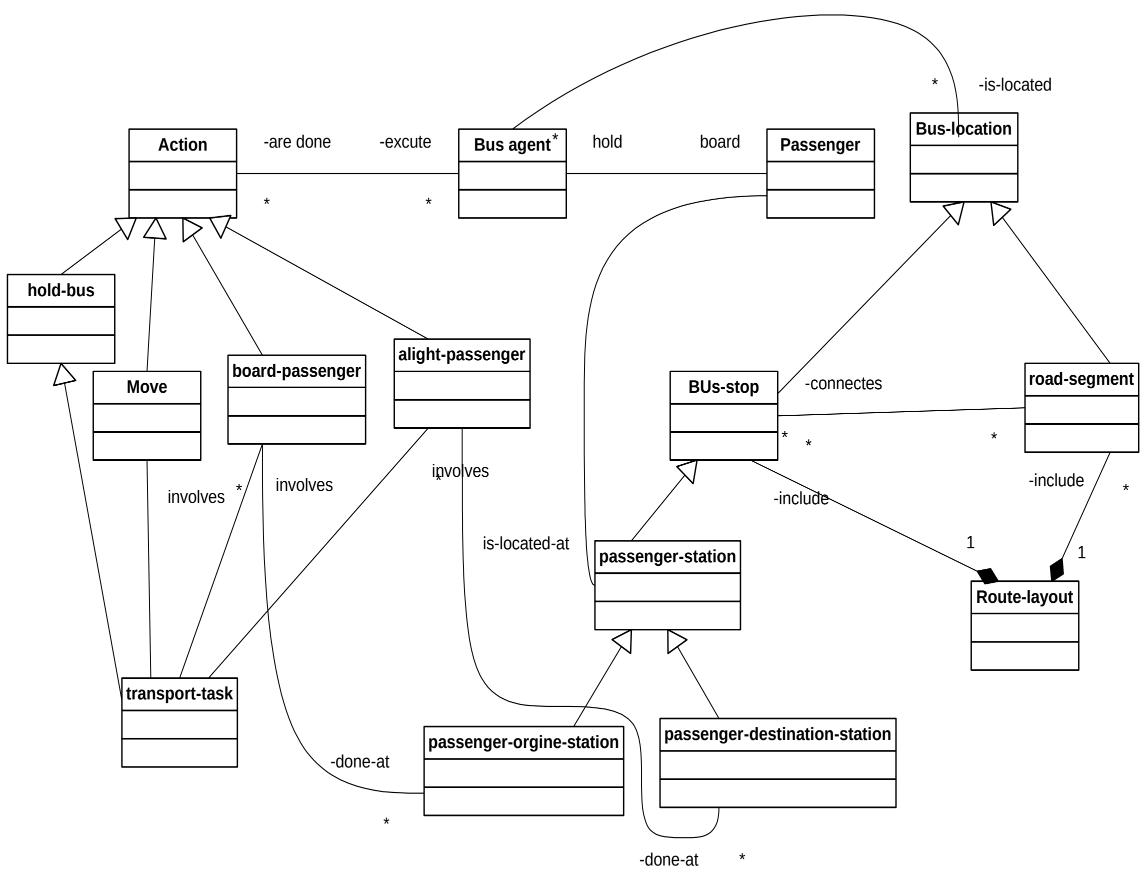


Figure 4.2 Domain model of the bus transit operation

4.6 Software Development Process Model

The software development methodology that we are going to use for implementing the conceptual model is rapid iterative prototyping. This methodology involves making (choosing) a working functional prototype, then iteratively evaluating and evolving it until it fulfils all our requirements. It allows functionality to be added in a systematic manner to a working version of the system, and allows requirements to be assessed at every stage of the development process. The effort to systematize such iterative methods leads to the development of iterative models that includes KISS model ("keep it simple stupid" model), KIDS model ("Keep it Descriptive, Stupid"), TAPAS model ("Take a previous model, add something")[46]. For our purpose we use a TAPAS model ("Take a previous model, add something") because it has advantage over other models especially when the multi-agent simulation is part of scientific research. When developing multi-agent simulation whose objective is to show the performance of some algorithm, we should implement not only our method but also other comparative methods that we are going to compare with. This is not only takes unnecessary extra time, but also is very error prone. For this reason, we are going to use TAPAS iterative multi-agent simulation model development methods whose pseudo code is shown in the table below.

Table 4.3 TAPAS iterative model development

1. Select an appropriate existing model M
2. If M is not implemented, implement it and validate it using available data from M
3. Add new additional aspects to produce M
4. Test and validate M_{add}
    a. if sufficient ready
    b. else go back to step 3 or step 1

We use the TAPAS iterative multi-agent simulation model for two purpose; for the development of multi-agent simulation of the bus transit system on which the SWARM AI metaheuristics is designed and for iteratively design the SWARM AI coordination mechanism which will help to improve the reliability of the bus transit reliability problem using the huge real-time data from ITS.

In this chapter we are going to use TAPAS model for designing and implementing the multi-agent simulation model taking the initial starting model from our re-implementation of [20] Metro NetLogo model and iteratively adapt to our requirements which will be shown in the next section.

4.7 User Analysis & Use Cases

As already pointed out, the objective of developing this multi-agent simulation models is twofold. First to use it as a test bed for evaluating the performance the SWARM AI coordination metaheuristics that we are going to design in the next chapter. And the Second use of this model is to use it as a framework on which the SWARM AI coordination mechanism is iteratively designed. That is why we have to design and implement the starting multi-agent simulation model of the bus transit simulation system before starting the design of the proposed SWARM AI coordination methods.

As a result, it is expected that the user of this multi-agent simulation is the SWARM AI designer and the simulation experimenter that compare the performance of the designed metaheuristics with other comparable methods. Moreover the developers of this multi-agent simulation are group of people that have expertise at least in three fields; i.e.

- Problem Domain expert (and also called thematicians, in our case bus transit system expert)
- Computer programming expert (that change the conceptual model into computer executable program)
- Modelers (operational research experts, develop conceptual model)

However in our research the same person is used both as a developer and a user. That is why iterative development of the simulation is recommended because the same person cannot reasonably be an expert in all these domains at the same time. However this is the main challenge that the researcher in this interdisciplinary research will face and he should do his best to address this challenge.

Taking the above description into consideration, the following use case diagram of the multi-agent simulation system is developed.

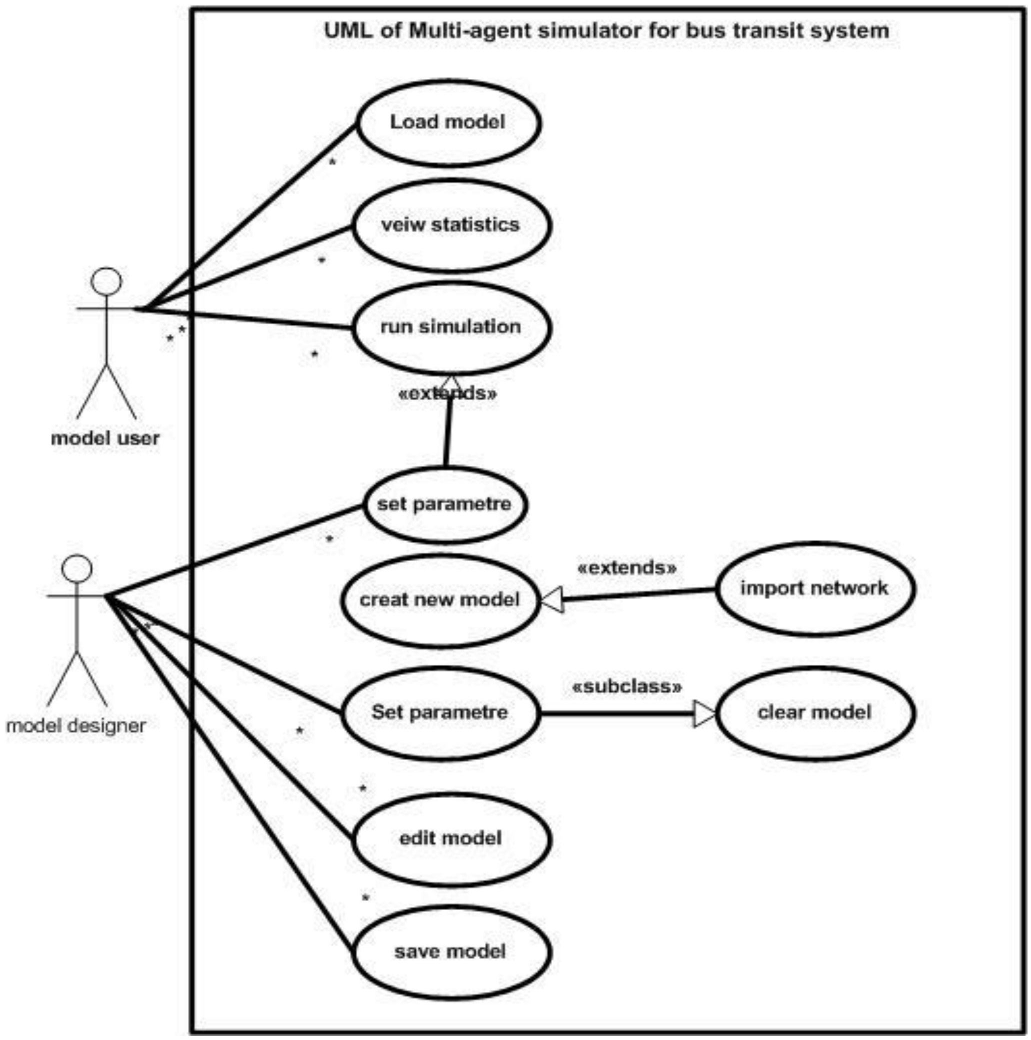


Figure 4.3 Use case diagram of the multi-agent simulation model of bus transit system.

Although all of the features which are shown in the above use case diagram is not necessary, it can give us a good starting point for the development of the multi-agent simulation we are going to use the following chapters. The functionalities given in the use case diagram is self-evident and, the following user requirements can be extracted.

User requirement

User Requirements

1. The system must provide graphical model creation/editing utilities.
2. The system must allow the user to load/save the current state of the model from/to a file.
3. The system must allow the user to alter simulation parameters, before and during a simulation run.

4. The system must provide a visualization of the current state of the simulation/model, and animate this during simulation runs.
5. The system must show the simulation statistics graphically using the implemented indicators.
6. The system must deliver realistic (verified) microscopic and macroscopic traffic behavior in both urban and motorway scenarios.
7. The system must be able to simulate a period of at least 30 minutes in simulation time.
8. The system must run at a speed equal to or greater than real-time.
9. The system must simulate the following additional features: queue spill back, HOV lanes and coordinated traffic signals.
10. The system must implement the following indicators: travel time, congestion level and AVO (average vehicle occupancy; for evaluating HOV lanes).
11. The system must be able to model a length of 2-lane road in excess of 1.5km (the length of road of interest in the study area) and be able to simulate the appropriate number of vehicles on this road.

4.8 Integrated Development Environment (IDE)

In order to implement to change a conceptual modeling of a multi-agent simulation model, we have to be selective in choosing a programming integrated development environment. This is because the demand this modeling paradigm put on the capabilities of IDE is very high. These capability demands includes ease of use (user friendliness), output reporting, ease of debugging, statistical support, rich documentation, application specific modules and so on.

There are varies agent modeling IDE available for both commercial and educational purpose. Some these IDE includes NetLogo, Swarm, Repast, Cybele, JAS and MASON and so on. Although all of them have their own strength, each of them are developed taking special objective in mind. However for our purpose we choose NetLogo because it is developed for modeling emergent and self-organizing phenomena by which the local interaction of mobile agents leads to the emergent of desire system level behaviors.

Moreover, as our research objective has a focus of modeling and design of SWARM AI metaheuristics, and has fewer things to do with the programming issue, NetLogo [39] is a good choice because it abstracts away many of the programming effort if we had to program using other low level language such as C++, JAVA and so on. For this reason we selected NetLogo, which is introduced in the next section, for our purpose.

