DEVELOPMENT OF EFFICIENT ALGORITHMS SUITABLE FOR VLSI CIRCUITS AND WIRELESS SENSOR NETWORKS

Thesis Submitted by

Subhrapratim Nath

DOCTOR OF PHILOSOPHY (ENGINEERING)

Department of Computer Science and Engineering
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata, India

2023

JADAVPUR UNIVERSITY
KOLKATA- 700032

INDEX NO. 95/17/E

1. Title of the thesis:

DEVELOPMENT OF EFFICIENT ALGORITHMS SUITABLE FOR VLSI CIRCUITS AND WIRELESS SENSOR NETWORKS

2. Name, Designation & Institution of the Supervisors:

Prof. Jamuna Kanta Sing
Professor
Department of Computer Science and Engineering
Jadavpur University
Kolkata-700032, India

Prof. Subir Kumar Sarkar
Professor
Department of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata-700032, India

3. List of Publications:

Journal Publications (6)

1. **S. Nath**, J.K. Sing, and S.K. Sarkar, "Wire length optimization of VLSI circuits using IWO algorithm and its hybrid," *Circuit World*, Emerald Publishing, vol. Aop, no. Aop, July 2021, doi: 10.1108/CW-08-2020-0215. [0.875, SCI Indexed]
2. **S. Nath**, A. Dey, P. Pachal, J.K. Sing, and S.K. Sarkar, "Performance analysis of gas sensing device and corresponding IoT framework in mines," *Microsystem Technologies*, Springer, vol. 27, pp. 3977–3985, September 2019, doi:10.1007/s00542-019-04621-x. [2.01, SCI Indexed]
3. **S. Nath**, and S.K. Sarkar, "Metaheuristics-based routing optimization, balanced workload distribution and security strategy in IoT environment," *International Journal of Advanced Intelligence Paradigms*, vol. 19, no. 1, pp. 101-117, April 2021, doi:10.1504/IJAIP.2021.114586 [1.1, Scopus Indexed]
4. **S. Nath**, A. Seal, A. Bhattacharya, and S.K. Sarkar, "A New Approach to Routing Optimization for Cluster-Based Wireless Networks using Swarm Intelligence," *Advances in Industrial Engineering and Management*, American Scientific Publisher, vol. 6, no. 2, pp. 83-89, 2017, ISSN: 2222-7059 (Print); eISSN: 2222-7067 (Online). [1.60, DOAJ Indexed]
5. **S. Nath**, A.K. Chakravarty, S. Ghosh, and S.K. Sarkar, "FPGA Placement Optimization using Firefly Algorithm," *Advances in Industrial Engineering and Management*, American Scientific Publisher, vol. 6, no. 2, pp. 83-89, 2017. ISSN: 2222-7059 (Print); eISSN: 2222-7067 (Online). [1.60, DOAJ Indexed]
6. S. Ghosh, **S. Nath**, and S.K. Sarkar, "PSO Algorithm with self-tuned parameter for efficient routing in VLSI design," *SSRG International Journal of Electronics and Communication Engineering*, pp. 60-63, 2015, ISSN: 2348 – 8549. [0.72, Scopus Indexed]

Conference Publications (15)

1. **S. Nath,** S. Ghosh, and S.K. Sarkar, "A novel approach to discrete Particle Swarm Optimization for efficient routing in VLSI design," *in Proceedings of 4th International conference on Reliability, Infocom Technologies and Optimization*, Noida, India, 2015, pp. 1-4, doi: 10.1109/ICRITO.2015.7359375**.**
2. **S. Nath,** S. Sarkar, R. Bandyopadhyay, and S.K. Sarkar, "A New Optimization Algorithm Based on Physarum BioNetwork for VLSI Global Routing," *in Proceedings of IEEE International conference on Computing, Communication and Sensor Network*, Kolkata, India, 2016, vol 1, pp 119-123, ISBN : 81-85824-46-0.

3. **S. Nath**, A. Seal, T. Banerjee, and S.K. Sarkar, "Optimization Using Swarm Intelligence and Dynamic Graph Partitioning in IoE Infrastructure: Fog Computing and Cloud Computing", *in Proceedings of International conference on Computational Intelligence, Communications, and Business Analytics*, Kolkata, India, 2017 Communications in Computer and Information Science, Springer, vol 775, pp. 440-452, doi: 10.1007/978-981-10-6427-2_36

4. **S. Nath**, A.K. Chakravarty, S. Ghosh, and S.K. Sarkar, "Invasive weed optimization approach to VLSI routing," *in Proceedings of IEEE International conference on Devices for Integrated Circuit*, Kalyani, India, 2017, pp. 615-619, doi: 10.1109/DEVIC.2017.8074024.

5. S. Ghosh, **S. Nath,** R. Biswas, P. Venkateswaran, J.K. Sing, and S.K. Sarkar, "PSO Variants and its Comparison with Firefly Algorithm in Solving VLSI Global Routing Problem," *in Proceedings of IEEE Electron Devices Kolkata Conference,* Kolkata, India, 2018, pp. 513-518, doi: 10.1109/EDKCON.2018.8770397.

6. **S. Nath**, P.P.G. Neogi, J.K. Sing, and S.K. Sarkar, "VLSI Routing Optimization based on modified Constricted PSO with Iterative RLC Delay model," *in Proceedings of IEEE International Conference on Current Trends towards Converging Technologies*, Coimbatore, India, 2018, pp. 1-7, doi: 10.1109/ICCTCT.2018.8550839

7. **S. Nath**, A. Dey, P. Pachal, S.R. Chowdhury, J.K. Sing, and S.K. Sarkar, "Nano Structured Gas Sensing Device and Its Application in Underground Mines," *in Proceedings of IEEE Electron Devices Kolkata Conference,* Kolkata, India, 2018, pp. 445-449, doi: 10.1109/EDKCON.2018.8770439.

8. **S. Nath**, S. Banik, A. Seal, and S.K. Sarkar, "Optimizing MANET routing in AODV: A hybridization approach of ACO and firefly algorithm," *in Proceedings of IEEE International Conference on Research in Computational Intelligence and Communication Network*, Kolkata, India, 2016, pp. 122-127, doi: 10.1109/ICRCICN.2016.7813643.

9. **S. Nath,** S. Gupta, S. Biswas, R. Banerjee, J.K. Sing, and S.K. Sarkar, "GPSO Hybrid Algorithm for Rectilinear Steiner Tree Optimization*," in Proceedings of IEEE VLSI Device Circuit and System,* Kolkata, India, 2020, pp. 365-369, doi: 10.1109/VLSIDCS47293.2020.9179861.

10. **S. Nath,** R. Bandyopadhyay, S. Biswas, J.K. Sing, and S.K. Sarkar, "A New Global Routing Optimization Algorithm based on Pigeon Inspired Optimization," *in Proceedings of IEEE Calcutta Conference (CALCON)*, Kolkata, India, 2020, pp. 184-188, doi: 10.1109/CALCON49167.2020.9106507.

11. **S. Nath**, A. Shankar, R. Sarkar, S. Banerjee, J.K. Sing, and S.K. Sarkar, "Minimizing Wirelength with Bend Reduction using Gradient Descent PSO

Hybrid in VLSI Global Routing," *in Proceedings of Devices for Integrated Circuit*, Kalyani, India, 2021, pp. 401-405, doi: 10.1109/DevIC50843.2021.9455851.

12. **S. Nath**, I. Das, S. Roy, A. Jana, J.K. Sing, and S.K. Sarkar, "Enhancing Packet Delivery Rate in Flying Ad-Hoc Network Routing," *in Proceedings of IEEE R10 Humanitarian Technology Conference (R10-HTC)*, Depok, West Java, Indonesia 2019, pp. 66-70, doi: 10.1109/R10-HTC47129.2019.9042457.

13. **S. Nath**, A. Paul, R. Banerjee, S. Bhaumik, J.K. Sing, and S.K. Sarkar, "Optimizing FANET Routing Using a Hybrid Approach of Firefly Algorithm and ACO-Lévy Flight," *in Proceedings of IEEE VLSI Device Circuit and System*, Kolkata, India, 2020, pp. 378-383, doi: 10.1109/VLSIDCS47293.2020.9179956.

14. P. Nath, S. Dey, **S. Nath,** A. Shankar, J.K. Sing, and S.K. Sarkar, "VLSI Routing Optimization Using Hybrid PSO Based on Reinforcement Learning," *in Proceedings of IEEE VLSI Device Circuit and System,* Kolkata, India, 2022, pp. 238-243, doi: 10.1109/VLSIDCS53788.2022.9811434.

15. **S. Nath,** A. Dey, P. Das, D. Mohapatra, J.K. Sing, and S.K. Sarkar, "Application of Soil Sensors for Maximizing Productivity Using IoT Framework," *in Proceedings of IEEE VLSI Device Circuit and System,* Kolkata, India, 2022, pp. 220-224, doi: 10.1109/VLSIDCS53788.2022.9811456.

4. List of Patents: NIL

5. List of Presentations in National/International Conferences:

1. **S. Nath,** S. Ghosh, and S.K. Sarkar, "A novel approach to discrete Particle Swarm Optimization for efficient routing in VLSI design," *in Proceedings of 4th International conference on Reliability, Infocom Technologies and Optimization*, Noida, India, 2015, pp. 1-4, doi: 10.1109/ICRITO.2015.7359375**.**
2. **S. Nath**, P.P.G. Neogi, J.K. Sing, and S.K. Sarkar, "VLSI Routing Optimization based on modified Constricted PSO with Iterative RLC Delay model," *in Proceedings of IEEE International Conference on Current Trends towards Converging Technologies*, Coimbatore, India, 2018, pp. 1-7, doi: 10.1109/ICCTCT.2018.8550839

3. **S. Nath,** S. Gupta, S. Biswas, R. Banerjee, J.K. Sing, and S.K. Sarkar, "GPSO Hybrid Algorithm for Rectilinear Steiner Tree Optimization," *in Proceedings of IEEE VLSI Device Circuit and System,* Kolkata, India, 2020, pp. 365-369, doi: 10.1109/VLSIDCS47293.2020.9179861.

"Statement of Originality"

I, Subhrapratim Nath, registered on 15th February, 2017, do hereby declared that the thesis entitled "***Development of Efficient Algorithms suitable for VLSI Circuits and Wireless Sensor Networks***" contains literature survey and original research work done by undersigned candidates as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declared that I have checked this thesis as per the "Policy on Anti Plagiarism, Jadavpur University, 2019", and the level of similarity as checked by iThenticate software is **4 %.**

Subhrapratim Nath
Signature of Candidate:

Date: 13.03.23

Certified by Supervisor (s):
(Signature with date, seal)

1. 13/03/2023

Jamuna Kanta Sing, Ph.D.
Professor
Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-700032

2. 13.03.2023

Dr. Subir Kumar Sarkar
Professor and Former Head, Dept. of Electronics
& Telecommunication Engg.
Jadavpur University, Kolkata-32

CERTIFICATE FROM THE SUPERVISORS

This is to certify that the thesis entitled "**Development of Efficient Algorithms suitable for VLSI Circuits and Wireless Sensor Networks"** submitted by Sri. Subhrapratim Nath, who got his name registered on **15.02.2017** [**D-7/E/124/17**] for the award of Ph.D. (Engg.) degree of Jadavpur University is absolutely based upon her own work under the supervision of **Prof. Jamuna Kanta Sing** and **Prof. Subir Kumar Sarkar** and that neither his thesis nor any part of the thesis has been submitted for any degree/ diploma or any other academic award anywhere before.

