MAXIMIZING CONNECTIVITY AND PERFORMANCE IN MOBILE AD HOC NETWORKS USING MOBILE AGENTS

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Orhan Dengiz

Certificate of approval:

Robert L. Bulfin
Professor
Industrial and Systems Engineering

Alice E. Smith, Chair
Professor
Industrial and Systems Engineering

Jorge Valenzuela
Associate Professor
Industrial and Systems Engineering

Abdullah Konak
Assistant Professor
Information Sciences and Technology
Penn State Berks, Reading, PA

Joe F. Pittman
Interim Dean
Graduate School

www.manaraa.com

www.manaraa.com

MAXIMIZING CONNECTIVITY AND PERFORMANCE IN MOBILE AD HOC NETWORKS USING MOBILE AGENTS

Orhan Dengiz

A Dissertation
Submitted to
the Graduate Faculty of
Auburn University
in Partial Fulfillment of the
Requirement for the
Degree of
Doctor of Philosophy

Auburn, Alabama
December 17, 2007

المنارة للاستشارات

www.manaraa.com

UMI Number: 3301879

UMI Microform 3301879
Copyright 2008 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

المنارة للاستشارات

www.manaraa.com

MAXIMIZING CONNECTIVITY AND PERFORMANCE IN MOBILE

AD HOC NETWORKS USING MOBILE AGENTS

Orhan Dengiz

Permission is granted to Auburn University to make copies of this dissertation at its discretion, upon request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

المنارة للاستشارات

www.manaraa.com

VITA

Orhan Dengiz, son of Berna Dengiz and Ahmet Tahir Dengiz, was born in Ankara, Turkey on April 28, 1978. He graduated from TED Ankara College High School in 1995, and received his bachelor's degree in Civil Engineering at Middle East Technical University (METU) in 2000. After receiving his bachelors degree in civil engineering, he entered Auburn University, Department of Industrial and Systems Engineering for graduate studies, and received a MISE degree in 2002. He continued his education at Auburn University as a PhD student, and as a graduate research assistant. He is a student member of INFORMS.

المنارة للاستشارات
www.manaraa.com

DISSERTATION ABSTRACT

MAXIMIZING CONNECTIVITY AND PERFORMANCE IN MOBILE AD HOC NETWORKS USING MOBILE AGENTS

Orhan Dengiz

Doctor of Philosophy, December 17, 2007
(M.I.S.E., Auburn University, December 2002)
(B.S., Middle East Technical University, Ankara, Turkey, June 2000)

215 Typed Pages

Directed by Alice E. Smith

Mobile wireless ad hoc networks are instantaneous, autonomous telecommunication networks that provide service to users wherever and whenever the service is needed. The communication depends on wireless links that are formed between the users. A link is formed between two users if they are within each other’s wireless communication range. The mobility in these networks can cause links to disconnect, disrupting communications. A new strategy is proposed which controls the movements of some mobile agents to maintain network connectivity. The main objective of these mobile agents is to maximize network data flow, which is formulated as an all-pair maximum flow problem. This is accomplished by optimizing the movements of the agents to their next locations as the user nodes travel freely in the field. The representation of ad hoc network performance in terms of an all-pair maximum flow

المنارة للاستشارات

www.manaraa.com

problem is novel as is dynamically optimizing the agent nodes using heuristic algorithms integrated with network flow algorithms. Two evolutionary inspired, population based heuristic algorithms; a genetic algorithm and a particle swarm are developed along with an approximate linear programming model as optimizer tools. The results show the advantage of employing heuristic algorithms due to the complexity of the problem. While the approximate linear model could only solve small static and medium dynamic problems with poor results, the heuristics performed successfully for problems two to four times larger. These heuristic approaches will enable robust and physically self organizing networks with superior connectivity properties. The approach proposed in this research can be applied to static scenarios and dynamic situations. This is important because there are practical static applications of ad hoc networks, mainly in sensor networks. The novel models and algorithms developed should enable new research and and commercial opportunities in ad hoc wireless networking.

www.manaraa.com

ACKNOWLEDGEMENTS

The author would like to thank Dr. Alice E. Smith and Dr. Abdullah Konak for their invaluable help and guidance throughout this research. Also, the author expresses his gratitude to Dr. Robert L. Bulfin and Dr. Jorge Valenzuela for their consent to serving on the dissertation committee, and Dr. Gerry Dozier for his discussions. Special thanks to all of the Industrial and Systems Engineering faculty, staff and friends who have helped in many ways.

The author is grateful to his parents, Dr. Berna Dengiz and Mr. A. Tahir Dengiz, his sister E. Burçin Dengiz Olin, and his brother-in-law Mr. Samuel Olin for their continuous support, encouragement guidance and love.

المنارة للاستشارات

www.manaraa.com

Style manual or journal used: Elsevier, Ad Hoc Networks.

Software used: MS Windows, MS Word, MS Excel, Matlab, MS Visual Studio – Visual C++, Minitab, BOOST C++ Libraries.

Computer used: Intel P4 3.0 GHz CPU, 1GB RAM, Dell Dimension XPS series PC

www.manaraa.com

TABLE OF CONTENTS

LIST OF FIGURES xiii

LIST OF TABLES xx

CHAPTER 1. INTRODUCTION 1

1.1 Wireless and Wireline Telecommunication Networks 1

1.2 Ad Hoc Networks 4

CHAPTER 2. BACKGROUND 5

2.1 Wireless Mobile Ad hoc Networks (MANET) 5

2.2 MANET Performance 7

2.3 Applications of MANET 10

2.4 Routing in Ad Hoc Networks 11

2.4.1 Proactive Routing 12

2.4.2 Source-initiated On-demand Routing 12

2.4.3 Hybrid Routing Protocols 13

2.5 MANET Connectivity 13

2.5.1 Connectivity versus Routing 16

2.5.2 Connectivity and Performance Measures 16

2.5.3 Basic Graph Theory 17

المنارة للاستشارات

www.manaraa.com

2.6 MANET Mobility 19

2.7 Future Location Prediction 22

2.7.1 Cellular Models 22

2.7.2 Ad hoc Models 24

2.7.3 Kinematics Approach 26

2.8 Heuristic Optimization with Evolutionary Algorithms: Genetic Algorithm and Particle Swarm Optimization 27

2.8.1 Genetic Algorithms 27

2.8.2 Particle Swarm Optimization (PSO) 30

2.9 Evolutionary Algorithms in Dynamic Environments 34

CHAPTER 3. THE PPROPOSED MANET MANAGEMENT SYSTEM: PROBLEM DESCRIPTION AND MATHEMATICAL MODEL 38

3.1 Introduction 38

3.2 The Proposed MANET Management System 39

3.3 The Mobile Agent Location Problem 40

3.3.1 The Maximum Flow Analogy 40

3.3.2 The Mathematical Model 44

3.3.3 The Objective Function 47

3.3.4 The Mobile Agent Velocity Constraints 50

3.4 Future Location Prediction Using Kinematics 51

المنـــارة للاستشارات

www.manaraa.com

CHAPTER 4. A SPECIAL GENETIC ALGORITHM WITH NON-DETERMINISTIC BINARY DECODING FOR CONTINUOUS PROBLEMS 57

4.1 Background 58

4.2 NDBGA Algorithm 59

4.2.1 Motivation 59

4.2.2 Non-deterministic Binary Decoding 59

4.2.3 Mapping Rearrangement Mechanism 64

4.2.4 NDBGA Algorithm Structure 65

4.2.5 Gray Coding 67

4.2.6 Testing NDBGA 68

4.2.7 NDBGA Performance 74

CHAPTER 5. AN APPROXIMATE LINEAR MODEL WITH PIECEWISE LINEAR APPROXIMATIONS FOR DISTANCE AND DATA FLOW RATE 78

5.1 Mathematical Model 78

5.2 Notation 81

CHAPTER 6. THE MOBILE AGENT LOCATION OPTIMIZATION SYSTEM AND THE SIMULATION ENVIRONMENT 88

6.1.1 Dynamic Scenarios 89

6.1.2 Static Scenarios 90

6.2 The NDBGA for Mobile Agent Location Optimization 90

6.2.1 Tuning the NDBGA parameters 94

المنارة للاستشارات

www.manaraa.com

6.3 The PSO for Mobile Agent Location Optimization 97
6.3.1 PSO Parameters 100
6.4 Stopping Criteria 101
6.5 Semi-intelligent Agent Behavior 102
CHAPTER 7. TEST PROBLEMS AND RESULTS 106
7.1 Generation of the Static Test Problems 107
7.2 Generation of the Dynamic Test Problems 107
7.2.1 User Mobility Model 108
7.3 Performance Metrics and Use of Dynamic Information 109
7.3.1 Dynamic Problem Analyses 111
7.3.2 The Effect of Future Location Prediction 112
7.3.3 The Effect of Using the Best Solutions From the Previous Time Step .. 118
7.4 Algorithm Performances 124
7.4.1 Comparisons on Static Scenarios 125
7.4.2 Comparisons on Dynamic Scenarios 142
7.5 Cost-benefit Analyses 167
CHAPTER 8. CONCLUSIONS 172
REFERENCES 175
APPENDIX 184

المنارة للاستشارات
www.manaraa.com

LIST OF FIGURES

Figure 2-1 A MANET with six nodes, each with wireless transmission range = 1.7,........ 8

Figure 2-2 (a) A connected graph, $d_{min} = 1$, $d_{mean} = 8/5$. (b) A 2-connected graph, $d_{min} = 2$, $d_{mean} = 12/5$. (c) A disconnected graph, $d_{min} = 1$, $d_{mean} = 8/5$. 19

Figure 2-3 A binary represented multi-variable solution.. 28

Figure 2-4 (a) Single point crossover. (b) Uniform crossover.. 29

Figure 2-5 Pseudo code for basic PSO mechanism. 32

Figure 3-1 Normalized distance vs. normalized data rate, range = 450 m. 43

Figure 3-2 The pseudo code for the components of the objective function calculation ... 49

Figure 3-3 Pseudo code for predicting the location at time (t+H) 52

Figure 3-4 Real trajectory (H = 0) 53

Figure 3-5 Location prediction with H = 2 53

Figure 3-6 Location prediction with H = 4 54

Figure 3-7 Location prediction with H = 6 54

Figure 3-8 Location prediction with H = 8 54

Figure 4-1 Two dimensional binary coded search space 60

Figure 4-2 Regions represented by chromosomes c_a and c_b in NDBGA, a two dimensional case 60

Figure 4-3 Gaussian mapping from binary representation grid to the search space 62

www.manaraa.com

Figure 4-4 Graphical representation of binary mapping rearrangement for the 2D case . 63

Figure 4-5 NDBGA flowchart 67

Figure 5-1 The piecewise linear approximation of $f(x) = x^2$. An example is plotted for $x \in [0, 2]$, $P = 3$ 84

Figure 6-1 The pseudo code for the mobile agent location optimizer 89

Figure 7-1 Prediction horizon H versus the average minimum bandwidth 113

Figure 7-2 Prediction horizon H versus the average total bandwidth............................ 113

Figure 7-3 Prediction horizon H versus the average total solution time......................... 114

Figure 7-4 Change of (a) % user connectivity (b) all pair minimum bandwidth with simulation time.. 115

Figure 7-5 Change of total bandwidth with simulation time 116

Figure 7-6 Mobile agent behavior with location prediction 117

Figure 7-7 The effect of feeding the best solutions from (t-1) to t on average % user connectivity.. 119

Figure 7-8 The effect of feeding the best solutions from (t-1) to t on average minimum bandwidth between the MANET user pairs.. 119

Figure 7-9 The effect of feeding the best solution from (t-1) to t on average total bandwidth between the MANET user pairs.. 120

Figure 7-10 The effect of feeding the best solution from (t-1) to t on average solution time 120

Figure 7-11 Change of (a) % user connectivity (b) all pair minimum bandwidth with simulation time.. 122

Figure 7-12 Change of total bandwidth with simulation time 123

المنارة للاستشارات

www.manaraa.com

Figure 7-13 Solution time for each time step.. 123

Figure 7-14 Boxplot figure description... 125

Figure 7-15 An example problem (a) and its solution (b) for a small static scenario..... 126

Figure 7-16 The performance of the GA and the PSO algorithms on small scale static test problems in terms of the % user connectivity. (a) Boxplot (b) Best, average and worst performances over all problem instances 127

Figure 7-17 The performance of the GA and the PSO algorithms on small scale static test problems in terms of the minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances 129

Figure 7-18 The performance of the GA and the PSO algorithms on small scale static test problems in terms of the total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances................................. 130

Figure 7-19 The performance of the GA and the PSO algorithms on small scale static problems in terms of the solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances.. 131

Figure 7-20 The performance of the GA and the PSO algorithms on medium scale static test problems in terms of the % user connectivity. (a) Boxplot (b) Best, average and worst performances over all problem instances 133

Figure 7-21 The performance of the GA and the PSO algorithms on medium scale static test problems in terms of the minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances 134

المنارة للاستشارات

www.manaraa.com

Figure 7-22 The performance of the GA and the PSO algorithms on medium scale static test problems in terms of the total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances......................... 135

Figure 7-23 The performance of the GA and the PSO algorithms on medium scale static problems in terms of the solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances.. 136

Figure 7-24 The performance of the GA and the PSO algorithms on large scale static test problems in terms of the minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances..... 138

Figure 7-25 The performance of the GA and the PSO algorithms on large scale static test problems in terms of the total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances................................. 139

Figure 7-26 The performance of the GA and the PSO algorithms on large scale static problems in terms of the solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances.. 140

Figure 7-27 An example small scale dynamic scenario shown at (a) *t* = 1, (b) *t* = 25, (c) *t* = 50 and (d) *t* = 75, diamond shape represents user nodes and round shape represents agent nodes... 143

Figure 7-28 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on small scale dynamic test problems in terms of the average % user connectivity. (a) Boxplot (b) Best, average and worst performances over all problem instances..... 144

Figure 7-29 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on small scale dynamic problems in terms of the average minimum (nonzero)

المنارة للاستشارات

www.manaraa.com

bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances 145

Figure 7-30 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on small scale dynamic problems in terms of the average total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances 146

Figure 7-31 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on small scale dynamic problems in terms of the average solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances 147

Figure 7-32 Change in % user connectivity over simulation time 148

Figure 7-33 Change of (a) minimum bandwidth and (b) total bandwidth with simulation time 149

Figure 7-34 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on medium scale dynamic problems in terms of the average % user connectivity. (a) Boxplot (b) Best, average and worst performances over all problem instances 151

Figure 7-35 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on medium scale dynamic problems in terms of the average minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances 153

Figure 7-36 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on medium scale dynamic problems in terms of the average total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances 154

المنارة للاستشارات

www.manaraa.com

Figure 7-37 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on medium scale dynamic problems in terms of the average solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances 155

Figure 7-38 Change in % user connectivity over simulation time................................. 156

Figure 7-39 Change of (a) minimum bandwidth and (b) total bandwidth with simulation time 157

Figure 7-40 The performance of the GA and the PSO algorithms on large scale dynamic problems in terms of the average % user connectivity. (a) Boxplot (b) Best, average and worst performances over all problem instances 159

Figure 7-41 The performance of the GA and the PSO algorithms on large scale dynamic problems in terms of the average minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances 160

Figure 7-42 The performance of the GA and the PSO algorithms on large scale dynamic problems in terms of the average total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances 161

Figure 7-43 The performance of the GA and the PSO algorithms on large scale dynamic problems in terms of the solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances 163

Figure 7-44 Change in % user connectivity over simulation time................................. 164

Figure 7-45 Change of (a) minimum bandwidth and (b) total bandwidth with simulation time 165

Figure 7-46 Number of mobile agents versus average % user connectivity.................. 167

المنارة للاستشارات

www.manaraa.com

Figure 7-47 Number of mobile agents versus average minimum nonzero bandwidth ... 168

Figure 7-48 Number of mobile agents versus average total bandwidth 168

Figure 7-49 Number of agents versus the change in % user connectivity over simulation time 169

Figure 7-50 Number of agents versus the change of (a) minimum bandwidth and (b) total bandwidth with simulation time... 171

www.manaraa.com

LIST OF TABLES

Table 3-1 Path Loss verus Data Rate 41

Table 4-1 Test Functions F1-F10 71

Table 4-2 Test Functions F11-F20 72

Table 4-3 NDBGA Population Parameters 73

Table 4-4 Results for F1-F5 Compared With DPE 75

Table 4-5 Results for F5-F14 Compared With SZGA 75

Table 6-1 ANOVA analysis for NDBGA parameters 96

Table 7-1 Paired-t tests for NDBGA and PSO on static scenarios 141

Table 7-2 Paired-t tests for NDBGA and PSO on dynamic scenarios 166

المنارة للاستشارات

www.manaraa.com

CHAPTER 1

INTRODUCTION

With the increasing availability of computers with high processing speeds and large storage capabilities, individuals as well as businesses heavily depend on them. Moreover, the high data rate networks that connect computers and networks of computers to each other have enabled many services, but have increased the dependency of users to these interconnected networks. Computing is moving towards a time when a non-networked computer will be nearly useless.

1.1 Wireless and Wireline Telecommunication Networks

With the development of network technologies, wireline networks (networks built by cable connections) have become very fast, and reliable. Many researchers have studied the problems that arise in the design of wireline networks. Numerous methods and algorithms have been developed and tested which have more or less the same objectives; maximize reliability, speed, connectivity, and minimize cost [3, 4, 32, 33, 57, 82]. These objectives are sometimes seen in the form of constraints, where for example a minimum reliability value should be satisfied while minimizing the total cost.

المنارة للاستشارات

www.manaraa.com

Wireline networks have provided users with very fast and reliable networks over time and a large portion of the world economy now relies completely on these telecommunication networks. It is important to note that almost any form of information can be represented digitally today. Video, voice, and paper documents are common examples of data that are digitized every day. Wireline networks are fast, secure and reliable but they do have limitations. A significant limitation is that the users have to be near a data terminal so that they can connect their computer devices. However, it is crucial in today's world for individuals to travel and have access to a computer that is connected to a network. This is important for businesses or governments to be responsive to dynamically changing conditions and environments. Another limitation of wireline networks is that in case of a link failure, the repair job may involve the surrounding infrastructure.

With the recent developments in technology, very portable computer devices that have considerable processing speeds and data storage capabilities have become available. Such devices inherently have the ability to travel, and also manage individual or business tasks. Of course, portable devices need wireless data connectivity.

Wireless data communication networks can be divided into two main categories; wireless local area networks (WLAN) and wireless wide area networks (WWAN). Wireless local area networks have been highly developed and commercialized. In WLANs, users connect to local wireless access points and thus access local or wide area networks. The local access points have limited ranges, and the users need to be within the range of an access point in order to access the network. WLANs can provide users with

المنارة للاستشارات

www.manaraa.com

very fast data transmission speeds and reliability equivalent to wireline networks. However, range and mobility limitations exist. Users can only move within a few hundred feet, losing connectivity when they go out of the access points' range. It is possible to maintain connectivity by "handoff" to other access points covering the new location, provided that they remain in the same subnet.

The second type of wireless data telecommunication networks (WWAN) is centered on cellular networks, or the global system of mobile communications (GSM). GSM has made its way into the lives of countless people. There is currently an estimated 1.5 billion GSM subscribers worldwide [47]. This is a huge increase from the 170 million wireless subscribers in year 2000 [24]. Almost all mobile telephones that are in use today are using cellular network technology. Similar to WLAN, cellular networking technology requires base stations, located so a certain area is covered. Base stations are connected to a central switching office which also keeps a database of users that are currently using the network so that the necessary routings can be done. These systems are technologically advanced and provide users excellent service for voice communications. However, cellular systems have limited data transfer. While the data rates in cellular networks are enough for good quality voice communications, they are still very slow for simple networking tasks involving multimedia or large file transfers.

The newer generation cellular network technologies, i.e. 3G networks, offer higher data rates than previous cellular networks, but costly investments both in network infrastructure and subscriber equipment are necessary.

المنارة للاستشارات
www.manaraa.com

1.2 Ad Hoc Networks

A different type of wireless networking technology that was developed a few decades ago has become popular again and is attracting research interest [24, 81]. Called *ad hoc* networks, this type of wireless network does not require any fixed infrastructure and user devices communicate among themselves via the arbitrary and temporary “ad hoc” network topologies that they form [35, 41, 81]. Mobile ad hoc networks will be referred to as MANET in the literature throughout the rest of this paper, following the common practice in the literature. While the infrastructure topology is fixed and stable in a traditional WLAN, it is potentially very dynamic in a MANET [53]. Ad hoc networking had been used in combat fields and by emergency response teams but the wider availability of wireless capable computers and improved routing protocols have made it an emergency telecommunication network alternative [7, 24, 35, 41]. Ad hoc networks are considered essential in 4G wireless network architectures. 4G systems aim to provide ultra-high transmission speed of up to 100 Mbps, which is 50 times faster than those in 3G networks [24]. More detailed information on the MANET technology is given in the background section.

المنارة للاستشارات

www.manaraa.com

CHAPTER 2

BACKGROUND

In this chapter, some background information about wireless mobile ad hoc networks, their possible application areas and research challenges are described. The literature review is given for routing, different approaches for modeling and measuring mobility and connectivity, and finally for the prediction of mobility in mobile ad hoc networks.

2.1 Wireless Mobile Ad hoc Networks (MANET)

Ad hoc networks are networks formed without a central administration. They consist of nodes which use wireless interfaces to send data packets. The nodes in ad hoc networks can serve as both routers and hosts and they can forward packets on behalf of other nodes in the networks [41].

The roots of ad hoc networking can be traced back to the late 1960s and early 1970s but the technology had not been developed for the consumer market [24, 41]. The reason that ad hoc networks are now drawing attention is because of the availability and popularity of high performance portable handheld computers with wireless

www.manaraa.com

communication capabilities. An added motivation is that MANET has almost no initial investment cost.

The importance of being able to form instant, autonomous telecommunication networks is highlighted by the natural disasters that devastated different parts of the world in 2004 and 2005. Damage assessment and emergency response teams needed reliable telecommunication capability where almost all fixed infrastructure was damaged and non-operational for weeks or longer. The following quote is taken from [29], which describes the situation in New Orleans after the hurricane Katrina in August 2005:

> *"The devastation was so complete, so comprehensive ... that we couldn't figure out how bad it was," said Adm. Timothy Keating, chief of the U.S. military's Northern Command, which oversaw the Pentagon's Katrina effort. "On Tim Keating's list of things we need to work and to analyze very carefully, communications is at the top of that list."*

This aspect of ad hoc networking is enough by itself to justify the need for research to develop its technology and reliability, i.e. its usability.

A main advantage of ad hoc networks is that no infrastructure investment is necessary. This is a huge economic advantage from the point of view of investors. It also opens up possibilities in underdeveloped countries where infrastructure investment is lacking. Ad hoc networks are dynamic and flexible in terms of the coverage area. Network connectivity is a function of user movements and their relative locations. This is

المنارة للاستشارات
www.manaraa.com

an advantageous property from the users’ standpoint but the network dynamics need to be properly managed.

A reliable network is usually the number one priority for users. A very crude definition of a reliable network can be given as; the network is always expected to be available when access is required, given that the user equipment has no physical defects.

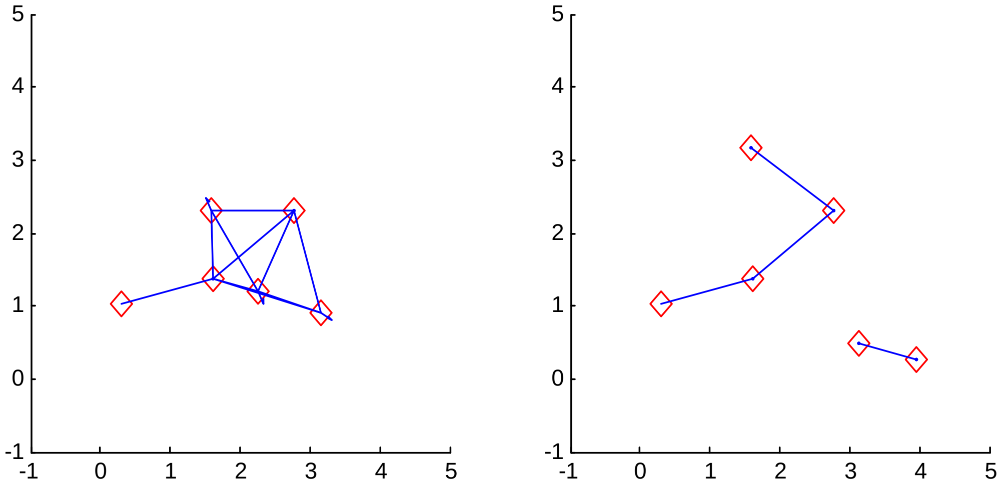
2.2 MANET Performance

There are many factors that affect the performance and reliability of a mobile ad hoc network. Links between the mobile devices sometime exist, and sometime not, depending on their locations relative to each other, their transmission power and the surrounding environment. New mobile devices can enter the system, or existing devices can disappear for various reasons including loss of battery power or loss of signal strength due to distance or other environmental causes. Under totally random user behavior, it is very likely that one or more users will lose their connectivity with the network or with the parts of the network due to their positions relative to other users. If a user is outside the range of its nearest neighbor in the network in terms of signal strength, then its access to the rest of the network will be unavailable.

Communication between the nodes of a MANET can be in a multi hop fashion, meaning data can be sent to a destination node not directly connected to the source node using an available route through other nodes. Each user can communicate directly to other users within its range. To communicate with nodes beyond its range, it needs to use intermediate nodes to relay the data packets, hop by hop [24]. Figure 2-1 shows a

www.manaraa.com

MANET with six nodes, each with a transmission range of 1.7 units at two different times.



(a) **(b)**

Figure 2-1 A MANET with six nodes, each with wireless transmission range = 1.7, (a) Connected (b) Disconnected

Gupta and Kumar have shown that if n identical stationary ad hoc network nodes, each with a data transmission rate of W bits/sec and a fixed range, are randomly located to form a wireless network using a non-interference protocol, the data throughput realized by any node for a randomly chosen destination has an upper bound of $\left(W/\sqrt{n\log n}\right)$ [48]. Even if all parameters such as transmission ranges, traffic patterns and node placements are optimally arranged, the bound on the throughput becomes $\left(W/\sqrt{n}\right)$. The apparent trade-off between the number of nodes and individual throughput rates is due to the multi-hop nature of wireless ad-hoc networks. Each node generates a certain traffic burden for other nodes, thus every node uses some of its capacity to relay other nodes' data. In order to

www.manaraa.com

decrease the number of hops necessary to reach a destination node from a source node, one might think of increasing the transmission range, but this causes increased interference. The mobility of an ad hoc network introduces additional variations to its capacity. Grossglauser and Tse suggest that mobility can actually improve the capacity of an ad hoc network when compared with a fixed network [46]. They propose a communication model by relaying the data to its destination using a number of relay nodes, delivering only when the relay node is closer to the destination, within a two-hop path. This eliminates excessive multi hop requirements. The drawback of this proposal is that the applications need to be delay tolerant. The communication waits until the mobile relay nodes are close to the destination nodes. This causes large delays, increasing with the size of the system, making it unsuitable for real time applications such as voice communications or remote control.

Whether fixed or mobile, the capacity of an ad hoc network is mainly bounded by individual transmission capacities. In this research, a method to maximize the individual data transmission rates between all user pairs and the total data transmission rate of a mobile ad hoc network is proposed. The actual capacity of the network will depend on the routing, scheduling and relaying of the communicated data, which are not addressed in this dissertation.

There have been some studies that investigate reliability in ad hoc networks by addressing data packet routing algorithms. This is an important problem for ad hoc network reliability. Different routing algorithms have been developed, each trying to optimize data packet routes by assessing network connectivity [1, 5, 11, 35, 54, 55, 60,

المنارة للاستشارات
www.manaraa.com

72, 73, 89]. Detailed aspects of the different routing protocols will be given in the following sections. However, regardless of what type of routing protocol is used, the first and most important requirement for communication between any two nodes is having at least one path linking them, which is the basic definition of network connectivity. Network connectivity is at a high level in the reliability hierarchy. If a single user or a part of a network has no connectivity, then data packet routing reliability becomes of secondary importance. The availability of paths between the nodes depends on network topology. Since mobile ad hoc networks are formed by mobile devices, they have continuously changing, dynamic topologies which is a distinguishing feature as well a challenge [53].

Due to the dynamic connectivity nature of ad hoc networks, special care needs to be taken in studying the connectivity problem as well as the routing reliability problem. This dissertation aims to develop a method that will optimize the network topology dynamically such that the network connectivity is maximized. Network connectivity is a broad term used to represent different objectives by different researchers. The term *network connectivity* is purposely used here, because a special connectivity measure will be developed taking into account the characteristic properties of ad hoc networking.

2.3 Applications of MANET

The most commonly envisioned application of MANET is military communications including combat, emergency response, search and rescue, maneuvers, etc. [7, 24, 35, 41, 81]. Besides the commonly envisioned uses of MANET, it can be used

المنارة للاستشارات
www.manaraa.com

where there is no telecommunications network infrastructure available. This could be because it has never existed, or there might be an existing infrastructure which is inoperable due to disaster damage. Rescue operations, rural construction sites, or rural land survey teams are examples. Ad hoc networks can also be used when the existing infrastructure is not capable of handling a short-term demand increase. An event area where tens or hundreds of thousands of people gather is an example [81]. Such a concentration in a town, in a concert hall, or in a stadium creates a short-term demand that is beyond the maximum available capacity of the local network infrastructure. Another important aspect of MANET is its ability to form an independent network within its users only. This could be useful if the communications need to be secured. For example, for military operations, without using any existing infrastructure in either friendly or hostile territory, secure communications can be established between military vehicles, mobile or stationary teams.

2.4 Routing in Ad Hoc Networks

The definition of routing in a telecommunication network is as follows: routing is the mechanism of directing data packet flow from the source to the destination. There are many different routing protocols, and different algorithms under those protocols, for fixed topology wireline or wireless networks with different constraints and objectives such as maximum path capacities or minimized costs. Similarly, there are different routing protocols and algorithms for ad hoc networks, with different objectives. In an ad hoc network since there is no fixed topology, managing routing is a very important task

المنارة للاستشارات
www.manaraa.com

to maintain the quality of service (QoS). Routing in ad hoc networks is much harder than routing in fixed topology networks. There are three main classifications of MANET routing protocols and each approach has its own advantages and disadvantages, according to the realized mobile network scenario [53]. These three protocols can be summarized as follows:

2.4.1 Proactive Routing

Proactive routing algorithms or table-driven algorithms work on the basis of a well maintained, i.e. frequently updated, routing table kept by every node in the network. The routing tables are always available and whenever a packet needs to be sent, the source node will send the packets via the best route found by a certain algorithm. The disadvantage of this protocol is that due to the dynamic nature of the network topology, the maintenance of the routing tables consumes a lot of the network bandwidth. Common examples of this routing protocol are destination-sequenced distance-vector routing (DSDV) [71], clusterhead gateway switch routing (CGSR) [23], and optimized link state routing (OLSR) [54].

2.4.2 Source-initiated On-demand Routing

On-demand routing is a reactive protocol, and paths are constructed only when there is a need to send a packet. Rather than continuously updating the routing tables, the source node initiates a path discovery algorithm before sending the packet. When the path discovery reaches the destination node, the information is sent back to the source node

المنارة للاستشارات
www.manaraa.com

and the data packet is then sent via the constructed path. Although the on-demand routing protocol does not use up valuable bandwidth like the proactive routing, a delay is incurred while constructing a route from the source to the destination nodes. Some examples of this type of algorithm are ad hoc on-demand distance vector (AODV) [72] and dynamic source routing (DSR) [55].

2.4.3 Hybrid Routing Protocols

The first two types of routing protocols have their weaknesses as described in their summaries. Hybrid protocols have emerged to form a MANET routing protocol that combine the advantages and minimize the weaknesses of the proactive and reactive protocols. Zone routing protocol (ZRP) is based on a hybrid approach. A node uses a proactive type routing for its neighboring nodes within a certain number of hops. Routing for more distant destinations is done using a reactive path discovery [70].

2.5 MANET Connectivity

Bettstetter [9] investigates node degree and connectivity characteristics of MANET, and terms these the two fundamental characteristics. The node degree and connectivity concepts for MANET are explained in more detail in Section 2.5.3. To give a basic definition, node degree is the number of links that a node has in the network, and connectivity is a measure of the total possible disjoint paths between node pairs. Bettstetter defines a simulation model which consists of three stages. First, a total of n MANET nodes are placed on a two-dimensional simulation area A using a uniform

www.manaraa.com

random distribution. Second, wireless transmission is modeled for each node based on omnidirectional, or circular, transmission with transmission range, R, and a certain path loss or signal attenuation model. Every user is assumed to have the same transmission range. Finally, a third model is defined for the mobility of the nodes, such as random waypoint or random direction.

Bettstetter represents the MANET as a graph $G = (V,E)$ where vertices set V is the nodes, and the edges set E is the links formed between the nodes within each other's range. More detailed information on how a MANET is modeled as a graph is given in Section 2.5.3. He develops analytical expressions for the minimum required R value such that the probability of having no isolated nodes is a high probability P. In that study, the probability that a node is isolated is given as in equation (2-1).

$$P(\text{a node has no neighbors}) = e^{-\rho\pi R^2} \tag{2-1}$$

where $\rho = n / A$, and similarly the probability that the network is connected is:

$$P(\text{every MANET node is connected}) = 1 - e^{-\rho\pi R^2} \tag{2-2}$$

Bettstetter provides the analyses of the transmission range R versus the probability that the MANET is connected. From his results, it is clearly seen that there is a certain threshold range value immediately below which the P(every MANET node is connected) is almost zero, whereas immediately above the critical range the probability is almost one. Known as the "phase transition," this behavior is fairly common in many graph measures [9, 38].

المنارة للاستشارات

www.manaraa.com

Similarly, Xue and Kumar [91] studied the number of nodes that each node needs to have in its neighborhood to keep the network connected. They showed that instead of some constant magic number, connectivity is almost certainly established if every node is connected to its nearest $5.1774 \cdot \log(n)$ neighbors, where n is the number of MANET nodes. Both of these studies assume uniformly randomly distributed nodes in a certain area and approach the connectivity in a probabilistic manner.

Cook and Marquez [28] proposed a two-terminal reliability calculation approach for a MANET with a random waypoint mobility model. The analytical expression for the expected number of neighbors is used to calculate the probability of link existence and a Monte Carlo based simulation calculates a two terminal reliability measure. Their results indicate that the two terminal reliability of a MANET increases with increasing node density, however it is bounded by the square of the node reliabilities.

Bettstetter also analyzes the impact of mobility on the measures he derived for MANET. However, his basic assumption for mobile node scenarios is that $n >> 1$, nodes are always distributed uniformly in the area and $A >> R^2 \cdot \pi$ at each time step. Further, all node movements are independent and not confined to a certain sub portion of the simulation area. The mobility model that is used in that study [9] is the random waypoint model in which a node randomly chooses a destination point and moves towards it with a certain velocity, pauses for a certain time when it reaches the destination and then chooses the next destination point. This behavior is sufficient to model a completely random behavior but certainly not suitable for the case of mobile agents whose primary aim is to analyze the network continuously and move to their next best location.

www.manaraa.com

2.5.1 Connectivity versus Routing

Although routing is a very important task in ad hoc networking, network connectivity (which determines the ability of the user nodes being able to form single or multi hop paths among themselves) is a more fundamental requirement [53]. If there are no possible links between the source and the destination nodes, communications will be disrupted no matter which routing protocol is used. This dissertation is primarily aimed at developing a method that maximizes the connectivity of the network such that communication disruptions due to link unavailability are minimized. However, the developed method will also be useful for routing protocols to maintain or generate routing tables as needed.

2.5.2 Connectivity and Performance Measures

A mobile ad hoc network at any instant can modeled as an undirected graph with nodes being the vertices and links being the edges, as given in [9]. If any two nodes are within each other's range, a link is formed between these two nodes in the ad hoc network. The principles of graph theory applied to telecommunication networks are also applicable to MANET. Since a MANET is a dynamic network with changing node locations, a discrete time model is used to represent the network state at any time t. Let UN be the set of the user nodes and AN be the set of mobile agents, and UN_t and AN_t be the sets of active user nodes and mobile agents at time t, respectively. Let graph $G_t = G(N_t,E_t)$ be an undirected graph with n_t nodes (vertices) and m_t edges (links), at time t. At

www.manaraa.com

any time t, the set $N_t = UN_t \cup AN_t = \{1,2,\ldots,n_t\}$ denotes the set of active nodes on the network, and set $E_t = \{1,2,\ldots,m_t\}$ denotes the set of established links between node pairs.

2.5.3 Basic Graph Theory

As discussed above, there are some graph measures that are useful indicators of the state and performance of a MANET [9]. They will be briefly summarized in the following sections.

2.5.3.1 Node Degree

The node degree of a node *i*, denoted by *d(i)*, is the number of neighboring nodes with a direct link to *i*. Another definition for the node degree of node *i* is the number of links it has. The minimum node degree of a graph *G* is defined as shown in equation (2-3).

$$d_{\min}(G) = \min_{\forall i \in G}\{d(i)\} \tag{2-3}$$

The average, or mean, node degree of a graph *G* is:

$$d_{mean}(G) = \frac{1}{n}\sum_{i=1}^{n} d(i) \tag{2-4}$$

which for undirected graphs is equal to:

$$d_{mean} = \frac{2m}{n} \tag{2-5}$$

المنارة للاستشارات
www.manaraa.com

Thus, the minimum, and the average node degree of graph G_t at time t is:

$$d_{\min}(G_t) = \min_{\forall i \in G_t} \{d(i)\} \tag{2-6}$$

$$d_{mean}(G_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} d(i) \tag{2-7}$$

or:

$$d_{mean} = \frac{2m_t}{n_t} \tag{2-8}$$

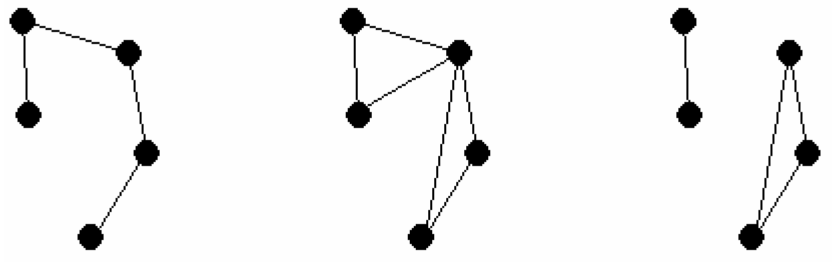
2.5.3.2 Graph Connectivity

Connectivity is defined either for a pair of nodes, or for the entire graph, or for the network. A graph is said to be connected if every node can be reached from every other node by traveling through the links between nodes, and it is fully connected if all node pairs have links between them. In a typical WLAN or WWAN it is sufficient for a mobile node to have a link to at least one access point or to a base station. In a MANET however, connectivity is a function of the number and locations of the nodes and the wireless transmission range [9].

A common way to represent connectivity is the *k*-connected synonym. If a graph is *k*-connected, then every node pair has at least *k* disjoint paths between them. For a graph to be connected all nodes should be at least 1-connected, i.e. the links of the graph should form at least a spanning tree.

المنارة للاستشارات
www.manaraa.com

A disconnected graph or an isolated node can result in degraded network performance. The main aim of the method proposed in this dissertation is to maintain at least a spanning tree at all times in the MANET. All links in the MANET are assumed bidirectional, i.e. the data can flow in any direction. Figure 2-2 represents three different graphs with different node degrees and connectivity properties.



(a) **(b)** **(c)**

Figure 2-2 (a) A connected graph, $d_{min} = 1$, $d_{mean} = 8/5$. (b) A 2-connected graph, $d_{min} = 2$, $d_{mean} = 12/5$. (c) A disconnected graph, $d_{min} = 1$, $d_{mean} = 8/5$.

2.6 MANET Mobility

The mobility of a MANET has been addressed by researchers in many different ways. For example, in Shukla's [80] and Camp *et al.*'s [18] studies, the average velocities of MANET nodes are taken as the mobility measure. In Ishibashi and Boutaba's work [53] the effect of maximum speed is strongly emphasized. Kwak *et al.* [58] propose a different measure called *remoteness* suggesting that the average or the maximum velocities are insufficient to reflect the movements of the nodes relative to each other. The remoteness measure that is proposed is a function of the distance between any two

المنارة للاستشارات
www.manaraa.com

nodes, and it assigns a greater importance to ones that are just at each other's range or near the border.

Ishibashi and Boutaba [53] consider the mobility and MANET topology relationship using a random waypoint model. Throughout the life of the network, links are created and broken as the nodes move in and out of the range of one another. There are different time definitions to describe the life of a link. The first is the optimum lifetime of the link. This is the time from when the nodes first move within each other's range so the link can be formed, until the link is broken when they move out of range. This is the maximum stable and usable period for the link. However, it is not the actual time that the link is available for use. In order for the link to be available for use, it has to be detected by a node. Similarly, the breakage of the link has to be detected and this happens when a neighbor does not respond for a certain timeout period. The time elapsed from the first detection to the link breakage detection is termed the perceived link lifetime, which usually extends beyond the end of the existence of a usable link. Data packets sent during that time are wasted effort. A final definition that is described in [53] is the time the link is first included in a path by the routing protocol. This process may occur at any time during the link's lifetime therefore the expected time to failure for the link, from the arbitrary time of route discovery, is half of the perceived link lifetime. Here, the term failure is used for the event that the link is lost due to the nodes moving out of range, not due to an equipment failure.

The quality of a link can be explained as its sustainable data transmission rate, which depends on the amount of signal attenuation mainly due to the distance between

المنارة للاستشارات

www.manaraa.com

nodes. This is explained in more detail in Section 4.3.1. Although the node density affects the transmission quality of the links, the lifetime only depends on the mobility model and the transmission range. Ishibashi and Boutaba show that average link lifetimes exponentially decrease with increasing maximum velocity. For a transmission range of 250 m, at a maximum speed of 5 m/s (18 km/h) the links last about 165 seconds on average, and only 40 seconds when the maximum speed is 108 km/h. These are average link lifetimes. They report that link lifetime distributions have long but light tails and a significant weight around near-zero lifetimes. This means that a significant portion of the links formed in a MANET with randomly moving nodes fail in a very short period of time.

Similarly, Chu and Nikolaidis analyze mobility versus connectivity of a MANET [25]. Their analyses reveal that the higher the velocities are, the better the connectivity. This might seem contradicting at first sight, due to the fact that the average link lifetimes are expected to be shorter as stated by Ishibashi and Boutaba [53], but the observed behavior is explained in terms of connectivity. The explanation is that at low speeds, nodes in weaker covered regions tend to stay longer and thus decrease the overall connectivity of the network over time. However, with increasing speeds, the link lifetimes could become shorter but new links are formed as the older ones dissipate, and the node distribution tends to be more uniform, both contributing to overall better connectivity. The phase transition phenomena is also seen in Chu and Nikolaidis's paper [25] with changing wireless transmission range.

المنارة للاستشارات
www.manaraa.com

Stepanov and Rothermel [83] proposed an urban scenario simulation for a MANET to take into account mobility and wireless transmission differences that are realized in city environments. The mobility model considers movement constraints, obstacles, road networks and the transmission model considers propagation in city areas. Their study shows that realistic simulations for urban environments differ from simpler models and provide a better estimation of urban performance. However, this comes at the expense of computational complexity and the requirement of detailed data to reflect the physical conditions of the simulation area.

2.7 Future Location Prediction

There have been a few studies that investigate prediction of future locations of mobile users. The interest in estimating the future locations of users in wireless telecommunication networks falls in two main categories; 1) the cell that the user will enter in cellular networks, and 2) the future geographical location of the user or users, in ad hoc networks.

Papers that address the first group include: [10, 61, 62, 63, 66, 93]. Papers that address the latter group include: [8, 30, 67, 84, 85, 86, 89]. Discussion about these studies are presented below.

2.7.1 Cellular Models

Liu and Maguire model the movement of mobile users within cells as movement circle (MC), movement track (MT) and Markov chain models [61, 62]. The MC model is

المنارة للاستشارات
www.manaraa.com

based on the assumption that users will eventually return to their initial positions. The MT model is a uni-directional model, less constrained than the MC model. They use the MC and MT models to describe the regular or structured movements of the users, and the Markov chain model to describe additional randomness.