- **NetLogo**

NetLogo[36] is an agent-based modeling environment designed for simulating complex natural and social phenomena. It is written in Java and is therefore cross-platform, it is freeware, and has a large friendly user group. It enables the user to model any number of agents in a variable-size environment using a simple programming language derived from Logo. It is designed for use by students and researchers to explore the behavior of programmed agents under varying conditions. It follows the philosophy of 'low threshold, no ceiling', meaning that new users should find it easy to get started, but advanced users should not find it limiting.

NetLogo comes with extensive documentation, tutorials, and over 300 sample models that demonstrate all aspects of the tool. Models can be saved as Java Applets and run on web pages by any user with the Java Virtual Machine installed. It is possible to view the current state of the environment and agents in 2D or 3D, and agents can be given any vector shape to display their type. Commands can be in the form of procedures that are called by buttons on the interface, or entered directly in the command console on the main panel.

NetLogo also has a well-designed graphical interface and interface builder in one that allows the novice and expert alike to run, alter and develop multi-agent models with ease. It provides many built-in widgets to alter simulation parameters at runtime, including sliders, buttons, and drop-down menus, and allows output in the form of graphs and variable monitors. Simulation time is measured in discrete 'ticks', and simulation speed can be adjusted by a slider above the display. It also has many add on tools to do scientific research such as BehaviorSpace which enable researcher to do scientifically valid simulation experiments on any agent models.

4.9 Class Diagram

As pointed before, the simulation programming method chosen for this research is a TAPAS iterative development whose pseudo code is given in table . Using

- NetLogo model developed by [20] as a starting model,
- The user requirement developed
- The domain model developed and
- The inbuilt agent of NetLogo (i.e. turtle, observer, and link agent)

and with some debugging, adaptation and so on a working multi-agent simulation model with the following class diagram is developed. The next figure shows the NetLogo interface implementation of the model. This model has many feature which include visualization, data input (number of bus, number of bus stop, passenger arrival rate for each direction of the rout), and data output (graph, counter, visualization, and so on). In chapter 6 we will calibrate the model constraint by accessibility of data using a case study from Line number 31 of Ambessa city bus organization in Addis Ababa, capital city of Ethiopia.

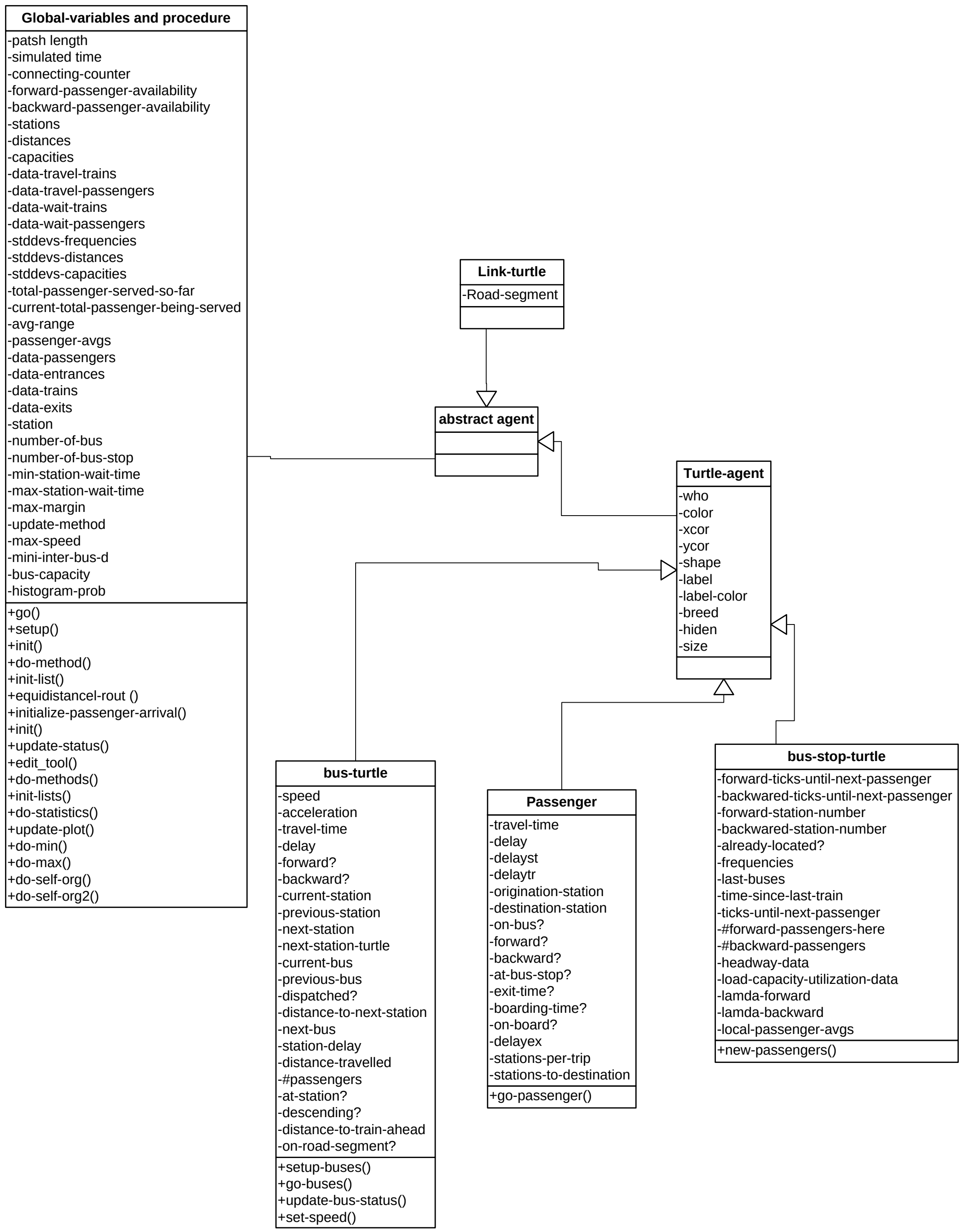
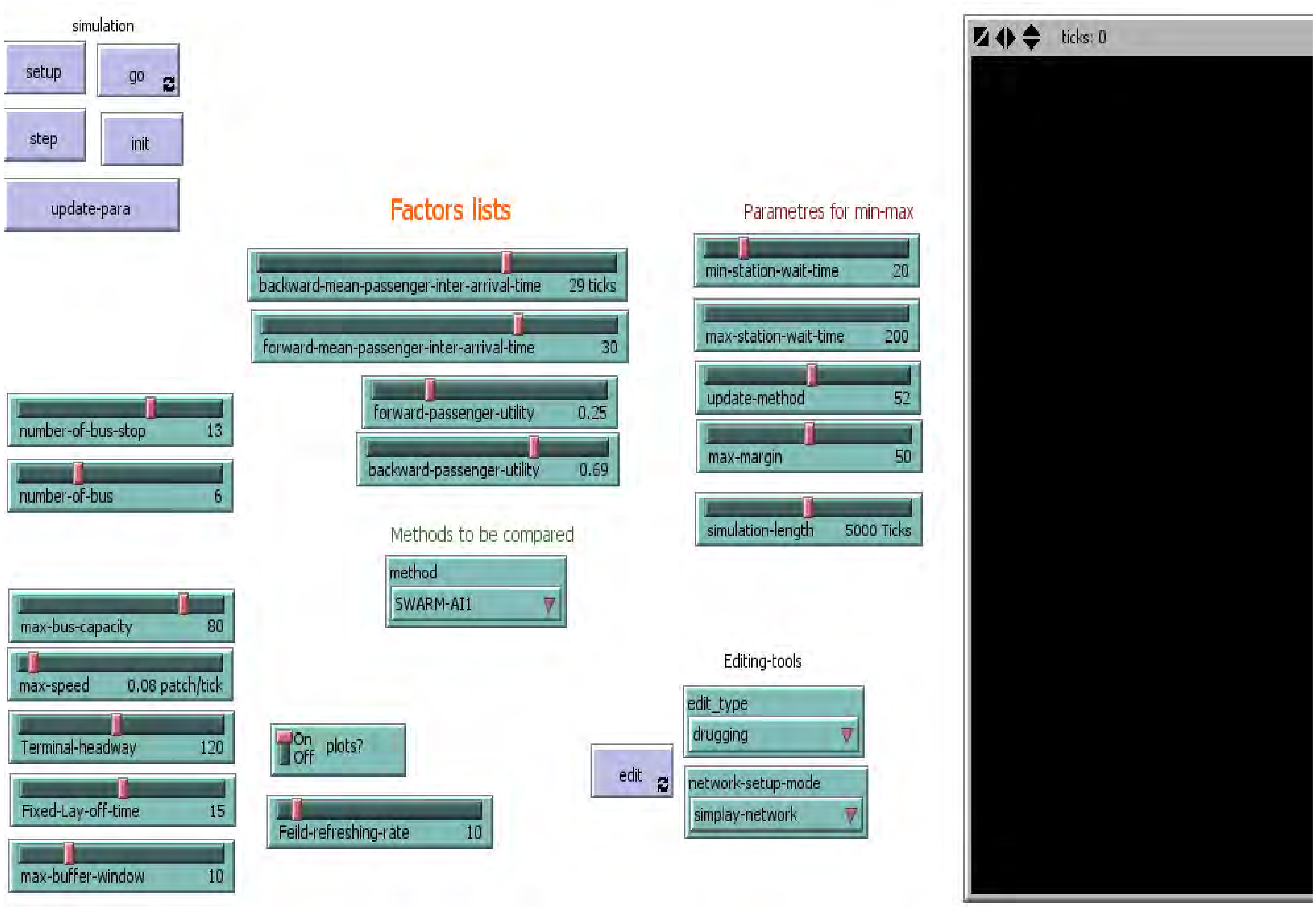
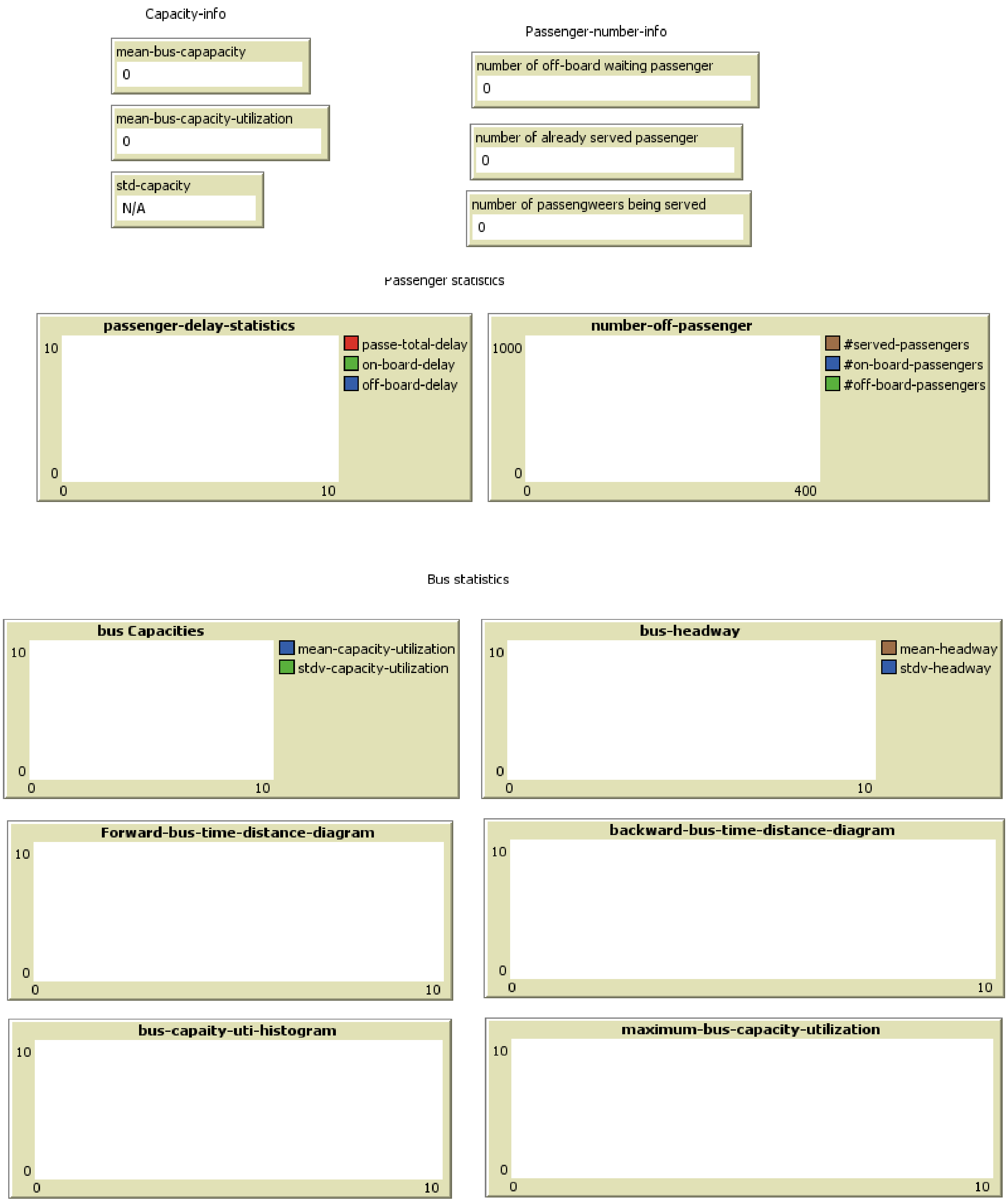


Figure 4.4. Class diagram of multi-agent simulation model of bus transit operation of single rout.