13/03/2023

[Prof. Jamuna Kanta Sing]
Signature of Supervisor
Date with Office seal

Jamuna Kanta Sing, Ph.D.
Professor
Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-700032

13.03.2023

[Prof. Subir Kumar Sarkar]
Signature of Supervisor
Date with Office seal

Dr. Subir Kumar Sarkar
Professor and Former Head, Dept. of Electronics
& Telecommunication Engg.
Jadavpur University, Kolkata-32

Dedicated To

My Grand Parents

For their heavenly blessings,

My Parents and My family

For their constant support,

cooperation and sacrifice.

ACKNOWLEDGEMENTS

This work would not have been done without the support and inspiration of my supervisors **Prof. Jamuna Kanta Sing**, Professor, Department of Computer Science and Engineering, Jadavpur University and **Prof. Subir Kumar Sarkar**, Professor, Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata-700032, whose expertise, guidance, constant encouragement, and support helped me finish my dissertation. I feel a deep sense of respect, gratitude, and debt to them for their helpful advice, constructive comments, clear instructions, inspiring conversations, and never-ending source of ideas during this work.

I would like to offer my most sincere gratitude to the Faculty Council of Engineering and Technology, the Head of the Department, and all of the faculty members of the Department of Computer Science and Engineering at Jadavpur University for their assistance and support throughout the course of my research work.

I would like to thank Dr. Manash Chanda from the bottom of my heart for his helpful advice, constructive suggestions, and technical support, all of which helped my research work. I owe him a lot for always being there for me and cheering me on.

I want to thank my fellow research scholars at Jadavpur University, especially Mr. Sudipta Ghosh, Dr. Anup Dey, Mr. Bijoy Goswami, and Mr. Abir Jana, as well as everyone else who has helped me directly or indirectly with my work. They have been a good way to make friends, get good advice, and work together.

My parents, my family, and my friends have always shown me unconditional love, support, and encouragement. Saying "thank you" is not enough to show how much I appreciate them.

Finally, I would like to thank the Almighty God for giving me that inner strength to complete this work.

Subhrapratim Nath

(Subhrapratim Nath)

March 2023
Jadavpur University
Kolakata -700032

ABSTRACT

In the past few decades, technology has evolved in leaps and bounds. From wired pcs and basic electronic circuits to wireless interconnectivity among devices over a local network, that can solve real time problems. The fields of mathematics, networking, circuit design and artificial intelligence are evolving the most.

Due to its numerous advantages, VLSI electronic circuit design is the most important and pursued technology for developing efficient and dependable digital systems. Modern VLSI designs necessitate a diverse set of capabilities, including simple hardware size, high reliability, high speed, and power efficiency. Rapid advances in nanometre IC fabrication technology have enabled the manufacture of hundreds of thousands of transistors on a single chip. This provides a significant physical design issue for VLSI, particularly in the realm of VLSI routing. It used to be enough to regulate gate latency to improve the performance of VLSI circuits. However interconnecting latency is now becoming more important among researchers working on the deep sub-micron technological node, in addition to the goal of wire-length minimization.

A grid graph can be used to depict the interconnects of a circuit, with the graph terminals denoting the positioning of the individual circuit blocks. The link length is optimised using a graph's minimal spanning tree (MST) by adding Steiner points to the MST and building a Rectilinear Minimum Steiner Tree allows for even more optimization (RMST). However, finding the number and placement of Steiner points is an NP-hard task, making constructing the RMST for real-world circuits unfeasible by brute force and improbable to be efficient when employing simple Steiner point allocation criteria. The solution space, also known as the search space, is made up of all conceivable combinations of valid Steiner point placements. The quantity of Steiner points to be inserted raises the computational intricacy rapidly, which is dependent in part on the circuit parameters. In a search space with such a large parameter, some logic or intelligence is required to guide the search in the right path. In conjunction

with swarm intelligence technology, population-based metaheuristics have showed great promise in this direction.

Heuristic methods are a greater optimization approach for finding best solution in a search space. Because of its non-deterministic iterations, it is able to escape local optima and remains relatively independent of the individual problem to optimise. Swarm intelligence is a type of group activity in which a number of decentralised agents share information with one another according to a set of simple rules, resulting in the establishment of a swarm knowledge or global knowledge that no single agent is aware of over time. Particle swarm optimization (PSO), firefly algorithm (FA), invasive weed optimization (IWO), and Physarum Bio Network are only a few of the swarm intelligence-based metaheuristic algorithms being investigated to overcome the RMST problem, hence reducing wirelength and delay in VLSI routing. There are several modifications of the algorithms that solve some of the flaws and lead to better outcomes than the initial formulation, rather than the usual variant, of the method. Placing control conditions has been proved to be efficiently yielding outcomes for some of the specific problems because the meta heuristic algorithm employs non deterministic identities. These trial-and-error algorithms leads to better efficient optimising algorithms

Given the inputs of various research communities, modern Wireless Sensor Networks (WSNs) are extremely robust, cheap, and easy to monitor, given the constant development of more efficient protocols, techniques for better resource usage, and improved quality over large variations in topologies, and so on. WSNs eventually laid the groundwork for the Internet of Things (IoT), a framework in which important and common objects are connected via a network system, making it easier for us to collect data, interact with the devices themselves, and simplify our duties. WSNs are essentially deployed in large numbers, and clusters are created in agreement with other nodes, with substantial information about topologies being communicated back, usually to an access point. It's difficult to build a robust network in WSN systems to simplify data transfers because of the various topologies. Various conventional protocols and Ad hoc communication protocols are employed to overcome design and operation challenges. AODV overcomes the majority of the problems, resulting in a stable protocol that may be used in the system. Although the conventional AODV protocol employs blind flooding of cluster members with RREQ requests in the route discovery phase, which allows for hopping to the best adjacent nodes, WSNs with limited resources may incur large loads for the nodes to work correctly, and can even cause system delays. When the full set of nodes shifts, the AODV protocol may reject better routes due to non-use of the routing table and a high path drop rate. Ant-Colony Optimization and Particle Swarm Optimizations are perfect meta-heuristics that have been published in the context of implementing Swarm Intelligence and denouncing the clustering phenomena and WSN routing optimization. While meta-heuristics such as the Directed Artificial Bat Algorithm (DABA), ACO can be used for routing in WSN, ACO and Constricted PSO, versions of PSO, can be used for clustering. With the use of meta-heuristics and correct swarm intelligence-based routing algorithms, these address some of the challenges that the IoT paradigm has been dealing with, such as real-time delay and network congestion. The bandwidth utilisation and workload distribution are controlled by Fog computing. This helps in maintaining an effective line of intercommunication between the IoE clusters. When different algorithms are tried upon WSNs to create a hybrid algorithm they are tested in the emulators.

TABLE OF CONTENTS

Contents **Page No.**

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

IC	Integrated Circuit
SSI	Small Scale Integration
MSI	Medium Scale Integration
ULSI	Ultra-Large-Scale Integration
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
TFET	Tunnel Field Effect Transistor
MST	Minimum Spanning Tree
RSMT	Rectilinear Steiner Minimum Tree
NP	Non-deterministic Polynomial-time
ACO	Ant Colony Optimization
PSO	Particle Swarm Optimization
FA	Firefly Algorithm
ABC	Artificial Bee Colony
WSN	Wireless Sensor Network
CH	Cluster Head
BS	Base Station
IIoT	Industrial Internet of Things
IoT	Internet of Things
IWO	Invasive Weed Optimization
DABA	Directed Artificial Bat Algorithm
RISC	Reduced Instruction Set Computer
CAD	Computer-Aided Design
ILP	Integer Linear Programming
BPSO	Binary PSO
MRST	Minimum Rectilinear Steiner Tree
LEACH	Low Energy Adaptive Clustering Hierarchy
SPIN	Sensor Protocol for Information via Negotiation
AODV	Ad-hoc On-demand Distance Vector
DSR	Dynamic Source Routing
RREP	Route Reply
OLSR	Optimal Link State Routing
QoS	Quality of Service
P	Polynomial Time
SI	Swarm Intelligence
TSP	Travelling Salesman Problem
RISC	Reduced Instruction Set Computer
PSO-W	Weighted PSO
PSO-C	Constricted PSO
DPSO	Discrete PSO
PSO-ST	Self-Tuned PSO
DIWO	Discrete Invasive Weed Optimization
MOPSO	Multi-objective PSO

CHAPTER 1

INTRODUCTION AND ORGANIZATION OF THE THESIS

1.1. Introduction and motivation

In the past decades technology has evolved along an exponential graph. It is a long way from simple chips to complicated smart systems communicating over a network making real time decisions to achieve a goal. Today everyone is connected globally and as its effect technological advances are taking place in leaps and bounds. The fields of circuit design, mathematical optimization, networking, and artificial intelligence have had a great positive impact due to this advancement.

This revolution is predicated on a convergence of computing and communication technologies, both of which are fuelled by advancements in Integrated Circuit (IC) technology. In computers, integrated circuits are used for processing units' memory components, and interface chips, among other things. Computer networking, network switching systems, communication systems, vehicles, aero planes, and even household appliances nowadays have integrated circuits embedded in them.

The existence of ICs is traced to early 1949. Werner Jacobi (de) (Siemens AG) [1.1] filed a patent for an IC-like design for a semiconductor amplifier [1.2]. Kilby recorded an initial idea regarding the IC in July 1958 [1.3]. Kilby described the new device as "a body of semiconductor material wherein all the components of the electronic circuit are completely integrated" in the patent application. The initial ICs had very basic functionalities with transistors numbered in a few tens and with one or two logic gates. These came to known as Small Scale Integration (SSI) chips [1.4]. The Plessey SL201 or the Philips TAA320 had only two transistors. The phrase Large Scale Integration was coined by IBM scientist Rolf Landauer who described the concepts for SSI, MSI, VLSI, and ULSI. The advent of VLSI began in 1980s and even today its various functionalities are implemented. The transformation of these chips began when the number of transistors increased from hundreds of thousands to billions over from 1980 to 2009. The International Technology Roadmap for Semiconductors [1.5] is discussed on the efficient path designs. To bring down power consumption to an agreeable limit the research has moved along from Negative Channel Metal-Oxide Semiconductor to Positive Channel Metal-Oxide Semiconductor and now to Complementary Metal Oxide Semiconductor. In 1986 the first on-megabit Random Access Memory chips were introduced which had more than a hundred thousand transistors embedded in them. Microprocessor chips moved beyond the million-transistor mark in 1989 and the billion-transistor mark in 2005 [1.6]. The trend is still going strong, with devices having trillions of memory transistors [1.7] being announced in 2007. Wafer-scale integration is a massive system of VLSI circuits combined onto a single 'super-chip' using reduced packaging. For some systems, like parallel supercomputers [1.8], it is an effective circuit. A system-on-a-chip is a type of integrated circuit that has all of the components required for a computer or other system on a single chip. It is quite popular [1.9] due to decreased manufacturing and assembly costs, as well as a significantly reduced power budget. Circuits with more than 1 million transistors on a chip are known as ULSI [1.10]. A three-dimensional integrated circuit (or $3D$-IC) is made up of two or more layers of activated electronic components that are stacked vertically and horizontally to form a single circuit. On-die signalling is used to connect the layers, resulting in lower power consumption than isolated circuits. Total wirelength can be lowered in such ICs by carefully employing tiny vertical wires, resulting in faster operation [1.11]. Several works in modified MOSFET [1.12] - [1.14] and TFT [1.15] - [1.17] in semiconductor technology have been reported as technology improves. Finally, these novel ideas will lead to further scaling in semiconductor technology, increasing the number of gates per chip and the density of cells. [1.18]

Due to its numerous advantages, VLSI electronic circuit design is the most important and requested technology for developing efficient and dependable digital systems. Modern VLSI designs necessitate a diverse set of capabilities, including small device size, high efficiency, high speed, and low power consumption. The production of billions of transistors on a single chip has been made possible by rapid advances in IC manufacturing technology in the nanometre range [1.19]. VLSI physical design is now faced with a significant difficulty. Before production, the stage of physical design [1.20] - [1.22] converts Register-Transistor Level code to Graphical Data System II. Partitioning, Floor layout, Placement, Clock Tree Synthesis, Routing, and other physical design elements are included in VLSI routing.

