

AUS Repository

MANET Cluster Optimization Using ILP/SAT Techniques

Item Type	Thesis
Authors	Zahidi, Syed Zohaib Hussain
Download date	2026-06-08 17:04:54
Link to Item	http://hdl.handle.net/11073/2763



MANET CLUSTER OPTIMIZATION USING ILP/SAT TECHNIQUES

A THESIS IN COMPUTER ENGINEERING
Master of Science in Computer Engineering

Presented to the faculty of the American University of Sharjah
College of Engineering
in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE

by
SYED ZOHAIB HUSSAIN ZAHIDI
B.S. 2008

Sharjah, UAE
Jan 2012

© 2012

SYED ZOHAIB HUSSAIN ZAHIDI

ALL RIGHTS RESERVED

We approve the thesis of Syed Zohaib Hussain Zahidi

Date of signature

Dr. Fadi Ahmed Aloul
Associate Professor
Department of Computer Science and Engineering
Thesis Advisor

Dr. Assim Sagahyoon
Associate Professor, & Department Head
Department of Computer Science and Engineering
Thesis Co-Advisor

Dr. Taher Abualrub
Professor
Department of Mathematics and Statistics
Graduate Committee Member

Dr. Rana Ejaz Ahmed
Associate Professor
Department of Computer Science and Engineering
Graduate Committee Member

Dr. Tarik Ozkul
Associate Professor
Department of Computer Science and Engineering
Graduate Committee Member

Dr. Hany El-Kadi
Associate Dean
College of Engineering

Dr. Yousef Al Assaf
Dean
College of Engineering

Dr. Khaled Assaleh
Director, Graduate Studies

AN ABSTRACT IN AN AMERICAN
UNIVERSITY OF SHARJAH THESIS:

ASSESSING THE FEASIBILITY OF SAT/ILP SOLVERS IN SOLVING THE
CLUSTERING PROBLEM IN MOBILE AD-HOC NETWORKS

Syed Zohaib Hussain Zahidi, Candidate for the Master of Science Degree

American University of Sharjah, 2012

ABSTRACT

In recent years, there have been several improvements in the performance of Integer Linear Programming (ILP) and Boolean Satisfiability (SAT) solvers. These improvements have encouraged the modeling of complex engineering problems as ILP problems. These engineering problems are diverse in nature and include genetics, optimization of power consumption, scheduling, cryptography, and more. One such problem is the ‘clustering problem’ in Mobile Ad-Hoc Networks (MANETs). The clustering problem in MANETs consists of selecting the most suitable nodes of a given MANET topology as clusterheads and ensuring that regular nodes are connected to clusterheads in such a way that the network lifetime is maximized.

This thesis focuses on assessing the performance of state-of-the-art generic ILP and 0-1 SAT-based ILP solvers in solving ILP formulations of the clustering problem. The thesis consists of four parts. The first part of this thesis consists of improving the existing ILP formulations of the clustering problem. The second part involves enhancing the ILP formulation of the clustering problem through the addition of intra-cluster communication, coverage constraints and multihop links. The third part focuses on the development of an improved tool to enable conversion of user-created on-screen topologies to an ILP formulation. The fourth and final part of this thesis is the detailed performance comparison of a selected set of Generic ILP and 0-1 SAT-based ILP solvers in solving the improved ILP formulations of the clustering problem generated using the tool.

The results obtained indicate that from our selected set of solvers, generic ILP solvers are able to handle relatively large scale MANET topologies, while 0-1 SAT-based ILP solvers are the fastest, for small scale networks. For small scale networks the proposed ILP formulations, such as the Star-Ring base model, together with the high performance solvers would be suitable for use in real-world environments. However for large scale networks, as the time to cluster the network grows exponentially, the solvers will be unable to cluster the network in accordance with the demands of a real-world environment.

CONTENTS

ABSTRACT	iii
LIST OF FIGURES	vi
LIST OF TABLES	viii
ACKNOWLEDGEMENTS	x
 Chapter	
1. INTRODUCTION.....	1
1.1 Background.....	2
1.2 Literature Review.....	8
1.3 Research Methodology.....	15
2. INVESTIGATION.....	17
2.1 Existing Model Analysis.....	17
2.2 Analysis and Selection of Solvers.....	25
2.3 Analysis of Existing Tool.....	29
3. DESIGN AND IMPLEMENTATION.....	32
3.1 Proposed Model And Enhancements.....	32
3.2 Design of New and Improved Tool.....	49
4. TESTS.....	53
4.1 Initial Testing Using The EEC-FCB Model.....	53
4.2 Test 1: Comparing the Performance of the Selected Set of Solvers Across all Proposed ILP Formulation Models of the Clustering Problem.....	58
4.3 Test 2: Testing the Ability of Solvers to Solve ILP Formulations of the Clustering Problem for Large Scale Networks.....	59
5. EVALUATION OF RESULTS.....	61
5.1 Test 1: Results.....	61
5.2 Evaluation of the Results of Test 1.....	70
5.3 Test 2: Results.....	73
5.4 Evaluation of the Results of Test 2.....	75
5.5 Recommendations for Further Testing.....	75
5.6 Feasibility of Using Generic-ILP or Pseudo-Boolean SAT Solvers to Solve the Clustering Problem in a Practical Setting.....	76
6. CONCLUSION.....	77
7. FUTURE WORK.....	79
REFERENCE LIST.....	81
 Appendix	
A. THE m^* VARIABLE.....	86
B. EXAMPLE SOLVED WITH PROPOSED MODEL AND ENHANCEMENTS.....	88
VITA.....	100

FIGURES

Figure	Page
1. SAT Problem Showing 2 ⁿ Different Possible Variable Assignments.....	3
2. Improvement in SAT Solver Performance Over the Last Decade.....	3
3. Performance Comparison of ILP Model of the Clustering Problem and Virtual Grid Architecture.....	8
4. Different Network Topologies: Fully Connected Backbone, Connected Backbone and Redundant Models.....	9
5. Performance Comparison of ILP Models (Fully Connected Backbone and Connected Backbone) with non ILP models (FEER).....	10
6. Wireless Sensor Network Layout: From the Sensor to the End-User.....	11
7. Routing in a Grid of Sensors and Demand Points.....	12
8. Different Possible Sensor States: Active, Sleep and Clusterhead.....	13
9. Cost of Solutions Generated using ILP model vs Search Heuristic (TABU-RCC) for Energy-Optimal Model of Wireless Sensor Networks.....	14
10. Fully Connected Backbone Topology.....	19
11. Connected Backbone Topology Using a Master Clusterhead.....	19
12. Performance Comparison of SCIP with Commercial and Non-Commercial ILP Solvers.....	27
13. PseudoBoolean Contest Results Comparing Solver Performance in Terms of Number of Instances Solved in a Given Time Period.....	28
14. Screenshot of Existing Tool.....	29
15. Solved Topology With a Star-Ring Backbone.....	34
16. Multihop Network Topology.....	39
17. Intra-Cluster Communication in a Network Topology.....	43
18. A 12 Node Network Topology Displaying Coverage.....	47
19. Solving a 12 Node Topology With a Combined Model Including Coverage Constraints, Intra-Cluster Communication and Multihop Connections.....	48
20. Example Topology To Be Solved With All Enhancements.....	48
21. Proposed Tool Screen 1: Sample Topology Shown with Corresponding Solution.....	50

22. Proposed Tool Screen 2: Viewing Different Aspects of a Generated Solution.....	51
23. Proposed Tool Screen 3: Connection Dialog Box Displaying Status Messages Allowing Users to Monitor Progress	52
24. Example of Conflict in FCB - Original Topology and Solved Topology.....	53
25. Test 1 Results: EEC-FCB Model in Configurations with 3 Clusterheads.....	62
26. Test 1 Results: EEC-FCB Model in Configurations with 4 Clusterheads.....	63
27. Test 1 Results: EEC-FCB Model in Configurations with 5 Clusterheads.....	63
28. Test 1 Results: SR Model in Configurations with 3 Clusterheads.....	65
29. Test 1 Results: SR Model in Configurations with 4 Clusterheads.....	65
30. Test 1 Results: SR Model in Configurations with 5 Clusterheads.....	66
31. Test 1 Results: SR+IC Model in Configurations with 3 Clusterheads.....	66
32. Test 1 Results: SR+IC Model in Configurations with 4 Clusterheads.....	67
33. Test 1 Results: SR+IC Model in Configurations with 5 Clusterheads.....	68
34. Test 1 Results: SR+MH Model in Configurations with 3 Clusterheads.....	69
35. Test 1 Results: SR+MH Model in Configurations with 4 Clusterheads.....	69
36. Test 1 Results: SR+MH Model in Configurations with 5 Clusterheads.....	70
37. The Effect of Increasing the Number of Clusterheads in a Fixed Size Topology on the Time Taken by Solvers to Solve the Corresponding SR ILP Formulation	72
38. The Effect of Increasing the Number of Clusterheads in a Fixed Size Topology on the Time Taken by Solvers to Solve the Corresponding SR+IC ILP Formulation.....	72
39. Test 2 Results: SR Model in Configurations with 3 Clusterheads.....	74
40. Test 2 Results: SR Model in Configurations with 4 Clusterheads.....	74
41. Test 2 Results: SR Model in Configurations with 5 Clusterheads.....	75

TABLES

Table	Page
1. Variable Assignment for EEC-FCB Conflict Example.....	54
2. Test Set 1: Network Configurations to Evaluate the Performance of all Solvers Across all Proposed ILP Formulations.....	58
3. Test Set 2: Network Configurations to Test the Ability of Solvers to Handle Large Networks	60
4. Test 1 Results: Time Taken by all Solvers to Solve Network Configurations using the EEC-FCB Model	62
5. Test 1 Results: Time Taken by all Solvers to Solve Network Configurations using the SR Model.....	64
6. Test 1: Time Taken by all Solvers to Solve Network Configurations using the SR + IC Model.....	67
7. Test 1: Time Taken by all Solvers to Solve Network Configurations using the SR+MH Model.....	68
8. Results of Test Set 2: Testing the Ability of Solvers to Handle Large Networks	73

ABBREVIATIONS

FCB – Fully Connected Backbone
CB – Connected Backbone
SR – Star-Ring
GUI – Graphical User Interface
MH – Multihop
IC – Intra-Cluster
CH – Clusterhead
BS – Base Station
WSN – Wireless Sensor Network
MANET – Mobile Ad-Hoc Network

ACKNOWLEDGEMENTS

I would like to thank my advisors, Dr. Fadi Aloul and Dr. Assim Sagahyoon, for their continuous support and helpful advice but more so for their patience, guidance and constructive criticism which helped me mold this thesis into its final form.

I would like to thank Dr. Wassim El Hajj for helping me understand the application of Integer Linear Programming formulation in the context of Mobile Ad Hoc Networks, and the potential areas of focus.

My sincere gratitude and heartfelt thanks are extended to all the professors in the Department of Computer Engineering and Science at the American University of Sharjah, for their support and advice over the course of my graduate study.

This thesis is dedicated to my parents and my sister for their patience, love and encouragement. Without their support this thesis would never have been completed.

CHAPTER 1

INTRODUCTION

Over the past decade extensive research has led to improvements in generic Integer Linear Programming (ILP) solvers and 0-1 Boolean Satisfiability (SAT) based ILP solvers. The introduction, and development, of new techniques has improved the performance of generic ILP and SAT solvers and enabled them to handle a wider range of engineering problems. While generic ILP Solvers have been applied to solving ILP models of several real-life optimization problems, comparatively few attempts have been made using SAT solvers. One such problem is the ‘clustering problem’ in Mobile Ad-Hoc Networks (MANETs). MANETs are used in wide-ranging applications such as battlefield communication, law enforcement operations and disaster recovery. The proposed solution to the scalability issue in ‘flat’ MANET networks is the concept of ‘clustering’ or the creation of a hierarchical network where the network is divided into clusters with certain nodes in each cluster being chosen to be ‘clusterheads’. The process of selecting which nodes would be best suited to be clusterheads and which regular nodes should be assigned (connected) to which clusterhead is known as the ‘clustering problem’. The clustering problem can be modeled as an optimization problem in ILP.

Our research aims to examine the capabilities of current state of the art Generic ILP and SAT solvers when used to solve formulations of the clustering problem. This consists of three distinct parts. The first is the creation of enhanced ILP models of the ‘clustering problem’ to be able to generate topologies of varying characteristics. The second is the design of a custom tool to create topologies and convert them into ILP formulations for different requirements (different models). Additionally, the tool should be able to integrate with a selected set of solvers to solve the generated problems and to be able to display the solutions produced, all through an intuitive Graphical User Interface (GUI). The third is the testing of the selected set of the state of the art commercial and non-commercial Generic ILP and 0-1SAT-based ILP solvers in solving these different models of the clustering problem through the tool created.

1.1 BACKGROUND

In this section, a better understanding of the foundations of the research and the context of the research will be provided. Section 1.1.1 will describe Integer Linear Programming (ILP) and its use in real world applications. It will also describe the use of Generic ILP solvers and Boolean Satisfiability (SAT) solvers, the similarities and differences between them and will also touch on the improvements in the performance of solvers over the years. Section 1.1.2 will outline Mobile Ad-Hoc Networks (MANETs) and present the “clustering problem”. Section 1.1.3 will begin to present the use of ILP formulation to model the Clustering Problem in MANETs. Section 1.1.4 will detail the focus of our research.

1.1.1 INTEGER LINEAR PROGRAMMING: GENERIC ILP AND SAT

This section will describe the basics of ‘Linear Programming’ and its applications. Linear Programming involves maximizing or minimizing a function with respect to certain restrictions or constraints where the functions and constraints are linear. Integer Linear Programming (ILP) is the area of linear programming where the variables in the linear function to be maximized or minimized and its constraints can only take integer values [1]. ILP can be divided into generic ILP and 0-1 ILP where the variables take only binary values. In Boolean Satisfiability (SAT), given a formula f , the objective is to identify an assignment to a set of Boolean variables that will satisfy a set of constraints. The difference between ILP and SAT is that in ILP the constraints consist of mathematical equations, whereas in Boolean Satisfiability the constraints between variables are represented using what is called propositional logic where the AND, OR and NOT operations are used to construct formulas in the Products-of-Sums form (also called the Conjunctive Normal Form (CNF)). The variables can only take Boolean values (0 or 1) and the CNF formulas may evaluate to either 0 or 1. A satisfying assignment of variables is such that when the values of the variables are substituted into the formula, it evaluates to 1 at the same time satisfying all constraints for that formula. Given such a problem, the goal is to either find a satisfying variable assignment or prove that none exists. Using SAT to determine whether a satisfying assignment exists is an NP complete problem which is solvable but with runtimes growing exponentially in the worst case scenarios [2]. There are several applications which can be modeled mathematically using ILP and SAT. When modeled mathematically, some of these applications have potentially conflicting constraints which need to be satisfied. In order to

‘solve’ such models we have to determine the values of variables in accordance with these constraints.

Figure 1 shows the search space of a sample propositional logic formula. Given n variables there are 2^n different possible variable assignments. In order to ‘solve’ or rather ‘satisfy’ the formula shown above the tree, Boolean Satisfiability will go through this search space (using intelligent search techniques) and determine whether there is a satisfying variable assignment (shown as green in Figure 1) or prove that no satisfying assignments exist.

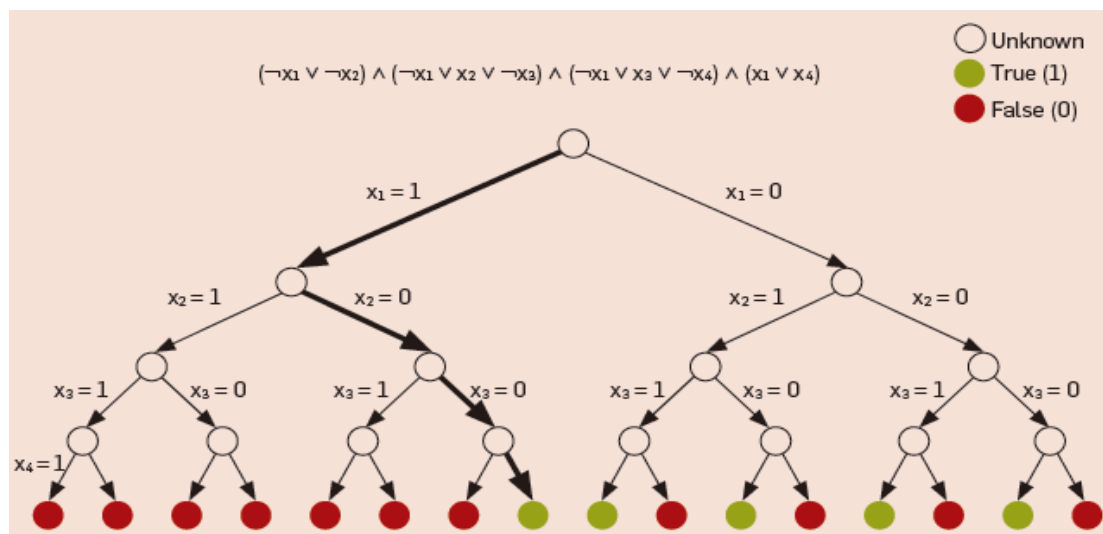


Figure 1: SAT Problem Showing 2^n Different Possible Variable Assignments [2].

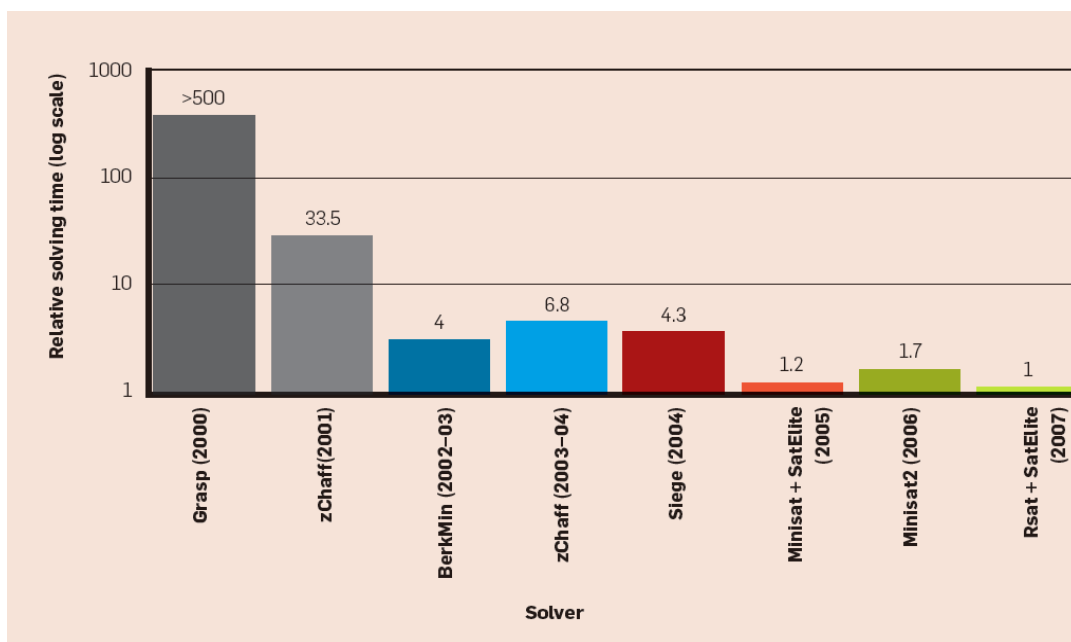


Figure 2: Improvement in SAT Solver Performance Over the Last Decade [2].

Over the last decade, there has been significant improvement in SAT/ILP Solver capabilities and performance as shown in Figure 2. Research in the area has been primarily along two lines.

The first is the development of new solving techniques or improvement in existing solving techniques in order to create new solvers or enhance the performance of existing solvers. Much research has been directed at creating and improving generic ILP and 0-1 SAT solvers and also at determining which technique/solver is better or faster at solving specific instances or problems. (Throughout this thesis 0-1 SAT based ILP solvers, will be referred to as ‘SAT’ solvers) One particular improvement or extension in SAT solver capability has been the ability to solve problems which have Pseudo-Boolean constraints [3]. Several contests are held to assess the performance of SAT Solvers [4]. The SAT solver contest has been running regularly since 2002 although the first one was held a decade earlier in 1991/1992 [6] and the contest focused on Pseudo-Boolean constraint based problems [5] has been running regularly since 2005 [7]. Studies have shown that SAT solvers can compete with generic ILP solvers in solving 0–1 ILP problems arising in specific applications [3, 8, 9].

The second area of research has been the formulation of ILP/SAT equivalents of complex problems in different areas and determining how effectively they can be solved using ILP/SAT Techniques. Improvements in ILP/SAT solving techniques, as well as the availability of increasingly affordable high computational power, have resulted in several challenging engineering problems being modeled using ILP and/or SAT. Some of them are listed below:

- FPGA [10]
- Network Intrusion [11, 12]
- Access Control [13]
- Cryptography [14]
- Software verification and debugging [15]
- Application Mapping [16]
- Genetics [17]
- Scheduling [18, 19]
- Optimization of power consumption [9]

1.1.2 MANETS: THE CLUSTERING PROBLEM

In this section, Mobile Ad-Hoc Networks (MANETs) will be introduced, the challenges involved in setting up a MANET topology will be discussed, and the clustering problem will be presented.

MANETs are wireless, self-organizing networks consisting of mobile nodes with generally a limited supply/store of energy. These nodes can be for example, laptops, mobile radio terminals or other devices, generally those which are used by humans [20]. MANETs are used in wide-ranging applications such as battlefield communication, law enforcement operations, and disaster recovery [21]. There are several challenges faced in enabling MANETs to communicate through a stable, scalable, and flexible topology. Over the years much research has been undertaken in enabling MANETS to operate in the optimum state, i.e. minimizing energy consumption and essentially attempting to achieve the maximum network lifetime. This research has focused on many different challenges in MANETs such as cluster formation, routing and communication.

Initially MANET topologies were ‘flat’ networks or non-hierarchical networks where all nodes had identical roles. Through various tests and simulations conducted it was proven that as the number of nodes in flat networks increases the throughput falls drastically [22]. In addition several factors such as frequent route breakage, unpredictable topology changes, routing overhead make it difficult for a ‘flat’ topology to be scalable [23].

The proposed solution to the scalability was the concept of ‘clustering’ or a hierarchical network, where the network is divided into clusters with certain nodes in each cluster being chosen to be ‘clusterheads’. This is similar to concept of an IP Subnet and results in reduced control overhead [24]. The clusterheads have the responsibility of managing communication and routing for their particular cluster and because of this the selection of clusterheads is particularly important [25].

Selection of clusterheads is not trivial. There are several issues that need to be considered when selecting clusterheads, one of which is that the clusterheads are not selected for the lifetime of the network but rather are re-selected and the topology is re-generated at certain intervals. This is because of the fact that since clusterheads are responsible for routing and communication they use more energy than regular nodes and so if they remain clusterheads they will be the first nodes to be depleted. In order to maximize network lifetime, the responsibility of being a clusterhead is rotated between nodes. Another reason for re-clustering is that since the nodes are mobile, some nodes may move out of range of one

clusterhead and in range of another and so the topology must adjust accordingly. Another factor to consider when clustering a network is the nature of the ‘backbone’ formed by connecting clusterheads. Factors such as redundancy must be considered and implemented without adding unmanageable or expensive overhead. Another area that has been researched is the optimal ‘cluster size’ that a clusterhead can handle.

One can think of the MANET network formation problem in three parts:

- Clusterhead Selection
- Cluster maintenance
- Re-election and reformation.

Numerous algorithms have been proposed for different aspects of the clustering problem with different unique approaches put forward with different areas of focus. Several algorithms take into account a combination of parameters or multiple metrics, such as mobility, residual energy, when electing clusterheads [23, 26]. Certain algorithms focus more on optimizing routing in a clustered network [27, 28, 29]. Other algorithms focus on cluster ‘stability’ which can be defined as the time for which a cluster structure is constant [24, 30]. Some algorithms implement a 1-hop topology [21] where each node is connected to its clusterhead directly, while others implement Multihop topologies where nodes are not connected directly to the clusterhead but through intermediate nodes [31, 32].

1.1.3 MODELING ‘THE CLUSTERING PROBLEM’

This section will touch briefly on the different ways of modeling the clustering problem. The authors in [33] point out that: *“Different clustering algorithms have different optimizations, such as minimum clusterhead election and maintenance overhead, maximum cluster stability, maximum node lifespan, etc. There are probably contradictions among these optimizations. In addition, lots of the optimizations and their combinations are an NP-hard problem. Thus, heuristic clustering algorithms are used to find sub-optimal solutions in common.”*

The key points here are as follows.

- Heuristic algorithms aim to obtain ‘sub-optimal’ solutions that are as close as possible to optimum solution.

- A lot of optimizations and combinations are ‘NP-hard’. NP-hard refer to problems that are non-deterministic polynomial-time hard problems. While these problems can be solved by an ‘exhaustive’ search, i.e. searching through all possible combinations, determining if a solutions exists and if necessary going through all possible solutions and determining the optimum solution. The time taken to do this search increases exponentially as the size of the problem increases. This exponential increase in solving time is one of the key reasons why heuristic algorithms are preferred.

The advantage is using Integer Linear Programming (ILP) or Boolean Satisfiability (SAT) Solvers is that for such a problem they provide the best possible solution, (i.e. the optimum solution). However, as mentioned, since the problem is NP-complete, the complexity increases exponentially with an increase in the number of variables (in case of the clustering problem this would be as a result of an increase in the size of the network).

1.1.4 RESEARCH FOCUS

Integer Linear Programming (ILP) formulations of optimization problems in Mobile Adhoc Networks (MANETs) are few and have attempted to target different areas such as the clustering problem [23, 34], energy efficient routing problems, broadcast/multicast routing problems, data extraction and gathering problems and QoS topology control [35].

To our knowledge there are no papers which attempt to formulate the clustering problem as a Boolean Satisfiability (SAT) problem and solve it using a SAT solver. As already mentioned, there has been a significant improvement in both SAT and ILP solvers over the last decade and the results of using solvers to solve the clustering problem would yield significantly improved results compared to earlier efforts.

Our research is directed at assessing the current state of the art ILP and 0-1 SAT-based ILP Solvers (which we will be referring to as SAT solvers) in solving an ILP formulation of the clustering problem. As will be explained in the literature review, the work put forward by the authors in [34] is one of the initial efforts at solving the clustering problem in MANETs by mathematical modeling and solving the generated models using ILP Solvers. Our research will aim to build primarily on the work put forward by the authors in [34], by extending/improving the ILP model of the clustering problem developed in [34], and then comparing the performance of different generic ILP and SAT Solvers as they solve the newer/improved models. Our research will also include the design of an improved tool

compared to the one developed in [34], in order to formulate ILP problems for different solvers when given a visual topology. The proposed tool would also be able to display as a connected network the solution provided by the selected solvers. The improvements/enhancements and the need for them, in both the tool and the model of the clustering problem will be discussed in depth in the subsequent sections of this thesis.

1.2 LITERATURE REVIEW

As mentioned earlier, the application of Integer Linear Programming (ILP) in modeling and solving optimization problems in Mobile Ad-Hoc Networks has been limited. In this section significant contributions made in ILP formulations in optimization problems in MANETS will be described in detail.

In 2004, the authors, in [23], compared the performance of a clustering mechanism called Virtual Grid Architecture (VGA) with an ILP formulation aimed at minimizing the set of connected clusterheads (finding the minimum connected dominating set). The goal of VGA was “to create a fixed rectilinear virtual topology on which routing and network management functions can be performed easily and efficiently” [23]. The authors focused on network management, i.e. identifying the set of clusterheads. VGA consisted of three parts; ‘Zoning’ whereby the area is divided into rectangular zones/clusters, ‘clusterhead selection’ where the algorithm to select clusterheads is executed in each zone and last is the ‘Routing’ which can be restricted to horizontal/vertical routing or enhanced by using Diagonal –VGA (D-VGA) and enabling diagonal routing. The topologies generated were 1-hop and due to the capabilities of the solver, the network size was limited to 30 nodes when comparing VGA to ILP. The ILP formulation for their 30 node network took 1011.5 seconds (~17 minutes).

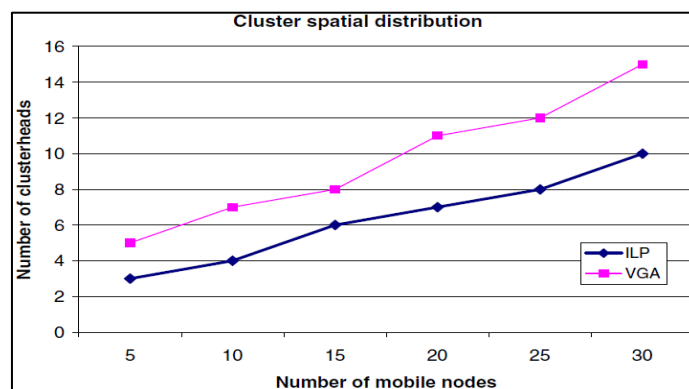


Figure 3: Performance Comparison of ILP Model of the Clustering Problem and Virtual Grid Architecture [23].

This was the first attempt, to the best of our knowledge, at using ILP to model a version of the clustering problem that didn't aim to find which nodes were best suited to being clusterheads, but rather it aimed to find the minimum number of clusterheads. As shown in Figure 3, when using ILP formulations, topologies with a fewer number of clusterheads were generated as compared to the VGA algorithm.

In 2005, the authors of [35] put forward ILP formulations to tackle various challenges in MANETS and sensor networks. The areas of focus were energy efficient routing, data extraction and gathering, and (Quality of Service) QOS topology control. Although the clustering problem was not one of the problems addressed, the work put forward by the authors in [35] provided an understanding of the potential of ILP and how within a single field such as MANETS, ILP formulation can be applied to so many areas.

While the work put forward by the authors in [23] could be considered the first attempt at using ILP formulation in relation to the clustering problem, the first truly significant attempt at applying ILP formulation to the clustering problem was the work put forward by the authors in [34] in 2006. Unlike the model presented in [23], the authors did not focus on obtaining the minimum number of clusterheads but rather the selection of the most suitable nodes to be clusterheads such that network lifetime was maximized.