A similar two-level model is used to describe human motion in a cellular environment by Liu *et al.* in [63]. The top level is a global mobility model (GMM) whose resolution is in terms of cells crossed by the mobile user rather than user coordinates. The second level is a local mobility model (LMM) which is used to describe the movement within a cell, using speed, direction and position information. GMM is a deterministic model whereas LMM is a stochastic model that interacts with the GMM model. The GMM is motivated by the fact that the users show some regular patterns during daily movements.

Yavaş *et al.* propose a data mining approach to extract inter-cell movement history regularities and combine that with the current trajectory to estimate the next position in [93]. A similar data mining application to location prediction is proposed by Ming-Hui *et al.* in [66]. Bilurkal *et al.* [10] propose a neural network (NN) algorithm to predict the next location of users. They emulated 6 weeks of data with 30 observations per day of (time, *x*-coordinate, *y*-coordinate) and trained a NN with backpropagation using the next *x* and *y* coordinates as the output.

المنارة للاستشارات

www.manaraa.com

2.7.2 Ad hoc Models

Wang and Chang [89] propose a mobility prediction model to be used for a reliable routing protocol. Their model is based on the assumption that the position and velocity of a node is known at some time t, the path loss is a free-space loss and all devices have the same wireless transmission range R. A node at (x,y) at time t_o, is expected to be in a circular region with center (x,y) and a radius of $v \cdot (t_1 - t_o)$ at time t_1. By using this circular region to find the farthest possible point that the node can be at, and assuming a constant velocity and direction between $(t_1 - t_o)$, the estimated link duration time between any pair of nodes can be calculated. This information is then utilized to route packets via longer duration links. The same principles are used by Su *et al.* and Tang *et al.* [85, 86].

Ashbrook and Starner [8] propose a learning algorithm for significant locations and motion prediction with GPS. GPS was used to record data for a period of 4 months in Atlanta, GA. They only recorded the location data when the subjects were steady. Thus, the stations, not the motion, is of interest in their model. They use a Markov model with transitions from each location to another.

Mitrovic [67] proposes a model to predict short term user motion to help vehicle navigation. A time-delay neural network is developed which allows information about signal history be available as an input to the NN. He uses longitudinal and lateral acceleration data gathered from two accelerometers, vehicle rotation data, changes in the road slope, and GPS position data as inputs.

Creixell and Sezaki [30] propose a time series method with the least squares lattice (LSL) method to estimate the parameters. The time series to represent the trajectory are

www.manaraa.com

$v=\{v_0, v_1, v_2,\ldots, v_n\}$ and $Q=\{Q_0, Q_1, Q_2,\ldots, Q_n\}$, where Q is the amount of displacement angle from the horizontal axis, and v is the velocity vector. The prediction model is given in equation (2-9)and (2-10).

$$v_{i+1}=\alpha_1^{v}(\mathrm{i})v_i+\alpha_2^{v}(\mathrm{i})v_{i-1}+w^{v} \tag{2-9}$$

$$Q_{i+1}^{d}=\alpha_1^{Q}(\mathrm{i})Q_i^{d}+\alpha_2^{Q}(\mathrm{i})Q_{i-1}^{d}+w^{Q} \tag{2-10}$$

The method does not use the first 20 observations for prediction because LSL needs about 20 iterations to converge. The prediction horizon is 10 steps into the future.

As a summary, it can be stated that the first few papers by Wang and Chang [89] Su *et al.* [85] and Tang *et al.* [86] utilize a simple location and velocity based expected position to help the routing protocol. The method does not make use of direction, change in direction nor change in velocity. The significant location learning approached proposed by Ashbrook and Starner [8] only predicts the next important station that the user will be in, and does not utilize velocity or direction information. It also is only functional over the specific area that is used for the learning. The NN model proposed by Mitrovic [67] is aimed at predicting car motion. It is trained only using specific maneuvers on certain road conditions. However, the proposed neural network approach can be adapted for a more generalized motion pattern. Although satisfactory results are achieved, the time series method of Creixell and Sezaki [30] has a time series parameter prediction problem at every time step, adding to the computational burden.

www.manaraa.com

In a more recent study, Huang and Zaruba [52] proposed a method that enables non GPS equipped ad hoc nodes to estimate their approximate locations by using the information from GPS equipped nodes. This would be advantageous in situations where GPS equipment or satellite signals are unavailable. Their model involves multiple GPS enabled nodes. Other nodes on the MANET approximate their locations by using the known node location data and the signal strength between them to estimate a location distribution.

2.7.3 Kinematics Approach

Motion is inherently continuous. Where an object stands at a time instant greatly depends on where it was a moment ago, and is highly correlated with the space it will occupy moments later. Kinematics is a branch of mechanics which describes the motion of objects only by means of geographical coordinates, i.e. with no consideration of the forces acting on the bodies.

The position of an object is described by its coordinates. The rate of change of position is defined as the velocity and the rate of change of velocity is described as the acceleration of an object. By using the velocity and the acceleration information, it is possible to calculate how the position of an object changes.

In this study, a location prediction method based on kinematics principles is developed for MANET users. The location prediction system is integrated into the mobile agent location optimizer and it enables the system to utilize past user location

المنارة للاستشارات

www.manaraa.com

information. The details of location prediction using kinematics and its effects on algorithm performance are shown in Section 3.4 and Section 7.3.2.

2.8 Heuristic Optimization with Evolutionary Algorithms: Genetic Algorithm and Particle Swarm Optimization

Perhaps the most commonly used general purpose heuristics are evolutionary algorithms (EA) that mimic the dynamics of natural evolution where the fittest individuals survive and transfer their genetic information to the future generations.

2.8.1 Genetic Algorithms

Genetic algorithm (GA) is one of the evolutionary computation methods that researchers use when attempting to find approximate solutions to large, complex problems. GA was introduced by Holland, and its performance on both combinatorial and continuous problems has been studied extensively [31, 44, 51, 65, 78].

A genetic algorithm maintains a population of individuals and applies selection, crossover and mutation to the population over generations mimicking the natural evolution process. The individuals in the population are represented as a string of digits or alphabetical characters, synonymous with the genotype. A typical binary represented multi-variable individual is given in Figure 2-3.

المنارة للاستشارات
www.manaraa.com

$$\underbrace{1100101}_{\text{Var 1}}\underbrace{1010110}_{\text{Var 2}}\underbrace{1001100}_{\text{Var 3}}\ldots\underbrace{1010110}_{\text{Var } n}$$

Figure 2-3 A binary represented multi-variable solution.

In each generation, individuals with relatively better fitness values are given higher chances to mate with each other and transfer parts of their genotype to the children in the following generations. The fitness of an individual is correlated to its objective function value. The correlation should be positive for maximization problems and negative for minimization problems. To calculate the fitness of an individual, first its genotype needs to be converted from encoding space to the variable space, or phenotype, using a decoding function. A binary bit representation for the chromosomes is common among GA researchers. A typical decoding function from binary to real space is given in equation (2-11).

$$decode(c_i) = \frac{(x_{ub_i} - x_{lb_i}) \times decimal(c_i)}{2^{l_i} - 1} + x_{lb_i} \qquad (\,2\text{-}11\,)$$

where c_i is the chromosome, x_{lb_i} is the lower bound, x_{ub_i} is the upper bound of the i^{th} variable, l_i is the length of the c_i, and *decimal*(c_i) is the decimal value of c_i as in equation (2-12).

$$decimal(c_i) = \sum_{j=1}^{l_i} c_{i_j} \times 2^{j-1} \qquad (\,2\text{-}12\,)$$

where c_{i_j} is the j^{th} bit of c_i.

www.manaraa.com

Populations generally become fitter through the generations as a result of the mimicked evolution. Initial populations are often created randomly. Search for the global optimal is conducted by selecting parent members from the parent population and letting them create an offspring population by combining their genetic information. Combination of genetic information is done by means of crossover operators. The parent members go through crossover with a certain crossover probability, p_c. The offspring are mutated after crossover, which is also a natural phenomenon. Parent and survivor selections follow certain rules, which can differ from one GA application to other. In general, there are random, roulette wheel, and tournament selection methods for parent selection. Some examples of the crossover operators are single-point, multiple point and uniform crossover. Examples of single point and uniform crossover are given in Figure 2-4.

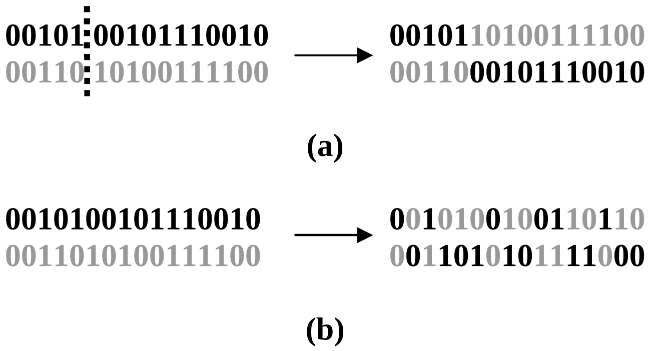


Figure 2-4 (a) Single point crossover. (b) Uniform crossover.

Survival selection can follow either a generational strategy or a steady-state strategy. In the generational strategy the entire population is merged with the offspring population, whereas in the steady state survival strategy offspring merge with existing

المنارة للاستشارات
www.manaraa.com

members in the parent population immediately after being created. GAs are often elitist, i.e. the best individual(s) in the population is(are) preserved and usually not mutated from one generation to another.

Population based heuristics have been applied to dynamically changing objectives in the literature [14, 15, 92]. Since the movements of mobile network nodes create a different topology within the proximity of the previous topology in successive time increments, the change in objective function is also incremental. Further, the new optimal locations of the mobile agents will be within the proximity of their previous locations due to velocity and geographical constraints. Once the population is stabilized, GA's or any other EA's response to an incremental change in the objective function is expected to be relatively fast, benefiting from previous superior solutions [92]. More detailed information about EAs in dynamic environments is given in Section 2.9, and the analysis of this behavior can be found in Section 7.3.3.

2.8.2 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population based optimization tool which emulates the social behavior of species that live in the form of swarms in the nature. These swarms are capable of exchanging valuable information such as food locations in the habitat. PSO was developed by Eberhart and Kennedy in 1995 [56]. The swarm particles in the algorithm communicate and direct the search towards areas in the search space with better fitness values.

المنارة للاستشارات

www.manaraa.com

PSO has many common aspects with evolutionary algorithms. Like a GA, PSO has a population of randomly initialized candidate solutions. Different than the evolutionary algorithms, the members of a PSO population do not mate or mutate to create offspring. Instead, they swarm over the search space by moving in the solution hyper plane while communicating with each other and using the information from superior individuals in the swarm as well as their own best positions in the past. The value of their positions are evaluated in terms of the objective function.

2.8.2.1 The PSO Mechanism

A swarm particle changes its velocity at each iteration, or time step, aiming towards the superior particle in the neighborhood and its best history. This change in the particle velocity is also weighed by random factors to provide a robust and diverse search.

Each member particle in the swarm is represented by three vectors ***X***, ***P*** and ***V***. Vector ***X*** represents the current particle location, ***P*** represents the location of the particle's historic best fitness and ***V*** is the velocity vector that defines the direction and magnitude the particle will travel if not disturbed. ***V*** is used to update ***X*** every iteration. The swarm has a global or neighborhood best fitness location vector, ***G***, which is used in conjunction with individual ***P*** vectors while updating particle coordinates.

Maintaining the ***G*** vector relies on a communication scheme within the swarm. As mentioned above, the ***G*** for a particle is the best found in its neighborhood of particles and ***G*** is the global best if the swarm employs a global neighborhood. Different neighborhood topologies have been studied in the literature and global neighborhoods seem to perform better in terms of computational costs [19].

www.manaraa.com

The Pseudo code of the PSO mechanism is given in Figure 2-5. In Figure 2-5, the term ω is the inertia factor, c_1 and c_2 are the cognition and the social coefficients, respectively, and U(a,b) is a uniform random number between [a, b].

```
Initialize population {
  $X = U(X_{min}, X_{max})$
  $V = U(V_{min}, V_{max})$
  $P = X$
}
Do While (Stopping criteria not met) {
  $V = \omega \cdot V + c_1 \cdot U(0,1) \cdot (P - X) + c_2 \cdot U(0,1) \cdot (G - X)$
  $X = X + V$
  if( $f(X)$ is better than $f(P)$ ) then $P = X$
  if( $f(X)$ is better than $f(G)$ ) then $G = X$
}
```

Figure 2-5 Pseudo code for basic PSO mechanism.

The PSO has successfully been applied to problems in the continuous domain. It has few parameters that require adjustment, which makes the development process relatively easy and fast. Implementation is also easy due to its simple but robust mechanism. PSO has also been applied to dynamic problems. Eberhart and Shi [37]

المنارة للاستشارات
www.manaraa.com

tested PSO on random error introduced problems with multi variables. They show the ability of PSO to track the changing objective successfully. Carlisle and Dozier [20] modified the memory property (***P*** vector maintenance strategy) of the swarm for dynamically changing environments. When a change in the problem environment is detected, all memory positions are reevaluated and set to either the old memory or to the current particle position, whichever is better. Their results show that the PSO can successfully track a time dependent objective function.

2.8.2.2 Enhancing PSO’s Performance

Since the major search component in the PSO is the modification of particle velocities, controlling the changes in the velocity is a major issue. If left unbounded, magnitudes of the particle velocities can reach quite large numbers [56]. There are two main methods developed to control the changes in the velocities:

1) Implementing a dynamically adjusted inertia coefficient
2) Using a constriction coefficient

The inertia method employs a dynamically changing ω coefficient. Initially, ω is set to 1 and is decreased gradually as the PSO iterations advance [56]. With a relatively high inertia coefficient, the current direction and magnitude of the particle’s motion are weighed highly. As the iterations advance and the inertia coefficient is decreased,

www.manaraa.com

changing the direction and magnitude of the particle velocities toward the self and global best particles become easier.

The constriction coefficient was developed by Clerc in 1999 [26]. The constriction coefficient K improves PSO's ability to control the growth in velocity magnitudes. It scales the velocity updates such that a theoretical convergence is guaranteed. It has been found that K combined with V_{max} constraints improved the PSO performance significantly [36]. The constriction coefficient K and its application to control PSO velocities are given as:

$$V = K \cdot \left[v + \varphi_1 R_1 \cdot (P - X) + \varphi_2 R_2 \cdot (G - X) \right] \tag{2-13}$$

$$\varphi = \varphi_1 R_1 + \varphi_2 R_2 \tag{2-14}$$

$$K = \begin{cases} \dfrac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}} & \text{for } \varphi > 4 \\ 1 & \text{otherwise} \end{cases} \tag{2-15}$$

In equation (2-15), R_1 and R_2 are random numbers drawn from a uniform distribution between [0, 1].

2.9 Evolutionary Algorithms in Dynamic Environments

In many real world optimization problems, the objective function, the problem instance or the constraints may change over time, also changing the optimal solution to

المنارة للاستشارات
www.manaraa.com

the problem [13, 15, 16, 92]. Yaouchu *et al.* gives four main reasons that problem uncertainties might be taken into account:

1) Noise: The fitness function is subject to noise. This can happen due to sensory measurement errors or randomized simulations.

2) Robustness: The design variables are subject to perturbation after the optimal is determined. The solution is expected to be robust and still be satisfactory with the changed design variables.

3) Fitness approximation: The fitness function is very expensive to calculate exactly. A simpler meta-model is used to approximate the fitness function value.

4) Time-varying fitness functions: The fitness function is deterministic at any point in time, but is dependent on time *t*, as shown in equation (2-16),

$$F(X) = f_t(X) \tag{2-16}$$

Therefore, the optimum also changes over time. The optimization algorithm is expected to track and locate the changing optimum in each time step. The challenge is to reuse the information from the previous environments to speed up the solution process.

المنارة للاستشارات
www.manaraa.com

The mobile agent location optimization problem in a MANET falls into the fourth category in the above classification. The successive location problem at each time step should be solved in real time in an actual application.

In dynamic optimization algorithms, an explicit “solve from scratch” approach can be time consuming. Using previously gained knowledge about the search space can speed up the next optimization process. If the new optimal solution is guaranteed to be within a certain distance of the old one, then restricting the search to only that space will certainly be beneficial [92]. As described in the previous section, the new optimal locations of the mobile agents need to be within the proximity of their previous locations due to maximum velocity and possible geographical constraints. This inherent characteristic makes the mobile agent motion optimization problem suitable for dynamic environment solution methods.

Many ways can be devised to transfer knowledge from the previous search space. A common way is to keep the individuals in the final population of the previous problem state [14, 92]. However, explicit actions or strategies are needed to increase diversity and facilitate the shift of the population towards the new optimum when a change in the environment occurs. There are various ways to accomplish this. The EA can be run in a standard fashion, but the diversity can be increased for a short period of time after a change is observed. Some examples of this strategy are hypermutation [27] and variable local search [88], where the mutation rate is gradually increased after a change in the environment is detected. Another method is to maintain diversity throughout the runs. This can be accomplished by accepting random individuals, i.e. random immigrants [45],

www.manaraa.com

into the population at every generation. Memory-based approaches are useful when the optimum repeatedly returns to past stages [68, 79].

Some meta-heuristics have been applied to dynamic optimization problems. The particle swarm optimization (PSO), which is described in Section 2.8.2, is among those. More information on evolutionary heuristic algorithms for problems with changing environments can be found in the recent survey paper [92].

Considering all the discussions above, a GA and a PSO with dynamic objective functions are developed as the mobile agent location optimizer heuristics, to dynamically manage the motion of a number of mobile agents in the MANET in order to maximize its connectivity. Detailed model of the proposed method is given in Chapter 3, followed by the detailed descriptions of the GA and the PSO implementations in Chapter 6.

المنـــارة للاستشارات

www.manaraa.com

CHAPTER 3

THE PPROPOSED MANET MANAGEMENT SYSTEM: PROBLEM DESCRIPTION AND MATHEMATICAL MODEL

A MANET management system is developed that helps maintain connectivity by using a number of controlled ad hoc network nodes (agents). Brief descriptions of the proposed system, the problem it solves, and its solution operators are given in this chapter.

3.1 Introduction

The proposed method consists of managing the directions and magnitudes of velocities of a group of mobile agents that have predefined wireless communication capabilities similar to the other mobile nodes that form the MANET. The agents actually become an integral part of the ad hoc network. Their movements and thus their locations are remotely controlled dynamically as the entire MANET topology changes to optimize network connectivity. To our knowledge, such a method or algorithm for MANET networks has not yet been proposed.

Since the ad hoc network nodes are mobile, only their current and past location data are available. The proposed method is designed to make use of the current and the past

www.manaraa.com

location data available by the use of a global positioning system (GPS). GPS is a satellite based system that can provide readily available and accurate position information almost anywhere on Earth. Many GPS implementations are available including integrated GPS receivers in mobile phones or mobile network devices [24]. To achieve a real time response to the changing network topology, a fast and dynamic algorithm is required to continuously optimize the locations of the mobile agents. This problem is a complex, non-linear problem that requires a heuristic algorithm with a continuously changing objective function. A population based heuristic with a time varying fitness function is therefore applied as the heuristic optimizer.

3.2 The Proposed MANET Management System

There are two main types of MANET nodes; user nodes and agent nodes. User nodes are the nodes that demand network service. Mobile agents are responsible for helping the user nodes experience the best network service possible. The user nodes in the MANET move at their own will and it is assumed that their future positions are unknown. Also, location data is assumed to be available to the agent control system for all times that there exists a communication path between a node nodes and the control system. This is technically possible by broadcasting the location information provided by the GPS. Finally, every node has a certain wireless connection range and a maximum velocity.

المنارة للاستشارات
www.manaraa.com

3.3 The Mobile Agent Location Problem

The locations of the mobile agents are under control of a location optimizer that is responsible for maximizing the performance objective as a function of the current coordinates of the nodes in the MANET. The objective function gauges performance that the user nodes experience in terms of being able to communicate with other users and the speed of the data transmission rates. Only the users are used to assess network performance and only the mobile agent movements are controlled by the centralized optimizer.

The problem, which is formulated below, is a non-linear problem and is appropriate for a heuristic algorithm. Heuristic optimization algorithms need an objective function that responds well to the decision variables. Measures such as minimum node degree or connectedness usually show sudden changes in certain regions of the search space, depending on the graph's characteristics. These measures typically show a steady or flat behavior over large portions of the search space. This is a phenomenon known as phase transition as mentioned earlier [38]. In order to overcome this issue, an objective function is needed that is responsive to small changes in mobile agent locations and that also reflects network connectivity and performance. This is accomplished using a maximum flow approach as detailed in Section 3.3.1.

3.3.1 The Maximum Flow Analogy

In a wireless network, a link's performance depends on the signal strength, which is a function of the link distance and some external factors. In general, the link distance and

المنارة للاستشارات
www.manaraa.com

signal strength, and thus the link data rate, are negatively related. The wireless IEEE 802.11 standard is capable of linking MANET nodes [41]. Using this protocol, it is technically possible to create multi-hop networks that cover several square kilometer areas [24]. The 802.11 standard operates at 2.4 GHz, or for some applications at 5.0 GHz. The signal attenuation for 2.4 GHz in free space environments is given in [64], as in equation (3-1).

$$Path\ Loss = 32.4 + 20 \cdot log(f) + 20 \cdot log(d_{ij}) \qquad (3\text{-}1)$$

where f is the frequency in megahertz (MHz), and d_{ij} is the distance between nodes i and j in km, and *Path Loss* is in decibels (dB).

The path loss model is used along with a product specification sheet of a wireless access point manufactured by one of the industry leaders to calculate the data transfer rate versus distance [69]. The path loss versus data rate chart is given in Table 3-1.

Table 3-1 Path Loss verus Data Rate

Data rate (Mbps)	Receive Sensitivity (dBm)
54	-75
48	-76
36	-80
24	-84
18	-88
12	-90
9	-90
6	-93
2	-93

المنارة للاستشارات
www.manaraa.com

If a normalized wireless transmission range is considered, equation (3-2) provides a reasonable normalized data rate estimation for location optimization purposes.

$$DataRate(i,\,j) = \left(1 + e^{10\cdot\left(d_{ij} - 0.5\right)}\right)^{-1} \tag{3-2}$$

where d_{ij} is the Euclidean distance between nodes *i* and *j*.

The function given in equation (3-2) may not be the most accurate estimation of the normalized data rates at intermediate distances, but the path loss and the data rate estimation models are all estimates assuming constant interference and certain environmental conditions. Also, technical capabilities such as antenna reception of devices differ. The device that is presented here is just example. It is expected that the manufacturer’s specification curve in Figure 3-1 will shift left or right for different products. This is why a general centralized estimation curve is devised. The data rate function conforms to the basic requirements of a normalized distance versus data rate. When the distance is close to zero, the normalized data rate should be close to one, and when the distance is close to one, i.e. the distance is close to the wireless transmission range, the data rate should be close to zero. Also exponential decrease of the data rate occurs as the distance increases, as observed in practice. The graph of the data rate estimation function is given in Figure 3-1 with comparison to a device manufacturer’s specifications.

المنارة للاستشارات
www.manaraa.com

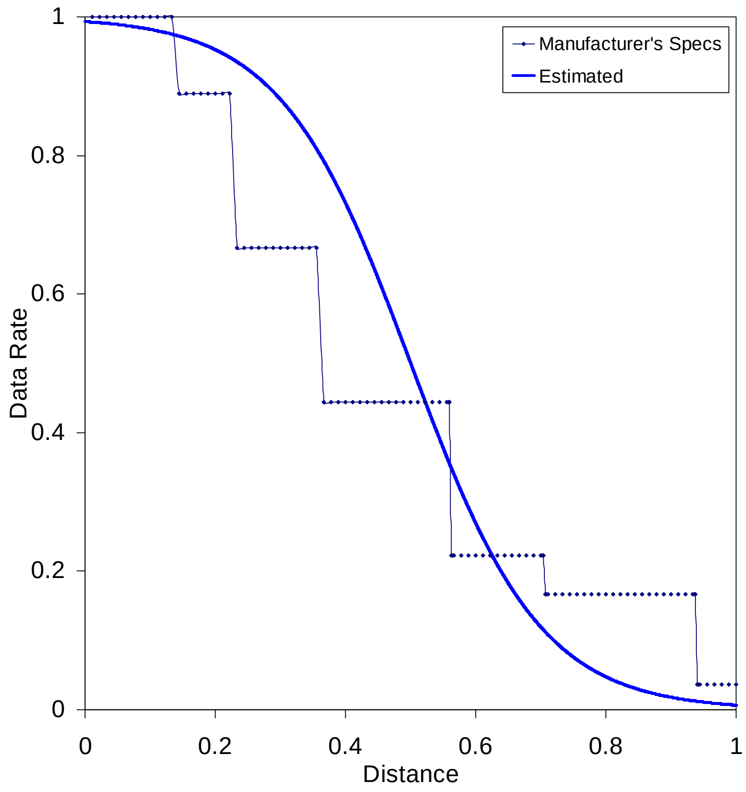


Figure 3-1 Normalized distance vs. normalized data rate, range = 450 m.

At any time *t*, a MANET is modeled as a transportation network with flow capacities equal to the data rates of the wireless links. An intuitive first measure is then the maximum flow values between the pairs of nodes. The maximum flow values between every node pair give a good sense of the overall network performance. Trying to maximize the minimum of those maximum flows between every user pair is a responsive objective function, and is suitable for the mobile agent location optimizer. The maximum flow problem is a well known network optimization problem and there are various algorithms readily available to optimally solve it, including ones in polynomial time [2, 40]. Maximizing the minimum of maximum flow values between user node pairs is very

www.manaraa.com

similar to a problem that exists in the literature, the all-pairs maximum flow (or minimum-cut) problem [2, 6, 49, 50].

Let's consider a capacitated network $G_t = (N_t, E_t)$ with a non-negative capacity u_{ijt}, associated with each link (i,j) at time t. Further, two special nodes in network G_t are specified; a source node S, and a target node T. The formulation of the maximum flow problem between the source S and the target T at time t is as follows:

$MaxFlow(G_t,S,T)$ = Maximize f

Subject to

$$\sum_{\{j:(i,j)\in E_t\}} x_{ij} - \sum_{\{j:(j,i)\in E_t\}} x_{ji} = \begin{cases} f & \text{for } i = S, \\ 0 & \forall\, i \in N_t - \{S \text{ and } T\}, \\ -f & \text{for } i = T \end{cases} \tag{3-3}$$

$$0 \le x_{ij} \le u_{ijt} \tag{3-4}$$

where x_{ij} is the amount of flow from node i to node j and u_{ijt} is the capacity of link (i,j) at time t.

3.3.2 The Mathematical Model

Notation

UN_t	the set of user nodes at time t
AN_t	the set of mobile agent nodes at time t
N_t	the set of all MANET nodes at time t, $N_t = UN_t \cup AN_t$

المنارة للاستشارات
www.manaraa.com

n_{u_t}	the number of mobile user nodes at time t, $\lvert UN_t \rvert$
n_{a_t}	the number of mobile agent nodes at time t, $\lvert AN_t \rvert$
n_t	the total number of MANET nodes at time t, $\lvert N_t \rvert$
E_t	the set of links between all MANET nodes at time t
t_o	initial time
R_{it}	the wireless connection range of the i^{th} node at time t
(x_{it},y_{it})	x and the y coordinates of the i^{th} MANET node at time t
XY_t	$\{(x_{it},y_{it}) : i \in N_t\}$, the set of x and y coordinates of the MANET nodes at time t
XY_{destin}	$\{(x_{jdestin},y_{jdestin}) : j \in UN\}$, the set of x and y coordinates of user node final destination points
(X_{min}, X_{max})	x-axis boundaries
(Y_{min}, Y_{max})	y-axis system boundaries
r_{it}	rotation angle (rad, counter clockwise) from the x-axis of the i^{th} mobile agent at time t
v_{it}	the speed of the i^{th} mobile agent at time t
$v_{\max_i}$	the maximum speed of node i user or agent
$e_{t_{(i,j)}}$	1 if there exists a link between (i,j) at time t, 0 otherwise
u_{ijt}	the capacity of the link between (i,j) at time t, i.e. *DataRate*(i,j)
$U(a,b)$	uniformly distributed random number between a and b
z_{ijt}	1 if there exists a path between (i, j) at time t

www.manaraa.com

M	a big enough number for computational use

For any time t, and set $XY_t = \{(x_{it},y_{it}) : i \in N_t\}$, the graph $G_t = (N_t,E_t)$ is formed as follows:

$$e_{t_{(i,j)}} = \begin{cases} 1 & \text{if } R_{it} \geq d_{ijt} \text{ and } R_{jt} \geq d_{ijt} \\ 0 & \text{otherwise} \end{cases}$$

(3-5)

where $i,j \in N_t$ and d_{ijt} is the Euclidean distance between nodes i and j at time t as given in equation (3-6).

$$d_{ijt} = \sqrt{(x_{it} - x_{jt})^2 + (y_{it} - y_{jt})^2}$$

(3-6)

The formation of a link is a function of the signal attenuation between the nodes, and can depend on factors other than distance. In that case, the proper attenuation model will replace equation (3-2). The remainder of the model will not be affected. The mathematical model for the mobile agent location optimizer is then written as given in equations (3-7) through (3-9).

المنارة للاستشارات
www.manaraa.com

$$O_1 = \min_{i,j \in UN_t : j > i} \{ MaxFlow(G_t, i, j) : MaxFlow(G_t, i, j) > 0 \}$$

$$O_2 = \frac{\sum_{i,j \in UN : j > i} MaxFlow(G_t, i, j)}{n_u \cdot (n_u - 1)/2}$$

$$O_3 = \left(\sum_{i,j \in UN : j > i} z_{ijt} \right) \cdot M$$

Maximize $O_1 + O_2 + O_3$

(3-7)

Subject to

$$0 \le r_{it} \le 2\pi \qquad \forall i \in AN_t$$

(3-8)

$$0 \le v_{it} \le v_{\max_i} \qquad \forall i \in AN_t$$

(3-9)

$$z_{ijt} = \begin{cases} 1 & \text{if there is a path between the } i^{th} \text{ and the } j^{th} \text{ user at time } t, \\ 0 & \text{otherwise} \end{cases} \qquad \forall i, j \in UN_t$$

3.3.3 The Objective Function

The term O_1 in equation (3-7) is the user pair with the worst possible maximum flow value. O_2 is the total maximum possible flows between all user pairs, scaled down by the maximum number of possible direct links among the users. This scaling ensures that O_2 is not given more importance than O_1. Finally, the connectivity term O_3 ensures that no communication path between a user pair is sacrificed for better O_1 or O_2 values. This gives connectivity the greatest importance among the three factors.

www.manaraa.com

An algorithm is coded for the calculation of the maximum flow value between all user node pairs and is invoked during each objective value calculation. The maximum flow between the pairs of nodes is calculated by an implementation of the highest-label push-relabel algorithm. The push-relabel algorithm was proposed by Goldberg and Tarjan [43]. Cheriyan and Maheshwari [22] show that the algorithm runs in $O\left(n^2\sqrt{m}\right)$ which is a tighter bound than the $O\left(n^3\right)$ which Goldberg previously stated [42]. The implemented version is the most efficient maximum flow algorithm in practice [2] and is available from the BOOST C++ libraries, which is a peer-reviewed, freely available software library collection [12].

The *MaxFlow* algorithm is called to calculate the flow between every user pair unless a direct link exists in between with a larger data transmission capacity than the user pair with the current lowest maximum flow. The Pseudo code is given in Figure 3-2. The F_t^{Min} and F_t^{Tot} are used to calculate the O_1 and O_2 values in the objective function.

المنارة للاستشارات
www.manaraa.com

```
Start{
    Set $F_t^{Tot} = 0$
    Set $F_t^{Min} = M$
    for ( $S$ in $UN_t$ = 1 to $n_u$-1 ){
        for ( $T$ in $UN_t = S + 1$ to $n_u$ ){
            If ( $e_{S,T} = 1$ AND $u_{S,T} > F_t^{Min}$ ) then {
                $F_t^{Tot} = F_t^{Tot} + u_{S,T}$
            }
            Else {
                $F_{ST} = MaxFlow(G,S,T)$
                $F_t^{Tot} = F_t^{Tot} + F_{ST}$
                If ( $0 < F_{ST} < F_t^{Min}$ ) then {
                    $F_t^{Min} = F_{ST}$
                }
            }
        }
    }
}End
```

Figure 3-2 The pseudo code for the components of the objective function calculation

In the worst case, the $MaxFlow(G_t,S,T)$ algorithm is executed for a total of $n_{u_t}(n_{u_t}-1)/2$ times. The all-user-pairs maximum flow calculation is made more efficient by excluding node pairs that are guaranteed to have a larger flow than a known lowest maximum flow pair.

www.manaraa.com

3.3.4 The Mobile Agent Velocity Constraints

There is an important advantage of implementing a polar coordinate system when optimizing the mobile agent relocations. If the rectangular coordinates of the agents are taken as the decision variables, satisfying the velocity constraints would require calculation of Euclidean distances and taking necessary measures within the algorithm such as penalization of unwanted or infeasible solutions. On the other hand, using polar coordinate components for the direction and magnitude of agent velocity vectors resolves this issue.

The velocity vector rotation r_i bounded by $[0, 2\pi]$ and the magnitude v_i bounded by $[0, v_{max}]$ allow the search method to move the mobile agents freely within a circle of radius v_{max}, thus automatically complying with the velocity constraint. The Cartesian coordinates at time (t+1) can then be calculated as given in equations (3-10) and (3-11).

$$x_{i(t+1)} = x_{it} + cos(r_{it}) \cdot v_{it} \qquad \forall i \in AN_t \tag{3-10}$$

$$y_{i(t+1)} = y_{it} + sin(r_{it}) \cdot v_{it} \qquad \forall i \in AN_t \tag{3-11}$$

Once the Cartesian coordinates of the mobile agents are known for time t+1, then the graph G_{t+1} can be drawn and its connectivity and data flow capacity properties can be calculated for the objective function.

المنارة للاستشارات
www.manaraa.com

3.4 Future Location Prediction Using Kinematics

Since the MANET users envisioned in this study are allowed to move freely within no preset boundaries or paths, an accurate and practical future position estimation method is developed by making use of the laws of kinematics. The only data needed for the future position prediction of an ad hoc user is its position history from three time steps back. With position data at each time step from time (*t*-3), it is possible to calculate the rate of change of acceleration, which is equivalent to the third derivative of the position. Any older time observations do not affect the practical accuracy of future location prediction. The GPS systems that are assumed to be available to all MANET users provide accurate position information that is used by the location prediction method. The components of the kinematics based location prediction method are given below:

$$v_{t-2} = (x, y)_{t-2} - (x, y)_{t-3} \tag{3-12}$$

$$v_{t-1} = (x, y)_{t-1} - (x, y)_{t-2} \tag{3-13}$$

$$v_{t} = (x, y)_{t} - (x, y)_{t-1} \tag{3-14}$$

$$a_{t-1} = \frac{v_t - v_{t-1}}{\Delta t} \tag{3-15}$$

$$a_{t-2} = \frac{v_{t-1} - v_{t-2}}{\Delta t} \tag{3-16}$$

www.manaraa.com

$$\Delta a_{t-1} = \frac{a_{t-1} - a_{t-2}}{\Delta t}$$

(3-17)

$$\hat{a}_t = a_{t-1} + \Delta a_{t-1} \cdot \Delta t$$

(3-18)

$$\hat{v}_{t+1} = v_t + \hat{a}_t \cdot \Delta t$$

(3-19)

$$\Delta xy_t = v_t + \frac{1}{2} \hat{a}_t \cdot \Delta t$$

(3-20)

$$XY_{t+1}^{p} = XY_t + \Delta xy_t$$

(3-21)

In the formulations of equations (3-12) through (3-19), v_t indicates the calculated velocity for time t, a_t indicates the change in velocity, i.e. acceleration, between time t and t+1. Δa_t is the rate of change of acceleration between t-1 and t. Finally, Δxy_t is the change in x and y coordinates, and XY_{t+1}^{p} is the set of predicted x and y coordinates at time t+1.

```
Set $t$ = current time
for ( $repeat$ = 1:$H$ ){
        $XY_{t+1}^{p} = XY_t + \Delta xy_t$
        $XY_{t+1} = XY_{t+1}^{p}$
        $t = t+1$;
}
```

Figure 3-3 Pseudo code for predicting the location at time (t+H)

المنارة للاستشارات

www.manaraa.com

Figure 3-4, Figure 3-5, Figure 3-6, Figure 3-7 and Figure 3-8 represent example cases of future location prediction of a single user in a time frame of 100 time steps for prediction horizons (*H*) of 0, 2, 4, 6 and 8 time steps, respectively. An *H* value of 0 means no prediction is performed.

In Figure 3-5, Figure 3-6, Figure 3-7 and Figure 3-8, the trajectories marked by + show the predicted locations of the user. Increasing prediction error is observed clearly as *H* increases from 2 to 8 time steps.

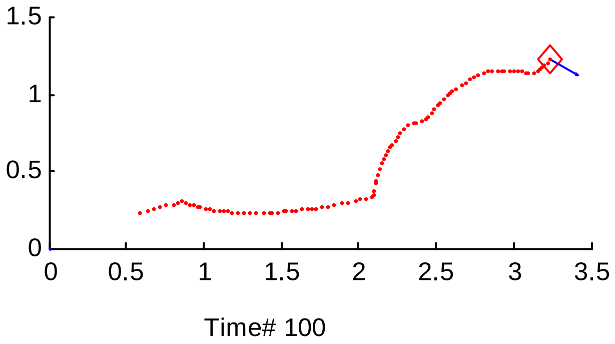


Figure 3-4 Real trajectory (*H* = 0)

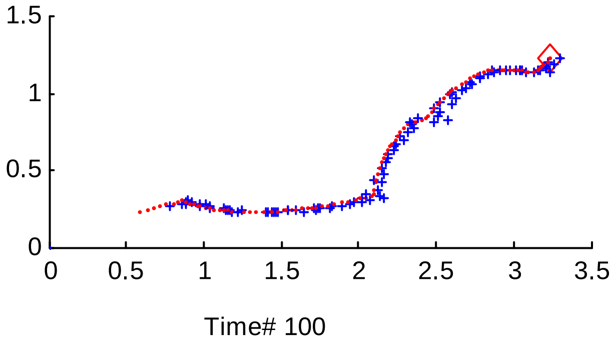


Figure 3-5 Location prediction with *H* = 2

المنارة للاستشارات
www.manaraa.com

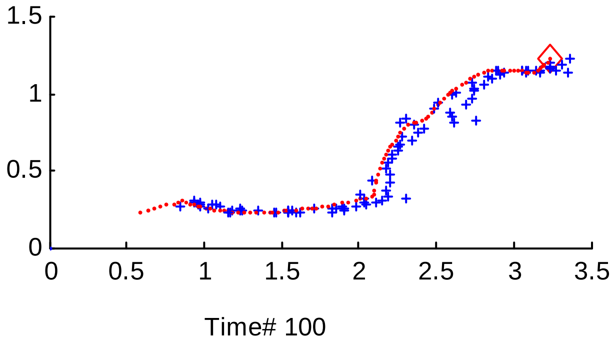


Figure 3-6 Location prediction with *H* = 4

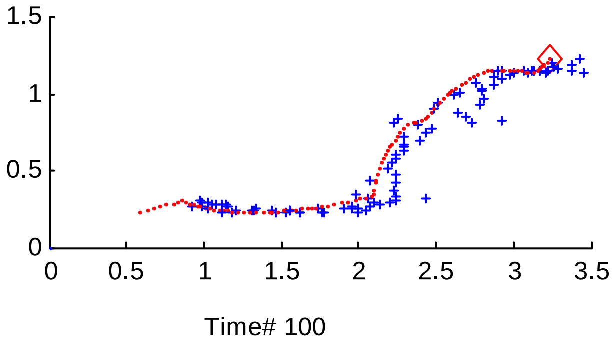


Figure 3-7 Location prediction with *H* = 6

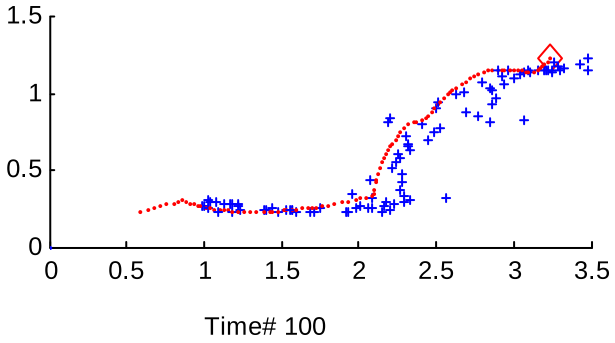


Figure 3-8 Location prediction with *H* = 8

المنارة للاستشارات
www.manaraa.com

When the mobile agent location optimization is done with future location data, two minor changes to the algorithm are necessary. One is the conversion of the velocity constraints into travel distance constraints for a time span of *H*. This is required because the system with future location prediction tries to relocate the agents from their positions at time *t* to their optimized locations at time *t*+*H*. The second modification is the interpretation of the result at *t*+*H* and its application at time *t*+1, which is where the agents are going to be deployed next.

The velocity constraints of mobile agents are converted to travel distance constraints as shown in equation (3-22), and the coordinates of the mobile agents at time *t*+1 are calculated according to equations (3-23) and (3-24).

$$0 \le v_{it} \le v'_{\max_i} = v_{\max_i} \cdot H \qquad \forall i \in AN_t \tag{3-22}$$

$$x_{i(t+1)} = x_{it} + cos(r_{it}) \cdot v_{it} / H \tag{3-23}$$

$$y_{i(t+1)} = y_{it} + sin(r_{it}) \cdot v_{it} / H \tag{3-24}$$

In equations (3-22), (3-23) and (3-24), the maximum travel distances that the mobile agents can cover are calculated. Then, the optimized movement at *H* time steps is scaled down to a single time unit by keeping the direction constant and scaling the travel distance down by *H*.

In this chapter, the mobile agent location problem is defined and modeled as a maximum flow problem variant. An objective function that reflects MANET connectivity

المنارة للاستشارات

www.manaraa.com

and overall data transmission speed, and which is sensitive to small changes in mobile agent locations is developed.

The agent velocity constraints are successfully handled with the use of polar coordinate transformations. This allows any heuristic algorithm to perform an effective search without violating velocity constraints.

The results of the kinematics based future location prediction are satisfactory. While the prediction error is expected to increase as the prediction horizon increases, the mobile agent location optimization is found to benefit from the additional information gained by a modest prediction horizon. A prediction horizon, *H*, of 4 time steps is found most beneficial. Analysis of the effect of prediction horizon on agent location optimization is given in Section 7.3.2.

www.manaraa.com

CHAPTER 4

A SPECIAL GENETIC ALGORITHM WITH NON-DETERMINISTIC BINARY DECODING FOR CONTINUOUS PROBLEMS

For a continuous domain, a genetic algorithm (GA) is usually encoded using a binary string because of simplicity and established common use. However, when a continuous domain is represented using a binary string only a finite number of discrete points are actually represented [44]. The number of represented discrete points relates to the number of binary digits so the length of binary string defines the resolution of the binary to continuous domain mapping, as well as the precision of the returned solution. For a problem with 100 variables, the required number of binary digits for a precision of six digits after the decimal point can reach thousands. For such problems, the performance of GAs is quite poor [65].

Various approaches have been developed to address the representation precision problem [59, 76]. A GA that uses a real number coding can be used but requires different crossover and mutation operators [65]. Also, the theory of genetic search is currently better established for binary string representations than for real representations [65].

المنارة للاستشارات

www.manaraa.com

4.1 Background

A few studies focus on the binary representation resolution deficiency. Schraudolph and Belew [76] introduced a method, dynamic parameter encoding (DPE), which repeatedly improves the precision by re-mapping the genes to promising smaller search regions and thus searching a finer resolved section. DPE divides its search interval into a certain number of sections and zooms into a "good" region that is identified by statistical information collected during previous generations. Kwon *et al.* [59] recently proposed a similar algorithm, the successive zooming genetic algorithm (SZGA), using continuous zooming factors. In their method, the search space is zoomed around the best point of the last 100 generations. Both methods sacrifice some portion of the search space as evolution progresses, in order to achieve better resolution. This might result in loss of the search space that contains the global optimum.

The method proposed here enables a GA to search the regions that are left out by conventional decoding functions as a result of finite resolution. Binary strings are decoded with a small Gaussian perturbation instead of being decoded on the same discrete points every time. This enables the GA to search the region between two adjacent discrete points of a conventional decoding. The non-deterministic decoding is coupled with a mapping rearrangement mechanism that continuously uses the information gathered from GA's evolution such that the best known solution is the expected decoded value of the corresponding best chromosome.