It is used to be enough to regulate gate delay to improve the performance of VLSI circuits; however, interconnect delay is now becoming more important among researchers working on the deep sub-micron technological node.

In the physical design flow, interconnection optimization is crucial. Topologies for all of the nets in cutting-edge VLSI design will be created and optimised with performance and power consumption in mind. Interconnection efficiency is becoming more difficult as technology advances for two reasons:

1. Since RC connectivity latency is growing while gate delay is decreasing with CMOS scaling, the percentage of overall delay attributable to RC interconnections is rapidly growing. [1.23]

2. Optimizing connectivity gets increasingly challenging as overall wirelength and cell density rise, raising concerns about congestion and routability.

After installation, the logical blocks and gates remain in place, but the wiring between them does not. It has already been explained that optimising interconnections entails arranging all cables in a way that maximises performance. The fundamental aspect of the interconnection planning process is routing, particularly global routing. Routing is divided into two components, global routing and detail routing. This division divides the problem into two sub-problems, each of which is reasonably simple. A coarse-grained grid depicting an approximation of each net's shape is used to do the global routing. It provides a smaller solution space with a coarse-grain grid, resulting in a shorter runtime [1.24].

The sequential and concurrent ways in solving the global routing issues are the two most common. As a result, a practical router, in conjunction to the sequencing phase, includes an improvement phase to remove bottlenecks when more routing is not practicable. However, it's possible that none of these workarounds will be able to eliminate the issue with sequential routing. Examples of these stages of development include the 'rip-and-reroute' and 'shove-aside' approaches [1.25]. Another strategy [1.26] is to route the simple nets (two or three terminals) first and then Steiner Tree Algorithm is deployed to sort out the intermediary nodes. The output of the global routing is fed into a very fine sieve-like grid, this is known as Detail Routing. Therefore, the final routes' feasibility is tied to the global routing solution, since they are found based on it.

Therefore, the global routing must yield a workable result in terms of wirelength, routing, and time for the approach to be considered successful. There are two types of global routing techniques in today's world. [1.27] In the preceding ways, Fairly Good Router uses the former method, in which routing is done in a $3D$ grid and the routing problem is solved in an easy manner by employing full $3D$ labyrinth routing. Direct $3D$ method, it is hypothesised, should produce better outcomes. In reality, however, $2D$ routing using layer assignment algorithms outperforms $3D$ routing in terms of performance and runtime [1.38]. Because of the complexity of modern layouts, full $3D$ routing takes longer. The routing issue is extended onto a $3D$ plane where routing can be accomplished with less effort in $2D$ routing. The solution is then projected from the $2D$ plane to the initial many layers using layer assignment. Box Router 2.0 [1.29], Archer [1.30], Maize Router [1.31], and NTHU-Route [1.32] are some of the routers that use this technology. Box Router

[1.33] is a computer software that employs the concept of box expansion and an advanced Integer linear programming technique.

A grid graph can be used to depict the interconnects of a circuit, with the graph terminals denoting the positioning of the individual circuit blocks. The link length is optimised using a graph's Minimum Spanning Tree (MST). Adding steiner points to the MST and building a Rectilinear Steiner Minimum Tree (RSMT) allows for even more optimization. The refining of RSMT [1.34] - [1.38] has been done in several ways. However, finding the number and placement of steiner points is an NP-hard [1.39] problem, making constructing the RSMT for real-world circuits unfeasible by overwhelming force and unlikely to be optimal when employing simple steiner point allocation criteria. The solution space, also known as the search space, is made up of all conceivable variations of legitimate steiner point placements. Since high-performance VLSI design necessitates more than just a reduction in wire length, and RSMT just addresses that aspect. When it comes to the timing-wire-length trade-off, timing-driven RSMT [TD RSMT] is the clear winner. As a result of scaling, interconnection delay now accounts for between 50 and 70 percent of the clock cycle in high-performance circuits [1.40]. At the current RST design process, crucial path information is typically available because Static timing analysis is performed during the installation and routing stages. Using this criticality data, TD-RSMT will make wire length trades to reduce delay on critical lines. After the RSMT is created, buffering will be performed over each tree to protect branch capacitance and linearize the interconnection latency over long linkages. Buffers can be used to restore the power of weak signals and to shorten delay times. Due to the dominance of connectivity latency, the critical length is decreasing, meaning that more and more buffers must be included in a chip to get the same interconnect delay reduction. Reports indicate that with 32-nm technology, 70% of cells are buffers [1.41]. What's more, empirical evidence suggests that slew mode buffering is more widely used in practise than timing mode buffering [1.42] - [1.45].

The number of steiner points to be inserted raises the combinatorial complexity rapidly, which is dependent in part on the circuit dimensions. To identify nearly optimum solutions in such a high-dimensional search space, some logic or heuristics

[1.46] must be used to drive the search in the appropriate direction. To reduce the complexity of VLSI circuits, the neural network and genetic algorithm concepts are used. [1.47], [1.48] In conjunction with swarm intelligence technology, population-based metaheuristics have shown a lot of promise in this area. Metaheuristics is a greater optimization approach for finding optimal solutions in a search space. Even with its non-deterministic iterations, it is able to escape local optima and remains relatively independent of the individual problem to optimise. Swarm intelligence [1.49] - [1.52] is a type of group action in which a number of decentralised agents share information with one another according to a set of simple rules, resulting in the establishment of an information sharing or global knowledge that no single agent is aware of over time. Ant colony optimization (ACO) is a pervasive metaheuristics calculation that has been used to address higher complexity problems such as RSMT in VLSI circuits [1.53] - [1.55]. Using the ACO [1.56] as their foundation, Arora and Moses developed both a Manhattan and a non-Manhattan routing scheme. Particle Swarm Optimization (PSO) [1.57] - [1.58], which researchers have extensively utilised in optimising RSMT and in several domains [1.59] - [1.62], is another robust swarm approach. Dong et al. [1.63] employed an ingenious programming and updating strategy for a distinct version of PSO to address the routing problem in VLSI, which was the first time PSO was used in routing. In order to solve the RSMT problem, Sarkar et al. [1.64], [1.65] worked on wirelength minimization of VLSI circuits. Ayob et al. [1.66] suggested a PSO-based routing strategy combined with delay insertion to reduce connection delay overall. An improved PSO technique for significantly lowering bends while routing was introduced by Liu et al. [1.67]. Shen et al. [1.68] reported another obstacle-avoidance routing system based on a modified PSO algorithm. Another unique strategy for design space exploration was reported [1.69] - [1.70], which used a two-step PSO-based scheme to optimise the number of virtual channel buffers. In IC layout design, a modified PSO-based method was employed for grid-less net routing [1.71]. The mating behaviour of fireflies, which is facilitated by their bio-luminescence property [1.72], also encourages the optimization method. Researchers have used the Firefly Algorithm (FA) to optimise benchmark functions in the past [1.73] - [1.74]. Falcon et al. [1.75] developed another application of the binary FA that encoded potential solutions into binary form utilising an adjustable light absorption coefficient to speed up the search and problem-specific data to deal with unworkable solutions. For unconstrained optimization problems, Subotic et al. [1.76] created the parallelized FA. Based on firefly optimization, Ayob et al. suggested an obstacle-avoiding VLSI routing algorithm [1.77].

To address the issue of routing, the swarm algorithm, also known as the Artificial Bee Colony (ABC) algorithm, has been implemented. This work was done by Zhang and Ye in [1.78] and Sarkar et al. [1.79], [1.80] Researchers have improved the efficiency of these heuristic algorithms by incorporating the fundamentals of other common approaches like as Genetic Algorithm (GA) and Differential Evolution (DE) selection, mutation, and crossover. This hybrid approaches were implemented because these algorithms provide a viable solution to the population explosion and premature convergence conflicts. PSO-GA [1.81], [1.82], PSO-DE [1.83], ACO-PSO [1.84], [1.85], ACO-ABC [1.86], PSO-ABC [1.87], FA-DE [1.88], FA-Cuckoo [1.89] are hybrid algorithms depends on the application, but in every case, hybridization algorithm outperforms its forerunners.

Recently developed wireless technology allows access from everywhere on Earth. Wireless communication technologies such as pagers, cellphones, laptop computers, and personal digital assistants are used by individuals all over the world to communicate information. The widespread usage of mobile phone voice and text messaging has opened the door for the integration of wireless networking into other domains, such as consumer and enterprise computing and security monitoring. Information may now be accessed and shared anywhere in the world, even in previously infeasible circumstances, thanks to the elimination of many of the drawbacks of traditional wired networks. These kinds of developments have made it possible to set up ad hoc networks in places and circumstances where it would have been extremely challenging, if not downright impossible, to set up a wired network. In conjunction with more recent developments in miniaturization, simple, cost-effective, low-power circuit design, and the development of smaller-sized batteries, these improvements have made it possible for a new technological vision to emerge: Wireless Sensor Network (WSN). [1.90] – [1.94]

WSN networks use wireless communication and low-power computation to monitor environmental conditions and provide with real-time feedback on things like temperature, pressure, and vibration. [1.94]

In layman's terms, a WSN is a circuit board with a variety of sensors (depending on the application), relaying capabilities, and, most crucially, usability due to its wireless nature. WSNs are essentially deployed in large numbers, and clusters are created in agreement with other nodes, with substantial information about topologies being communicated back, usually to a base station. Constant researches are being conducted in the fields of WSNs pertaining to 4 main aspects:

i. Network Capabilities,
ii. Power Consumption,
iii. Better Data Congregation, and
iv. Security

As a result of recent improvements in algorithms and simulations, it is now possible to simply imitate and deploy modifications in essential aspects of WSNs, allowing for a wider range of experimentation. To meet the asset limits of tiny sensor hubs, Sarkar et al. developed a lightweight trust component and upward investigation for grouped WSN. [1.95] - [1.102].

Clustering is another possible solution suggested by the researchers to reduce energy consumption. Each cluster has a cluster head (CH) who collects data from its nodes and communicates with other CHs in order to report data to a centralized base station (BS). So far, several clustering protocols have been reported [1.103] - [1.106]. The determination of the best number of clusters for a provided wireless sensor network is one issue highlighted in the clustering algorithm. It is generally accepted that a smaller number of clusters is preferable when the cluster size is constrained by k-hop communication. Depending on how large the cluster is, this could be the case. Overhead from too many intra-cluster communications can result in inefficiency if the cluster sizes are too large. However, no research has been conducted to determine the optimal cluster size. [1.107]

Currently, industry-wide research is being conducted with a particular emphasis on power consumption, network topology maintenance, and finding a more efficient route for data transmission in order to optimize network life time. Energy-efficient routing [1.108] - [1.110], query optimization [1.111], [1.112] and security advancement [1.113], [1.114] have made significant contributions to WSN energy efficiency through green computing. The authors suggested a low-energy Industrial Internet of Things (IIoT) architecture. [1.114]

Wireless Sensor Network established the groundwork for the Internet of Things (IoT), a platform in which important and common items are connected via a shared network, enabling to acquire data, communicate with the devices themselves, and simplify the duties. It laid the groundwork for crucial technologies such as IoT [1.115], [1.116], and it doesn't stop there. Mobile Ad-Hoc Networks, Flying Ad-Hoc Networks [1.117] - [1.118], Vehicular Ad-Hoc Networks, and other types of ad-hoc networks are all deployed using WSNs. Because of its platform, there are a lot of IoT applications [1.119] - [1.120] and a lot of different ways to use it.