The authors modelled the clustering problem in three different ways. Looking at MANETS, the authors classified topologies into three categories:

- EEC-FCB (Energy Efficient Clustering - Fully Connected Backbone)
- EEC- CB (Energy Efficient Clustering - Connected Backbone)
- EEC-R (Energy Efficient Clustering – Redundancy)

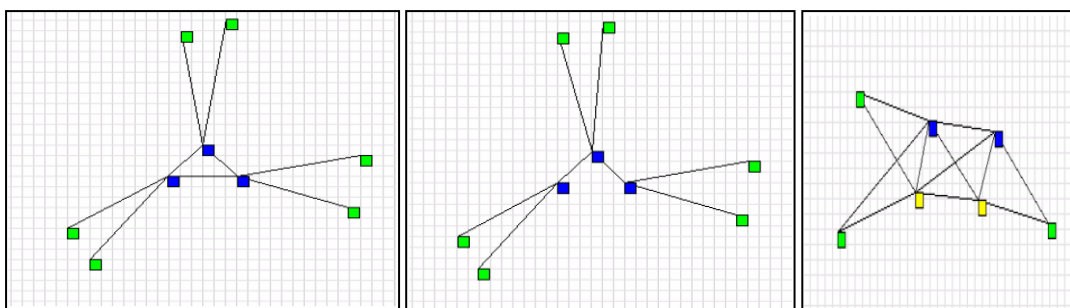


Figure 4: Different Network Topologies: a) Fully Connected Backbone, b) Connected Backbone, and c) Redundant Models [34].

The three different topologies are shown in Figure 4. The first case has a fully connected backbone (FCB). In this case all the clusterheads are connected to each other in a

mesh topology. The second case is that of a connected backbone (CB). In this case all clusterheads are not connected to each other but are all connected to one particular clusterhead. In the third case, the concept of a ‘Backup’ clusterhead is introduced in order to address redundancy concerns. The backup clusterhead takes over the role of the clusterhead in case the clusterhead fails. A fully connected backbone is assumed with clusterheads and backup clusterheads being connected in a mesh topology.

An ILP model was devised for each of these cases. The authors believed this to be the first case of applying ILP to the network clustering problem. An in-depth look at the model(s) is provided in the later sections. Also the ILP models could not be tested for networks greater than 9 nodes as the solver used in [34] could only handle small scale networks; up to 9 nodes. The authors do however note that ILP formulation presented in [34], was purely theoretical and not for use in a practical environment. It is important to note here that extensive evaluation was not conducted.

The results obtained were not compared to other clustering algorithms such as those mentioned earlier (non ILP based models). They did compare it to an earlier work of theirs: fuzzy based hierarchical energy efficient routing protocol (FEER). The results obtained are a comparison of the cost of the solution produced for all three models. This is shown in Figure 5. What is clear is that compared to FEER, the clustering algorithms produce low ‘cost’ solutions.

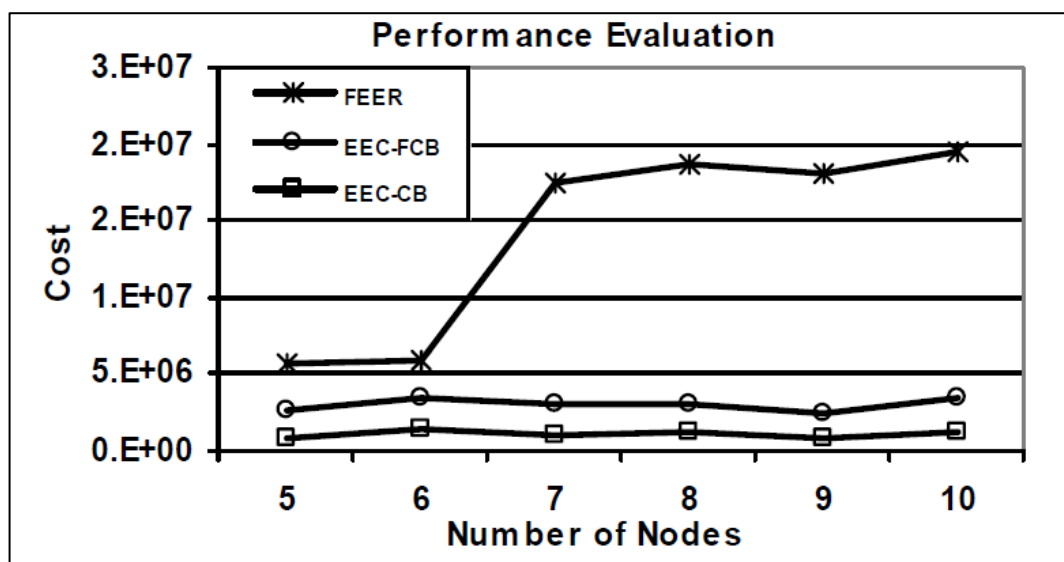


Figure 5: Performance Comparison of ILP Models (Fully Connected Backbone and Connected Backbone) with non ILP models (FEER) [34].

Additionally, the authors in [34] created a JAVA-based tool which allowed users to create topologies visually on a grid. These topologies could then be formulated as ILP instances. The optimum selection of clusterheads and connections between nodes could be determined by solving the ILP problem through the solver integrated with the tool. The solution of the ILP Problem could then be displayed on the grid using the tool.

While attempts to use ILP formulation in optimization problems have been limited, there have been applications of ILP formulations in optimization problems in Wireless Sensor Networks (WSNs). WSNs can be considered to be a special case of MANETs. Although there are several similarities between MANETs and WSNs there are significant differences between them as well, and these are explored in detail in [20].

WSNs are networks which consist of numerous ‘nodes’ which have similar restrictions to MANETs, such as limited resources (energy reserves), and the need to make the most of them. Both are self-organizing wireless networks with wide-ranging applications. However, the nature of the nodes and the applications differ. Some of the key differences, especially when considered in the context of ILP formulation are described below.

Base Stations are present in a WSN topology. In MANETs, there are no Base Stations. Nodes in a MANET communicate with each other and exchange information/data with each other. In a WSN, nodes have a specific objective. The objective of these nodes is to sense phenomena occurring within their coverage radius and to transmit the data gathered to a central point known as the ‘sink’ as shown in Figure 6. They use each other to transmit the data they have gathered back to the sink or Base Station. This has to be taken into account when formulating an ILP model.

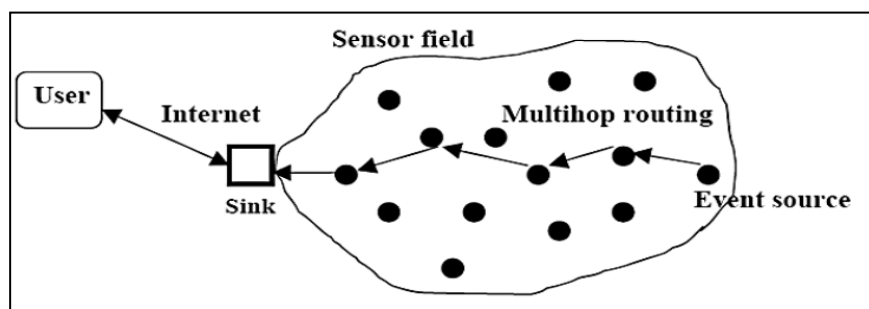


Figure 6: Wireless Sensor Network Layout: From the Sensor to the End-User [20].

There is a difference in the size of MANETs and WSNs. WSNs are larger in scale, generally consisting of a lot more nodes as compared to MANETs. There is also a difference in the extent of mobility between WSNs and MANETs. Wireless Sensor networks are

relatively less mobile. The authors, in [20], state that although there are some cases where the nodes are attached to mobile objects such as automobiles or buoys, the majority of the applications use a WSN with static nodes that may have, for example, been deployed from a plane.

In several environments where WSNs are deployed there is increased likelihood of nodes going down, failing and necessitating a network reconfiguration. Due to the need to save energy and also because sometimes multiple nodes may overlap in the areas they cover, some nodes are put to sleep while others are awake. As this sleep-wake cycle puts some nodes to sleep and wakes others up, network reconfiguration is required.

Keeping these similarities and differences in mind there have been attempts to create ILP formulations of optimization problems in WSNs. The authors in [36], extended and strengthened an ILP based model for optimizing the energy consumption in wireless sensor networks. The model aimed to minimize total energy consumption, penalizing any areas lacking coverage as well as penalizing unnecessary activation of nodes. This was done through the concept of ‘demand points’ which were areas that require coverage and had to be covered by at least one sensor at all times. The assumption was made that the area where the sensors are deployed was plain with no obstacles. The data being generated at the demand point had to be ‘sensed’ and routed back to the sink.

The ILP model presented in [36] was tested using a grid of 10 by 10 meters, with 100 demand points in place as shown in Figure 7. Sixteen nodes were deployed throughout the area. Both the demand points and the sensors were deployed using one of two types of sensor placements; grid based and randomly generated positions.

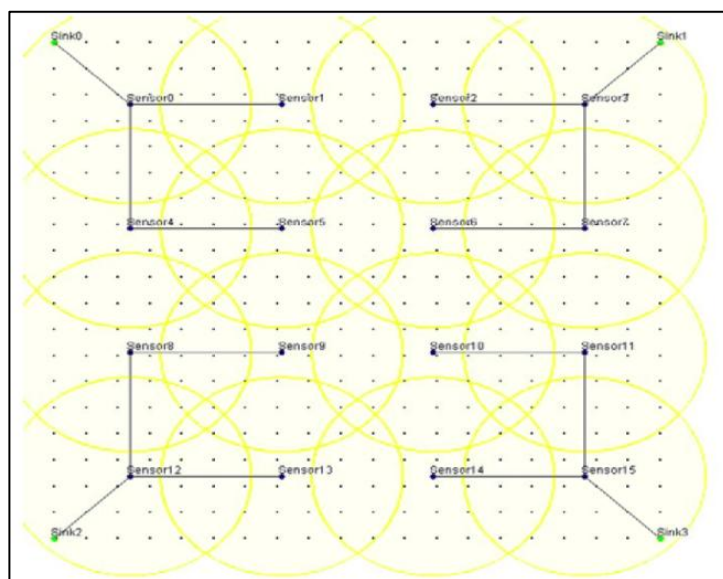


Figure 7: Routing in a Grid of Sensors and Demand Points [36].

The CPLEX [37] solver was used to solve and test the effectiveness of the model. Testing indicates that in grid-position based instances, no demand points were left uncovered but in random-position based instances some demand points were left uncovered. Additionally, although the model could handle “situations found in real WSNs”, it was limited in size by the exponentially increasing nature of the problem. However, an interesting idea put forward by the authors was the use of a hybridized model combining ILP with Genetic Algorithms (GA).

The authors, in [38], presented an ILP formulation aimed at implementing an energy optimal topology that maximized network lifetime while simultaneously ensuring full area coverage and sensor connectivity to clusterheads. In the proposed model, any node could be active, switched off or selected as a clusterhead and only clusterheads were allocated the task of routing data.

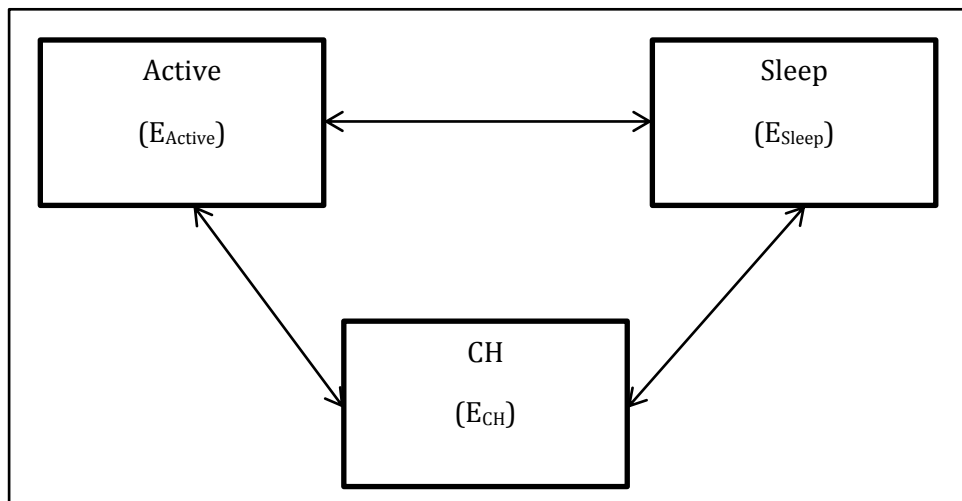


Figure 8: Different Possible Sensor States: Active, Sleep and Clusterhead [38].

The factors that determined whether a node could become a clusterhead or a regular node were the node’s residual energy, the distance of the node from non-clusterhead neighboring nodes and also the position of the node in the clusterhead backbone. Since in this mechanism there was the possibility of having non-active sensors (sensors switched off as shown in Figure 8), there was a clear trade-off between energy consumption and coverage.

The ILP model that was formulated was solved using a proposed TABU search heuristic called TABU-RCC. TABU is a search technique that aims to find a solution close to the optimal for combinatorial problems where finding the optimal solution is extremely difficult [39]. TABU was to be run periodically to reconfigure the network after a pre-defined time period T . This required all the information to be present at the Base Station or PN

(Processing Node). Simulations were run, where the ILP model was solved using CPLEX and the performance of the proposed TABU-RCC search heuristic was compared against an existing TABU search algorithm called EESH. The performance of the model was evaluated for different cluster sizes and sensing ranges.

Simulations showed that although centralized, acceptable results within low computational times were obtained and the authors believed it to be feasible for practical implementation. The difference between the costs obtained using ILP formulation and using the search heuristic are shown in Figure 9. As can be seen, the differences are small and hence the search heuristic TABU-RCC was preferred to CPLEX because of the low computation times associated with TABU-RCC.

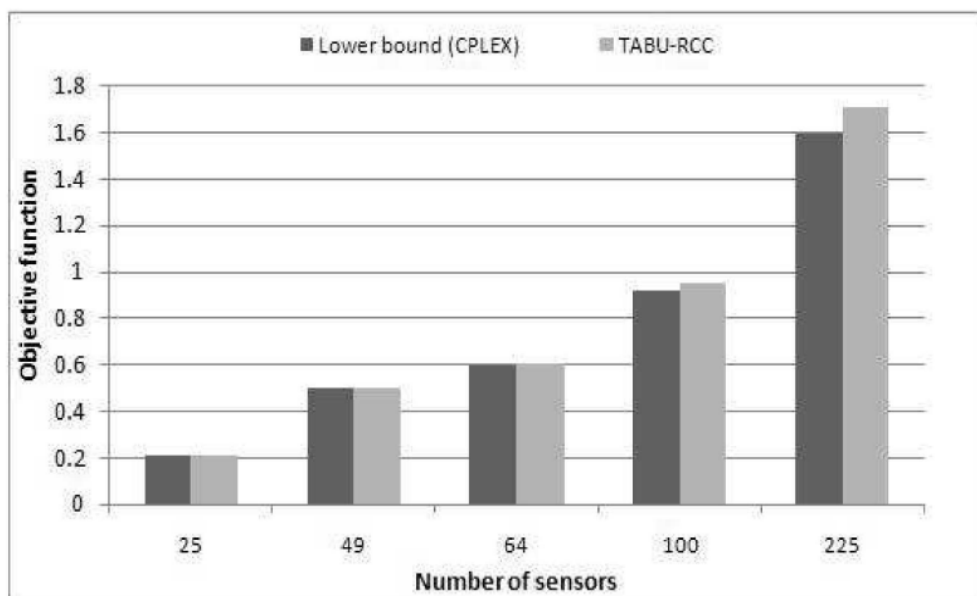


Figure 9: Cost of Solutions Generated using ILP model vs Search Heuristic (TABU-RCC) for Energy-Optimal Model of Wireless Sensor Networks [38].

This section covered the key contributions in the area of ILP formulations, mainly of the clustering problem, in MANETs and WSNs. CPLEX was the preferred solver used to solve the ILP formulations. As mentioned earlier, the most significant attempt to formulate the clustering problem in MANETs as an ILP formulation, was the work put forward by the authors in [34]. The model and the tool, presented in [34], create a solid foundation to use to begin the construction of an improved model and tool. Additionally, it is possible to compare the performance of newer solvers such as CPLEX as well as Pseudo-Boolean SAT Solvers using the model put forward in [34].

1.3 RESEARCH METHODOLOGY

This section will describe the research methodology followed and the organization and structure of the thesis into phases.

The first phase of research is the ‘Investigation Phase’. This will consist of detailed analysis of the existing model, tool and solvers. The outcome of this phase will include the identification of the constraints, assumptions and capabilities of the Integer Linear Programming (ILP) models and tool presented in [34] which can and should be preserved. It will also include identification of the potential enhancements to the model and tool presented in [34] which should be included in the proposed ILP formulation and proposed tool. Also, the preferred set of effective and available solvers to be used in conjunction with the proposed tool and ILP models will be identified.

The second phase of the research is the ‘design and implementation phase’. This phase will involve the design and implementation of the proposed ILP formulation of the clustering problem and the proposed tool. The tool will be designed to generate the ILP formulation of the clustering problem of a custom topology created by the end user. The generated ILP formulation should be in the format required by the selected set of solvers. This will enable analysis of results produced by different solvers at different stages of modelling. These results will be examined both mathematically and visually through the tool. This will aid in identifying flaws in the model, programming errors in the tool as well as identifying and correcting discrepancies between the mathematical results produced by the solver and the visual representation of the results produced by the tool.

The third phase of the research is the testing phase which will consist of establishing a testing procedure and carrying out detailed tests using the proposed ILP formulation, proposed tool and selected set of solvers. A set of ILP formulations of the clustering problem for a varied set of topologies will be generated using the tool and solved using the selected set of solvers.

The fourth phase of the research is the ‘evaluation phase’ which will involve detailed analysis of the results obtained from the testing phase. The outcome of this phase will include identification of the fastest solvers from the selected set of solvers for each of the different ILP models used in testing. In addition, trends and dependencies between the proposed ILP formulations and the performance of the selected set of solvers will be identified and areas of improvements and further testing will be discussed. Based on this detailed analysis, the

feasibility of using the proposed ILP formulations to solve clustering problems in MANETs in a practical environment will be examined.

This thesis is structured as follows: Chapter 2 will describe the investigation phase. The design and implementation phase will be detailed in Chapter 3 of this thesis. Chapter 4 will describe the testing phase. The evaluation phase will be covered by Chapter 5. Chapter 6 will summarize and conclude the thesis. Chapter 7 will detail the areas of future work in the three key areas; ILP formulation of the clustering problem, design and implementation of the tool and solver testing and performance.

CHAPTER 2

INVESTIGATION

In this chapter, the model and tool presented by the authors in [34] will be examined, and the set of solvers to be used in the research will be selected. Section 2.1 will describe the model presented by the authors in [34], its strengths and limitations and the potential areas of improvement. Section 2.2 will detail the current state of the art solvers, both generic Integer Linear Programming (ILP) and Boolean Satisfiability (SAT) solvers, and will list the solvers selected for the research as well as the criteria used to select them. Section 2.3 will describe the strengths and limitations of the tool designed by the authors in [34], and identify the potential areas of improvement as well as the new features and design requirements.

2.1 EXISTING MODEL ANALYSIS

In this section, the models presented by the authors in [34] will be examined in detail. In [34], the authors formulate Integer Linear Programming (ILP) models for three cases. These models are the first significant effort at modeling the clustering problem using ILP techniques.

- The first case is the Energy Efficient Clustering (with a) Fully Connected Backbone Model (EEC-FCB). In this case all the clusterheads are connected to each other, forming a mesh backbone.
- The second case is the Energy Efficient Clustering (with a) Connected Backbone Model (EEC-CB). In this case all clusterheads are connected to one central Master clusterhead. The number of connections within the backbone is significantly less than the FCB.
- The third case is the Energy Efficient Clustering (with) Redundancy (EEC-R) Model. In this case there is improved reliability through the use of a backup clusterhead, with one backup clusterhead assigned to each regular clusterhead.

The EEC-FCB case produces too many redundant links, through its mesh backbone, and the EEC-CB case has a single point of failure, which is the Master clusterhead. If the Master clusterhead fails then each clusterhead will be isolated from all others. Our goal, in

this work, is to find a middle ground between these two, where the level of redundancy is higher than the EEC-CB case but not overly infused with redundant links such as the EEC-FCB model.

2.1.1 EXISTING ILP FORMULATION OF THE CLUSTERING PROBLEM

In this section, the functions and constraints which are used to formulate the Energy Efficient Clustering (with a) Fully Connected Backbone (EEC-FCB) model and Energy Efficient Clustering (with a) Connected Backbone (EEC-CB) model are analyzed. The authors in [34] construct the ILP formulations for these models (EEC-FCB and EEC-CB) with the the goal of maximizing network lifetime, i.e. the selection of nodes to be clusterheads (CHs), the creation of links between regular nodes and clusterheads, and the creation of backbone links between clusterheads should be such that the network is able to operate for the longest possible time before a node runs out of power.

The authors in [34], make the assumption that the radio used by the nodes in Mobile Ad-hoc Network (MANET) is based on the power-attenuation model where *“The signal power falls as $1/r^k$, where r is the distance between the transmitter/receiver nodes and k is a real constant dependent on the wireless environment, typically between 2 and 4. In our case, we set $k = 2$ for communication between normal nodes and CHs, and $k = 3$ for communication between CHs.”* [34]

Given below are the variables used in the EEC-FCB and EEC-CB models presented in [34].

- N : Total number of nodes in the network-predetermined
- P : Number of clusterheads (CHs) – predetermined
- d_{ij} : Euclidean distance between nodes i and j
- K_j : Max nodes that can be connected to CH- j – predetermined
- c_{ij} : cost of connecting a regular node i to CH j (proportional to d_{ij}^2)
- h_{jk} : Cost of connecting CH j to CH k (proportional to d_{jk}^3)
- b_j : Weight associated with CH j . The authors in [34] describe b_j as follows: *“ b_j is the output of the fuzzy logic controller discussed in [27]. The higher the value of b , the better the node is. Since the objective function is a minimization function, each value in the b array is multiplied by -1.”*
- $x_{i,j} = \begin{cases} 1, & \text{if node } i \text{ is connected to clusterhead } j \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$

- $y_j = \begin{cases} 1, & \text{if node } j \text{ is selected to be a clusterhead} \\ 0, & \text{otherwise} \end{cases} \quad (2.2)$

- $w_{i,j} = \begin{cases} 1, & \text{if } x_{i,j} \text{ and } y_j \text{ are both equal to 1.} \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$

- $z_{j,k} = \begin{cases} 1, & \text{if nodes } j \text{ and } k \text{ are clusterheads and interconnected} \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$

2.1.1.1 THE EEC-FCB MODEL

In the EEC-FCB case presented by the authors in [34], as shown in Figure 10, the backbone forms a mesh, with each clusterhead connected to all other clusterheads.

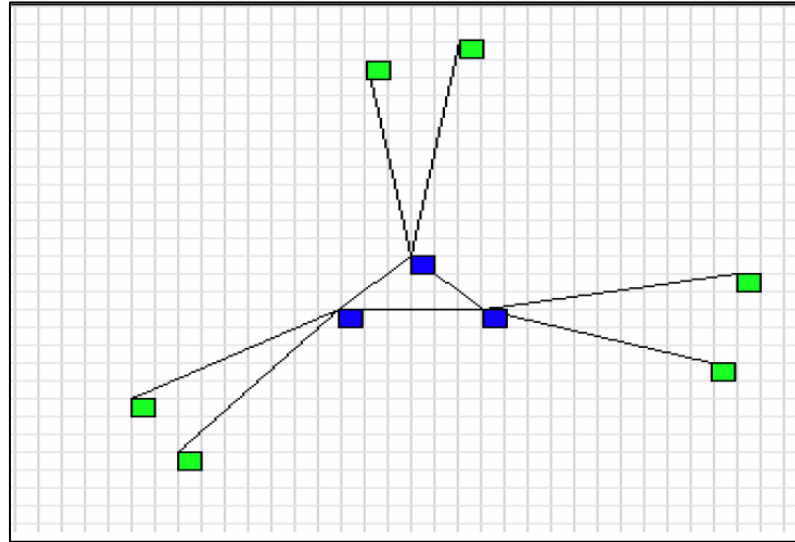


Figure 10: Fully Connected Backbone Topology [34].

The objective function to be minimized for the EEC-FCB model is:

$$\text{Min}(x, y, z): \left(\sum_{i=1}^N \sum_{j=1}^N c_{i,j} x_{i,j} + \sum_{j=1}^N b_j y_j + \sum_{j=1}^N \sum_{k=1}^N (h_{j,k} - c_{j,k}) z_{j,k} \right) \quad (2.4)$$

The first term of the objective function represents the connections between nodes and clusterheads. The second term represents the selection of nodes to be clusterheads. The last term represents the connections between clusterheads. The objective function aims to minimize the cost of connections, or links, between nodes, thereby minimizing the cost of sending/receiving data along these connections.

Constraint 2.5 is used to define the backbone. If there are P clusterheads and a node is selected to be a clusterhead then it must connect to $(P-1)$ other clusterheads in order to implement the mesh backbone.

$$\sum_{i=1}^N x_{i,j} \geq (P-1)y_j + (1-y_j) \quad \forall j \quad (2.5)$$

Constraint 2.6 indicates that if node j is chosen to be a CH, it should be connected to at most $(P-1)$ CHs and K_j regular nodes.

$$\sum_{i=1}^N x_{i,j} \leq (K_j + P - 1)y_j + (1 - y_j) \quad \forall j \quad (2.6)$$

Constraint 2.7 is used to specify the total number of connections which must be present in the generated topology. This is made up of the mesh connections for the backbone, and the connections of each regular node to a selected clusterhead.

$$\sum_{i=1}^N \sum_{j=i+1}^N x_{i,j} = (N - P) + P(P - 1)/2 \quad (2.7)$$

Constraint 2.8 indicates that the total number of clusterheads is P .

$$\sum_{j=1}^N y_j = P \quad (2.8)$$

Constraints 2.9 and 2.10 are used to specify that the connections that make up the backbone of the network are only the connections which interconnect the clusterheads.

$$\sum_{\substack{j=1 \\ j \neq k}}^N z_{j,k} = (P-1)y_j \quad (2.9)$$

$$\sum_{\substack{k=1 \\ j \neq k}}^N z_{j,k} = (P-1)y_k \quad (2.10)$$

Constraint 2.11 is used to enforce the restriction that a node cannot be connected to itself.

Constraint 2.12 is used to indicate that the matrix of connections between nodes is symmetric and diagonal, i.e. If node 1 is connected to node 2, that implies that node 2 is also connected to node 1.

$$\sum_{i=1}^N x_{ii} = 0 \quad (2.11)$$

$$x_{i,j} = x_{j,i} \quad \forall i, \forall j \quad (2.12)$$

Constraints 2.13, 2.14 and 2.15 state that w_{ij} is 1 only if x_{ij} is 1 and y_j is 1.

$$w_{i,j} \leq x_{i,j} \quad \forall i \forall j \quad (2.13)$$

$$w_{i,j} \leq y_j \quad \forall i \forall j \quad (2.14)$$

$$w_{i,j} \geq x_{i,j} + y_j - 1 \quad \forall i \forall j \quad (2.15)$$

Constraints 2.16 and 2.17 are used to ensure that regular nodes are connected only to clusterheads and not to regular nodes.

$$\sum_{i=1}^N w_{i,j} \geq 1 \quad \forall j \quad (2.16)$$

$$\sum_{i=1}^N w_{i,j} \leq (P - 1)y_j + (1 - y_j) \quad \forall j \quad (2.17)$$

Constraints 2.18, 2.19 and 2.20 are used to indicate that variables $x_{i,j}$, y_j and $z_{j,k}$ have binary values.

$$x_{i,j} \in \{0,1\} \quad \forall i \forall j \quad (2.18)$$

$$y_j \in \{0,1\} \quad \forall j \quad (2.19)$$

$$z_{j,k} \in \{0,1\} \quad \forall j \forall k \quad (2.20)$$

2.1.1.2 LIMITATIONS OF THE EEC-FCB MODEL

The following are the assumptions made by the authors in [34] when formulating the Energy Efficient Clustering-Fully Connected Backbone (EEC-FCB) model, and the limitations of the EEC-FCB model.

- Coverage is not considered. It is assumed that all nodes can communicate with each other.
- Localization is not considered. It is assumed that nodes are able to determine each other's position, either through the use of a Global Position System (GPS), or other localization techniques.
- The variable b_j gets its value from an external source (algorithm, tool, etc). This is useful as multiple approaches/algorithms, which determine the suitability of a node in acting as a clusterhead, can be combined with this model without changing the equations, although this is out of the scope of our research.
- All connections are single hop, i.e. All nodes are directly connected to a clusterhead.
- The mesh backbone provides a high number of redundant links between clusterheads.
- There is no direct communication between nodes in the same cluster. Communication between nodes in the same cluster also goes through the clusterhead.

2.1.1.3 THE EEC-CB MODEL

In the Energy Efficient Clustering – Connected Backbone (EEC-CB) model presented by the authors in [34], the network backbone, which consists of the clusterheads, is not fully connected. In order to maintain network connectivity, while not having a fully connected backbone, the concept of a Master clusterhead (MCH) is introduced. All clusterheads are connected to one clusterhead which is designated to be the Master clusterhead. This is illustrated by Figure 11.

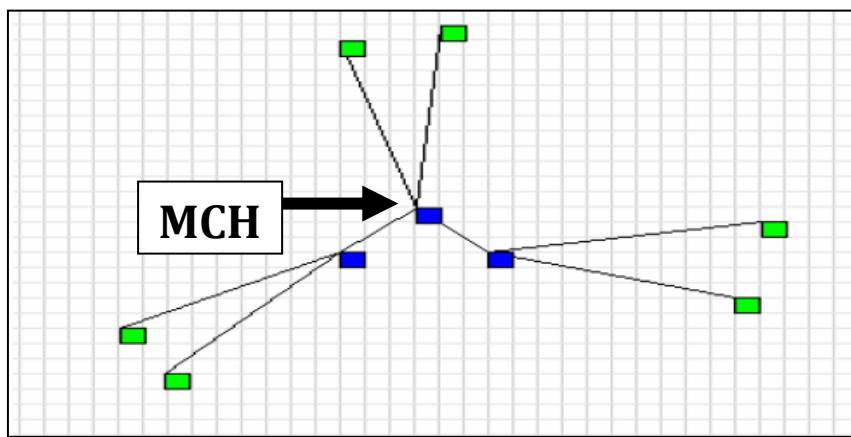


Figure 11: Connected Backbone Topology Using a Master Clusterhead [34].

The objective function to be minimized is:

$$\text{Min } (x, y, z): \left(\sum_{i=1}^N \sum_{j=1}^N c_{i,j} x_{i,j} + \sum_{j=1}^N b_j y_j + \sum_{j=1}^N \sum_{k=1}^N h_{j,k} z_{j,k} \right) \quad (2.21)$$

In the case of EEC-CB the first term of the objective function represents the connections between nodes and clusterheads. The second term represents the selection of nodes to be clusterheads. The last term represents the connections between clusterheads.