It will be used as a framework and agent architecture on which the SWARM AI will be designed and on which its performance is evaluated.



a) Input interface



b) Output interface

Figure 4.5 Graphical interface of the multi-agent simulation model of bus transit system of a single route a. Input interface, b. Output interface

4.10 Summary

In this chapter we have seen how we are able to design and implement the multi-agent simulation model of the bus transit system of a single rout. This model will be used as an architectural framework on which a SWARM AI metaheuristics will be iteratively designed in the next chapter and on which the performance evaluation of the proposed method will be tested in chapter 6. In the next chapter we are going to design the SWARM AI metaheuristics using TAPAS iterative design methods and the multi-agent simulation methods designed in this chapter.

Chapter 5

Design of the Proposed SWARM AI Coordination Methods

5.1 Introduction

In the last chapter we have seen how we have developed the multi-agent simulation model that we are going to use both for designing of the required coordination mechanism using SWARM AI and for testing its performance. The developed simulation model encapsulates the agent and the architecture of the SWARM AI system whose interaction space will be designed in this chapter. The interaction of the system, whose problem solving power of the SWARM AI resides, will be designed as a coordination mechanism between the bus transit systems so that the reliability of the system will be improved.

5.2 Interaction Design

The interaction of the multi-agent simulation is the most important components of the SWARM AI model. This is because most interaction based solution, such as SWARM AI which usually are designed and evaluated using multi-agent simulation model, have their solution embedded in the interaction of the constituent agents. That is, the problem solving power of interaction based solution, such as SWARM AI metaheuristics, is mainly found in the interaction and coordination of the agents and this interaction is modeled using the interaction part of the multi-agent modeling.

The design of the interaction part of the model has two components, the first is the natural interaction of agents in the original target system (bus transit operation), and the second is the artificially designed interaction based on some methods (SWARM AI in our case) so that the problem of the original system will be addressed (i.e. reliability problem in our case).

As the natural interaction of the target model was addressed in the previous chapter, while we are designing the multi-agent simulation model of the system, which can also be used as a starting point for the design of the agents and architecture of the system, in the following section we are going to design the artificial interaction that address the coordination problem of bus transit system so that its reliability will be improved. However before designing the artificial interaction of the system, we have to select the coordination mechanism which is the subject of the next section.

5.2.1 Selection of Coordination Mechanism

As already stated before, the problem that we are going to tackle is how to adaptively orchestrate in a decentralized way the spatial movement of bus agents so that a global pattern is formed which improve the reliability of the bus transit system. The coordination mechanism has to be robust and flexible in the face of frequent passenger demand changes in the environments in which the bus transit system operate.

As this is a distributed motion coordination problem with the intention of pattern formation by the bus agents, we choose a Co-field coordination mechanism [47-53] because according to the pattern description given by [54-56], which is shown in the table below, it has already proved its effectiveness in such type of problem. In the next section we are going to see about a Co-field coordination methods.

Table 5.1: Co-field Pattern description format

Pattern name	Gradient Fields (co-fields) [47-53])
Context/ applicability	A solution is needed to coordinate multiple autonomous entities, situated in an environment, in a decentralized way to achieve a globally coherent spatial movement of the agents. The coordination mechanism has to be robust and flexible in the face of frequent changes. Local estimates of global information are the only possible way to coordinate.
Problem/intent	Spatial Movement, Pattern Formation, Structure Formation, Routing, Integration of Contextual Information
Forces	• Exploration vss. exploitation • centralized vss. decentralized • optimality vss. robustness • Agent vss. environment responsibility • Greedy vss. focused
Solution	Spatial, contextual, and coordination information is automatically and instantaneously propagated by the environment as computational fields. Agents simply follow the "waveform" of these fields to achieve the coordination task, i.e. no explicit exploration.

5.2.2 Co-Field Coordination Mechanism

Inspiration

Co-field (gradient field) coordination mechanism takes its inspiration from physical and biological world. In physics, masses and particles adaptively move and globally self-organize their movements accordingly to the locally perceived magnitude of field values. In the same way, biological organisms cooperate using some form of chemicals.

Conceptual Description

To use this decentralized coordination mechanism in a computational systems, we can observe the following guidelines which may help us in designing the system adapted from [47-53]

- The field has to be translated into an artificial data structure representing the Gradient Field.
- Each field is characterized by a unique identifier, with
  - A location-dependent contextual numeric value
  - A field Propagation Rule
- Procedurally a gradient field is started, initiated, or injected into the environment from a certain "source" location by a Gradient Initiator which conveying some application-specific information about the local environment and/or about the initiator itself
- The gradient fields are propagated by the Environment, according to its Propagation rule, from the starting location to the neighbors of that location.
- In turn, the neighboring locations modify the strength and re-broadcast the gradient to their neighbors which is repeated until the gradient has propagated far enough.
- Each intermediate location stores and forwards only the gradient part with the minimum strength value it has received for that particular gradient field.
- As such a "waveform" gradient map is formed in the environment which conveys useful context information for the coordination task.
- Then agents follow the waveform (deterministically or with some probability) by moving to a neighboring location. This allows agents to coordinate their movement with respect to the gradient initiator.

The conceptual description of this method can be summarized in the UML diagram as shown in the fig. 5.1 which is adapted from [54].

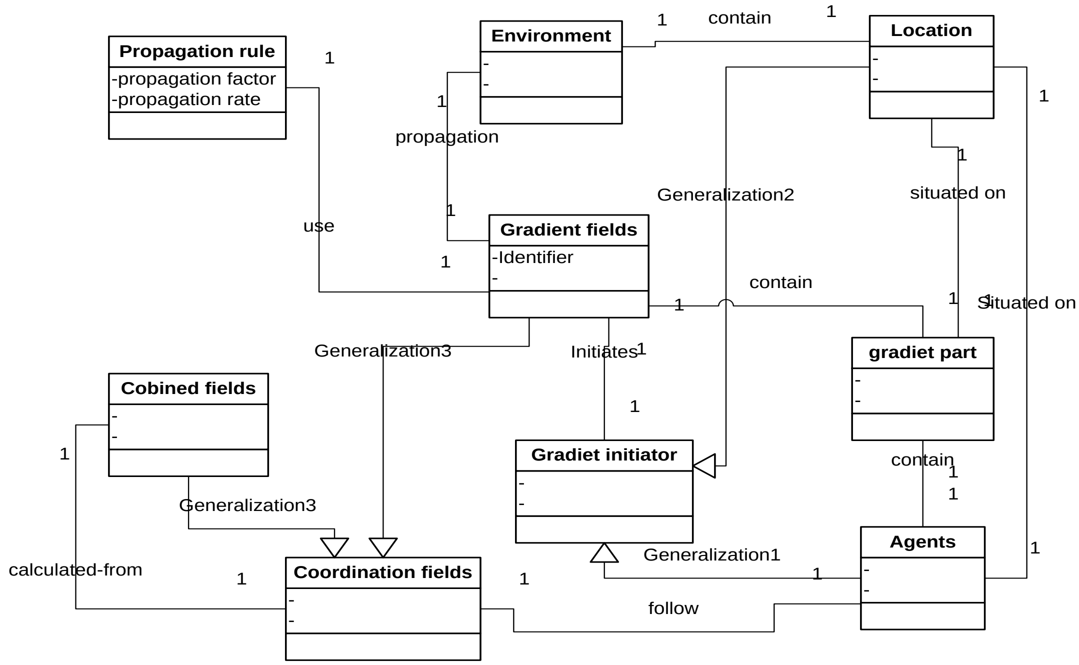


Figure 5.1 A conceptual descriptive model of the co-fields (gradient) coordination mechanism

5.3 A Headway Regulation Using Co-field Coordination Mechanism

Problem Formulation

The hypothetical rout that we use as an illustration of designing the required SWARM AI solution is shown in the figure below. In order to formulate the problem to be solved, a lot of assumptions concerning such factors as, distribution and arrival rate of passenger at each stop, is taken. For the purpose of suitability, we listed our component of the problem such as assumption, objective function below

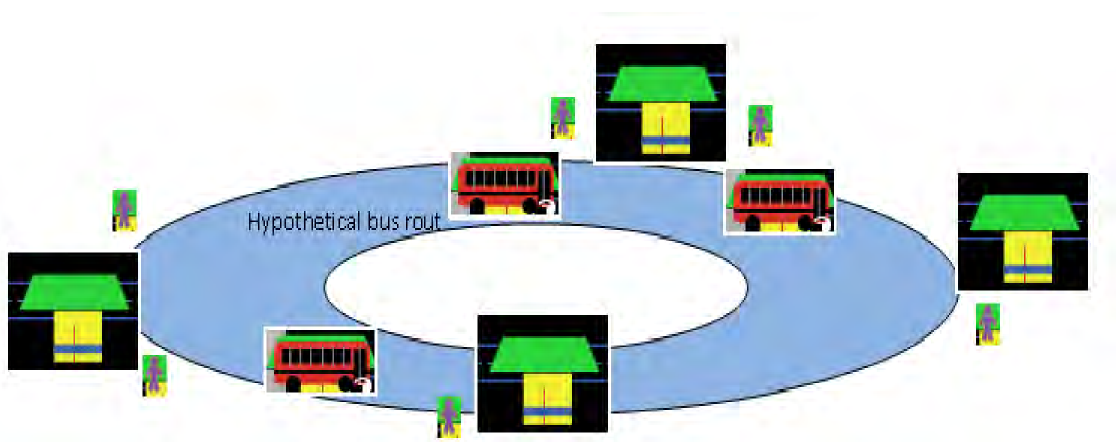


Figure 5.2 Conceptual route used to represent the environment of ITS

- **Assumption**
  - Constant inter-stop time
    - Only deal the effect of passenger distribution on the reliability of the ITS
  - Alighting passenger are not influenced by control methods
  - Only boarding passengers are subjected to control methods
  - Passenger board and alight in sequence
    - First alighting
    - Then boarding
- **Measure of effectiveness (objective function)**
    - Macroscopic system requirements
      - Throughput
        - Number of passenger served per simulation time (measuring passenger delay indirectly)
        - Capacity utilization of each bus
          - Histogram
          - standard deviation
          - Mean
  - **Major decision variables (major strategy)**
    - When should the bus depart (determining appropriate holding time)
- **Input to the simulation**
  - Fixed input variables for the simulation
    - unit time
    - unit space
    - number of
      - buses
      - bus stops
    - terminal headway
    - lay-off time
    - rout length
    - inter-stop time
    - bus capacity
    - both boarding and alighting time per passenger
    - sequence of passenger boarding and alighting
      - first alighting
      - Then boarding
  - Variable input variables for the simulation
    - Passenger arrival rate (rate of passenger arrival at each stop)
      - distribution
      - values of distribution function (mean, variance)
    - Passenger utility (distribution of passenger arrival across all bus stops)

5.4 The Proposed Coordination Method Design

Three types of fields has been hypothetically has been identified. These are exploration fields, exploitation fields and combination fields. However the identification of required fields, the context to be represented by each of them, their rule of propagation, combination, injection and so on, is still an open research gap and needs further research. For this reason we has only tentatively identified these fields which of course, needs further tinkering in the multi-agent simulation on which it will be designed. This is because the required emergent and self-organization effect can be only being known after the simulation is finished and there is no step by step procedure to design it. The purpose and function each field is described in the next section.