Metaheuristics are used often and fine-tuned to reduce packet loss in a variety of contexts. To save power and acquire data, meta-heuristics and peculiar physics are employed. WSNs are becoming more affordable and portable thanks to advancements in VLSI. Swarm intelligence is an important component of modern-day WSN success. ACO [1.21] - [1.25] and PSO [1.126] - [1.130] are be-fitting metaheuristics to be employed in implementing Swarm Intelligence and condemning the clustering issue in WSNs. When applied to WSNs, metaheuristics can produce successful results and can help to optimise their workings to a large extent. Different emulators, or simulators, are used to simulate the behavioural tendencies of WSNs as they are worked on by various algorithms.

The above descriptions are summarized below as the key aspects.

- Miniaturization of devices is a primary factor in the development of VLSI design technology, where "scaling" goals include increasing packaging density, operating speed, power consumption, and cost per function.

- Researchers at the deep sub-micron technological node have prioritized the reduction of interconnecting latency in VLSI systems with the reduction of wire length.

- The previous two decades of WSN research have focused on improving data collection, routing, network longevity, and sensor node communication. In addition to scalability and power efficiency, routing optimization is one of the main concerns. This involves determining the most effective path from origin to destination.

- Combining population-based metaheuristics with swarm intelligence technologies has demonstrated promising results in this area recently.

Thus, it can be deduced from the above illustration that there are vast scopes and opportunities for researchers to investigate the scope of optimization techniques in VLSI circuits and Wireless Sensor Networks and created primarily metaheuristics algorithms that can efficiently optimize wirelength and reduce delay in VLSI circuits and optimize route discovery in WSN without degrading performance with the ever-increasing technological demand of modern society.

1.2. Organization of the thesis

The present thesis has been organized as different chapters demonstrating metaheuristic, swarm intelligence and its hybrid employed in optimization of VSLI routing and optimization of Route discovery and clustering in Wireless Sensor Network.

Chapter 1 provides an overview the growing development in VLSI domain and subsequent challenges in VLSI routing followed by contemporary research trend and outcomes in optimizing VLSI routing problem. The chapter also gives an overview of the rapid expansion of WSN over the last decade with significant need for efficient, scalable routing protocols and existing research in this domain. The Outline of the thesis is given at the end of this chapter.

Chapter 2 covers deeply the fundamental concepts related to VLSI physical design automation emphasizing the routing phase and the basic challenges pertaining to VLSI routing is elaborated. A general overview of the WSN which helps to obtain a clear understanding of the basic concepts of WSN and various issues in WSN and its applications in different areas are discussed. Furthermore, the chapter also focusses on the basic concepts of metaheuristics and swarm intelligence algorithms which leads to use of such algorithms in optimizing the problems in VLSI and WSN domain.

Chapter 3 discusses minimization of wirelength in VLSI global routing using a self-adaptive system based on PSO by monitoring the acceleration coefficient parameters and analyses its characteristics to those of the present acceleration coefficient-controlled PSO and also enhanced with genetic algorithm in a variety of terminal node allocation topologies inside a definite VLSI layout. This new metaheuristic algorithm is tested with a limiting factor over a wide range of input data to test its least VLSI route output. Furthermore, multiple studies have shown that using state-of-the-art approaches, it is possible to address the routing and time optimization problems in VLSI. Constricted PSO is employed in the RLC delay model to find the global optimum. Using an iterative RLC delay model, this work proposes a two-stage technique for lowering interconnect latency, with the first stage achieving wire minimization and appropriate buffer inclusion with concurrent wire-sizing in the last stages. This method compared to the current Binary PSO technique and is observed that it provides a superior option for optimizing VLSI interconnect latency.

Chapter 4 uses a new metaheuristic approach to wirelength optimization in VLSI that is inspired by the Invasive Weed Optimization (IWO) and resembles the colonization behavior of weeds. This suggests that superior solutions will eventually replace inferior ones during the optimization of the steiner issue for global routing in circuits during the VLSI design process. For the presented method with PSO, a new hybridization strategy is investigated, in which the effectiveness of both approaches is combined in a novel way to improve global optimization. For performance assessment, both of the aforesaid methods were compared to test problems as well as conventional PSO. When compared to previous metaheuristics algorithms and benchmarks, the presented approach delivers a far more viable output and generates results in less time.

Chapter 5 present an alternate solution to VLSI wirelength minimization with routing algorithms based on bio-inspired metaheuristics. In this section, the behavior of the single celled, amoeboid organism Physarum polycephalum slime mould has been used to inspire a new method that uses BioNetwork to solve the VLSI global routing issue more successfully. In addition, a unique hybridization technique for the suggested algorithm with PSO is used, in which the benefits of both algorithms are combined in a novel way to improve global optimization. The hybrid algorithm proves effective when compared to available benchmarks.

Chapter 6 encompasses use of PSO and ACO in large scale cluster-based WSNs to improve node cluster connection while lowering power consumption. In the study, a unique method employs hybridization of Constricted PSO and ACO with Levy Flight to optimize cluster forms and node-clustering connectivity, allowing for better clustering use in WSN. The second section aims to address some of the issues raised by the IoT paradigm by using a new metaheuristics-based data routing hybrid optimization technique on Directed Artificial Bat Algorithm (DABA) and PSO to improve connection issues like real-time delay and network congestion. Using the Dynamic Graph Partitioning Algorithm, this solution also incorporates the clustering notion in conjunction with Fog Computing to spread network stress and improve bandwidth utilization.

Chapter 7 provides the final outcome of research work as well as the possible future research directions.

References

1.1 A. N. Saxena, "*Invention of Integrated Circuits,*" Rensselaer, USA, World Scientific, 2009.

1.2 W. Jacobi, Halbleiterverstärker, DE patent 833366 priority filing on April 14, 1949, published on May 15, 1952.

1.3 J. S. Kilby, "Turning Potential into Realities: The Invention of the Integrated Circuit," http://nobelprize.org/nobel_prizes/physics/laureates/2000/kilby-lecture.pdf, 2000. Mindell, David A (2008). Digital Apollo: Human and Machine in Spaceflight. MIT Press.

1.4 K. Shirriff, ''The surprising story of the first microprocessors,'' *IEEE Spectrum*, vol. 53, no. 9, pp. 48-54, September 2016.

1.5 R. Schaller, ''Technological innovation in the semiconductor industry: a case study of the International Technology Roadmap for Semiconductors (ITRS)," *in Proceeding of IEEE Portland International Conference (PICMET),* Portland, OR, USA, 2001, pp. 195.

1.6 P. Clarke, "Intel enters billion-transistor processor era," EE Times: November 2005.

1.7 A. Gonsalves, "Samsung begins production of 16-Gb flash," EE Times, April 2007.

1.8 B. LaMacchia, and G. Redinbo, "RNS Digital Filtering Structures for Wafer-Scale Integration," *IEEE Journal on Selected Areas in Communications*, vol. 4, no. 1, pp. 67-80, January 1986.

1.9 J. Plummer, "Foreword integrated systems on a chip," *in Proceedings of IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, San Francisco, CA, USA, 1980, pp. 3-3.

1.10 J. Meindl, K. Ratnakumar, L. Gerzberg, and K. Saraswat, "Invited: Circuit scaling limits for ultra-large-scale integration," *in Proceedings of IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, New York, NY, USA ,1981, pp. 36-37.

1.11 Y. Akasaka, and T. Nishimura, "Concept and basic technologies for 3-D IC structure," *in Proceedings of International Electron Devices Meeting*, Los Angeles, CA, USA ,1986, pp. 488-491.

1.12 S. Naskar, and S. K. Sarkar, "A quantum analytical model for inversion charge & threshold voltage of Short channel DMDG SON MOSFET," *IEEE Transactions on Electron Devices*, vol.60, no.9, pp. 2734-2740, Sept. 2013.

1.13 K. K. Young, "Short-channel effects in fully depleted SOI MOSFET's," *IEEE Trans. Electron Devices*, vol. 36, no. 2, pp. 399-402, February 1989.

1.14 S. Sarkhel, B. Manna, and S. K. Sarkar, "A Compact Two Dimensional Analytical Modeling of Nanoscale Fully Depleted Dual Material Gate strained SOI/SON MOSFETs for subdued SCEs," *Journal of Low Power Electronics, ASP* ,vol. 10, no. 3, pp. 383-391, Sept. 2014.

1.15 Upasana, R. Narang, M. Saxena, and M. Gupta, "Modeling and TCAD Assessment for Gate Material and Gate Dielectric Engineered TFET Architectures: Circuit-Level Investigation for Digital Applications," *in IEEE Transactions on Electron Devices*, vol. 62, no. 10, pp. 3348-3356, Oct. 2015.

1.16 P. Saha, S. K. Sarkar, "Drain current modeling of dual material elliptical Gate-All Around heterojunction TFET for enhanced device performance," *Superlattices and Microstructures*, vol. 130, pp. 194-207, Jun. 2019.

1.17 R. Dutta, and S. K. Sarkar, "Analytical Modeling and Simulation based Optimization of Broken Gate TFET Structure for Low Power Applications," *IEEE Transactions ON Electron Devices*, vol. 66, no. 8, pp. 3513 - 3520, August 2019.

1.18 Jack Y.-C. Sun., "System Scaling and Collaborative Open Innovation," *in Proceedings of Symposium on VLSI Technology (VLSIT),* Kyoto, Japan, 2013, pp. 2-7.

1.19 J. Wu, Y. L, Shen, K. Reinhardt, H. Szu, and B. Dong, "A Nanotechnology Enhancement to Moores Law," *Applied Computational Intelligence and Soft Computing*, vol. 2013(2), Jan. 2013.

1.20 W. R. Heller, "Contrasts in Physical Design between LSI and VLSI," *in Proceedings of 18th Design Automation Conference,* Nashville, TN, USA, 1981, pp. 676-683.

1.21 M. E. Daniel, and C. W. Gwyn, "CAD Systems for IC Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 1, no. 1, pp. 2-12, January 1982.

1.22 S. June Hong, and R. Nair, "Wire-routing machines—New tools for VLSI physical design," *Proceedings of the IEEE*, vol. 71, no. 1, pp. 57-65, Jan. 1983.

1.23 G. Yeap, "Smart Mobile SoCs Driving the Semiconductor Industry: Technology Trend, Challenges and Opportunities," *in Proceedings of IEEE International Electron Devices Meeting (IEDM),* Washington, DC, USA*,* 2013,pp. 1.3.1 – 1.3.8.

1.24 R. M. Karp, *Reducibility among Combinatorial Problems. In Complexity of Computer Computations*, Berkeley, USA: Springer, Boston, MA ,1972.

1.25 W. A. Dees, and P. G. Karger, "Automated rip-up and reroute techniques," *in Proceedings of Design Automation Conference*, Las Vegas, NV, USA,1982, pp. 432–439.

1.26 L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels, "The self-organizing exploratory pattern of the Argentine ant," *Journal of Insect Behaviour*, pp. 159-168, March 1990.

1.27 Yen-Jung Chang; Yu-Ting Lee; Jhih-Rong Gao; Pei-Ci Wu; Ting-Chi Wang, "NTHU-route 2.0: a robust global router for modern designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no.12, pp. 1931-1944, Dec. 2010.