Constraints 2.22 and 2.23 state that the variable M_j can take only values from 0 and 1 and that a node may be a Master clusterhead only if it is also a regular clusterhead.

$$M_j \leq y_j \quad \forall j \quad (2.22)$$

$$M_j \in \{0,1\} \quad \forall j \quad (2.23)$$

Constraint 2.24 states that there can be only one MCH.

$$\sum_{j=1}^N M_j = 1 \quad (2.24)$$

Constraint 2.25 specifies that the Master clusterhead is connected to all other clusterheads.

$$\sum_{i=1}^N x_{i,j} \geq (P-2)M_j + 1 \quad \forall j \quad (2.25)$$

Constraint 2.26 ensures that the backbone is not fully connected. Unlike the previous model (EEC-FCB) the “=” sign is replaced by the “≤” symbol. In this way the model does not enforce complete connectivity, but lets the solver decide based on the cost of connectivity.

$$\sum_{i=1}^N \sum_{j=i+1}^N x_{i,j} \leq (N-P) + P(P-1)/2 \quad (2.26)$$

Constraints 2.27 and 2.28 ensure that the MCH is connected to all clusterheads and that clusterheads are connected to at least one other clusterhead (MCH). Clusterheads may be connected to other clusterheads as well.

$$\sum_{\substack{k=1 \\ j \neq k}}^N z_{j,k} \geq (P-2)M_j + y_j \quad \forall j \quad (2.27)$$

$$\sum_{\substack{k=1 \\ j \neq k}}^N z_{j,k} \leq (P-1)y_j \quad \forall j \quad (2.28)$$

Constraint 2.29 ensures that the backbone is symmetric.

$$z_{j,k} = x_{j,k} \quad \forall j, \forall k \quad (2.29)$$

Constraints in the EEC - CB model which are the same as the Energy Efficient Clustering-Fully Connected Backbone (EEC-FCB) model are as follows.

Constraint 2.30 indicates that normal nodes should be connected to only one clusterhead.

$$\sum_{i=1}^N x_{i,j} \leq (K_j + P - 1)y_j + (1 - y_j) \quad \forall j \quad (2.30)$$

Constraint 2.31 indicates that the total number of CHs is P .

$$\sum_{j=1}^N y_j = P \quad (2.31)$$

Constraint 2.32 and 2.33 state that the matrix x is symmetric and diagonal.

$$\sum_{i=1}^N x_{ii} = 0 \quad (2.32)$$

$$x_{i,j} = x_{j,i} \quad \forall i, \forall j \quad (2.33)$$

Constraint 2.34, 2.35 and 2.36 state that $w_{i,j}$ is 1 only if $x_{i,j}$ is 1 and y_j is 1.

$$w_{i,j} \leq x_{i,j} \quad \forall i \forall j \quad (2.34)$$

$$w_{i,j} \leq y_j \quad \forall i \forall j \quad (2.35)$$

$$w_{i,j} \geq x_{i,j} + y_j - 1 \quad \forall i \forall j \quad (2.36)$$

Constraint 2.37 and 2.38 state that regular nodes are not connected to other regular nodes and only connected to one CH.

$$\sum_{i=1}^N w_{i,j} \geq 1 \quad \forall j \quad (2.37)$$

$$\sum_{i=1}^N w_{i,j} \leq (P-1)y_j + (1-y_j) \quad \forall j \quad (2.38)$$

Constraints 2.39, 2.40 and 2.41 indicate that that variables $x_{i,j}$, y_j and $z_{j,k}$ have binary values.

$$x_{i,j} \in \{0,1\} \quad \forall i \forall j \quad (2.39)$$

$$y_j \in \{0,1\} \quad \forall j \quad (2.40)$$

$$z_{j,k} \in \{0,1\} \quad \forall j \forall k \quad (2.41)$$

2.1.1.3 LIMITATIONS OF THE EEC-CB MODEL

The following are the assumptions made by the authors in [34] when formulating the Energy Efficient Clustering-Connected Backbone (EEC-CB) model, and the limitations of the EEC-CB model.

1. Intra-Cluster Communication is not possible. Nodes in the same cluster do not connect directly to each other. Communication still goes through the clusterhead. In a MANET, two nodes in the same cluster could be exchanging data, but all this data would be routed through the clusterhead,
2. The model does not generate Multihop topologies. Clusters consist of only 1 hop nodes. In certain cases, nodes may not be able to connect directly to a clusterhead or it may be more efficient to hop using another node which is closer than the clusterhead.
3. The model is not scalable when solved (times out for networks above 9 nodes. This is more of a solver limitation rather than a model limitation)
4. There is no redundancy. Although the model may, depending on the topology, generate a solution with some redundancy, it is not a controlled redundancy, and there is no guarantee that the solution will have redundancy, in which case the master clusterhead will be a central point of failure and if the master clusterhead fails then each clusterhead becomes isolated from all other clusters.

5. Coverage is not considered (assumed that a node can communicate with all other nodes).
6. Localization is not considered (it is assumed that a node can determine the positions of all nodes through localization techniques).

2.1.2 RECOMMENDED ENHANCEMENTS FOR PROPOSED MODEL

Based on the detailed analysis of the Energy Efficient Clustering – Fully Connected Backbone (EEC-FCB), and the Energy Efficient Clustering – Connected Backbone (EEC-CB) models, the following enhancements will be made and integrated to the proposed model.

- The backbone of our model will be similar to the EEC-CB model, in that there will be a master clusterhead to which all other clusterheads will be connected (Star), but in addition there will also be a ring between the clusterheads (each clusterhead being connected to 2 other clusterheads, apart from the master clusterhead). The worst case of the EEC-CB model is that the Master clusterhead (MCH) fails and each cluster is isolated from all other clusters. The worst case in our model will be based on the cost of links. This will result in either a complete redundant set of links to go through if the MCH fails or the network split into groups of clusters and not completely isolated clusters.
- The proposed model will be such that it will consist of a base model with the redundant ‘backbone’ listed above, and will also have enhancements, which can be present as a single extension or a set of extensions. The proposed enhancements are described below.
 - Intra-Cluster communication will be enabled, allowing nodes within the same cluster to communicate directly without going through the clusterhead.
 - It will be possible to generate Multihop topologies, allowing nodes which are much further away from the clusterhead to connect to the clusterhead by hopping using other regular nodes.
 - Coverage restrictions will be taken into account. The proposed model will allow the coverage radius of all nodes to be specified, ensuring that only connections which are possible are established, i.e. Nodes are connected to nodes which are within their radius of coverage.

2.2 ANALYSIS AND SELECTION OF SOLVERS

In this section we will specify the criteria considered when selecting the set of solvers to use in the research. The final set of selected solvers will be listed and described.

2.2.1 SOLVER SELECTION CRITERIA

From the large number of solvers that are available, a manageable subset of solvers had to be selected to integrate with the proposed tool and to use as the basis for our performance comparison. These solvers had to be from both the generic Integer Linear Programming (ILP) and Boolean Satisfiability (SAT) area so that the feasibility of both could be assessed. The following factors were considered when selecting solvers:

- The first factor considered was the performance of solvers. This was assessed based on general consensus, wide-spread use and evaluations, and the results of contests such as the Pseudo-Boolean Solver contest [7].
- The second factor considered was the availability of solvers, i.e. whether or not the solvers are already available for use, or can be obtained, installed and run on a local machine.
- The final factor considered was the execution environment required to run the solver. It is important for comparison purposes that solvers be installed and executed on the same machine, under the same OS and under the same conditions. If one solver can only run on Windows and the other one only on Linux it is not possible to draw definite conclusions from the runtimes obtained.

2.2.2 SELECTED SET OF SOLVERS

From the set of solvers available we selected the following five solvers:

- CPLEX [37] is perhaps the most well-known and widely used commercial Integer Linear Programming (ILP) solver. It has been used by researchers attempting to solve wide ranging optimization problems in various fields and it is being used in various industrial applications [37]. In addition, CPLEX is developed, supported and maintained commercially and has frequent and regular upgrades. Although it is a commercial optimization suite, academic editions are also available for use. Additionally, as shown on their website, CPLEX has “a long history of constant performance improvement”, some of which are shown below [40]:
 - CPLEX 12.2 (2010): 50% overall, 2.7X on 1000 seconds and up
 - CPLEX 12.0 (2009): 30% overall, 2X on 1000 seconds and up

- CPLEX 11 (2007): 15% under one minute, 2X on 1-60 minutes, 10X on one hour and up
- CPLEX 10 (2006): 35% overall, 70% on ‘particularly difficult models’
- SCIP [41] is one of the best performing non-commercial ILP solvers. Figure 12 is dated 2nd November 2011 and taken from the SCIP website [42]. The results are from the Mixed Integer Linear Programming Benchmark 2010 [43]. Over the course of this research, SCIP v1.2 was used initially and then v2.01 was used.

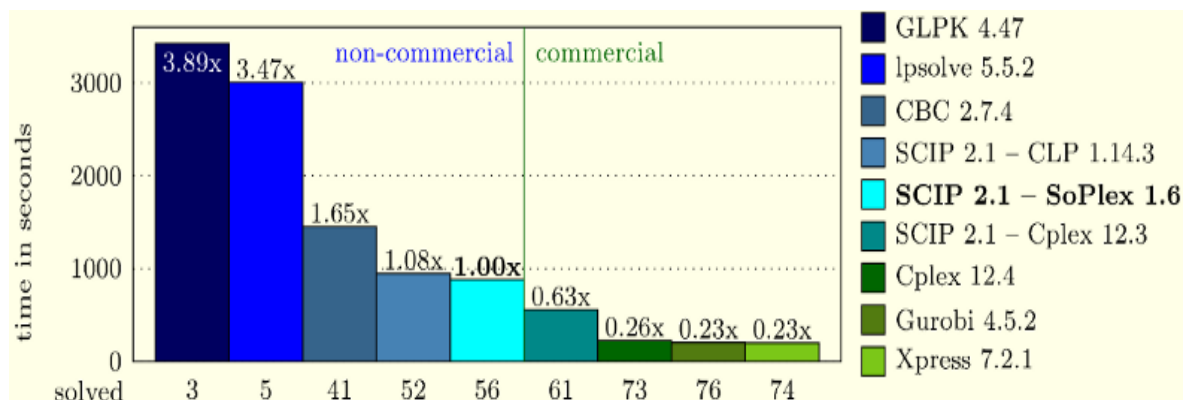


Figure 12: Performance Comparison of SCIP with Commercial and Non-Commercial ILP Solvers [41].

- BSOLO [44] is a Boolean Satisfiability (SAT) Solver capable of handling the optimization of Pseudo-Boolean constraints, and has performed consistently well in the Pseudo-Boolean Contests being held. It ranked high on the list between 2005 and 2010, for test cases involving optimization, particularly those with linear constraints. The results of these evaluations are available (2005-2010) are accessible at [5] .
- Minisat+ [45] is a SAT solver modified to handle Pseudo-Boolean constraints and has also performed consistently well over the Pseudo-Boolean contests held [5].
- Pueblo [46] is a ‘hybrid’ Pseudo-Boolean SAT Solver. It fared well in the 2005, 2006 and 2007 Pseudo-Boolean Contests, particularly when optimizing instances in the ‘Small Integers’ category.

Figure 13 gives an indication of the performance of BSOLO, Minisat+ and Pueblo. It is a graph from the 2007 Pseudo-Boolean contest [47] and is a comparison of number of instances solved within a given interval of time. As can be seen in Figure 13, Pueblo,

Minisat+ and BSOLO, all rank high in terms of the number of instances solved within the given time interval.

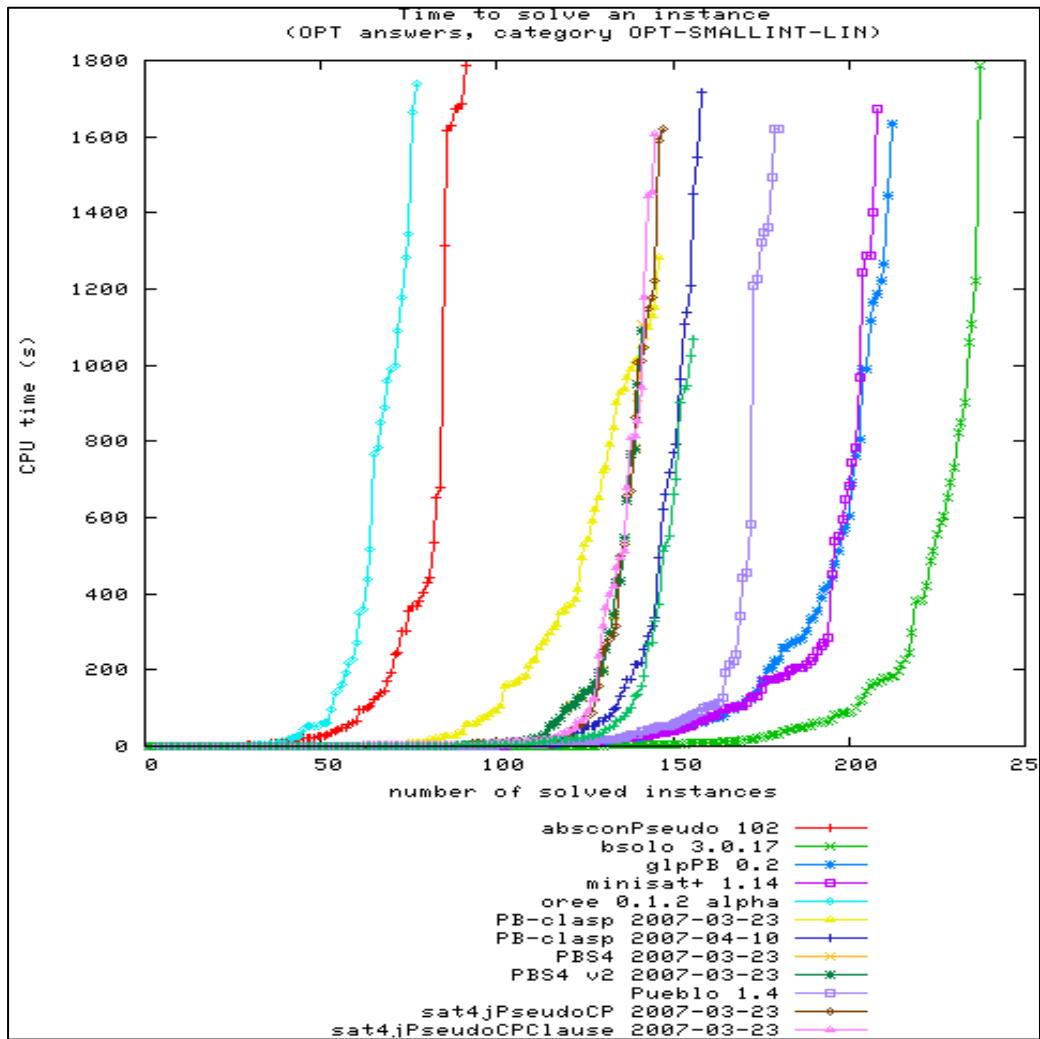


Figure 13: Pseudo-Boolean Contest Results Comparing Solver Performance in Terms of Number of Instances Solved in a Given Time Period [47].

2.3 ANALYSIS OF EXISTING TOOL

In this section, the tool designed by the authors in [34] will be described and its capabilities assessed. Additionally, the recommended set of enhanced features for the proposed tool will be listed.

The tool developed in [34] is a basic tool with an easy-to-use interface which would allow simplified topology creation, problem modeling and solution display. Figure 14 is a screenshot of the interface of tool developed in [34]. The interface consists of a grid for node placement and basic buttons in order to place and move nodes.

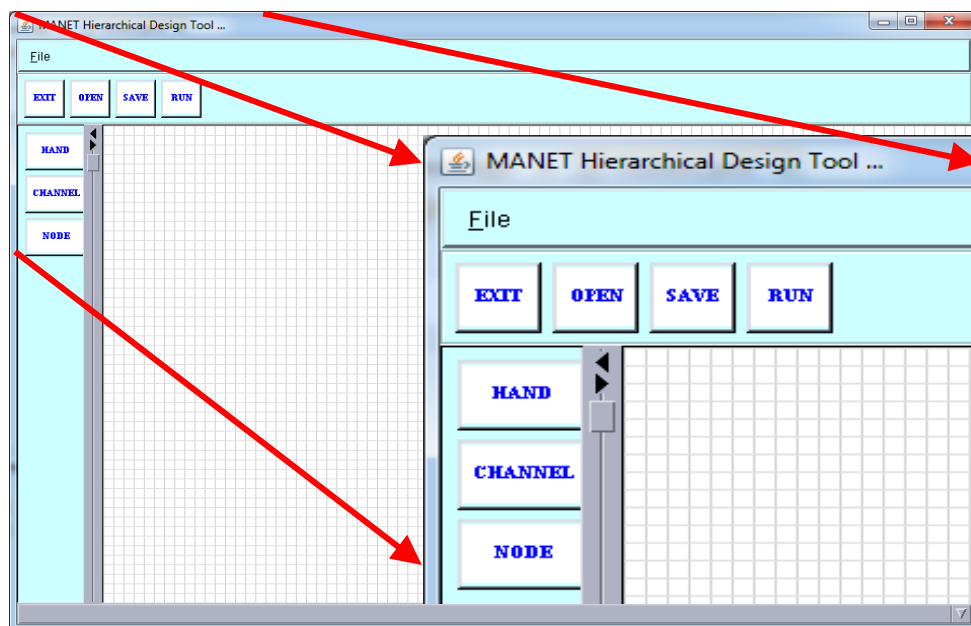


Figure 14: Screenshot of Existing Tool [34].

In order to understand how the tool in [34] functions, the steps which need to be followed to construct a topology, convert it to an Integer Linear Programming (ILP) model, and solve the ILP model using the integrated solver are listed as follows:

- The first step is to decide whether or not to use an existing topology or to create a new topology.
- If a new topology is being created, nodes can be placed by first selecting 'Node' from the left side panel and then placing the nodes where desired by clicking on the square on the grid.

- Once all nodes have been placed the topology is complete. The “Hand” tool can be used to move nodes around if they have not been placed correctly or as desired.
- Once the desired node placement has been achieved, the “Run” button is pressed. This will convert the topology on screen into its corresponding ILP formulation.
- The ILP formulation obtained for the topology will then be solved by the integrated solver.
- The solution to the ILP formulation obtained will then be displayed. The nodes selected to be clusterheads will be colored blue and the connections between nodes will be shown on the grid.
- It is important to note that all models, Energy Efficient Clustering – Fully Connected Backbone (EEC-FCB), Energy Efficient Clustering – Connected Backbone (EEC-CB) and Energy Efficient Clustering – Redundancy (EEC-R), are not integrated in the same tool. There are three versions of the tool; one for each model (i.e. one for EEC-FCB, one for EEC-CB, and one for EEC-R).

2.3.1 FEATURES AND LIMITATIONS OF EXISTING TOOL

The tool has the following features and capabilities:

- The ability to create a new custom node configuration.
- The ability to open the already saved node configuration.
- The ability to save the existing node configuration.
- The ability to place a node and move it to a desired position.
- The ability to convert the on-screen topology to its corresponding ILP formulation and then solve the resulting formulation using the integrated solver.

The tool has the following limitations:

- The tool has only one save slot. Saving a configuration overwrites any configuration present in the slot.
- The tool only uses only one solver to run the algorithm which times out for topologies greater than 9 nodes.
- The source code is extremely difficult to modify.
- The library dependencies have been implemented in such a way that the program is difficult to run on different machines (too many manual file placements need to be made).

- There are actually three versions of the tool. One each for the EEC-FCB (fully connected backbone), EEC-CB (connected backbone) and EEC-R (redundant backbone) algorithms, instead of having each algorithm as an option in one tool.

2.3.2 DESIRED CAPABILITIES FOR THE NEW TOOL

The proposed tool should have the following set of enhanced features (in addition to improved versions of features present in the tool developed in [34]):

- The proposed tool should be able to view the original problem (topology) side-by-side with the solution.
- The proposed tool should have the ability to save as many topologies as necessary and open them for later use.
- The proposed tool should be able to open previously generated solutions. Not only should the topologies be available but also solutions should be saved for reference.
- The proposed tool should be able to integrate with many solvers and handle the required file formats.
- The proposed tool should be able to display various aspects of the solution topology generated. This should include the following:
 - The ability to display only the backbone of the network topology.
 - The ability to displaying only the different clusters generated.
 - The ability to display specific connections selected by the user.
- The proposed tool should be able generate multiple problem instances of a given type which can be solved in one batch with a selected solver.
- The proposed tool should allow the user to adjust the size of nodes on the screen for visual convenience based on size of network)

This concludes the investigation phase of the research. At this point, the following three key pieces of information required to begin design and development of the proposed model have been gathered. The first is the set of requirements and improvements needed in order to design the proposed ILP formulation. The second key piece of information is the set of requirements and improvements to be implemented in the proposed tool. The final piece of information is the set of selected solvers to integrate with the tool and to use for testing purposes.

CHAPTER 3

DESIGN AND IMPLEMENTATION

In this section, the design and implementation of the proposed Integer Linear Programming (ILP) formulation and the proposed tool will be described in detail. It is important to keep in mind that the proposed model and proposed tool were designed and developed in parallel. This was so that the proposed tool could be used to test the proposed ILP model formulation at different stages and evaluate the correctness and capabilities of the model through an intuitive visual interface. Section 3.1 will detail the design and implementation of the proposed ILP formulation and its different enhancements including Intra-Cluster communication, Coverage constraints, and Multihop topologies. Section 3.2 will consist of a comprehensive example, illustrating how a single topology can be solved using the different ILP formulation and enhancements to yield unique solutions. Section 3.3 will describe the design and implementation of the proposed tool.

3.1 PROPOSED MODEL AND ENHANCEMENTS

In this section, the design and implementation of the proposed ILP formulation and enhancements will be described in detail. The following are the improvements that the proposed model should implement:

- The proposed ILP model should have a redundant backbone. The backbone of the proposed ILP model will be similar to the one in the Energy Efficient Clustering – Connected Backbone (EEC-CB) model, in that there will be a Master clusterhead to which all other clusterheads will be connected (Star). However, in addition to the star connections there will also be an open ring between the clusterheads (each clusterhead being connected to 2 other clusterheads, apart from the Master clusterhead). This will provide a layer of redundancy, without as many redundant connections as the Energy Efficient Clustering – Fully Connected Backbone (EEC-FCB), should the Master clusterhead fail. By having an open ring, the most expensive path (which closes the ring) is left open.

- Intra-Cluster communication should be enabled as an enhancement. Allowing nodes within the same cluster to communicate directly without going through the clusterhead will reduce the burden on the clusterhead, and allow it to conserve energy for when communication between clusters must occur.
- Multihop connections should be enabled as an enhancement. Allowing nodes which are much further away from the clusterhead to connect to the clusterhead by hopping using other closer nodes will result in more cost effective connections. The advantage of Multihop connections is that nodes do not have to waste energy communicating over longer distances.
- Nodes should be restricted to connect only to other nodes within their coverage radius. This is to reflect a real life environment where nodes can only communicate when they are both in each other's radius of coverage.

The three enhancements to the base model (Intra-Cluster Communication, Multihop Connections and Coverage Constraints), which will be discussed in detail in subsequent sections, will be implemented in such a way that they can be used individually or as a combination.

3.1.2 ILP FORMULATION OF THE PROPOSED BASE MODEL

This section will outline the development of the proposed Integer Linear Programming (ILP) formulation of a 'base model', i.e. a basic topology with an improved backbone, but with no enhancements such as Intra-Cluster communication and Multihop connections. As stated previously, our formulation of the clustering problem aims to modify the base formulation put forward in [34], through re-modeling with new equations and modifying the backbone.

The proposed model will use a star-ring backbone. The concept of a Master clusterhead (MCH), from the Energy Efficient Clustering – Connected Backbone (EEC-CB) model presented in [34], will be maintained, and all clusterheads will be connected to the Master clusterhead. Additional redundancy will be implemented in the proposed model by interconnecting the clusterheads in an 'open-ring' formulation, i.e. The most expensive link of the ring will be left open, as shown in Figure 15. The 'Star' is formed by connecting Node 9 (MCH) to the other clusterheads (Nodes 1, 10 and 5). The 'Ring' is formed between the clusterheads by connecting Node 1 to Node 10 and Node 10 to Node 5. However, Node 5 is

not connected to Node 1 leaving the ring ‘open’. This selection is not made randomly. As we proceed to detail the equations of the model, it will be made clear how these links were selected because based on low-cost criterion.

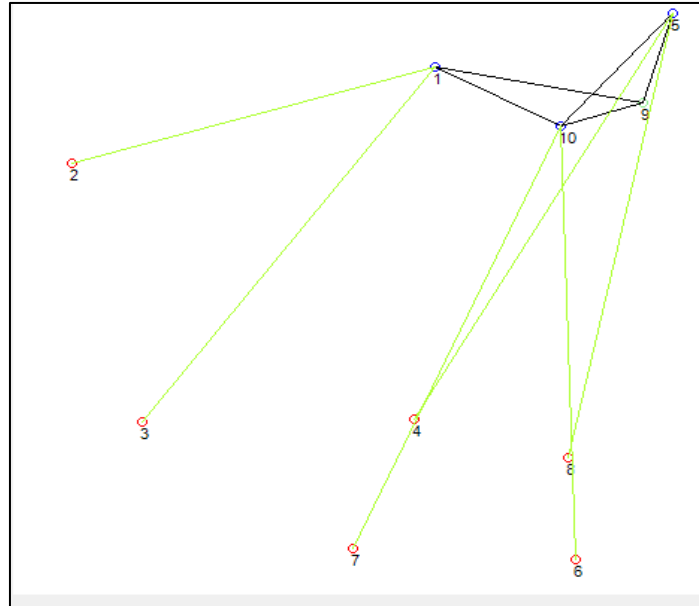


Figure 15: Solved Topology With a Star-Ring Backbone.

The benefits of the increased level of redundancy can be seen in that if the Master clusterhead (which is Node 9) fails, clusterhead 1 will still be able to communicate directly with clusterhead 10. Clusterhead 1 will also be able to communicate with clusterhead 5 through clusterhead 10. In this way the MCH is no longer the central point of failure. The number of backbone links is also less than the EEC-FCB model which produces a mesh of backbone network links.

The following assumptions which were made in the ILP formulations in [34] are also applicable to the proposed ILP formulation:

1. Localization is not considered. It is assumed that nodes are able to determine each other’s position, either through the use of GPS, or other localization techniques.
2. The variable a_j (named b_j in the EEC-FCB model) gets its value from an external source (algorithm, tool, etc). This is useful as multiple approaches/algorithms, which determine the suitability of a node in acting as a clusterhead, can be combined with this model without changing the equations, although this is out of the scope of our research.
3. All connections are single hop. All regular nodes are directly connected to a clusterhead.
4. There is no direct communication between nodes in the same cluster. Communication between nodes in the same cluster also goes through the clusterhead.

5. The cost of forming the topology (objective function) is kept the same.

The following are the equations which constitute the ILP formulation of the proposed Base Model.

The objective function to be minimized:

$$\text{Min } (x, y, z): \left(\sum_{i=1}^N \sum_{j=1}^N c_{i,j} x_{i,j} + \sum_{j=1}^N a_j y_j + \sum_{j=1}^N a_j M_j + \sum_{j=1}^N \sum_{k=1}^N h_{j,k} z_{j,k} \right) \quad (3.1)$$

The first term in the objective function represents the connections between nodes and clusterheads. The second term represents the selection of nodes to be clusterheads. In the proposed formulation, the Master clusterhead is no longer also a regular clusterhead. This results in the need for a new term, as the selection cost of the Master clusterhead still has to be taken into account. This new term is the third term in Equation 3.1. The last term represents the connections between clusterheads. The objective function aims to minimize the cost of connections or links between nodes, thereby minimizing the cost of sending/receiving data along these connections.

Constraint 3.2 is to enforce the restriction that there is only one Master clusterhead.

$$\sum_{j=1}^N M_j = 1 \quad (3.2)$$

Constraint 3.3 is to enforce the restriction that the total number of CHs is $P - 1$. Note: This is different from the EEC-CB model in that we do not count the Master clusterhead as also a regular clusterhead. That is to say that if there are a total of P clusterheads, there will be 1 Master clusterhead and $P-1$ regular clusterheads.

$$\sum_{j=1}^N y_j = P - 1 \quad (3.3)$$

Constraint 3.4 is the upper limit on the total number of connections a node has. If a node is a regular node it can at most be connected to one other node (this node will be clusterhead as enforced by later constraints). If a node is a clusterhead, it will be connected at most to K other regular nodes (this enforces the restriction of maximum cluster size).

$$\sum_{i=1}^N x_{i,j} \leq 1 + (K - 1)y_j \quad \forall j \quad (3.4)$$

Constraint 3.5 is the lower limit on the total number of connections a node has. If a node is a regular node it must be connected to at least one other node (which will be a clusterhead as

enforced by later constraints). If a node is a clusterhead it must support at least one node. If a node is a Master clusterhead it is not restricted to ‘1 connection to a regular node’. Rather, it can have (and in this case it should have) no connections to regular nodes.

$$\sum_{i=1}^N x_{i,j} \geq 1 - M_j \quad \forall j \quad (3.5)$$

Constraint 3.6 is the upper limit on the maximum number of backbone connections. If a node is a clusterhead it cannot have more than 3 backbone connections. (1 will be to a Master clusterhead for the star connection, and 2 will be to other regular clusterheads in order to establish the ring links). If a node is a Master clusterhead, it will be connected to all the regular clusterheads ($P-1$).

$$\sum_{\substack{k=1 \\ j \neq k}}^N z_{j,k} \leq (P - 1)M_j + 3y_j \quad \forall j \quad (3.6)$$

Constraint 3.7 is used to enforce the lower limit on the number of backbone connections. If a node is a regular clusterhead then it has to be connected to at least two other nodes. One other regular clusterhead and one master clusterhead. If a node is a Master clusterhead, it has to be connected to all the regular clusterheads ($P-1$).

$$\sum_{\substack{k=1 \\ j \neq k}}^N z_{j,k} \geq (P - 1)M_j + 2y_j \quad \forall j \quad (3.7)$$

Constraint 3.8 is used to enforce the restriction that backbone connections are only between the master clusterhead and regular nodes, or between regular clusterheads. The connections between regular nodes and clusterheads are not counted as backbone connections.

$$\sum_{\substack{k=1 \\ j \neq k}}^N z_{j,k} \leq \frac{M_j + y_j + M_k + y_k}{2} \quad \forall j \quad (3.8)$$

Constraint 3.9 is used to enforce the restriction that if a node is selected to be a regular clusterhead, it cannot be the master clusterhead and vice versa. The node can only be one of the two.

$$\sum_{j=1}^N (y_j + M_j) \leq 1 \quad (3.9)$$

Constraint 3.10 is used to ensure that nodes are not connected to themselves and Constraint 3.11 is used to diagonalize the matrix x which represents the connections between regular nodes and regular clusterheads. That is to say that if clusterhead 1 is connected to node 2, it is the same as saying node 2 is connected to clusterhead 1. Constraint 3.12 does the same for the z matrix which represents the interconnections between clusterheads.

$$\sum_{i=1}^N x_{ii} = 0 \quad (3.10)$$

$$x_{i,j} = x_{j,i} \quad \forall i, \forall j \quad (3.11)$$

$$z_{i,j} = z_{j,i} \quad \forall i, \forall j \quad (3.12)$$

Constraint 3.13 restricts the total number of connections between regular nodes and clusterheads to the same number as the number of regular nodes; each regular node must be connected to at least one other clusterhead.

$$\sum_{i=1}^N \sum_{j=i+1}^N x_{i,j} = (N - P) \quad (3.13)$$

Constraint 3.14 is used to restrict the total number of backbone connections to $2(P-1) - 1$. 1 is deducted because the ring will be left ‘open’ as described earlier.

$$\sum_{i=1}^N \sum_{j=i+1}^N z_{i,j} = 2(P - 1) - 1 \quad (3.14)$$

Constraint 3.15 is used to ensure that clusterheads do not connect to themselves.

$$\sum_{i=1}^N z_{ii} = 0 \quad (3.15)$$

Constraint 3.16 is used to ensure that regular nodes are not connected to each other. When x (non-backbone) connections are made, at least one of the nodes must be a clusterhead.

$$\sum_{\substack{k=1 \\ j \neq k}}^N x_{j,k} \leq \frac{1 + y_j + y_k}{2} \quad \forall j \quad (3.16)$$

Constraints 3.17 – 3.20 are used to restrict the variables to binary values.

$$x_{i,j} \in \{0,1\} \quad \forall i, \forall j \quad (3.17)$$

$$z_{j,k} \in \{0,1\} \quad \forall j, \forall k \quad (3.18)$$

$$y_j \in \{0,1\} \quad \forall j \quad (3.19)$$

$$M_j \in \{0,1\} \quad \forall j \quad (3.20)$$

The limitations of the proposed Base Model are as follows:

- All topologies must have at least one master clusterhead and two regular clusterheads.
- All regular clusterheads must support at least one regular node.
- Master clusterheads do not connect to any regular nodes.