The proposed algorithm will be referred to as the non-deterministic binary decoding GA, or NDBGA, in the following sections. NDBGA is introduced and its

المنارة للاستشارات
www.manaraa.com

details are given in Section 4.2. Tests and analysis of its performance over a variety of test functions, as well as comparisons to DPE and SZGA are presented in Sections 4.2.6 and 4.2.7.

4.2 NDBGA Algorithm

NDBGA is a binary coded GA for the optimization of continuous multi-dimensional functions. Details on the modified binary decoding function and other NDBGA operators are explained in this section.

4.2.1 Motivation

When GAs are used for continuous optimization problems, the parameters are often encoded as binary strings. A typical binary encoding/decoding scheme works as follows; let x_i be the i^{th} variable of a function in a continuous search domain. A binary string chromosome of length l, used to encode x_i will represent 2^l discrete values of x_i, starting at its lower bound and ending at its upper bound. Thus, the search space for x_i is divided into 2^l-1 intervals. Conventional decoding, or mapping, from binary to continuous space is done as given in equations (2-11) and (2-12).

4.2.2 Non-deterministic Binary Decoding

The NDBGA algorithm uses a decoding function that maps a certain chromosome not to one specific point, but to a neighborhood or region around it, by adding a Gaussian offset with zero mean to the decoded value. This is identical to adding a Gaussian offset

المنارة للاستشارات
www.manaraa.com

to the lower bound of the variable that is used in the decoding function and then decoding the variable using this new lower bound. Every chromosome is assigned a specific area in the search space and these areas do not overlap. In a two-dimensional search space (Figure 4-1) each grid intersection point represents the center points of the rectangular regions that chromosomes are responsible for. This can be better visualized in Figure 4-2. In Figure 4-2, chromosomes c_a and c_b are given chances to represent points in the dotted and shaded areas, respectively. With conventional decoding, only the center points would have been represented.

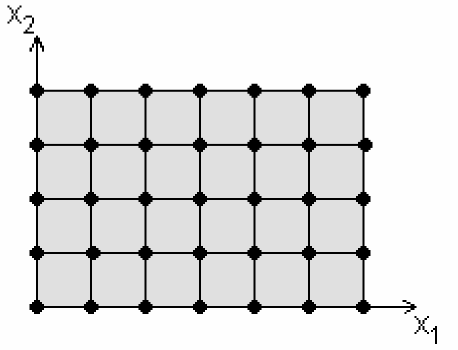


Figure 4-1 Two dimensional binary coded search space

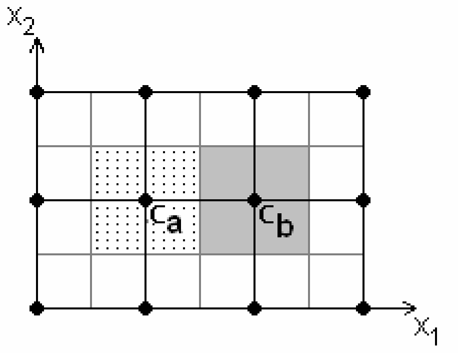


Figure 4-2 Regions represented by chromosomes c_a and c_b in NDBGA, a two dimensional case

المنارة للاستشارات
www.manaraa.com

Let c_i be the binary chromosome string for the i^{th} variable and let r_{ci} be the decoded real value of c_i. With non-deterministic decoding, c_i represents a Gaussian random number with mean r_{ci}, and a standard deviation of Kh_i, where h_i is the resolution half width of the i^{th} variable given by equation (4-2) and K is a user defined scaling factor.

$$w_i = \frac{(x_{ub_i} - x_{lb_i})}{2^{l_i} - 1}$$

(4-1)

$$h_i = \frac{w_i}{2}$$

(4-2)

where;

w_i	is the interval width of the i^{th} variable
x_{lb_i}	is the lower bound of the i^{th} variable
x_{ub_i}	is the upper bound of the i^{th} variable
l_i	is the chromosome length for the i^{th} variable

The NDBGA decoding function can be derived by implementing a dynamic lower bound value and a Gaussian offset in a conventional decoding function as in equation (4-3).

$$ndecode(c_i) = \frac{(x_{ub_i} - x_{lb_i}) \times decimal(c_i)}{2^{l_i} - 1} + lb_i + N(0, Kh_i)$$

(4-3)

www.manaraa.com

where;

x_{lb_i}	is the lower bound of the i^{th} variable
x_{ub_i}	is the upper bound of the i^{th} variable
c_i	is the chromosome for the i^{th} variable
l_i	is the chromosome length for the i^{th} variable
$decimal(c_i)$	is the decimal value of c_i
lb_i	is the lower bound of the i^{th} variable used for decoding
$N(0,Kh_i)$	is a Gaussian random number with mean 0, and standard deviation Kh_i
K	is a user defined constant, used to scale the standard deviation of the Gaussian offset
h_i	is the interval half-width of the i^{th} variable

Figure 4-3 illustrates the Gaussian mapping onto the search space. Each peak is the expected decoded value of the corresponding chromosome. The X_1 and X_2 axes are variable axes and, the *pdf* is the Gaussian probability density function value.

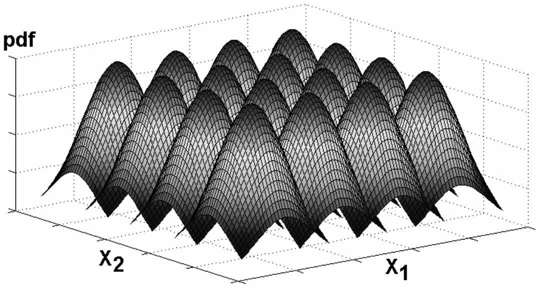


Figure 4-3 Gaussian mapping from binary representation grid to the search space

المنارة للاستشارات
www.manaraa.com

Initially, the peak points correspond to the points that would have been represented by the regular binary mapping. As NDBGA progresses, information from the individual with the best fitness in each generation is continuously used to update the decoding lower bound values so that the optimum peak point in the updated mapping correspond to the best known solution. Thus, the decoded value of the best known chromosome will always be based on the best known solution so far. This process is referred to as the *mapping rearrangement* property of NDBGA. Change of expected values of chromosomes can be visualized as a geometrical rearrangement of the binary representation grid.

The mapping rearrangement is determined by the best known individual. This is illustrated in Figure 4-4 for a two dimensional case. The black grid represents the initial binary mapping and the gray grid represents an intermediate stage, which has been rearranged to position the superior individual *S* at its corresponding grid intersection point. After the rearrangement, if the chromosome of *S* is decoded with zero offsets, point *S* will be precisely located.

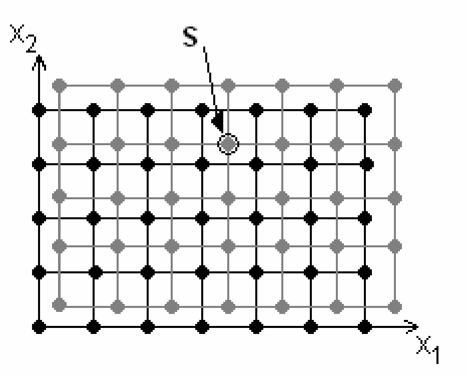


Figure 4-4 Graphical representation of binary mapping rearrangement for the 2D case

www.manaraa.com

The binary to real mapping is rearranged continuously as the population evolves. It is important to note that rearrangement does not alter regions that the chromosomes are responsible for. If we consider Figure 4-2, if the best known solution is in the dotted region, it will be represented by c_a. Similarly, if it is in the shaded region, it will be represented by c_b, etc.

The mapping rearrangement is done by updating elements of *lb* whenever a solution that is superior to the best known is created. The procedure is described in Section 4.2.3.

4.2.3 Mapping Rearrangement Mechanism

A binary mapping can be rearranged by altering the variable lower bounds vector, *lb*. When an improvement on the best-known solution is realized, the *lb* vector is updated according to equation (4-4).

$$(lb_i)_{\text{new}} = (lb_i)_{\text{old}} + N_{si} \qquad \forall i \tag{4-4}$$

where;

$(lb_i)_{new}$	is the updated lower bound of the i^{th} variable
$(lb_i)_{old}$	is the previous lower bound of the i^{th} variable
N_{si}	is the Gaussian offset that was generated for the i^{th} variable of the superior individual

المنارة للاستشارات
www.manaraa.com

4.2.4 NDBGA Algorithm Structure

While the non-deterministic decoding idea could be used with any binary GA, the one that is designed for experimentation is described here.

Notation

varsize	the number of variables in problem
l	the length of the binary string for a variable
μ	population size
q	tournament size
n_{couple}	the number of parent couples
λ	total number of offspring created ($\lambda = 2 \times n_{couple}$)
p_c	crossover probability
b_m	bit mutation probability
$\lambda_{replace}$	the number of worst individuals replaced by offspring
PopBest	best individual of the current population
BestSoFar	best individual found so far
N_i	Gaussian offset generated when decoding the i^{th} variable of an individual

Each individual in NDBGA has a total of ($varsize \cdot l$) genes as its genotype, and also stores a real value for each of its variables for the mapping rearrangement mechanism. Initially all members are generated randomly by assigning 0 or 1 to their binary genes, with equal probabilities.

www.manaraa.com

By changing the n_{couple} parameter, NDBGA can behave as a steady-state GA, a generational GA, or anything in between. Each couple contains two parents. The same parent member can appear in more than one couple but it cannot appear more than once in one couple. Parent selection is done using a tournament selection of size q. Each couple produces two children by crossover with a probability of p_c. If a couple is to undergo crossover, either a uniform crossover or a single-point crossover takes place with equal probability. Every child is subject to mutation with a bit mutation probability of b_m. Each of the bits in a child's chromosome is flipped with a probability of b_m. Figure 4-5 shows the flowchart of the NDBGA.

المنارة للاستشارات

www.manaraa.com

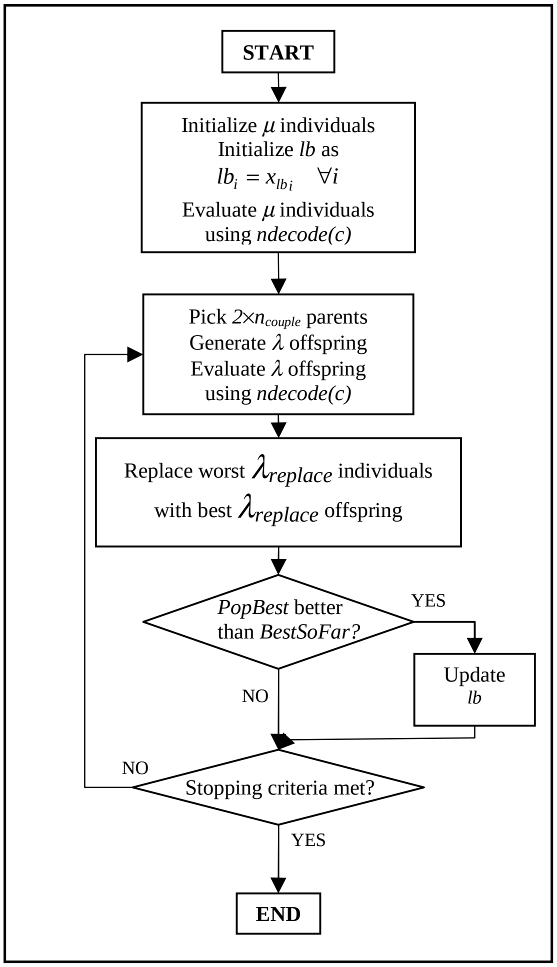

START
Initialize μ individuals
Initialize *lb* as
$lb_i = x_{lb\,i} \quad \forall i$
Evaluate μ individuals
using *ndecode(c)*
Pick $2\times n_{couple}$ parents
Generate λ offspring
Evaluate λ offspring
using *ndecode(c)*
Replace worst $\lambda_{replace}$ individuals
with best $\lambda_{replace}$ offspring
PopBest better than *BestSoFar?*
YES
Update *lb*
NO
NO
Stopping criteria met?
YES
END

Figure 4-5 NDBGA flowchart

Every time an individual superior to the current *BestSoFar* is generated**,** the *lb* vector is updated.

4.2.5 Gray Coding

Gray coding is a commonly applied method to transform a binary mapping such that adjacent points in the search space differ by one bit only in genotype. This eliminates the problem of small mutations producing solutions far away from the original point [44,

المنارة للاستشارات
www.manaraa.com

65]. NDBGA can be used with any binary encoding scheme, either Gray coded or not. Gray coding has been shown to improve GAs' performance on many types of continuous problems [21]. There are many different ways to Gray code binary strings and below is the conversion rule used in this study;

Let c_b be the binary chromosome string, c_g be the gray coded chromosome, and l be the chromosome length. Start with assigning the higher ordered bit in *Step1*:

Step1 $c_{b_l} = c_{g_l}$

Step2 $i = l-1$

Step3 $c_{b_i} = \left(c_{b_{i+1}}\right)^{c_{g_i}}$

Step4 decrease i by 1.

Step5 Goto *Step3* if $i > 0$, otherwise stop.

Where c_{b_i} and c_{g_i} are the i^{th} elements of c_b and c_g, respectively.

4.2.6 Testing NDBGA

NDBGA was tested on a set of 20 test problems, including the test sets used by Schraudolph *et al.* for DPE and test sets used by Kwon *et al.* for SZGA to see how NDBGA compares to these algorithms [59, 76]. The remainder of the test suite is compiled from well known multi-modal and multi-dimensional continuous optimization problems. The complete test suite is in Table 4-1 and Table 4-2.

www.manaraa.com

All problems are multi-dimensional and many have numerous local extrema that challenge search algorithms. All problems were tested for minimization. Functions F1 through F5 are De Jong's test functions and were used by Schraudolph *et al.* to test DPE. Function F5 was also used by Kwon *et al.* Functions F6 through F14 are the remainder of Kwon *et al.*'s functions used to test SZGA. F4 in Kwon *et al.*'s paper could not be tested due to its unbounded variables.

Function F1 is a convex three-dimensional parabola with a minimum at the origin. It is a relatively easy function to optimize. F2 is a widely studied test function which was first proposed by and named after Rosenbrock. It is a non-convex, unimodal function with a deep parabolic valley along the curve $x_2 = x_1^2$ [31]. F3 is a 30 dimensional, discontinuous step function. F4 is a convex and unimodal 30 dimensional quadric function with Gaussian noise. It is useful to test the performance of an optimization algorithm under the presence of noise. F5 is known as Shekel's foxholes [31]. It is an interesting two-dimensional multi-modal function with 25 local minima. F6 was proposed by Bohachevsky *et al.* and is a two-dimensional function with numerous local minima and a global optimum at the origin [39]. When (x_1,x_2) is far from origin, the quadratic terms of F6 dominate the cosine terms, thus giving an overall quadratic shape to the function [39]. F7 is the second function in Kwon *et al.*'s test suite. It is a multi-modal function with 16 local minima. F8 is commonly known as the Branin-RCOS function and has a global minimum at three different locations [34]. F9 is known as the six-hump camelback function and has three conjugate pairs of local optima, one of which is the global minimum [87]. F10, known as Goldstein-Price function, has four local

المنارة للاستشارات
www.manaraa.com

minima and a global minimum [87]. F11, also known as the Shubert function, has 760 local minima, 18 of which are global minima [65]. F12, also known as the Colville function, is a 4 dimensional function with a global minimum at (1,1,1,1), a stationary point at (1,-1,1,-1), and further local minima. A very narrow valley runs from the stationary point to the minimum [78]. F13 and F14 are the last two test functions in Kwon *et al.*'s test suite. They are both 20 dimensional functions with global optimal values of 0. The remainder of our test suite consists of a variety of functions ranging from 2 to 30 dimensions.

F15 is a 2 dimensional, highly multi-modal function proposed by Schaffer [75]. F16 and F17 are 5 and 20 dimensional variants of the generalized Rastrigin function, respectively. The function was first proposed by Rastrigin as a 2-dimensional problem, and generalized by Rudolph as a test function for distributed parallel evolutionary strategies [77, 87]. F18 is a 30 dimensional version of the sphere function F1. F19, known as Schwefel's problem, is a continuous and unimodal problem [77]. Finally, F20 is another function by Bohachevsky with 10 dimensions. It has numerous local optima and a global optimum of 0. Optimizing F20 requires a simultaneous minimization of 10 multi-modal functions [39].

www.manaraa.com

Table 4-1 Test Functions F1-F10

Fn.	Equation	Dim.	Variable range	*Theoretical optimum (f*)*
F1	$\sum_{i=1}^{3} {x_i}^2$	3	[-5.12,5.12]	f^* = 0.0 at x_i^* = 0
F2	$100({x_1}^2 - x_2)^2 + (1 - x_1)^2$	2	[-2.048,2.048]	f^* = 0.0 at x_i^* = 1
F3	$\sum_{i=1}^{30} \lfloor x_i \rfloor$	30	[-5.12,5.12]	F^* = -30.0 at $x_i^* \in$ [5.12, -5)
F4	$\sum_{i=1}^{30} i x_i^4 + Gaussian(0,1)$	30	[-1.28,1.28]	f^* = 0.0 (underlying function) at x_i^* = 0
F5	$\left(0.002 + \sum_{j=1}^{25} \left(j + \sum_{i=1}^{2} (x_i - a_{ij})^6 \right)^{-1} \right)^{-1}$	2	[-65.536,65.536]	f^* = 0.9980038 at x_i^* = -32
	$a = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & -16 & -16 & -16 & 0 & 0 & 0 & 0 & 0 & 16 & 16 & 16 & 16 & 16 & 32 & 32 & 32 & 32 & 32 \end{bmatrix}$			
F6	${x_1}^2 + 2{x_2}^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	2	[-1.28,1.28]	f^* = 0.0 at x_i^* = 0
F7	$[\cos(2\pi x_1) + \cos(2.5\pi x_1) - 2.1] \times [2.1 - \cos(3\pi x_2) - \cos(3.5\pi x_2)]$	2	[-1,1]	f^* = -16.0917200 at (0.4388, -0.3058)
F8	$(x_2 - \frac{5.1}{4\pi^2}{x_1}^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	2	$x_1 \in$ [-5,10] $x_2 \in$ [0,15]	f^* = 0.3978873 at (-3.142, 12.275) (3.142, 2.275) (9.425, 2.245)
F9	$\left(4 - 2.1x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1 x_2 + (4x_2^2 - 4)x_2^2$	2	[-5,5]	f^* = -1.0316285 at (0.08983, -0.7126) (-0.08983, 0.7126)
F10	$\{1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)\} \times$ $\{30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)\}$	2	[-2,2]	f^* = 3.0 at (0, -1)

المنارة للاستشارات
www.manaraa.com

Table 4-2 Test Functions F11-F20

Fn.	Equation	Dim.	Variable range	*Theoretical optimum (f*)*
F11	$\left\{\sum_{i=1}^{5} i\cos((i+1)x_1+i)\right\}\times\left\{\sum_{j=1}^{5} j\cos((j+1)x_2+j)\right\}$	2	[-10,10]	*f** = - 186.7309088 at (-1.425, -0.800)
F12	$100\left(x_2-x_1^2\right)^2+\left(1-x_1\right)^2+90\left(x_4-x_3^2\right)^2+\left(1-x_3\right)^2+$ $10.1\left(\left(x_2-1\right)^2+\left(x_4-1\right)^2\right)+19.8\left(x_2-1\right)\left(x_4-1\right)$	4	[-10,10]	*f** = 0.0 at x_i* = 1
F13	$\sum_{i=1}^{19}\left[\left(x_i^2\right)^{\left(x_{i+1}^2+1\right)}+\left(x_{i+1}^2\right)^{\left(x_i^2+1\right)}\right]$	20	[-1,4]	*f** = 0.0 at x_i* = 0
F14	$\frac{\pi}{20}\left\{10\sin^2(\pi x_1)+\sum_{1}^{19}\left[\left(x_i-1\right)^2\left(1+10\sin^2(\pi x_{i+1})\right)\right]+\left(x_{20}-1\right)^2\right\}$	20	[-10,10]	*f** = 0.0 at x_i* = 1
F15	$0.5+\frac{\left[\sin(\sqrt{x_1{}^2+x_2{}^2})\right]^2-0.5}{\left[1.0+0.001\times(x_1{}^2+x_2{}^2)\right]^2}$	2	[-100,100]	*f** = 0.0 at (0,0)
F16	$\sum_{i=1}^{5}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	5	[-5.12,5.12]	*f** = 0.0 at x_i* = 0
F17	$\sum_{i=1}^{20}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	20	[-5.12,5.12]	*f** = 0.0 at x_i* = 0
F18	$\sum_{i=1}^{30} x_i^2$	30	[-5.12,5.12]	*f** = 0.0 at x_i* = 0
F19	$\sum_{i=1}^{20}\left[\sum_{j=1}^{i} x_j\right]^2$	20	[-65.536,65.536]	*f** = 0.0 at x_i* = 0
F20	$\frac{1}{10}\left\{\begin{array}{l}\sum_{i=1}^{9}\left(x_i{}^2+2x_{i+1}^2-0.3\cos(3\pi x_i)\cos(4\pi x_{i+1})+0.3\right)+\\ x_{10}{}^2+2x_1^2-0.3\cos(3\pi x_{10})\cos(4\pi x_1)+0.3\end{array}\right\}$	10	[-50,50]	*f** = 0.0 at x_i* = 0

General performance measures that are considered can be listed as; the average number of function evaluations before the population best reaches the global optimal

www.manaraa.com

within a tolerance limit, the standard deviation of number of function evaluations, the objective value of the population best, the average objective value of the population best, and the standard deviation of the objective value of the population best. All of the statistics are calculated over 20 repetitions with different random number seeds for each function.

For fair comparisons with DPE and SZGA, NDBGA was limited to the corresponding number of function evaluations, where applicable, for test functions F1 through F14. For all other results provided, the function evaluation limit is set to 500,000 for each run.

It is well known that GAs generally require larger population sizes as the number of variables increase, for better performance. After preliminary experiments, the population size of NDBGA was set to 30 for functions with less than 10 variables, and to 100 for others. Using a generational strategy provides better diversification and performed well on problems with less than 10 variables but it greatly decreased the convergence ability for larger problems. An elitist generational strategy was used for problems with less than 10 variables, and a steady-state strategy was employed for the larger problems. A fixed bit string length of 8 was used for all test functions, unless indicated otherwise. Other fixed parameters related to the evolution dynamics are given in Table 4-3.

Table 4-3 NDBGA Population Parameters

Population size (μ)	Number of couples (n_{couple})	Tournament size (q)	Offspring size (λ)	$\lambda_{replace}$	p_c	b_m	K
30	15	2	30	29	0.85	0.05	0.050
100	1	2	2	2	0.85	0.01	0.001

المنارة للاستشارات
www.manaraa.com

4.2.7 NDBGA Performance

NDBGA performed satisfactorily throughout the entire test suite Table 4-4 and Table 4-5 summarize the performance comparisons of NDBGA with DPE and SZGA on test functions F1 through F5 and F5 through F14, respectively. Both tables are structured the same way.

Table 4-4 presents the best and the average results of NDBGA along with the average results of DPE. NDBGA performs at least as good as DPE except for F4, the noisy problem. It is not clear from Schraudolph *et al.'s* paper if the reported values of F4 were noise-free or not. NDBGA's performance measures for the noisy F4 function is calculated by using the noise-free values of the function since it is the underlying function that is optimized. A completely noise-free version of F4 was also tested and presented along with the original version.

Table 4-5 presents the best and the average NDBGA results in comparison with best results reported for SZGA [59]. Again, NDBGA performs at least as good as SZGA with the same number of function evaluations allowed.

www.manaraa.com

Table 4-4 Results for F1-F5 Compared With DPE

Fn.	Theoretical minimum	Number of function evaluations	DPE results[a]	NDBGA results[a]	
			Population average	Population best	Average population best
F1	0.0	12,000	1×10^{-18}	$<1\times10^{-30}$	$<1\times10^{-30}$
F2	0.0	6,000	$3\times10^{-2}<\text{DPE}<1\times10^{-1}$	2.1×10^{-8}	1.66×10^{-2}
F3	-30.0	4,000	1×10^{-2}	-30.0	-30.00
F4[b]	0.0	10,000	3×10^{-1}	5.1×10^{-1}	5.16×10^{-1}
F4[c]	0.0	10,000	Not available	2.42×10^{-1}	2.77×10^{-1}
F5	0.9980038	18,000	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}

[a] NDBGA and DPE results are with 3 bits per variable for F1 through F4 and 6 bits per variable for F5.

[b] Values for the underlying function are presented.

[c] Values for F4 optimized with no noise are presented.

Table 4-5 Results for F5-F14 Compared With SZGA

Fn.	Theoretical minimum	Number of function evaluations	SZGA results[a]	NDBGA results [a,b]	
			Population best	Population best	Average population best
F5[c]	0.9980038	8,000	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}
F6	0.0	4,000	2.98×10^{-8}	5.55×10^{-17}	5.55×10^{-17}
F7	-16.0917200	4,000	-16.09172	-16.0917200	-16.0037882
F8	0.3978873	4,000	0.39789	0.3978874	0.3979254
F9	-1.0316285	3,000	-1.03163	-1.0316284	-1.0316249
F10	3.0	4,000	3.0	3.0000000	3.0000000
F11	-186.7309088	3,000	-186.73091	-186.7309088	-186.7282441
F12	0.0	228,000	1.3074×10^{-6}	1.04×10^{-6}	0.1629989
F13	0.0	500,000	2.5422×10^{-8}	5.08×10^{-10}	7.61×10^{-10}
F14	0.0	668,000	2.3033×10^{-4}	7.85×10^{-10}	1.41×10^{-9}

[a] NDBGA and SZGA results are with 8 bits and 12 bits per variable, respectively.

[b] Presented results are calculated over 20 runs with different random number seeds.

[c] NDBGA was limited to 8,000 function evaluations for F5 when compared with SZGA, which is different than 18,000 function evaluation limit used for comparison with DPE.

المنارة للاستشارات
www.manaraa.com

For the comparison of NDBGA with DPE given in Table 4-4, NDBGA was set to 3 bits per variable for the functions F1 through F4 and 6 bits per variable for F5. DPE was tested with a fixed binary string size of 3 per variable for functions F1 through F5 [76]. This resolution of DPE was not enough for F5, but with a resolution of 6 bits per variable DPE was able to converge to the global optimal in 18,000 function evaluations. For the F5 function, NDBGA converged to the global optimal with 6 bits per variable in 1,400 function evaluations on average. SZGA was tested with 12 bits per variable [59]. NDBGA's binary string size was kept at 8 bits per variable for comparisons with SZGA as given in Table 4-5. The high number of function evaluations of DPE might be a result of the "folding" process where the individuals falling out of the zoomed in region are recalculated using new bounds, and this requires objective function evaluations equal to the number of individuals recalculated. Similarly, in SZGA, the population is reset every time the intervals are changed. NDBGA does not re-initialize individuals like DPE and SZGA and it can therefore conserve objective function evaluations while converging to global optimal.

The general performance measures of the NDBGA on the entire test suite, also compared with a conventionally decoded GA (CDGA) are tabulated in Appendix Ap- 1, page 184. CDGA is identical to NDBGA algorithmically and parameter-wise except for the decoding method. It also uses the same Gray coding scheme but it has the conventional binary to real decoding function.

The tolerance limit, *t*, is set to 10^{-7} for all functions except the noisy F4, which had a tolerance limit of 0.25. When the best individual in the population reaches an objective

www.manaraa.com

function value that is closer than *t* to the theoretical minimum, it is considered a successful convergence. All problems successfully converged to the theoretical optimum 100% of the time, except the noisy F4, which had a success rate of 55%.

The encoding mechanism presented in this chapter enables a binary coded genetic algorithm to perform an efficient and effective search in the continuous domain. It is very easy to convert a standard binary coded GA into NDBGA. The standard binary coded GA in the continuous domain usually suffers from long chromosome strings, which can have adverse effects in its performance due to some second order effects. i.e. effects due to the algorithm mechanism, not due to problem size or complexity. Moreover, the number of generations it takes to converge to a superior solution increases with increasing chromosome length.

The NDBGA mechanism helps the GA in two ways. One is the shorter chromosome length. The other is the fine search undertaken by the mapping rearrangement mechanism without increasing the number of function evaluations per generation. This helps keep down the overall number of generations and thus the total number of objective function evaluations, which is critical for the computationally intense functions in this study. As shown in Section 7, the NDBGA performs very satisfactorily over different types and sizes of test instances.

المنارة للاستشارات
www.manaraa.com

CHAPTER 5

AN APPROXIMATE LINEAR MODEL WITH PIECEWISE LINEAR APPROXIMATIONS FOR DISTANCE AND DATA FLOW RATE

The mobile agent location optimization model proposed in this study involves a complex and non-linear objective function as well as non-linear constraints. Although this is a typical scenario where researchers and practitioners benefit from using heuristic optimization methods, a similar model is developed as a mixed-integer programming (MIP) model by approximating the non-linear parts of the original model by piecewise linear curves. This model is then used to compare the performances of the heuristics up to medium sized (10-15 nodes) problems.

5.1 Mathematical Model

The approximate mixed integer model is developed by modification of a standard maximum flow model to incorporate the all-pair maximum flow calculation. This allows relocation of agent nodes considering the velocity constraints and enables varying link capacities with varying link distances.

The basic idea for the all-pair maximum flow calculation is to let virtual commodities, equal to the number of node pairs, flow through the network without

المنارة للاستشارات
www.manaraa.com

sharing link capacities between commodity types. All distance calculations are approximated to the Euclidean distances by mapping the orthogonal components of the link and agent travel distances to their second power and summing them to get the square of the said distances. The velocity and data rate constraints are then handled using the distances in their squared forms. The model given below is followed by the description of the notation used.

$$\text{Maximize } F + \frac{\sum_{S,T\in UN:T>S} F_{ST}}{(n)\cdot(n-1)/2}$$

(5-1)

Subject to

$$F \le F_{ST} \qquad \forall S,T \in UN\text{: } T>S$$

(5-2)

$$C_{ijST} \le u_{ij} \qquad \forall (i,j) \in E\ ,\ \forall (S,T) \in UN\text{: } T>S$$

(5-3)

$$\sum_{\{j:(i,j)\in E_t\}} C_{ijST} - \sum_{\{j:(j,i)\in E_t\}} C_{jiST} = \begin{cases} F_{\mathrm{ST}} & \text{for } i = S, \\ 0 & \forall\, i \in N \text{ - } \{S \text{ and } T\}, \\ -F_{ST} & \text{for } i = T \end{cases}$$

(5-4)

$$d_{ij}^{x} \ge |x_i - x_j| \qquad \forall \{i \in N, j \in AN, (i,j) \in E\}$$

(5-5)

$$d_{ij}^{y} \ge |y_i - y_j| \qquad \forall \{i \in N, j \in AN, (i,j) \in E\}$$

(5-6)

المنارة للاستشارات

www.manaraa.com

$$D^{e}_{ij} \geq a^{dx}_{p_{ex}} + b^{dx}_{p_{ex}} \cdot d^{x}_{ij} + a^{dy}_{p_{ey}} + b^{dy}_{p_{ey}} \cdot d^{y}_{ij}$$

$$\forall \{ i \in N, j \in AN, (i,j) \in E \},\ L^{dx}_{p_{dx}-1} \leq d^{x}_{ij} < L^{xx}_{p_{dx}} \ ,$$

$$L^{dy}_{p_{dy}-1} \leq d^{y}_{ij} < L^{dy}_{p_{dy}} \ ,\ L^{dx}_{0} = L^{dy}_{0} = 0 \ ,$$

and $0 < p_{dx}, p_{dy} < P_d$

(5-7)

$$u_{ij} \leq a^{u}_{p_u} + b^{u}_{p_u} \cdot D^{e}_{ij} \qquad \forall \{ i \in N, j \in AN, (i,j) \in E \},$$

$$L^{u}_{p_u-1} \leq D_{ij} < L^{u}_{p_u} < \left(R_j\right)^2 \ ,\ L^{u}_{0} = 0 \ ,$$

and $0 < p_u < P_u$

(5-8)

$$u_{ij} \leq u_{ji} \qquad \forall \{ i \in N, j \in AN, (i,j) \in E \}$$

(5-9)

$$v^{x}_{j} \geq \left| x_j - xl_j \right| \qquad \forall \{ j \in AN \}$$

(5-10)

$$v^{y}_{j} \geq \left| y_j - yl_j \right| \qquad \forall \{ j \in AN \}$$

(5-11)

$$a^{vx}_{p_{vx}} + b^{vx}_{p_{vx}} \cdot v^{x}_{j} + a^{vy}_{p_{vy}} + b^{vy}_{p_{vy}} \cdot v^{y}_{j} \leq \left(v_{\max j}\right)^2$$

$$\forall \{ j \in AN \},\ L^{vx}_{p_{vx}-1} \leq v^{x}_{j} < L^{vx}_{p_{vx}} \ ,$$

$$L^{vy}_{p_{vy}-1} \leq v^{y}_{j} < L^{vy}_{p_{vy}} \ ,\ L^{vx}_{0} = L^{vy}_{0} = 0 \ ,$$

and $0 < p_{vx}, p_{vy} < P_v$

(5-12)

www.manaraa.com

5.2 Notation

x_i	is the x-coordinate of node i
y_i	is the y-coordinate of node i
F	is the flow value between the user pair that has the lowest maximum flow value among all pairs
F_{ST}	is the flow value of the virtual commodity ST from source node S to target node T
N	is the set of nodes
n	is the number of nodes
C_{ijST}	is the flow value of virtual commodity ST through the link ij
d_{ij}^{x}	is the absolute difference between the x coordinates of nodes i and j
d_{ij}^{y}	is the absolute difference between the y coordinates of nodes i and j
$a_{p_{ex}}^{dx}$	is the constant factor of the p_{ex}th segment of the piecewise approximation curve for $(d_{ij}^{x})^2$
$b_{p_{ex}}^{dx}$	is the rate factor of the p_{ex}th segment of the piecewise approximation curve for $(d_{ij}^{x})^2$
$a_{p_{ey}}^{dy}$	is the constant factor of the p_{ey}th segment of the piecewise approximation curve for $(d_{ij}^{y})^2$
$b_{p_{ey}}^{dy}$	is the rate factor of the p_{ey}th segment of piecewise approximation curve for $(d_{ij}^{y})^2$

المنارة للاستشارات

www.manaraa.com

D_{ij}	is the approximated value of the link (i,j)'s distance squared
$L_{p_{dx}}^{dx}$	is the boundary of the $p_{dx}{}^{th}$ segment of the piecewise approximation curve for $(d_{ij}^{x})^2$, in terms of (d_{ij}^{x})
$L_{p_{dy}}^{dy}$	is the boundary of the $p_{dy}{}^{th}$ segment of the piecewise approximation curve for $(d_{ij}^{y})^2$, in terms of (d_{ij}^{y})
P_d	is the number of line segments used for approximating the link distances between the agent nodes and other nodes that are within their transmission range, or may become by agent relocation within velocity constraints
$a_{p_u}^{u}$	is the constant factor of the $p_u{}^{\text{th}}$ segment of the piecewise approximation curve for the link data rate
$b_{p_u}^{u}$	is the rate factor of the $p_u{}^{\text{th}}$ segment of the piecewise approximation curve for the wireless link data rate
$L_{p_u}^{u}$	is the boundary of the $p_u{}^{th}$ segment of the piecewise approximation curve for the wireless link data rate u, in terms of D_{ij}
P_u	is the number of line segments used to approximate the wireless link data rate curve
v_j^x	is the travel amount of mobile agent j in the x-direction
v_j^y	is the travel amount of agent j in the y-direction
$a_{p_{vx}}^{vx}$	is the constant factor of the $p_{vx}{}^{\text{th}}$ segment of the piecewise approximation curve for $(v_j^x)^2$

المنارة للاستشارات

www.manaraa.com

$b_{p_{vx}}^{vx}$	is the rate factor of the $p_{vx}{}^{\text{th}}$ segment of the piecewise approximation curve for $(v_j^x)^2$
$a_{p_{vy}}^{vy}$	is the constant factor of the $p_{vy}{}^{\text{th}}$ segment of the piecewise approximation curve for $(v_j^y)^2$
$b_{p_{vy}}^{vy}$	is the rate factor of the $p_{ey}{}^{\text{th}}$ segment of piecewise approximation curve for $(v_j^y)^2$
v_{maxj}	is the maximum velocity for agent j
$L_{p_{vx}}^{vx}$	is the boundary of the $p_{vx}{}^{th}$ segment of the piecewise approximation curve for $(v_j^x)^2$, in terms of (v_j^x)
$L_{p_{vy}}^{vy}$	is the boundary of the $p_{vy}{}^{th}$ segment of the piecewise approximation curve for $(v_j^y)^2$, in terms of (v_j^y)
P_v	is the number of line segments used to approximate the magnitude of the agent velocity squared

Decision variables

xl_j	is the x-coordinate of the relocated agent node j
yl_j	is the y-coordinate of the relocated agent node j

As seen in Figure 5-1, the calculation of the linear coefficients *a* and *b* for a piecewise linearization procedure is fairly straightforward, given that the number of line segments *P*, and the variable limits L^x_p, p $\in$ [1,2,..,*P*-1] are known. This figure is

www.manaraa.com

presented for demonstration purposes only for $f(x) = x^2$, $x \in [0, 2]$ and $P = 3$. The dashed line segments represent the linear curves that are used to constrain the decision variable that is approximated to x^2 given x. The slope for the $(P+1)^{\text{th}}$ segment is assumed to be infinity. The AMPL modeling language which is used to model the linear approximate problem has a built-in function to automatically approximate non-linear functions. In this study this functionality is used, information on which can be found in the AMPL user's manual [74].

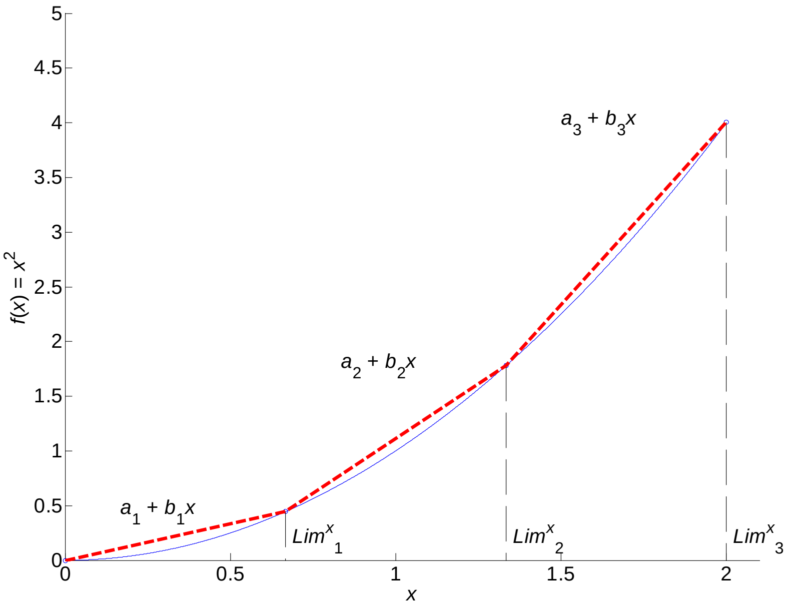


Figure 5-1 The piecewise linear approximation of $f(x) = x^2$. An example is plotted for $x \in [0, 2]$, $P = 3$

المنارة للاستشارات
www.manaraa.com

The AMPL model file, run file and an example data file can be found in Appendices Ap- 4, Ap- 5, Ap- 6 on pages 189 through 195. The descriptions of the constraints of the above model are as follows:

The constraint given in equation (5-2) bounds the all-pair minimum flow decision variable by all flows between any node pair.

The constraints given in equation (5-3) limit any virtual commodity's flow on a particular link by the link capacity. The flow of different virtual commodities are exclusive and do not take up each others flow resources.

The constraints given in (5-4) maintain the flow equalities separately for every virtual commodity flow.

The constraints given in (5-5) and (5-6) set the link distances in the x and y directions, respectively, to the relevant decision variables by imposing a lower bound. A lower bound is necessary because for larger link capacities, the link distance variables would tend to become smaller.

The constraints given in (5-7) bound the decision variable used to approximate the square of the link distances by the corresponding piecewise linear curve.

The constraints given in (5-8) bound the decision variable used to approximate the link flow capacities by the piecewise linear D_{ij} versus u_{ij} curve.

The constraints given in (5-9) maintain the equality in the link flow capacities in the inverse directions.

المنارة للاستشارات

www.manaraa.com

The constraints given in (5-10) and (5-11) set the mobile agent travel distances in the *x* and *y* directions, respectively, to the relevant decision variables by imposing a lower bound.

The constraints given in (5-12) are used to approximate the square of the distance traveled to the corresponding piecewise linear curve.

The mixed integer approximate linear model for mobile agent location optimization is a complex model, with the solver being able to return solutions for problems up to medium size in reasonable time. It is used for verification of the heuristics and comparisons on small scale problems. The performance of the mixed integer model, even for small test problems, is far below the two heuristics developed in this study. The results of the MIP model and comparisons with the NDBGA and PSO heuristics can be found in Section 7.4.

One reason for the poor performance of the mixed integer model is the lack of flexibility, especially in defining the objective function. The heuristics are designed to favor solutions with total connectivity in any situation, or utilize the non-zero minimum flow if this is not possible. The first term in the objective function given in (5-1) is zero when the network is not fully connected. This is sometimes not preventable for sparse networks. The heuristics, however still consider the non zero minimum flow within the network and continue with the optimization accordingly.

Additionally, the mixed integer program solutions had to be limited with a 10% optimality gap in order to return solutions in practical time (although still slower than the

المنارة للاستشارات
www.manaraa.com

two heuristics). The model, however, proves how complex and hard to solve a problem this is and provides a reasonable basis for algorithm comparison.

المنارة للاستشارات
www.manaraa.com

CHAPTER 6

THE MOBILE AGENT LOCATION OPTIMIZATION SYSTEM AND THE SIMULATION ENVIRONMENT

In the field where the MANET is active, the locations of the mobile agents determine the links between them and any other node on the network. This affects the link capacities and therefore the maximum possible data flow rates between the user nodes, which is an important consideration when maximizing the network performance.

An optimization engine needs two decision variables per mobile agent- its direction of motion and its magnitude of motion- and an objective function as defined in equation (3-7) that can be calculated when the node coordinates are known. Both the NDBGA and PSO are population based heuristic optimization tools. Their general working principle for mobile agent location optimization is given in Figure 6-1 as a Pseudo code.

In Figure 6-1, the variables marked with * show the best values returned by the optimizer. The implementation details of the NDBGA and the PSO algorithms are described in Section 6.2 and 6.3, respectively.

المنارة للاستشارات
www.manaraa.com

```
Start{
    Initialize $t = t_o$
    Do{
      Read $XY_t$
      Optimize with NDBGA or PSO $r_{it}$ and $v_{it}$ using $XY_t$ , find $r_{it}^*$ and $v_{it}^*$
      Update $(x_{i(t+1)},y_{i(t+1)})$ using $r_{it}^*$ and $v_{it}^*$   $\forall i \in AN_t$
      Deploy mobile agents to $(x_{i(t+1)},y_{i(t+1)})$ $\forall i \in AN_t$
      Set $t = t + 1$
    }While (User nodes are active)
}End
```

Figure 6-1 The pseudo code for the mobile agent location optimizer

A MANET can perform in a stationary situation as well as in a dynamic one. Therefore, a second type of problem exists which involves locating the mobile agents to optimize the connectivity and performance of static users.

6.1.1 Dynamic Scenarios

The problems in this group are multiple time step problems with varying user locations in each time step, as mainly considered in this study. Mobile agents are bound with a velocity constraints which limit their movement each time step.

المنارة للاستشارات
www.manaraa.com

6.1.2 Static Scenarios

There are times that a MANET might need to be set up in a stationary fashion. Such applications can be seen in sensor networks, or at temporary establishments such as military or disaster emergency camps, short-term housings, etc.

Modifying the mobile agent optimizer algorithm for the static case is fairly simple. These can be regarded as single time step problems where agents are not bound with velocity limits. The objective is to locate the agents such that the connectivity and network performance among the stationary users are maximized.

6.2 The NDBGA for Mobile Agent Location Optimization

The general structure and parametric settings of the NDBGA algorithm for the mobile agent location problems are as follows:

Due to the inherent characteristics of the static and the dynamic problems, two different NDBGA generation strategies are followed. For the static type, a generational strategy is adopted. A generational strategy means that the child population size is equal to the population size, and the children replaces all parent population members except the best two of the parent population, following an elitist strategy.