1.28 Min Pan, Yue Xu, Yanheng Zhang, and Chris Chu, "FastRoute: an efficient and high-quality global router," *VLSI Design,* Volume 2012, pp 14, January 2012.

1.29 M. Cho, K. Lu, K. Yuan, and D. Z. Pan, "Boxrouter 2.0: architecture and implementation of a hybrid and robust global router," *in Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2007, pp. 503–508.

1.30 M. M. Ozdal, and M. D. F. Wong, "Archer: a history-driven global routing algorithm," *in Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2007, pp. 488–495.

1.31 M. D. Moffitt, "Maizerouter: engineering an effective global router," *in Proceedings of Asia and South Pacific Design Automation Conference*, Seoul, Korea, 2008, pp. 226–231.

1.32 J.-R. Gao, P.-C. Wu, and T.-C. Wang, "A new global router for modern designs," *in Proceedings of Asia and South Pacific Design Automation Conference*, Seoul, Korea, 2008, pp. 232–237.

1.33 M. Cho, L. Katrina, Y. Kun, and P. David, "BoxRouter 2.0: A hybrid and robust global router with layer assignment for routability," *ACM Transactions on Design Automation of Electronic Systems*, Volume 14, Issue 2, Article 32 , pp 1-21, April 2009.

1.34 K. Sakai, K. Tsuji, and T. Matsumoto, "An efficient approximation algorithm for the Steiner tree problem in rectilinear graphs," 1989 IEEE International Symposium on Circuits and Systems (ISCAS), Portland, OR, USA, 1989, pp. 339-342.

1.35 C. Chiang, M. Sarrafzadeh, and C. K. Wong, "A powerful global router: based on Steiner min-max trees," *in Proceedings of IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*, Santa Clara, CA, USA, 1989, pp. 2-5.

1.36 J. Ho, G. Vijayan, and C. K. Wong, "New algorithms for the rectilinear Steiner tree problem," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 2, pp. 185-193, Feb. 1990.

1.37 D. Du, Y. Zhang, and Q. Feng, "On better heuristic for Euclidean Steiner minimum trees," *in Proceedings of 32nd Annual Symposium of Foundations of Computer Science*, San Juan, PR, USA, 1991, pp. 431-439.

1.38 J. L. Ganley, and J. P. Cohoon, "Routing a multi-terminal critical net: Steiner tree construction in the presence of obstacles," *in Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*, London, UK, 1994, pp. 113-116.

1.39 M.R.Garey, and D.S.Johnson, "The Rectilinear Steiner Tree Problem is NP-Complete" *SIAM Journal on Applied Mathematics*, vol. 32, no. 4, pp. 826–834, Jun. 1977.

1.40 J. Cong, L. He, K. Khoo, C. K. Koh, and D. Z. Pan, "Interconnect Design for Deep Submicron ICs," *in Proceedings of ICCAD*, San Jose, CA, USA, 1997, pp. 478–485.

1.41 P. Saxena, N. Menezes, P. Cocchini, and D. A. Kirkpatrick, "Repeater Scaling and Its Impact on CAD." *IEEE TCAD*, vol. 23, no. 4, pp. 451–463, April 2004.

1.42 S. Hu, C.J. Alpert, J. Hu, S.K. Karandikar, Z. Li, W. Shi, and C.N. Sze, "Fast algorithms for slew-constrained minimum cost buffering," *in Proceedings of IEEE TCAD*, San Francisco, CA, 2007, pp. 2009–2022.

1.43 P. J. Osler, "Placement driven synthesis case studies on two sets of two chips: hierarchical and flat," *in Proceedings of ISPD*, Phoenix, Arizona, USA, 2004, pp. 190–197.

1.44 Z. Md-Yusof, M. Khalil-Hani, N. Shaikh-Husin, and M. N. Marsono, "Iterative RLC Models for Interconnect Delay Optimization in VLSI Routing Algorithm," *in Proceedings of 2008 Student Conference on Research and Development (SCOReD 2008)*, Johor, Malaysia, Nov. 2008, pp. 270.1-4.

1.45 N. Shaikh-Husin, and M. K. Hani, "Optimal routing algorithm for minimizing interconnect delay in VLSI layout design," *in Proceedings of International Conference on Robotics, Vision, Information and Signal Processing (ROVISP),* Penang, Malaysia, 2007, pp. 345–349.

1.46 A. Kahng, and G. Robins, "A new class of Steiner tree heuristics with good performance: the iterated 1-Steiner approach," *in Proceedings of IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*, Santa Clara, CA, USA, 1990, pp. 428-431.

1.47 S. K. Sarkar, A. Ghosh, G. M. A., S. Rani A, and D. Samanta, "An efficient technique of integrating parallel neural networks for faster and power efficient nanodevices for ultra-dense VLSI circuits," *in Proceedings of 14th International Workshop on the Physics of Semiconductor Devices (IWPSD)*, Mumbai, India, 2007, pp 232-235.

1.48 S. Ding, and N. Ishii, "An online genetic algorithm for dynamic Steiner tree problem," *in Proceedings of 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies*, Nagoya, Japan, 2000, pp. 812-817.

1.49 P. Tarasewich, and P. R. McMullen, "Swarm intelligence: power in numbers," *Communications of the ACM,* vol. 45, no. 8, pp. 62-67, August 2002.

1.50 N. B. Singh, S. Deb, and S. K. Sarkar, "Swarm Approach towards Better Quantum Well Nanostructure Modelling," *in Proceedings of International Conference & Workshop on Emerging Trends in Technology (ICWET 2010),* Thakur College of Engineering, Mumbai, India, 2010, pp. 667-669.

1.51 C. Jacob, J. Litorco, and L. Lee. "Immunity through swarms: Agent-based simulations of the human immune system," *in Proceedings of Artificial Immune Systems*, Springer Berlin Heidelberg, 2004, pp. 400-412.

1.52 T. Chakrabarti, S. Manna, U. Sharma, T. Chakrabarti, and S. K. Sarkar, "Design of Intelligent Maximum Power Point Tracking (MPPT) technique based on Swarm Intelligence based Algorithms," *in Proceedings of International Conference on Power and Advanced Control Engineering (ICPACE),* Bengaluru, India, 2015, pp. 173-177.

1.53 Y. Hu, T. Jing, X. Hong, Z. Feng, X. Hu, and G. Yan, "An efficient rectilinear Steiner minimum tree algorithm based on ant colony optimization," *in Proceedings of International Conference on Communications, Circuits and Systems,* Chengdu, China, 2004, pp. 1276-1280.

1.54 Y. Hu, T. Jing, X. Hong, Z. Feng, X. Hu, and G. Yan, "An efficient rectilinear Steiner minimum tree algorithm based on ant colony optimization," *in Proceedings of Int. Conf. Communication, Circuits and Systems,* Chengdu, China, Oct. 2004, pp. 1276-1280.

1.55 Y. Hu, T. Jing, Z. Feng, X.-L. Hong, X.-D. Hu, and G.-Y. Yan, ''ACO-Steiner: Ant colony optimization based rectilinear Steiner minimal tree algorithm,'' *J. Computer Science and Technology,* vol. 21, no. 1, pp. 147 152, Jan. 2006.

1.56 T. Arora, M.E. Moses, "Ant Colony Optimization for power efficient routing in Manhattan and non-Manhattan VLSI architectures," *In Proceedings of Swarm Intelligence Symposium, 2009. SIS '09. IEEE*, Nashville, TN, USA, April 2009, pp. 137-144.

1.57 R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," *in Proceedings of Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, Oct. 1995, pp. 39-43.

1.58 M. Clerc, J. Kennedy, "The particle swarm explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation. 2002*, vol. 6, no.1, pp. 58 - 73.

1.59 J. Liang, A. Qin, P. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,'' *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281 295, Jun. 2006.

1.60 N. B. Singh, S. Deb, G.P. Mishra, S. K. Sarkar, and S. K. Sarkar, "Application of Particle Swarm Optimization Algorithm for better Nano-devices," *in Proceedings of 2nd Int. Conf. on Contemporary Computing (IC3)*, Noida, India, 2009, pp. 349–357.

1.61 S. Deb, N. B. Singh, D. Das, and S. K. Sarkar**,** "Particle swarm approach for parameter optimization of quantum well nano structure," *Expert Systems with Applications, Elsevier*, vol. 38, no. 10, pp. 12999-13004, Sept. 2011.

1.62 S. Laha, J. Chowdhury, A. Khan, and S. K. Sarkar**,** "A watermarking Scheme based on singular value decomposition and particle swarm optimization," *in Proceedings of IEEE International Advance Computing Conference*, Ghaziabad, India, 2013, pp. 888-892.

1.63 C. Dong, G. Wang, Z. Chen, S. Sun, & D. Wang, "A VLSI routing algorithm based on improved DPSO," *in Proceedings of ICIS 2009 IEEE International Conference*, Shanghai, China, 2009, pp. 802-805.

1.64 A. Khan, S. Laha, and S. K. Sarkar**,** "A Novel Particle Swarm Optimization approach for VLSI Routing," *in Proceedings of IEEE International Advance Computing Conference*, Dr. A.K Garg Institute, Ghaziabad, India, 2013, pp. 258-262.

1.65 A. Khan, P. Bhattacharya, and S. K. Sarkar, "A Swarm Based Global Routing Optimization Scheme", *in Proceedings of International Conference on Advances in Electrical Engineering (ICAEE'2014),* Vellore, India, 2014, pp 1-4.

1.66 M. N. Ayob, Z. M. Yusof, A. Adam, A. F. Z. Abidin, I. Ibrahim, Z. Ibrahim, & M. K. Hani, "A particle swarm optimization approach for routing in VLSI," *in Proceedings of CICSyN*, Liverpool, UK, 2010, pp. 49-53.

1.67 G. Liu, G. Chen, W. Guo, & Z. Chen, "DPSO-based Rectilinear Steiner Minimal Tree construction considering bend reduction," *in Proceedings of ICNC*, Shanghai, China, 2011, pp. 1161-1165.

1.68 Y. Shen, Q. Liu, & W. Guo, "Obstacle-avoiding rectilinear Steiner minimum tree construction based on Discrete Particle Swarm Optimization," *in Proceedings of ICNC*, Shanghai, China, 2011, pp. 2179-2183.

1.69 A. Faruque, M. Abdullah, and J. Henkel. "Minimizing virtual channel buffer for routers in on-chip communication architectures," *in Proceedings of Design, Automation and Test in Europe*, Munich, Germany, 2008, pp. 1238-1243.

1.70 Z. Md. Yusof, et al, "A two-step binary particle swarm optimization approach for routing in VLSI," *in Proceedings of ICIC Express Letters*, Langkawi, Malaysia, 2012, pp. 771-776.

1.71 X.-c. Huang; Z.-j. Liu, "Gridless net routing of integrate circuit with Particle Swarm optimization algorithm," *in Proceedings of Electricity Distribution (CICED),* Nanjing, China, Sept. 2010, pp. 1-6.

1.72 X. S. Yang, "Firefly algorithms for multimodal optimization," *in Proceedings of Stochastic algorithms: foundations and applications*, Springer, Germany, 2009, pp. 169-178.

1.73 Chai-Ead, P. Aungkulanon, and P. Luangpaiboon, "Bees and firefly algorithms for noisy non-linear optimization problems," *in Proceedings of IMECS 2011*, Hong Kong, China, 2011, pp. 1449-1454.

1.74 C. Liu, Z. Gao, and W. Zhao, "A New Path Planning Method Based on Firefly Algorithm," *in Proceedings of CSO,* Harbin, China, 2012, pp. 575-578.