5.5 Informal Description of the Proposed Methods

Exploration fields: Each bus stop agent with a passenger waiting for bus agent emits an exploration field which reflects the current state of the bus transit system. The context that it has is the number of passengers waiting at that stop. The field propagates backward to the starting terminal of the rout. While it propagate back ward, each intermediate bus stop agent add its passenger demand to what it receive and propagate it again. The purpose of this field is to inform any bus agent with the current dynamics of passenger spatial and temporal distribution in the system.

Exploitation field: The exploration field only represents the current state of the bus transit system. However responding to every fluctuating state is not efficient way of responding in stochastically dynamic system. Some sort of learning about repetitive spatial and temporal pattern may make the co-field coordination model less greedy. Here as all bus stop agent do not serve as a boarding stop (especial at the end of the rout), we use this field as a sign about the number of boarding bus stop agent left so that unnecessary holding time do not incurred which increase the cost of on-board passengers.

Combined field: This is the field that the bus agents directly use in their coordination methods. It is the combination of both exploration field and exploitation fields. Bus agent consult the gradient of this field in their coordination activity. By simply following this field, we will have simple agent with simple rule base system which improves the scalability and maintainability issue of the solution.

We may summarize the whole process of the model as shown below,

- At regular intervals, and concurrently with the bus transit operation, from each injection bus stop artificial computational fields are propagated toward destination bus stop (sinking bus stop) as shown in table 5.1 and table 5.2
- Each bus agent act concurrently and independently, and communicate in an indirect way with each other using the bus stop agent through the computational fields (co-fields) they read and write locally on the bus stop agent.

- Each bus stop agent searches for a minimum holding time at each bus stop so that the global objective of the whole system (improves reliability, minimize total passenger delay and maximize capacity utilization of each serving bus).
- Each bus agent moves step by step toward its destination station. At each intermediate bus stop a combinational fields is used to calculate the holding time that the bus uses. The method makes use of (1) bus stop-local combined co-field, (2) bus stop-local problem-dependent heuristic information(number of passenger), which is represented by exploration fields and (3) the bus stop agents memory about past decision of previous bus agents (exploitation fields) and so on.

The complete algorithm

The overall algorithm which includes both the simulation framework and the metaheuristic part is shown in table 5.2. The main algorithmic component of this overall metaheuritic is the calculation of holding time (thold) which uses the co-field dynamics to calculate its value. As described in chapter 2, by regulating the time that the bus holds at bus stop even without doing anything, the relative position of each bus, which plays a major factor in the reliability of the whole bus transit system, the reliability of the system will improve.

The method is similar to that of developed by [60], but our method use a co-field dynamics instead of pheromone dynamics of ant colony metaheuritic. As the agents of the swarm ai system (bus agent in our case) are simple without any global knowledge of the whole system, there should be a mechanism to bring relevant global information to the locality of these agents. That is just what the co-field construct is doing in our case. By propagating at the interval of field-refreshing-time, the co-field construct (exploration field in our case) bring contextual information (passenger distribution in our case) to the locality of the bus agent so that optimum holding time will be calculated.

Moreover as the bus system is a stochastic system with random passenger arrival rate another type of co-field (exploitation field) will be use to create a window of recently calculated values so that recent past calculated values of holding time has some significance in the calculation of the current holding time. The propagation heuristics which is shown in table 5.3 whose code segment in NetLogo implementation shown in table 5.4 is responsible for bringing contextual information (number of passengers not yet served in our case) to the locality of the bus agent which is calculating the require holding time.

Table 5.2 The multi-agent implementation of the proposed methods

Procedure()
setup() // initialize world and indeviduales

- while current-simulation time != length-off simulation-time
  - foreach time step
    - for each agent
      - foreach bus-stop agent // concurrent activity on each bus stop
        - if t mod field-refreshing-time = 0 then
          - Propagate-fields(number-of-passenger)
          - Update-field-parametre()
        - end-if
        - If time-untile-next-passenger-arrival-time = o
          - creat-new-passenger()
        - endif
        - if bus-stop = terminal-stop
          - handle-despatching-stops()
        - endif
      - endforeach
      - foreach bus agent
        - Thold = calculate-local-holding-time(local-combined-fields, local-passenger-dynamics)
        - If thold = 0
          - depart()
        - elseif thold > 0
          - wait for thold time
        - endif
      - end for each
      - for each passenger agent
        - if off-board and local capacity available
          - board-bus()
        - elseif on-board and current-station = destination-station
          - alight
        - else
          - stay-in-the-bus
        - end
      - end foreach
    - end foreach
    - ask world (observer) to do
        - measure statistics
        - do cost benefit analysis
        - do visualization
  - endforeach
- endwhile

Table 5.3 The Field propagation pseudo code

- Field-propagation ()
    - If (t mod refreshing-time = 0)
    - foreach (co-field-struct) do
        - while (current-tation != sinking-station) do
            - next-station = PopMemory-from-station
            - current = next_hop
            - UpdateLocalTrafficModel()
            - update-local-exploration-fields()
            - update-local-exploitation-fields()
            - Update-local-combinedfields()
        - end-while
    - end-foreach
    - endif
- end procedures

Table 5.4 The source code implementation the propagation function

```
to propagate fields
while [i > 0  and j > 0 ]
          [
            if any? bus_stops with [ forward-station-number = i] and any? bus_stops with
            [ backwared-station-number = j]
             [
              let x one-of bus_stops with [forward-station-number = i]
              let z  one-of bus_stops with [backwared-station-number = j]
              let p [#forward-of-board-passenger] of t
              if is-list? p
              [set p one-of p]
              let m [#backward-of-board-passenger] of k

              if is-list? m
              [set m one-of m]
              ask x
                 [set forward-exploration-field #forward-of-board-passenger + p]
                  ask z
                 [set backward-exploration-field #backward-of-board-passenger + m]
```

```
            set t x
            set k z
          ]
        set i (i - 1)
        set j (j - 1)
      ]
end
```

5.6 Characteristics of the Designed Coordination Methods

In section 3.2.4 we have mentioned some principles and requirements that any SWARM AI metaheuristics should fulfill which was summarized in table 3.2. In this section we will give some account of how the designed metaheuristic address those principle and requirements by formalism of template proposed by [44] for cross checking the requirements with our model.

- **Coupling** :
  - **Use a distributed environment**.—The bus transit system as part of a general transportation system is a distributed system over space (spatially). For the ITS system, each bus stop agent maintains a computational field (co-field) data structure to distribute global information so that bus agents access it locally.
  - **Use an active environment.—**The environment implements the basic co-field dynamics of propagation (at regular interval of time), combination (e.g. linear), and injection and sinking (at location of injection and sinking bus stops).
  - **Keep agents small.—**Both bus and passenger agents are small compared with the overall system, and all interactions are local. No single agent can solve the problem. No bus agent know on its own that it is part of a big system, nor can any bus agent access the global objective value at any time.
- **Autocatalysis**:
  - **Think Flows rather than Transitions**.—the fundamental information flow in this application is the artificial computational field flow (co-fields).
  - **Diversify agents to keep flows going.—**This architecture has three main co-field types, exploration fields, exploitation fields and the combined fields.
- **Function:**
  - **Provide a mechanism for selecting among alternative behaviors**.—bus agents adjust their holding time using the computation of combined field.

5.7 Implementation of the Co-field Based Headway Regulation

As already pointed out in the last chapter, the actual design of the metaheuristics is characterized by some form of iterative model development which resemble some sort of try-and-error strategy. Here there are two abstraction of iterative development. The first is a meta-iterative development which refers for its iteration the overall objective (requirement) of the whole system and contains in it many iteration of high-level specification of heuristics and many implementation level iterations in which a single heuristic is changed into a computational model. One example of meta-iterative development is shown in fig. 5.4 .

The second level of iterative development includes methods such as KISS, TAPAS and KIDS that we mentioned in the last chapter. For our purpose, we are going to use again the TAPAS method of iterative model development.

As already discussed in the last chapter, the implementation of this coordination mechanism using multi-agent simulation model will actually enable as to tune the variables of interests such as types of fields, what their contextual representation will be, the emission and refreshing rate should be, how do different field should combined to give the combined fields so that it will coordinate the activity of the agents so that the required macroscopic emergent and self-organized behavior of the system will result.

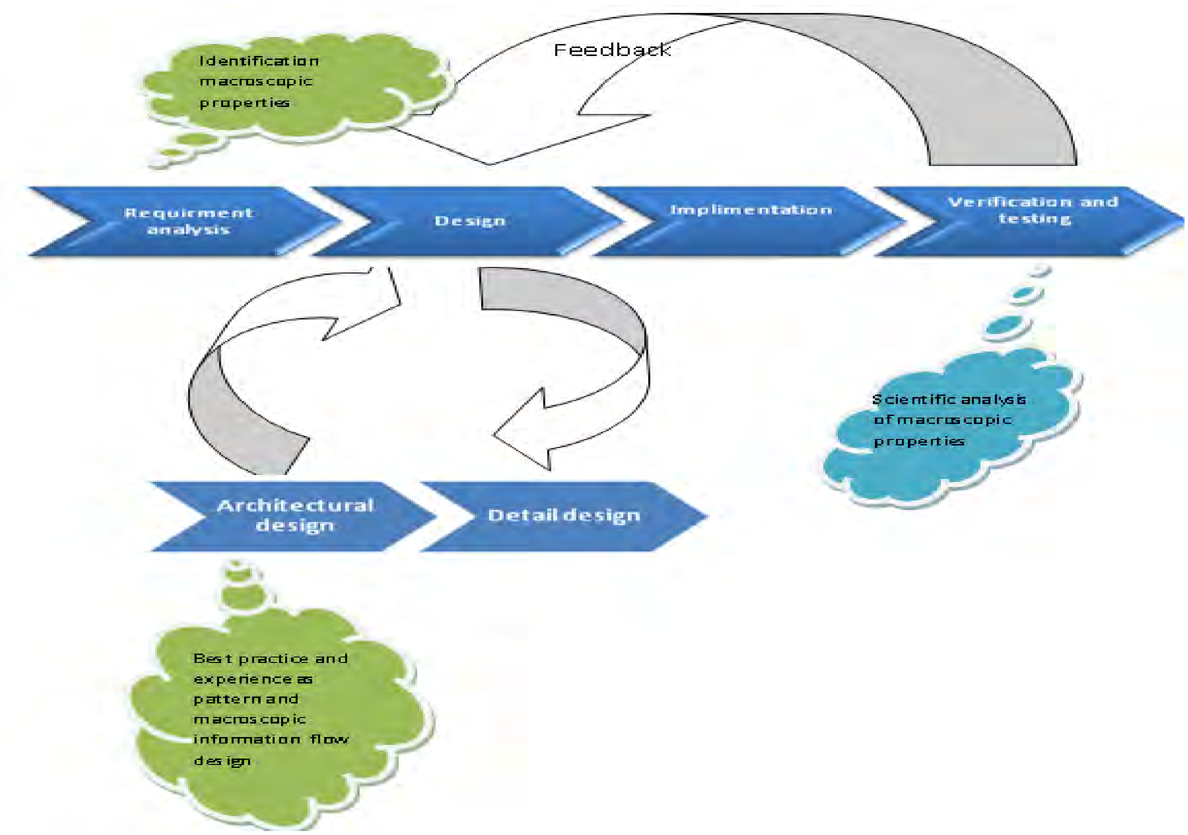


Figure 5.3 Adapted version of unified process for designing SWARM AI solution

Using the computational model developed in the last chapter, the TAPAS iterative model, the adaptive UML method shown in fig. 5.3, we have managed to implement the pseudo code shown in table 5.3. In the next chapter we are going to see its performance using a designed multi-agent simulation experiment.