For the dynamic type of problems, a steady state strategy is used, in which 20% of the population members are selected, following a tournament selection strategy, as couples and the child population replaces the worst 20% of the parent population. The steady state strategy is implemented for the dynamic problems because the search space is relatively small when compared to the static problems and a continuous incremental

www.manaraa.com

optimization is performed. The detailed pseudo code of the NDBGA for mobile agent location optimization is as follows:

```
Start{
     Initialize $t = t_o$
     Read $XY_t$
     $BestSoFar = -M$
     Set $r_{it}$ and $v_{it}$ for each agent $i$ as decision variables, encoded by $l$ binary digits each.
     Do{
       Initialize population for ( each population member ){
               for ( each $r_{it}$ variable ) $lb_{var} = r_{lb} = 0$
               for ( each $v_{it}$ variable ) $lb_{var} = v_{lb} = 0$
               if ( $t \neq t_o$ AND $mbr = 1$) then {
                       Transfer population best from time ($t$-1)
               }else{
                       Initialize population randomly by assigning each gene a binary digit randomly with equal chances for 0 and 1.
               }
       }
       Evaluate fitness for ( each population member ){
               Decode for ( each $r_{it}$ variable $var$){
                       $lb_{\text{var}}^{temp} = lb_{var} + N(0,Kh_{var})$
                       $r_{it} = \frac{(2\pi - r_{lb})\cdot decimal(c_{\text{var}})}{2^l - 1} + lb_{\text{var}}^{temp}$
               }
               Decode for ( each $v_{it}$ variable $var$){
                       $lb_{\text{var}}^{temp} = lb_{var} + N(0,Kh_{var})$
```

المنارة للاستشارات
www.manaraa.com

```
                    $v_{it} = \frac{\left(v_{\max i} - v_{lb}\right) \cdot decimal\left(c_{\text{var}}\right)}{2^{l} - 1} + lb_{\text{var}}^{temp}$
            }
            Calculate the $x_{i(t+1)}$ and $y_{i(t+1)}$ coordinates suggested by the solution at $(t+1)$
            for every agent $i$ using decoded $r_{it}$ and $v_{it}$ values as given in equations
            ( 3-10 ) and ( 3-11 ).
            Form MANET topology, $G_{t+1}$, and calculate link capacities.
            Calculate objective:
            Set fitness $F$ = MANET performance metric (Equation ( 3-7 ) )
            if ( Member fitness is better than BestSoFar ) then {
                    Update BestSoFar as the member and its solution for
                    Update $lb_{var}$ for each variable with $lb_{\text{var}}^{temp}$ of the BestSoFar solution
            }
    }
Loop{
        for ( $n_{couple}$ couples){
            Repeat for 2 parents: $P_1$, $P_2${
                    Randomly pick $q$ individuals from the population
                    Assign the one with best fitness as a parent
            }
        Crossover parents to create two offspring with probability $p_c${
                    DO with equal probabilities{
                            {Uniform crossover: Offspring get each chromosome from
                            either of the parents, $P_1$ and $P_2$, with equal probabilities.}
                    OR
                            {Singlepoint crossover:Offspring get a random length
                            sequence of genes from one parent, and the corresponding
                            rest from the other. The reverse is applied to create the
                            second offspring.}
                    }
```

المنارة للاستشارات

www.manaraa.com

```
                if (No crossover) then {
                        Transfer $P_1$ and $P_2$ to offspring population without crossover.
                }
                Mutate (bit flip) each offspring gene with probability $b_m$
            }
            Evaluate fitness for (each offspring member)
            Replace worst $\lambda_{replace}$ population members with best $\lambda_{replace}$ offspring
    }While (Loop) (stopping criteria is not met)

    Update $(x_{i(t+1)},y_{i(t+1)})$ $\forall i \in AN_t$ using BestSoFar solution
    Deploy mobile agents to $(x_{i(t+1)},y_{i(t+1)})$ $\forall i \in AN_t$
    Set $t = t + 1$
    }While (Do) (User nodes are active)
}End
```

The NDBGA takes two decision variables per agent, the direction and the magnitude of its velocity, binary encoded using 12 bits per variable. Every individual in the population represents one possible movement scenario for the mobile agent nodes. The fitness evaluation involves generating the network that corresponds to the movement scenario, find the link capacities and calculate the objective function, as described in the "Evaluate fitness" routine in the above pseudo code. This is followed by parent selection using tournament selection, crossover and mutation to create offspring, evaluation of the offspring fitness and replacement of the parent population accordingly. This loop continues until the stopping criteria is met, which is described in Section 6.4.

www.manaraa.com

The genetic algorithm specific parameters such as population size (μ), tournament size (q), bit mutation rate (b_m) and crossover rate (p_c) are determined by factorial experimentation as described in Section 6.2.1.

6.2.1 Tuning the NDBGA parameters

Like every other heuristic, a GA's performance depends to some degree eon its parameter settings. As Wolpert and Macready [90] state in their study, known as the "no free lunch" theorem, different search algorithms over the global problem space are indistinguishable, but might return superior solutions on some group of problems. In other words, it is shown that no single algorithm is perfect for comprehensive problem space.

Algorithms tailored for a specific group of problems are expected to perform better than others on that problem type. The behavior of an algorithm on any problem changes when the values of its certain parameters change. In NDBGA, population size, tournament size, mutation rate and crossover rate are the general GA parameters that are commonly tested among different problem types, and the value of the non-deterministic decoding error, K, is an additional parameter that has a potential effect on algorithm performance.

In order to investigate the significance of the NDBGA parameters on the mobile agent location optimization algorithm, a factorial experimentation is carried out. Parameters such as population size, tournament size, bit mutation rate, crossover rate and the decoding error K are tested with the following factor levels:

www.manaraa.com

Population size (μ):	four levels at	30	60	90	120
Tournament size (q):	four levels at	2	3	4	5
Bit mutation rate (b_m):	four levels at	0.02	0.03	0.04	0.05
Crossover rate (p_c):	two levels at	0.70	0.90		
K	two levels at	0.05	0.20		

These levels are selected per preliminary experimentation results and common practice among GA researchers [31, 44, 65]. The tests for the above factor levels are done on 5 stationary test problems with 7 users and 5 agents, with 5 replications per problem using different random seeds for a total of 25 runs per factor level combination (FLC), resulting in a total of 6400 runs. All runs are conducted with the same stopping criteria: If the best solution is not improved in 1000 consecutive objective function evaluations, the algorithm stops and returns the best solution found.

The three performance measures that are considered, in order of importance are:

1) The average percent of user nodes that one user node can communicate with, (P_1) (%).
2) All-pair minimum bandwidth, (P_2) (Mbps).
3) Total bandwidth, (P_3) (Mbps).

www.manaraa.com

Calculation of the above performance measures are shown in detail in Section 7.3, in equations (7-6), (7-7) and (7-8), respectively.

The results of the experiments are analyzed with Minitab software, and the analysis of variance (ANOVA) indicated that the factors, other than *K*, have a significant effect on the performance metrics. The p_c however only has a significant effect on the all-pair minimum bandwidth. Table 6-1 summarizes the ANOVA results.

Table 6-1 ANOVA analysis for NDBGA parameters

Factor	DF	Metric	SS	MS	F	P
μ	3	P_1	12187	4062	16.76	0.000
		P_2	9805	3268.4	67.05	0.000
		P_3	1401707	467236	179.31	0.000
q	3	P_1	1796	599	2.47	0.060
		P_2	11576.7	3858.9	79.16	0.000
		P_3	2048604	682868	262.06	0.000
b_m	3	P_1	20205	6735	27.79	0.000
		P_2	19166.7	6388.9	131.06	0.000
		P_3	3441697	1147232	440.26	0.000
p_c	1	P_1	3	3	0.01	0.915
		P_2	241.8	241.8	4.96	0.026
		P_3	24343	24343	9.34	0.002
K	1	P_1	120	120	0.49	0.482
		P_2	36.4	36.4	0.75	0.388
		P_3	665	665	0.26	0.613
Problem Instance	4	P_1	1204605	301151	1242.61	0.000
		P_2	65963.3	16490.8	338.30	0.000
		P_3	37736641	9434160	3620.46	0.000
Error	6384	P_1	1547188	242		
		P_2	311199.1	48.7		
		P_3	16635388	2606		
Total	6399	P_1	2786104			
		P_2	417989.3			
		P_3	61289046			

المنارة للاستشارات
www.manaraa.com

Selecting a parameter level combination with three performance measures requires a decision making process since this is a multi-objective criteria. A factor level combination only dominates another one if it performs better in terms of all objectives. The analysis of the above FLC revealed 34 non-dominated options. The non-dominated factor level combination set is then sorted with respect to the performance metric P_1, and the one with the following settings is selected as the NDBGA parameter set due to its satisfactory performance with respect to P_2 and P_3 while being among the top with respect to P_1.

Population size (μ):	90
Tournament size (q):	3
Bit mutation rate (b_m):	0.03
Crossover rate (p_c):	0.70
K	0.05

6.3 The PSO for Mobile Agent Location Optimization

Due to PSO's advantages discussed in Section 2.8.2, it is selected to be an alternative tool for the mobile agent location optimization problem. Since the PSO is proven to be an effective tool for continuous optimization, it provides a good comparison opportunity for the NDBGA. The benefits and disadvantages of using PSO over NDBGA or vice versa are discussed in Section 7.3.

المنارة للاستشارات
www.manaraa.com

The pseudo code of the PSO implementation for mobile agent location optimization:

```
Start{
    Initialize $t = t_o$
    Read $XY_t$
    $G = -M$
    $P = -M$ for all particles
    Set $r_{it}$ and $v_{it}$ for each agent $i$ as decision variables, each encoded as a real number.
    Do{
      Initialize swarm particles' $X$, $V$ for ( each swarm particle ){
            if ( $t \neq t_o$ AND $mbr = 1$) then {
                  Transfer best particle's $X$, $P$ from time ($t$-1)
            }else{
                  for ( each $r_{it}$ variable ) $X \leftarrow U(0,2\pi)$
                  for ( each $v_{it}$ variable ) $X \leftarrow U(0, v_{\max_i})$
            }
            for ( each $r_{it}$ variable ) $V_{max} = 2\pi$, $V \leftarrow U(-2\pi,2\pi)$
            for ( each $v_{it}$ variable ) $V_{max} = v_{max}$, $V \leftarrow U(v_{\max_i}, v_{\max_i})$
      }
      Evaluate fitness for ( each swarm particle ){
            Calculate the $x_{i(t+1)}$ and $y_{i(t+1)}$ coordinates suggested by the solution at ($t$+1)
            for every agent $i$ using decoded $r_{it}$ and $v_{it}$ values as given in equations (
            3-10 ) and ( 3-11 ).
            Form MANET topology, $G_{t+1}$, and calculate link capacities.
            Calculate objective:
            Set fitness $F$ = MANET performance metric (Equation ( 3-7 ) )
            if ( $F$ is better than particle's $P$ ) then {
```

www.manaraa.com

```
            Set $P = F$
        }
        if ( $P$ is better than $G$ ) then {
            Set $G = P$
        }
    }
    Set $\omega = 1.5$
    Loop{
        for ( each swarm particle ){
            if (U(0,1) < 0.02) then {
                Set $\omega = 1.5$
            }
            Set $R_1$ = U(0,1)
            Set $R_2$ = U(0,1)
            Set $\varphi = \varphi_1 R_1 + \varphi_2 R_2$
            Set $K = \begin{cases} \frac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}} & \text{for } \varphi > 4 \\ 1 & \text{otherwise} \end{cases}$
            Set $V = K \cdot [\omega \cdot V + \varphi_1 R_1 \cdot (P - X) + \varphi_2 R_2 \cdot (G - X)]$
            Scale $V$ down, if any of its elements is beyond the corresponding $V_{max}$ limit, by $\max_{\forall \mathrm{var}} \left\{ V_{\mathrm{var}} \big/ (V_{\max})_{\mathrm{var}} \right\}$.
            Set $X = X + V$
        }
        Evaluate fitness for ( each swarm particle )

    }While $^{(Loop)}$ (stopping criteria is not met)

Update $(x_{i(t+1)}, y_{i(t+1)})$ $\forall i \in AN_t$ using $G$
```

www.manaraa.com

```
    Deploy mobile agents to $(x_{i(t+1)},y_{i(t+1)})\ \forall i \in AN_t$
    Set $t = t + 1$
    }While $^{(Do)}$ (User nodes are active)
}End
```

The PSO for mobile agent location optimization takes two decision variables per agent, the direction and the magnitude of its velocity. Every particle in the swarm represents one possible movement scenario for the mobile agent nodes. The fitness evaluation involves generating the network that corresponds to the movement scenario, finding the link capacities and calculating the objective function, as described in the "Evaluate fitness" routine in the above pseudo code. This is followed by updating the ***P***, ***G***, and the ***V*** vectors, and finally the ***X*** vector for each particle. This loop continues until the stopping criteria is met, which is described in Section 6.4.

6.3.1 PSO Parameters

As discussed in Section 2.8.2.2, the PSO performance is enhanced by the introduction of the constriction coefficient, and setting the social and cognition parameters such that they sum up to a number larger than 4. The common practice is to set them to 2.05 each, which has been done in this study also [17]. Additionally, a dynamic weight inertia strategy is applied with an initial ω_0 value of 1.5, and decreased geometrically by a coefficient of 0.98 at every iteration. Although the common practice is to let the inertia coefficient decrease monotonically, in preliminary experimentation it was found beneficial to randomly reset it back to its high level. This improves the ability

المنارة للاستشارات

www.manaraa.com

to escape local optima by “exciting” the particles every now and then, helping them swarm to other regions. The inertia coefficient is reset to its original value of 1.5, with a probability of 0.02, at each iteration.

The swarm particle velocities are limited to the variable limits, which is also a common practice. One important thing that needs to be noted on limiting the velocities is that the swarm particle velocity vector is scaled down entirely. That is, all of its elements are scaled down rather than only the ones that exceed the limit. The scale factor is calculated using the velocity element that has the largest deviation from the maximum allowed velocity limit.

The global neighborhood was found to be the most efficient over a wide variety of continuous test problems as suggested by Carlisle and Dozier [19]. A global neighborhood topology is used for the PSO in this study. For all comparisons, the PSO and the NDBGA population sizes are kept equal.

6.4 Stopping Criteria

Since the heuristic optimization algorithms are expected to return a “good solution” which usually cannot be tested for optimality, there is one or more stopping criteria used. In this study, a strategy is used which records the function evaluation count every time there is an improvement on the best known solution. If no improvements are detected within a preset number of consecutive function evaluations, the algorithm stops and returns the best found solution so far.

المنارة للاستشارات
www.manaraa.com

The mobile agent location optimizer is designed for two types of scenarios, static and dynamic, which are explained in detail in Section 6. Here, the importance of their differences in terms of the stopping criteria is discussed.

The static problems and the dynamic problems are almost identical except for the problem complexity. The static problems have a significantly larger search space due to the larger velocity constraints of the mobile agents. On the other hand, the dynamic problems are made up of many static problems that have agents with infinitesimal velocity constraints.

This property brings an advantage when solving individual time steps of the dynamic problems because the stopping criteria requirements can be lowered drastically without a significant loss of performance. According to preliminary experiments, the sufficient stopping criteria are found as follows: For the static problems, both the NDBGA and the PSO algorithms stop and return a solution if the number of objective function evaluations without an improvement of the best solution found so far is 1000 and for dynamic problems, the requirement for each time step is set equal to 200.

6.5 Semi-intelligent Agent Behavior

The heuristic mobile agent location optimizer requires the agents to be an integral part of the MANET in order to calculate the objective function. If, for some reason, an agent falls back, or becomes disconnected from the network, it has to be able to move independently and catch up with the MANET to be properly utilized again.

المنارة للاستشارات

www.manaraa.com

To achieve this, a simple method is developed. Every mobile agent keeps track of the coordinates of the users it can communicate with, including multi-hop communications. The center point of the coordinates of these users is treated as an attractive target to move towards in case an agent becomes isolated. The last recorded velocity of the target location, which is the difference between the target at time t and time t-1, is also preserved in order to be able to predict the change in the target location while the agent is isolated.

An agent that cannot communicate with any user node, or an agent that has a node degree of 1 or 0 ($n_d < 2$), it is considered an isolated agent. If an agent becomes isolated, then it will self deploy towards the last recorded target coordinate until it catches up with the network, becomes a node with $n_d \geq 2$ and it is able to communicate with at least one user node.

www.manaraa.com

The pseudo code for the semi-intelligent agent behavior is as follows:

```
For any agent j{
    If( $t = t_0$ ) then { $XY_{jt}^{T} = (0,0)$ }
    At any time t{
        If( $|UN_t^j| > 1$ ) then {
                $XY_{jt}^{T} = C(UN_t^j)$
        }else{
            If( $t > 2$ ) then {
                $XY_{jt}^{T} = 2 \cdot XY_{j(t-1)}^{T} - XY_{j(t-2)}^{T}$
            }else{
                $XY_{jt}^{T} = (x_{jt}, y_{jt})$
            }
        }
        If( $d(j) \leq 1$ OR $|UN_t^j| \leq 1$ ) then {
            Self deploy agent j to $XY_{jt}^{T}$
        }else{
            Obey mobile agent location optimizer system
        }
    }
}
```

Where XY_{jt}^{T} is the calculated target coordinate for self deployment of agent *j* at time *t*, UN_t^j is the set of user nodes that agent *j* can communicate with (not necessarily within

www.manaraa.com

the immediate range), $C(UN_t^j)$ is the coordinate center of all users in UN_t^j, (x_{jt}, y_{jt}) is the position of agent j at time t, and $d(j)$ is the node degree of agent j.

In this chapter, the two different problem scenarios for the mobile agent location optimization problem, static and dynamic, are described. The details of the NDBGA and the PSO algorithms are given. The determination of the NDBGA algorithm parameters is done by a full factorial experiment and a multi-criteria decision is made by taking a very high performing parameter combination.

The dynamic mobile agent location optimization can be enhanced by making use of future user location prediction and information gained during past time steps. These are explained in sections 7.3.2 and 7.3.3 in the next chapter.

المنارة للاستشارات

www.manaraa.com

CHAPTER 7

TEST PROBLEMS AND RESULTS

The performance analyses of this study are done in a computerized simulation environment. Test problems of static and dynamic natures are generated and the performance of the developed algorithms are analyzed using computer simulation. Test problems for the dynamic cases are generated in three groups based on problem size: small, medium and large. Small sized problems employ 4 users and 3 agents, medium sized problems employ 8 users and 6 agents, and large size problems employ 16 users with 12 agents. The dynamic test problems are generated with a span of 100 time steps.

All test problems are generated and the computer simulations are carried out using the MATLAB technical computing package. To achieve better computational performance, all heuristic optimization computations are coded in C++ and embedded into the simulation platform within MATLAB. The MIP model was constructed using the AMPL modeling language and solved with CPLEX version 9.1, which is also embedded in the MATLAB simulation platform.

للاستشارات المنارة
www.manaraa.com

7.1 Generation of the Static Test Problems

Test problem data involve the locations of the users as well as the initial coordinates of the mobile agents. The mobile agent location optimizer system then deploys the agents within the simulation area. The test problem instance generation code and the test problem parameters necessary to create the test problems are given in Appendix Ap- 2 and Ap- 3, pages 185-188.

7.2 Generation of the Dynamic Test Problems

Test problem data specify the starting locations of the users and the agents, followed by all future users' locations in discrete time steps. Naturally, the mobile agent location optimizer system is only given the user location data of the current time step.

The simulation dimensions are set so that the wireless transmission ranges of all MANET nodes are 1.0 distance unit. Velocity constraint v_{max} for user nodes is 0.05 and v_{min} for user nodes is 0.02. User nodes come to a stop when they reach their destinations. Velocity constraint v_{max} for mobile agents is 0.06 and v_{min} for mobile agents is 0.0. The velocity values indicate Euclidean distance units traveled per one time increment. The simulation area is a two dimensional rectangular area with $X_{min} = 0$, $X_{max} = 5$, $Y_{min} = 0$. $Y_{max} = 5$. Test problem instance generation code and the parameters necessary to create the test problems are given in Appendix Ap- 2 and Ap- 3, pages 185-188.

المنارة للاستشارات
www.manaraa.com

7.2.1 User Mobility Model

User nodes are assigned random destination points and they follow a random path with random perturbations to their directions. Each user is assigned a random velocity U[v_{min}, v_{max}] at each time step. Let uv_{jt} be the unit vector in the direction of the motion of the j^{th} user node at time *t*, and $uvdestin_{jt}$ be the unit vector in the direction of its final destination point from its location at time *t*, as given in equation (7-1). The initial direction of user motion, i.e. uv_{jt_0} , is created randomly for all users.

$$\overset{\rightarrow}{uvdestin}_{jt} = \frac{(x_{jdestin}, y_{jdestin}) - (x_{jt}, y_{jt})}{|(x_{jdestin}, y_{jdestin}) - (x_{jt}, y_{jt})|}$$

(7-1)

The direction angle of each user node, uv_{jt}, is perturbed by a uniformly distributed random number between [-π/4 , π/4] with a 10% probability in each time step before calculating its direction of motion for the next time step, $uv_{j(t+1)}$. The random rotation procedure is given in equations (7-2) through (7-5).

$$\theta = U(-\pi / 4, \pi / 4)$$

(7-2)

$$RotMatrix = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

(7-3)

$$uv^*_{jt} = uv_{jt} \times RotMatrix$$

(7-4)

المنارة للاستشارات
www.manaraa.com

where θ is the random rotation in the direction of the motion, *RotMatrix* is the corresponding rotation matrix and uv^{*}_{jt} is the perturbed unit vector in the direction of motion of the j^{th} user at time t.

The directions of user nodes are calculated for each successive time step as shown in equation (7-5).

$$uv_{j(t+1)} = (\alpha)\cdot uv_{jt} + (1-\alpha)\cdot uvdestin_{jt} \tag{7-5}$$

where $\alpha = 0.95$ is the weight factor for the current motion direction, $uv_{j(t+1)}$ is the unit direction vector of the simulated motion of the j^{th} user in the next time step, $uvcurrent_{jt}$ is the current direction vector, and $uvdestin_{jt}$ is the direction towards the destination of the j^{th} user from its current location.

This mobility model is similar to the random waypoint model, but different in the sense that the user nodes try to reach a certain destination. The simulated motion resembles the case as if the users are searching for their destination or making their way around forbidden areas or obstacles, which is a reasonable representation of a search and rescue or a military operation. Other mobility models could be readily used since the motion strategy is not an input.

7.3 Performance Metrics and Use of Dynamic Information

This section is organized as follows; first, analyses are presented on future location prediction and the usage of the best solution from the previous time step in Section 7.3.1.

المنارة للاستشارات
www.manaraa.com

Then, comparisons of the NDBGA and PSO heuristics and CPLEX as the MIP solver are made on static and dynamic test problem cases in Section 7.4.

The performance criteria that the comparisons are based on are the three metrics given in Section 6.2.1 and, in the order of importance, they are:

1) The average percent of other users that one user can communicate with, P_1 (%), given in equation (7-6).

$$P_1 = \frac{\sum_{t=1}^{t_f}\left(\frac{\sum_{i,j\in UN:j\neq i} z_{ijt}}{n_u\cdot(n_u-1)}\right)}{t_f}\cdot 100$$

(7-6)

where,

t_f is the final time step in the problem simulation. For static problems, $t_f = 1$.

$$z_{ijt} = \begin{cases} 1 & \text{if there is a path between the } i^{th} \text{ and the } j^{th} \text{ user at time } t, \\ 0 & \text{otherwise} \end{cases} \qquad \forall i, j \in UN_t$$

2) The average all-pair minimum bandwidth, P_2 (Mbps), given in equation (7-7).

$$P_2 = \frac{\sum_{t=1}^{t_f}\min_{i,j\in UN_t:j>i}\{MaxFlow(G_t,i,j):MaxFlow(G_t,i,j)>0\}}{t_f}$$

(7-7)

www.manaraa.com

3) The average total bandwidth, P3 (Mbps), given in equation (7-8).

$$P_3 = \frac{\sum_{t=1}^{t_f} \left(\sum_{i,j \in UN : j>i} MaxFlow(G_t, i, j) \right)}{t_f} \qquad (7\text{-}8)$$

The performance measures given in equations (7-6) through (7-8) are indicators of algorithm performance over the entire simulation time span. P_1 is the average percent user connectivity metric, in which it is possible to weight to give more or less importance to users' connectivity properties. For dynamic problems, the P_1, P_2 and P_3 metrics reflect the average performances over the entire time span, which form the basis of algorithm performance comparisons for dynamic problems.

All of the above listed performance measures are "the higher the better" type of performance measures. A fourth performance measure, the time for the optimizers to return a solution, is also recorded.

7.3.1 Dynamic Problem Analyses

The dynamic have unique characteristics that, if made use of, enable some additional inputs to the optimizer to enhance performance. This performance increase is realized in both the solution found and in the solution time.

The two additional inputs used are the future predicted locations of MANET users and the transfer of the best solution at time *t* to time *t*+1 during the initialization of the

www.manaraa.com

population at time *t*+1. The results of these two analyses are given in Sections 7.3.2 and 7.3.3.

7.3.2 The Effect of Future Location Prediction

The optimization of the mobile agents in the MANET is done according to the user location data. At each time step, the user locations are collected, and the optimum agent locations are determined. As the agents relocate to their calculated locations, users also move to their next coordinates, which are used at the next time step.

Optimizing agent locations using current user coordinates can be seen in a sense as the agents following the users' movements on the field. This actually can be aided by forecasting the coordinates of the users at a specific prediction horizon by using the past location information, as described in Section 3.4, and supplying this information to the agent location optimizer instead of the current user location data.

To test for the effect of using predicted user location in mobile agent location optimization, prediction horizons (H) of 1, 2, 3, 4, 5, 6, 7 and 8 time steps are analyzed on 5 medium size test problems, with 5 replications per problem with different random seeds for the heuristic algorithm, for a total of 200 runs.

The location prediction did not have any adverse effects on the average percent of users that could communicate. Furthermore, the results show that a horizon of $H = 4$ is an optimal setting which not only helps increase the average total and the minimum bandwidth of the MANET but also decrease the average time it takes to return the solutions. Figure 7-1 and Figure 7-2 show the average total and minimum bandwidth of

www.manaraa.com

all runs at corresponding *H* levels. Figure 7-3 shows the average time it takes to solve a 50 time step problem with corresponding levels of *H*.

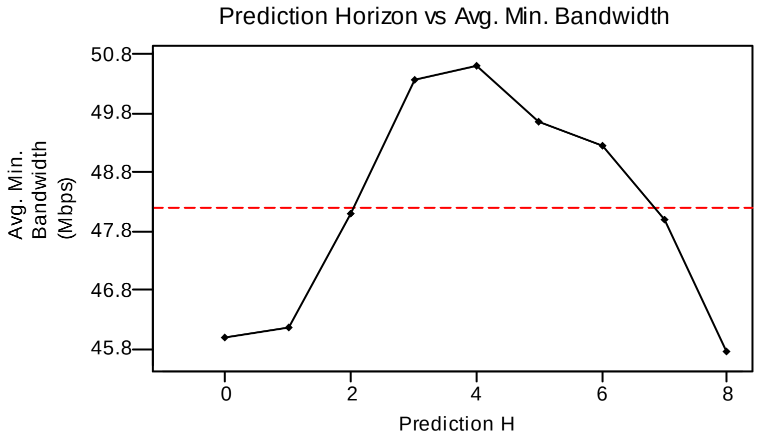


Figure 7-1 Prediction horizon *H* versus the average minimum bandwidth

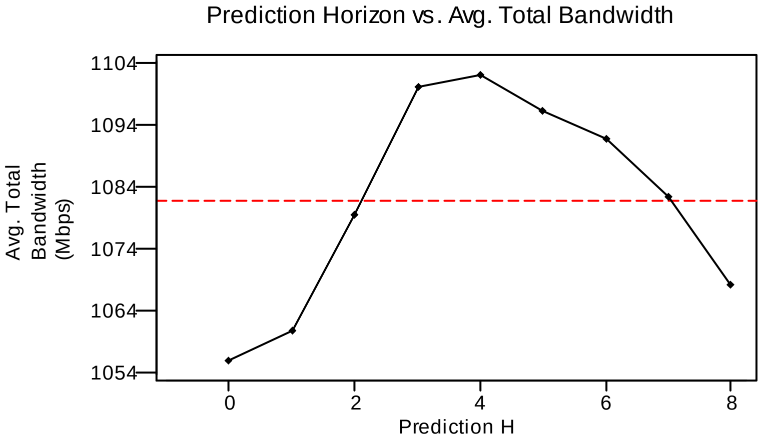


Figure 7-2 Prediction horizon *H* versus the average total bandwidth

المنارة للاستشارات
www.manaraa.com

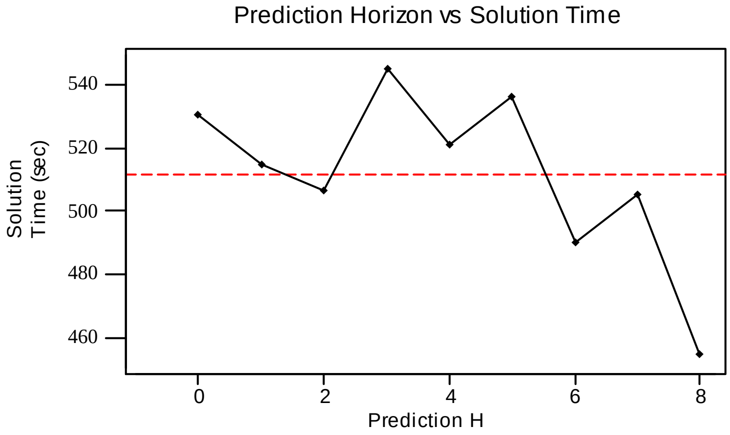


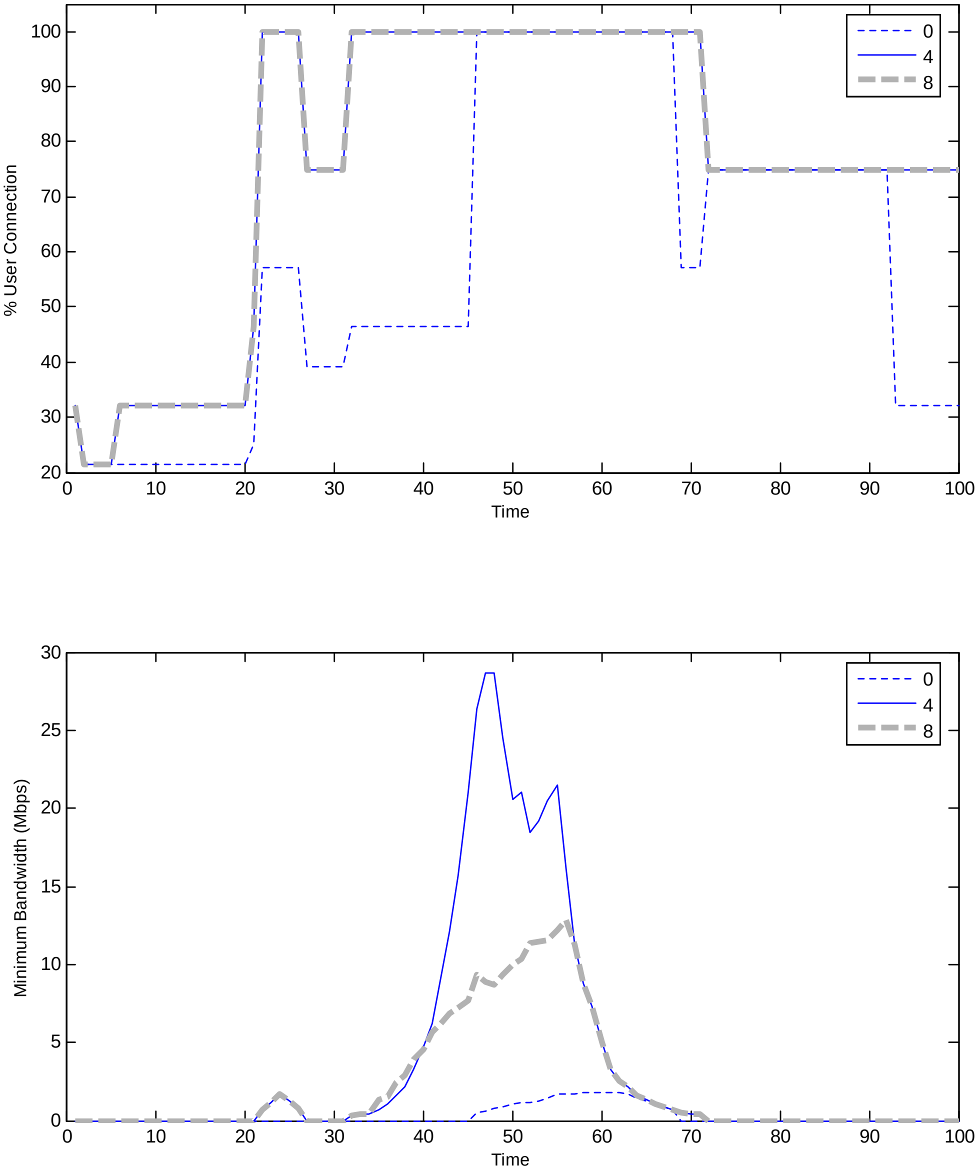
Figure 7-3 Prediction horizon *H* versus the average total solution time

In Figure 7-3, it seems as if there is an improvement in the solution time as H increases. Interpreting this as an improvement would be misleading. It is true that solutions are returned more quickly, but this is because the algorithm cannot improve its best solution. This best solution is inferior to the best solution with smaller prediction horizons.

7.3.2.1 Time Varying Effects

In this part, the effect of future location prediction on the performance metrics at each time step is analyzed for an example problem. Figure 7-4 and Figure 7-5 show the change in performance metrics over time for cases with no location prediction (*H*=0), with prediction of 4 time steps into the future (*H*=4) and with prediction of 8 time steps into the future (*H*=8).

www.manaraa.com



(a)

(b)

Figure 7-4 Change of (a) % user connectivity (b) all pair minimum bandwidth with simulation time

www.manaraa.com

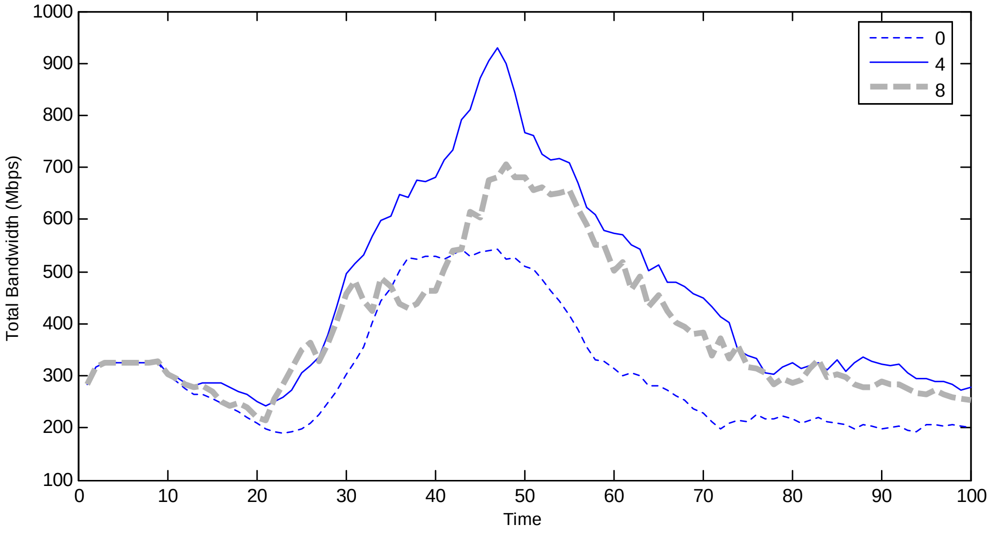


Figure 7-5 Change of total bandwidth with simulation time

Figure 7-6 shows the actual optimized locations of mobile agents for the example problem with different prediction horizons. Three different optimizations are overlaid in the figures. Diamond shape represent user nodes while Δ, o and + represent agents optimized with *H*=0, *H*=4 and *H*=8, respectively.

www.manaraa.com

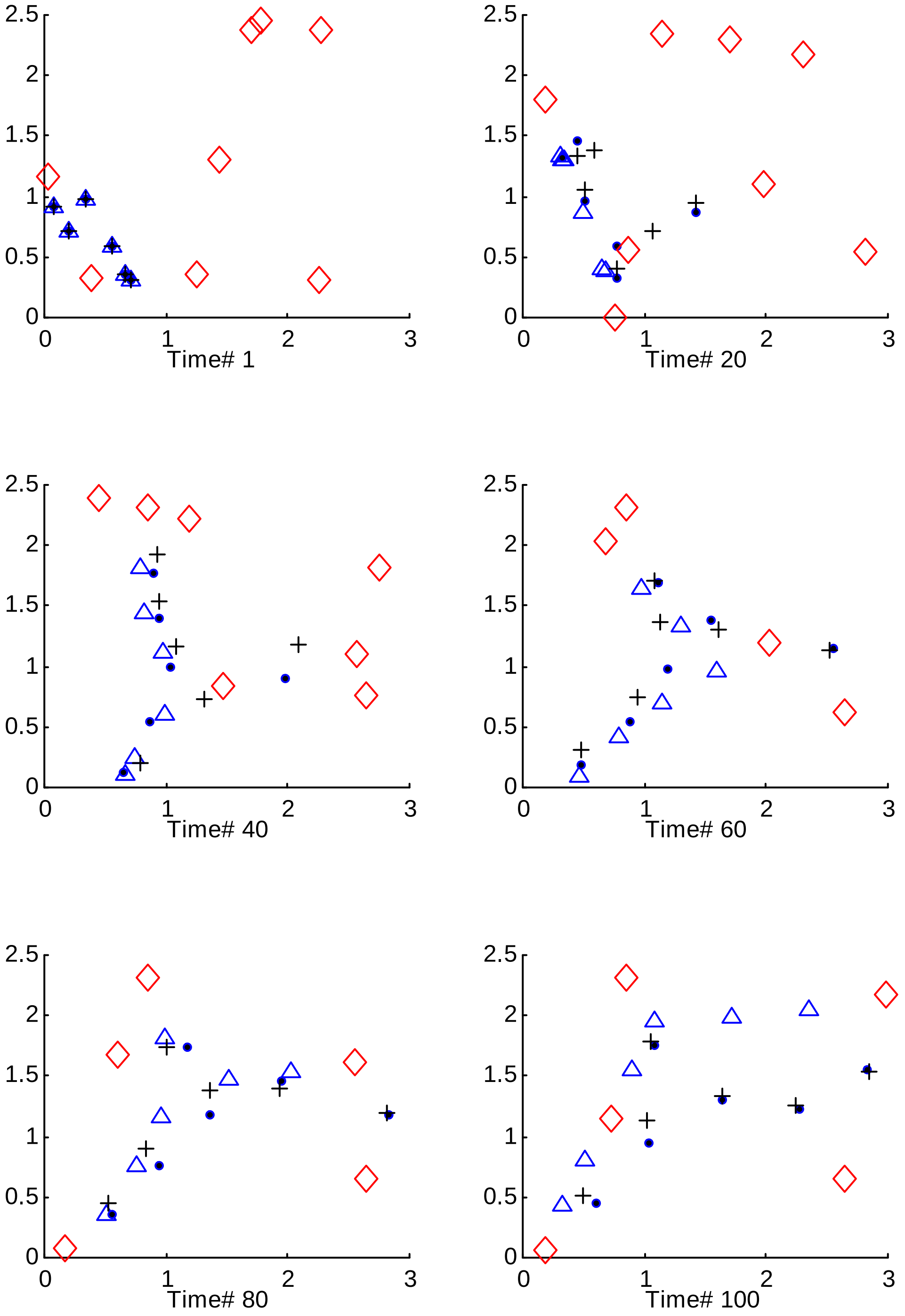


Figure 7-6 Mobile agent behavior with location prediction

www.manaraa.com

7.3.3 The Effect of Using the Best Solutions From the Previous Time Step

Dynamic mobile agent location optimization inherently models a continuous time-space relationship. If the best solution vector from the previous time step (t-1) is fed into the new population when solving for the agent velocity vectors of time t, it is expected to speed up the search process because the best velocities of time t are likely to be correlated with the best mobile velocities of time t-1.