1.75 R. Falcon, M. Almeida, A. Nayak, "Fault identification with binary adaptive fireflies in parallel and distributed systems," *in Proceedings of Evolutionary*

Computation (CEC), 2011 IEEE Congress on, IEEE, New Orleans, LA, USA, 2011, pp. 1359-1366.

1.76 M. Subotic, M. Tuba, N. Stanarevic, "Parallelization of the firefly algorithm for unconstrained optimization problems," *in Proceedings of Latest Advances in Information Science and Applications*, 2012, pp. 264-269.

1.77 Ayob, M. N., Hassan, F., Ismail, A. H., Basri, H. H., Azmi, M. S., Abidin, "A Firefly Algorithm approach for routing in VLSI," *in Proceedings of Computer Applications and Industrial Electronics (ISCAIE)*, Kota Kinabalu, Malaysia, 3-4 Dec. 2012, pp. 43-47.

1.78 H. Zhang, and D. Ye, "An artificial bee colony algorithm approach for routing in VLSI," *in Proceedings of Advances in Swarm Intelligence,* Springer, Berlin Heidelberg, 2012, pp. 334-341.

1.79 P. Bhattacharya, A. Khan, S. Sarkar, and S. K. Sarkar, "An Artificial Bee Colony Optimization Based Global Routing Technique," *in Proceedings of International Conference On Control, Instrumentation, Energy, Communication (CIEC 14),* Calcutta, India, 2014, pp. 621-625.

1.80 A. Khan, P. Bhattacharya, and S. K. Sarkar, "A Swarm Based Global Routing Optimization Scheme," *in Proceedings of International Conference On Advances In Electrical Engineering (ICAEE'2014),* Vellore, India, 2014, pp 1-4.

1.81 B. Kantha, and S. K. Sarkar, "Comparative Study Analysis of Particle Swarm Optimization and Genetic Algorithm for the Optimization of Parameter of MEMS based Micro-heater," *Journal of Computational and Theoretical Nanoscience, ASP,* vol. 12, no. 8, pp. 1-6, 2015.

1.82 J. Robinson, S. Sinton, and Y.R. Samii, "Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna," *in Proceedings of the IEEE International Symposium in Antennas and Propagation Society*, San Antonio, TX, USA, 2002, pp. 314–317.

1.83 T. Hendtlass, "A Combined Swarm differential evolution algorithm for optimization problems," *in Proceedings of 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert*

Systems, Lecture Notes in Computer Science, Springer Verlag, German, 2001, pp. 11–18.

1.84 P. S. Shelokar, P. Siarry, V. K. Jayaraman, B. D. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," *Applied Mathematics and Computation*, vol. 188, no.1, pp.129–142.

1.85 A. M. Aaref, S. Mohammed, A. B. Noori, and A. M. Humada, "A hybrid (ACO-PSO) algorithm based on maximum power point tracking and its performance improvement within shadow conditions," *Int. J. Eng. Technol.,* vol. 7, no. 4.37, pp. 43-47, 2018.

1.86 M. Kefayat, A. L. Ara, and S. N. Niaki, ''A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources,'' *Energy Convers. Manage.,* vol. 92, pp. 149-161, Mar. 2015.

1.87 M. Li, H. Duan, and D. Shi. "Hybrid Artificial Bee Colony and Particle Swarm Optimization Approach to Protein Secondary Structure Prediction," *in Proceedings of Intelligent Control and Automation (WCICA), 2012 10th World Congress on. IEEE*, Beijing, China, 2012, pp 5040-5044.

1.88 L. Zhang, L. Liu, X.-S. Yang, and Y. Dai, ''A novel hybrid algorithm for global optimization,'' *PLoS ONE*, vol. 11, no. 9, Sep. 2016.

1.89 T. Chakrabarti, T. Chakrabarti, U. Sharma, and S. K. Sarkar, "Extraction of Efficient Electrical Parameters of Solar Cell using Firefly and Cuckoo Search Algorithm," *in Proceedings of IEEE-7th India International Conference on Power Electronics (IICPE) 2016,* Punjab, India, Nov. 2016, pp 1-5.

1.90 S. K. Sarkar, *Wireless Sensor and Ad-Hoc Networks under Diversified Network Scenarios,* St. Norwood, MA, USA: Artech House, 2012.

1.91 K. Romer, and F. Mattern, "The design space of wireless sensor networks," *Wirel.Commun. IEEE*, vol. 11, no. 6, pp. 54-61, 2004.

1.92 S. Gowrishankar, T. G. Basavaraju, and S. K. Sarkar, "Performance Analysis of Routing Protocols in Wireless Sensor and Actor Networks from an Actor-to-Actor Perspective," *in Proceedings of the International Conference on Computer Networks and Security ICCNS 08*, VIT, Pune, MS, India,2008.

1.93 S. Gowrishankar, T. G. Basavaraju, and S. K. Sarkar, "Effect of IEEE 802.15.4/Zigbee on the Performance of Wireless Sensor Networks," *in Proceedings of IEEE International Conference on future Networks (ICFN 2009),* Bangkok, Thailand, 2009.

1.94 P. Calhoun, M. Montemurro, D. Stanley, and D. Stanley, "Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11, " RFC 5416, March 2009.

1.95 S. Gowrishankar, S. K. Sarkar, and T. G. Basavaraju, "Scenario Based Simulation Study of Ad Hoc Routing Protocol's Behavior in Wireless Sensor Networks," 2009 International Conference on Future Computer and Communication, Kuala Lumpar, Malaysia, 2009, pp. 527-532.

1.96 Basilis Mamalis, Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou, "*Clustering in Wireless Sensor Networks,"* Taylor & Francis Group, 2009.

1.97 J.Bag, R. R. Sahoo, P.K. Dutta, and S. K. Sarkar, "Design and VLSI Implementation of Power Efficient Processor for object localization in Large WSN," *Int. J. High Performance Systems Architecture, Inderscience Journal,* vol. 4, no. 4, pp. 204-217, 2013.

1.98 R. R. Sahoo, M. Singh, B. M. Sahoo, K. Majumder, S. Ray, and S. K. Sarkar, "A Light Weight Trust based Secure and Energy Efficient Clustering in Wireless Sensor Network: Honey Bee Mating Intelligence Approach,*" Elsevier, Procedia Technology*, vol. 10, pp. 515-523, 2013.

1.99 K.S. Arathy, and C.N. Sminesh," A novel approach for detection of single and collaborative black hole attack in MANET," *Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology*, vol. 25, pp. 264–271, 2016.

1.100 R. R. Sahoo, A. R. Sardar, M. Singh, S. Ray, and S. K. Sarkar, "Trust Based Secure and Energy Efficient Clustering in Wireless Sensor Network: A Bee Mating Approach" *in Proceedings of Pattern Recognitions and Machine Intelligence International Conference, PreMI 2013,* LNCS 8251, Springer, Berlin, Heidelberg, 2013, pp. 100–107.

1.101 M. Singh, A. R. Sardar, R. R. Sahoo, K. Majumder, S. Ray, and S. K. Sarkar, "Lightweight Trust Model for Clustered WSN," *in Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014,* Advances in Intelligent Systems and Computing, Springer, 2015, pp. 765-773.

1.102 M. Singh, A. R. Sardar, K. Majumder, and S. K. Sarkar, "A Lightweight Trust Mechanism and Overhead Analysis for Clustered WSN," *IETE Journal of Research*, vol. 63, no. 3, pp. 297-308, Feb. 2017.

1.103 L. Qing, Q. Zhu, and M. Wang, "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks," *In: Computer Communications*, vol. 29, no. 12, pp. 2230 –2237, 2006.

1.104 W. Xu, G. Chen, T. Liu, J. Peng, and J. Yang, "Avoidance of energy hole problem based on feedback mechanism for heterogeneous sensor networks," *In: International Journal of Distributed Sensor Networks*, vol. 13, no. 6, 2017.

1.105 M. Pinotti, A. Papadopoulos, A. Navarra, and J. A. McCann, "VIBE: An energy efficient routing protocol for dense and mobile sensor networks," *In: J. Network and Computer Applications*, vol. 35, no. 4, pp. 1177–1190, July 2012.

1.106 M. Pinotti, A. Formisano, and A. Navarra, "Distributed colorings for collision-free routing in sink-centric sensor networks," *In: J. Discrete Algorithms*, vol. 14, pp. 232–247, July 2012.

1.107 R. K. Yadav, D. Gupta, and D. K. Lobiyal, "Energy Efficient Probabilistic Clustering Technique for Data Aggregation in Wireless Sensor Network," *Wireless Personal Communications*, vol. 96, no.3, pp. 4099-4113, Oct. 2017.

1.108 K. Guleria, and A. K. Verma," Comprehensive review for energy efficient hierarchical routing protocols on wireless sensor networks," *Wireless Networks*, vol. 25, no. 3, pp. 1159-1183, Apr. 2019.

1.109 M. Khabiri, and A. Ghaffari,"Energy-Aware Clustering-Based Routing in Wireless Sensor Networks Using Cuckoo Optimization Algorithm," *Wireless Personal Communications*, vol. 98, no. 3, pp. 2473-2495, 2018.

1.110 B. Chen, W. Liang, and J. X. Yu, "Energy-efficient skyline query optimization in wireless sensor networks," *Wireless Networks*, vol. 18, no. 8, pp. 985-1004, Nov. 2012.

1.111 V. Jindal, A.K. Verma, and S. Bawa, "Quantitative Reduction in Communication Load for Energy Efficiency in WSN," *Wireless Personal Communications*, vol. 85, no.4, pp. 2795-2810, Dec. 2015.

1.112 S. Prasanna, and S. Rao," An overview of wireless sensor networks applications and security," *International Journal of Soft Computing and Engineering (IJSCE),* vol. 2, no. 2, May, 2012.

1.113 M. K. Garg, D. Vir, and S. K. Agarwal, "Simulation analysis of AODV, DSR and ZRP routing protocols in manet using qualnet 5.0 simulator," *International Journal of Scientific and Research Publications*, vol. 3, no. 3, pp. 1-6, March 2013.

1.114 A. Wang, D. Yang, and D. Sun, "A clustering algorithm based on energy information and cluster heads expectation for wireless sensor networks," *Computers & Electrical Engineering*, vol. 38, no. 3, pp. 662-671, 2012.

1.115 S. Savazzi, V. Rampa, and U. Spagnolini," Wireless Cloud Networks for the Factory of Things: Connectivity Modeling and Layout Design," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp.180–195, 2014.

1.116 Jeffrey Pawlick, and Quanyan Zhu, "Strategic Trust in Cloud-Enabled Cyber-Physical Systems with an Application to Glucose Control," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2906-2919, Dec. 2017.

1.117 H. Ghazzai, A. Feidi, H. Menouar, and M. L. Ammari, "An exploratory search strategy for data routing in flying ad hoc networks," *in Proceedings of IEEE 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Montreal, QC, Canada, 2017, 1–7.

1.118 H. Yang, and Z. Liu, "An optimization routing protocol for FANETs," *EURASIP Journal on Wireless Communications and Networking*, Article 120, May, 2019.

1.119 B. Biswas, S. Bhowmick, M. Biswas, S. Roy, A. Dey, and S. K. Sarkar, "IoT-Based Smart Heart-Health Monitoring System," *in Proceedings of IEEE-*

International Conference on New Trends in Engineering & Technology (ICNTET), Chennai, India, September 2018.

1.120 S. Mukherjee, A. Ghosh, and S.K Sarkar, "Arduino based Wireless Heart-rate Monitoring system with Automatic SOS Message and/or Call facility using SIM900A GSM Module," *in Proceedings of 2019International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN),* Vellore, India, March 2019, pp. 1-5.