The following differences can be observed between the proposed ILP formulation and the EEC-CB ILP formulation presented in [34]:

- The presence of additional redundancy in the backbone through the use of the ‘Ring’ to supplement the ‘Star’.
- The equations are reworked. The variable ‘ w ’ which was present in EEC-FCB and EEC-CB formulations in [34] has been removed.
- The Master clusterhead is not counted as one of the regular clusterheads. The number and nature of connections has been adjusted accordingly.
- The additional term in the objective function has been added to account for the cost of selection of the Master clusterhead selection, which is now no longer also counted as a regular clusterhead.

3.1.3 ENABLING MULTIHOP CONNECTIONS

In this section, the Multihop enhancement to the Base Model will be described. In the original Energy Efficient Clustering – Fully Connected Backbone (EEC-FCB) and Energy Efficient Clustering – Connected Backbone (EEC-CB) models presented in [34], all networks were generated with 1-hop connections which meant that all regular nodes were directly connected to clusterheads. The proposed Base Model is enhanced by enabling Multihop connections, allowing for 2-hop connections as shown in Figure 16. This allows for nodes to hop using a node that is closer to the clusterhead. The Multihop enhancement to the model has two parts. The first is to enable Multihop connectivity and the second part is to take into account the cost of enabling these connections. Enabling Multihop connections will require

modifying several constraints in the proposed Base Model, to allow for regular nodes to connect to other regular nodes as the hopping mechanism to connect to clusterheads.

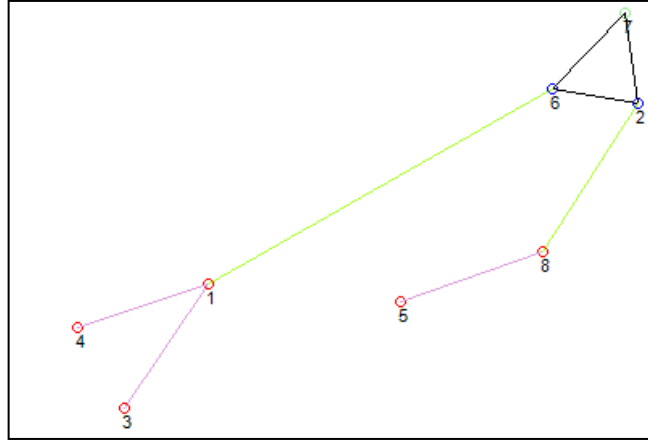


Figure 16: Multihop Network Topology.

Enabling Multihop connections requires the introduction of the following new variables:

- $b_{i,j,k} = \begin{cases} 1, & \text{if node } i \text{ is connected to clusterhead } k \text{ through regular node } j \\ 0, & \text{otherwise} \end{cases} \quad (3.21)$

- $q_{i,j} = \begin{cases} 1, & \text{if regular node } i \text{ is connected to regular node } j \\ 0, & \text{otherwise} \end{cases} \quad (3.22)$

Variable $b_{i,j,k}$ and $q_{i,j}$ are two new variables used when enabling multihop. These variables are required because the cost of the ‘hop’ connection will be different from regular connections represented by variable $x_{i,j}$. It is also important to remember that certain restrictions must be kept in place, for example, regular nodes can only hop using the regular nodes to connect to clusterheads, since it is illogical if they hop using one clusterhead to connect to another clusterhead.

3.1.3.1 CONSTRAINTS ENABLING MULTIHOP CONNECTIONS

The constraints unless shown here are kept the same as the proposed Base Model.

Constraint 3.23 is the updated version of Constraint 3.6. The maximum connections node ‘ i ’ can have is K which occurs when node ‘ i ’ is a clusterhead. In Multihop, the maximum cluster size must also include nodes that are connected to the clusterhead through hops. This is taken into account by Constraint 3.23.

$$\sum_{\substack{j=1 \\ j \neq k \\ j \neq i}}^N \sum_{\substack{k=1 \\ k \neq i}}^N b_{i,j,k} + x_{i,j} \leq K \quad \forall i \quad (3.23)$$

Constraint 3.24 is the updated version of Constraint 3.5. The minimum number of connections that node ‘ i ’ should have is 1 if it is a regular node and 0 if it is the master clusterhead.

$$\sum_{i=1}^N x_{i,j} + q_{i,j} \geq 1 - M_j \quad \forall j \quad (3.24)$$

Constraint 3.25 is the updated version of Constraint 3.13 and ensures that the total number of non-backbone connections is equal to $N-P$. This includes both hop based and direct connections.

$$\sum_{i=1}^N \sum_{j=i+1}^N x_{i,j} + q_{i,j} = (N - P) \quad (3.25)$$

Constraint 3.26 is used to ensure that only those nodes that are connected to the clusterhead ($x_{ik}=1$) can be used as hopping nodes.

$$\sum_{i=1}^N \sum_{j=1}^N \left(\sum_{\substack{k=1 \\ i \neq j \\ j \neq k \\ i \neq k}}^N t_{i,j,m^*} - x_{i,k} \right) \leq 0 \quad (3.26)$$

m^* is an index starting from 0, incremented when three conditions are satisfied ($i \neq j$, $j \neq k$, $i \neq k$) and used to indicate a potential hop path. m^* is used to indicate the number of potential hope path, not the identity of the possible hop path which would be $t_{i,j,k}$. The former is used because the emphasis is on whether or not a ‘hop’ path was taken and to simplify the coding of the model.

Constraint 3.27 and 3.28 are used to ensure that it is not possible to hop off of a clusterhead. That if y_j is 1 or y_i is 1 then all potential hops through y_j and y_i are deemed not possible because y_j or y_i is a clusterhead.

$$\sum_{i=1}^N \sum_{j=1}^N \left(\sum_{\substack{k=1 \\ i \neq j \\ j \neq k \\ i \neq k}}^N t_{i,j,m^*} + y_j \right) \leq 1 \quad (3.27)$$

$$\sum_{i=1}^N \sum_{j=1}^N \left(\sum_{\substack{k=1 \\ i \neq j \\ j \neq k \\ i \neq k}}^N t_{i,j,m^*} + y_i \right) \leq 1 \quad (3.28)$$

Constraint 3.29 is added to ensure that only either a direct connection to the clusterhead or a hop connection to a clusterhead exists from a particular node. The node cannot be connected to the clusterhead both directly and by hopping through another node.

$$\sum_{i=1}^N x_{i,j} + q_{i,j} \leq 1 \quad \forall j \quad (3.29)$$

Constraint 3.30 and 3.31 are used to identify that node i is connected to node j if it has hopped taken one of the potential hop paths. (N = total number of nodes)

$$\sum_{i=1}^N \sum_{j=1}^N q_{i,j} - \left(\sum_{\substack{k=1 \\ i \neq j \\ j \neq k \\ i \neq k}}^N t_{i,j,m^*} \right) \leq 0 \quad (3.30)$$

$$\sum_{i=1}^N \sum_{j=1}^N Nq_{i,j} - \left(\sum_{\substack{k=1 \\ i \neq j \\ j \neq k \\ i \neq k}}^N t_{i,j,m^*} \right) \geq 0 \quad (3.31)$$

Constraint 3.32 is used to enforce the restriction that a node cannot connect to itself through a hop.

$$\sum_{i=1}^N q_{ii} = 0 \quad (3.32)$$

Constraint 3.33 is used to state that node i cannot hop to j if j has hopped to i . (q connection matrix is not diagonal). Saying that node i has hopped to clusterhead j is not the same as saying the node j has hopped to clusterhead i . Which node is the clusterhead matters unlike with the x connection where just the presence of the connection matters.

$$\sum_{i=1}^N q_{i,j} + q_{j,i} \leq 1 \quad \forall j \quad (3.33)$$

Constraint 3.34 and constraint 3.35 are used together to implement an ‘AND’ logic. Node k can hop using node j to clusterhead i , if i is a clusterhead and j is connected to i and connecting k to j is possible.

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{\substack{k=1 \\ j \neq i \\ j \neq k \\ i \neq k}}^N 2b_{i,j,k} - x_{i,j} - q_{j,k} \leq 0 \quad \forall j \quad (3.34)$$

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{\substack{k=1 \\ j \neq i \\ j \neq k \\ i \neq k}}^N x_{i,j} + q_{j,k} - b_{i,j,k} \leq 1 \quad \forall j \quad (3.35)$$

Altogether these constraints have now enabled Multihop (2-hop) connections while keeping earlier restrictions, such as the connection count and the maximum cluster size constraint, in place.

3.1.3.2 INCORPORATING THE COST OF MULTIHOP CONNECTIONS

The second part of the Multihop Connection enhancement involves taking into account the cost of Multihop connections. Changes need to be made to the objective function. This is because now a new cost is in place; the cost of ‘hopping’. If node ‘ i ’ is connecting to clusterhead ‘ j ’ by hopping through node ‘ k ’ then node k is now also a second tier clusterhead supporting one regular node. The cost incurred by node k for routing node i ’s data to clusterhead j must be taken into account.

This is done by adding another term to the objective function as this cost must also be taken into account as part of the total cost which must be minimized.

$$\sum_{i=1}^N \sum_{j=1}^N c_{ij}x_{ij} + \sum_{j=1}^N a_j y_j + \sum_{j=1}^N a_j M_j + \sum_{j=1}^N \sum_{k=1}^N h_{jk} z_{jk} + \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N B_{ij} b_{ijk} \quad (3.36)$$

B represents the cost of connecting node j and node k . This cost is similar to the costs in the original objective function in the proposed Base Model and in the Energy Efficient Clustering – Fully Connected Backbone (EEC-FCB) and Energy Efficient Clustering-Connected Backbone (EEC-CB) models presented by the authors in [34]. It is similar in that it is again proportional to the distance between the hopping node and the intermediate node used to hop to the clusterhead as shown in Equation 3.37.

$$B_{j,k} \propto d_{j,k}^n \quad (3.37)$$

The value of n depends on several factors which are as follows:

- The value of n depends on the degree to which Multihop connections should be encouraged over direct connections.
- It depends on how mobile the nodes are, for example, if the node being used to hop is moving, then the hopping node may lose connectivity if it moves out of range.
- It depends on the distances between the nodes and the size of the grid (location) over which the nodes are deployed.
- It also depends on the coverage radius of nodes.

However, the value of n is not proportional to the square of the distance as with the regular node-clusterhead connections ($n=2$), and it is not proportional to the cube of the distance as with the clusterhead-clusterhead connections ($n=3$). Rather, it is somewhere in between. A suitable starting value of n can be taken to be 2.5. This value can then be adjusted or tuned through simulation based on how preferred Multihop connections are over direct connections.

3.1.4 ENABLING INTRA-CLUSTER CONNECTIONS

In the Energy Efficient Clustering – Fully Connected Backbone (EEC-FCB) and Energy Efficient Clustering-Connected Backbone (EEC-CB) models presented by the authors in [34], all nodes in a cluster were connected only to the clusterhead for that cluster. The cluster was responsible for both Intra-Cluster and Inter-Cluster routing. By routing Intra-Cluster communications, the clusterhead is losing energy that could be used for Inter-Cluster routing. The model is enhanced by enabling Intra-Cluster node-node connections. A sample topology is shown in Figure 17.

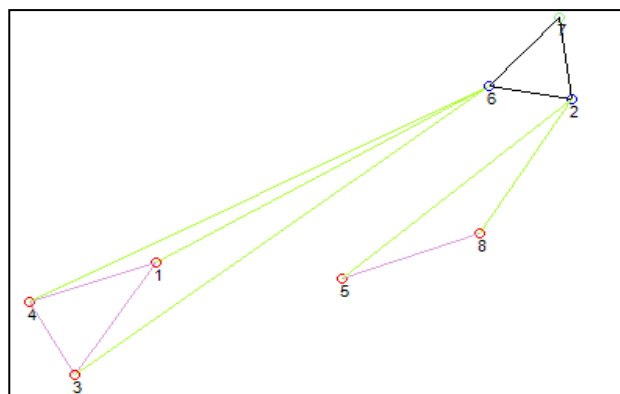


Figure 17: Intra-Cluster Communication in a Network Topology.

The proposed enhancement requires the introduction of the following new variables. Variable $v_{i,j}$ and f_{i,j,m^*} are two new variables used when enabling Intra-Cluster connections.

$$\bullet \quad f_{i,j,m^*} = \begin{cases} 1, & \text{if node } i \text{ and node } j \text{ are connected to the same clusterhead} \\ 0, & \text{otherwise} \end{cases} \quad (3.38)$$

$$\bullet \quad v_{i,j} = \begin{cases} 1, & \text{if regular node } i \text{ is connected to regular node } j \\ 0, & \text{otherwise} \end{cases} \quad (3.39)$$

m^* is an index starting from 0, incremented when three conditions ($i \neq j$, $j \neq k$, $i \neq k$) are satisfied and used to indicate a possibility of 2 nodes being connected to the same clusterhead. m^* is used to indicate the number of possibility, not the identity of nodes involved. There will always be $N-2$ possibilities. For example: 7 node network. When considering whether node i and node j , one must check if they are both connected to the same clusterhead which could be anyone of the 5 remaining nodes (should they be selected to be clusterheads).

3.1.4.1 CONSTRAINTS ENABLING INTRA-CLUSTER COMMUNICATION

Constraints 3.40 and 3.41 are used to identify that node i is connected to node j if one of the possibilities of the both of them being connected to the same clusterhead has occurred. ($N =$ total number of nodes)

$$\sum_{i=1}^N \sum_{j=1}^N (v_{i,j} - \sum_{\substack{k=1 \\ i \neq j \\ j \neq k \\ i \neq k}}^N f_{i,j,m^*}) \leq 0 \quad (3.40)$$

$$\sum_{i=1}^N \sum_{j=1}^N (Nv_{i,j} - \sum_{\substack{k=1 \\ i \neq j \\ j \neq k \\ i \neq k}}^N f_{i,j,m^*}) \geq 0 \quad (3.41)$$

Constraint 3.42 is used to enforce the restriction that a node cannot connect to itself through a hop.

$$\sum_{i=1}^N v_{ii} = 0 \quad (3.42)$$

Constraint 3.43 is used to state that node i being connected to node j in the same cluster also implies that node j is connected to node i (Matrix is diagonal).

$$\sum_{i=1}^N v_{i,j} + v_{j,i} = 0 \quad \forall j \quad (3.43)$$

Constraints 3.44 and Constraint 3.45 are used together to implement an ‘AND’ logic. Node i and node j are connected through an Intra-Cluster connection if they are both connected to clusterhead k , satisfying the m *th possible clusterhead connection.

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{\substack{k=1 \\ j \neq i \\ j \neq k \\ i \neq k}}^N 2f_{i,j,m^*} - x_{i,k} - x_{j,k} \leq 0 \quad \forall j \quad (3.44)$$

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{\substack{k=1 \\ j \neq i \\ j \neq k \\ i \neq k}}^N x_{i,k} + x_{j,k} - f_{i,j,m^*} \leq 1 \quad \forall j \quad (3.45)$$

3.1.4.2 INCORPORATING THE COST OF INTRA-CLUSTER CONNECTIONS

The Intra-Cluster communication enhancement to the proposed Base Model also has two parts (similar to Multihop connections). The first is to enable it. The second part is to take into account the cost of enabling Intra-Cluster connections. Equation 3.46 is the updated objective function.

$$\sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij} + \sum_{j=1}^N a_j y_j + \sum_{j=1}^N a_j M_j + \sum_{j=1}^N \sum_{k=1}^N h_{jk} z_{jk} + \sum_{j=1}^N \sum_{k=1}^N A_{jk} v_{jk} \quad (3.46)$$

A is the weight associated with connecting node j and node k . Since the nodes being connected are regular nodes and there is no additional routing involved, the cost (A) will be the same as node-clusterhead connections. (i.e. proportional to the square of the distance between the 2 nodes). This is shown in Equation 3.47 below.

$$A \propto d_{j,k}^2 \quad (3.47)$$

3.1.4 ADDING COVERAGE RESTRICTIONS

The proposed Base Model can be extended to take into account the coverage radius of the nodes in the network, and ensure that connections are established only between nodes that are within each other's coverage radius. This will no longer assume that all nodes can communicate with each other irrespective of where they may be located.

Similar to the manner in which distances between nodes determine the cost of the connections, they can also be compared to the coverage radius of each node and used to obtain a matrix of nodes to which each node can connect to (1) and to which it can't (0).

In equation 3.48, the variable $cv_{i,j}$ is the binary value which represents whether or not node j and i are in each other's coverage radius. If they are not, then they cannot be connected. If they are, then they can be connected but may not necessarily be connected. The actual connection will depend on the cost (which is proportional to the distance).

$$cv_{i,j} = \begin{cases} 1, & \text{if node } i \text{ and node } j \text{ are in each other's coverage radius} \\ 0, & \text{otherwise} \end{cases} \quad (3.48)$$

These values can then be used to enforce the possibility of connectivity between nodes using constraints 3.49 and 3.50 as given below.

$$\sum_{i=1}^N x_{i,j} \leq cv_{i,j} \quad \forall j \quad (3.49)$$

$$\sum_{i=1}^N z_{i,j} \leq cv_{i,j} \quad \forall j \quad (3.50)$$

Additionally, if the coverage constraints are also to be implemented together with Multihop and/or Intra-Cluster Communication enhancements, then the coverage constraints should also be applied to the connection variables for those enhancements (q for Multihop connections and v for Intra-Cluster connections) as shown in constraints 3.51 and 3.52.

$$\sum_{i=1}^N q_{i,j} \leq cv_{i,j} \quad \forall j \quad (3.51)$$

$$\sum_{i=1}^N v_{i,j} \leq cv_{i,j} \quad \forall j \quad (3.52)$$

3.1.5 USING MULTIPLE ENHANCEMENTS

It is possible to have the proposed Base Model combined with all the proposed enhancements. The list of possible combinations of enhancements is as follows (where IC=Intra-Cluster Communication, MH = Multihop Connections, CV=Coverage Constraints):

- S-R Only
- S-R with IC
- S-R with MH
- S-R with CV
- S-R with IC and MH
- S-R with IC and CV
- S-R with MH and CV
- S-R with all 3: MH, IC and CV

An example topology is shown in Figure 18, with a possible solution of the combined enhancement of SR with MH, IC and CV shown in Figure 19.

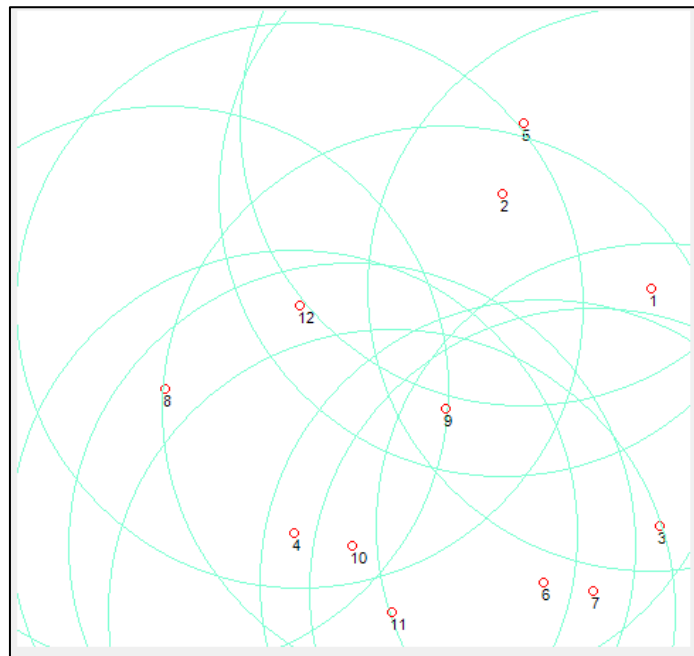


Figure 18: A 12 Node Network Topology Displaying Coverage.

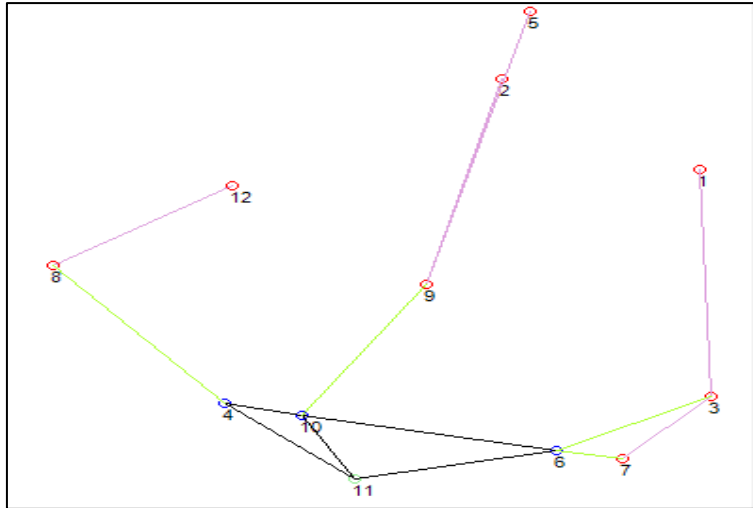


Figure 19: Solving a 12 Node Topology With a Combined Model Including Coverage Constraints, Intra-Cluster Communication and Multihop Connections.

3.1.6 EXAMPLE WALKTHROUGH WITH ALL PROPOSED ENHANCEMENTS

In order to illustrate the application of the proposed formulation and enhancements, using small scale networks, the topology shown in Figure 20, is used as an example. The ILP formulations of the SR, SR+CV, SR+CV+MH, SR+CV+IC (where IC=Intra-Cluster Communication, MH = Multihop Connections, CV=Coverage Constraints) models corresponding the topology are generated. The complete formulations and the corresponding solutions for each formulation are provided in Appendix B.

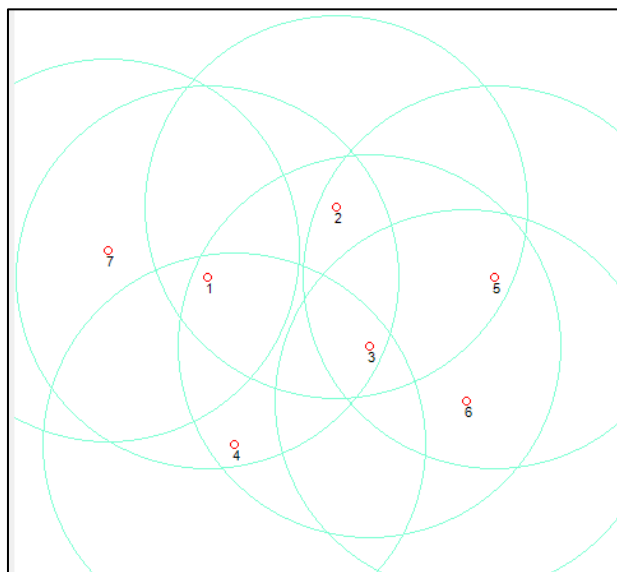


Figure 20: Example Topology to Be Solved With All Enhancements.

3.2 DESIGN OF NEW AND IMPROVED TOOL

In this section, the design and implementation of the proposed tool will be described. This will include the features implemented, the limitations of the proposed tool, and the areas for further enhancement.

3.2.1 MAJOR DESIGN DECISIONS

In this section, the major decisions taken in designing the proposed tool and the reasoning behind each decision will be discussed. The following are the main design decisions:

- The first decision was the choice of programming language to design and develop the proposed tool. The original tool in [34] was designed using JAVA. While using it, we realized that the primary purpose of our tool is to provide an intuitive Graphical User Interface (GUI), for users to be able to generate topologies, open and save previously generated topologies and solutions. Keeping this in mind we chose to use Visual Basic as the language of choice in order to generate a simple but effective Graphical User Interface (GUI).
- The second decision was to determine how the proposed tool would connect to the set of selected solvers. The original tool in [34] had a solver integrated into the tool. Commercial solvers provide the JAVA libraries but at a price and not all solvers, particularly the non-commercial SAT solvers have this option. In addition, in order to be able to compare the performance of different solvers, it would make sense if all solvers were running on the same machine. We chose to have all selected solvers installed on a Linux server and allow our tool to solve problems in 2 ways:
 - Generate the problems and allow the end user to manually take/transfer the file to the solver, generate the solution using the solver of choice and retrieve the solution to feed it back into the tool. This way is manual and slow.
 - The second and more practical option in the tool is the ability of the tool to connect to pre-configured servers with pre-configured but customizable credentials, send the generated problem through FTP, execute the solver and retrieve the solution. This process all happens in the background and the user seeing only the retrieved solution displayed as a topology beside the problem.

- The third decision was to have a database in order to store the server parameters, solver locations, and user log in credentials, so that the user does not need to enter them repeatedly and can adjust them easily if they change. The advantage of this is that when new versions of solvers added, or if paths change, they can be reconfigured easily.

3.2.2 TOOL IMPLEMENTATION DETAILS

Below are the screenshots and descriptions of various functions that make up the finished tool. The screen shown in Figure 21 is that of the main screen with a sample topology created and solved. When using the tool there are two initial options available at the screen for the user, to either start by creating a new topology or opening/editing an existing topology. It is possible to do both, by clicking on the appropriate button on the toolbar or selecting the required option from the menu. Initially, when a new topology is created or opened, it will be in a ‘flat’ state prior to clustering. Once generated, the user can save the topology.

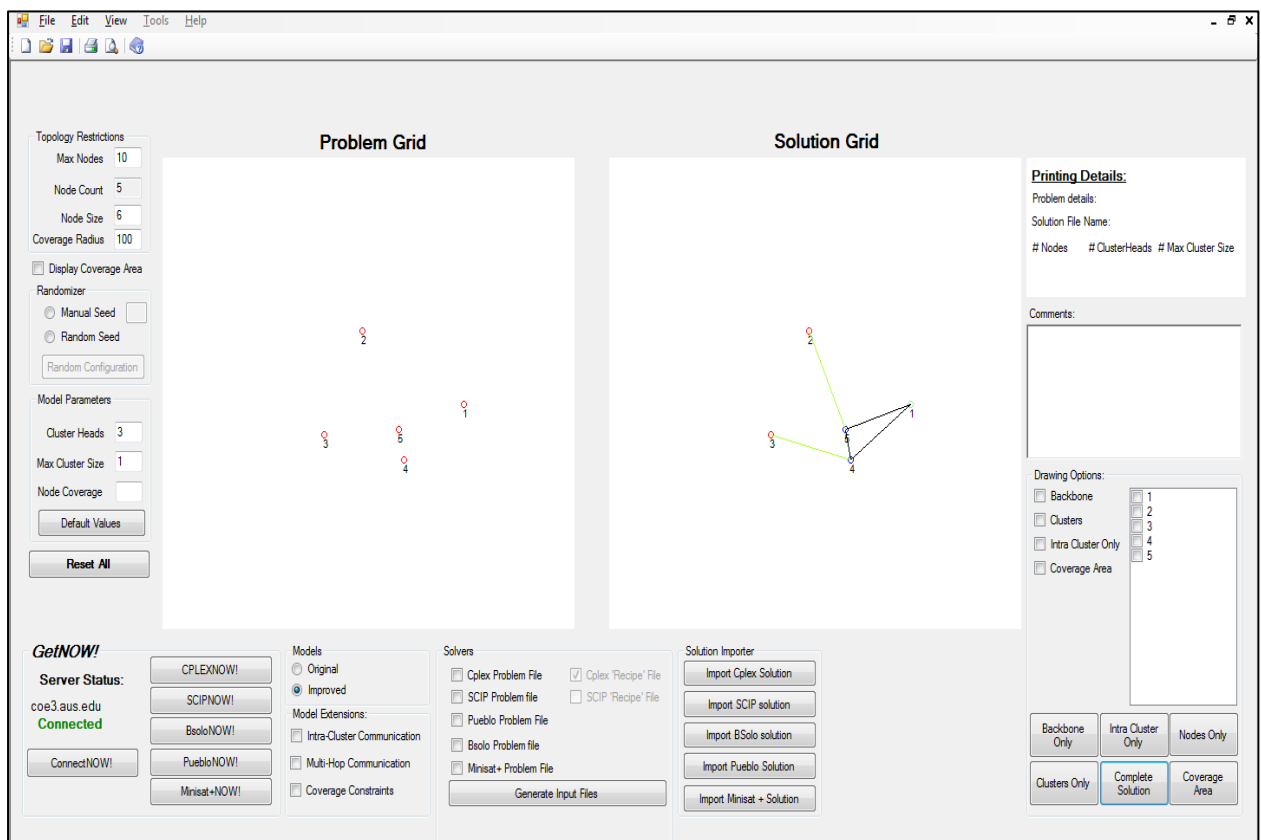


Figure 21: Proposed Tool Screen 1: Sample Topology Shown with Corresponding Solution

The problem can be generated by filling in the required parameters of the solution topology and selecting the different solvers for which the problem is required. These parameters include Node Size (for visual purposes only), Number of Nodes, Number of clusterheads and the maximum cluster size the solution should have. All that is then required is to click on the ‘Generate’ button. This process will generate the ILP formulations for the on-screen topology. These files can now be manually taken to the server and executed on the solver of choice by the end-user.