The effect of feeding the last time step's best solutions into the new population is investigated on 5 medium scale test problems with 5 replications per problem. Transferring the only the best solution, top 5 solutions, best half (45) and the entire population (90) is tested for each instance for a total of 125 runs. The results show that the inclusion of the best solutions -no matter how many- from the previous time step has only a minor effect, if any, on the performance measures but a significant reduction, about 30-40%, on the solution time. Figure 7-8, Figure 7-9, and Figure 7-10 show the effects on the three performance measures and the solution time as boxplots.

www.manaraa.com

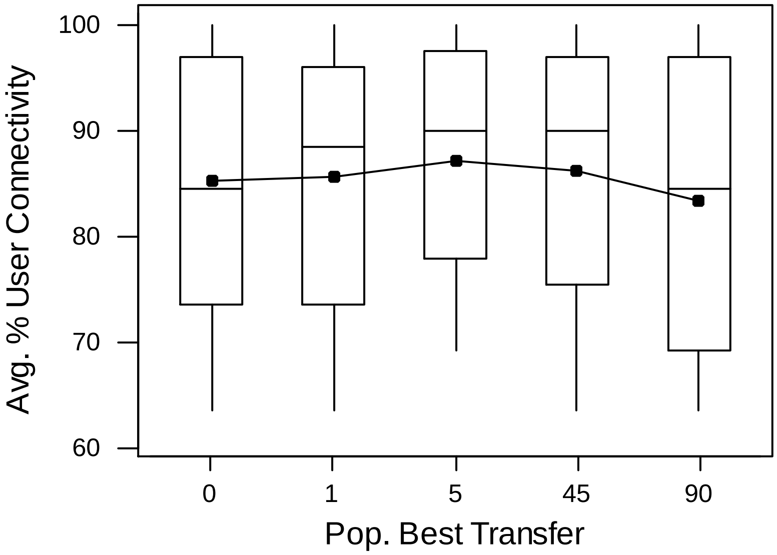


Figure 7-7 The effect of feeding the best solutions from (*t*-1) to *t* on average % user connectivity

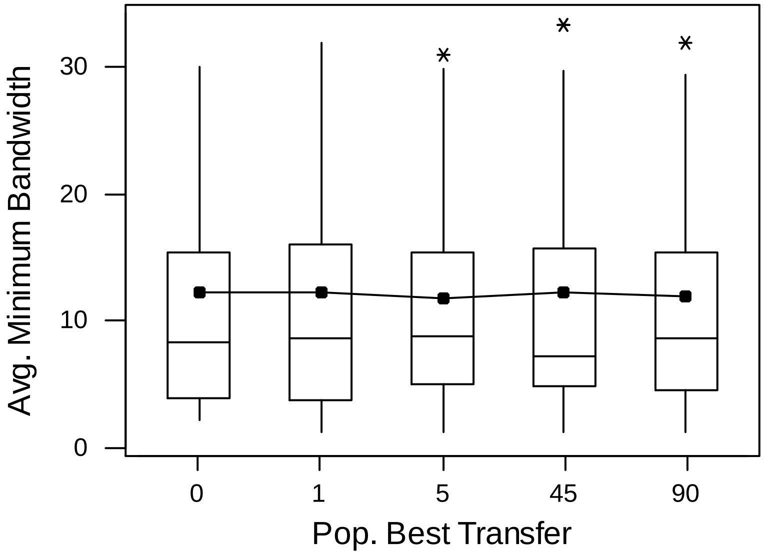


Figure 7-8 The effect of feeding the best solutions from (*t*-1) to *t* on average minimum bandwidth between the MANET user pairs

المنارة للاستشارات

www.manaraa.com

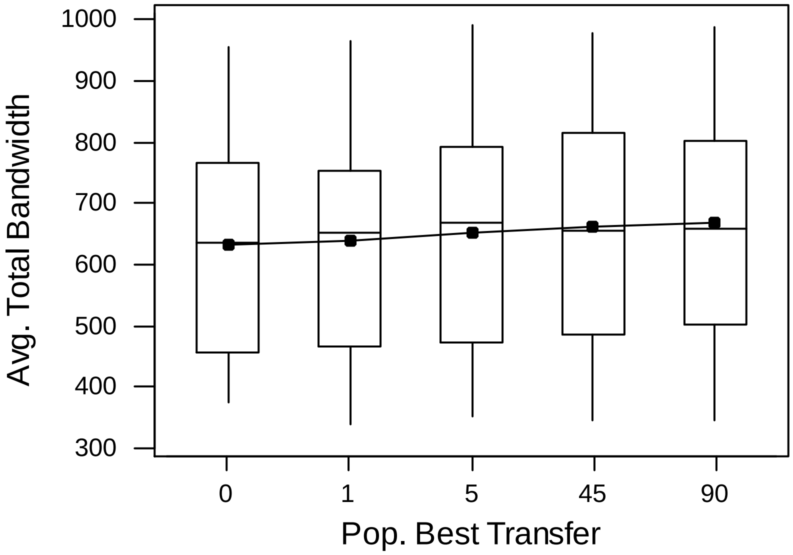


Figure 7-9 The effect of feeding the best solution from (*t*-1) to *t* on average total bandwidth between the MANET user pairs

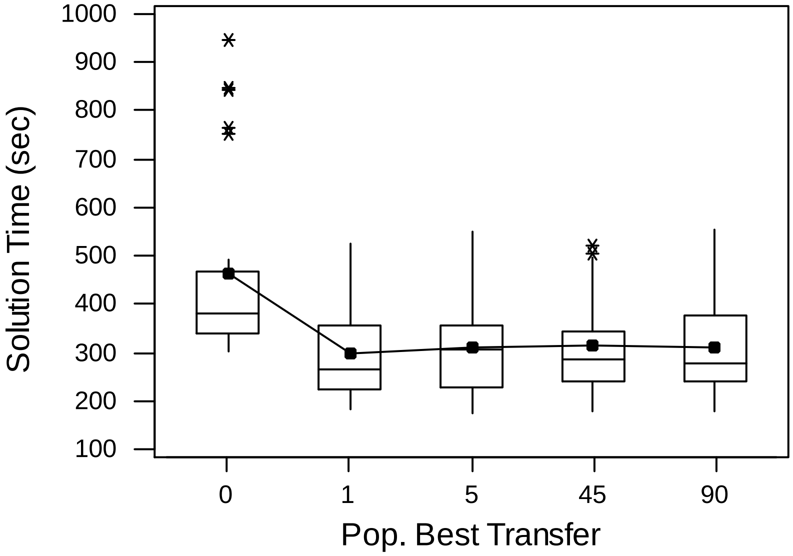


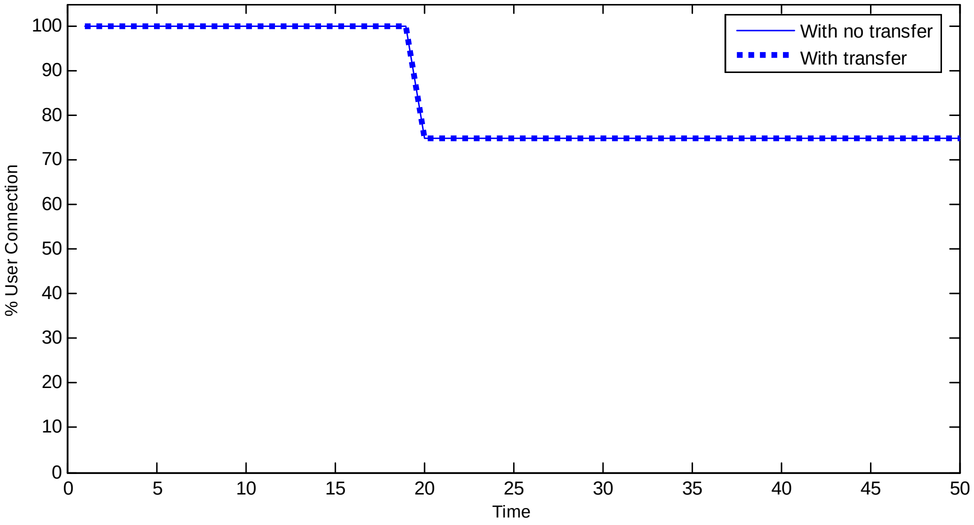
Figure 7-10 The effect of feeding the best solution from (*t*-1) to *t* on average solution time

www.manaraa.com

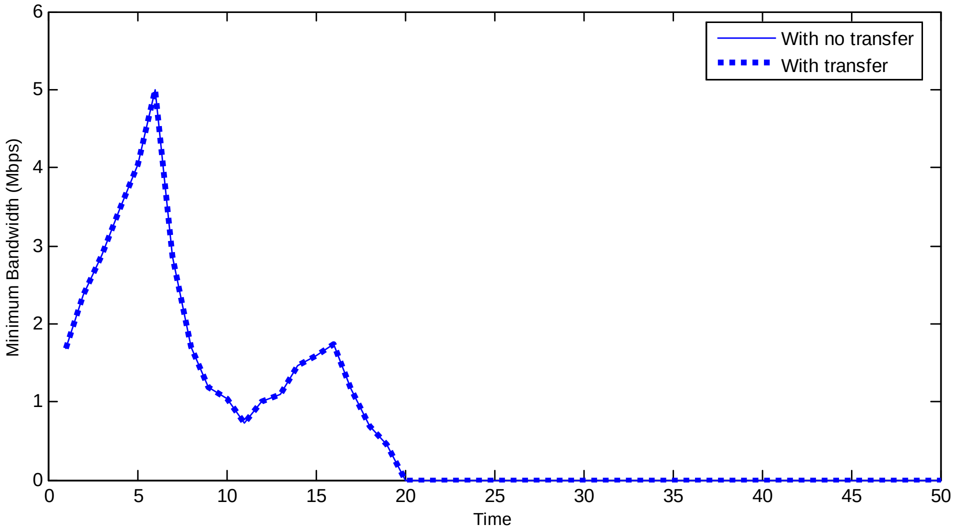
7.3.3.1 Time Varying Effects

In this part, the effect of transferring the population best is investigated at each time step for an example problem. Figure 7-11, Figure 7-12 and Figure 7-13 show the change in performance metrics over simulation time and solution times for cases with no transfer and with transfer of the population best.

المنارة للاستشارات
www.manaraa.com



(a)



(b)

Figure 7-11 Change of (a) % user connectivity (b) all pair minimum bandwidth with simulation time

المنارة للاستشارات
www.manaraa.com

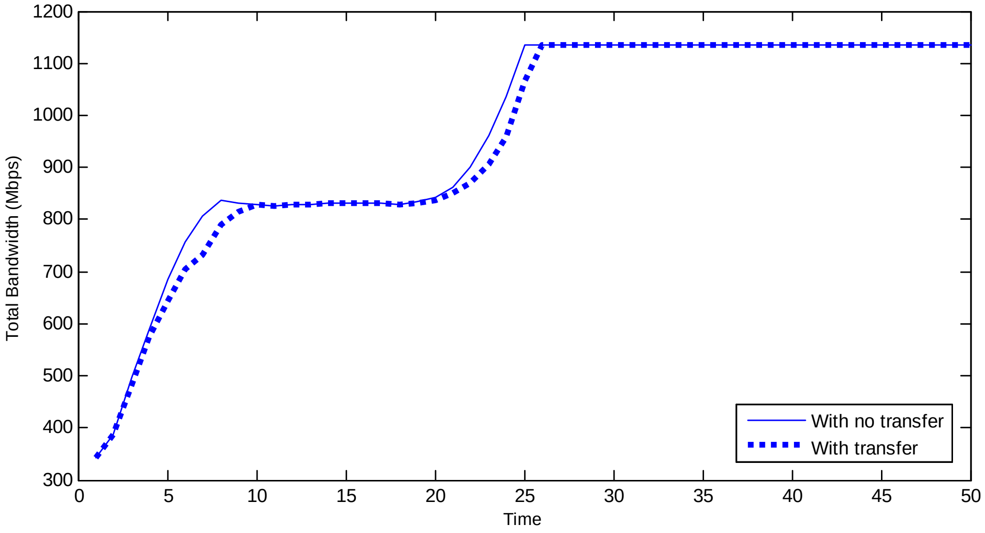


Figure 7-12 Change of total bandwidth with simulation time

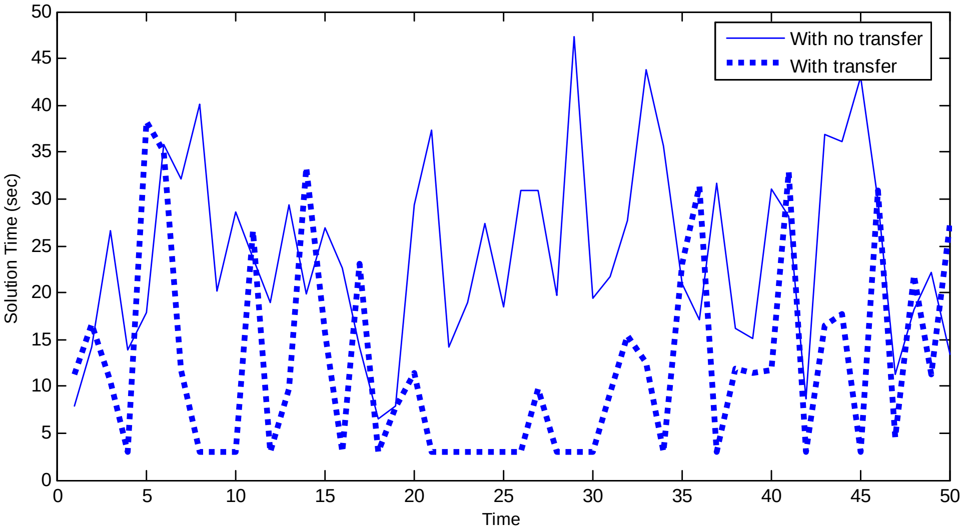


Figure 7-13 Solution time for each time step

As a conclusion, it can be stated that the transfer of the information of the best solution from *t*-1, which is actually the current velocity direction and magnitudes of the

المنارة للاستشارات
www.manaraa.com

mobile agents, improves the solution time significantly. When the effects of transferring the best solution, the top 5, best half or the entire population are analyzed with paired-t tests, the effects of transferring any more than one solution are not statistically significant. Therefore, in order to keep any possible bias to a minimum, transfer of only the population best is set as the default for the rest of this study.

Another result that can be drawn when the effects of future user location prediction and population best transfer are analyzed is as follows: The prediction provides estimation of unknown information and therefore it helps improve solution quality. Transfer of the previous best solution provides memory to maintain previously gained knowledge and it helps speed up reaching a high quality solution in the future time step, which is very important for real time applications.

7.4 Algorithm Performances

In this section, the performances of the NDBGA, the PSO and the CPLEX solvers are compared on various sizes of static and dynamic problems. The boxplot figures are produced with Minitab and a typical boxplot is given in Figure 7-14.

المنارة للاستشارات
www.manaraa.com

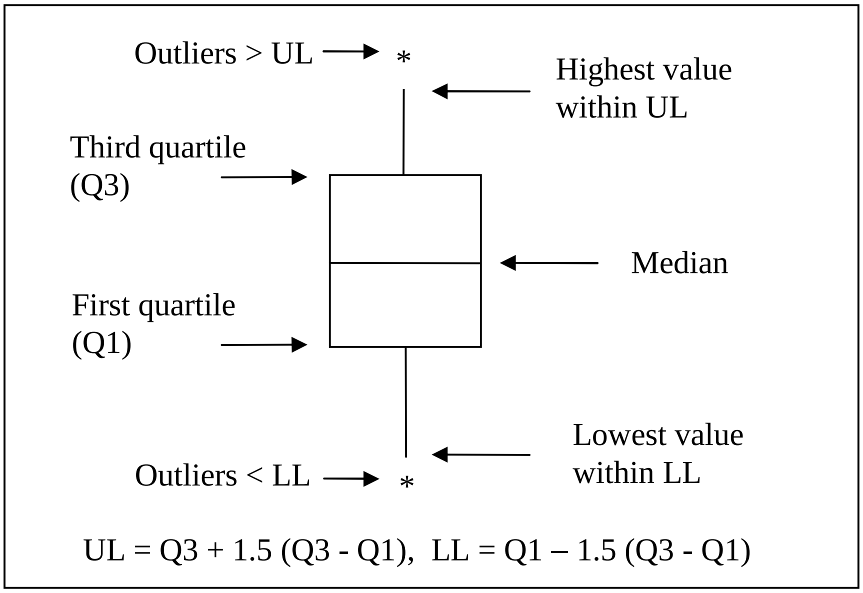


Figure 7-14 Boxplot figure description

7.4.1 Comparisons on Static Scenarios

The algorithms' performance are analyzed and compared with respect to the three measures defined in Section 7.3. The heuristic algorithms are run for 5 replications per problem.

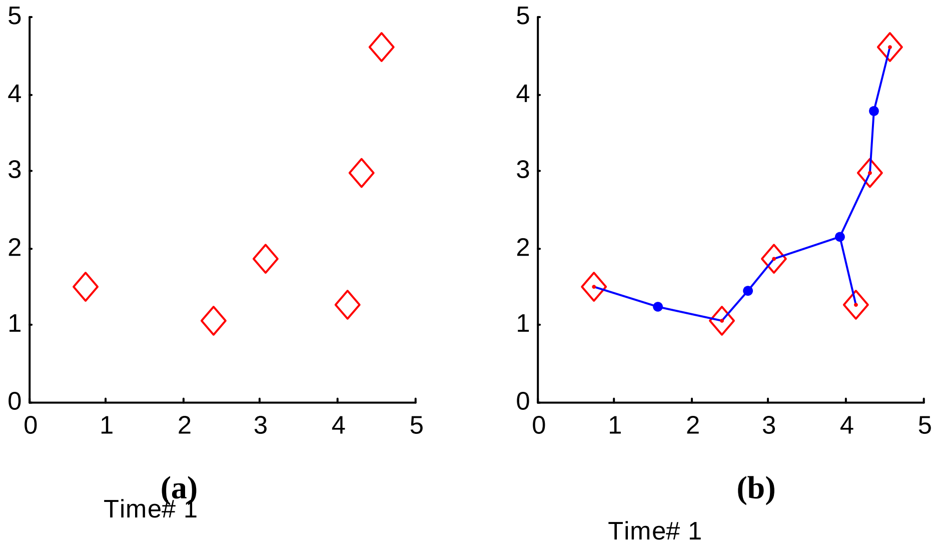
The analyses are grouped into three problem categories; small, medium and large, depending on the number of nodes. The results and comparisons of the algorithm performance are presented as boxplots of all the performance data from all problem instances of same type and size, followed by the graph of the best, average and the worst performances over replications for each problem instance. The problem instances are generated randomly, with the code and random seed data provided in appendices Ap- 2 and Ap- 3. Static users are assigned x and y coordinates drawn from a uniform

المنارة للاستشارات
www.manaraa.com

distribution U(0,XY_{max}), where XY_{max} = 5 units, representing a 5 by 5 square simulation area.

7.4.1.1 Small Static Problems

The comparisons on small scale problems are done on 20 instances with 6 users and 4 agents. Problem instances are generated as explained in Section 7.4.1. An example problem and its solution is given in Figure 7-15. In the figure, the diamond shaped nodes represent user nodes and the solid round shaped nodes represent the agent nodes.



(a)

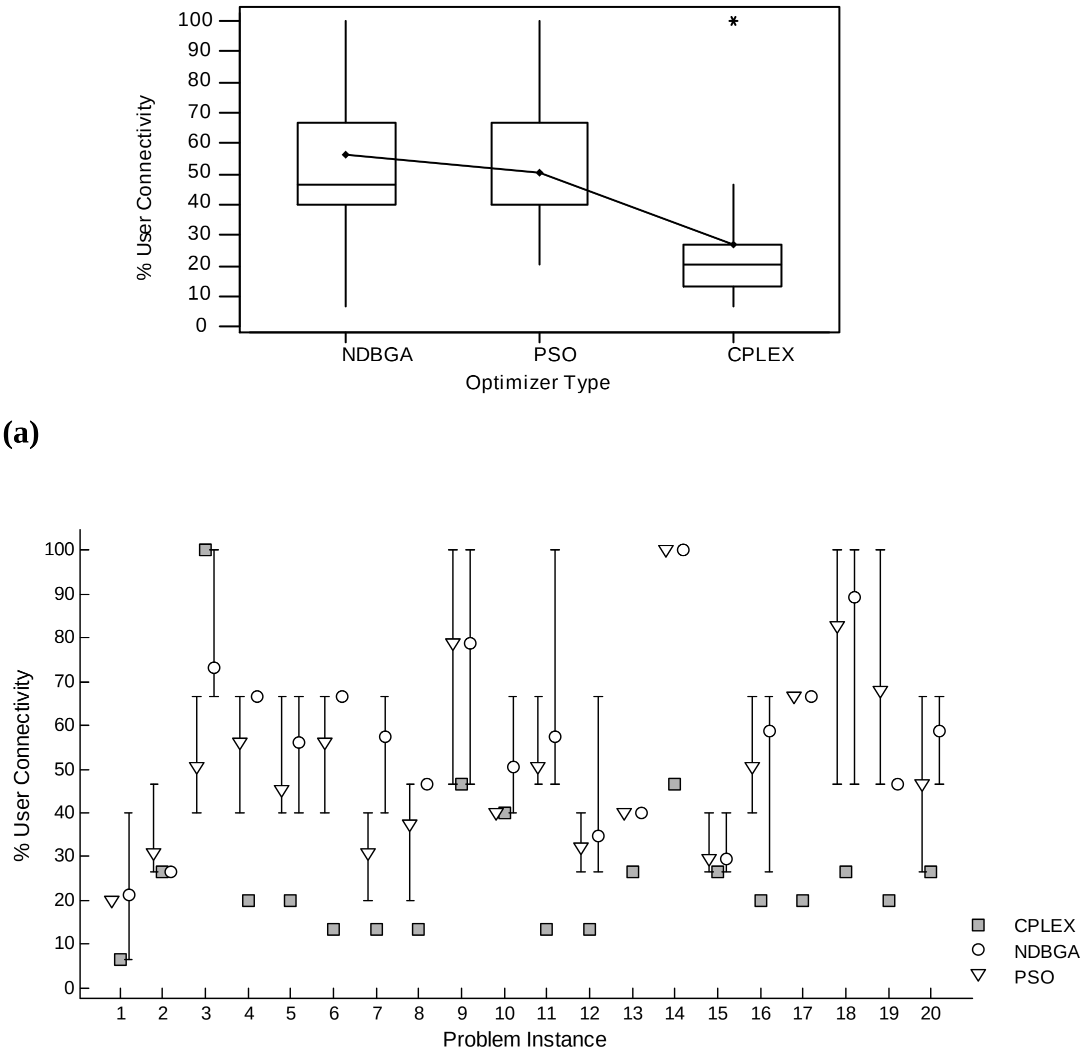
(b)

Figure 7-15 An example problem (a) and its solution (b) for a small static scenario

When the average connectivity results are analyzed, as seen in Figure 7-16, the NDBGA based mobile agent location optimizer performed the best in terms of the average % of user connectivity. NDBGA is followed by the PSO with approximately 9%

المنارة للاستشارات
www.manaraa.com

gap. The CPLEX's performance is much poorer with nearly 50% gap when compared to the NDBGA.



(a)

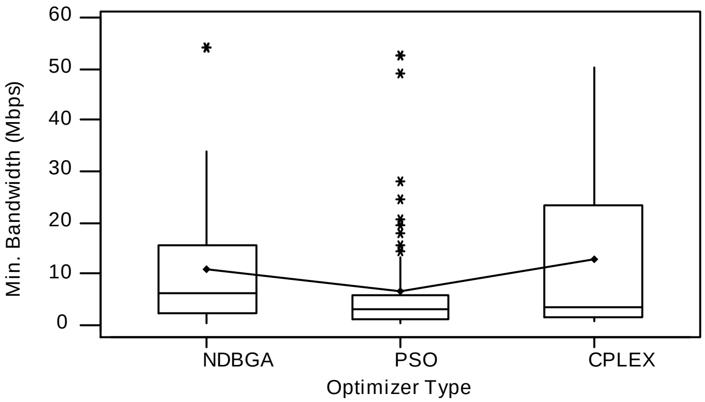
(b)

Figure 7-16 The performance of the GA and the PSO algorithms on small scale static test problems in terms of the % user connectivity. (a) Boxplot (b) Best, average and worst performances over all problem instances

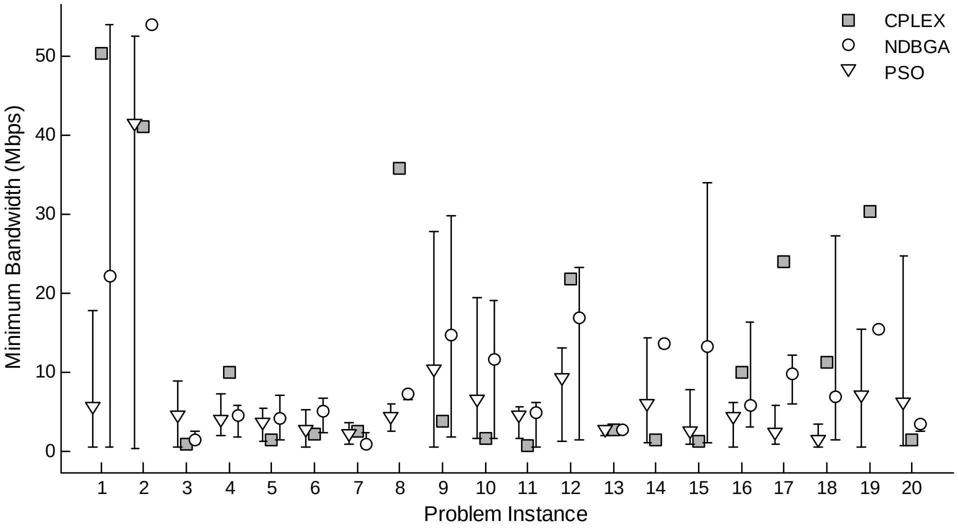
المنارة للاستشارات
www.manaraa.com

In terms of the minimum bandwidth between all user pairs, CPLEX seemingly has done a better job than the heuristics as presented in Figure 7-17, but given its much poorer performance on the most important first criteria, the improvement in the average minimum bandwidth is not a real benefit. The NDBGA lead over the PSO is still valid in this criteria, with approximately 3.5% gap, although the PSO has generated some outliers towards the higher minimum bandwidth.

المنارة للاستشارات
www.manaraa.com



(a)

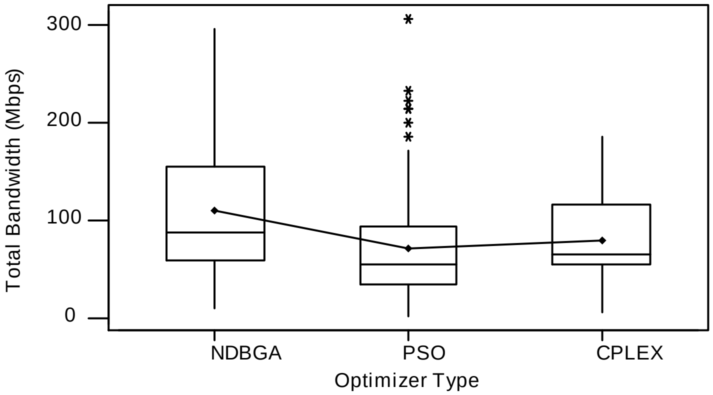


(b)

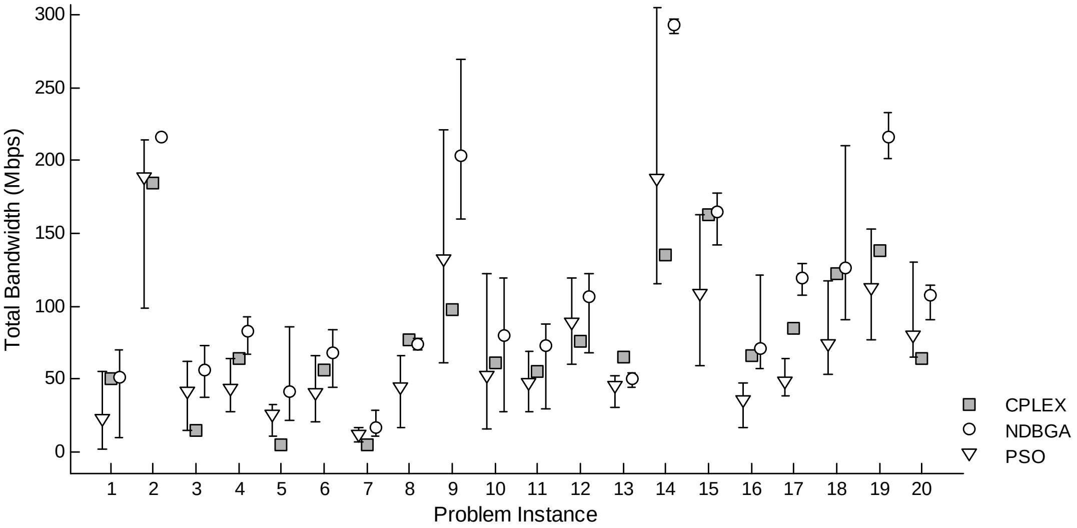
Figure 7-17 The performance of the GA and the PSO algorithms on small scale static test problems in terms of the minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

When the average total bandwidth results are compared, the NDBGA has outperformed the PSO and CPLEX on an average basis, while PSO has produced some better outliers. These results can be seen in Figure 7-18.

www.manaraa.com



(a)



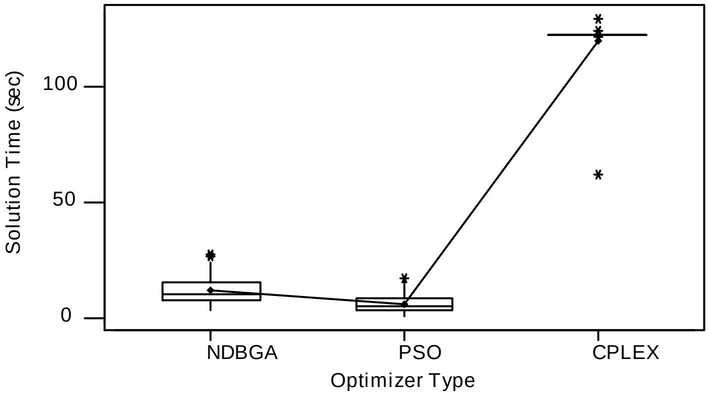
(b)

Figure 7-18 The performance of the GA and the PSO algorithms on small scale static test problems in terms of the total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

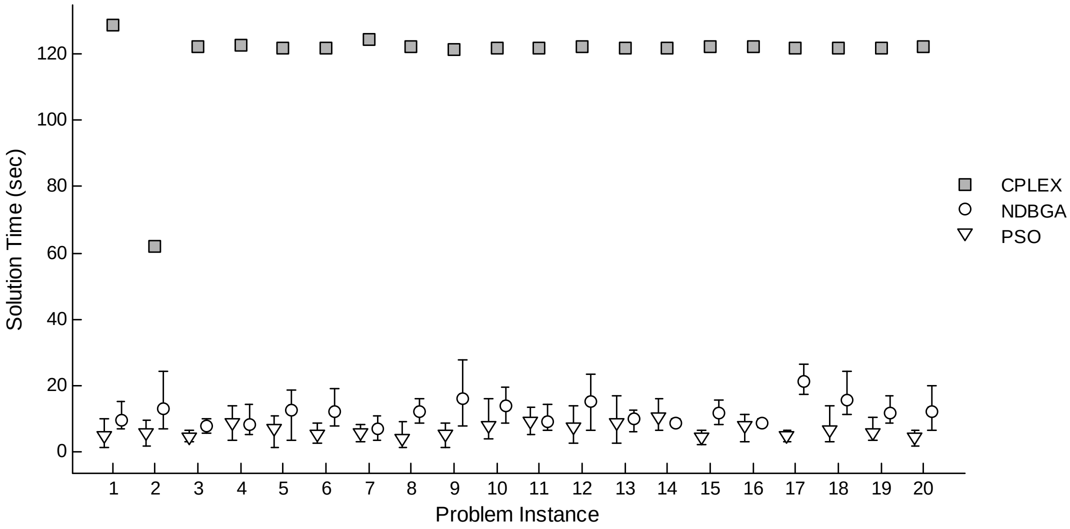
In terms of the solution times, CPLEX recorded the worst performance, which is not surprising. Although the PSO returned slightly quicker solutions on the average, its

المنارة للاستشارات
www.manaraa.com

poorer performance on the connectivity metrics makes the NDBGA the preferred optimizer for this group of problems. The solution time results are presented in Figure 7-19.



(a)



(b)

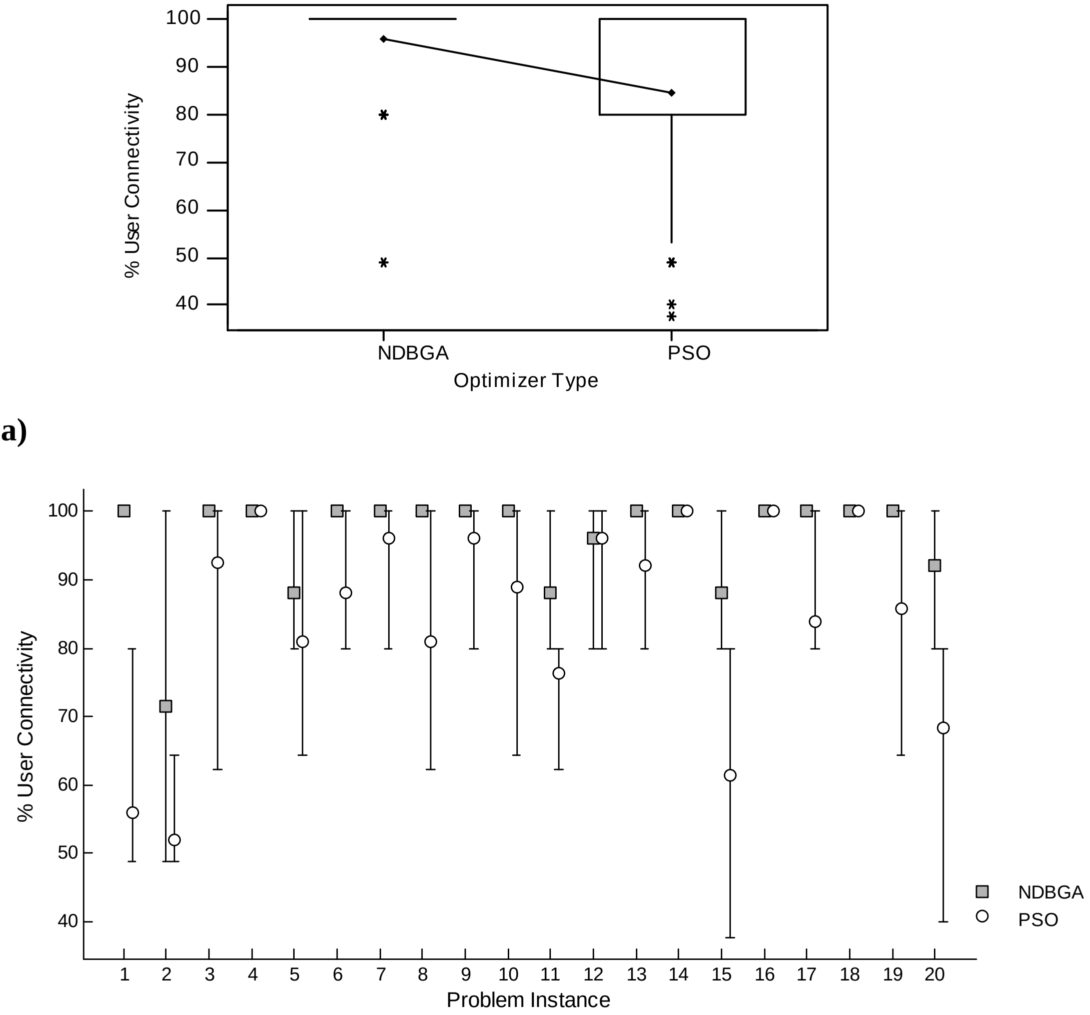
Figure 7-19 The performance of the GA and the PSO algorithms on small scale static problems in terms of the solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances

المنارة للاستشارات
www.manaraa.com

7.4.1.2 Medium Static Problems

The comparisons on medium scale problems are done on 20 test problems with 10 users and 10 agents. Problem instances are generated as explained in Section 7.4.1. Only the heuristic algorithms could be tested on this scale, due to the complexity of the MIP model. For the NDBGA and PSO solutions, each problem is solved 5 times with different random number seeds.

المنارة للاستشارات
www.manaraa.com



(a)

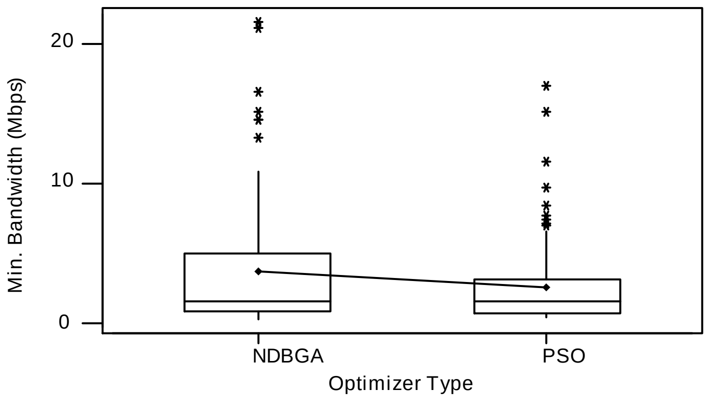
(b)

Figure 7-20 The performance of the GA and the PSO algorithms on medium scale static test problems in terms of the % user connectivity. (a) Boxplot (b) Best, average and worst performances over all problem instances

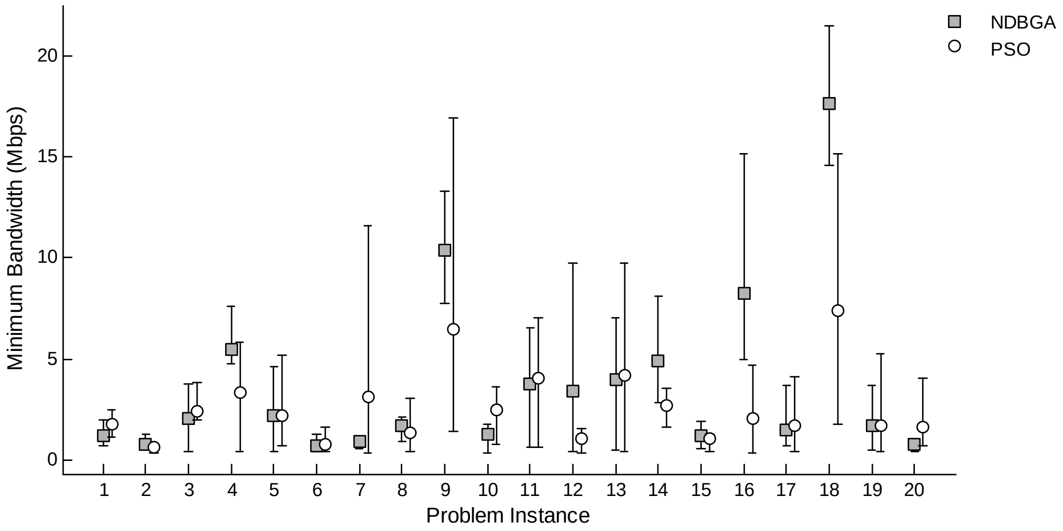
Figure 7-20 presents the average % user connectivity among the network. For this group of problems, the performance of the NDBGA is clearly superior to the PSO. There is an approximately 10% performance gap between the two. The NDBGA has successfully returned solutions with full communication except a few outliers.

المنارة للاستشارات

www.manaraa.com



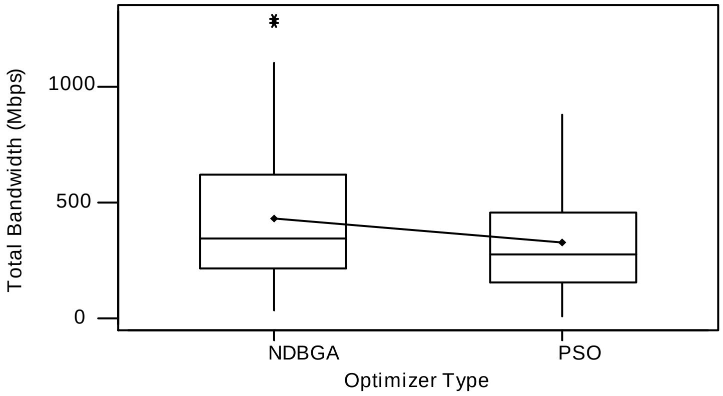
(a)



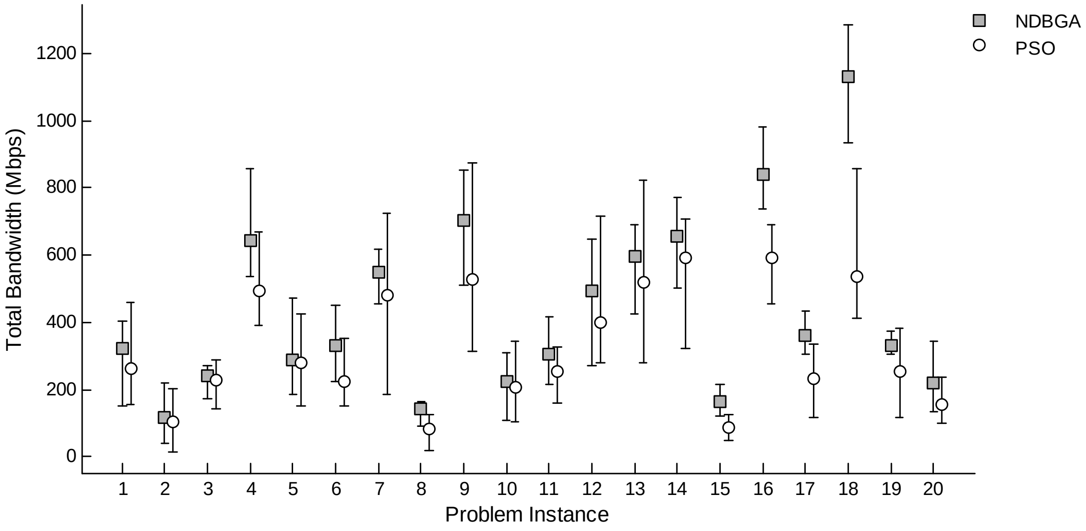
(b)

Figure 7-21 The performance of the GA and the PSO algorithms on medium scale static test problems in terms of the minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

www.manaraa.com



(a)



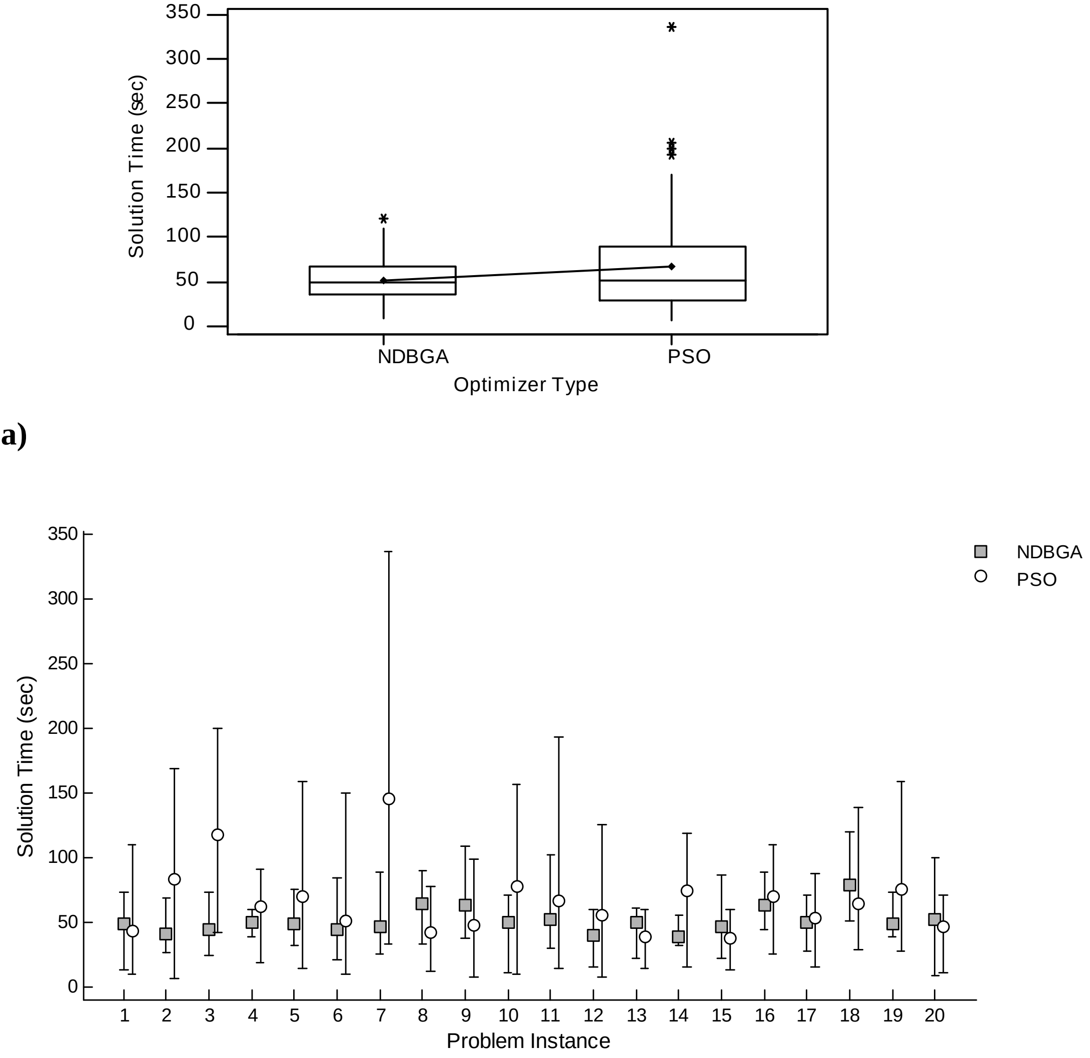
(b)

Figure 7-22 The performance of the GA and the PSO algorithms on medium scale static test problems in terms of the total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

The performance of the NDBGA is also superior in terms of the remaining performance criteria. The average minimum and total bandwidth are both approximately

المنارة للاستشارات
www.manaraa.com

25% higher than the PSO while the solution time is approximately 20% quicker. The results are presented in Figure 7-21, Figure 7-22 and Figure 7-23.



(a)

(b)

Figure 7-23 The performance of the GA and the PSO algorithms on medium scale static problems in terms of the solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances

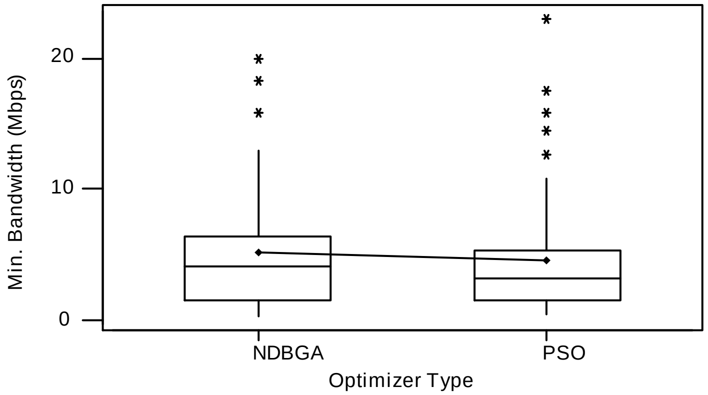
المنارة للاستشارات
www.manaraa.com

7.4.1.3 Large Static Problems

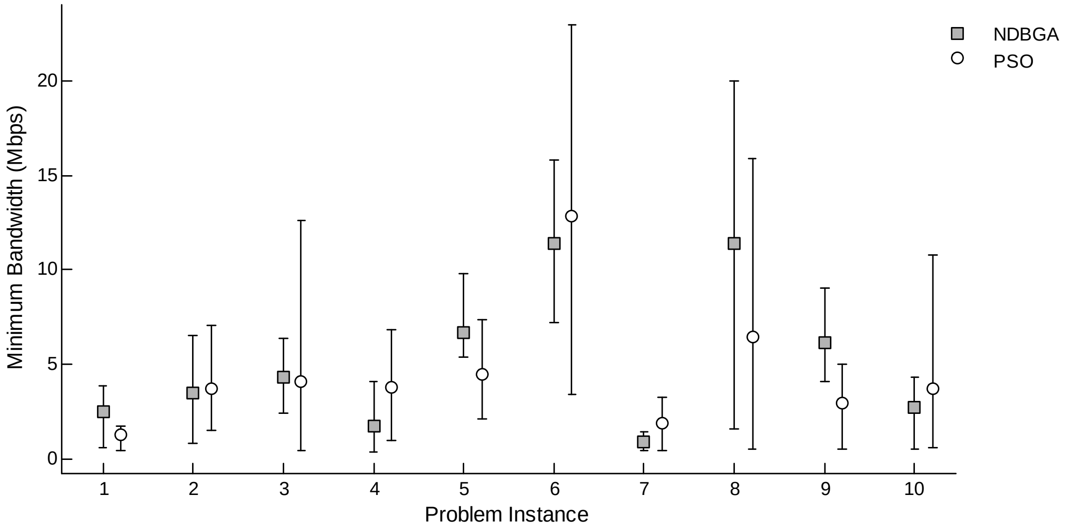
The comparisons on large scale problems are done on 10 test problems with 20 users and 20 agents. Problem instances are generated as explained in Section 7.4.1. Only the heuristic algorithms could be tested on this scale, due to the complexity of the MIP model. For the NDBGA and PSO solutions, each problem is solved 5 times with different random number seeds.