5.8 Summary

In this chapter we have seen how the required interaction of the multi-agent system, on which the problem solving power of SWARM AI solution resides, is addressed using the co-field (gradient field) approach which takes its inspiration from how particles rearrange themselves using electrical and magnetic fields so that some sort of pattern will be formed. In the next chapter we are going to see about some experimental issue such as case study description which will improves the credibility of the developed multi-agent simulation, experimental design of the simulation and the result analysis.

CHAPTER 6

Experimental Setup, Result Analysis and Discussion

6.1 Introduction

In the last chapter we have seen how we have designed the proposed methods of regulating the headway of the bus transit system. In this chapter we are going to see how we are going to evaluate the performance of the proposed method and compare it with other comparable method. Finally we are going to discuss the result of which we get from the multi-agent simulation developed in chapter 4 by calibrating the simulation with a case study from Line 31 of Ambessa city bus organization, which is the largest city bus operator in Addis Ababa.

6.2 Description of a Case Study

The purpose of this case study is to calibrate the implemented multi-agent simulation described in the last chapter so that it reflect real-world scenario with constraint of real-world data availability. It also test the reliability of the simulator developed as to reflect the real world transit system which increases its effectiveness as a workbench to evaluate different algorithms representing real world system. Here we modify the simulation setting so that it reflect a specified real-world routes (Line 31) from Ambessa city bus enterprise.

6.2.1. AMBESSA CITY BUS ORGANIZATION

AMBESSA City Bus organization is the biggest public transport enterprise that serves Addis Ababa since 1942 G.C. and currently it is serving more than 104 routes and many hundreds vehicles [58].

6.2.2 Route 31 (From Sheromeda to Legehar)

The present study is based on a case study line number 31, Addis Ababa. The line is shown in fig 3 below. According to the figure the line connects the city northern part (Shiro Meda) to the central part of Legehar, passing through Addis Ababa Stadium, Meskel Scure, Arat kilo university , Amst kilo university , sidist kilo university and finally to shiromeda. The line consists of about 13 stops with a total distance of 7.4 km and an average single total travel time of around 30 minutes. Each day about 8 buses serve the route and these buses do on average a total of 210 trip per day [58]. Until recently the type of bus used in this line is an articulated one with 12 meters length, 38 seats and a maximum capacity of 100 passengers.

Table 6.1 Stop number and location of line 31

Index	Stop name	Index	Stop name
1	Legehar	1	Shiromeda
2	Meskel square	2	America embassy
3	Estifanos	3	Sadist kilo 2
4	Filwuha	4	Sadist kilo 1
5	National palace1	5	Ambesa gibi
6	National palace2	6	National museum
7	Arate kilo university	7	Arate kilo university
8	National museum	8	National palace2
9	Ambesa gibi zoo	9	National palace1
10	Sadist kilo 1	10	Filwuha

11	Sadist kilo 2	11	Rufaele
12	America embassy	12	Meskel square
13	Shiromeda	13	Legehar

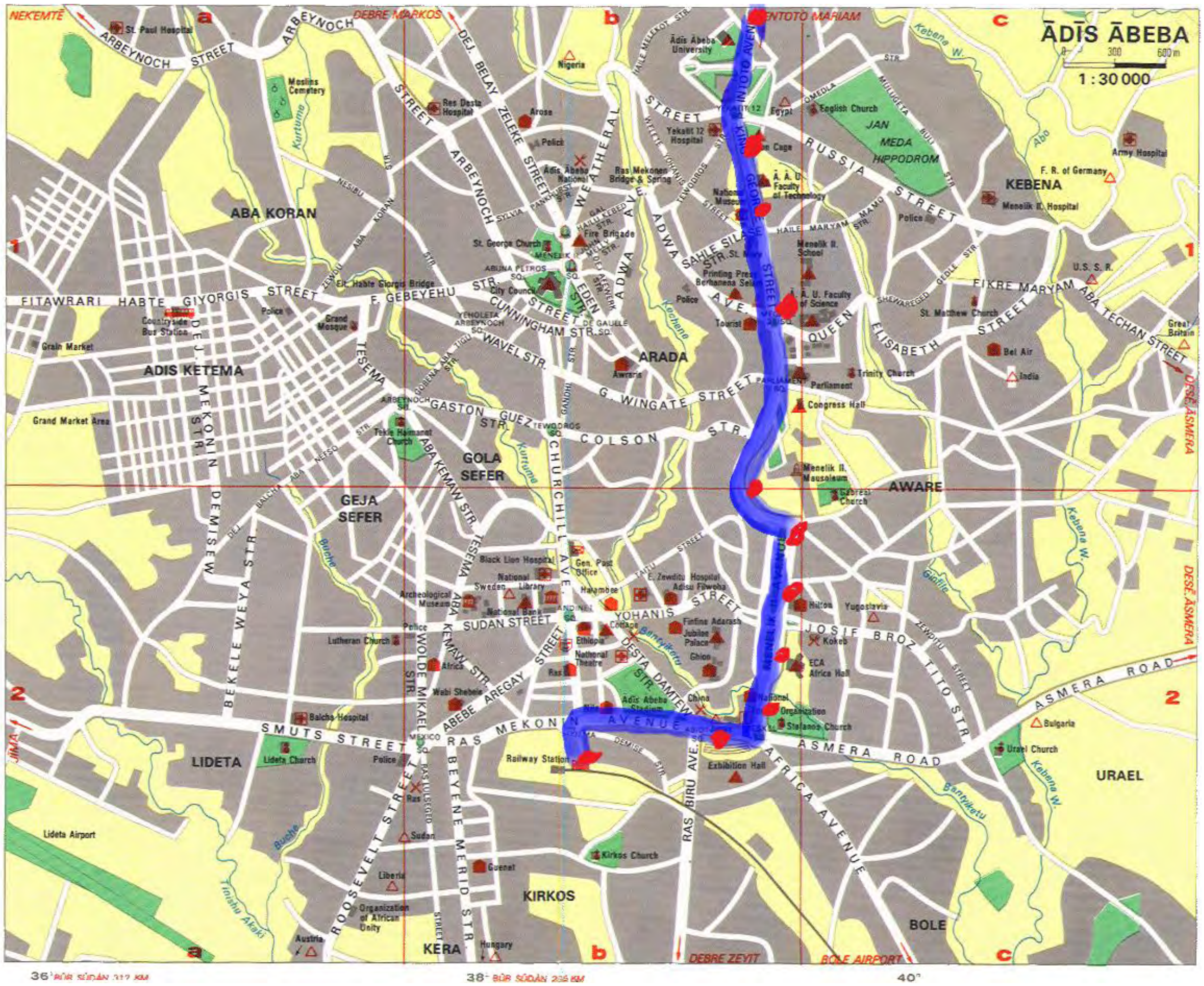


Figure 6.1: AMBESSA CITY BUS Number 31 service line map

6.2.3 Timetable

The current operational strategy of line 31 of AMBESSA city bus since as in any of its line is based on schedule-based. However this line so overloaded, it is schedule time is so short (around 8 minute), we can consider it a headway operational strategy with short headway (<= 12 minute) so that a random passenger arrival rate assumption can be justified.

6.3 Calibration of the Simulation

Model assumption:

- All passengers with a destination of the current stop must alight before any boarding passenger is allowed to board and the boarding and alighting time per passenger is equal to one time step in the simulation and around 3 sec. practically.
    - I.e.1 tick (time step) in the simulation = 3sec. in real time
- Average speed of the buses along the road segment is constant.
- Assuming 1/3 of travel time of a bus is spent on different bus stops of the rout (specially at terminal and crowded connection points), in line 31, 20 minute is spent actually moving along the bus route.
- As 3 sec in real-time = 1 tick in simulation, in each inter-stop road segment a bus spent an average time of 20/13 = 1.5385 minutes which will be 1.5385 *60/3 =30 tick
- This means in each road segment the bus agent spent 30 tick traveling the road segment and
    - The rout distance in the implemented our simulation is 30.4615384 which is equvalent to the distance of route 31 of our case study described in section 6.2.2.
    - Homogenous bus stop spacing, the distance of each simulated road segment will have 30.4615384/13 = 2.34 patch length.
    - This mean while the bus on the road segment is will move a simulated distance of 2.34/30 = 0.08 patch length.

6.4 Simulation Experiment Design

Assumption

- Fixed price bus service (fixed passenger utility)
- Constant inter-stop trip time (constant velocity)

From the case study described in section 6.6.2 we design the following experment shown in table 6.2.

Table 6.2 Variable setting for the experiment

Variable	Description	Value (range)
Terminal headway	Constant for methods	60 tick
Number of bus stops	Constant for all methods	13
Number of buses	Constant	6
Inter-stop distance	Constant	Equidistance
Vehicle capacity	Constant	100
Passenger arrival rate	Independent variable	[20-80]
Passenger utility	Independent variable	[0-1]

Vehicle speed in the simulation	Constant	0.08 patch/tick
Reliability regulation methods	Independent variables	• No-control (default) • Min [19] • Min-max [19] • SWARM-AI
Simulation length		10000 tick
Number of Replication		100

6.5 Experimental Result and Analysis

Although we cannot measure the phenomena of emergence and self-organization directly, we can measure it by the effect it have on the performance of the problem on which it is applied. In our case as our problem domain is the reliability of the bus transit system, so, we can measure the performance of the metaheuristics with the usual measure of effectiveness (MOE) with which the reliability of the bus transit system is measured. As there are two kinds of MOE of a bus transit system reliability problem (i.e. from the passenger side and the operator side), we take a measure from both sides. From the passenger side we take the delay measure of effectiveness (on-board delay and off-board delay) and from the operator side we take capacity utilization (the ratio of current load and its maximum load capacity), and headway distribution. The multi-agent simulation run is done on window 7 32-bit operating system on a hardware of del OPTIPLEX 755 with 2.39 GHz and 1GB RAM.

In all performance measures we have shown the total number of passengers already served in the fixed simulation time as a main MOE. This is because other MOE such as on-board passenger delay, off-board passenger, bus capacity utilization are only the derivative and part of this MOE.