The solution returned by the solver can similarly be manually copied back to the software and placed in the folder of choice. The solution folders have also been auto-created and it is preferred to place the solution in one of these folders. The solution can then be opened by clicking on the ‘Import solution’ button for the solver of choice. The solution will then be parsed and the output of the variable assignments displayed as a solution topology as shown by the test topology solution displayed in Figure 21.

Figure 22 shows a particular solution viewed in two different ways. The left side is the complete/entire solution and the right side is the same solution with only the network backbone connections displayed. This illustrates some of the features available in the tool to customize how the solution is displayed on the screen. Selected connections can be displayed, clusters only can be displayed, or just the network backbone can be displayed.

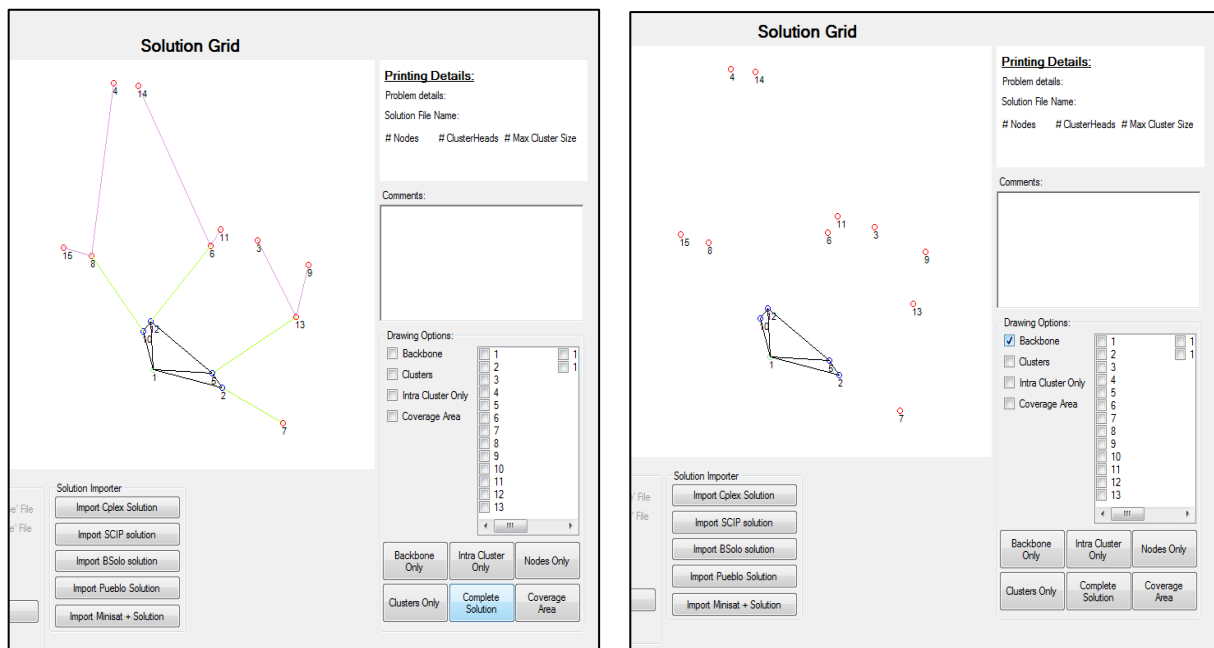


Figure 22: Proposed Tool Screen 2: Viewing Different Aspects of a Generated Solution

3.2.3 IMPLEMENTING THE ‘GETNOW!’ FEATURE IN THE PROPOSED TOOL

Using a library called ‘sharpSsh’, available at [48], a feature called ‘GetNOW!’ has been implemented in the proposed tool. The sharpSsh library allows for Secure Shell (SSH) and Secure File Transfer Protocol (SFTP) connections to be made to servers. This library provides a great deal of functionality, allowing it to integrate seamlessly with the proposed tool to be able to open connections to the Linux server on which the solvers have been installed. In addition to being able to transfer files to and from the server it is possible to execute commands on the server. With the help of the sharpSsh library it is possible for the tool to open an SFTP connection and an SSH connection to the server configured earlier. On clicking ‘*NOW’ button where * is the name of the solver, ex: CPLEXNOW, the solver will open two connections to the Linux server, one SFTP session and one SSH session. The tool will then generate the ILP formulation for the solver of choice (on the local machine), and send it to the server by SFTP. Then using the SSH session, it will execute the desired solver and retrieve the solution. Each stage of the process is visible in a dialog box on the screen shown in Figure 23

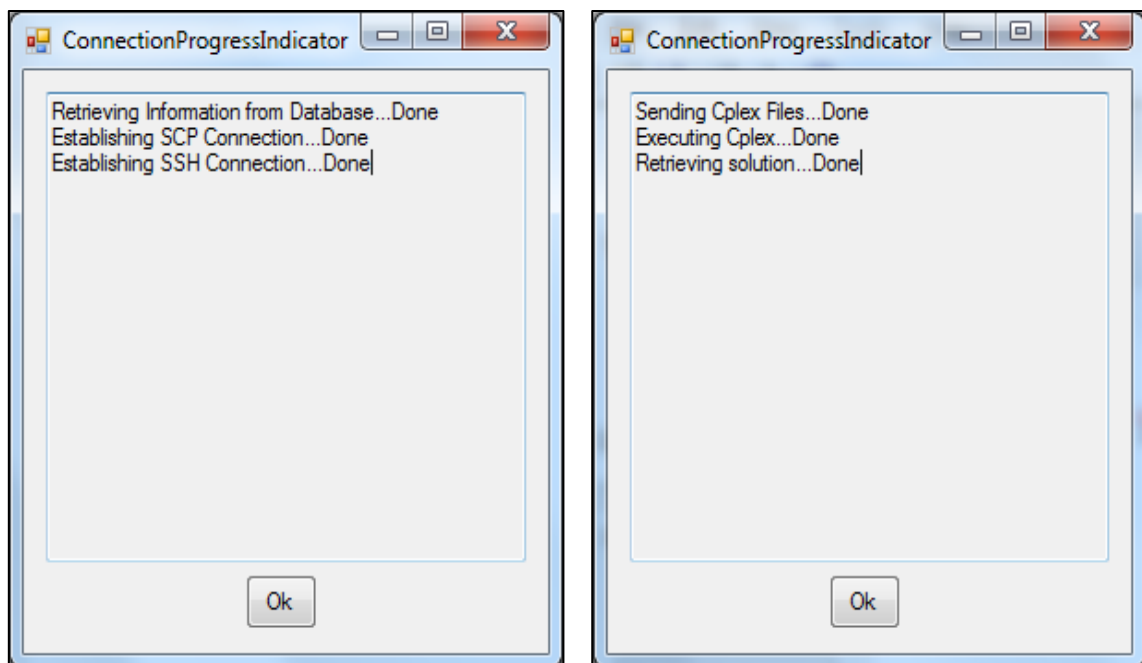


Figure 23: Proposed Tool Screen 3: Connection Dialog Box Displaying Status Messages Allowing Users to Monitor Progress

CHAPTER 4

TESTING

In this section we will detail the tests carried out over the course of the research. This will include initial testing using an early prototype of the proposed tool with the Energy Efficient Clustering – Fully Connected Backbone (EEC-FCB) model proposed in [34]. It will also cover the testing procedures for the final set of comprehensive tests conducted with the proposed ILP formulations, using the proposed tool and the selected set of solvers.

4.1 INITIAL TESTING USING THE EEC-FCB MODEL

In this section, we will describe the initial set of tests conducted with an early prototype of the proposed tool with the EEC-FCB model presented in [34].

Once the proposed tool was able to generate ILP formulations of the EEC-FCB model, sample topologies were created, and their corresponding EEC-FCB formulations were generated. The generated formulations were then fed to the CPLEX [37] solver to obtain the solutions. What we found was that CPLEX was unable to solve any instance and declared all instances ‘infeasible’. After a detailed study of various different instances and manual attempts to solve for the optimum solution, the point of conflict was identified and will be illustrated through the following example. Assume a sample network with the following configuration; four nodes to be configured such that there are two regular clusterheads with each clusterhead supporting one regular node. The sample topology and the sample solution to this topology are shown in Figure 24.

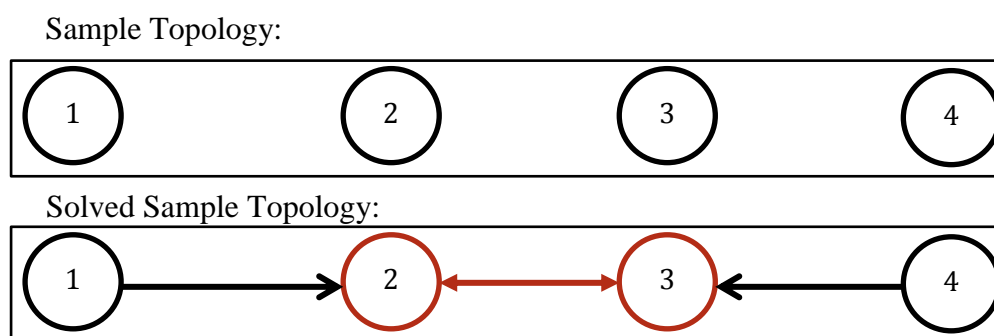


Figure 24: Example of Conflict in FCB - Original Topology and Solved Topology

For the EEC-FCB model, this is arguably the most straightforward topology to solve, and if the topology is exactly as shown above, visually one can identify one probable solution (possibly the ideal solution) as the one shown in Figure 32. In the solution, nodes 2 and 3 are the clusterheads which produce the combination of least cost backbone connections and regular-node to clusterhead connections.

However CPLEX is unable to obtain this solution (or any solution for that matter) and immediately says that the problem is “infeasible”. This implies that an equation or set of equations is causing an unresolvable conflict.

4.1.1 IDENTIFYING THE CONFLICT IN THE EEC-FCB MODEL

When solving the problem manually, it is possible to work backwards by taking the solution in Figure 32, which should satisfy the problem, determining its variable assignment, and plugging this assignment back into the original equations to determine the conflict. The description of the variables is as follows:

$$x_{i,j} = \begin{cases} 1, & \text{if node } i \text{ is connected to Clusterhead } j \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

$$y_j = \begin{cases} 1, & \text{if node } j \text{ is a clusterhead} \\ 0, & \text{otherwise} \end{cases} \quad (4.2)$$

$$w_{i,j} = \begin{cases} 1, & \text{if node } x_{i,j} \text{ is 1 and } y_j \text{ is 1} \\ 0, & \text{otherwise} \end{cases} \quad (4.3)$$

Based on the definition of the variables and the potential solution shown in Figure 32, one can determine the values of the variables x , y and w and these are shown in Table 1. All values not shown are equal to zero.

Table 1

Variable Assignment for EEC-FCB Conflict Example

x_{ij} values	y values	w_{ij} values
$x_{12}=1$	$y_2=1$	$w_{12}=1$
$x_{23}=1$	$y_3=1$	$w_{23}=1$
$x_{32}=1$	$y_2=1$	$w_{32}=1$
$x_{43}=1$	$y_3=1$	$w_{43}=1$

When plugging these variable values back into the original constraints of the EEC - FCB model, the source of the conflict can be identified as being constraints 2.13 and 2.14, shown again here, for convenience, as constraint 4.4 and constraint 4.5.

$$\sum_{i=1}^N w_{i,j} \geq 1 \quad \forall j \quad (4.4)$$

$$\sum_{i=1}^N w_{i,j} \leq (P-1)y_j + (1-y_j) \quad \forall j \quad (4.5)$$

Substituting the values from Table 1 into constraint 2.13/4.4 we get the following:

$$W_{11} + W_{21} + W_{31} + W_{41} \geq 1$$

$$0 + 0 + 0 + 0 \geq 1$$

$$\mathbf{0 \geq 1}$$

$$W_{12} + W_{22} + W_{32} + W_{42} \geq 1$$

$$1 + 0 + 1 + 0 \geq 1$$

$$\mathbf{2 \geq 1}$$

$$W_{13} + W_{23} + W_{33} + W_{43} \geq 1$$

$$0 + 1 + 0 + 1 \geq 1$$

$$\mathbf{2 \geq 1}$$

$$W_{14} + W_{24} + W_{34} + W_{44} \geq 1$$

$$0 + 0 + 0 + 0 \geq 1$$

$$\mathbf{0 \geq 1}$$

As we can see here that 0 can never be greater than 1. When attempting to solve the problem, CPLEX [37] detects the conflict in this equation and declares all problem instances infeasible.

Similarly for constraint 2.14/4.5 when substituting the values from Table 1, we get the following:

$$W_{11} + W_{21} + W_{31} + W_{41} \leq (P-1)y_j + (1-y_j)$$

$$0 + 0 + 0 + 0 \leq (2-1)(0) + (1-0)$$

$$\mathbf{0 \leq 0}$$

$$W_{12} + W_{22} + W_{32} + W_{42} \leq (P-1)y_j + (1-y_j)$$

$$1 + 0 + 1 + 0 \leq (2-1)(1) + (1-1)$$

$$\mathbf{2 \leq 1}$$

$$W_{13} + W_{23} + W_{33} + W_{43} \leq (P-1)y_j + (1-y_j)$$

$$0 + 1 + 0 + 1 \leq (2-1)(1) + (1-1)$$

$$\mathbf{2 \leq 1}$$

$$W_{14} + W_{24} + W_{34} + W_{44} \leq (P-1)y_j + (1-y_j)$$

$$0 + 0 + 0 + 0 \leq (2-1)(0) + (1-0)$$

$$\mathbf{0 \leq 0}$$

Here as well, when solving CPLEX detects the conflict and declares all problem instances infeasible.

4.1.2 CORRECTING THE CAUSE OF CONFLICT IN THE EEC-FCB MODEL

The definition of Constraints 2.13/4.4 and 2.14/4.5 as described in [34] is as follows.

“Constraints 12 and 13 ensure that if a node is chosen to be a regular node, it should be connected to only one CH and not to another regular node.” [34]

Constraint 4.4 is setting the lower bound for the number of connections from a node.

Constraint 4.5 is setting the upper bound for the number of connections from a node.

Based on this we modify constraints 4.4 and 4.5 as follows:

For Constraint 4.4, which represents the lower-bound of connections to/from a node, one can assume the following:

- If a node is a regular node it must have at least 1 connection from it and this must be to a clusterhead.
- If a node is a clusterhead it will have a minimum of $(P-1)$ connections from it in order to ensure the fully connected backbone.

This is translated into the following equation.

$$\sum_{i=1}^N w(i,j) \geq y(j) \quad (4.6)$$

Applying Equation 4.6 to the example shown in Figure 32 and substituting the values in Table 1 the following equations are obtained.

- $W_{11} + W_{21} + W_{31} + W_{41} \geq y_j$

- $0 + 0 + 0 + 0 \geq 0$
- $\mathbf{0} \geq \mathbf{0}$
- $W_{12} + W_{22} + W_{32} + W_{42} \geq y_j$
 - $1 + 0 + 1 + 0 \geq 1$
 - $\mathbf{2} \geq \mathbf{1}$
- $W_{13} + W_{23} + W_{33} + W_{43} \geq y_j$
 - $0 + 1 + 0 + 1 \geq 1$
 - $\mathbf{2} \geq \mathbf{1}$
- $W_{14} + W_{24} + W_{34} + W_{44} \geq y_j$
 - $0 + 0 + 0 + 0 \geq 0$
 - $\mathbf{0} \geq \mathbf{0}$

All equations are satisfied and constraint 4.6, which is the improved version of constraint 4.4, does not cause a conflict.

Similarly for constraint 4.5 which represents the upper bound of connections to/from a node, one can assume the following:

- If a node is a regular node it must have no more than 1 connection and this must be to a clusterhead.
- If a node is a clusterhead it must have no more than $(P-1) + (K)$ where the $(P-1)$ connections are to other clusterheads and the K connections are to regular nodes such that the maximum cluster size K is not exceeded.

This is translated into the following equation.

$$\sum_{i=1}^N w(i, j) \leq (K + P - 1)y_j \quad (4.7)$$

Applying this to the problem in Figure 32 and substituting the values from Table 1 results in the following equations.

- $W_{11} + W_{21} + W_{31} + W_{41} \leq (K+P-1)y_j$
 - $0 + 0 + 0 + 0 \leq (1+2-1)(0)$
 - $\mathbf{0} \leq \mathbf{0}$
- $W_{12} + W_{22} + W_{32} + W_{42} \leq (K+P-1)y_j$
 - $1 + 0 + 1 + 0 \leq (1+2-1)(1)$
 - $\mathbf{2} \leq \mathbf{2}$

- $W_{13} + W_{23} + W_{33} + W_{43} \leq (K+P-1)y_j$
 - $0 + 1 + 0 + 1 \leq (1+2-1)(1)$
 - $2 \leq 2$
- $W_{14} + W_{24} + W_{34} + W_{44} \leq (K+P-1)y_j$
 - $0 + 0 + 0 + 0 \leq (1+2-1)(0)$
 - $0 \leq 0$

After implementing the above changes, CPLEX was then able to solve instances of problems.

4.2 TEST 1: COMPARING THE PERFORMANCE OF THE SELECTED SET OF SOLVERS ACROSS ALL PROPOSED ILP FORMULATION MODELS OF THE CLUSTERING PROBLEM

In the first test, the topologies were generated according to the configurations in Table 2.

Table 2

Test Set 1: Network Configurations to Evaluate the Performance of all Solvers Across all Proposed ILP Formulations

#Nodes	#Clusterheads	MaxClusterSize
5	3	1
7	3	2
9	3	3
11	3	4
13	3	5
15	3	6
7	4	1
9	4	2
11	4	3
13	4	3
15	4	4
9	5	1
11	5	2
13	5	2
15	5	3

The first test involved comparing the performance of all solvers in solving all proposed ILP formulations and enhancements for small scale network topologies. The topologies were generated according to the configurations in Table 2. For each configuration (row) in Table 2, 100 random topologies were generated for each solver (Total 5 Solvers) and the average time the solver took to solve each configuration for each model (FCB, SR, SR+IC, SR+MH) was noted. The timeout for each solver was set to be 15 minutes. A total of 5 Solvers * 4 models * 15 configurations * 100 instances = 30000 test cases were run. The results of the tests will be discussed in detail in Section 5.1.

4.3 TEST 2: TESTING THE ABILITY OF SOLVERS TO SOLVE ILP FORMULATIONS OF THE CLUSTERING PROBLEM FOR LARGE SCALE NETWORKS

The set of tests specified in Section 4.2 will allow for the comparison of the performance of the entire selected set of solvers, and will also allow each proposed Integer Linear Programming (ILP) formulation and enhancement to be analysed from a feasibility perspective. However, a test is required to determine how the solvers fare in larger scale networks. The next set of tests to be carried out is to determine the scalability of the problems handled by the best solvers from the results of the previous test. The configurations for this set of tests are shown in Table 3.

In this set of tests, in order to test the ability of the selected solvers to handle networks of a larger scale, topologies ranging from 10 nodes to a maximum of 50 nodes will be generated in increments of 5 nodes. For each configuration, 100 random topologies will be generated. The average time taken to solve the 100 instances of each configuration will be found. The timeout will again be set to a period of 15 minutes (or 900 seconds).

For this test, only the standard Star Ring Base Model model with no enhancements was used. As with the previous test set, different clustering configuration for the same total node count were generated (15 node topologies with 3, 4 and 5 clusterheads) to determine if the clustering configuration had an impact on the time taken by the solver to cluster the topology.

Table 3

Test Set 2: Network Configurations to Test the Ability of Solvers to Handle Large Networks

#Nodes	#Clusterheads	MaxClusterSize
7	3	2
10	3	4
15	3	6
20	3	9
25	3	11
30	3	14
35	3	16
40	3	19
45	3	21
50	3	24
7	4	1
10	4	2
15	4	4
20	4	6
25	4	7
30	4	9
35	4	11
40	4	12
45	4	14
50	4	16
9	5	1
10	5	2
15	5	2
20	5	4
25	5	5
30	5	7
35	5	8
40	5	9
45	5	10
50	5	12

CHAPTER 5

EVALUATION OF RESULTS

In this section, the results of the tests specified in Section 4.2 and 4.3 will be shown in a tabular format and as comparison graphs. The results will subsequently be discussed in detail and the relationship between the performance of solvers and aspects of the Integer Linear Programming (ILP) formulation will be analyzed. All experiments were conducted on an Intel Xeon 3.2 Ghz workstation running Linux with 4 GB of RAM.

5.1 TEST 1: RESULTS

In this section, the results of the tests described in Section 4.2 will be detailed. As specified in Section 4.2, all solvers were used to solve a set of instances of each ILP formulation and enhancement. The results obtained will be broken down by ILP formulation in order to better understand and interpret the results. It is important to keep in mind that for each network configuration shown (for all tests in Table 4 Table 5, Table 6 and Table 7), 100 random topologies were generated, and their ILP formulations were solved by each solver. The times in the subsequent tables are the averages of the times taken to solve the 100 instances. A “Timeout” occurs when the solver attempted to solve the problem and was unable to solve it within the 15 minutes (900seconds) allotted. A “Cannot Solve” occurs when the solver is unable to attempt to solve the problem. Detailed evaluation of this will be discussed in the subsequent sections. The following abbreviations will be used throughout this section: N – Nodes, CH-Clusterheads, MCS-Maximum Cluster Size.

5.1.1 TEST 1: EEC-FCB MODEL RESULTS

The following are the results obtained by executing all the test instances for the Energy Efficient Clustering - Fully Connected Backbone (EEC- FCB) model. The network configurations of the test instances and corresponding times for each solver are shown in Table 4. The results in Table 4 are also converted into charts and shown in Figure 25, Figure 26 and Figure 27.

Table 4

Test 1 Results: Time Taken by all Solvers to Solve Network Configurations using the EEC-FCB Model

#N	#CH	#MCS	CPLEX	SCIP	BSOLO	Pueblo	Minisat+
5	3	1	0.019	0.022	0.006	0.001	0.040
7	3	2	0.030	0.047	0.129	0.012	0.345
9	3	3	0.070	0.187	1.338	0.053	4.428
11	3	4	0.160	0.609	26.432	0.377	60.101
13	3	5	0.420	2.884	Timeout	CannotSolve	571.607
15	3	6	0.617	6.977	Timeout	CannotSolve	Timeout
7	4	1	0.377	0.090	0.110	0.031	0.585
9	4	2	0.369	0.286	1.637	0.071	9.865
11	4	3	0.453	0.708	24.093	0.682	190.022
13	4	3	0.605	1.962	Timeout	CannotSolve	Timeout
15	4	4	0.820	3.613	Timeout	CannotSolve	Timeout
9	5	1	0.175	0.817	1.144	0.074	14.352
11	5	2	0.202	1.281	24.981	0.531	236.457
13	5	2	0.280	2.048	Timeout	CannotSolve	Timeout
15	5	3	0.423	4.395	Timeout	CannotSolve	Timeout

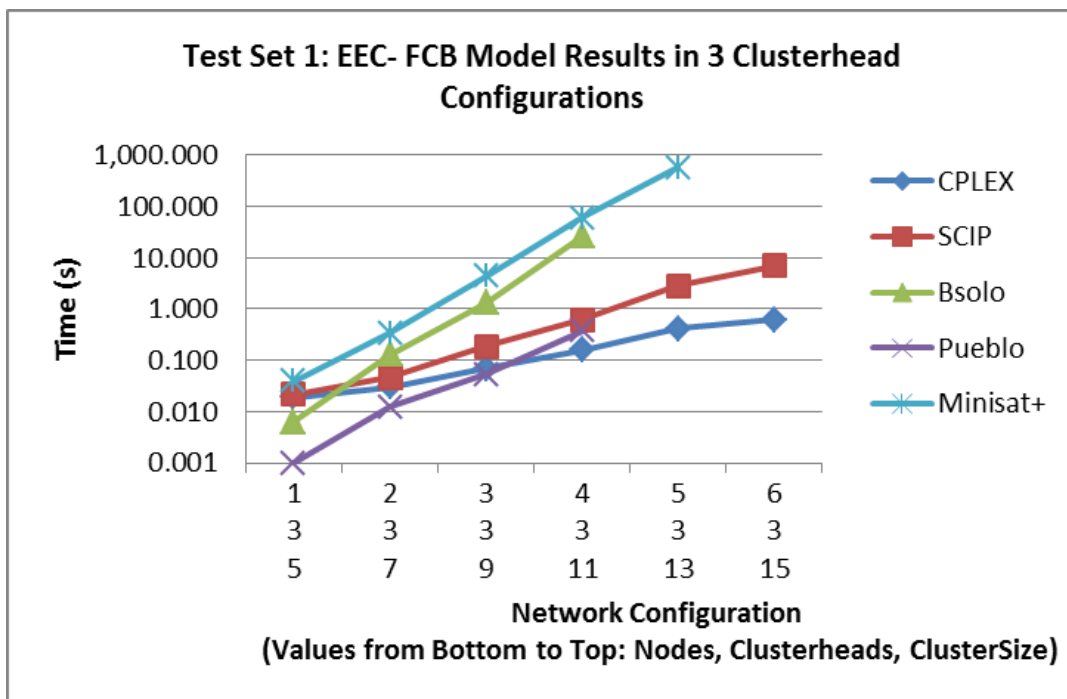


Figure 25: Test 1 Results: EEC-FCB Model in Configurations with 3 Clusterheads.

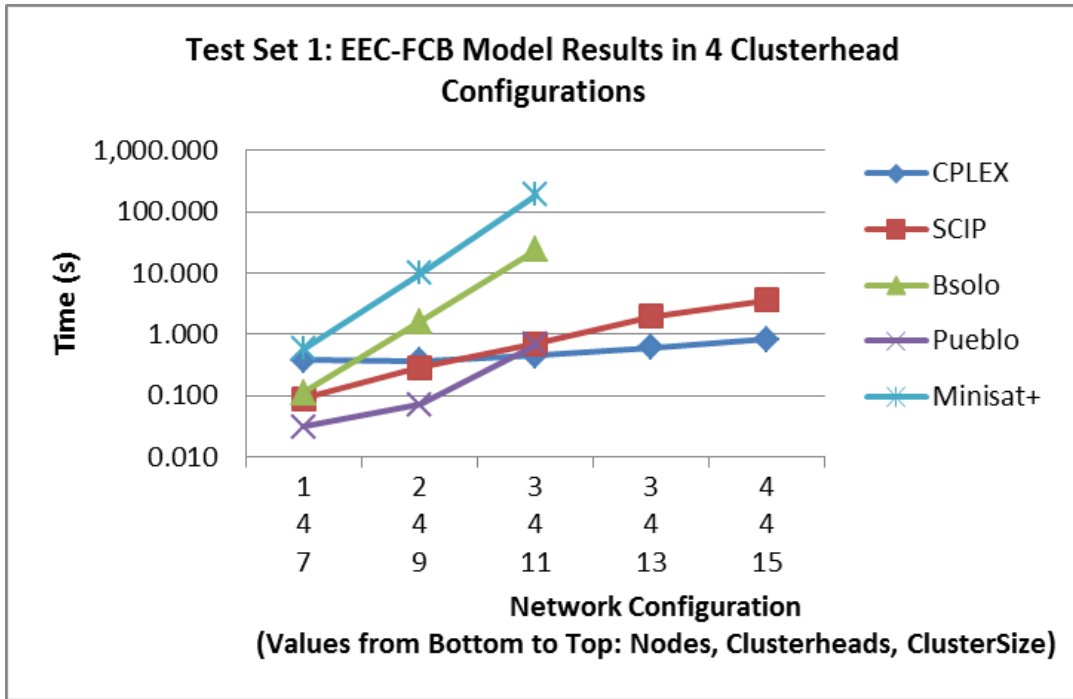


Figure 26: Test 1 Results: EEC-FCB Model in Configurations with 4 Clusterheads.

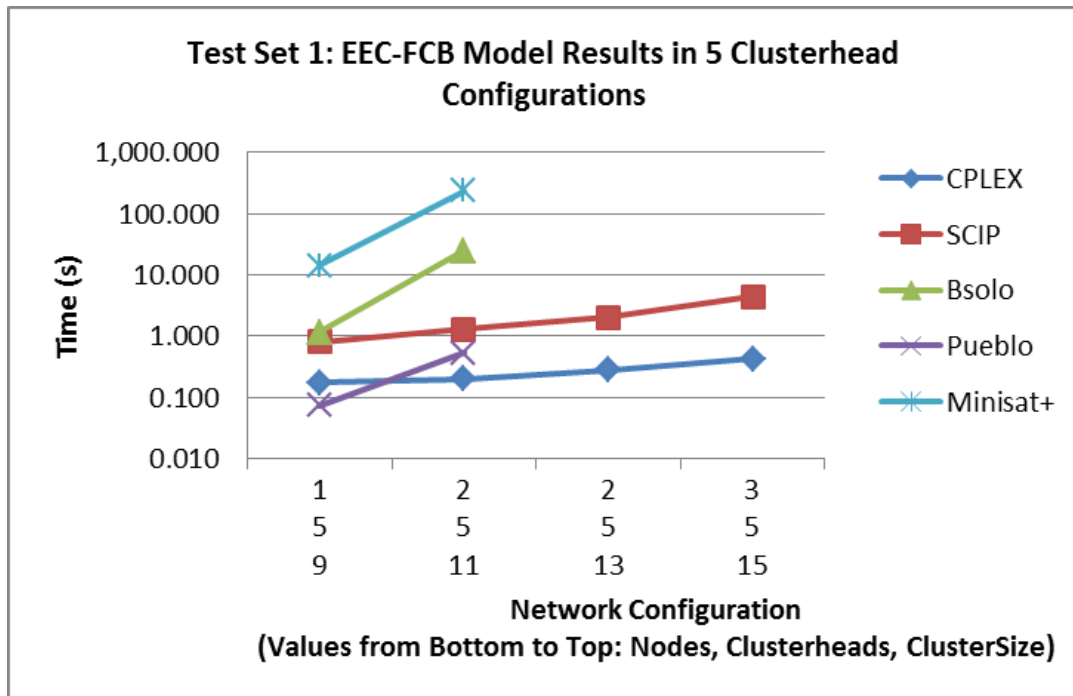


Figure 27: Test 1 Results: EEC-FCB Model in Configurations with 5 Clusterheads.

5.1.2 TEST 1: SR MODEL RESULTS

The following are the results obtained by executing all the test instances for the Star-Ring (SR) model. The network configurations of the test instances and corresponding times for each solver are shown in Table 5.