Both algorithms returned solutions with 100% user connectivity for all test runs. Therefore only comparisons on the average minimum bandwidth, average total bandwidth and solution time are presented.

www.manaraa.com



(a)



(b)

Figure 7-24 The performance of the GA and the PSO algorithms on large scale static test problems in terms of the minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

www.manaraa.com

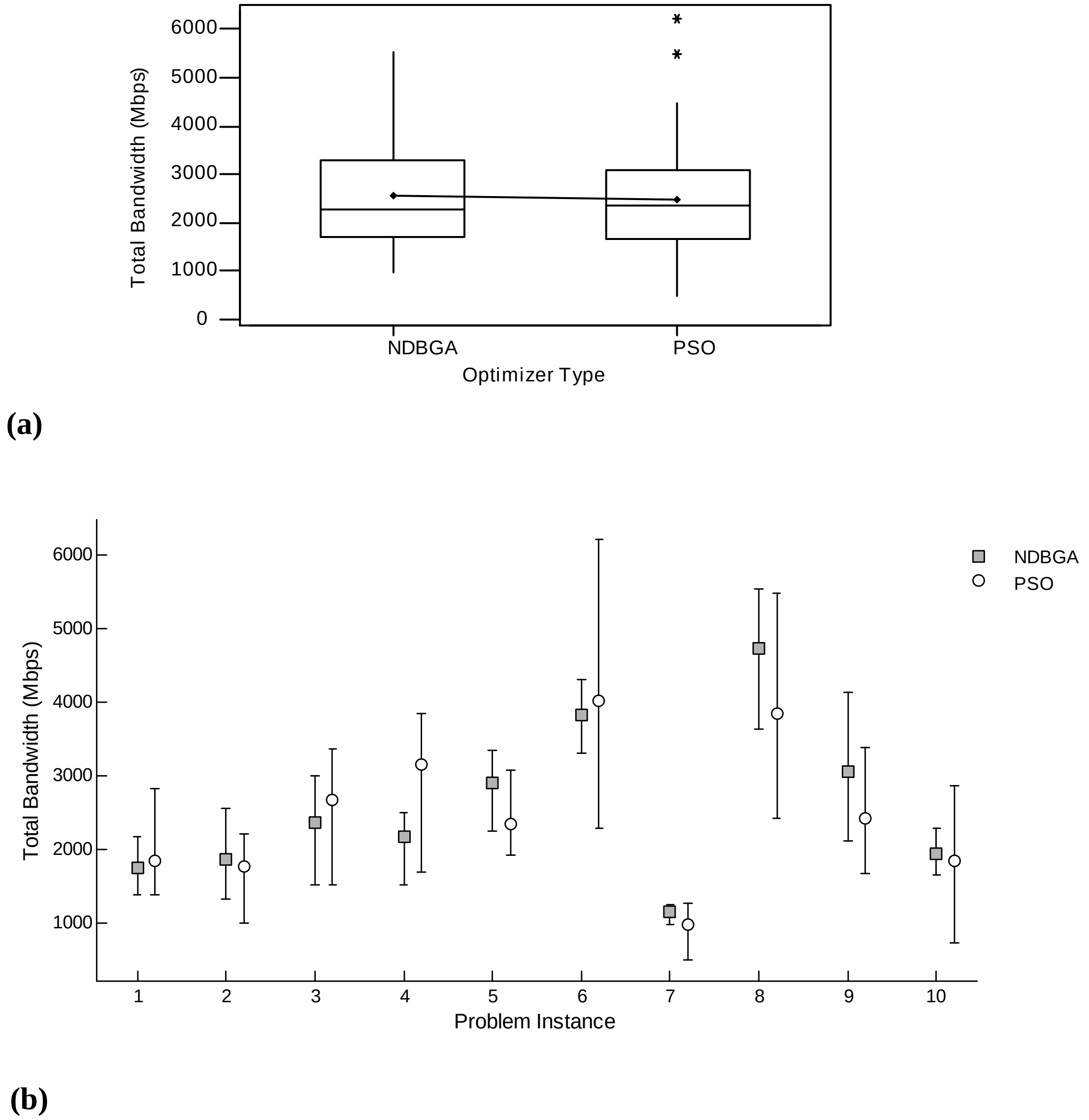


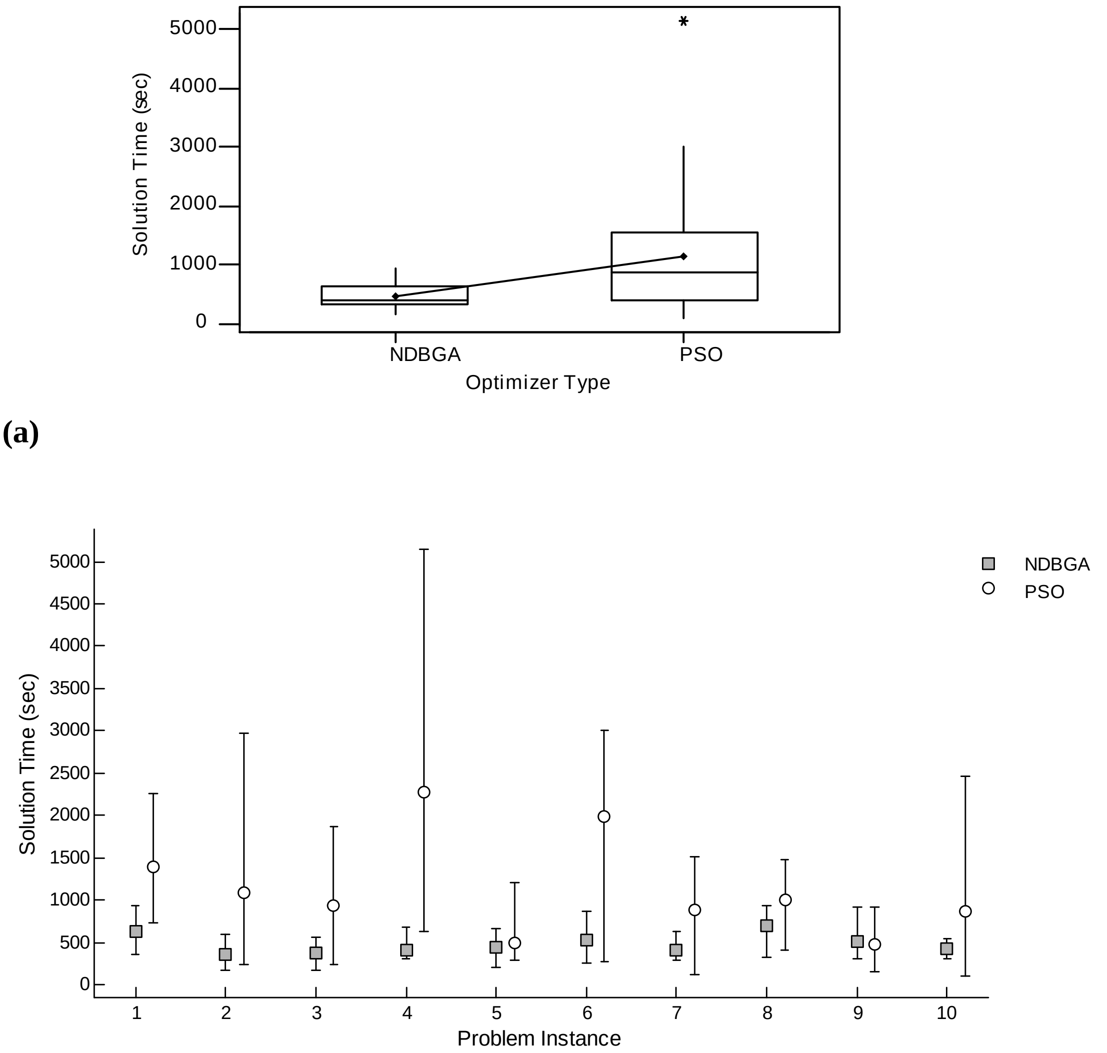
Figure 7-25 The performance of the GA and the PSO algorithms on large scale static test problems in terms of the total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

As seen in Figure 7-25 and Figure 7-26, the average performance of the NDBGA is approximately 10% better than the PSO in terms of the minimum bandwidth between

المنارة للاستشارات

www.manaraa.com

user pairs, and about 3% better then PSO in terms of the total bandwidth. In addition to this, the average solution return time is about twice as long for the PSO.



(a)

(b)

Figure 7-26 The performance of the GA and the PSO algorithms on large scale static problems in terms of the solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances

المنارة للاستشارات
www.manaraa.com

Table 7-1 presents paired-t test results for NDBGA and PSO performances on static test problem instances.

Table 7-1 Paired-t tests for NDBGA and PSO on static scenarios

Problem Size		Mean Performance		Paired-t Test p-value
		NDBGA	PSO	
Small	% User Connectivity	56.267	50.600	0.003
	Min. Bandwidth (Mbps)	10.924	6.541	0.000
	Total Bandwidth (Mbps)	110.685	71.331	0.000
	Solution Time (sec)	11.699	6.226	0.000
Medium	% User Connectivity	96.178	84.756	0.000
	Min. Bandwidth (Mbps)	3.674	2.604	0.006
	Total Bandwidth (Mbps)	432.840	326.446	0.000
	Solution Time (sec)	51.401	66.383	0.006
Large	% User Connectivity	100.000	100.000	N/A
	Min. Bandwidth (Mbps)	5.156	4.533	0.192
	Total Bandwidth (Mbps)	2573.723	2490.506	0.289
	Solution Time (sec)	477.789	1141.468	0.000

Table 7-1 suggests that almost all practically significant differences in performances of NDBGA and PSO are statistically significant as well. The PSO suffers from long solution times for larger size problems with no improvement on the network performance measures when compared with the NDBGA.

المنارة للاستشارات
www.manaraa.com

7.4.2 Comparisons on Dynamic Scenarios

The performance analyses of the three optimizers on the dynamic problems are presented in this section. Similar to the static case, the results and the four performance comparisons are presented as boxplots of all the performance data from all problem instances of same type and size, followed by the graph of the best, average and the worst performances over replications for each problem instance. The analyses are grouped into three problem categories; small, medium and large, depending on the number of nodes.

The connectivity performance metrics for the dynamic problems are calculated at the end of each time step and reported as the average over the time span of the problem. The solution time for the dynamic problems reflect the time it takes to complete a full simulation over the entire time span.

7.4.2.1 Small Dynamic Problems

The comparisons on small scale problems are done on 20 dynamic problems with 4 users and 3 agents in a time span of 100 time steps. The problem instances are generated as explained in Section 7.2 and the test problem generation code given in appendices Ap- 2 and Ap- 3. For the NDBGA and PSO solutions, each problem is solved 5 times with different random number seeds.

An example problem and its solution is given in Figure 7-27. In the figure, the diamond shaped nodes represent user nodes and the round shaped nodes represent the agent nodes. The time caption shows the simulation time which runs from 1 to 100 for (a) at time *t* = 1, (b) at *t* = 25, (c) at *t* = 50 and (d) at *t* = 75.

المنارة للاستشارات
www.manaraa.com

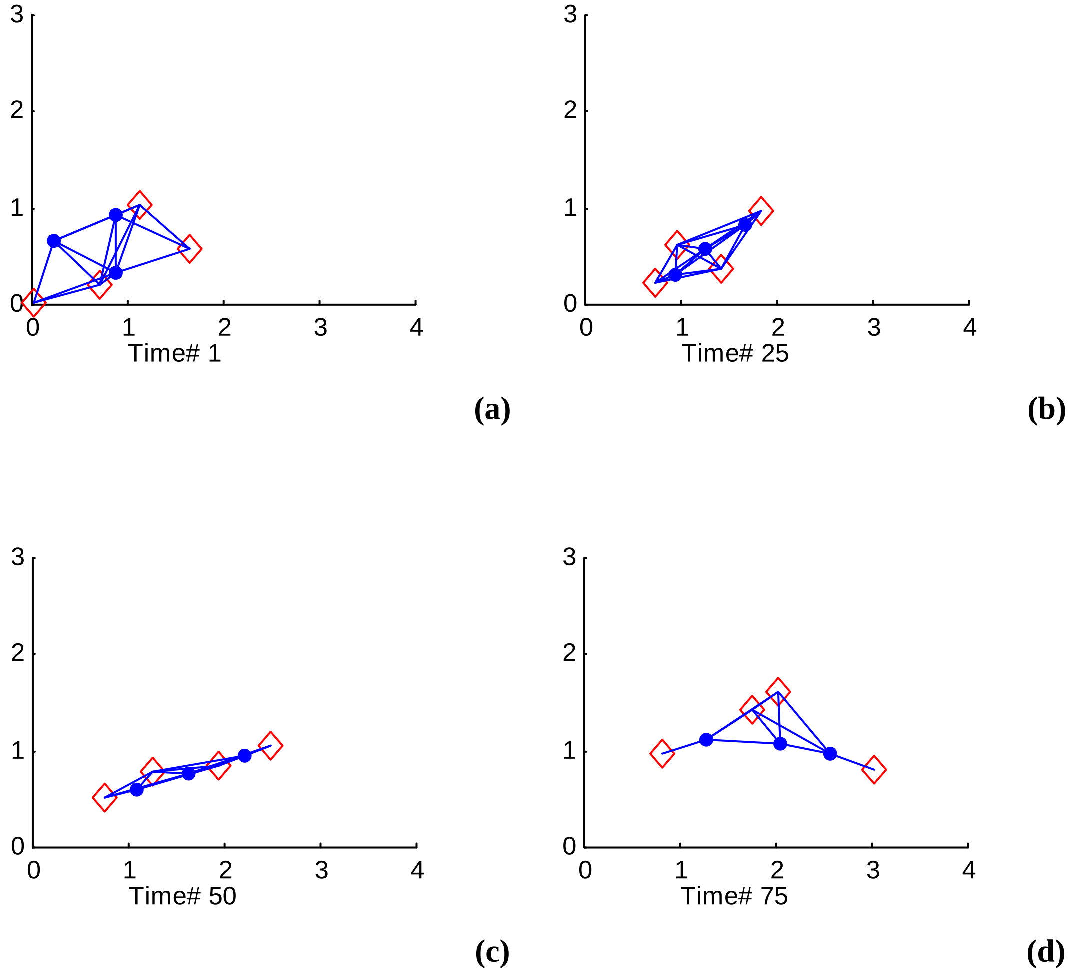
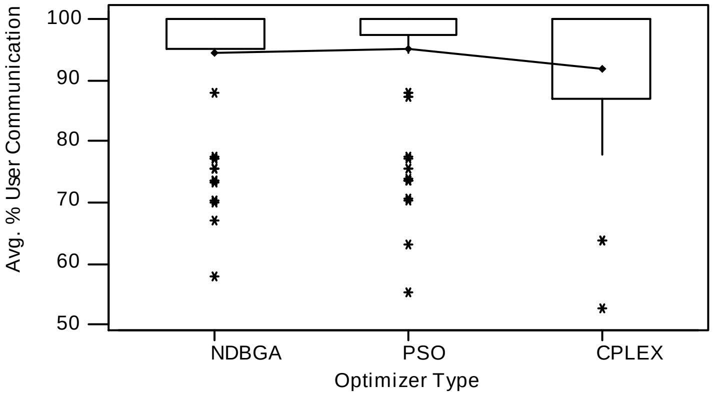


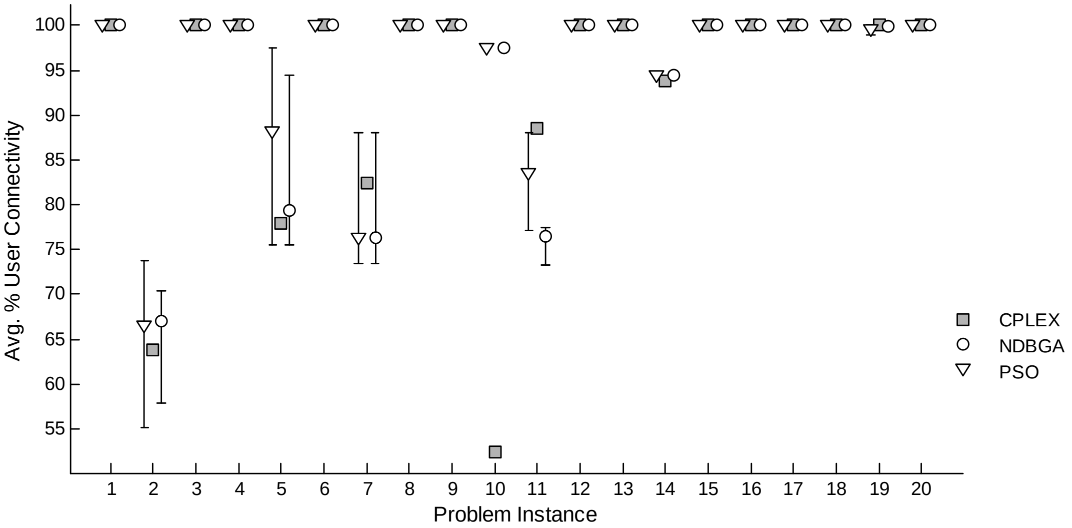
Figure 7-27 An example small scale dynamic scenario shown at (a) *t* = 1, (b) *t* = 25, (c) *t* = 50 and (d) *t* = 75, diamond shape represents user nodes and round shape represents agent nodes

When the average results are analyzed, as seen in Figure 7-28 the PSO based mobile agent location optimizer performed the best in terms of the average % user connectivity. PSO is followed by the GA with approximately a 1% gap and finally, the MIP model with approximately a 3.5% gap.

المنارة للاستشارات
www.manaraa.com



(a)



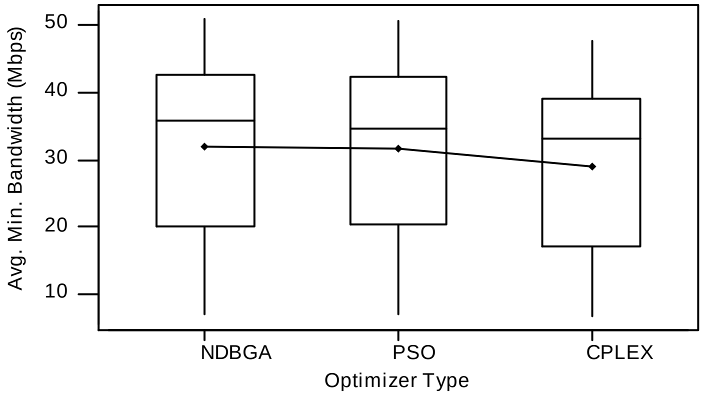
(b)

Figure 7-28 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on small scale dynamic test problems in terms of the average % user connectivity. (a) Boxplot (b) Best, average and worst performances over all problem instances

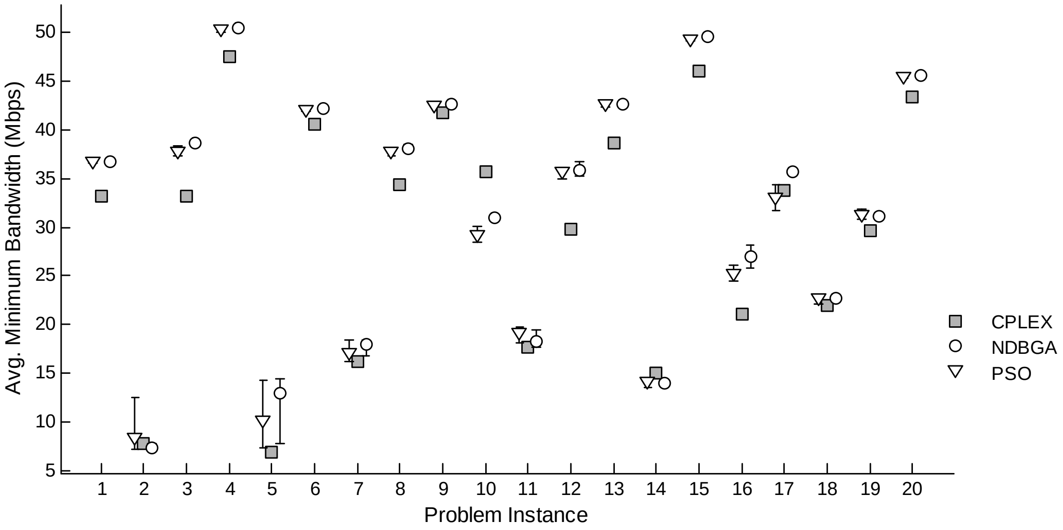
The GA based optimizer takes the lead in terms of the average minimum and total bandwidth among all MANET users. Figure 7-29 gives the performances of the

المنارة للاستشارات
www.manaraa.com

algorithms with respect to the average minimum bandwidth metric. The GA has the lead with a gap of approximately 1.5% over the PSO and CPLEX is behind the two heuristics again with an approximate performance gap of 9%.



(a)



(b)

Figure 7-29 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on small scale dynamic problems in terms of the average minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

المنارة للاستشارات
www.manaraa.com

The performance of the three algorithms with respect to the total bandwidth between MANET users is similar to the minimum bandwidth metric. Again, as seen in Figure 7-30, the GA leads with a 2.5% performance gap over the PSO, and around 7% over the CPLEX solver.

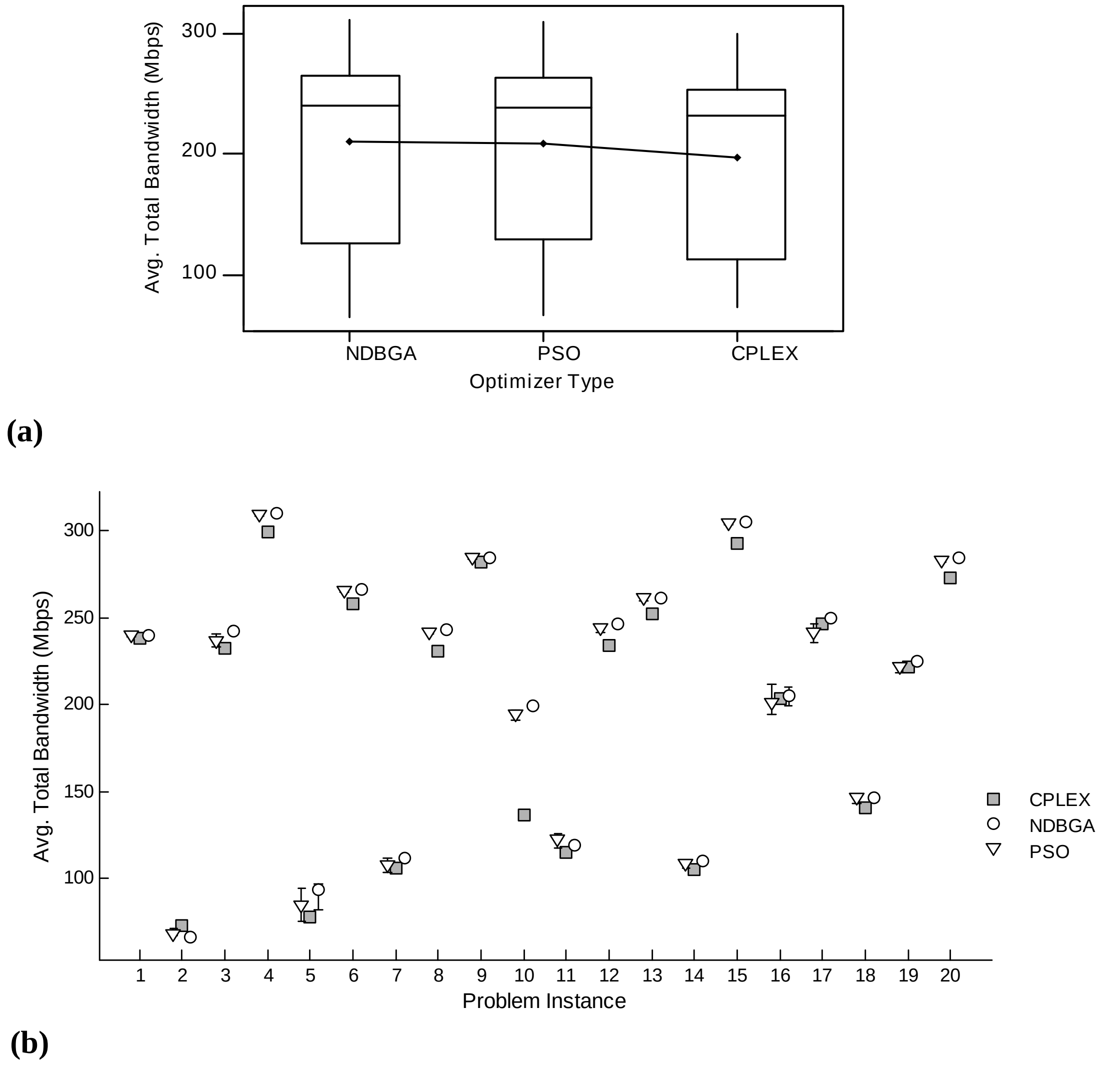
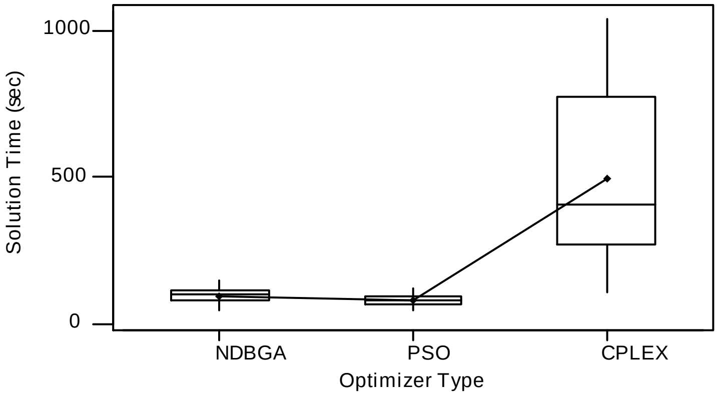
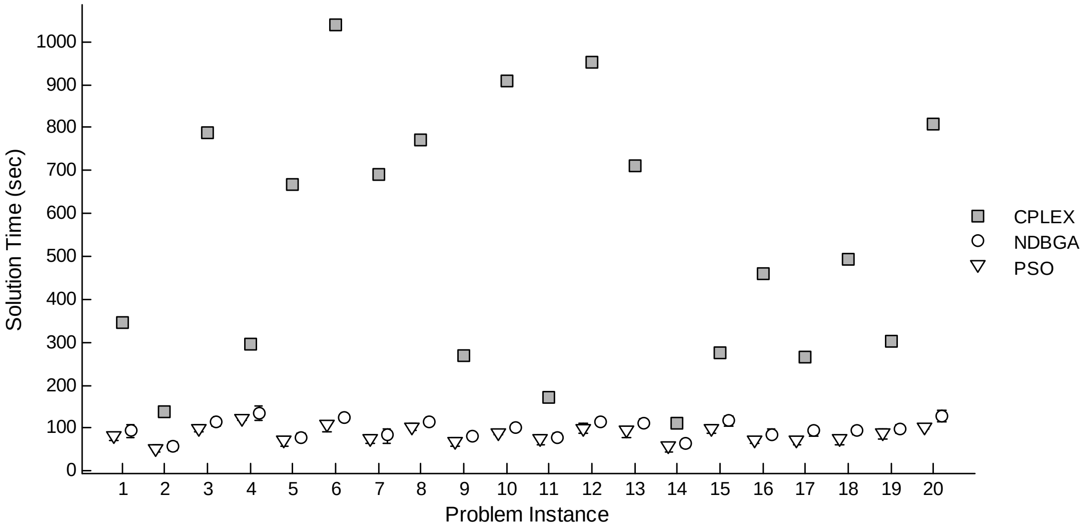


Figure 7-30 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on small scale dynamic problems in terms of the average total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

المنارة للاستشارات
www.manaraa.com



(a)



(b)

Figure 7-31 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on small scale dynamic problems in terms of the average solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances

www.manaraa.com

Finally, the solution time of the heuristics for the small scale problems is about 1/5th of the CPLEX model. As seen in Figure 7-31 with PSO being the fastest with a minor gap when compared to the GA.

All three algorithms managed to sustain an average user connectivity of at least 90%, heuristics being at the higher 90's.

7.4.2.1.1 Time Varying Performance:

In this part, the analyses of the algorithms' performance at each time step is presented for a typical small size problem. Figure 7-32 and Figure 7-33 show the change in % user connectivity, and change in minimum and total bandwidth with time, respectively. The dashed lines represent a MANET with no agents present and solid lines represent a MANET with 3 mobile agents and 4 users.

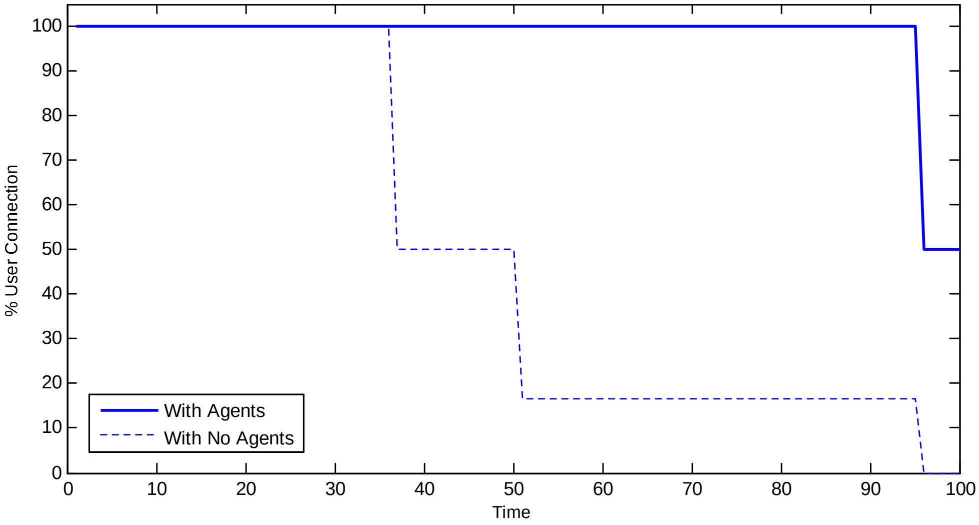
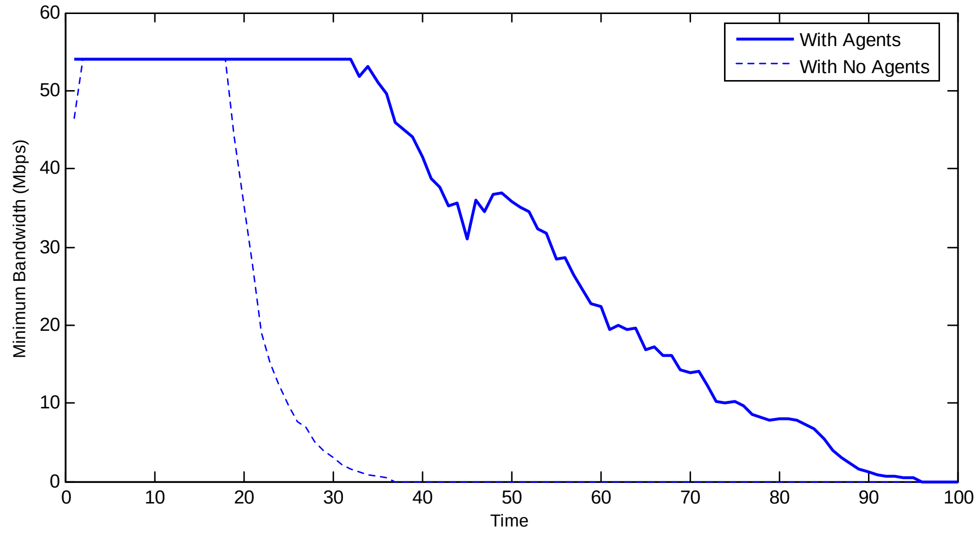
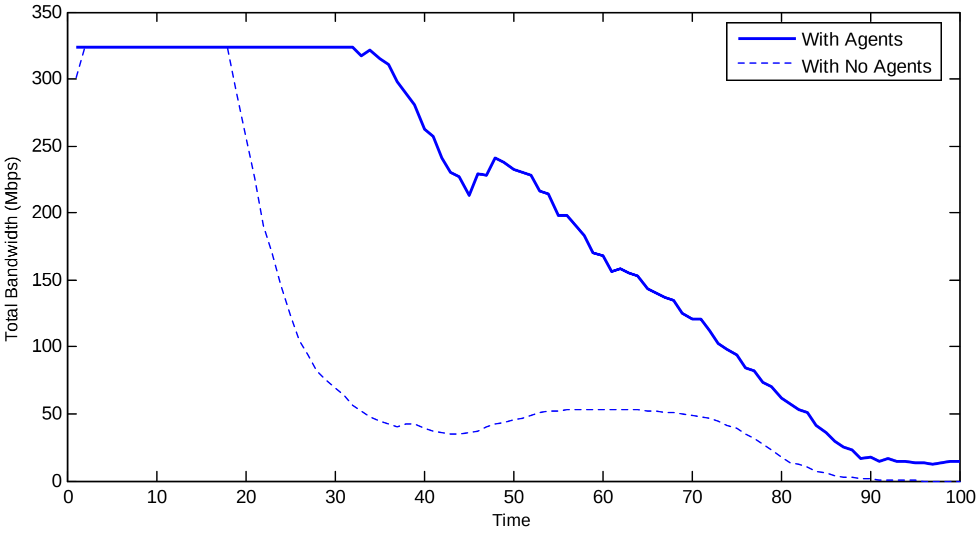


Figure 7-32 Change in % user connectivity over simulation time

المنارة للاستشارات
www.manaraa.com



(a)



(b)

Figure 7-33 Change of (a) minimum bandwidth and (b) total bandwidth with simulation time

In the above figures, the positive impact of having the mobile agents in the MANET can be clearly seen. The network without mobile agents starts to lose connectivity around

المنارة للاستشارات
www.manaraa.com

t=40, whereas full connectivity is maintained until *t*=95 with agents. The imporovement on the minimum bandwidth is also clearly visible before *t*=40.

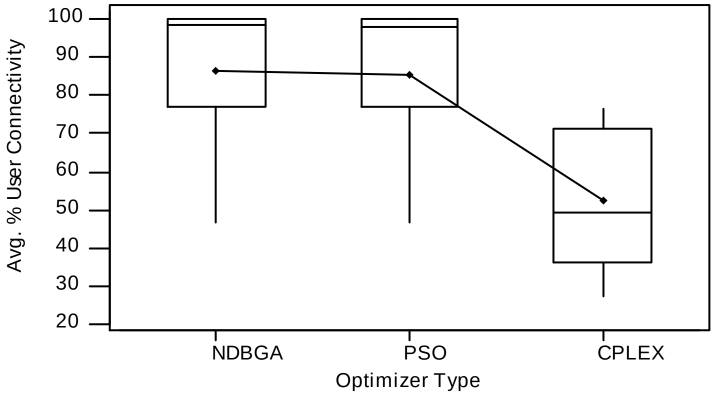
7.4.2.2 Medium Dynamic Problems

The comparisons on medium scale problems are done on 10 dynamic problems with 8 users and 6 agents in a time span of 100 time steps. For the NDBGA and PSO, each problem is solved 5 times with different random number seeds.

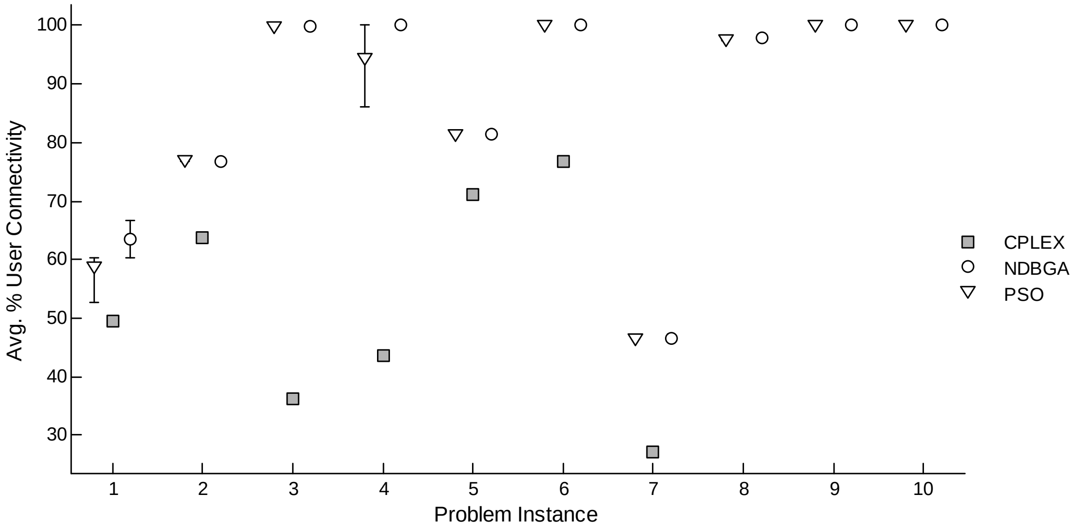
When the results are analyzed, unlike the small scale problems, for the medium scale the genetic algorithm performed superior to the PSO with respect to all three performance measures, and the solution time. However, as it can be observed from Figure 7-34, that the performance gap for the average % user connectivity measure is quite narrow.

المنارة للاستشارات

www.manaraa.com



(a)



(b)

Figure 7-34 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on medium scale dynamic problems in terms of the average % user connectivity. (a) Boxplot (b) Best, average and worst performances over all problem instances

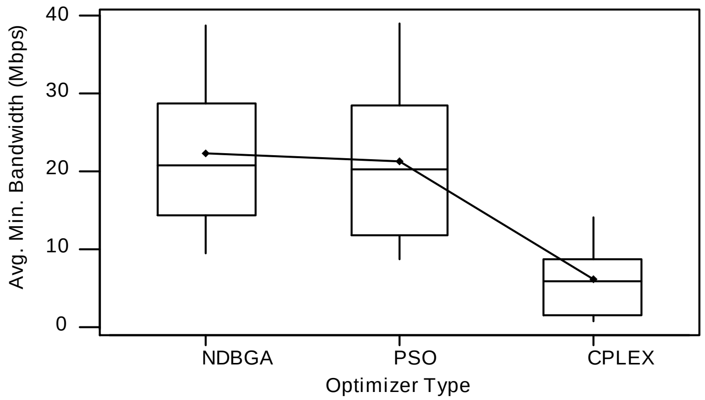
Figure 7-35 shows the performance of the algorithms on the average minimum bandwidth between all MANET user pairs. In terms of the minimum bandwidth, the GA

www.manaraa.com

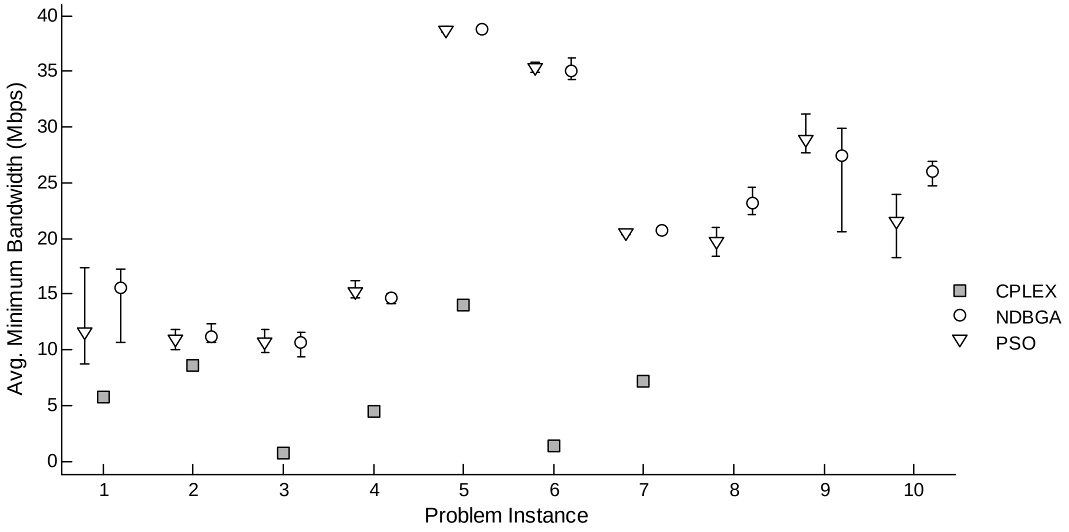
lead is followed by the PSO by a 10% gap, while the MIP model suffers from significant lack of performance.

With the medium scale problems, the MIP model became significantly overwhelmed and unstable. The CPLEX solver was unable to solve 3 out of 10 test problems, even with the time limitation removed. On the ones it could solve, it suffered from lack of performance due to the problem scale. Due to this issues, the large scale problems are only tested with the heuristic algorithms.

المنارة للاستشارات
www.manaraa.com



(a)

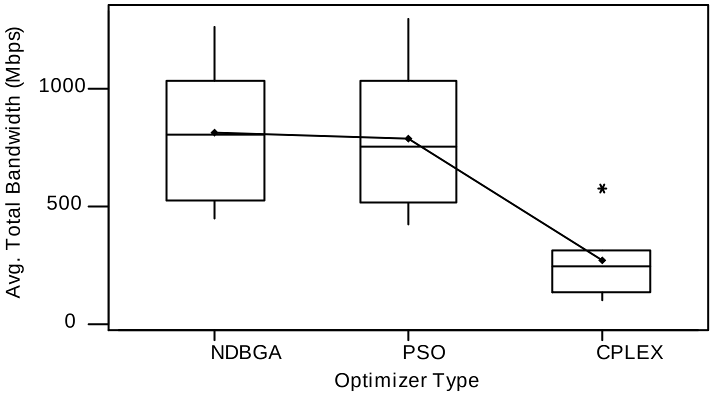


(b)

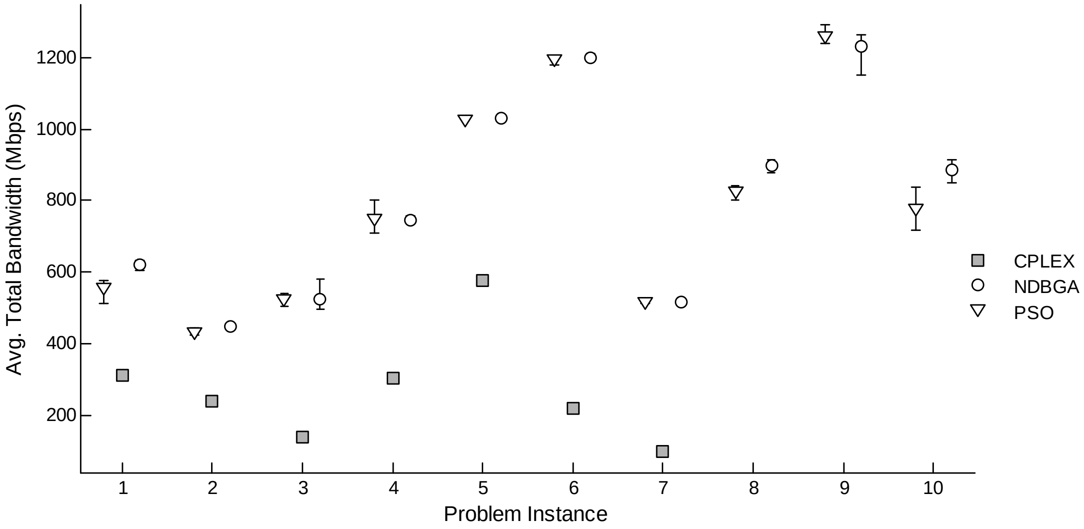
Figure 7-35 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on medium scale dynamic problems in terms of the average minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

www.manaraa.com

A performance similar to the minimum bandwidth measure is seen on the average total bandwidth. Again, as seen in Figure 7-36, the GA has a performance lead over the PSO by approximately 3%.