1.121 S. Okdem, and D. Karaboga, "Routing in Wireless Sensor Networks Using Ant Colony Optimization," *in Proceedings of First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06),* Istanbul, Turkey, 2006, pp. 401-404.

1.122 Z. Zhong, Z. Tian, Z. Li, and P. Xu, "An Ant Colony Optimization Competition Routing Algorithm for WSN," *in Proceedings of 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, 2008, pp. 1-4.

1.123 C. Li, and Z. Deng, "Feedback-enhanced ant colony routing algorithm for wireless sensor networks," *in Proceedings of 2010 5th International ICST Conference on Communications and Networking in China*, Beijing, China, 2010, pp. 1-5.

1.124 M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 26, no. 1, pp. 29-41, Feb. 1996.

1.125 J. Du, and L. Wang, "Uneven clustering routing algorithm for Wireless Sensor Networks based on ant colony optimization," *in Proceedings of 2011 3rd International Conference on Computer Research and Development*, Shanghai, 2011, pp. 67-71.

1.126 R. V. Kulkarni, and G. K. Venayagamoorthy, "Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey," *in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 41, no. 2, pp. 262-267, March 2011.

1.127 M. N. Rahman, and M. A. Matin, "Efficient algorithm for prolonging network lifetime of wireless sensor networks," *in Tsinghua Science and Technology*, vol. 16, no. 6, pp. 561-568, Dec. 2011.

1.128 Y. Zhou, N. Wang, and W. Xiang, "Clustering Hierarchy Protocol in Wireless Sensor Networks Using an Improved PSO Algorithm," *in IEEE Access*, vol. 5, pp. 2241-2253, Dec, 2016.

1.129 Z. Han, Y. Li, and J. Liang, "Numerical Improvement for the Mechanical Performance of Bikes Based on an Intelligent PSO-ABC Algorithm and WSN Technology," *in IEEE Access*, vol. 6, pp. 32890-32898, June 2018.

1.130 A. Mukherjee, P. Goswami, Z. Yan, L. Yang, and J. J. P. C. Rodrigues, "ADAI and Adaptive PSO-Based Resource Allocation for Wireless Sensor Networks," *in IEEE Access*, vol. 7, pp. 131163-131171, Sept 2019.

CHAPTER 2

FUNDAMENTALS OF VLSI PHYSICAL DESIGN, WSN AND METAHEURISTICS

2.1. VLSI Physical Design: Overview

Advancement of tools and technology enables VLSI technology to give the rights to create systems on a chip with millions and even billions of transistors. The Intel 80286 microprocessor has over 10 transistors, the RISC processor from National Semiconductor NS32SF641 has over 10 transistors [2.1], The Pentium processor has over 310 transistors. Computer programs that perform design work to reduce the complexity of the design cycle are known as CAD. The design process can be completely computerized so that there is no or very little human intervention and that is called Design Automation.

2.1.1. VLSI Design Cycle and Physical Design

The main objective of VLSI design cycle is to manufacture a packaged chip. As given in the Figure 2.1., typical design cycle is may be represented by a flow chart incorporating a number of steps [2.2]

VLSI physical design the geometric representation, also called layout of chip, should be developed from the netlist representation of the chip. It is created by converting part of each logic into a geometric body of different layers. Design rules are very important to consider in this section. In most cases, the visual design is fully or partially automatic when the structure is generated directly from the net list using Layout Synthesis tools. Structural integration tools have certain limitations such as location and performance penalty. Thus, a hand-made geometric modification, although deliberate and abundant, produces a better area and function than the composite structure.

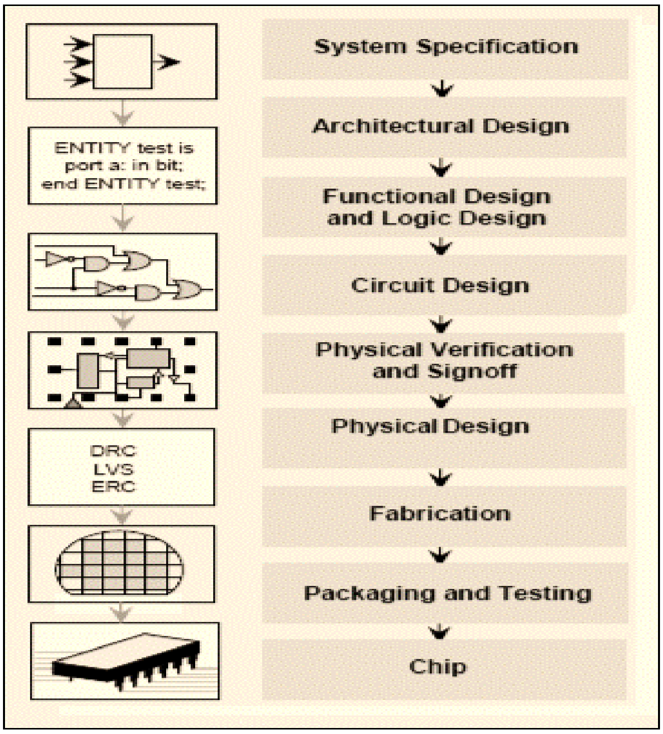

Figure 2.1. VLSI design cycle

Many factors need to be considered in this phase of the design cycle. Integrated circuits have two areas, a functional area and a wiring area. Functional modules such as registers, multiplexers, flip-flops etc. they take the circuit work and the wire used to connect these functional modules to use the wiring area. In a group of manufacturing chips, the number of defect-free chips produced is called yield. If the chip area is larger, the yield is poorer which results in higher production cost and increased selling cost of the chip. The physical design process is divided into steps to turn the problem into manageable minor issues. The steps are given below and shown in Figure 2.2.

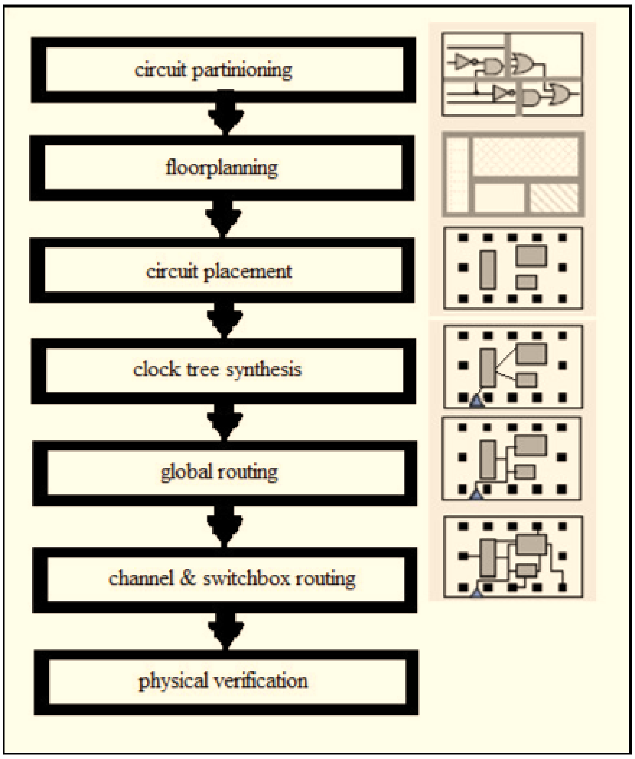

Figure 2.2. VLSI physical design cycle

A. Partitioning: Every design in the RTL design phase is subdivided into smaller blocks and each module is designed. These modules are linked to one of the main modules called the Top-level Module. This separation is mainly done to make the next steps easier. This is a crucial stage in creating a physical prototype.

B. Floor planning: At this stage IP cores, macros used in construction, routing possibilities, area of the whole design are considered and a suitable floorplan is identified. There is a trade-off between area and design speed due to the routing resources used in the system. Many routing resources slow down the design process. Area configuration allows the design to use fewer resources. This brings the design phases closer to each other and reduces the contact distances that provide faster signal routes from end to end. At the end of this section, the entire route circuit is divided into channels and switch boxes.

C. Placement: At this point the actual position of the circuit components is confirmed. Four stages of preparation are:

i. Pre-placement optimisation - Improves netlist before placement and lowers cells.

ii. In-placement optimisation - This sub-section makes cell size, cell movement, cell pass, net separation, gate duplication, buffer installation, area acquisition etc.

iii. Post-placement Optimization before clock tree synthesis - This section uses netlist setting with appropriate clocks and can fix set, capture, major trans or cap violations. Make ground-based placement arrangements

iv. Post-placement Optimization before clock tree synthesis - This section tries to keep the skew of the clock.

D. Clock tree synthesis: This section seeks to reduce skew delays and installations.

E. Routing: The routing is made up of two small steps that are a global routing and detailed routing. Global Routing creates a flexible route for the entire chip by providing router resources that are used for communication. A detailed routing lists routes to a certain metal layer and global routing resource routes. The Global Routing finds a complex route system through the channels through which the nets will be delivered. Then it is important to determine the order of the routing paths using the channels and switch boxes. After this the actual wire assignment on the routing tracks is done in detail routing.

F. Physical Verification: Virtual authentication confirms the following features.

i. Design Rule Checking - That the design complies with all technical requirements.

ii. Layout vs Schematic - Is it similar to a real netlist.

iii. Antenna Rule Checking - If it has any antenna effects.

iv. Electricity Rule Checking – that if its compliance with all electrical requirements.

2.1.2. VLSI Routing

In a VLSI chip the number of cells to be connected is very large and thus routing is accomplished using computer programs called routers. The pins which are electrically equivalent are connected by means of conductors that carry electrical signals. These conductors or wiring segments are assigned a fixed path during routing. Routing takes up almost 30 percent of design time and a large portion of layout area [2.3].

Routing is a NP-hard problem [2.3] which is factorized in a stepwise manner as a hierarchy of smaller problems which are solvable in polynomial time. But this decomposition however depreciates the global optimality. VLSI routing incorporates two-step approach as given in Figure 2.3.

i. global routing

ii. detailed routing

The purpose of the global route is to specify a route plan so that each net is placed in a specific route. Global routing algorithms generally improve the function of a specific purpose (average total length of wire or total power consumption or the number of vias used in multi-layer connections.).

During detail routing wires are used for communication and wire resistance, capacitance is measured. A detailed routing path is used between router regions, such as channels, switch boxes, and global route cells. Chip time requirement and chip production rules are examined in this section. The purpose of the detailed routing is to allocate signal segments of signal networks to specific track tracks and to determine vias, and metal layers in a manner consistent with the given global route effect. Thus, detailed routing does a fine-grain assignment of routes [2.4] of each net.

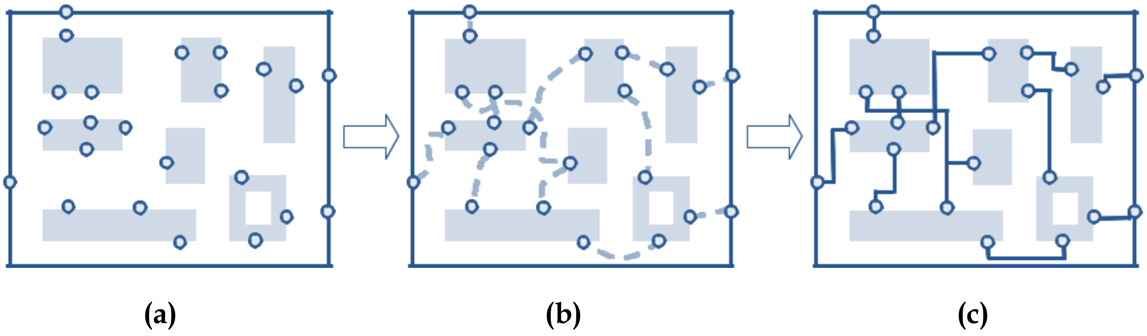
(a) (b) (c)

Figure 2.3. (a) A VLSI layout with blocks and pins. (b) Global routing. (c) Detailed routing.