Table 5

Test 1 Results: Time Taken by all Solvers to Solve Network Configurations using the SR Model

#N	#CH	#MCS	CPLEX	SCIP	BSOLO	Pueblo	Minisat+
5	3	1	0.257	0.014	0.002	0.001	0.038
7	3	2	0.285	0.023	0.007	0.008	0.180
9	3	3	0.376	0.060	0.026	0.040	1.180
11	3	4	0.468	0.148	0.063	0.349	5.793
13	3	5	0.637	0.428	0.281	CannotSolve	31.852
15	3	6	0.725	1.017	0.950	CannotSolve	242.520
7	4	1	0.356	0.051	0.013	0.011	0.378
9	4	2	0.546	0.150	0.055	0.072	6.736
11	4	3	0.571	0.297	0.152	0.531	76.198
13	4	3	0.795	0.967	1.030	CannotSolve	349.541
15	4	4	0.903	1.709	4.753	CannotSolve	Timeout
9	5	1	0.532	0.353	0.098	0.107	8.595
11	5	2	0.745	0.900	0.366	1.833	200.956
13	5	2	0.834	2.058	1.633	CannotSolve	Timeout
15	5	3	1.035	3.212	7.433	CannotSolve	Timeout

The results in Table 5 are also converted into charts. The charts are all formatted as follows. The time on the y axis is on a logarithmic scale. The network configuration is shown on the x axis (Nodes, Clusterheads and Clustersize). The charts are shown in Figure 28, Figure 29 and Figure 30.

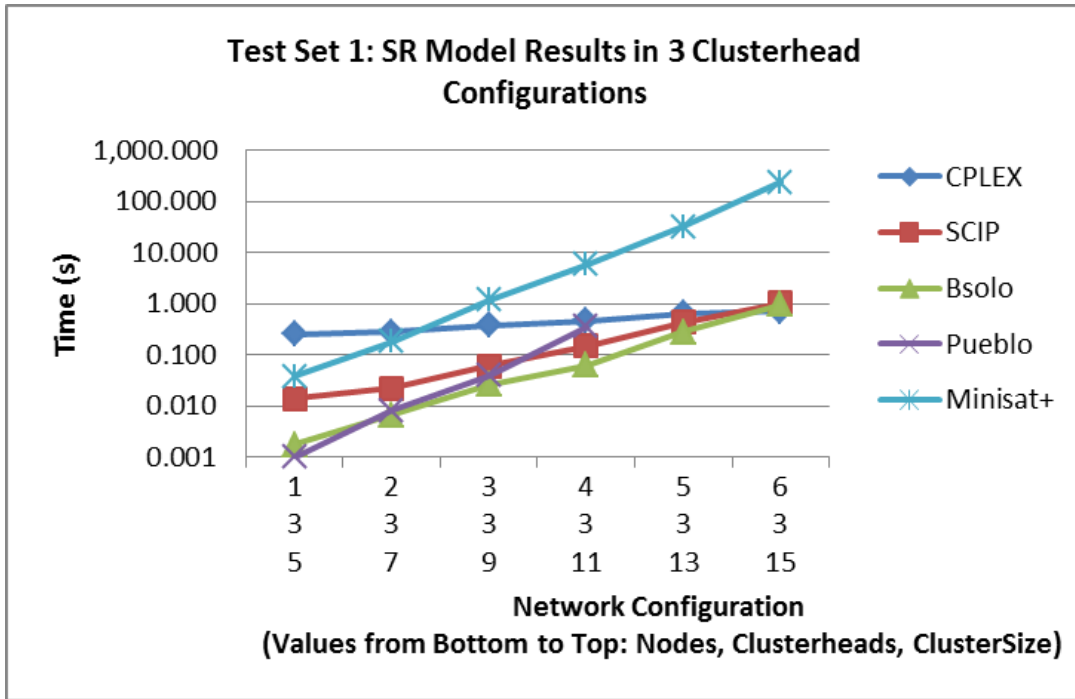


Figure 28: Test 1 Results: SR Model in Configurations with 3 Clusterheads.

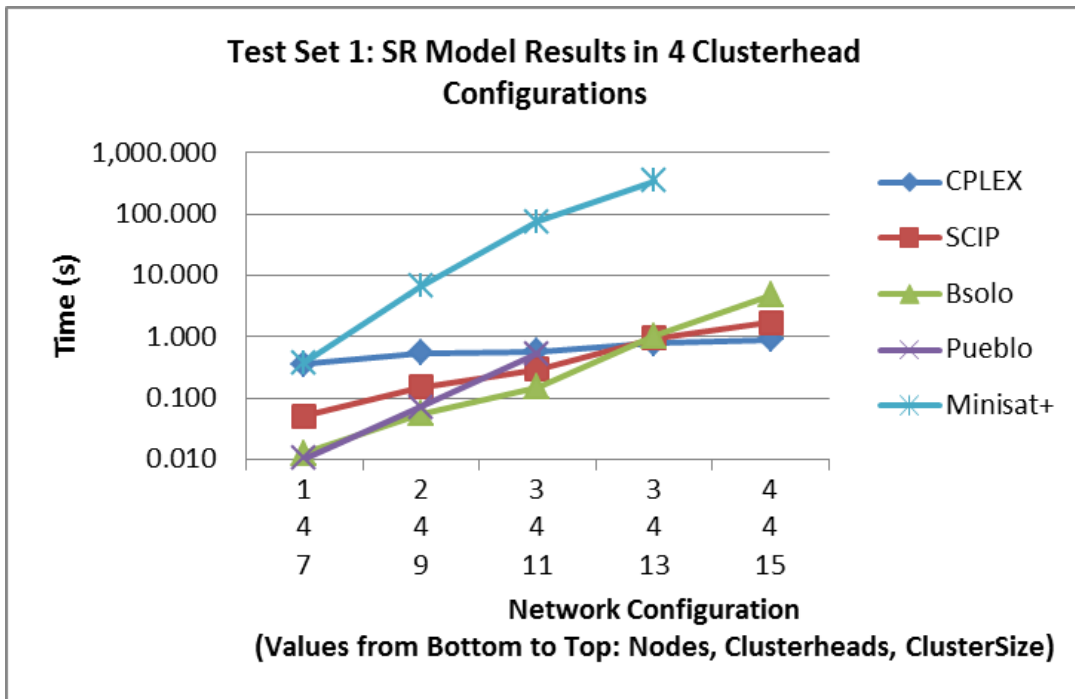


Figure 29: Test 1 Results: SR Model in Configurations with 4 Clusterheads.

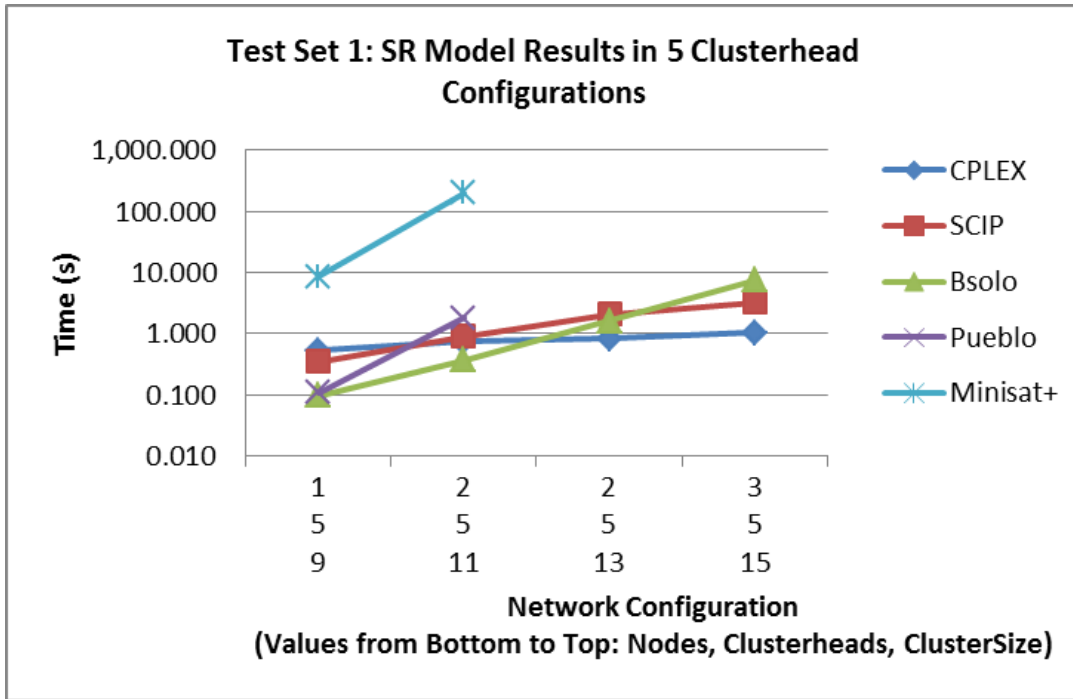


Figure 30: Test 1 Results: SR Model in Configurations with 5 Clusterheads.

5.1.3 SR+ IC MODEL TESTS AND RESULTS

The results obtained by executing all the test instances for the Star-Ring with Intra-Cluster Communication (SR+IC) model are shown in Table 6. The results in Table 6 are also converted into charts. The charts are shown in Figure 31, Figure 32 and Figure 33

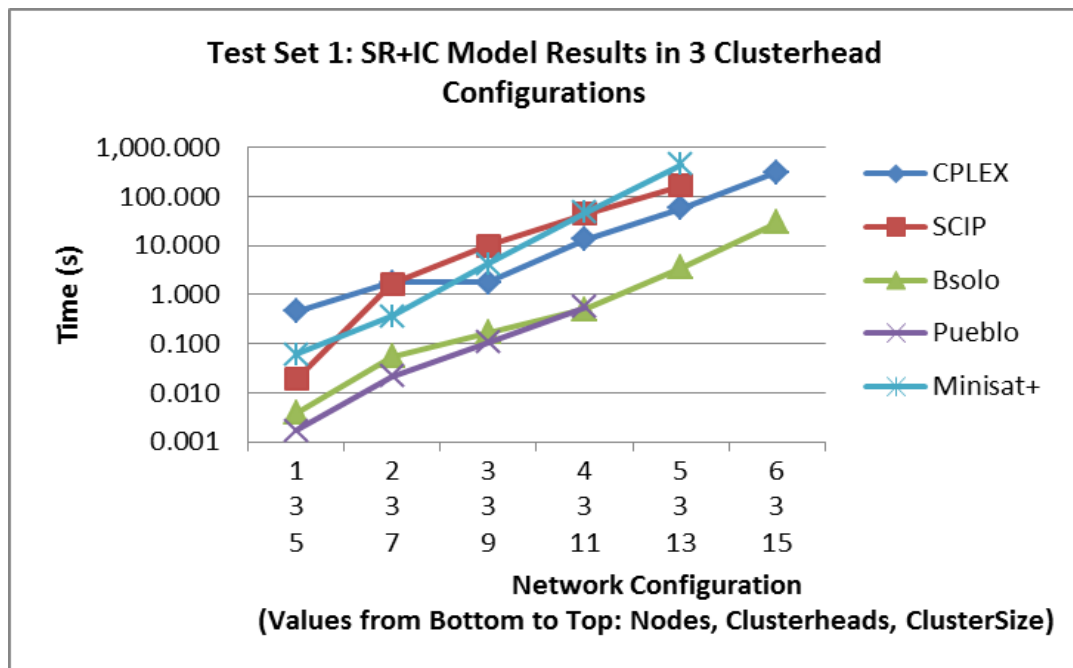


Figure 31: Test 1 Results: SR+IC Model in Configurations with 3 Clusterheads.

Table 6

Test 1 Results: Time Taken by all Solvers to Solve Network Configurations using the SR+IC Model

#N	#CH	MCS	CPLEX	SCIP	BSOLO	Pueblo	Minisat+
5	3	1	0.459	0.019	0.004	0.002	0.061
7	3	2	1.810	1.657	0.055	0.022	0.366
9	3	3	1.810	10.168	0.172	0.106	4.244
11	3	4	13.701	44.178	0.500	0.566	48.341
13	3	5	58.213	167.485	3.642	CannotSolve	453.634
15	3	6	310.034	Timeout	30.127	CannotSolve	Timeout
7	4	1	0.354	0.070	0.022	0.025	0.764
9	4	2	1.463	5.791	0.207	0.139	25.139
11	4	3	5.543	28.558	0.645	1.066	367.870
13	4	3	31.538	116.231	9.057	CannotSolve	Timeout
15	4	4	119.611	589.973	74.844	CannotSolve	Timeout
9	5	1	0.898	0.387	0.132	0.174	22.154
11	5	2	4.197	16.867	1.047	2.462	456.812
13	5	2	28.365	71.119	7.314	CannotSolve	Timeout
15	5	3	75.028	204.694	74.363	CannotSolve	Timeout

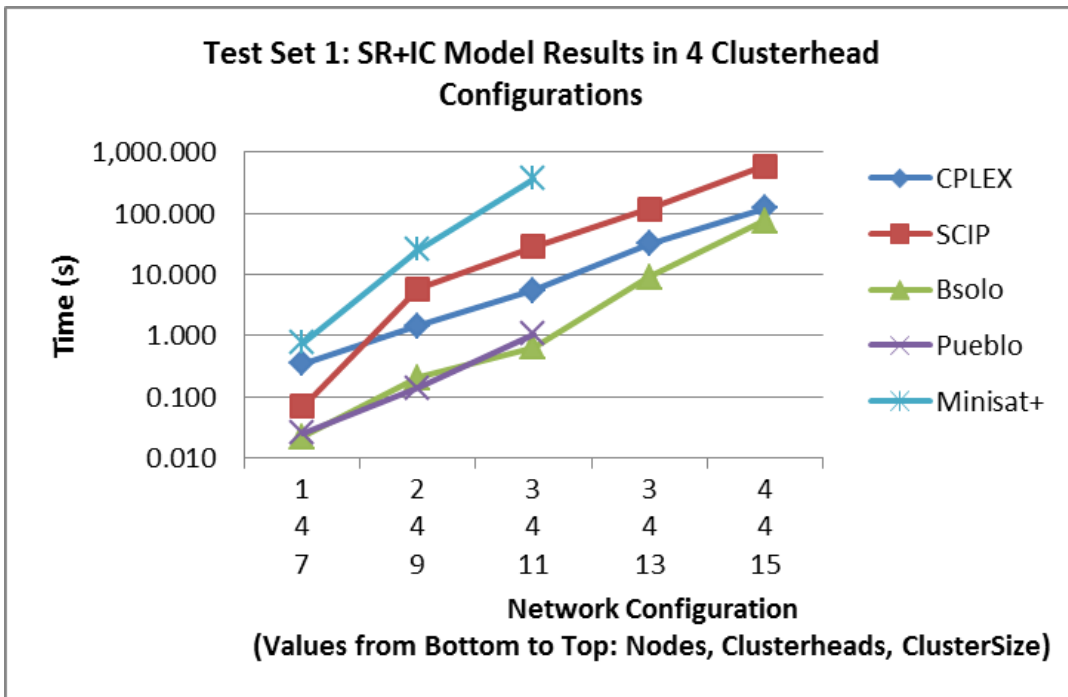


Figure 32: Test 1 Results: SR+IC Model in Configurations with 4 Clusterheads.

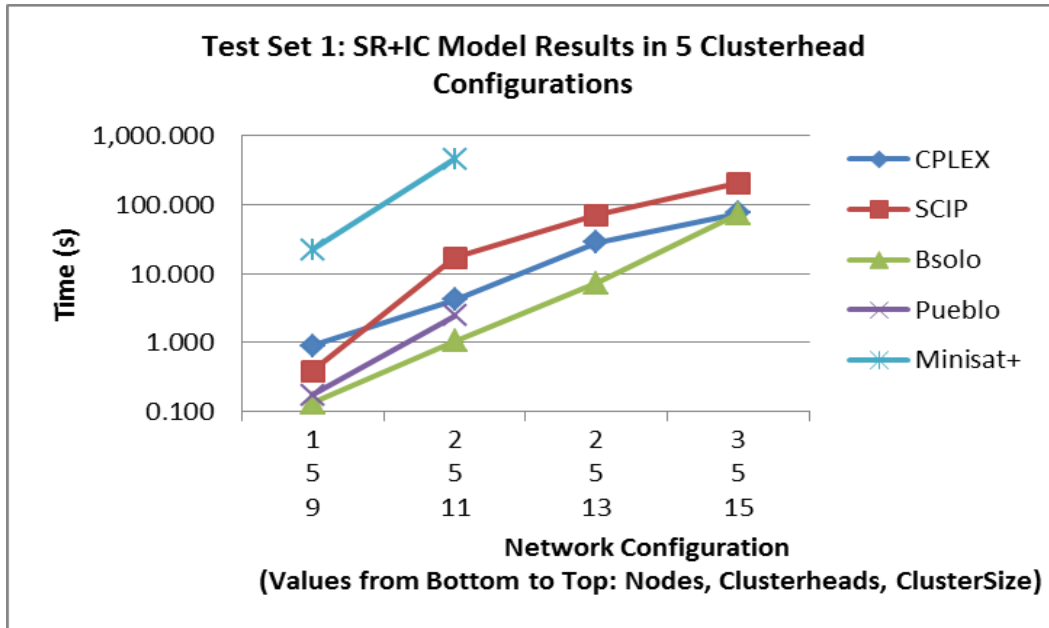


Figure 33: Test 1 Results: SR+IC Model in Configurations with 5 Clusterheads.

5.1.4 SR+ MH MODEL TESTS AND RESULTS

Table 7

Test 1 Results: Time Taken by all Solvers to Solve Network Configurations using the SR+MH Model

#N	#CH	#MCS	CPLEX	SCIP	BSOLO	Pueblo	Minisat+
5	3	1	0.277	0.038	0.007	CannotSolve	0.372
7	3	2	0.600	2.005	0.187	CannotSolve	237.852
9	3	3	1.621	9.485	2.451	CannotSolve	Timeout
11	3	4	11.116	28.294	75.725	CannotSolve	Timeout
13	3	5	50.044	125.698	765.245	CannotSolve	Timeout
15	3	6	168.975	329.926	Timeout	CannotSolve	Timeout
7	4	1	0.371	0.137	0.036	CannotSolve	7.098
9	4	2	1.897	14.021	1.251	CannotSolve	Timeout
11	4	3	11.911	30.518	30.351	CannotSolve	Timeout
13	4	3	50.225	150.820	688.563	CannotSolve	Timeout
15	4	4	190.577	401.778	Timeout	CannotSolve	Timeout
9	5	1	0.705	0.599	0.231	CannotSolve	135.943
11	5	2	12.102	68.562	8.449	CannotSolve	Timeout
13	5	2	69.610	255.456	310.908	CannotSolve	Timeout
15	5	3	138.273	312.762	Timeout	CannotSolve	Timeout

The results obtained by executing all the test instances for the Star-Ring with Multihop connections (SR+MH) model are shown in Table 7. The results in Table 7 are also converted into charts. The charts are shown in Figure 34, Figure 35 and Figure 36.

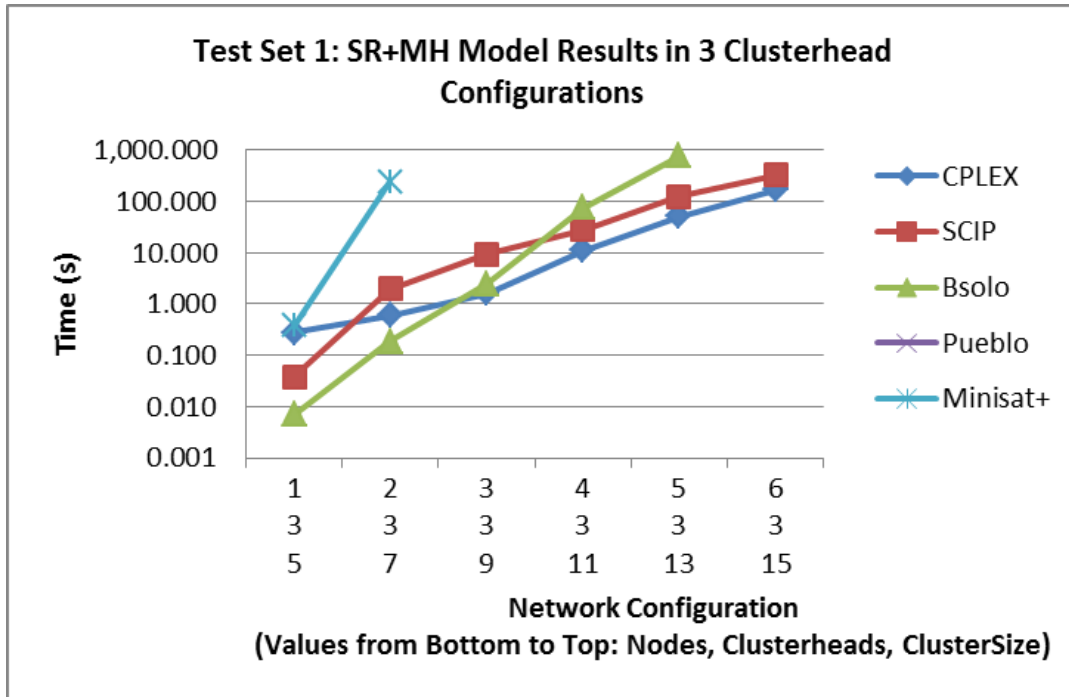


Figure 34: Test 1 Results: SR+MH Model in Configurations with 5 Clusterheads.

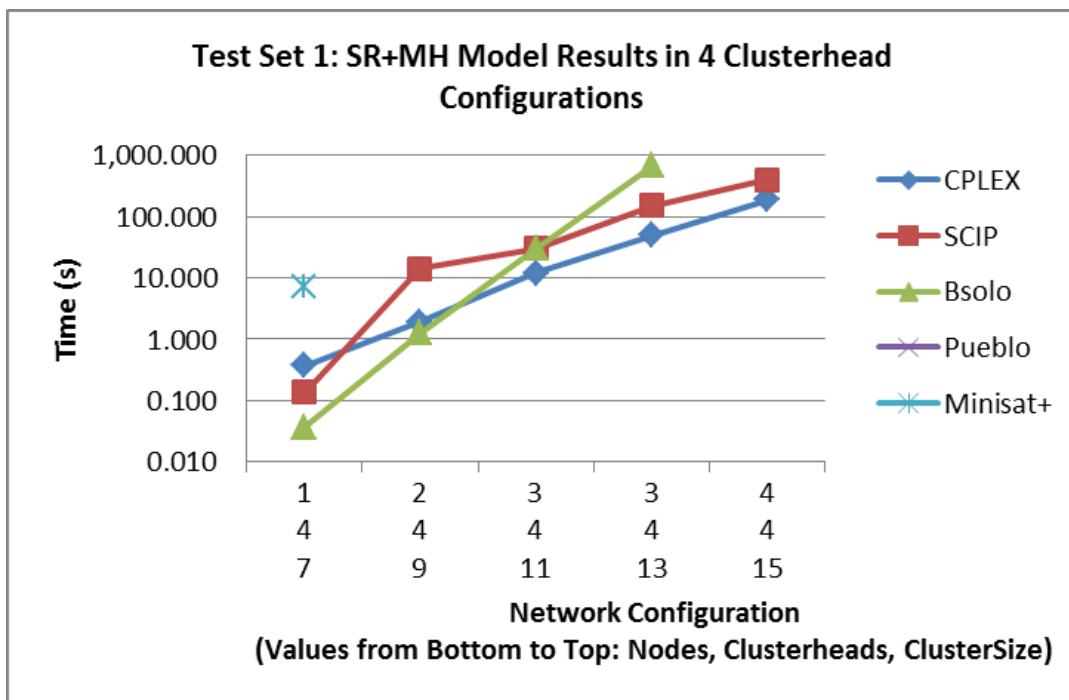


Figure 35: Test 1 Results: SR+MH Model in Configurations with 4 Clusterheads.

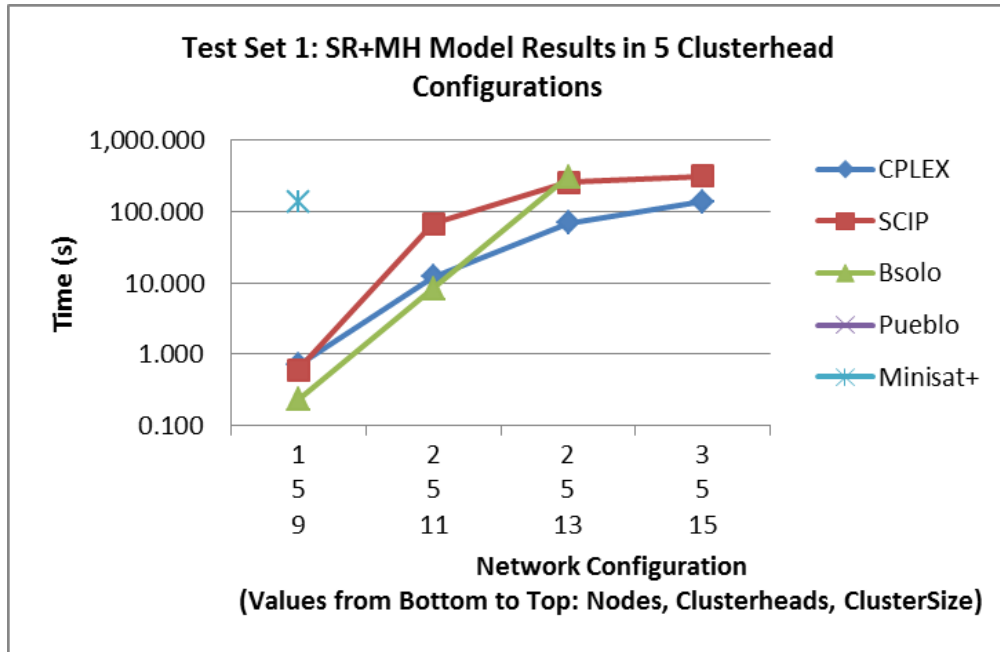


Figure 36: Test 1 Results: SR+MH Model in Configurations with 5 Clusterheads.

5.2 EVALUATION OF THE RESULTS OF TEST 1

In this section, based on the results obtained and shown in Section 5.1, the performance of the solvers will be discussed, and the trends and dependencies observed will be highlighted

5.2.1 GENERAL OBSERVATIONS

The following general observations can be made from the results of Test 1:

- Pueblo [46] is unable to handle certain instances and ends up in the “Cannot Solve” state. This is due to Pueblo’s inability to handle large coefficients. The large coefficients present are the costs associated with interconnecting nodes. (The cost of the link connecting a regular node to a clusterhead is proportional to the square of the distance between the nodes, and the cost of interconnecting clusterheads is proportional to the cube of the distance between the clusterheads [34].)
- Star-Ring (SR) model solutions are obtained faster than SR ‘enhanced’ with Intra-Cluster communication (SR+IC) or Multihop connections (SR+MH). In the case of SR+IC, solving the formulation takes much longer than SR, because not only does the optimal set of clusterheads needs to be identified, but it needs to be identified in conjunction with the

optimal connections between regular nodes and clusterheads since nodes need to connect to each other within the same cluster. The search space for the solutions is the same, but the process of identifying nodes connected to the same clusterhead, and accounting for their cost is computationally intensive. In the case of SR+MH, the Multihop formulation is the most computationally intensive and takes the most time to solve. This is for several reasons. The first is that the search space is increased. Additional solutions are possible as compared to the SR models because nodes can hop and connect to a clusterhead. The second reason is that calculating the cost of Multihop paths is very computationally intensive as it is done through a lengthy set of equations.

- CPLEX [37] and SCIP [42] perform well overall, with MINISAT+ [45] being the slowest solver. Among the set of selected solvers, CPLEX and SCIP handle the larger networks well as they almost never timeout.

5.2.2 MODEL SPECIFIC OBSERVATIONS

The following model-specific observations can be made from the results of Test 1:

- In the case of the Energy Efficient Clustering – Fully Connected Backbone (EEC-FCB) and the Star-Ring (SR) model, the Boolean Satisfiability (SAT) Solvers BSOLO [44], and in particular Pueblo [46] are very fast for the smaller networks, however as the size of the network increases, their time-to-solve increases faster than the generic ILP solvers; CPLEX [37] and SCIP [42]. MINISAT+ times out and CPLEX and SCIP are the fastest solvers for EEC-FCB and SR models.
- In the case of the Star-Ring with Intra-Cluster communication (SR+IC) Model, BSOLO emerges as the fastest solver, in some cases by a large margin. CPLEX is the next fastest, with the others far behind.
- In the case of the Multihop model, only CPLEX and SCIP do not timeout. BSOLO solves the smaller topologies well but is unable to handle the larger topologies. Pueblo is unable to handle any of the test cases because of the large coefficients generated in the Multihop enhanced ILP formulations.
- In the case of the SR model, it is observed that for a given number of nodes, all solvers take a longer time to solve topologies which have a larger number of clusterheads. This is shown in Figure 37 for the case of the CPLEX solver. The other solvers behave similarly. It can be seen clearly, that for a fixed number of nodes, if topologies with a larger number of clusterheads is to be generated, the solvers will take more time to generate the solution.

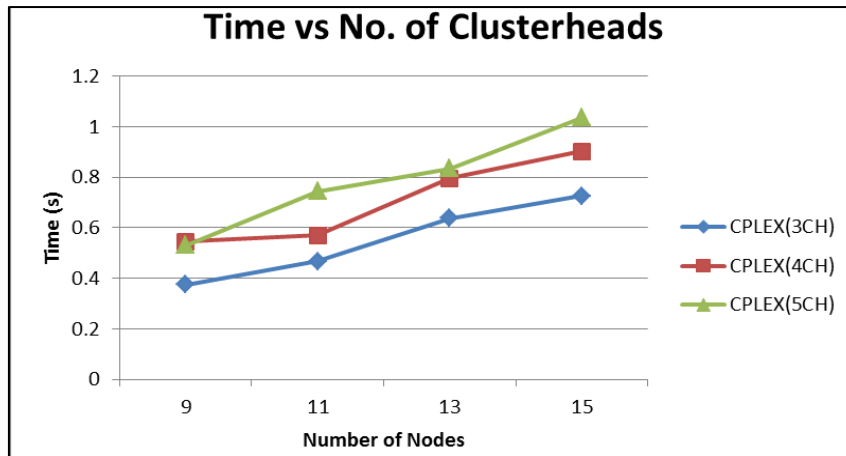


Figure 37: The Effect of Increasing the Number of Clusterheads in a Fixed Size Topology on the Time Taken by Solvers to Solve the Corresponding SR ILP Formulation.

- In the case of the SR+IC model, it is not the number of clusterheads but rather the clustersize which increases the time taken by all solvers to solve topologies of a given number of nodes. For the same network topology of 15 nodes, the average time taken to solve it with 3 clusterheads and a cluster size of 6 takes longer than solving it for a configuration of 4 clusterheads and a Cluster Size of 4, which again takes longer than solving it for a configuration of 5 clusterheads and a cluster size of 3. This is shown in the Figure 38 which summarizes the times taken by CPLEX to solve the different test configurations. The configurations are broken down by the total number of nodes in each topology. As can be seen, increasing the clustersize, while keeping the number of nodes constant results in higher time required to solve the SR+IC formulation.