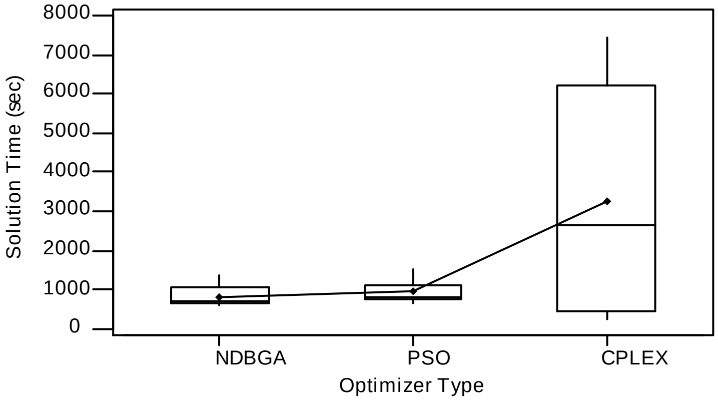
(a)



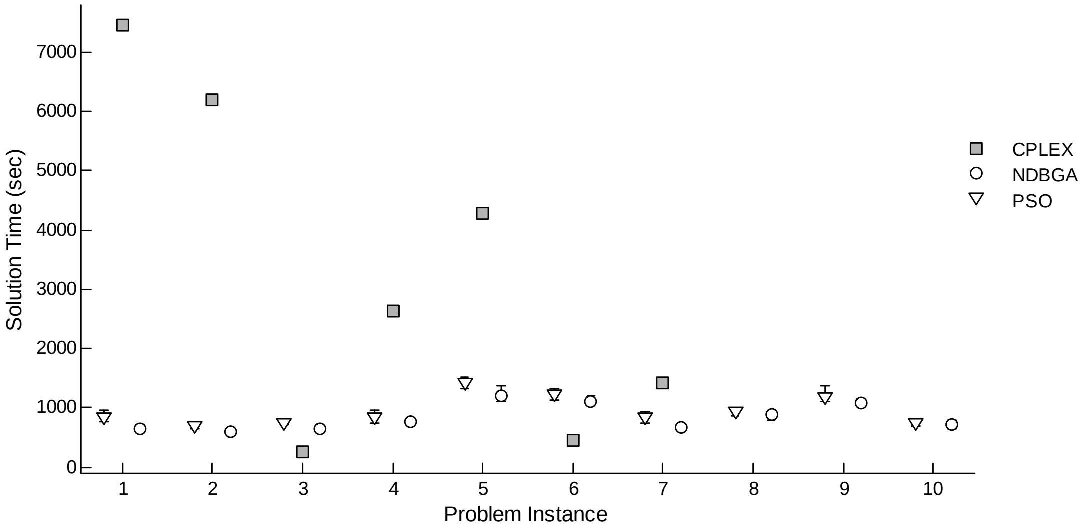
(b)

Figure 7-36 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on medium scale dynamic problems in terms of the average total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

www.manaraa.com



(a)



(b)

Figure 7-37 The performance of the GA, the PSO and the MIP (CPLEX) algorithms on medium scale dynamic problems in terms of the average solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances

www.manaraa.com

7.4.2.2.1 Time Varying Performance:

In this part, the analyses of algorithms' performance at each time step is presented for an example medium size problem. Figure 7-38 and Figure 7-39 show the change in % user connectivity, and change in minimum and total bandwidth with time, respectively. The dashed lines represent a MANET with no agents present and solid lines represent a MANET with 6 mobile agents and 8 users.

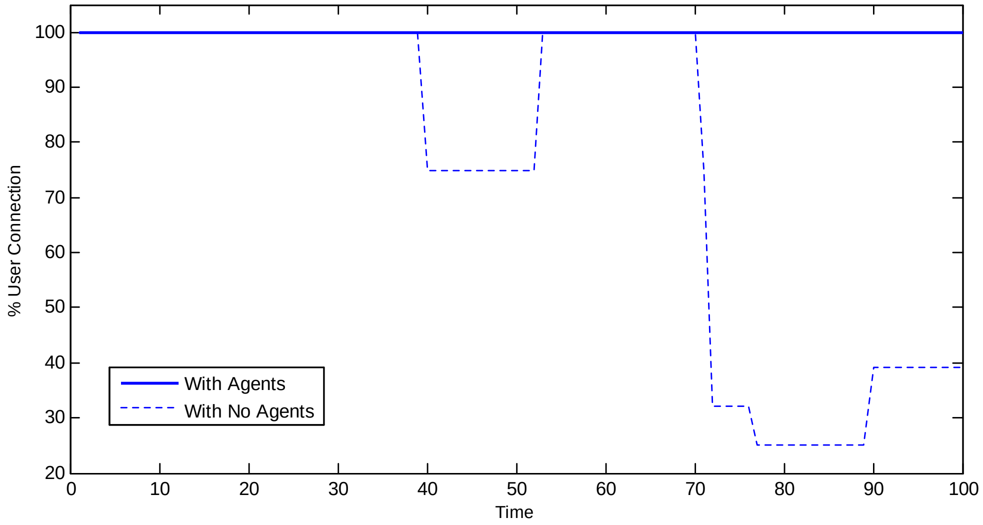
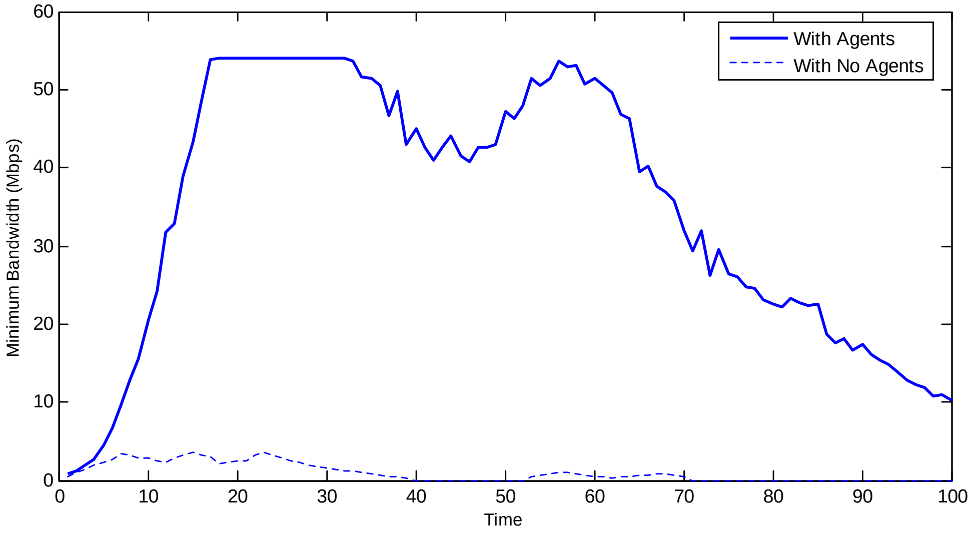


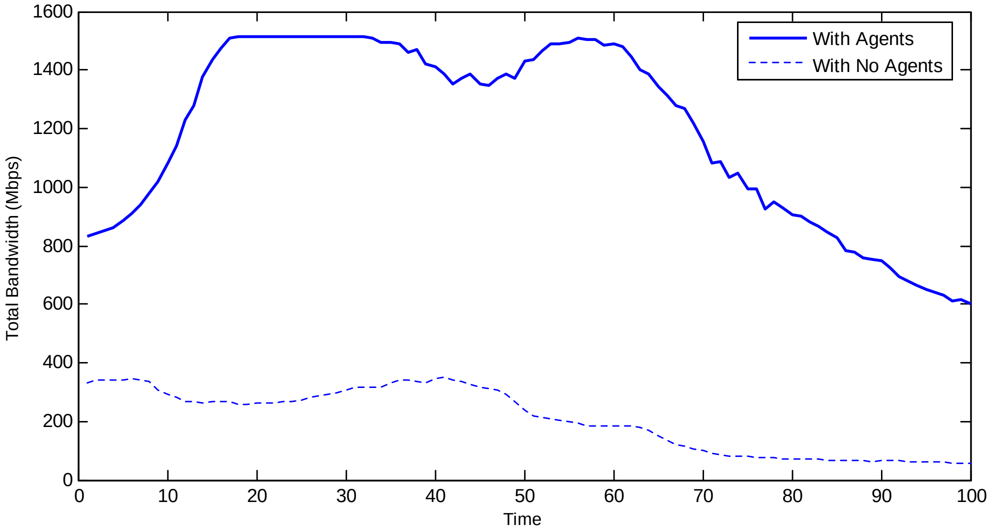
Figure 7-38 Change in % user connectivity over simulation time

The impact of mobile agents on the MANET performance in terms of connectivity is significant. The network with agents never loses connectivity. Moreover, the minimum and the total bandwidth performance is significantly improved as seen in Figure 7-39.

المنارة للاستشارات
www.manaraa.com



(a)



(b)

Figure 7-39 Change of (a) minimum bandwidth and (b) total bandwidth with simulation time

المنارة للاستشارات
www.manaraa.com

7.4.2.3 Large Dynamic Problems

The comparisons on large scale problems are done on 5 dynamic problems with 16 users and 12 agents over a time span of 100 time steps. For the NDBGA and PSO, each problem is solved 5 times with different random number seeds.

When the results are analyzed, it can be seen that the NDBGA algorithm performed slightly better in terms of average %user connectivity. The boxplot and the best, average and the worst performances for every problem instance can be seen in Figure 7-40 (a) and (b), respectively.

www.manaraa.com

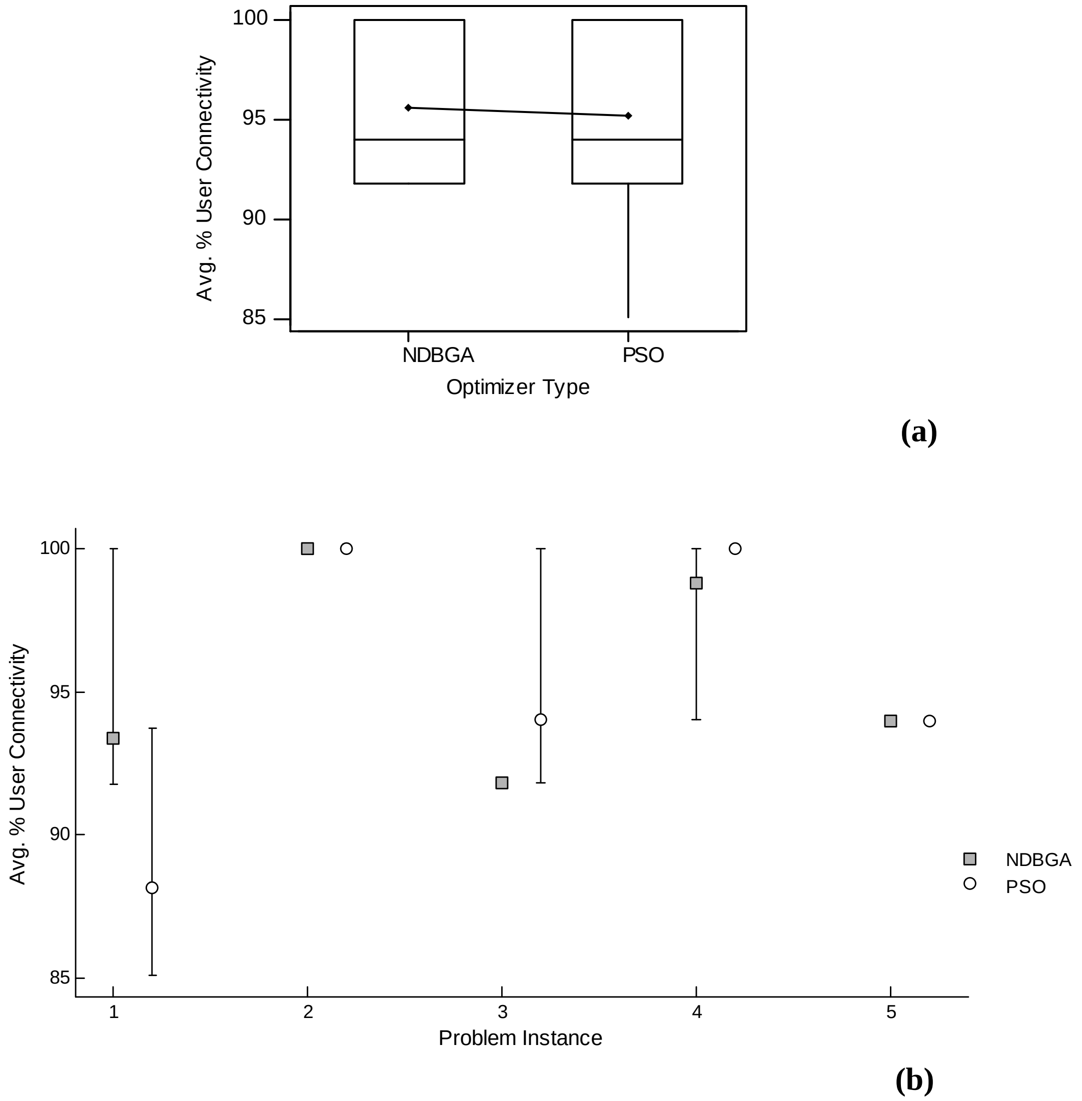


Figure 7-40 The performance of the GA and the PSO algorithms on large scale dynamic problems in terms of the average % user connectivity. (a) Boxplot (b) Best, average and worst performances over all problem instances

When the results for the average minimum bandwidth between user pairs are analyzed, the PSO shows a slightly better overall average performance, around 9%, with

المنارة للاستشارات
www.manaraa.com

larger variations within problem instances. The boxplot and the mean value graphs are given in Figure 7-41.

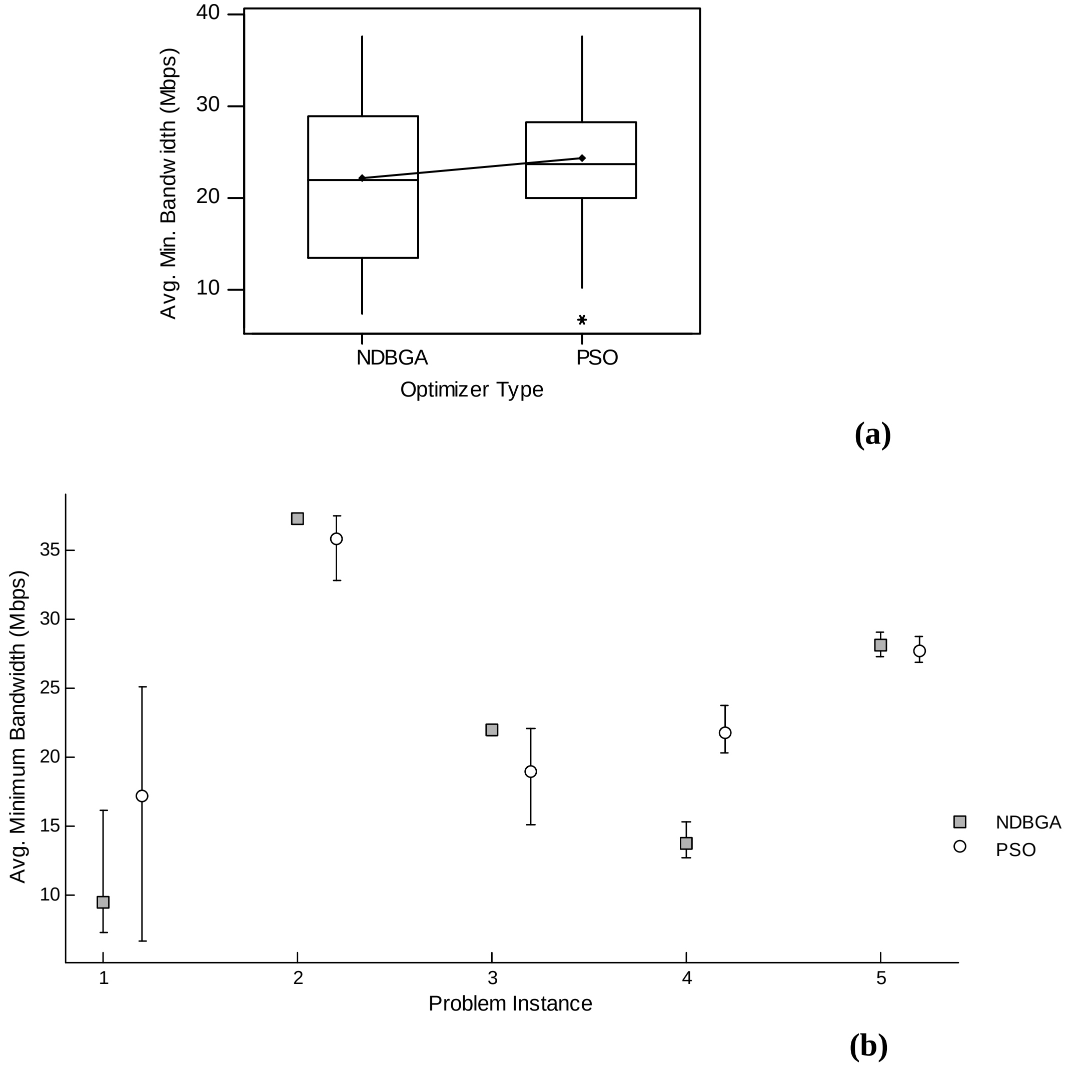


Figure 7-41 The performance of the GA and the PSO algorithms on large scale dynamic problems in terms of the average minimum (nonzero) bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

المنارة للاستشارات

www.manaraa.com

The NDBGA and the PSO algorithms perform almost the same in terms of the overall average total bandwidth between user pairs, with PSO having larger variance within problem instances. The boxplot and the mean value comparison graphs are given in Figure 7-43.

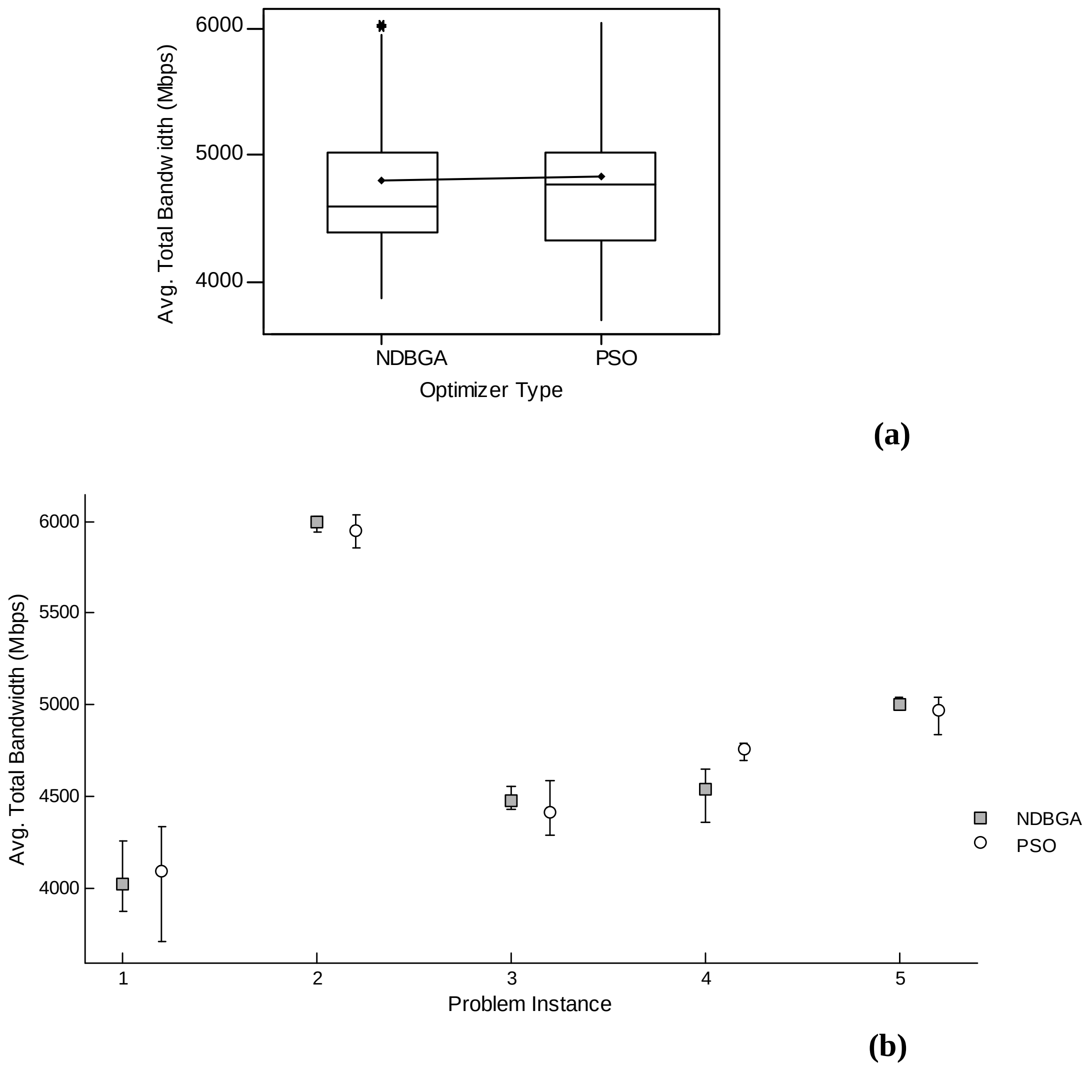


Figure 7-42 The performance of the GA and the PSO algorithms on large scale dynamic problems in terms of the average total bandwidth between all user pairs. (a) Boxplot (b) Best, average and worst performances over all problem instances

المنارة للاستشارات
www.manaraa.com

The average solution times for the NDBGA and the PSO for large scale dynamic problems are significantly different. Although the PSO’s performance for the average minimum and the total bandwidth measures is quite close to the NDBGA, the computation time is significantly higher. The PSO’s average performance is around 33% slower than the NDBGA. Furthermore, NDBGA’s performance for the shortest and the longest duration solutions are both superior to those of PSO’s. The boxplot and the mean value graphs for the solution time are given in Figure 7-43.

www.manaraa.com

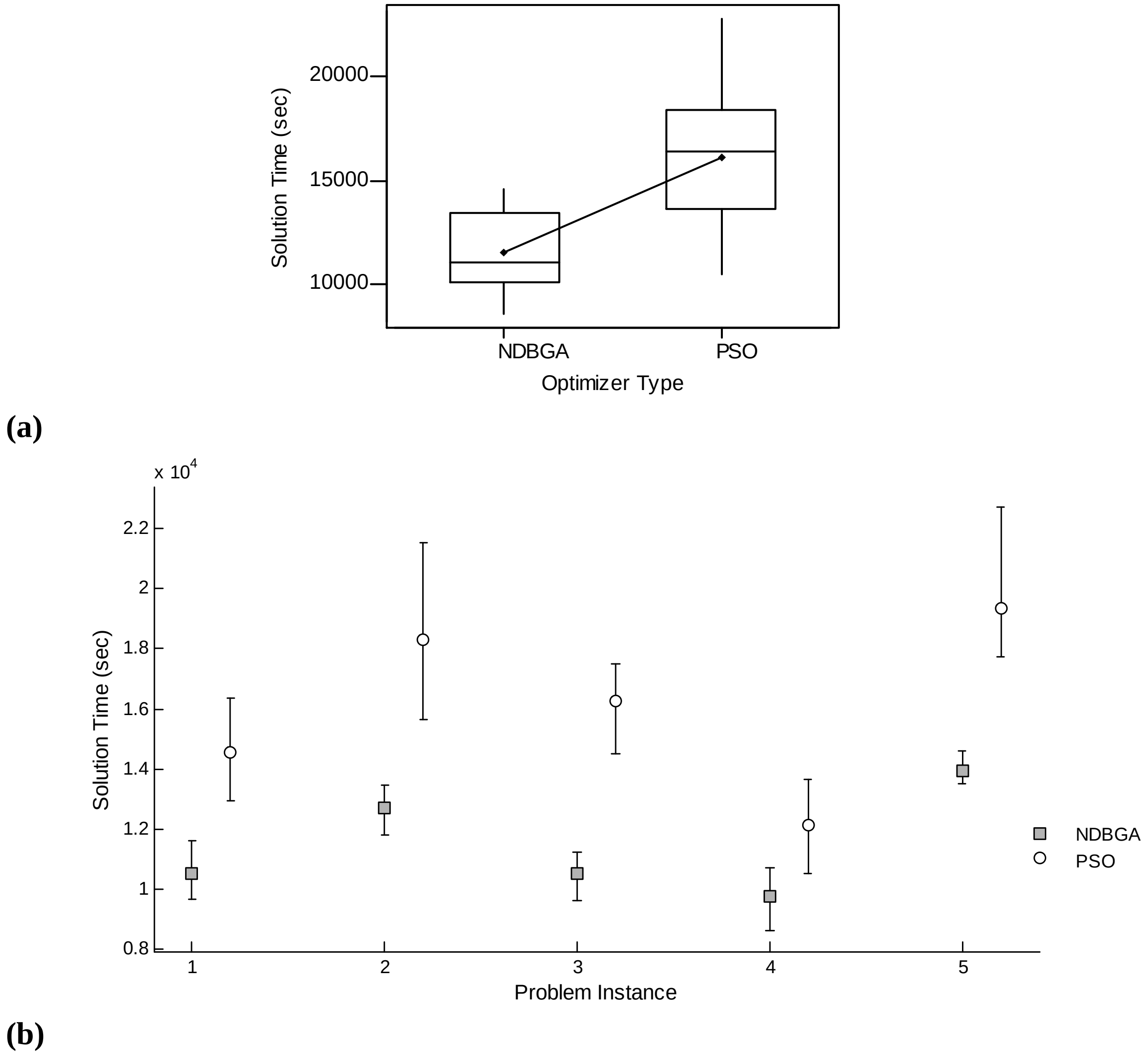


Figure 7-43 The performance of the GA and the PSO algorithms on large scale dynamic problems in terms of the solution time. (a) Boxplot (b) Best, average and worst performances over all problem instances

7.4.2.3.1 Time Varying Performance:

In this part, the analyses of the algorithms' performance at each time step is presented for an example large size problem. Figure 7-44 and Figure 7-45 show the

www.manaraa.com

change in % user connectivity, and change in minimum and total bandwidth with time, respectively. The dashed lines represent a MANET with no agents present and solid lines represent a MANET with 12 mobile agents and 16 users.

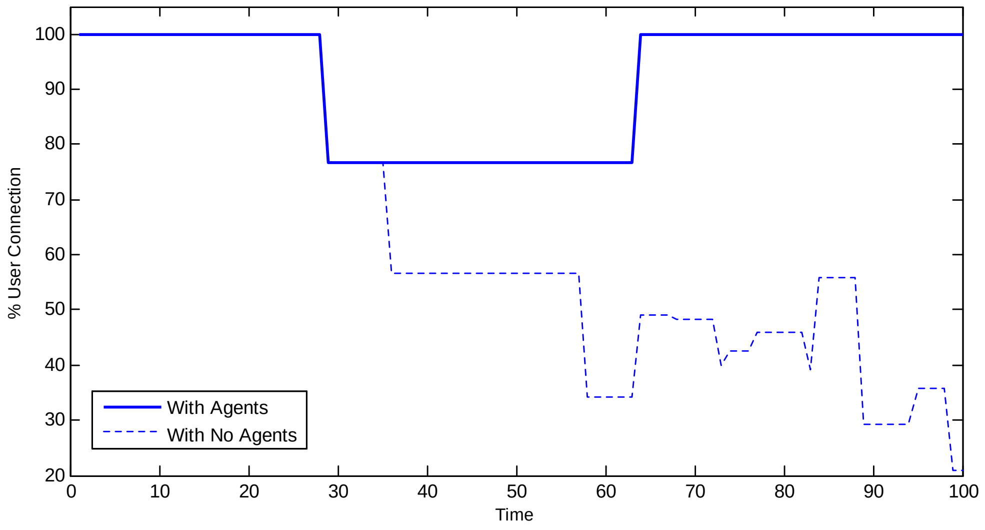
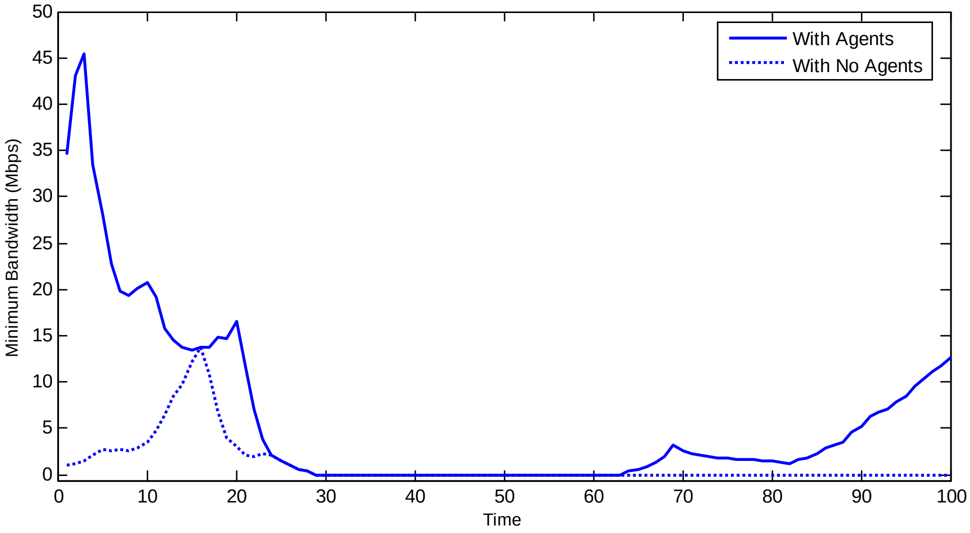


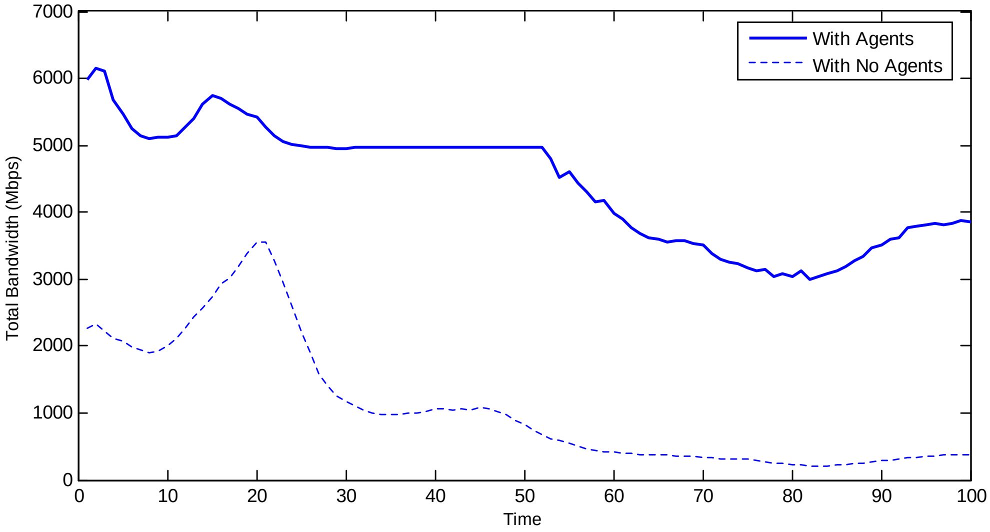
Figure 7-44 Change in % user connectivity over simulation time

The impact of mobile agents on the MANET performance in terms of connectivity is significant. The network with no agents loses full connectivity a little before *t* = 40, and constantly degrades after that while the network with mobile agents is able to regain and maintain connectivity. Moreover, the minimum and the total bandwidth performance is significantly improved as seen in Figure 7-45.

www.manaraa.com



(a)



(b)

Figure 7-45 Change of (a) minimum bandwidth and (b) total bandwidth with simulation time

Table 7-2 presents the paired-t test results for NDBGA and PSO performance on dynamic scenarios.

المنارة للاستشارات

www.manaraa.com

Table 7-2 Paired-t tests for NDBGA and PSO on dynamic scenarios

Problem Size		Mean Performance		Paired-t Test p-value
		NDBGA	PSO	
Small	Avg. % User Connectivity	94.558	95.318	0.082
	Avg. Min. Bandwidth (Mbps)	32.042	31.517	0.002
	Avg. Total Bandwidth (Mbps)	210.563	208.239	0.000
	Solution Time (sec)	98.428	84.049	0.000
Medium	Avg. % User Connectivity	86.586	85.561	0.013
	Avg. Min. Bandwidth (Mbps)	22.316	21.279	0.004
	Avg. Total Bandwidth (Mbps)	810.583	786.524	0.001
	Solution Time (sec)	832.144	944.363	0.000
Large	Avg. % User Connectivity	95.608	95.239	0.332
	Avg. Min. Bandwidth (Mbps)	22.147	24.323	0.040
	Avg. Total Bandwidth (Mbps)	4806.391	4835.774	0.169
	Solution Time (sec)	11492.028	16116.164	0.000

Table 7-2 suggests that the performances of NDBGA and the PSO for mobile agent location optimization are significantly different at the $\alpha=0.05$ level of significance for dynamic problems for almost all performance criteria. Although generally comparable results are achieved in terms of network performance, the PSO suffers from long solution times as the problem size increases.

المنارة للاستشارات
www.manaraa.com

7.5 Cost-benefit Analyses

When using mobile agents in real life, each agent will have an associated fixed cost and operating costs. In order to plan the required number of agents -or resources- prior to an operation, the proposed model can be used as a simulation tool to see the estimated network performance with different numbers of agents incorporated into the network.

The following example is a demonstration of such a simulation. A 20 user problem is simulated with the number of mobile agents ranging from 0 to 10. The number of agents versus the average network performance over 100 time steps is provided in the figures, followed by discussion.

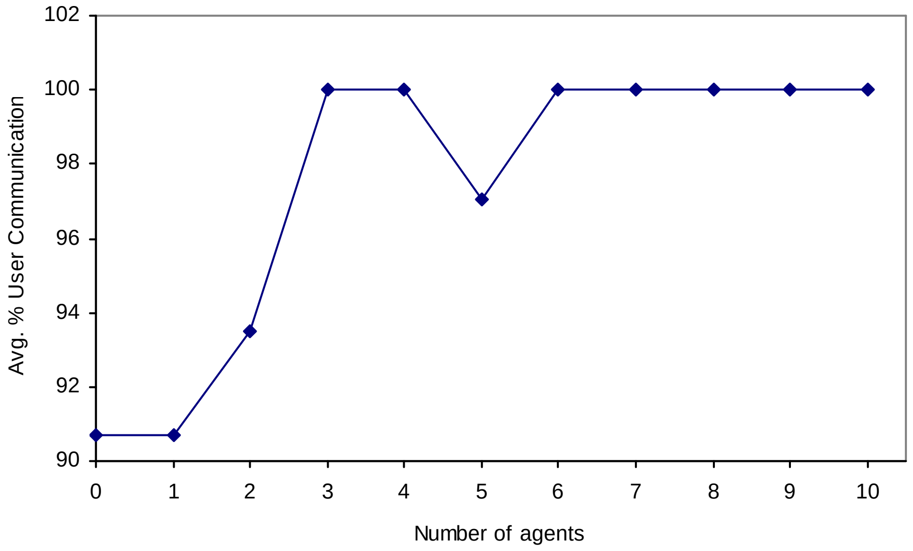


Figure 7-46 Number of mobile agents versus average % user connectivity

www.manaraa.com

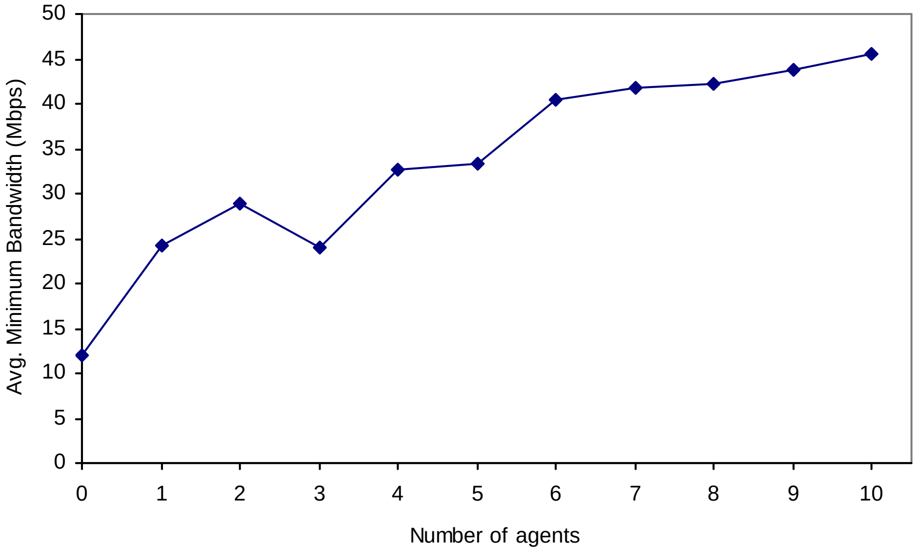


Figure 7-47 Number of mobile agents versus average minimum nonzero bandwidth

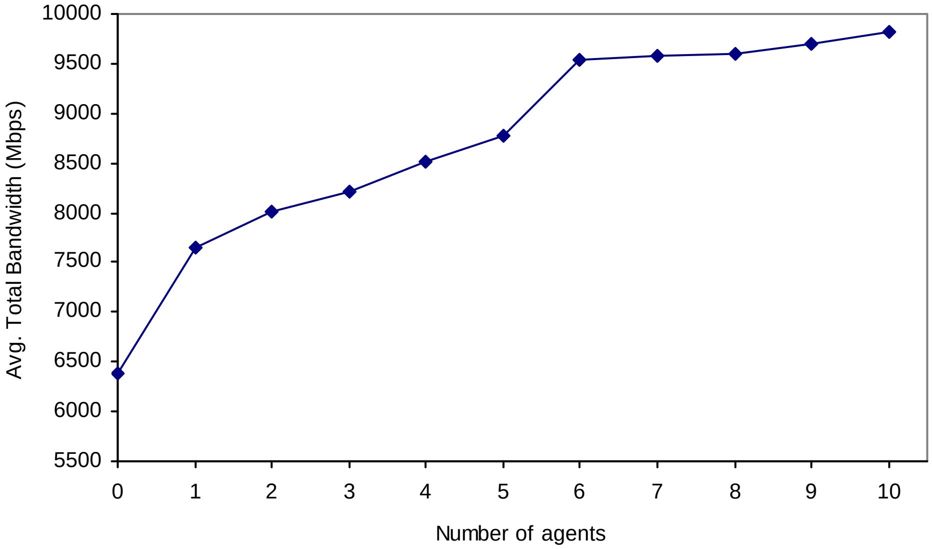


Figure 7-48 Number of mobile agents versus average total bandwidth

www.manaraa.com

In Figure 7-46, the change in average % user connectivity with increasing agent number is presented. Although full connectivity is achieved with only 3 agents, it is also true that full connection was lost with 5 agents. This is because the agents blend in and evolve with the network. The evolution with 3 agents can be different than with 5 agents because the best agent locations at each time step will change for each agent with changing number of agents. The graph suggests that number of agents should be on the greater side, preferably at least 6 in this case. As a general methodology, this also suggests that test simulations should be done with more agents than what is thought necessary to see whether a steady connectivity and performance is achieved, or not.

The following graphs present the time varying performances with 0, 3, 6 and 10 agents. In Figure 7-49, cases with 3, 6 and 10 agents are 100% connected all the time.

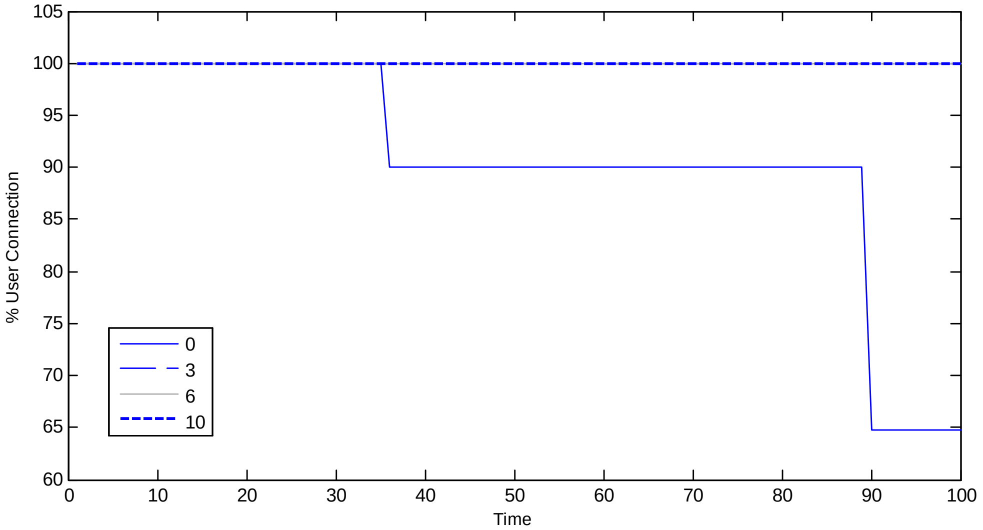


Figure 7-49 Number of agents versus the change in % user connectivity over simulation time

www.manaraa.com

In Figure 7-50, the effects on the minimum and the total bandwidth is seen. Although full connectivity is achieved, for the case with 3 agents, the bandwidth properties are much better with 6 or 10 agents. A greater benefit is experienced when the agent number is increased from 3 to 6 than 6 to 10. The scenario with 10 agents can maintain maximum possible network performance until *t*=70 and performs better than others towards the end of the simulation (*t*>70) when the network is the least dense, which is expected.

المنارة للاستشارات

www.manaraa.com

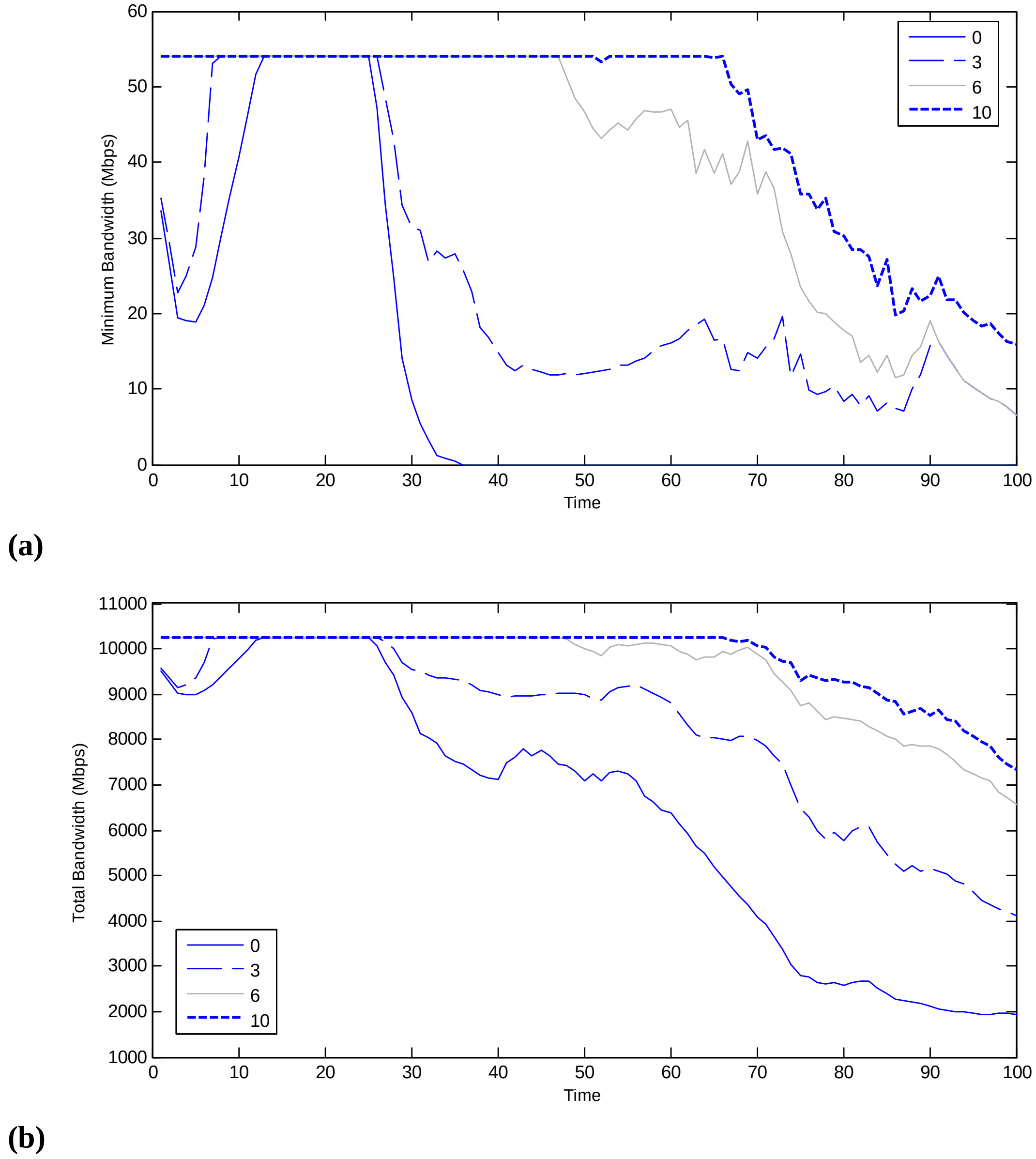


Figure 7-50 Number of agents versus the change of (a) minimum bandwidth and (b) total bandwidth with simulation time

www.manaraa.com

CHAPTER 8

CONCLUSIONS

In this research, a new model is proposed to conceptualize an autonomous topology optimization for mobile ad hoc networks. Mobile ad hoc networks are advantageous in many aspects. They do not require a costly infrastructure, and they are flexible and immediately available to serve the tasks and needs of the users. However, there are topological challenges that affect connectivity and performance due to their mobile nature. The proposed approach relocates a number of mobile agents within their mobility capabilities to help maintain a suitable level of communication service in the network.