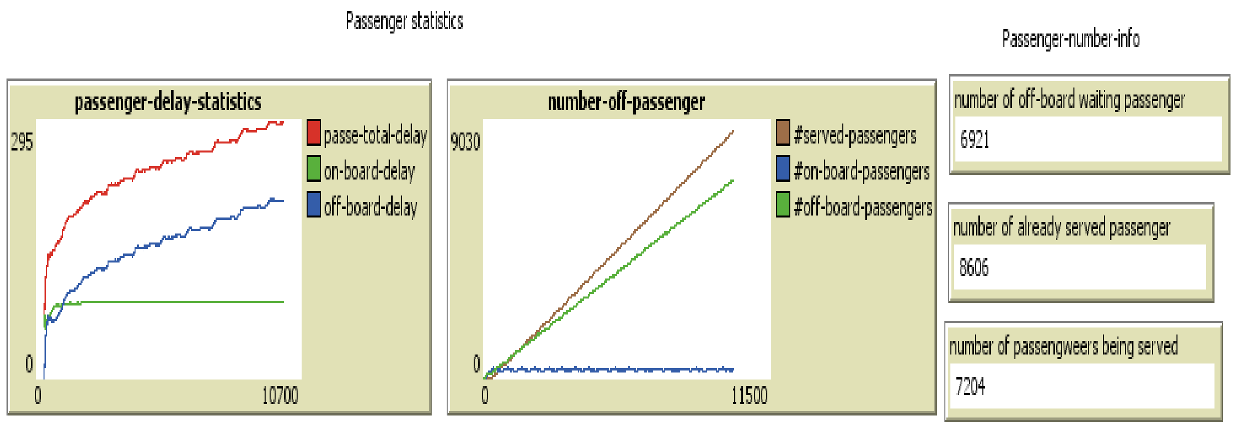


Figure 6.2 The default method Scenario 1 passenger delay statistics

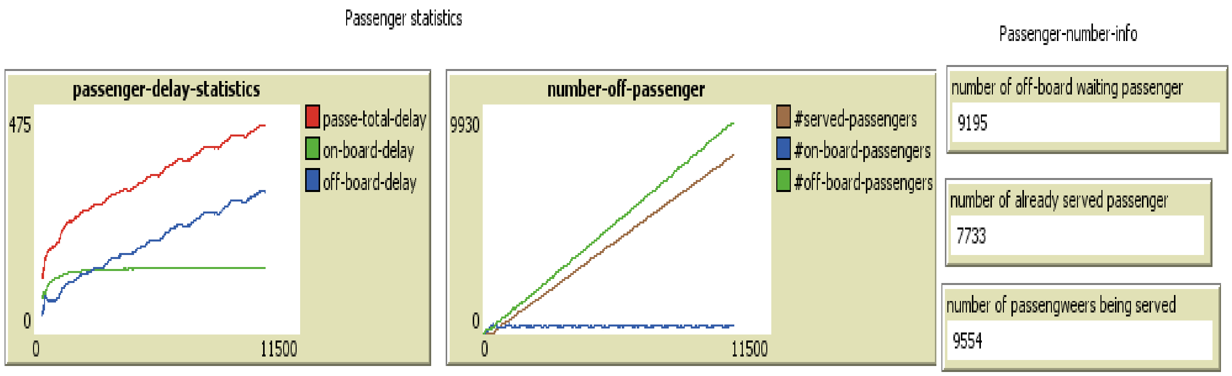


Figure 6.3 min methods Scenario 1 passenger delay statistics

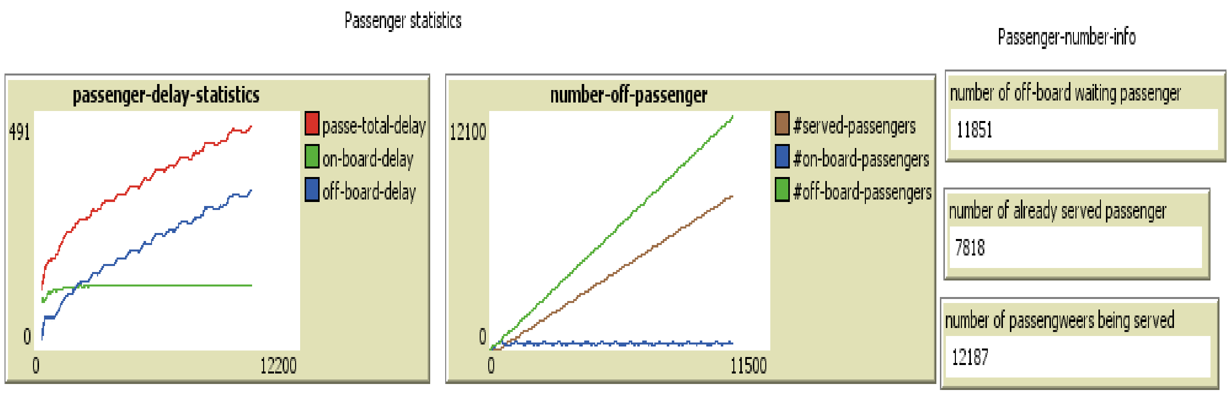


Figure 6.4 Max-min method Scenario 1 passenger delay statistics

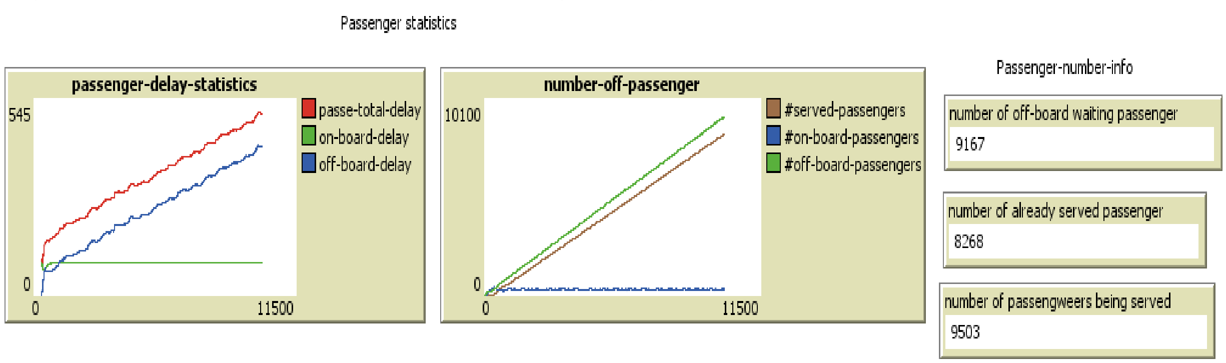


Figure 6.5 SWARM AI methods Scenario 1 passenger delay statistics

Four scenarios of passenger arrival rate and distribution has been tested with the multi-agent simulation model and experimental setting which is described in the last section. The result of the first scenario with the different methods (Default Min, Min-Max, and SWARM AI) are shown from fig. 6.2 to fig. 6.5. AS shown, the scenario is of high passenger arrival rate and denser arrival distribution of 0.5 passenger utility. This high arrival rate scenario is very difficult to monitor in all regulation methods. Even the no control of default method shows a high performance as shown in the number of passenger already served. And the SWARM AI coordination methods rate second next to the default method with small margin. This is because our method is almost similar to the no-control default method with the addition of some exploration and moving average exploitation (adaptive) activities. So the small margin difference can be taken as the cost of Adaptivity of the SWARM AI methods and the difference in the generated number of passengers from the random generator. Moreover both Min and Min-max method has difficulty of being adaptive to high passenger arrival rate and distribution. And that is why they perform poorly in this scenario.

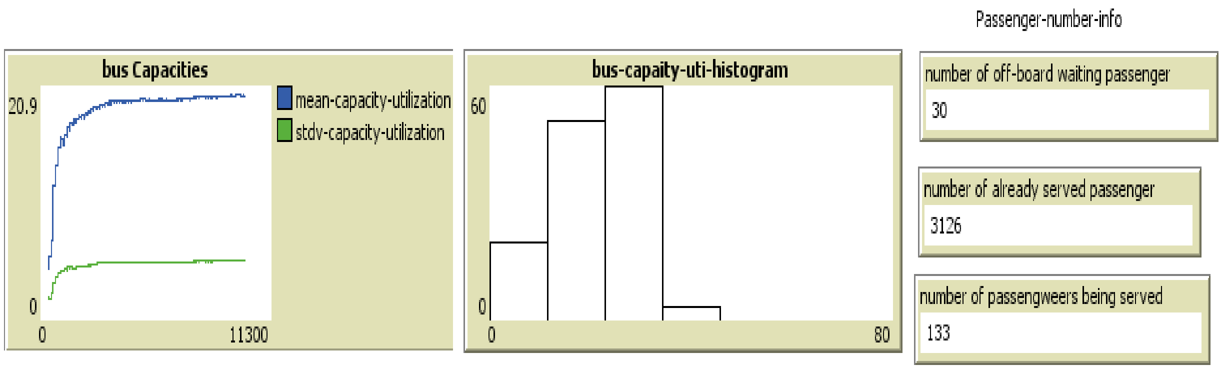


Figure 6.6 The default method capacity utilization scenario 2

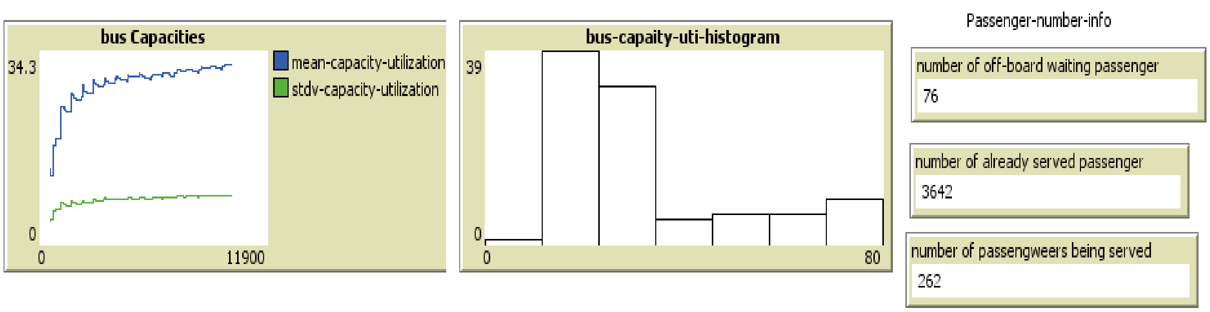


Figure 6.7 min methods capacity utilization scenario 2

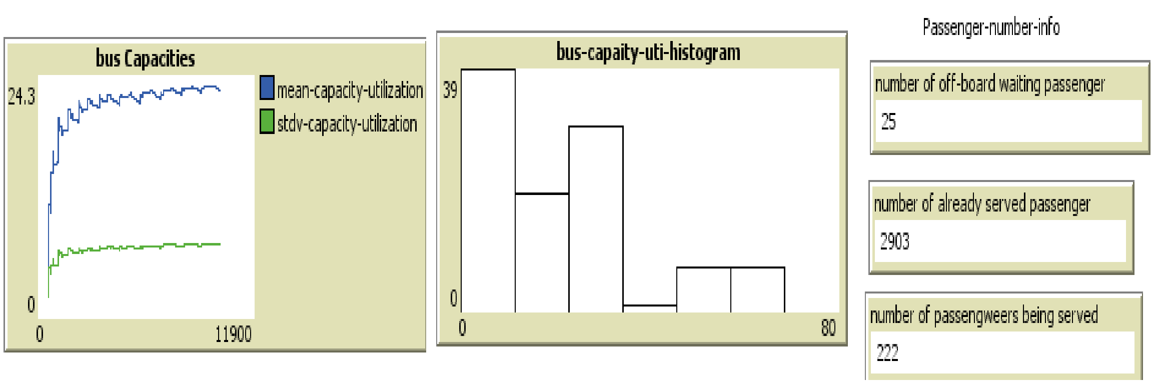


Figure 6.8Min-max method capacity utilization scenario 2

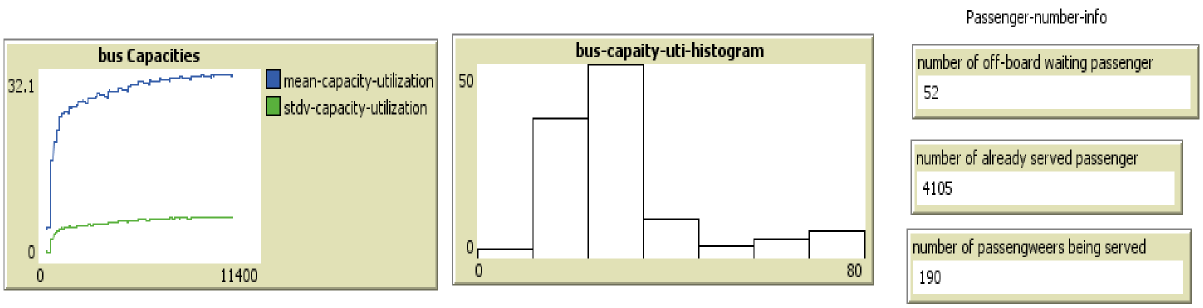


Figure 6.9 SWARM AI methods capacity scenario 2

In the second simulation experiment, a scenario of high passenger inter arrival rate and wider passenger distribution is considered. In addition to the passenger count statistics an operator side of measure of effectiveness which is capacity utilization is considered. As shown above from fig. 6.6 to fig. 6.9 in this scenario although the min method rate first in their average capacity utilization, the SWARM AI method perform best in the total count number of already served passengers. We also showed the Histogram of the capacity utilization. Histogram is the number of MOE being in some range of x-axis values. In our case the MOE is the capacity utilization (passenger load of a bus) with maximum capacity of 80. Here the Min method performs second best with min-max performing the worst. According to [44-45], this is to be expected as this is a low passenger density scenario which the min method performs well and adaptive methods such as our even perform better.