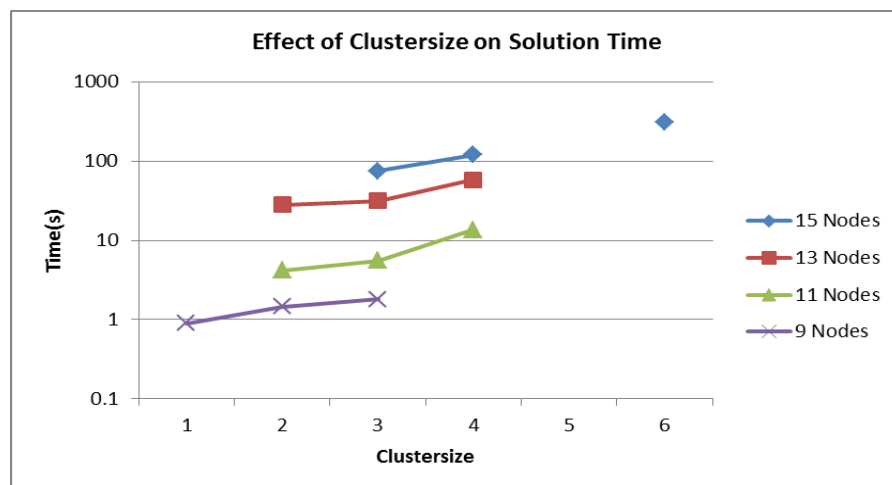


Figure 38: The Effect of Increasing the Clustersize in a Fixed Size Topology on the Time Taken by Solvers to Solve the Corresponding SR+IC ILP Formulation.

5.3 TEST 2: RESULTS

The results of the tests described in Section 4.3 are shown in Table 8.

Table 8

Results of Test Set 2: Testing the Ability of Solvers to Handle Large Networks

Nodes	Clusterheads	Cluster size	CPLEX	SCIP
10	3	4	0.033	0.0837
15	3	6	0.1234	0.8615
20	3	9	0.3372	4.3063
25	3	11	0.7118	12.0543
30	3	14	1.5052	25.9856
35	3	16	2.74	51.9671
40	3	19	5.7374	121.1817
45	3	21	8.274	162.5868
50	3	24	16.2945	303.2534
10	4	2	0.6089	0.273
15	4	4	0.8484	1.8967
20	4	6	1.2237	7.7969
25	4	7	2.2824	22.9498
30	4	9	4.6528	58.1925
35	4	11	10.4126	143.369
40	4	12	45.3313	304.8588421
45	4	14	94.3953	385.2515116
50	4	16	271.929	665.64
10	5	2	0.856	1.324
15	5	3	1.0919	3.5254
20	5	4	0.6938	12.9578
25	5	5	2.7112	33.5338
30	5	7	7.8036	77.4732
35	5	8	20.7042	172.2333
40	5	9	112.9257	489.40
45	5	10	195.1731	Timeout
50	5	12	427.5185	Timeout

As specified in Section 4.3, the two solvers which performed well in Test 1 were tested with large scale topologies. These two solvers were CPLEX [37] and SCIP [42] as both were able to handle the 15 node topologies without timing out. The results shown in Table 8 are also shown in the charts in Figure 39, Figure 40 and Figure 41.

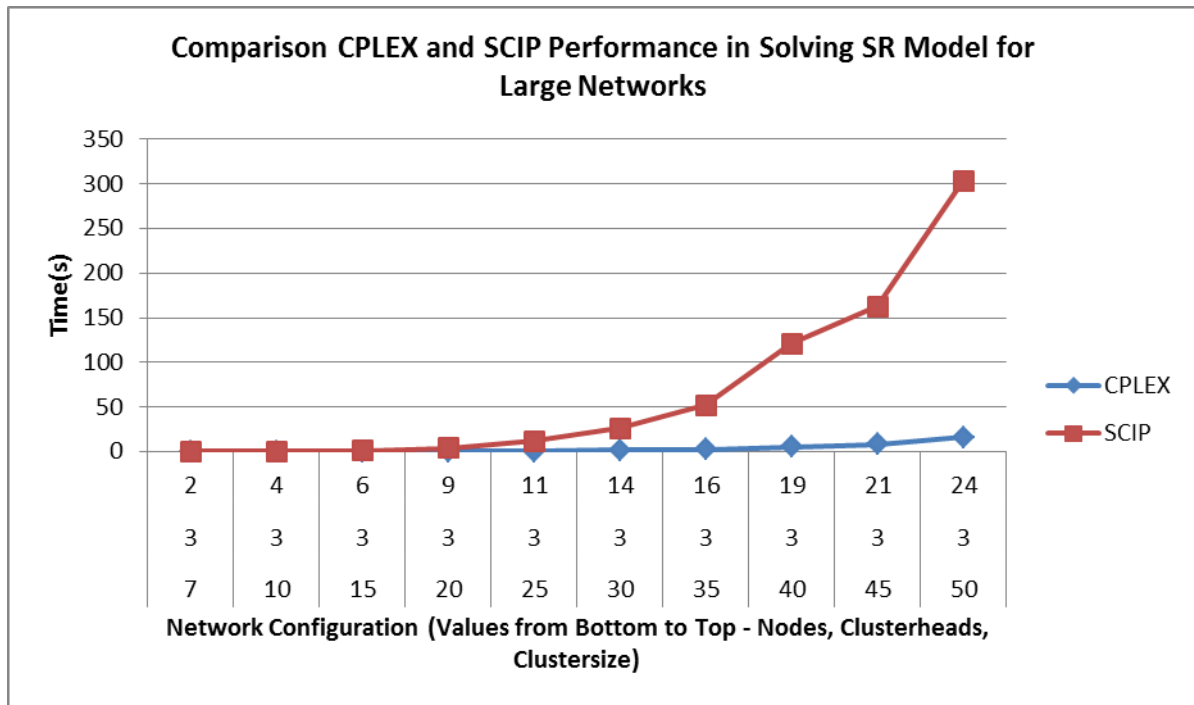


Figure 39: Test 2 Results: SR Model in Configurations with 3 Clusterheads.

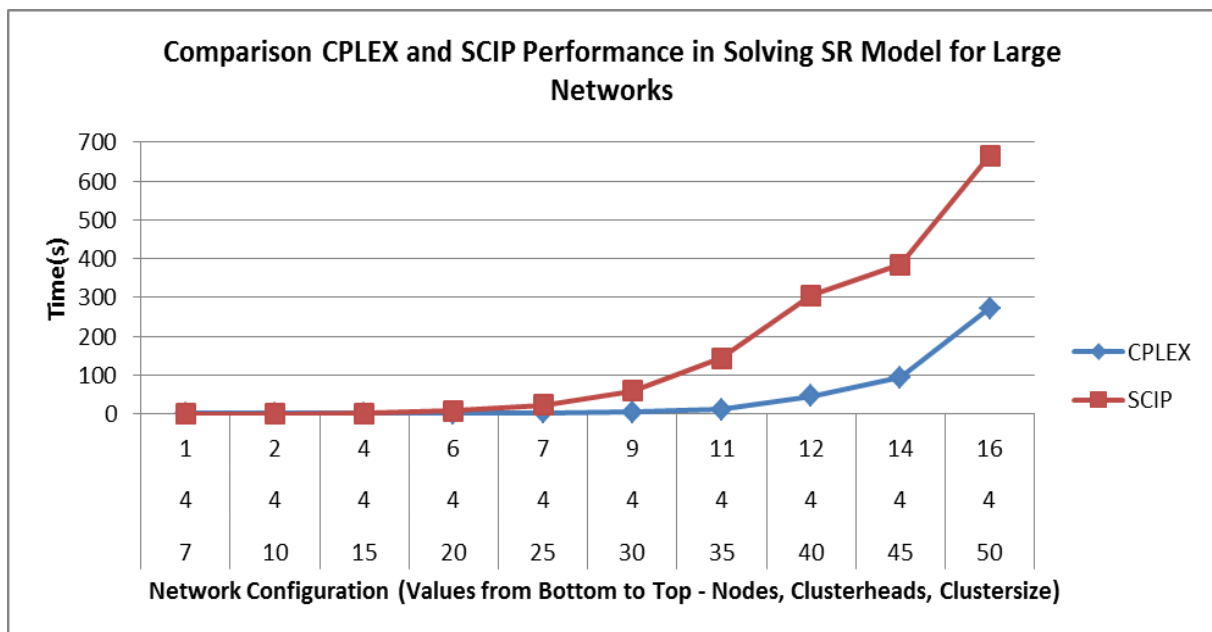


Figure 40: Test 2 Results: SR Model in Configurations with 4 Clusterheads.

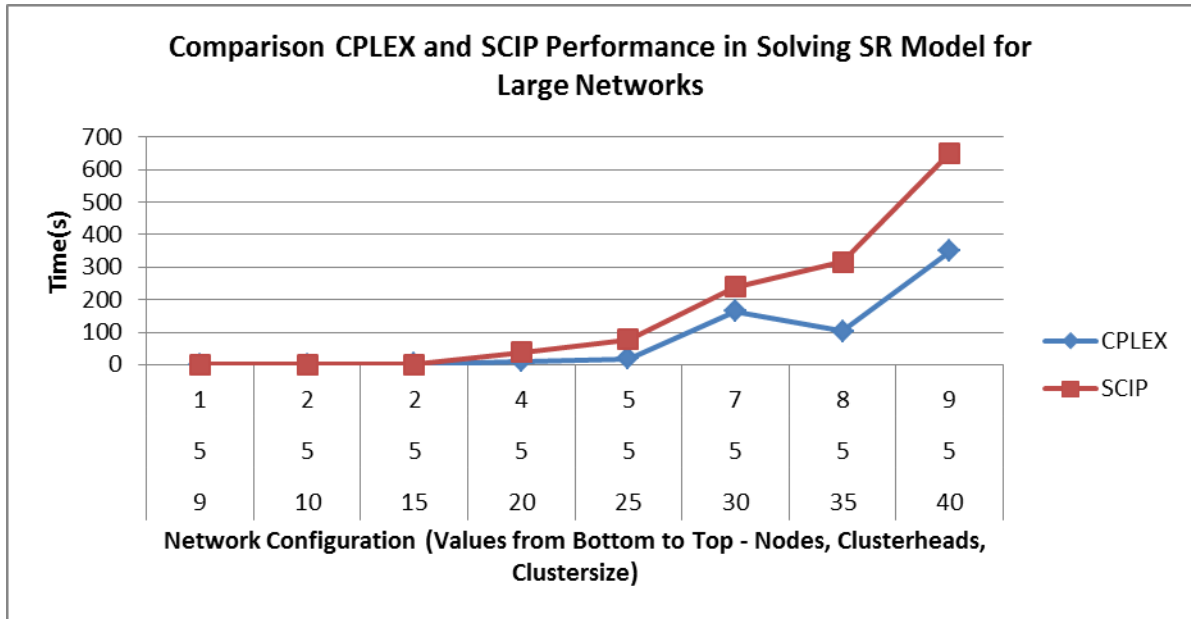


Figure 41: Test 2 Results: SR Model in Configurations with 5 Clusterheads.

5.4 EVALUATION OF THE RESULTS OF TEST 2

In this section, we will describe, analyse and interpret the results obtained in Test 2. Through the results obtained the following observations can be made:

- When using the SR Model to generate an ILP formulation of the clustering problem, both solvers are able to handle topologies up to 50 Nodes.
- As noted in the evaluations of the results of Test 1 in Section 5.2, when using the SR model, increasing the number of clusterheads increases the time required by the solver to solve the formulation. From the results of Test 2, this is confirmed with CPLEX [37] and SCIP [42] being able to solve 50 node topologies with 3 clusterheads and 4 clusterheads but timing out with 5 clusterheads at only 40 nodes.
- For configurations of 3 and 4 clusterheads, CPLEX is far from timing out at 50 nodes and should be able to handle even larger network topologies.

5.5 RECOMMENDATIONS FOR FUTURE TESTS

In this section, we will briefly describe additional tests which could be conducted in order to further study and possibly improve the performance of solvers in handling the proposed ILP formulations.

Dependencies on distribution of nodes in a topology require further study. Some topologies are dense, closely packed, while others are scattered. How this affects the time taken by the solver to generate the solution is something to test. Even when the coverage constraint is not applied the cost coefficients for scattered topologies are larger since the distance between the nodes is greater. Also, this affects Pueblo [46]’s ability to solve the problem.

Improving the performance of Pueblo, possibly trying to enable it to handle topologies with large coefficients is another aspect to further examine. One possibility to be investigated is the option of ‘equivalent topologies’. If a scattered topology can be scaled down to an equivalent relatively denser topology then the coefficients (which are proportional to the distance) will be smaller and Pueblo may be able to handle them.

5.6 FEASIBILITY OF USING GENERIC ILP OR PSEUDO-BOOLEAN SAT SOLVERS IN REAL WORLD CLUSTERING PROBLEM SOLVING

In this section, we will discuss the feasibility of using generic Integer Linear Programming (ILP) and Boolean Satisfiability (SAT) solvers to solve the clustering problem in MANETs, in a practical environment.

From the test results it can be seen that, as compared to the solver used in [34], solvers today are much faster and able to handle more complex ILP formulations of the clustering problem. The solver used to solve the ILP formulation presented in [34], timed out when solving for more than 9 node topologies. In the tests conducted with the proposed ILP formulations and enhancements, solvers such as CPLEX [37], SCIP [42] were able to handle ILP formulations of networks up to 50 nodes. Additionally, solvers including CPLEX, BSOLO [44] and SCIP were able to handle complex formulations such as the Multihop connections and Intra-Cluster communication enhancements proposed.

The timeout used in testing was 15 minutes (900 seconds) and does not accurately reflect a real-life setting, where clustering would need to be done much faster. It is also important to note, as mentioned before, that MANETs are generally small scale (as compared to Wireless Sensor Networks for example). For small scale networks the proposed ILP formulations and high performance solvers such as CPLEX would be suitable for use in real-world environments. However for large scale networks, as the time to cluster the network grows exponentially, the solvers will be unable to cluster the network in accordance with the demand of real-world environment.

CHAPTER 6

CONCLUSION

In this section we will summarize the results of our research. Over the course of this research, contributions were made in three key areas.

The first was the creation of the proposed Integer Linear Programming (ILP) formulation of the Clustering Problem for Mobile Ad-Hoc Networks (MANETs). This research presented the formulation of a Base Model, which was the Star-Ring Model, and proposed enhancements including coverage, Multihop and Intra-Cluster communication. Building on the work of [34], which was the first contribution in MANET modelling as an ILP problem, the Star-Ring model adds redundancy, bringing a compromise between the over-connected Energy Efficient Clustering – Fully Connected Backbone and the Energy Efficient Clustering - Connected Backbone (EEC-FCB/EEC-CB) models.

The second contribution consisted of the enhancements made to the proposed model. The proposed Coverage enhancement added constraints which allowed coverage restrictions to be taken into account, and solutions to be generated which did not violate the physical limitations of the node. The Intra-Cluster communication enhancement enabled nodes within the same cluster to communicate with each other directly without having to go through the clusterhead, allowing the clusterhead to conserve energy for Inter-Cluster communication. The Multihop connection enhancement enabled 2-hop connections which allow nodes to connect to clusterheads by ‘hopping’ using other closer nodes. The cost of establishing these connections was also accounted for. Additionally, formulations combining the various enhancements were generated, making it possible to have a model that enables Multihop connections, together with Intra-Cluster communication and at the same time, not violate the coverage constraints of the nodes. Also, through testing, conflicts arising from weaknesses in the implementation of the EEC-FCB model in [34], were identified and adjusted.

The third contribution was the development of the tool. The tool developed over the course of the research was able carry out a comprehensive set of functions. These functions included the ability to create custom MANET topologies in a user friendly manner, and the ability to connect and use different solvers. Additionally, the tool was able to use multiple

ILP formulations, create multiple instances for testing purposes, and print topologies with their corresponding solutions.

The third contribution involved using the proposed ILP formulations and enhancements together with the proposed tool, to test the performance and analyse the feasibility of generic ILP (CPLEX [37], SCIP [42]) and 0-1 SAT-based ILP solvers (BSOLO [44], Pueblo [46] and Minisat+ [45]) in solving the clustering problem for MANETs. CPLEX, and SCIP were able to handle the larger scale topologies and in most cases CPLEX was the fastest solver from the selected set of solvers. The exceptional case was the Star-Ring with Intra-Cluster Communication Model (SR+IC) model where BSOLO was the faster solver. For small scale networks, the Star-Ring (SR) ILP formulation and the CPLEX solver, can be used in a practical setting. However, the more complex formulations, including Intra-Cluster communication and Multihop connections, are too computationally intensive in their current state to be used in practical environments for larger networks as the time taken by the selected set of solvers to solve the enhanced formulations is too large to be feasible for practical environments.

CHAPTER 7

FUTURE WORK

In this section the key areas of future work and possible improvements in the three areas of research will be highlighted.

From a modelling perspective, while we have been able to create the SR model and devise enhancements which allow coverage, Multihop connections and Intra-Cluster communication, there are additional improvements which can be made. The key improvement would be the integration of the parameter which acts as the weight for selection of the clusterhead and Master clusterhead selection. Based on the model presented in [34], this parameter, b , is the weight obtained through the combination of factors such as mobility, residual energy and other factors which determine the suitability of the node to act as a clusterhead. At this point the value of b is being fed from an external source. Having a unified ILP formulation which also determines the value of this parameter, while computationally intensive, would be able to produce the optimum solution with all possible factors taken into account. Additionally, at present this parameter represents the combination of factors such as initial energy, residual energy, mobility and more to provide a single value indicating the capability of a node to handle the role of a clusterhead. Rather than using a single value, factors such as residual energy and mobility could be considered independently when generating the ILP formulation to more accurately represent the capabilities of the node. Using this unified model, it would be possible for the optimum solution, for any set of conditions and at any point in time of network operation, to be determined.

Additionally, another significant improvement would be to incorporate in the formulation, the ability to automatically determine the optimum number of clusterheads and also the optimum cluster size, both in terms of the cost of the solution generated and the time taken to find the solution. This would allow users the flexibility of being able to select a particular number of clusterheads and cluster size, or let the tool determine the best option for them.

From the perspective of the tool developed over the course of the research, there are several areas of improvement. At present, it is in its first version, with extensive changes and improvements still being made. These improvements are focused around increasing

functionality and providing a better user experience. Some of these improvements would include the ability to view solutions side by side and also to be able to connect and work with other solvers. One beneficial improvement would be the ability to adjust the topology to meet certain requirements such as coverage. If for example, a designed topology cannot be solved for user-specified requirements, the tool should be able to identify the nature of the conflict and adjust the topology or point out the adjustments required in order for a solution to be possible for the desired requirements (parameters).

Based on the results obtained from the tests conducted, there were several areas identified for improvement and further testing. Further tests could be conducted to assess the impact of the distribution of the nodes in a topology and their coverage radius on the solvers ability to solve the problem and the time taken to solve it. This is because the coefficients that determine the cost of connecting any two nodes (regular node to regular node, regular node to cluster head, clusterhead to clusterhead) are all dependent on the distance between the nodes.

This summarizes some of the potential improvements and points of future research with regard to ILP formulation of the clustering problem in MANETs, the development of the tool, and the feasibility of solvers in solving the ILP formulation of the clustering problem.

REFERENCES

- [1] Alexander Schrijver, *Theory of Linear and Integer Programming.*: John Wiley & Sons Ltd., 1999.
- [2] Sharad Malik and Lintao Zhang, "Boolean Satisfiability: From Theoretical Hardness to Practical Success," *Communications of the ACM - A Blind Person's Interaction with Technology*, vol. 52, no. 8, pp. 76-82, 2009.
- [3] F.A. Aloul, A. Ramani, I.L. Markov and K.A. Sakallah, "generic ILP versus specialized 0–1 ILP: an update," in *Proceedings of the International Conference on Computer-Aided Design (ICCAD)* , pp. 450–457, 2002.
- [4] (2011) The International SAT Competitions web page. [Online]. <http://www.satcompetition.org/>
- [5] Laurent Simon, Daniel Le Berre and Edward A. Hirsch. (2002, September) SAT2002 competition news. [Online]. <http://www.satcompetition.org/2002/online-report.pdf>
- [6] (2011) Pseudo-Boolean Competition 2011. [Online]. <http://www.cril.univ-artois.fr/PB11/>
- [7] Vasco Manquinho and Olivier Roussel. (2005, September) Pseudo Boolean Evaluation 2005. [Online]. <http://www.cril.univ-artois.fr/PB05/>
- [8] H. Dixon and M. Ginsberg, "Inference methods for a pseudo-boolean satisfiability solver," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)* , pp. 635–640, 2002.
- [9] Assim Sagahyoon and Fadi Aloul, "Using SAT-based techniques in power estimation," *Microelectronics Journal*, vol. 38, pp. 707-715, June 2007.
- [10] A.C. Ling, D.P. Singh and S.D. Brown, "FPGA Programmable logic block evaluation using quantified Boolean Satisfiability," *Computers and Digital Techniques IEEE Proceedings*, vol. 153, no. 3, pp. 165-172, May 2006.
- [11] Prahlad Fogla and Wenke Lee, "Evading Network Anomaly Detection Systems: Formal Reasoning and Practical Techniques," in *13th ACM Conference on Computer and Communications Security*, pp. 59-68, 2006.
- [12] Sunil Kim and Jun-yong Lee, "A system for high-speed deep packet inspection in signature-

- based network intrusion prevention," *Journal of Systems Architecture: The EUROMICRO Journal*, vol. 53, no. 5-6, pp. 310-320, May 2007.
- [13] Graham Hughes and Tefvik Bultan, "Automated Verification of Access Control Policies using a SAT solver," *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 10, no. 6, pp. 503-520, October 2008.
- [14] Ilya Mironov and Lintao Zhang, "Applications of SAT solver to Cryptanalysis of Hash Functions," in *International Conference on Theory and Applications of Satisfiability Testing*, pp. 102-115, 2006.
- [15] Fabio Massacci and Laura Marraro, "Logical Cryptanalysis as a SAT problem," *Journal of Automated Reasoning*, vol. 24, no. 1-2, pp. 165-203, February 2000.
- [16] S. Tosun, O. Ozturk, and M. Ozen, "An ILP formulation for application mapping onto Network-on-Chips," in *International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1-5, 2009.
- [17] Ana Graca, Ines Lynce, Joao Marques-Silva, and Arlindo L. Oliveira, "Haplotype inference with pseudo-Boolean optimization," in [*Annals of Operations Research*](#), 184(1), 137-162, 2011.
- [18] M.A. Centeno, R. Giachetti, R. Linn, and A.M. Ismail, "A simulation-ILP based tool for scheduling ER staff," in *Simulation Conference*, pp. 1930 - 1938, 2003.
- [19] Ahmed Wasfy and Fadi A. Aloul, "Solving the University Class Scheduling Problem," in *4th IEEE GCC Conference*, 2007.
- [20] JA Garcia-Macias and Javier Gomez, "MANET versus WSN," in *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications.*: Springer, ch. 17, pp. 369-388, 2006.
- [21] Suchismita Chinara and Santanu Kumar Rath, "Energy Efficient Mobility Adaptive Distributed Clustering Algorithm for Mobile Ad Hoc Network," in *16th International Conference on Advanced Computing and Communications*, pp. 265-272, 2008.
- [22] Kaixin Xu, Xiaoyan Hong, and Mario Gerla, "An Ad Hoc Network with Mobile Backbones," in *IEEE International Conference on Communications*, pp. 3138 - 3143, 2002.

- [23] Jamal N. Al-Karaki, Ahmed E. Kamal, and Raza Ul-Mustafa, "On the Optimal Clustering in Mobile Ad hoc Networks," in *Consumer Communications and Networking Conference*, pp. 71 - 76, 2004.
- [24] Hui Cheng, Jiannong Cao, Xingwei Wang, Sajal K. Das, and Shengxiang Yang, "Stability-aware multi-metric clustering in mobile ad hoc networks with group mobility," *Wireless Communications and Mobile Computing*, vol. 9, pp. 759-771, 2009.
- [25] Haidar Safa, Omar Mirza, and Hassan Artail, "A Dynamic Energy Efficient Clustering algorithm for MANETs," in *IEEE International Conference on Wireless & Mobile Computing, Networking and Communication*, pp. 51 - 56, 2008.
- [26] Roxana Zoican, "An Enhanced Performance Clustering Algorithm for MANET," in *15th IEEE Mediterranean Electrotechnical Conference*, pp. 1269 - 1272, 2010.
- [27] Wasim El-Hajj, Dionysios Kountanis, Ala Al-Fuqaha, and Mohsen Guizani, "A Fuzzy-Based Hierarchical Energy Efficient Routing Protocol for Large Scale Mobile Ad hoc Networks," in *IEEE International Conference on Communications*, pp. 3585-3590, 2006.
- [28] Wassim El-Hajj, Zouheir Trabelsi, and Dionysios Kountanis, "Fast Distributed dominating set based routing in large scale MANETs," *Computer Communications*, vol. 30, pp. 2880-2891, 2007.
- [29] Hui-Yao An, Ling Zhong, Xi-Cheng Lu, and Wei Peng, "A Cluster-Based Multipath Dynamic Source Routing in MANET," in *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, pp. 369 - 376, 2005.
- [30] Jing Wu, Go-chang Gu, and Guo-zhao Hou, "A Clustering Algorithm to Construct a Hierarchical Network Structure with Global Stability for Mobile Ad Hoc Networks," in *IEEE International Symposium on Knowledge Acquisition and Modeling Workshop*, pp. 301 - 304, 2008.
- [31] Abdelmajid Hajami, Kamal Ouididi, and Mohammed ElKoutbi, "An enhanced algorithm for MANET clustering based on Multi hops and Network Density," in *10th Annual International Conference on New Technologies of Distributed Systems*, pp. 181 - 188, 2010.
- [32] Giacomo Angione, Paolo Bellavista, Antonio Corradi, and Eugenio Magistretti, "A k-hop Clustering Protocol for Dense Mobile Ad-Hoc Networks," in *26th IEEE International*

Conference on Distributed Computing Systems Workshops, p. 10, 2006.

- [33] Zang Chunhua and Tao Cheng, "A Multi-hop Cluster Based Routing Protocol for MANET," in *International Conference on Information Science and Engineering*, pp. 2465 - 2468, 2009.
- [34] Wasim El-Hajj, Dionysios Kountanis, Ala Al-Fuqaha, and Hani Harbi, "Optimal hierarchical Energy Efficient Design for MANETs," in *International Conference on Communications and Mobile Computing*, pp. 287-292, 2006.
- [35] Xiaohua Jia and Hai Liu, "Linear Programming Approaches to Optimization Problems of Energy Efficiency in Wireless Ad Hoc Networks," in *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, Jie Wu, Ed.: CRC Press LLC, pp. 177-192, 2005.
- [36] Alexei Barbosa de Aguiar, Alvaro de M.S. Neto, Placido Rogerio Pinheiro, and Andre L. V. Coelho, "Applicability of a Novel Integer Programming Model for Wireless Sensor Networks," *International Journal of Computer Science and Information Security*, vol. 3, no. 1, pp. 7-13, 2009.
- [37] IBM. (2011) IBM ILOG CPLEX Optimizer. [Online]. <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>
- [38] Ali Chamam and Samuel Pierre, "On the Planning of Wireless Sensor Networks: Energy-Efficient Clustering under the Joint Routing and Coverage Constraint," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1077-1086, August 2009.
- [39] Gendreau, Michel, "An Introduction to TABU Search," Universite de Montreal, Montreal, Canada, Technical Report 2002.
- [40] IBM. (2011) IBM ILOG CPLEX Optimizer performance benchmarks. [Online]. <http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/cplex-optimizer/cplex-performance/>
- [41] Tobias Achterberg, "SCIP: solving constraint integer programs," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 1-41, 2009.
- [42] (2011, November) SCIP: An Open Source MIP Solver and constraint integer programming framework. [Online]. <http://scip.zib.de/>

- [43] H. Mittelman. (2011, November) Mixed Integer Linear Programming Benchmark (MILPB2010). [Online]. <http://plato.asu.edu/ftp/milpc.html>
- [44] V. M. Manquinho and J. Marques-Silva, "On using cutting planes in pseudoboolean optimization," *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, pp. 209–219, 2006. [Online]. <http://sat.inesc.pt/~vmm/research/index.html>
- [45] Niklas Sörensson and Niklas Eén, "An extensible SAT-solver," in *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pp. 502–508, 2003. [Online]. <http://minisat.se/MiniSat+.html>
- [46] Hossein M. Sheini and Karem A. Sakallah, "Pueblo: A Hybrid Pseudo-Boolean SAT Solver," *Journal on Satisfiability, Boolean Modelling and Computation*, vol. 2, no. 165-189, 2006.
- [47] (2007) PB'07 evaluation: time to solve an instance. [Online]. <http://www.cril.univ-artois.fr/PB07/results/timegraphs.php?idev=9>
- [48] Tamir Gal. (2008) SharpSSH - A Secure Shell (SSH) library for.NET. [Online]. <http://www.tamirgal.com/blog/page/SharpSSH.aspx>

Appendix A

The m^* Variable

Using the example in the example walkthrough earlier, the use of the variable m^* will be illustrated. The Intra-Cluster communication (IC) case is shown here. m^* is used in a similar way in the Multihop case.

$$\sum_{i=1}^N \sum_{j=1}^N (v_{i,j} - \sum_{\substack{k=1 \\ i \neq j \\ j \neq k \\ i \neq k}}^N f_{i,j,m^*}) \leq 0 \quad (3.35)$$

$$\sum_{i=1}^N \sum_{j=1}^N (Nv_{i,j} - \sum_{\substack{k=1 \\ i \neq j \\ j \neq k \\ i \neq k}}^N f_{i,j,m^*}) \geq 0 \quad (3.36)$$

From the example walkthrough for constraint 3.35 the following equations were obtained:

As

Constraint 3.35

$$v_{1,2} - f_{1,2,0} - f_{1,2,1} - f_{1,2,2} - f_{1,2,3} - f_{1,2,4} \leq 0$$

$$v_{1,3} - f_{1,3,0} - f_{1,3,1} - f_{1,3,2} - f_{1,3,3} - f_{1,3,4} \leq 0$$

$$v_{1,4} - f_{1,4,0} - f_{1,4,1} - f_{1,4,2} - f_{1,4,3} - f_{1,4,4} \leq 0$$

$$v_{1,5} - f_{1,5,0} - f_{1,5,1} - f_{1,5,2} - f_{1,5,3} - f_{1,5,4} \leq 0$$

$$v_{1,6} - f_{1,6,0} - f_{1,6,1} - f_{1,6,2} - f_{1,6,3} - f_{1,6,4} \leq 0$$

$$v_{1,7} - f_{1,7,0} - f_{1,7,1} - f_{1,7,2} - f_{1,7,3} - f_{1,7,4} \leq 0$$

$$v_{2,1} - f_{2,1,0} - f_{2,1,1} - f_{2,1,2} - f_{2,1,3} - f_{2,1,4} \leq 0$$

.....