The representation of the wireless ad hoc network communications as network flows and optimization using a maximum flow model is a novel approach. It is very responsive to small changes in topology when evaluating network connectivity and performance. Also, it can be used with any signal attenuation model when calculating the data flow rates.

The dynamic nature of the problem is a challenge, but it also enables the optimizer to gain additional information by leveraging the dynamism. The optimization at a new time step can benefit from the knowledge of the best solution from the previous time step. This results from the fact that motion has a time and space continuity and for small increments in time, the velocity of objects are usually correlated with predecessor and

المنارة للاستشارات
www.manaraa.com

successor velocities. Another benefit based on the dynamism is the ability to predict future user locations. By making use of the position data from a few time steps back, the optimizer can predict where the user nodes will be positioned over a specified prediction horizon and thus position the agents for better performance. The inclusion of this additional information is algorithm independent. The genetic algorithm, the particle swarm algorithm or any other algorithm that is programmed to solve the proposed model can benefit from the additional information.

The non-deterministic decoding for binary coded genetic algorithms was developed during this research but is applicable to a wide range of continuous optimization problems. It outperformed previous approaches to the resolution deficiency that is experienced when solving problems in continuous domain with binary encoded genetic algorithms. The non-deterministic decoding method enables the genetic algorithm to effectively work its crossover and mutation mechanisms without the need to increase the chromosome length for precision only.

The approximate MIP model proposed in this research is also new. It optimizes the locations of agent nodes in a network with an objective to maximize a function of the all-pair maximum flow and total maximum flows between node pairs. The movements of agent nodes affect the link capacities, which is incorporated into the model, as well as the agent travel distance constraints. The nonlinear link distance versus capacity relationship and the Euclidean link distance and agent travel distances are modeled using piecewise linear approximations. The model, while being not as effective as the heuristic optimizers tested, shows how complex the problems are, even very small sizes.

المنارة للاستشارات
www.manaraa.com

The heuristic algorithms have outperformed the MIP model, especially with respect to the solution time. They outperformed the MIP model with respect to the performance criteria most of the time because the heuristics allow greater flexibility when defining the objective function when the network is not fully connected. Among the heuristics, both the genetic algorithm and the particle swarm optimizations' performances were close in terms of solution quality, but the genetic algorithm in general, performed better in terms of the solution time.

The proposed approach, while developed for dynamic topology optimization, easily adapts to a static scenario by increasing the agent velocity constraints. The static scenario is useful when users want to improve an existing system of sensors or communication hubs already positioned in the field, or when designing a new static system.

The approach could also be used for "what if" purposes before launching an actual network in the field. The simulation is useful to plan for the most efficient number of mobile agents to serve under a certain scenario, and to consider cost / benefit trade offs.

For future research, a combination of the static model and the dynamic model might be adapted into a system which will deploy mobile agents into an already operating MANET. Another future topic is a slight modification to the objective function. In this study, all users are considered to be of equal importance. This could be readily changed, and weighted user importance objectives could be investigated. Different user mobility models could be used with the proposed approach. Finally, the model could be extended to three dimensions for agent and user motion and communication.

المنارة للاستشارات
www.manaraa.com

REFERENCES

[1] Abolhasan, M., Wysocki, T., and Dutkiewicz, E., "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks* 2(1) (2004) 1-22.

[2] Ahuja, R.K., Magnanti, T.L., and Orlin, J.B., Network flows, theory, algorithms, and applications, Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1993.

[3] Altiparmak, F., Dengiz, B., and Smith, A.E. "Reliability optimization of computer communication networks using genetic algorithms," in: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 1998, 4676-4681.

[4] Altiparmak, F., Dengiz, B., and Smith, A.E. "Reliability estimation of computer communication networks: Ann models," in: *Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC 2003)*, 2003, 1353-1358.

[5] Ammari, H. and El-Rewini, H. "A location information-based route discovery protocol for mobile ad hoc networks," in: *Proceedings of the IEEE International Conference on Performance, Computing, and Communications*, 2004, 625-630.

[6] Arikati, S.R., Chaudhuri, S., and Zaroliagis, C.D., "All-pairs min-cut in sparse networks," *Journal of Algorithms* 29(1) (1998) 82-110.

[7] Aschenbruck, N., Frank, M., Martini, P., and Tolle, J. "Human mobility in manet disaster area simulation - a realistic approach," in: *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, 2004, 668-675.

[8] Ashbrook, D. and Starner, T. "Learning significant locations and predicting user movement with gps," in: *Proceedings of the 6th International Symposium on Wearable Computers (ISWC 2002)*, 2002, 101-108.

[9] Bettstetter, C. "On the minimum node degree and connectivity of a wireless multihop network," in: *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC'02)*, Lausanne, Switzerland, ACM Press, 2002, 80-91.

المنارة للاستشارات

www.manaraa.com

[10] Bilurkar, P., Rao, N., Krishna, G., and Jain, R. "Application of neural network techniques for location predication in mobile networking," in: *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02),* 2002, 2157-2161.

[11] Boleng, J. and Camp, T. "Adaptive location aided mobile ad hoc network routing," in: *Proceedings of the IEEE International Conference on Performance, Computing, and Communications*, 2004, 423-432.

[12] BOOST.org, Boost c++ libraries, 2006, Accessed in 2006; Available at: http://www.boost.org.

[13] Branke, J., Kaußler, T., Schmidt, S., and Schmeck, H., A multi-population approach to dynamic optimization problems, in: I.C. Parmee (Ed.), Adaptive computing in design and manufacture, (ACDM 2000), Springer, 2000, 299-308.

[14] Branke, J. "Evolutionary approaches to dynamic optimization problems - updated survey," in: *Proceedings of the GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, 2001, 27-30.

[15] Branke, J. "Evolutionary approaches to dynamic optimization problems - introduction and recent trends -," in: *Proceedings of the Workshop on Evolutionary Algorithms for Dynamic Optimization Problems (EvoDOP-2003) held in conjunction with the Genetic and Evolutionary Computation Conference (GECCO-2003),* Chicago, IL, USA, 2003, 2-4.

[16] Branke, J., Salihoğlu, E., and Şima, U. "Towards an analysis of dynamic environments," in: *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, Washington DC, USA, ACM Press, 2005, 1433-1440.

[17] Bratton, D. and Kennedy, J. "Defining a standard for particle swarm optimization," in: *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007),* 2007, 120-127.

[18] Camp, T., Boleng, J., and Davies, V., "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications* 2(5) (2002) 483-502.

[19] Carlisle, A. and Dozier, G. "An off-the-shelf pso," in: *Proceedings of the 2001 Workshop on Particle Swarm Optimization*, Indianapolis, IN, 2001, 1-6.

www.manaraa.com

[20] Carlisle, A. and Dozier, G. "Tracking changing extrema with adaptive particle swarm optimizer," in: *Proceedings of the 5th Biannual World Automation Congress*, 2002, 265-270.

[21] Caruana, R.A. and Schaffer, J.D. "Representation and hidden bias: Gray vs. Binary coding for genetic algorithms," in: *Proceedings of the 5th International Conference on Machine Learning*, Los Altos, CA, Morgan Kaufmann, 1988, 153-161.

[22] Cheriyan, J. and Maheshwari, S.N., "Analysis of preflow push algorithms for maximum network flow," *SIAM Journal on Computing* 18(6) (1989) 1057-1086.

[23] Chiang, C.-C. "Routing in clustered multihop, mobile wireless networks with fading channel," in: *Proceedings of the IEEE Singapore International Conference on Networks*, 1997, 197-211.

[24] Chlamtac, I., Conti, M., and Liu, J.J.N., "Mobile ad hoc networking: Imperatives and challenges," *Ad Hoc Networks* 1(1) (2003) 13-64.

[25] Chu, T. and Nikolaidis, I., "Node density and connectivity properties of the random waypoint model," *Computer Communications* 27(10) (2004) 914-922.

[26] Clerc, M. "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," in: *Proceedings of the 1999 International Congress on Evolutionary Computation, (ICEC 1999)*, Washington, DC, 1999, 1951-1957.

[27] Cobb, H.G., An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time dependent nonstationary elements, Technical Report AIC-90-001, Naval Research Laboratory, Washington, DC., 1990.

[28] Cook, J.L. and Ramirez-Marquez, J.E., "Two-terminal reliability analyses for a mobile ad hoc wireless network," *Reliability Engineering & System Safety* 92(6) (2007) 821.

[29] Cooper, C., "Misinformation slowed federal response to Katrina," The Wall Street Journal, September 30, 2005, page A4.

[30] Creixell, W. and Sezaki, K. "Mobility prediction algorithm for mobile ad hoc network using pedestrian trajectory data," in: *Proceedings of the IEEE Region 10 Conference, (TENCON 2004)*, 2004, 668-671.

[31] De Jong, K.A., An analysis of the behavior of a class of genetic adaptive systems, PhD Thesis, University of Michigan, 1975.

المنارة للاستشارات

www.manaraa.com

[32] Dengiz, B., Altiparmak, F., and Smith, A.E., "Local search genetic algorithm for optimal design of reliable networks," *IEEE Transactions on Evolutionary Computation* 1(3) (1997) 179-188.

[33] Dengiz, B., Altiparmak, F., and Smith, A.E., "Efficient optimization of all-terminal reliable networks, using an evolutionary approach," *IEEE Transactions on Reliability* 46(1) (1997) 18-26.

[34] Dixon, L.C. and Szego, G.P., Towards global optimization 2, North-Holland Inc., Amsterdam, 1978.

[35] Dow, C.R., Lin, P.J., Chen, S.C., Lin, J.H., and Hwang, S.F. "A study of recent research trends and experimental guidelines in mobile ad-hoc network," in: *Proceedings of the 19th Intrernational Conference on Advanced Information Networking and Applications (AINA'05)*, 2005, 72-77.

[36] Eberhart, R.C. and Shi, Y. "Comparing inertia weights and constriction factors in particle swarm optimization," in: *Proceedings of the 2000 Congress on Evolutionary Computing*, 2000, 84-88.

[37] Eberhart, R.C. and Yuhui, S. "Tracking and optimizing dynamic systems with particle swarms," in: *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001, 94.

[38] Erdös, P. and Rényi, A., "On the evolution of random graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5 (1960) 17-61.

[39] Fogel, D.B., System identification through simulated evolution: A machine learning approach to modeling, Ginn Press, Needham Heights, MA, 1991.

[40] Ford, L.R. and Fulkerson, D.R., "Maximal flow through a network," *Canadian Journal of Mathematics* 8 (1956) 399-404.

[41] Forodigh, M., Johansson, P., and Larsson, P., "Wireless ad hoc networking-the art of networking without a network," *Ericsson Review* 4 (2000) 248-263.

[42] Goldberg, A.V., A new max-flow algorithm, Technical Report MIT/LCS/TM-291, Laboratory for Computer Science, MIT, Cambridge, MA., 1985.

[43] Goldberg, A.V. and Tarjan, R.E., "A new approach to the maximum-flow problem," *Journal of the ACM (JACM)* 35(4) (1988) 921-940.

المنارة للاستشارات

www.manaraa.com

[44] Goldberg, D.E., Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, Reading, MA, 1989.

[45] Grefenstette, J.J. "Genetic algorithms for changing environments," in: *Proceedings of Parallel Problem Solving From Nature (PPSN-2)*, Brussels, 1992, 137-144.

[46] Grossglauser, M. and Tse, D.N.C., "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking* 10(4) (2002) 477-486.

[47] GSM-Association, 2005, Accessed in 2005; Available at: www.gsmworld.com.

[48] Gupta, P. and Kumar, P.R., "The capacity of wireless networks," *IEEE Transactions on Information Theory* 46 (2000) 388–404.

[49] Hartvigsen, D. and Mardon, R., "The all-pairs min cut problem and the minimum cycle basis problem on planar graphs," *SIAM Journal on Discrete Mathematics* 7(3) (1994) 403-418.

[50] Hartvigsen, D., "Generalizing the all-pairs min cut problem," *Discrete Mathematics* 147(1-3) (1995) 151-169.

[51] Holland, J.H., Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, MI, 1975.

[52] Huang, R. and Zaruba, G.V., "Location tracking in mobile ad hoc networks using particle filters," *Journal of Discrete Algorithms* 5(3) (2007) 455-470.

[53] Ishibashi, B. and Boutaba, R., "Topology and mobility considerations in mobile ad hoc networks," *Ad Hoc Networks* 3(6) (2005) 762-776.

[54] Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., and Viennot, L. "Optimized link state routing protocol for ad hoc networks," in: *Proceedings of the IEEE International Multi Topic Conference (INMIC 2001), Technology for the 21st Century*, 2001, 62-68.

[55] Johnson, D.B. and Maltz, D.A., Dynamic source routing in ad-hoc wireless networks, in: H.K. T. Imielinski (Ed.), Mobile computing, Kluwer, Boston, 1996, 153-181.

[56] Kennedy, J., Eberhart, R., and Shi, Y., Swarm intelligence, Morgan Kaufmann, San Mateo, CA, 2001.

المنارة للاستشارات

www.manaraa.com

[57] Konak, A. and Smith, A.E. "A hybrid genetic algorithm approach for backbone design of communication networks," in: *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, 1999, 1817-1823.

[58] Kwak, B.J., Song, N.O., and Miller, L.E., "A standard measure of mobility for evaluating mobile ad-hoc network performance," *IEICE Transactions on Communications* E86-B (2003) 3236-3243.

[59] Kwon, Y.D., Kwon, S.B., and Kim, J.Y., "Convergence enhanced genetic algorithm with successive zooming method for solving continuous optimization problems," *Computers and Structures* 81 (2003) 1715-1725.

[60] Lee, S.-J., Hsu, J., Hayashida, R., Gerla, M., and Bagrodia, R., "Selecting a routing strategy for your ad hoc network," *Computer Communications* 26(7) (2003) 723-733.

[61] Liu, G. and Maguire Jr., G., "A class of mobile prediction algorithms for wireless mobile computing and communications," *Mobile Networks and Applications* 1 (1996) 113-121.

[62] Liu, G.Y. and Maguire, G.Q., Jr. "A predictive mobility management algorithm for wireless mobile computing and communications," in: *Proceedings of the Fourth IEEE International Conference on Universal Personal Communications*, Tokyo, Japan, 1995, 268-272.

[63] Liu, T., Bahl, P., and Chlamtac, I., "Mobility modeling, location tracking, and trajectory prediction in wireless atm networks," *IEEE Journal on Selected Areas in Communications* 16(6) (1998) 922-936.

[64] McLarnon, B., Vhf/uhf/microwave radio propagation: A primer for digital experimenters, 2005, Accessed on November 21, 2005; Available at: http://www.ictp.trieste.it/~radionet/ghana1998/LINKLOSS/INDEX.HTM.

[65] Michalewicz, Z., Genetic algorithms + data structures = evolution programs. third ed., Springer-Verlag, New York, NY, 1996.

[66] Ming-Hui, J., Eric Hsiao-Kuang, W., and Jorng-Tzong, H. "Location query based on moving behavior," in: *Proceedings of the 11th International Conference on Computer Communications and Networks*, 2002, 268-273.

[67] Mitrovic, D. "Short term prediction of vehicle movements by neural networks," in: *Proceedings of the 3rd International Conference on Knowledge-Based Intelligent Information Engineering Systems*, 1999, 187-190.

المنارة للاستشارات

www.manaraa.com

[68] Mori, N., Imanishi, S., Kita, H., and Nishikawa, Y. "Adaptation to changing environments by means of the memory-based thermodynamical genetic algorithm," in: *Proceedings of the 7th International Conference on Genetic Algorithms*, 1997, 299-306.

[69] Netgear, Prosafe™ 802.11g wireless access point, 2005, Accessed on November 21, 2005; Available at: http://www.netgear.com/products/details/WG302.php.

[70] Pearlman, M.R. and Haas, Z.J., "Determining the optimal configuration for the zone routing protocol," *IEEE Journal on Selected Areas in Communications* 17(8) (1999) 1395-1414.

[71] Perkins, C.E. and Bhagwat, P., "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *ACM SIGCOMM Computer Communication Review* 24(4) (1994) 234-244.

[72] Perkins, C.E. and Royer, E.M. "Ad-hoc on-demand distance vector routing," in: *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, 1999, 90-100.

[73] Quintero, A., Pierre, S., and Macabeo, B., "A routing protocol based on node density for ad hoc networks," *Ad Hoc Networks* 2(3) (2004) 335-349.

[74] Robert, F., Gay, D.M., and Kernighan, B.W., Ampl: A modeling language for mathematical programming, Thomson/Brooks/Cole, Pacific Grove, CA, 2003.

[75] Schaffer, J.D., Caruana, R.A., Eshelman, L.J., and Das, R. "A study of control parameters affecting online performance of genetic algorithms for function optimization," in: *Proceedings of the 3rd International Conference on Genetic Algorithms*, George Mason University, 1989, 51-60.

[76] Schraudolph, N.N. and Belew, R.K., "Dynamic parameter encoding for genetic algorithms," *Machine Learning* 9 (1992) 9-21.

[77] Schwefel, H.-P. and Manner, R. (Eds.), Lecture notes in computer science, Vol. 496, Springer-Verlag, New York, NY, 1991.

[78] Schwefel, H.-P., Evolution and optimum seeking, John Wiley & Sons Inc., New York, NY, 1995.

[79] Shengxiang, Y. "Memory-based immigrants for genetic algorithms in dynamic environments," in: *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, Washington DC, USA, ACM Press, 2005, 1115-1122.

المنارة للاستشارات

www.manaraa.com

[80] Shukla, D., Mobility models in ad hoc networks, Master's Thesis, KReSIT-IIT Bombay, 2001.

[81] Simplot-Ryl, D. and Stojmenovic, I., "Data communications and topology control in wireless ad hoc networks," *Ad Hoc Networks* 3(5) (2005) 507-508.

[82] Srivaree-ratana, C., Konak, A., and Smith, A.E., "Estimation of all-terminal network reliability using an artificial neural network," *Computers & Operations Research* 29(7) (2002) 849-868.

[83] Stepanov, I. and Rothermel, K., "Simulating mobile ad hoc networks in city scenarios," *Computer Communications* 30(7) (2007) 1466.

[84] Su, W., Lee, S.-J., and Gerla, M. "Mobility prediction in wireless networks," in: *Proceedings of the 21st Century Military Communications Conference, (MILCOM 2000)*, 2000, 491-495.

[85] Su, W., Lee, S.-J., and Gerla, M., "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management* 11 (2001) 3-30.

[86] Tang, J., Xue, G., and Zhang, W. "Reliable routing in mobile ad hoc networks based on mobility prediction," in: *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2004, 466-474.

[87] Torn, A. and Zilinkas, A. (Eds.), Lecture notes in computer science, Vol. 350, Springer-Verlag, New York, NY, 1989.

[88] Vavak, F., Jukes, K., and Fogarty, T.C. "Adaptive combustion balancing in multiple burner boiler using a genetic algorithm with variable range of local search," in: *Proceedings of the 7th International Conference on Genetic Algorithms*, 1997, 719-726.

[89] Wang, N.-C. and Chang, S.-W., "A reliable on-demand routing protocol for mobile ad hoc networks with mobility prediction," *Computer Communications* 29(1) (2005) 123-135.

[90] Wolpert, D.H. and Macready, W.G., No free lunch theorems for search, Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995.

[91] Xue, F. and Kumar, P.R., "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks* 10(2) (2004) 169-181.

المنارة للاستشارات

www.manaraa.com

[92] Yaochu, J. and Branke, J., "Evolutionary optimization in uncertain environments-a survey," *IEEE Transactions on Evolutionary Computation* 9(3) (2005) 303-317.

[93] Yavaş, G., Katsaros, D., Ulusoy, Ö., and Manolopoulos, Y., "A data mining approach for location prediction in mobile environments," *Data & Knowledge Engineering* 54(2) (2005) 121-146.

المنارة للاستشارات

www.manaraa.com

APPENDIX

Ap- 1 NDBGA Performance compared with conventionally decoded binary GA (CDGA) on the continuous test problems

Comparison of NDBGA With a Conventionally Decoded GA (CDGA)

	NDBGA						CDGA	
Fn.	Population best	Average population best	Standard dev. of population best	Average function eval.	Standard dev. of function Eval.	Population best	Average population best	Standard dev. of population best
F1	$<1\times10^{-30}$	$<1\times10^{-30}$	$<1\times10^{-30}$	5,538	938	1.21×10^{-3}	1.21×10^{-3}	4.45×10^{-19}
F2	5.08×10^{-12}	1.75×10^{-10}	3.56×10^{-10}	23,615	16,940	1.57×10^{-3}	1.57×10^{-3}	2.22×10^{-19}
F3	-30.0	-30.0	0.0	15,627	25,839	-30.0	-30.0	$<1\times10^{-30}$
F4	2.09×10^{-1}	2.73×10^{-1}	5.91×10^{-2}	169,918	99,905	9.29×10^{-2}	2.60×10^{-1}	1.10×10^{-1}
F4[c]	4.88×10^{-21}	2.00×10^{-20}	1.38×10^{-20}	59,003	2,253	2.95×10^{-7}	2.95×10^{-7}	5.43×10^{-23}
F5	0.99800384	0.99800384	1.33×10^{-14}	2,817	8,342	0.9980116	0.9980116	2.28×10^{-16}
F6	5.55×10^{-17}	5.55×10^{-17}	0.0	1,649	339	1.21×10^{-3}	1.21×10^{-3}	$<1\times10^{-30}$
F7	-16.0917200	-16.0917200	1.34×10^{-10}	3,076	1,716	-16.0832124	-16.0832124	7.29×10^{-15}
F8	0.3978874	0.3978874	1.30×10^{-9}	11,185	10,328	0.3989716	0.3989716	1.14×10^{-16}
F9	-1.0316285	-1.0316285	2.43×10^{-10}	3,953	6,587	-1.0311667	-1.0311667	$<1\times10^{-30}$
F10	3.0000000	3.0000000	1.85×10^{-13}	2,900	4,621	3.0154081	3.0154081	$<1\times10^{-30}$
F11	-186.7309088	-186.7309088	2.97×10^{-9}	17,156	9,094	-185.5815858	-185.5815858	2.92×10^{-14}
F12	1.90×10^{-9}	2.91×10^{-8}	2.31×10^{-8}	219,856	95,307	7.18×10^{-2}	8.60×10^{-2}	2.13×10^{-2}
F13	5.08×10^{-10}	7.61×10^{-10}	1.14×10^{-10}	12,475	1,056	$<1\times10^{-30}$	$<1\times10^{-30}$	$<1\times10^{-30}$
F14	9.46×10^{-10}	1.53×10^{-9}	3.23×10^{-10}	84,377	90,557	7.20×10^{-3}	1.51×10^{-2}	3.51×10^{-2}
F15	$<1\times10^{-30}$	$<1\times10^{-30}$	$<1\times10^{-30}$	8,243	7,569	4.77×10^{-3}	4.77×10^{-3}	8.90×10^{-19}
F16	$<1\times10^{-30}$	$<1\times10^{-30}$	$<1\times10^{-30}$	43,690	19,968	3.99×10^{-1}	3.99×10^{-1}	1.14×10^{-16}
F17	8.45×10^{-10}	1.55×10^{-8}	1.06×10^{-8}	125,786	40,860	1.5974995	1.5974995	6.83×10^{-16}
F18	6.13×10^{-10}	9.47×10^{-10}	2.11×10^{-10}	131,009	1,791	1.21×10^{-2}	1.21×10^{-2}	$<1\times10^{-30}$
F19	5.12×10^{-10}	2.58×10^{-8}	2.41×10^{-8}	313,455	44,343	6.61×10^{-1}	6.87×10^{-1}	8.13×10^{-2}
F20	$<1\times10^{-30}$	$<1\times10^{-30}$	$<1\times10^{-30}$	30,152	15,898	2.68×10^{-1}	2.68×10^{-1}	5.70×10^{-17}

www.manaraa.com

Ap- 2 Test problem generation code for Matlab

This function accepts rand_seed, NoOfUsers, NoOfAgents, and NumberOfTimeSlots as input parameters. NumberOfTimeSlots is set to 1 for static problems. The NodeXY cell stores an array of X and Y coordinates of the users for each time step *t* as NodeXY{t}(UserID,1) for the x-coordinate and NodeXY{t}(UserID,2) for the y-coordinate of user UserID ∈ [1,2,3,…,NoOfUsers]

```
rand('state',rand_seed); % Always the same results default

NoOfUsers = NoOfUsersLcl; %The number of users getting service
NoOfAgents = NoOfAgentsLcl; %The number of remote controlled agents
NodeSize = NoOfUsers + NoOfAgents;    %total number of moving nodes

MinX = 0;
MaxX = 5;
MinY = 0;
MaxY = 5;

if(NumberOfTimeSlots == 1)
    StartXY_User = unifrnd(0,MaxX,NoOfUsers,2);    %randomly create
starting XY coordinates default
else
    if(NoOfAgents <= 5)
        StartXY_User = unifrnd(0,MaxX/3,NoOfUsers,2);    %randomly
create starting XY coordinates default
    else
        StartXY_User = unifrnd(0,MaxX/2,NoOfUsers,2);    %randomly
create starting XY coordinates default
    end
end


DestXY_User = unifrnd(0,MaxX-1,NoOfUsers,2); %randomly create
destination XY coordinates default

StartXY_Agent_temp = zeros(NoOfAgents,2);

StartXY_temp = [StartXY_User %combine the start positions
    StartXY_Agent_temp] ;

NodeXY = cell(1, NumberOfTimeSlots);
```

www.manaraa.com

```
NodeXY{1} = StartXY_temp; %assign the current XY coordinates of the
nodes to the initial coordinates

Vmin = 0.02; %minimum linear speed of mobile nodes default
Vmax = 0.05; %maximum linear speed of mobile nodes default

Velocity = unifrnd(Vmin,Vmax,NoOfUsers,1); %randomly assign velocities
RangeNodes = ones(NodeSize,1) * 1.0; %everbody has the same range

DeltaXY = DestXY_User - NodeXY{1}(1:NoOfUsers,:);   %find delta X and Y
coordinates from destination to the current location
UV(:,1) = DeltaXY(:,1)./((DeltaXY(:,1).^2 + DeltaXY(:,2).^2).^0.5);
%normalize the deltas calculated above
UV(:,2) = DeltaXY(:,2)./((DeltaXY(:,1).^2 + DeltaXY(:,2).^2).^0.5);
%normalize the deltas calculated above

%start generate user trajectory
for time = 1:NumberOfTimeSlots

    %get the euclidian distance between current position and the
destination
    DeltaXY = DestXY_User - NodeXY{time}(1:NoOfUsers,:);   %find delta
X and Y coordinates from destination to the current location
    ScaleFactors = ( (DeltaXY(:,1).^2 + DeltaXY(:,2).^2).^0.5 );
%calculate the distance from current location to the destination

    alfa = 0.95; %weight of the current direction
    UV(:,1) = alfa*UV(:,1) + (1-alfa)*DeltaXY(:,1)./ScaleFactors;
%normalize the distance calculated above
    UV(:,2) = alfa*UV(:,2) + (1-alfa)*DeltaXY(:,2)./ScaleFactors;
%normalize the distance calculated above

    DestReached{time} = find(ScaleFactors<Velocity); %find the nodes
that have made it to the destination

    UV(DestReached{time},1) = 0;    %no more movement for the nodes at
their destinations
    UV(DestReached{time},2) = 0;    %no more movement for the nodes at
their destinations

    for j = 1:NoOfUsers %for every node
        if (rand<0.1) %change the path angle with a small probability
            teta = unifrnd(-pi/4,pi/4);%random rotation angle
            rot_matrix = [cos(teta) sin(teta)
                    -sin(teta) cos(teta)] ;%2D rotation matrix

            RotatedUVj = [UV(j,1) UV(j,2)] * rot_matrix; %rotate the
direction unit vectors
            UV(j,1) = RotatedUVj(1,1);%assign rotated direction vectors
            UV(j,2) = RotatedUVj(1,2);%assign rotated direction vectors
        end
    end
```

المنارة للاستشارات
www.manaraa.com

```
    Velocity = unifrnd(Vmin,Vmax,NoOfUsers,1); %randomly assign
velocities
    AgentVelocity = ones(NoOfAgents,1) * 0; %assign agent velocities to
zero

    NodeXY{time+1}(1:NoOfUsers,1) =
NodeXY{time}(1:NoOfUsers,1)+Velocity(:,1).*UV(:,1); %calculate the next
X coordinates
    NodeXY{time+1}(1:NoOfUsers,2) =
NodeXY{time}(1:NoOfUsers,2)+Velocity(:,1).*UV(:,2); %calculate the next
Y coordinates


end%generate user trajectory
```

المنارة للاستشارات
www.manaraa.com

Ap- 3 Test problem data

Static cases

NumberOfTimeSlots = 1

Problem Scale	NoOfUsers	NoOfAgents	rand_seed
Small	6	4	675, 948, 375, 468, 427, 843, 674, 241, 643, 846, 451, 944, 346, 466, 841, 523, 597, 411, 624, 977
Medium	10	10	675, 948, 375, 468, 427, 843, 674, 241, 643, 846, 451, 944, 346, 466, 841, 523, 597, 411, 624, 977
Large	20	20	675, 948, 375, 468, 427, 843, 674, 241, 643, 846

Dynamic cases

NumberOfTimeSlots = 100

Problem Scale	NoOfUsers	NoOfAgents	rand_seed
Small	4	3	675, 948, 375, 468, 427, 843, 674, 241, 643, 846, 451, 944, 346, 466, 841, 523, 597, 411, 624, 977
Medium	8	6	675, 948, 375, 468, 427, 843, 674, 241, 643, 846
Large	14	10	547, 862, 468, 142, 465

المنارة للاستشارات

www.manaraa.com

Ap- 4 AMPL model file

```
set NODES;              #The set of nodes
param x{NODES};               #user node x coordinates (including
initial agent coordinates)
param y{NODES};               #user node y coordinates (including
initial agent coordinates)

set AGENTS;             #The set of agent nodes

param Vmax{AGENTS};           #Maximum distance that agents can travel
in one time step

param range;            #Wireless transmission range
param MaxDataRate;            #Maximum data transmission rate
param M:=10000;


#Arcs between user nodes
set ARCSa:={i in NODES diff AGENTS, j in NODES diff AGENTS:  i<>j and
( sqrt( (x[i]-x[j])^2+(y[i]-y[j])^2 ) <= range) };


#Arcs between agents and user nodes, an arc is drawn if the agent can
reach a currently out of range user by travelling
set ARCSb:={i in NODES diff AGENTS, j in AGENTS:  i<>j and  ( sqrt(
(x[i]-x[j])^2+(y[i]-y[j])^2 ) <= ( range + Vmax[j] ) ) };

set ARCSc:={i in AGENTS, j in NODES diff AGENTS:  i<>j and  ( sqrt(
(x[i]-x[j])^2+(y[i]-y[j])^2 ) <= ( range + Vmax[i] ) ) };

#Arcs between agent nodes themselves, an arc is drawn if the agents
currently out of range can reach themselves by travelling
set ARCSd:={i in AGENTS, j in AGENTS:  i<>j and  ( sqrt( (x[i]-
x[j])^2+(y[i]-y[j])^2 ) <= ( range + Vmax[i] + Vmax[j] ) ) };

#Unite the arcs sets
set ARCS:=ARCSa union ARCSb union ARCSc union ARCSd;

param npiece_dist;            #number of points to approximate arc
distance
param npiece_Vmax;            #number of points to approximate travel
distance
param npiece_cap;             #number of points to approximate data
rate

param NumOfUsers;             #number of Ad Hoc users


#Arc distance approximation
param rate_x{i in NODES, j in AGENTS, p in 1..npiece_dist : i<>j };
```

www.manaraa.com

```
param limit_x{i in NODES, j in AGENTS, p in 1..(npiece_dist-1) : i<>j
};

param rate_y{i in NODES, j in AGENTS, p in 1..npiece_dist : i<>j };

param limit_y{i in NODES, j in AGENTS, p in 1..(npiece_dist-1) : i<>j
};


#Travel distance approximation
param rate_Vmax_x{i in AGENTS, p in 1..npiece_Vmax };

param limit_Vmax_x{i in AGENTS, p in 1..(npiece_Vmax-1) };

param rate_Vmax_y{i in AGENTS, p in 1..npiece_Vmax };

param limit_Vmax_y{i in AGENTS, p in 1..(npiece_Vmax-1) };


#Link capacity (data rate) approximation
param rate_cap{i in NODES, j in AGENTS, p in 1..npiece_cap : i<>j };

param limit_cap{i in NODES, j in AGENTS, p in 1..(npiece_cap-1) : i<>j
};



var d{ARCS} >=0;                                                  #arc
distance
var xl{AGENTS} >=0;                                               #new
location of agents x coordinate
var yl{AGENTS} >=0;                                               #new
location of agents y coordinate
var dx{ARCS} >=0;                                                 #arc
distance in x direction
var dy{ARCS} >=0;                                                 #arc
distance in y direction

var delta_x{AGENTS};
      #agent travel in x direction
var delta_y{AGENTS};
      #agent travel in y direction

var u{ARCS} >=0, <= MaxDataRate;    #data transmission rate

var d2{i in NODES, j in AGENTS : i <> j } >=0;                    #the
square of the arc distances


var minflow >= 0;
      #all-pair min flow value

set VirtualCommodity := {S in NODES, T in NODES: S<T};
      #virtual commodities, each belong to a node pair
```

www.manaraa.com

```
set VirtualCommodity2 := {S in NODES diff AGENTS, T in NODES diff
AGENTS: S<T};           #virtual commodities within users only


var CommodityFlows {ARCS,VirtualCommodity} >= 0, <= MaxDataRate;
      #virtual commodity flow values

var MaxFlowCommodityType {VirtualCommodity} <= MaxDataRate;
      #the maximum possible flow of virtual commodity


maximize AllPairMaxFlow : minflow + ( sum{(S,T) in VirtualCommodity2}
MaxFlowCommodityType[S,T] ) / ( NumOfUsers * (NumOfUsers - 1) / 2 );

#the minimum of maximum possible virtual commodity flows
subject to AllPairMin {(S,T) in VirtualCommodity2}: minflow <=
MaxFlowCommodityType[S,T];

#every virtual commodity flow need to be within bounds, and they do not
take up other virtual commodities bandwidth
subject to capacity_all_1 {(i,j) in ARCS, (S,T) in VirtualCommodity}:
CommodityFlows[i,j,S,T] <= u[i,j];


#if node not the source or the target of the specific virtual
commodity, flow in equals flow out
subject to flow_balance1 {i in NODES, (S,T) in VirtualCommodity: S<>i
and T<>i}: sum{j in NODES: (i,j) in ARCS} CommodityFlows[i,j,S,T] -
sum{j in NODES: (j,i) in ARCS} CommodityFlows[j,i,S,T] = 0;

#if node the source of the specific virtual commodity, flow difference
equals +flow
subject to flow_balance2 {(S,T) in VirtualCommodity}: sum{j in NODES:
(S,j) in ARCS} CommodityFlows[S,j,S,T] - sum{j in NODES: (j,S) in ARCS}
CommodityFlows[j,S,S,T] = MaxFlowCommodityType[S,T];

#if node the target of the specific virtual commodity, flow difference
equals -flow
subject to flow_balance3 {(S,T) in VirtualCommodity}: sum{j in NODES:
(T,j) in ARCS} CommodityFlows[T,j,S,T] - sum{j in NODES: (j,T) in ARCS}
CommodityFlows[j,T,S,T] = -1*MaxFlowCommodityType[S,T];


#Arc distance constraints (absolute value)

subject to Distance_x1a {i in NODES, j in AGENTS : (i,j) in ARCS }:
dx[i,j] >= x[i]-xl[j];

subject to Distance_x2a {i in NODES, j in AGENTS : (i,j) in ARCS }:
dx[i,j] >= xl[j]-x[i];

subject to Distance_y1a {i in NODES, j in AGENTS : (i,j) in ARCS }:
dy[i,j] >= y[i]-yl[j];
```

www.manaraa.com

```
subject to Distance_y2a {i in NODES, j in AGENTS : (i,j) in ARCS }:
dy[i,j] >= yl[j]-y[i];


#Arc distance constraints, approximate the squares of each distance
component and add to get arc (i,j) length (squared)

subject to InWirelessRange_1a {i in NODES, j in AGENTS : (i,j) in ARCS
}:
   << {p in 1..npiece_dist-1} limit_x[i,j,p];{p in 1..npiece_dist}
rate_x[i,j,p]>> dx[i,j] +<< {p in 1..npiece_dist-1} limit_y[i,j,p];{p
in 1..npiece_dist} rate_y[i,j,p]>>
dy[i,j] <= d2[i,j];


#Set arc capacities with the arc length squared

subject to Capacity_A {i in NODES, j in AGENTS : (i,j) in ARCS }:
u[i,j] <=  << {p in 1..npiece_cap-1} limit_cap[i,j,p];{p in
1..npiece_cap} rate_cap[i,j,p] >> (d2[i,j],range^2);

subject to Capacity_B {j in AGENTS, i in NODES : (j,i) in ARCS }:
u[j,i] <= u[i,j];


#Travel distance constraints (absolute value)

subject to Travel_x_1 {j in AGENTS}: delta_x[j] >= x[j]-xl[j];

subject to Travel_x_2 {j in AGENTS}: delta_x[j] >= xl[j]-x[j];

subject to Travel_y_1 {j in AGENTS}: delta_y[j] >= y[j]-yl[j];

subject to Travel_y_2 {j in AGENTS}: delta_y[j] >= yl[j]-y[j];


#Travel range constraints

subject to InMotionRange_1 {i in AGENTS}:
   << {p in 1..npiece_Vmax-1} limit_Vmax_x[i,p];{p in 1..npiece_Vmax}
rate_Vmax_x[i,p]>> delta_x[i] +<< {p in 1..npiece_Vmax-1}
limit_Vmax_y[i,p];{p in 1..npiece_Vmax} rate_Vmax_y[i,p]>>
delta_y[i] <= Vmax[i]^2;
```

www.manaraa.com

Ap- 5 AMPL run file

```
include in.inf;
model model_cap.mod;

#Data file
data NodeLocation.dat #from Matlab

#Mobility distance approximation
for {i in AGENTS} {
      for { p in 1..(npiece_Vmax-1)}{
            let limit_Vmax_x[i,p]:=p*(Vmax[i]/(npiece_Vmax-1));
            let limit_Vmax_y[i,p]:=p*(Vmax[i]/(npiece_Vmax-1));
            }

            let rate_Vmax_x[i,1]:=limit_Vmax_x[i,1];
            let rate_Vmax_y[i,1]:=limit_Vmax_y[i,1];

       for { p in 2..(npiece_Vmax-1)}{
            let rate_Vmax_x[i,p]:=(limit_Vmax_x[i,p]^2-
limit_Vmax_x[i,p-1]^2)/(limit_Vmax_x[i,p]-limit_Vmax_x[i,p-1]);
            let rate_Vmax_y[i,p]:=(limit_Vmax_y[i,p]^2-
limit_Vmax_y[i,p-1]^2)/(limit_Vmax_y[i,p]-limit_Vmax_y[i,p-1]);
            }

      let rate_Vmax_x[i,npiece_Vmax]:=M;
      let rate_Vmax_y[i,npiece_Vmax]:=M;
     }

#Range distance approximation
for {i in NODES, j in AGENTS : i <>j } {
      for { p in 1..(npiece_dist-1)}{
            let limit_x[i,j,p]:=p*(range/(npiece_dist-1));
            let limit_y[i,j,p]:=p*(range/(npiece_dist-1));
            }

            let rate_x[i,j,1]:=limit_x[i,j,1];
            let rate_y[i,j,1]:=limit_y[i,j,1];

      for { p in 2..(npiece_dist-1)}{
            let rate_x[i,j,p]:=(limit_x[i,j,p]^2-limit_x[i,j,p-
1]^2)/(limit_x[i,j,p]-limit_x[i,j, p-1]);
                  let rate_y[i,j,p]:=(limit_y[i,j,p]^2-limit_y[i,j,p-
1]^2)/(limit_y[i,j,p]-limit_y[i,j, p-1]);
            }

      let rate_x[i,j,npiece_dist]:=M;
      let rate_y[i,j,npiece_dist]:=M;
     }


#Data rate approximation
```

المنارة للاستشارات
www.manaraa.com

```
for { i in NODES, j in AGENTS : i <>j } {
      for { p in 1..(npiece_cap-1)}{
            let limit_cap[i,j,p]:=( p*(range/(npiece_cap-1)) )^2;
      }

      let rate_cap[i,j,1]:= MaxDataRate * ( 1.0/(1 + exp(10*(
limit_cap[i,j,1]^0.5/range - 0.5))) - 1.0 ) / limit_cap[i,j,1];

      for { p in 2..(npiece_cap-1)}{
            let rate_cap[i,j,p]:= MaxDataRate * ( 1.0/(1 + exp(10*(
limit_cap[i,j,p]^0.5/range - 0.5))) - 1.0/(1 + exp(10*(
limit_cap[i,j,p-1]^0.5/range - 0.5))) ) / (limit_cap[i,j,p]-
limit_cap[i,j,p-1]);
      }

      let rate_cap[i,j,npiece_cap]:=0;
}

#Fix the data rates for users
for { (i,j) in ARCS: i not in AGENTS and j not in AGENTS } {
      if ( ((x[i]-x[j])^2+(y[i]-y[j])^2)^0.5<=range) then{
            fix u[i,j] := MaxDataRate * 1.0/(1+exp(10*( ((x[i]-
x[j])^2+(y[i]-y[j])^2)^0.5/range - 0.5)));

      }
      else{
            fix u[i,j]:= 0;

      }
}


option cplex_options 'mipgap=0.10 timelimit=120';
solve;
printf"">agentLocation.out;
for {i in AGENTS}{
      printf"%f %f;\n",xl[i],yl[i]>>agentLocation.out;
}
```

www.manaraa.com

Ap- 6 Example data file for AMPL model (Stationary scenario)

```
param : NODES : x y :=
1 1.77560 2.85915
2 0.71296 4.84898
3 4.53495 4.72558
4 1.25040 3.95480
5 1.84809 2.74672
6 2.93035 1.23632
7 2.37250 2.88927
8 4.39605 2.40152
;

param range:= 1.00000;
param npiece_dist:= 8;
param npiece_Vmax:= 8;
param npiece_cap:= 8;
param MaxDataRate:= 54;

param NumOfUsers:= 5;

param : AGENTS : Vmax :=
6 4.23889
7 2.83694
8 4.42213
;
```

المنارة للاستشارات
www.manaraa.com