From fig. 6.10 to fig. 6.13 we see the headway distribution of the different methods. Although our method do not intend to control and monitor the headway of the bus transit system directly, we showed here only for completeness as the Min and Min-Max method are designed for regulating the headway of the system. In this medium passenger density scenario, although the headway graphs show similar trends, the SWARM AI method outperform in the main and cumulative MOE of total number of passengers already served. However the result shows that all methods shows good result although they differ in small margin with each other

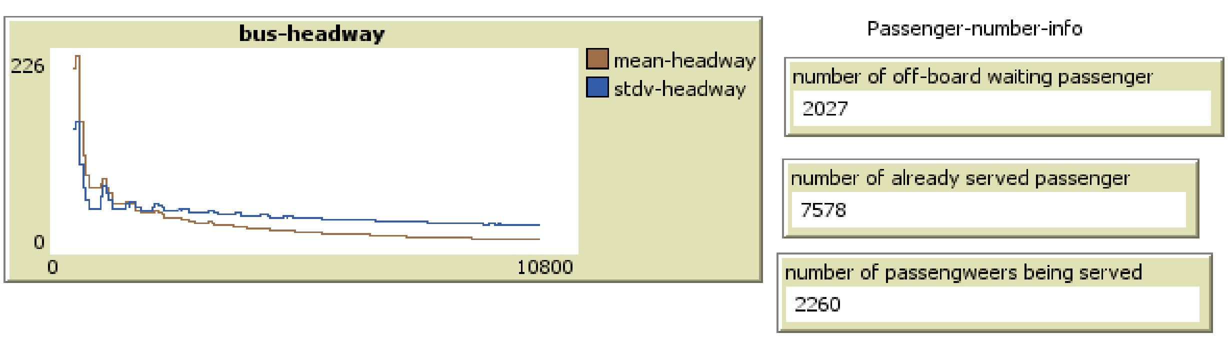


Figure 6.10 Default method scenario 3 headway measurements

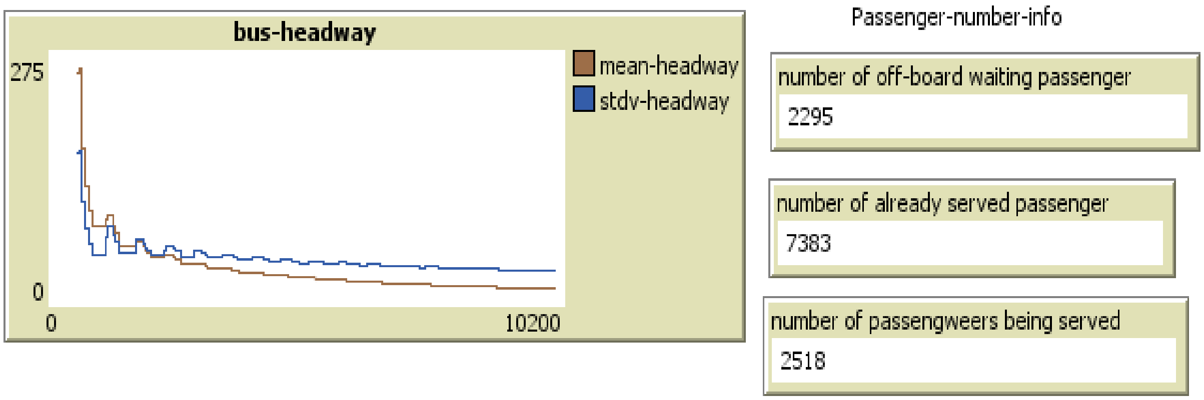


Figure 6.11 min methods scenario 3 Headway measurement

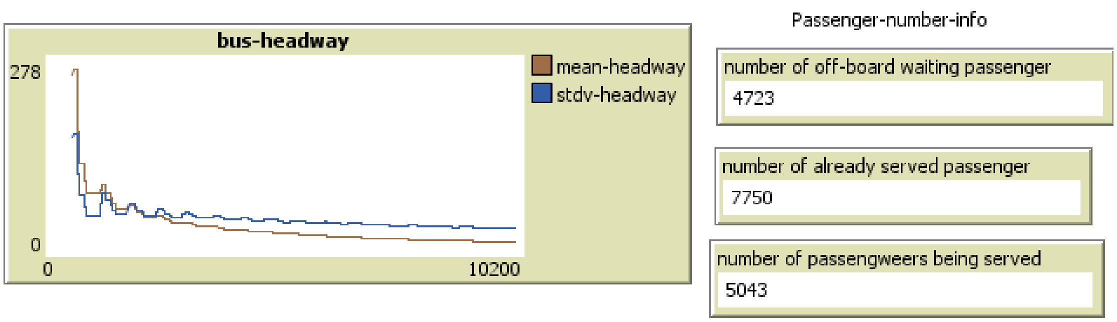


Figure 6.12 Max-min method scenario 3 headway measurements

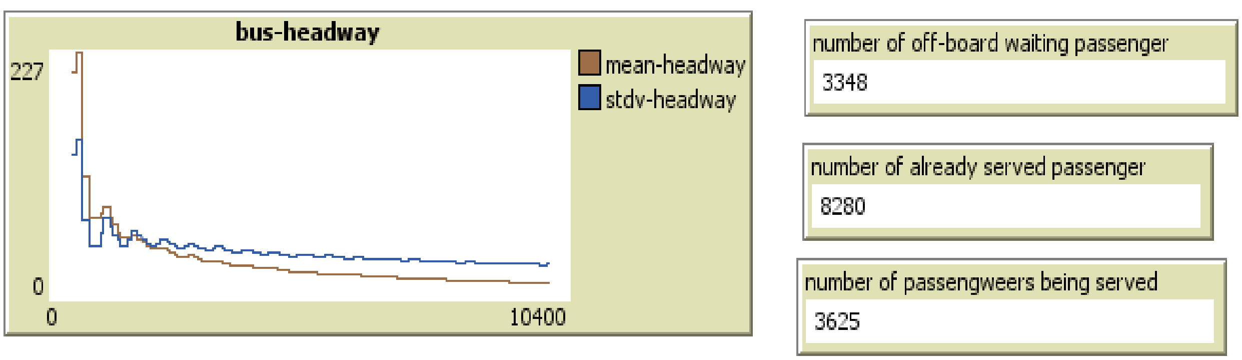


Figure 6.13 SWARM AI methods scenario 3 headway measurements

6.6 Discussion

The designed SWARM AI (co-field approach) method of distributed coordination for improving the reliability of the ITS bus transit system is a decentralized nature inspired metaheuristics. The idea is very simple but its implementation has a programming overhead which minimize the time needed to spent on the value added activities. The value adding activities in our case of the

SWARM AI distributed coordination metaheuristics design using the co-field approach form improving the reliability of the bus transit system includes:

- Identifying and selecting macro-scopic requirement of the bus transit reliability improvement from transportation engineering (operational science) literature
- Identifying and conceptual design of the natural operational dynamics of the bus transit system from the same source
- Implementation of the conceptually designed operational dynamics of the bus transit system in the previous step as a multi-agent simulation model which will be used both as a platform for designing the desired metaheuristics and as a platform for evaluating and comparing its performance with other comparable measures as described in the last chapter.
- The actual design of the metaheuristics using the implemented multi-agent simulation methods
- Finally evaluating and comparing the methods with other comparable methods using the multi-agent simulation experiment

The result from the simulation experiment is promising however it can be even be improved by further experimental try-and-error tuning and optimization.

6.7 Limitation

There are three types of limitation that this research has. The first limitation is concerned about the modeling of the bus transit operating dynamics. Here some assumptions are made concerning the simulated operation of the system. Some of them include:

- Only the effect of passenger dynamics on the reliability of the bus transit system is modeled (i.e. the inter-stop travel time kept constant)
- The distance between stop is equals
- All boarding station specified by the passenger utility parameter have homogeneous arrival time.

The second limitation is concerned about the design and implementation of the SWARM AI metaheuristics. Currently the design of SWARM AI metaheuristics is still in the process of development and no accepted universal process is available. The current trend is to use simulation model (such as multi-agent simulation model) both as a design and performance evaluation platform. This in turn makes the implementation (programming) process the most important but time consuming process. Moreover unless a standard computer platform (such as computer hardware, operating system, programming IDE and even the type and sequence of programming constructs), it is difficult to generalize our result to other scenarios.

The third limitation concerns about the simulation experiment carried out to evaluate the performance of the designed metaheuristics. Although we have done our best to do the experiment carful, still there are limitations. Some of them include:

- The calibration of the simulation model should be done with real-world data preferable automatically collected by ITS such as AVL, APC and so on rather than from literature research
- A more systematic and carful exploration (tuning) of parameters should be done the different parameters of the SWARM AI metaheuristics. These includes the refreshing rate of the co-field parameters, the exploration (alpha) and exploitation (beta) parameters, terminal headway of the bus transit system operation and so on.
- Moreover the value of the exploration (alpha) and exploitation (beta) parameters can be set based on the dynamics of the system (context) rather than being set randomly.

6.8 Practicality

As stated in chapter 1, the main motivation for proposed such types of decentralized coordination model is to effectively utilize the increasingly available near-real time data from ITS. This is because unless we can find a way of utilizing these data efficiently, we only incurring additional maintenance cost of these huge data. As a result if the bus transit system is already equipped by ITS apparatus such as AVL, APC, onboard computer and so on, little additional customizing technology is needed for putting the idea into practice.

6.9 Summary

In this chapter, we first introduced our system and application models and discussed several performance evaluation criteria with which we can evaluate the performance of the designed SWARM AI metaheuristics. Then, we presented several experimental results comparing the performance of different algorithms in different scenarios. The simulation results have shown the promising power of the proposed methods. The next chapter we are going to give our conclusion of the finding in this chapter and some recommendation of future work.

CHAPER 7

Conclusions and Recommendations

7.1 Conclusion

In this thesis we have examined the design of distributed coordination methods using SWARM AI metaheuristics (co-field) and applied to the problem of bus transit reliability problem. In order to do this we first model the bus transit operation with multi-agent simulation model using NetLogo IDE. This model was used for not only to measure the performance of the designed SWARM AI metaheuristics but also to design it.

After implementing the multi-agent simulation model, we iteratively design and implement the desired decentralized coordination methods using Co-field coordination method of SWARM AI. After calibrating the simulation model with a data using Line 31 of Ambessa Awtobis, Addis Ababa as a case study, we carried out a series of simulation experiments.

The proposed method of regulating the reliability of the bus transit system is compared with other comparable methods such as the no control methods, the Min-methods, and the Min-max methods [19]. Although different methods of effectiveness are taken from the simulation experiments such as on-board and off-board passenger delays, average and variance of bus capacity utilization, we use the total number of passengers served by the simulated bus transit system in a fixed simulation time as a main criterion of comparison. This is because the all other measure of effectiveness are in one way or other are a derivative of the number of passengers served by the system.

The result of the simulation shows that the proposed methods of regulating the reliability of the bus transit system is the most adaptive to wider passenger densities. It performs best in all tested scenario of passenger densities except the highest densities of passenger in which case the no control method outperform the other methods. Even in this case our proposed SWARM AI method performs worse only by a small margin. The result shows that decentralized heuristics (metaheuristics) of control methods without any sort of formal mathematical model can be an effective solution for improving the bus transit system reliability problem that operate in a complex distributed environment. More over this method also helps to solve the problem of how effectively to utilize the increasingly available huge near-time data from ITS.

7.2 Recommendations

Concerning the multi-agent simulation experiment, as described in the limitation section of the last chapter, our research can be improved in many way. The first improvement can be in the intensive and systematic parameter tuning (exploration) of the different parameters of the proposed methods. These parameters includes the exploration (alpha) and exploitation parameters which not only be tuned for optimum value but also their value can be set based on the some contextual information of the target system. This in turn helps the proposed method to be more adaptive. The next improvement is concerned with a rigorous statistical analysis of the output of the multi-agent simulation. This will help the result of the experiment to be more generalized.