As can be seen, the use of m^* is just for simplicity in coding as it is always from 0-4, keeping track of the option number and not the chosen option. If option 0 was chosen, option 0 for interconnecting node 1 and 2 is different from option 0 for interconnection node 1 and 3 and so on. m^* acts as an index of possible options.

Appendix B

Example Solved With Proposed Model and Enhancements

B.1 SOLVING THE EXAMPLE WITH THE SR MODEL

In this section, the example topology shown in Figure 20 will be solved using the Star-Ring model (base model – with no enhancements). Only a snapshot of each equation is shown as the complete set of equations are extensive and lengthy.

The following objective function is obtained when solving the example topology using the SR model:

$$\text{Minimize: } 0x_{1,1} + 13225x_{1,2} + \dots + 93025x_{7,6} + 0x_{7,7} - 1y_1 - \dots - 1y_7 - 1M_1 - \dots - 1M_7 + 0z_{1,1} + 1520875z_{1,2} + \dots + 28372625z_{7,6} + 0z_{7,7}$$

The following constraints are obtained when solving the example topology using the SR model:

Constraint 3.2: Number of Master ClusterHeads

$$M_1 + M_2 + M_3 + M_4 + M_5 + M_6 + M_7 = 1$$

Constraint 3.3: Number of Regular Clusterheads

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 = 2$$

Constraint 3.4: Upper limit of Connections (Maximum cluster size)

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} + x_{6,1} + x_{7,1} - 1y_1 \leq 1$$

...

$$x_{1,7} + x_{2,7} + x_{3,7} + x_{4,7} + x_{5,7} + x_{6,7} + x_{7,7} - 1y_7 \leq 1$$

Constraint 3.5: Lower Limit of Connections (ensuring no node is unconnected)

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} + x_{6,1} + x_{7,1} + 1M_1 \geq 1$$

...

$$x_{1,7} + x_{2,7} + x_{3,7} + x_{4,7} + x_{5,7} + x_{6,7} + x_{7,7} + 1M_7 \geq 1$$

Constraint 3.6: Upper limit of backbone connections

$$z_{1,1} + z_{2,1} + z_{3,1} + z_{4,1} + z_{5,1} + z_{6,1} + z_{7,1} - 3y_1 - 2M_1 \leq 0$$

...

$$z_{1,7} + z_{2,7} + z_{3,7} + z_{4,7} + z_{5,7} + z_{6,7} + z_{7,7} - 3y_7 - 2M_7 \leq 0$$

Constraint 3.7: Lower Limit of Backbone Connections

$$z_{1,1} + z_{2,1} + z_{3,1} + z_{4,1} + z_{5,1} + z_{6,1} + z_{7,1} - 2y_1 - 2M_1 \geq 0$$

...

$$z_{1,7} + z_{2,7} + z_{3,7} + z_{4,7} + z_{5,7} + z_{6,7} + z_{7,7} - 2y_7 - 2M_7 \geq 0$$

Constraint 3.8: Ensuring that backbone connections are only CH-CH or CH-MCH

$$2z_{1,2} - y_1 - M_1 - y_2 - M_2 \leq 0$$

$$2z_{1,3} - y_1 - M_1 - y_3 - M_3 \leq 0$$

...

$$2z_{5,7} - y_5 - M_5 - y_7 - M_7 \leq 0$$

$$2z_{6,7} - y_6 - M_6 - y_7 - M_7 \leq 0$$

Constraint 3.9: A node can be only CH or MCH

$$y_1 + M_1 \leq 1$$

...

$$y_7 + M_7 \leq 1$$

Constraint 3.10: Node is not connected to itself

$$x_{1,1} + x_{2,2} + x_{3,3} + x_{4,4} + x_{5,5} + x_{6,6} + x_{7,7} = 0$$

Constraint 3.11: the connection matrix is diagonal

$$x_{1,1} - x_{1,1} = 0$$

$$x_{1,2} - x_{2,1} = 0$$

...

$$x_{7,6} - x_{6,7} = 0$$

$$x_{7,7} - x_{7,7} = 0$$

Constraint 3.12: The backbone connection matrix is diagonal

$$z_{1,1} - z_{1,1} = 0$$

$$z_{1,2} - z_{2,1} = 0$$

...

$$z_{7,6} - z_{6,7} = 0$$

$$z_{7,7} - z_{7,7} = 0$$

Constraint 3.13: Restricting the total number of connections (Node-CH) in the topology

$$x_{1,2} + x_{1,3} \dots + x_{5,7} + x_{6,7} = 4$$

Constraint 3.14: Restricting the Backbone connection count (CH-CH, CH-MCH)

$$z_{1,2} + z_{1,3} + \dots + z_{5,7} + z_{6,7} = 3$$

Constraint 3.15: Backbone Node is not connected to itself

$$z_{1,1} + z_{2,2} + z_{3,3} + z_{4,4} + z_{5,5} + z_{6,6} + z_{7,7} = 0$$

Constraint 3.16: Regular nodes cannot connect to each other (One of the nodes has to be a clusterhead)

$$2x_{1,2} - y_1 - y_2 \leq 1$$

$$2x_{1,3} - y_1 - y_3 \leq 1$$

...

$$2x_{5,7} - y_5 - y_7 \leq 1$$

$$2x_{6,7} - y_6 - y_7 \leq 1$$

Figure 42 is a screenshot of the solution generated when solving the example topology using the SR model.

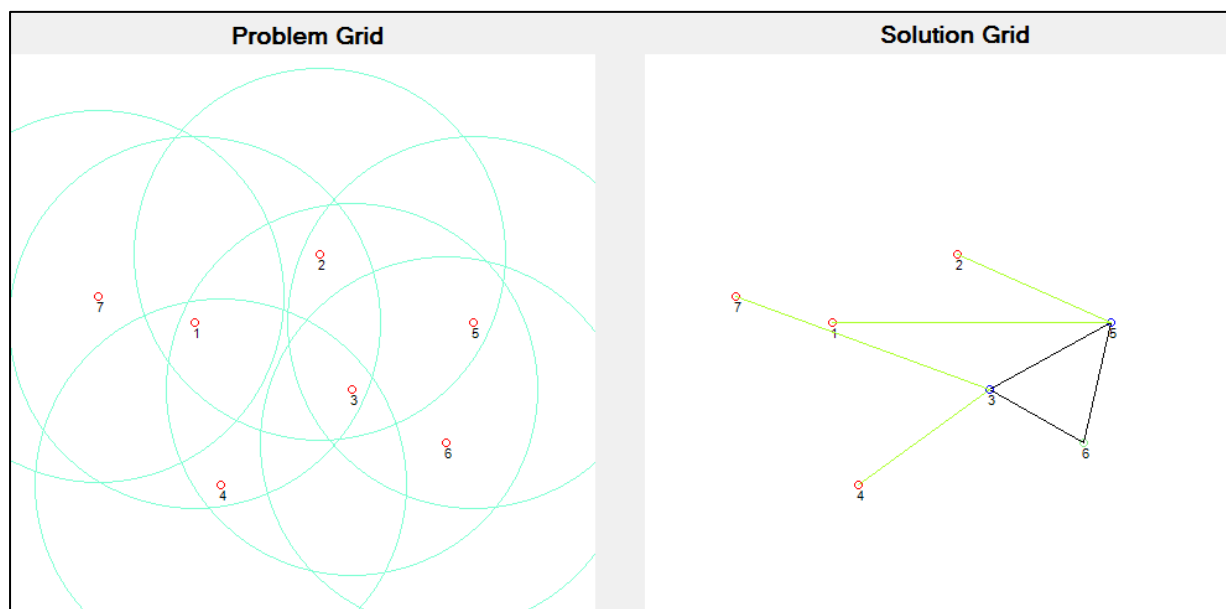


Figure 42: Example Solution Using SR Model.

B.2 SOLVING THE EXAMPLE WITH THE SR+CV MODEL

In this section, coverage constraints will be enforced when solving the example topology. This will restrict nodes to connect to other nodes which are within their coverage radius. There is no modification of the objective function or the other constraints. Only constraints 3.48 and 3.49 are added. These constraints automatically allow for the possibility or completely eliminate the possibility of certain connections depends on whether or not the corresponding nodes are in each other's coverage radius.

The following are the additional constraints when using the SR+CV model to solve the example topology:

Coverage Constraints (3.49 and 3.50)

$$x_{1,1} \leq 0$$

$$z_{1,1} \leq 0$$

$$x_{1,2} \leq 1$$

$$z_{1,2} \leq 1$$

$$x_{1,3} \leq 1$$

$$z_{1,3} \leq 1$$

...

$$x_{7,6} \leq 0$$

$$z_{7,6} \leq 0$$

$$x_{7,7} \leq 0$$

$$z_{7,7} \leq 0$$

Based on the restrictions imposed on certain connections that were allowed before, the solution will change. It should be kept in mind, that in certain cases where coverage is enforced and the nodes are far apart, there will be no possible solutions.

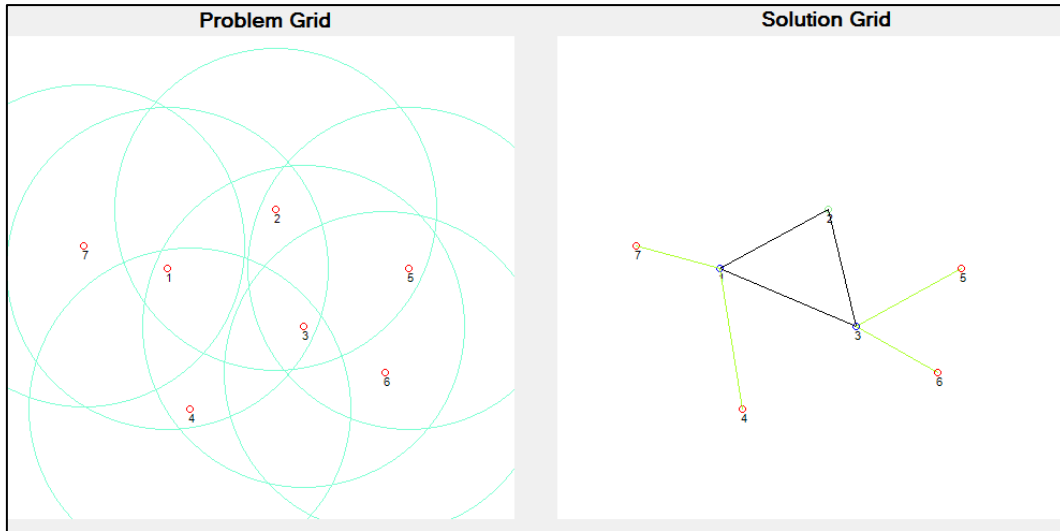


Figure 43: Example Solution with SR Backbone and Coverage Constraints.

As can be seen from the Figure 43, the solution is clearly different from the one in Figure 42 where coverage constraints were not present. While the solution obtained without coverage constraints is the lowest cost solution, it had connections which were not physically possible since the nodes were outside each other's coverage radius. This includes the connection between node 3 and node 7 which were previously connected and also the connection between node 1 and node 5.

B.3 SOLVING THE EXAMPLE WITH THE SR+IC MODEL

In this section, the example topology will be solved using the Star Ring Base model with the Intra Cluster communication enhancement (SR+IC). In this case, coverage will not be taken into account.

The following objective function is obtained when trying to solve the example topology with the SR+IC model:

$$\begin{aligned} \text{Minimize: } & 0x_{1,1} + 13225x_{1,2} + 93025x_{7,6} + 0x_{7,7} + 0v_{1,1} + 39675v_{1,2} + \dots + 279075v_{7,6} \\ & + 0v_{7,7} - 1y_1 - 1y_2 - \dots - 1y_6 - 1y_7 - 1M_1 - 1M_2 - \dots - 1M_6 - 1M_7 + 0z_{1,1} + 1520875z_{1,2} \\ & + \dots + 28372625z_{7,6} + 0z_{7,7} \end{aligned}$$

The following constraints are obtained when trying to solve the example topology with the SR+IC model:

Constraint 3.44: Identifying all possible intra cluster combinations
(2 nodes being in the same cluster, connected to the same clusterhead)

$$2f_{1,2,0} - x_{1,3} - x_{2,3} \leq 0$$

$$2f_{1,2,1} - x_{1,4} - x_{2,4} \leq 0$$

...

$$2f_{7,6,3} - x_{7,4} - x_{6,4} \leq 0$$

$$2f_{7,6,4} - x_{7,5} - x_{6,5} \leq 0$$

Constraint 3.45: Works together with the previous constraint to create the ‘And’ clause, ensuring that there is an Intra-Cluster connectin only if the two nodes are connected to the same clusterhead.

$$x_{1,3} + x_{2,3} - f_{1,2,0} \leq 1$$

$$x_{1,4} + x_{2,4} - f_{1,2,1} \leq 1$$

$$x_{1,5} + x_{2,5} - f_{1,2,2} \leq 1$$

...

$$x_{7,3} + x_{6,3} - f_{7,6,2} \leq 1$$

$$x_{7,4} + x_{6,4} - f_{7,6,3} \leq 1$$

$$x_{7,5} + x_{6,5} - f_{7,6,4} \leq 1$$

Constraint 3.40: Works with next constraint to create the ‘OR’ clause. The variable v is the ‘OR-ing’ of all the possibilities that could allow for the interconnection of selected 2 nodes. If any one of those possibilities is true (Example: Node 1 and Node 3 are connected to clusterhead 2, then v_{1,3} will be equal to 1)

$$v_{1,2} - f_{1,2,0} - f_{1,2,1} - f_{1,2,2} - f_{1,2,3} - f_{1,2,4} \leq 0$$

$$v_{1,3} - f_{1,3,0} - f_{1,3,1} - f_{1,3,2} - f_{1,3,3} - f_{1,3,4} \leq 0$$

...

$$v_{7,5} - f_{7,5,0} - f_{7,5,1} - f_{7,5,2} - f_{7,5,3} - f_{7,5,4} \leq 0$$

$$v_{7,6} - f_{7,6,0} - f_{7,6,1} - f_{7,6,2} - f_{7,6,3} - f_{7,6,4} \leq 0$$

Constraint 3.41: See explanation for 3.40 (previous constraint)

$$7v_{1,2} - f_{1,2,0} - f_{1,2,1} - f_{1,2,2} - f_{1,2,3} - f_{1,2,4} \geq 0$$

$$7v_{1,3} - f_{1,3,0} - f_{1,3,1} - f_{1,3,2} - f_{1,3,3} - f_{1,3,4} \geq 0$$

...

$$7v_{7,5} - f_{7,5,0} - f_{7,5,1} - f_{7,5,2} - f_{7,5,3} - f_{7,5,4} \geq 0$$

$$7v_{7,6} - f_{7,6,0} - f_{7,6,1} - f_{7,6,2} - f_{7,6,3} - f_{7,6,4} \geq 0$$

Constraint 3.42: Ensuring that 2 nodes cannot be linked to themselves

$$v_{1,1} + v_{2,2} + v_{3,3} + v_{4,4} + v_{5,5} + v_{6,6} + v_{7,7} = 0$$

Constraint 3.43: If node 1 is connected to node 2 in the same cluster that implies that node 2 is connected to node 1 (diagonal matrix).

$$v_{1,1} - v_{1,1} = 0$$

$$v_{1,2} - v_{2,1} = 0$$

$$v_{1,3} - v_{3,1} = 0$$

...

$$v_{7,5} - v_{5,7} = 0$$

$$v_{7,6} - v_{6,7} = 0$$

$$v_{7,7} - v_{7,7} = 0$$

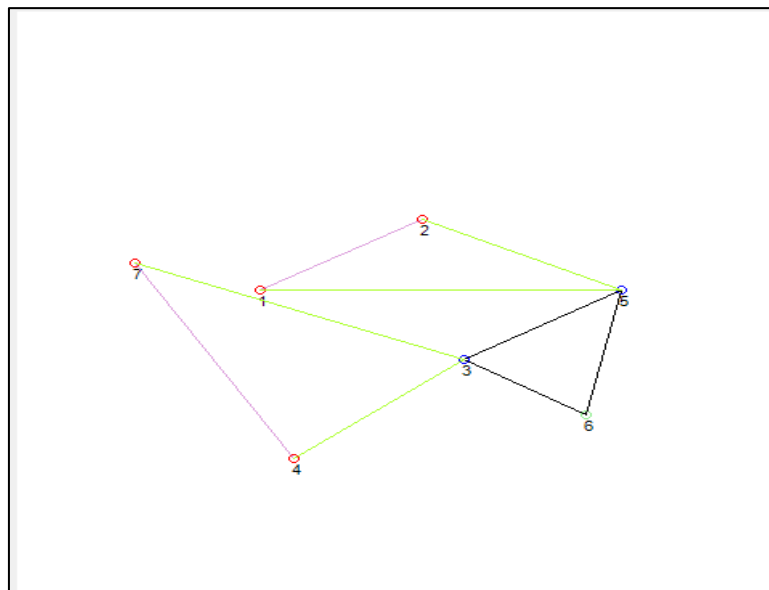


Figure 44: Example Solution with SR Backbone and Intra-Cluster Communication.

As can be seen from Figure 44, the solution is similar to the standard S-R solution shown in Figure 42, with the only difference being the new links created between the nodes in the same cluster. They are now interconnected with the purple links in Figure 23 (node 2 is connected to node 1, node 4 is connected to node 7).

B.4 SOLVING AN EXAMPLE TOPOLOGY WITH THE SR+MH MODEL

In this section, the example topology will be solved with the Star Ring Base Model with the Multihop connection enhancement enabled.

The following objective function is obtained when solving the example topology with the SR+MH model:

$$\begin{aligned} \text{Minimize: } & 0x_{1,1} + 13225x_{1,2} + 19044x_{1,3} + \dots + 93025x_{7,6} + 0x_{7,7} + 0q_{1,1} + 39675q_{1,2} + \\ & \dots + 279075q_{7,6} + 0q_{7,7} + 0b_{1,1,1} + 0b_{1,1,2} + \dots + 0b_{7,7,6} + 0b_{7,7,7} - 1y_1 - 1y_2 - \dots - 1y_6 \\ & - 1y_7 - 1M_1 - 1M_2 - \dots - 1M_6 - 1M_7 + 0z_{1,1} + 1520875z_{1,2} + \dots + 28372625z_{7,6} + 0z_{7,7} \end{aligned}$$

The following constraints are obtained when solving the example topology with the SR+MH model. Both, the constraints which are changed from the SR model and the additional constraints for the Multihop connection enhancement are shown:

Constraint 3.23: (enforcing the Upper limit for the maximum cluster size)

$$\begin{aligned} & b_{1,1,1} + b_{1,1,2} + b_{1,1,3} + b_{1,1,4} + b_{1,1,5} + b_{1,1,6} + b_{1,1,7} + x_{1,1} + b_{1,2,1} + b_{1,2,2} + b_{1,2,3} \\ & + b_{1,2,4} + b_{1,2,5} + b_{1,2,6} + b_{1,2,7} + x_{1,2} + b_{1,3,1} + b_{1,3,2} + b_{1,3,3} + b_{1,3,4} + b_{1,3,5} + \\ & b_{1,3,6} + b_{1,3,7} + x_{1,3} + b_{1,4,1} + b_{1,4,2} + b_{1,4,3} + b_{1,4,4} + b_{1,4,5} + b_{1,4,6} + b_{1,4,7} + x_{1,4} + \\ & b_{1,5,1} + b_{1,5,2} + b_{1,5,3} + b_{1,5,4} + b_{1,5,5} + b_{1,5,6} + b_{1,5,7} + x_{1,5} + b_{1,6,1} + b_{1,6,2} + b_{1,6,3} \\ & + b_{1,6,4} + b_{1,6,5} + b_{1,6,6} + b_{1,6,7} + x_{1,6} + b_{1,7,1} + b_{1,7,2} + b_{1,7,3} + b_{1,7,4} + b_{1,7,5} + \\ & b_{1,7,6} + b_{1,7,7} + x_{1,7} \leq 2 \end{aligned}$$

...

$$\begin{aligned} & b_{7,1,1} + b_{7,1,2} + b_{7,1,3} + b_{7,1,4} + b_{7,1,5} + b_{7,1,6} + b_{7,1,7} + x_{7,1} + b_{7,2,1} + b_{7,2,2} + b_{7,2,3} \\ & + b_{7,2,4} + b_{7,2,5} + b_{7,2,6} + b_{7,2,7} + x_{7,2} + b_{7,3,1} + b_{7,3,2} + b_{7,3,3} + b_{7,3,4} + b_{7,3,5} + \\ & b_{7,3,6} + b_{7,3,7} + x_{7,3} + b_{7,4,1} + b_{7,4,2} + b_{7,4,3} + b_{7,4,4} + b_{7,4,5} + b_{7,4,6} + b_{7,4,7} + x_{7,4} + \\ & b_{7,5,1} + b_{7,5,2} + b_{7,5,3} + b_{7,5,4} + b_{7,5,5} + b_{7,5,6} + b_{7,5,7} + x_{7,5} + b_{7,6,1} + b_{7,6,2} + b_{7,6,3} \end{aligned}$$

$$+ b_{7,6,4} + b_{7,6,5} + b_{7,6,6} + b_{7,6,7} + x_{7,6} + b_{7,7,1} + b_{7,7,2} + b_{7,7,3} + b_{7,7,4} + b_{7,7,5} + b_{7,7,6} + b_{7,7,7} + x_{7,7} \leq 2$$

Constraint 3.24: The new constraint for the Lower Limit (ensuring each node is connected to at least one node)

$$x_{1,1} + q_{1,1} + x_{2,1} + q_{2,1} + x_{3,1} + q_{3,1} + x_{4,1} + q_{4,1} + x_{5,1} + q_{5,1} + x_{6,1} + q_{6,1} + x_{7,1} + q_{7,1} + 1M_1 \geq 1$$

...

$$x_{1,7} + q_{1,7} + x_{2,7} + q_{2,7} + x_{3,7} + q_{3,7} + x_{4,7} + q_{4,7} + x_{5,7} + q_{5,7} + x_{6,7} + q_{6,7} + x_{7,7} + q_{7,7} + 1M_7 \geq 1$$

Constraint 3.25: The new constraint for restricting the total number of regular connections (which now includes hop connections as well).

$$x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6} + x_{1,7} + x_{2,3} + x_{2,4} + x_{2,5} + x_{2,6} + x_{2,7} + x_{3,4} + x_{3,5} + x_{3,6} + x_{3,7} + x_{4,5} + x_{4,6} + x_{4,7} + x_{5,6} + x_{5,7} + x_{6,7} + q_{1,1} + q_{1,2} + \dots + q_{7,7} = 4$$

Constraint 3.26: Taking into account all hopping possibilities

$$t_{1,2,0} - x_{1,3} \leq 0$$

$$t_{1,2,1} - x_{1,4} \leq 0$$

$$t_{1,2,2} - x_{1,5} \leq 0$$

$$t_{1,2,3} - x_{1,6} \leq 0$$

$$t_{1,2,4} - x_{1,7} \leq 0$$

...

$$t_{7,6,0} - x_{7,1} \leq 0$$

$$t_{7,6,1} - x_{7,2} \leq 0$$

$$t_{7,6,2} - x_{7,3} \leq 0$$

$$t_{7,6,3} - x_{7,4} \leq 0$$

$$t_{7,6,4} - x_{7,5} \leq 0$$

Constraint 3.27 and 3.28: Ensuring as part of the possibility the node is hopping to a clusterhead and not to a regular node.

$$t_{1,2,0} + y_2 \leq 1$$

$$t_{1,2,1} + y_2 \leq 1$$

$$t_{1,2,2} + y_2 \leq 1$$

$$t_{1,2,3} + y_2 \leq 1$$

$$t_{1,2,4} + y_2 \leq 1$$

...

$$t_{7,6,0} + y_7 \leq 1$$

$$\begin{aligned}
t_{7,6,1} + y_7 &\leq 1 \\
t_{7,6,2} + y_7 &\leq 1 \\
t_{7,6,3} + y_7 &\leq 1 \\
t_{7,6,4} + y_7 &\leq 1
\end{aligned}$$

Constraint 3.29: Ensuring that if a node is hopping to a clusterhead, it isn't also directly connecting to the clusterhead.

$$\begin{aligned}
q_{1,2} + x_{1,2} &\leq 1 \\
q_{1,3} + x_{1,3} &\leq 1 \\
&\dots \\
q_{7,5} + x_{7,5} &\leq 1 \\
q_{7,6} + x_{7,6} &\leq 1
\end{aligned}$$

Constraint 3.30: Implementing the OR constraint (together with the next constraint). If anyone of the possible hops is taken, the hop connection is registered for that node.

$$\begin{aligned}
q_{1,2} - t_{1,2,0} - t_{1,2,1} - t_{1,2,2} - t_{1,2,3} - t_{1,2,4} &\leq 0 \\
q_{1,3} - t_{1,3,0} - t_{1,3,1} - t_{1,3,2} - t_{1,3,3} - t_{1,3,4} &\leq 0 \\
&\dots \\
q_{7,5} - t_{7,5,0} - t_{7,5,1} - t_{7,5,2} - t_{7,5,3} - t_{7,5,4} &\leq 0 \\
q_{7,6} - t_{7,6,0} - t_{7,6,1} - t_{7,6,2} - t_{7,6,3} - t_{7,6,4} &\leq 0
\end{aligned}$$

Constraint 3.31: Completing the 'OR'

$$\begin{aligned}
7q_{1,2} - t_{1,2,0} - t_{1,2,1} - t_{1,2,2} - t_{1,2,3} - t_{1,2,4} &\geq 0 \\
7q_{1,3} - t_{1,3,0} - t_{1,3,1} - t_{1,3,2} - t_{1,3,3} - t_{1,3,4} &\geq 0 \\
&\dots \\
7q_{7,5} - t_{7,5,0} - t_{7,5,1} - t_{7,5,2} - t_{7,5,3} - t_{7,5,4} &\geq 0 \\
7q_{7,6} - t_{7,6,0} - t_{7,6,1} - t_{7,6,2} - t_{7,6,3} - t_{7,6,4} &\geq 0
\end{aligned}$$

Constraint 3.32: Preventing Nodes from hopping to themselves.

$$q_{1,1} + q_{2,2} + q_{3,3} + q_{4,4} + q_{5,5} + q_{6,6} + q_{7,7} = 0$$

Constraint 3.33: Hop is one way only, NOT diagonal.

$$\begin{aligned}
q_{1,2} + q_{2,1} &\leq 1 \\
q_{1,3} + q_{3,1} &\leq 1 \\
&\dots \\
q_{7,5} + q_{5,7} &\leq 1
\end{aligned}$$

$$q_{7,6} + q_{6,7} \leq 1$$

Constraint 3.34: Together with Equation 18b: Implementing the ‘AND’ clause. Ensuring that a hop is possible only if a regular node hops through a node connected to a clusterhead.

$$2 b_{1,2,3} - x_{1,2} - q_{2,3} \leq 0$$

$$2 b_{1,2,4} - x_{1,2} - q_{2,4} \leq 0$$

$$2 b_{1,2,5} - x_{1,2} - q_{2,5} \leq 0$$

...

$$2 b_{7,6,3} - x_{7,6} - q_{6,3} \leq 0$$

$$2 b_{7,6,4} - x_{7,6} - q_{6,4} \leq 0$$

$$2 b_{7,6,5} - x_{7,6} - q_{6,5} \leq 0$$

Constraint 3.35: Together with the previous constraint ensuring hop integrity.

$$x_{1,2} + q_{2,3} - b_{1,2,3} \leq 1$$

$$x_{1,2} + q_{2,4} - b_{1,2,4} \leq 1$$

$$x_{1,2} + q_{2,5} - b_{1,2,5} \leq 1$$

...

$$x_{7,6} + q_{6,3} - b_{7,6,3} \leq 1$$

$$x_{7,6} + q_{6,4} - b_{7,6,4} \leq 1$$

$$x_{7,6} + q_{6,5} - b_{7,6,5} \leq 1$$

The solution obtained is shown in Figure 45. We can see that unlike the standard SR solution in Figure 42, node 7, which is furthest from the clusterheads, is now able to hop through a closer node, rather than make a more expensive direct connection.

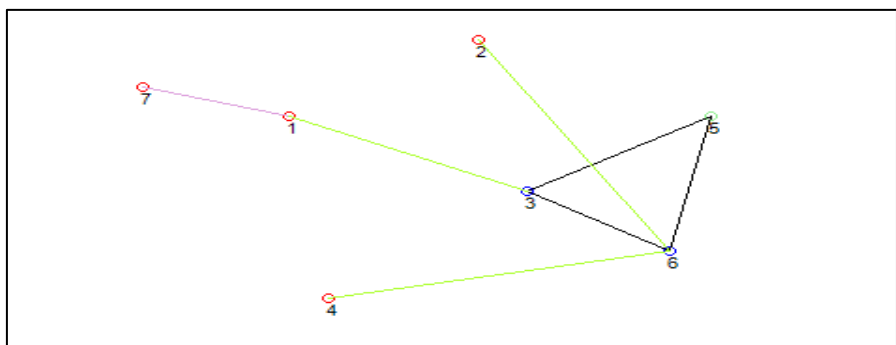


Figure 45: Example Solution with SR Backbone and Multihop Connections.

This section illustrated how a single topology can be solved using the different models resulting in different solutions depending on the requirements. It can be seen that as more and more ‘enhancements’ are made, the models become more complex.

VITA

Syed Zohaib Hussain Zahidi was born on October 18, 1986, in Kuwait, and moved to the United Arab Emirates (UAE) in 1992. He became a Microsoft Certified Professional in 2001 and went on to become a Microsoft Certified Systems Administrator and Microsoft Certified Systems Engineer. He graduated from high school in 2004 and subsequently joined the Computer Engineering program at the American University of Sharjah. Mr. Zahidi graduated magna cum laude from the American University of Sharjah in 2008 with a Bachelor of Science in Computer Engineering.

In 2008, Mr. Zahidi began the master's program in Computer Engineering at the American University of Sharjah. In November 2010, Mr. Zahidi joined IBM in Dubai (UAE) as part of the IBM Graduate Program. He was awarded the Master of Science degree in Computer Engineering from the American University of Sharjah in the Fall of 2011.

Mr. Zahidi's interests lie in Mobile Application Development and Mobile Enterprise offerings and he is a 'Social Business' enthusiast. Mr. Zahidi is an avid reader, an aspiring writer and enjoys playing squash in his spare time.