We can also improve this research in the direction of the methodology used for our research (i.e. SWRM AI methodology) which can be summarized as follow:

- As the main bottleneck for designing SWARM AI metaheuristics (any emergent self-organizing solution) is filling the gap between the Micro-Macro mapping problem a design support methodology in this direction may be done
- Moreover as we can only measure the phenomenon of emergence and self-organization indirectly through the measure of effectiveness of the target system, a theoretical research in this direction may be very attractive.
- A design support research that can abstract away the problem of time consuming programming overhead in the process of iterative try-and-error multi-agent simulation for discovering the required Macro-micro is discovered may be worth making.

Reference

1. Abkowitz, M. and Lepofsky, M." Implementing Headway-Based Reliability Control on Transit Routes", Journal of Transportation Engineering, Vol. 116, No. 1, 49-63. 1990
2. Ceder Avishai, "Public Transit Planning and Operation: Theory, Modeling and Practice". Public transit planning and operation, Civil and Environmental Faculty, 2007.
3. Pilachowski, J.M. "An Approach to Reducing Bus Bunching". UCTC Dissertation No. 165, University of California, Berkeley, 2009.
4. Moses, I. E., "A Transit Route Simulator for the Evaluation of Control Strategies Using Automatically Collected Data", Master Thesis, Massachusetts Institute of Technology, MA, 2005.
5. Federal Transit Administration, "Advanced Public Transportation Systems: The State of the Art Update 2000", Final Report, 2000.
6. Federal Transit Administration, "Advanced Public Transportation Systems: The State of the Art Update 2006", Final Report, USA, 2006.
7. Eberlein, X.J., Wilson, N.H.M. and Bernstein D. "The holding problem with real-time information available", Transportation Science, 35(1), 1-18, 2001.
8. Fu,L., Yang, X. "Design and implementation of bus-holding control strategies with real-time information", Transportation Research Record 1791, 6-12, 2002.
9. Cats, O., Burghout,W, Toledo, T., Koustopoulos, H.N. "Evaluation of real-time holding strategies for improved bus service reliability", Submitted ti IEEE, 2010a.
10. Carlos F. Daganzo. "Public Transportation Systems: Basic Principles of System Design, Operations Planning and Real -Time Control", Course Notes UCB-ITS-CN-2010-1, institute of transportation study, university of california, Berkeley, 2010
11. Dessouky, M., Hall, R., Zhang, L., Singh, A. "Real-time control of buses for schedule coordination at terminal", Transportation Research Part A 37. 145-164, 2003.
12. Delgado, F., Muñoz, J.C. and Giesen, R. "Real-Time Control of Buses in a Transit Corridor Based on Vehicle Holding and Boarding Limits", Transportation Research Record. Vol. 2090, pp. 59-67, 2009.
13. Zhao, J., S. Bukkapatnam, M. M. Dessouky. "Distributed architecture for real-time coordination of bus holding in transit networks", IEEE Transactions on Intelligent Transportation Systems 4(1) 43–51, 2003.
14. Zhao, J., Dessouky, M., and Bukkapatnam, S. "Distributed Holding Control of Bus Transit Operations", 2001 IEEE Intelligent Transportation Systems Conference Proceedings, 978-983, 2001.
15. John J. BARTHOLDI, Donald D. EISENSTEIN, "A self-coordinating bus route to resist bus bunching", Transportation Research, USA, 2011
16. Bartholdi, J.J. and Eisenstein, D.D. "A self-coordinating bus route to resist bus bunching", Transportation Research Part B: Methodological, In Print, 2012.
17. Daganzo, C.F., Pilachowski, J. "Reducing bunching with bus-to-bus cooperation", Working Paper UCB-ITS-VWP-2009-12, U.C. Berkeley Center for Future Urban Transport, 2009.
18. Fu,L., Yang, X. "Design and implementation of bus-holding control strategies with real-time information", Transportation Research Record 1791, 6-12, 2002.
19. Gershenson, C. "Self-organization leads to supraoptimal performance in public transportation systems", Submitted, 2011.

20. Gershenson, C. and Pineda, L. A. "Why Does Public Transport Not Arrive on Time? The Pervasiveness of Equal Headway Instability", PLoS ONE 4(10):e7292, 2009 Http://turing.iimas.unam.mx/~cgg/NetLogo/4.1/metro.htmlo
21. Daganzo, C.F. "A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons", Transportation Research Part B, 2009.
22. Teodorovic D. 'Transportation modeling by multi-agent system: a SWARM Artificial intelligence approach Transportation Planning and Technology, Vol. 26, No. 4, pp. 289–312, 2003.
23. St¨ utzle T, Hoos H TheMAX-MIN, "Ant System and Local Search for the Traveling Salesman Problem", Proceedings of ICEC'97 - 1997 IEEE 4th International Conference on Evolutionary Computation, IEEE Press, 308-313, 1997.
24. St¨ utzle T, Dorigo M. " ACO Algorithms for the Traveling Salesman Problem", In K. Miettinen, M. Makela, P. Neittaanmaki, J. Periaux, editors, Evolutionary Algorithms in Engineering and Computer Science, Wiley, 1999.
25. Dorigo, M. and Gambardella, L.M. 'Ant colonies for the traveling salesman problem', BioSystems 43, 73–81, 1997a.
26. Reimann, M., Doerner, K., & Hartl, R. F. "Insertion based ants for the vehicle routing problem with backhauls and time windows". In M. Dorigo, G. Di Caro, & M. Sampels (Eds.), Proceedings of ANTS 2002—From Ant Colonies to Artificial Ants: Third International Workshop on Ant Algorithms, vol. 2463 of Lecture Notes in Computer Science (pp. 135–148). Berlin, Springer-Verlag, 2002.
27. Schoonderwoerd, R., Holland, O., Bruten, J., & Rothkrantz, L. "Ant-based load balancing in telecommunications networks", Adaptive Behavior, 5(2), 169–207, 1996.
28. Bullnheimer B. "Ant Colony Optimization in Vehicle Routing", Doctoral thesis, University of Vienna, January 1999.
29. Schoonderwoerd, R., Holland, O., & Bruten, J. "Ant-like agents for load balancing in telecommunications networks", In Proceedings of the First International Conference on Autonomous Agents (pp. 209–216). New York, ACM Press, 1997.
30. GF Newell and RB Potts, "Maintaining a bus schedule". In Proc., Second Conference Australian Road Research Board, Melbourne, volume 2, pages 388–393, 1964.
31. K. Ling and A. Shalaby. "Automated transit headway control via adaptive signal priority", Journal of advanced transportation, 38(1):45–67, 2003.
32. Andersson, P-A. and Scalia-Tomba, G-P.,"A Mathematical Model of An Urban Bus Route", Transportation Research – B, Vol. 15B, No. 4, 249-266, 1981.
33. Hickman, M., "An Analytical Stochastic Model for the Transit Vehicle Holding Problem", Transportation Science, Vol. 35, No. 3, 215-237.240, 2001.
34. Morgan, D.J. "A Microscopic Simulation Laboratory for Advanced Public Transport System Evaluation". Master Thesis, Massachusetts Institute of Technology, 2002.
35. Hallowell, S. F. and Harker, "Predicting on-time performance in scheduled rail- road operations: Methodology and application to train scheduling", Transportation Research, 1998.
36. Adamski, A. and Turnau, "Simulation support tool for real-time dispatching con- trol in public transport", Transportation Research, 1998.

37. Federal Highway Administration, “Automatic Vehicle Location Successful Transit Applications: A Cross-Cutting Study”. Report No. FHWA-OP-99-022, U.S. Department of Transportation, Washington, DC, 2000.
38. Edmonds, B and Moss, “From kiss to kids - an ’anti-simplistic’ modeling approach”, In et al., P. D., editor, Multi-Agent Based Simulation, number 3415 in LNAI, pages 130–144. Springer, 2004.
39. Martijn C. Schut, “Scientific Handbook for Simulation of Collective Intelligence”, Version: 2, 2007.
40. Malzoumi, E., Currie, G., Rose, G., Ceder, A., Moridpour, “A Practical Tool to Optimize Transit Schedule Timing Points and Slack Times: An Ant Algorithm Application”. Submitted for Presentation at 90th Transportation Research Board Annual Meeting, 2010.
41. Rufí, “Optimization Analysis of the Number and Location of Holding Control Stops to prevent Bus Bunching, A simulation-based evaluation of line number 1, Stockholm”. University of catalonia, transportation engineering, Barcelonal, 2011.
42. E. Bonabeau, M. Dorigo, and G. Theraulaz, “Swarm Intelligence: From Natural to Artificial Systems”. New York, Oxford University Press, 1999.
43. S. Brueckner. “Return from the Ant: Synthetic Ecosystems for Manufacturing Control”, Dr.rer.nat. Thesis at Humboldt University Berlin, Department of Computer Science, 2000
44. A. Drogoul, D. Vanbergue, and T. Meurisse, “Multi-agent based simulation: Where are the agents?” in MABS, 2002.
45. Edmonds, B. and Moss, “From kiss to kids - an ’anti-simplistic’ modeling approach”, In et al., P. D., editor, Multi-Agent Based Simulation, number 3415 in LNAI, pages 130–144. Springer, 2004.
46. M. Mamei and F. Zambonelli, “Field-Based Coordination for Pervasive Multiagent Systems. Models,Technologies,andApplications”, Series in Agent Technology, Springer, 2006.
47. M. Mamei, F. Zambonelli, and L. Leonardi, “Co-fields: towards a unifying approach to the engineering of swarm intelligent systems,” in Proceedings of the 3rd International Workshop on Engineering Societies in the AgentsWorld (ESAW’02),P.Petta,R.Tolksdorf, F. Zambonelli et al., Eds., pp. 68–81, Madrid, Spain, 2002.
48. M. Mamei and F. Zambonelli, “Programming pervasive and mobile computing applications with the TOTA middleware,” in Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications (PerCom ’04), pp. 263–273, Orlando, Fla, USA,2004.
49. M. Mamei and F. Zambonelli, “Programming stigmergic coordination with the TOTA middleware”, in Proceedings of the 4th International Conference on Autonomous Agents and Multi agent Systems (AAMAS ’05), pp. 551–558, ACM, New York, NY, USA, 2005.
50. Mamei, M., Zambonelli, “Field-based Coordination for Pervasive Multiagent Systems”, Springer Verlag, 2005.
51. Mamei, M., Zambonelli, F., Leonardi, “Co-fields: A physically inspired approach to distributed motion coordination”, IEEE Pervasive Computing 3 52-61, 2004
52. Malzoumi, E., Currie, G., Rose, G., Ceder, A., Moridpour, “A Practical Tool to Optimize Transit Schedule Timing Points and Slack Times: An Ant Algorithm Application”. Submitted for Presentation at 90th Transportation Research Board Annual Meeting, 2004.
53. T. De Wolf, and T. Holvoet, “A Catalogue of Decentralized Coordination Mechanisms for Designing Self-Organizing Emergent Applications”, Department of Computer Science, K.U.Leuven, Report CW 458, Leuven, Belgium, 2006.

54. T. De Wolf and T. Holvoet, “Decentralized Coordination Mechanisms as Design Patterns for Self-Organizing Emergent Applications”, Proc. of the 4th International Workshop on Engineering Self-Organizing Applications, Editors: S. Brueckner, S.Hassas, M. Jelasity, D. Yamins, Future University-Hakodate, Japan, pp 40-61, 2006,
55. L. Gardelli,M. Viroli, and A.Omicini, “Design patterns for self-organizing systems,” in Multi-Agent Systems and Applications V: 5th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS ’07), Leipzig, Germany, 2007.
56. M. Dorigo, V. Maniezzo, and A. Colorni. “The ant system: Optimization by a colony of cooperating agents”, IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26:29{41, 1996.
57. Hailu Gemechu, “alleviating scheduling problem in ANBESSA CITY BUS service enterprise”, Addis Ababa university post graduate program in industrial engineering p. 29, 2009.
58. Demelash Abate, “Analyzing Public Transport Performance Using Efficiency Measures and Spatial analysis, the case of Addis Ababa, Ethiopia”, 2007.
59. Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL
60. Di Caro, G., & Dorigo, M. (1997). AntNet: A mobile agents approach to adaptive routing. Technical report IRIDIA/97-12, IRIDIA, Universite´ Libre de Bruxelles, Brussels.