2.1.3. VLSI Global Routing

The international router forms a loose line, connecting the components to the chip. The circuit on the chip is usually modelled as a $2D$ grid graph and the global path of each net is found on the graph [2.5]. On a global route, communication is terminated between appropriate (established or flexible) blocks in a circuit, regardless of the exact geometrical features of each wire. Global route quality impacts other important characteristics of the chip such as critical system delays, consumption of power etc. [2.6]. For each wire the international router regulates the list of mediums and switch boxes to be used as a transit point for that wire. [2.7]. So, the global router makes a complex navigation system for each network.

Rules which are followed during the global routing are:

i. Nets with significant distribution delays are delivered very carefully at the beginning. Their total length of connection is set to a minimum. These nets first get to use channels with shorter routes.

ii. Wires should be delivered in an orderly fashion. If the decisions are not good, the wires in the path are withdrawn, and new wires are moved. After that, the torn wires are re-inserted into the track.

There are four different ways to use the global route which are given below

i. Sequential approach

ii. Mathematical programming approach

iii. Hierarchical approach

iv. Stochastic iterative approach

In sequential technique the nets are chosen in a particular order and distributed one at a time. If the functional routing capacity is not updated it is called the order-independent approach or else it is called the order-dependent approach.

With a mathematical programming technique, the global route is designed as a complete $0-1$ integration system. In this way *0* or *1* is assigned to each net as well as to other potential trees for that net.

Hierarchical methods can be performed in a downward or upward manner. In a downward spiral, the grid cells are grouped together into a super cell. In each stage of ranking, a global path is achieved between individual cells grouped together. In the top-down procedure, the sequence of stages starting from a super cell to a different one can be either a separate grid cell or a small group of unified cells. This approach is usually guided by the configuration of the floorplan design.

The stochastic repetitive method steadily updates the contemporary solution by tearing and rearranging each net, until an adequate net bond is obtained. Simulated annealing is one such method [2.8].

2.1.3.1. Global Routers

Today Global Routing has some major obstacles to overcome such as delays, congestion. Designing on a nanometre scale has become a clever practice in the construction of the modern VLSI. The International Symposium on Physical Design (ISPD) arranged the Global Routing competition in 2007 and 2008 [2.9] where a few academic global routers were inaugurated. New international routers have appeared that have pushed the boundaries of finding a better solution quality and lowered operating time. To advance the investigation and development of new global routes many routers have contributed to the air race. The arisen-up routers in this contest are Box Router [2.10], Maize Router [2.11], Archer [2.12], and NTHU-Route [2.5]. Earlier the routers such as the Labyrinth Router, the Chi dispersion Router used a rip-off and redirect method to combat congestion which is a conventional complication on international routes. [2.13]. But these new routers have used the chat-based routing method presented in Path Finder [2.14].

The current route strategies can now be divided into two classes [2.15]:

i. Full $3D$ routing

ii. $2D$ routing followed by layer assignment.

Fuzz Route [2.16] uses the previous method in the above techniques where the route is performed on a $3D$ grid and resolves the route problem in a simple way employing a full $3D$ maze route. [2.13]. It is thought that a straightforward $3D$ approach should produce better result. However practically, a $2D$ routing with layer allocation approach is better in solution quality and performance time. Due to the complexity of modern designs, a full $3D$ line usually takes longer.

In the $2D$ route the routing problem is shown in the $2D$ plane where the routing can be done with minimal attempt. After that layer allocation method is used to forecast a solution from a $2D$ plane to the actual different layers. Some routers which use this process are Box Router 2.0, Archer, Maize Router, NTHU-Route.

Box Router, uses the idea of a box extension and an improved ILP (Integer Linear Programming) method. It uses a simple pre-route policy to target multiple congested regions with maximum accuracy. Formed on this box expansion process, Box Routing is later done with a flexible maze track. Eventually the post-routing procedure succeeds without ripping to achieve an effortless transition between the total length of the wires and the rout ability.

Maize Router [2.13] begins by greedily creating fully connected routes of all nets independently of each other. FLUTE is used to determine the topology of each net RSMT. Next this router makes an extreme transition, a process aimed at effectively reducing route congestion. It also uses a curbing mechanism associated with excessive edge displacement which helps to reduce unnecessary wire length. Another efficient and high-quality international router is Fast Route [2.17]. Modern algorithms attempt to reduce wire lengths, vias and capacity to reduce power consumption.

2.1.3.2. Grid Graph Model

During global routing the chip area is modelled as a $2D$ grid graph $G(v_g, e_g)$ [2.18]. Each cell is represented by a vertex (v_g) in the graph. Two neighbouring cells (v_{g1}) and (v_{g2}) are joined by an edge (e_g). Available Channel resources are represented by these edges. The demand (d_g). or utilisation of an edge (e_g) is the total number of nets that use that edge. The capacity (c_g) of an edge is the maximum number of nets that can travel through it. Whenever demand of an edge exceeds its capacity congestion occurs (c$_g$ / d_g). Overflow is measured as the difference between demand and capacity. Figure 2.4. (a) shows an integrated system on chip and Figure 2.4. (b) shows the corresponding grid graph model.

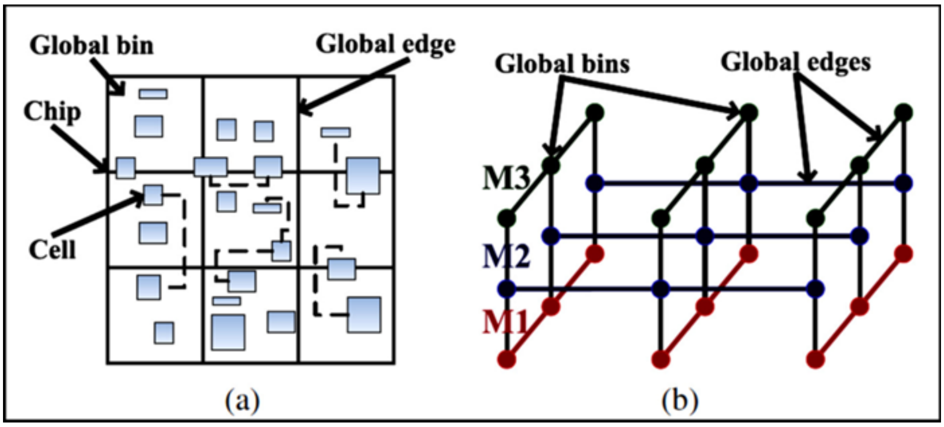

Figure 2.4 (a). An integrated system on chip. (b) Grid graph model.

Given a netlist $N = \{N_1, N_2, \dots., N_n\}$, a routing graph $G = (v_g, e_g)$, and a Steiner tree T_i for each net N_i where $i = \{1,2, \dots, n\}$, the objective of a global router [2.18] are:

i. $d_g - c_g$ for all $e_g \in E$, where E comprises of all edges. i.e., demand should not exceed capacity of an edge so that sum of the total overflow among all the edges is minimized.

ii. $\sum e'g$, where $e' \in E$ and e' are edges in the T_i are should be minimized, total wirelength of Steiner tree T_i, i.e., the total interconnecting wirelength of the chip should be minimized

2.1.3.3. Global Routing Problem

Global Routing Optimization Problem, which tries to improve worldwide steering cost inside a predefined search space, followed by a curve decrease issue if non-Euclidean space occurs. Non-Euclidean rectilinear search space drives this work in which the cost between two terminal points is intended as the Manhattan distance between the two points [2.19], as shown in (2.1).

$$dist = |x_1 - x_2| + |y_1 - y_2| \tag{2.1}$$

Here, (x_1, y_1) and (x_2, y_2) are corresponding (x, y) coordinates of the two vertices or terminal points (v_{g1}) and (v_{g2}) respectively. The foremost optimization objective is to diminish routing costs for creating a link between networks for a given set of terminal points. Netlist is the simplest form of such connection. It comprises the set of connected nodes designating both active and passive components of the electronic circuit and the edges connecting these nodes. The routing path amid the terminal points is represented by the edge. The optimization in a netlist is found by first founding an MST from connection, followed by RSMT or MRST.

Engineering geometry achieves reflective sub-micron levels, circuit proficiency, and dependability, which have become massively dependent on interconnect length impacting the device delay in the circuit. As routing delay or interconnect length depends upon aspects, algorithms often assist in bringing out enhancement. Historically, minimising wire length was the best strategy for reducing routing latency and increasing circuit throughput. Nevertheless, it is certain that the routing way in addition to buffer insertion fundamentally upgrades this methodology. Delay enhancement is rudimentary to achieve the timing conclusion of a top-notch VLSI format.

Elmore wire model offers the accuracy for estimating delay in the global interconnect routing methods [2.20]. In nanometre VLSI innovation [2.21], this has been shown to produce a 35% postpone misjudgement, but it avoids the inductance implications. Two-way RLC-based punishment, such as in S-RABILA [2.22], projects a progressive high-order RLC delay model now that VLSI configuration has achieved reflective submicron technology.

Fabrication technology enlists profound submicron levels of accuracy heightens its resistance and diminishes the wire size. Wire resistance is a non-trivial contributor to the connection time and must be accounted for. As a result, the lumped delay model has become irrelevant [2.22].

The RLC interconnect model of [2.22] is enhanced by the BPSO method developed by Md. Yusof et al. [2.23], which has shown its utility in the concurrent routing and buffer placement algorithm in S-RABILA. The RLC delay model is computed for a certain node i, using the formula (2.2).

$$t_{Di} = \frac{(1.047e^{-\left(\frac{\zeta_i}{0.85}\right)}+1.39\zeta_i}{\omega_i} \tag{2.2}$$

where,

$$\omega_i = \frac{1}{\sqrt{T_{LCi}}} \tag{2.3}$$

$$\zeta_i = \frac{T_{RCi}}{2\sqrt{T_{LCi}}} \tag{2.4}$$

$$T_{RCi} = \sum_k C_k R_{ik} \tag{2.5}$$

$$T_{LCi} = \sum_k C_k L_{ik} \tag{2.6}$$

Capacitance components at any given section k are indicated by C_k (2.5) and (2.6). R_{ik} and L_{ik} represent the input nodes' shared resistance and inductance respectively. Source-to-sink delay T_{RCi} is the sum of R_{ik} times C_k, and sink-to-source delay T_{LCi} is the same, but in the other direction. It is possible to reconsider (2.5) and (2.6) in the following way:

$$T_{RCi} = \sum_k C_{Tk} R_k \tag{2.7}$$

$$T_{LCi} = \sum_k C_{Tk} L_k \tag{2.8}$$

where L_k is the inductance of a specific section k and R_k is the resistance. For each given value of R_k and L_k, the resulting capacitance is denoted by C_{Tk}. When planning the time lag between the sink and the source, R_k (resistance of a segment) times C_{Tk} and L_k (inductance of a segment) times C_{Tk} are both taken into account to yield T_{RCi} and T_{LCi} respectively. The delay at a node can be measured by subtracting T_{RCi} and C_{Tk}. Two types have been decided upon for the link between wires, with or without a buffer. Below is a list of them:

(i) Source-to-sink wire-only connection latency estimation:

Component values in the tuples for the previous vertex $(r, l,\ T_{RCi},\ T_{LCi})$ and the current fragment wire parameters $(c_\omega,\ r_\omega,\ l_\omega)$ are used to determine the components in the tuples for the subsequent vertex $(r', l',\ T_{RCi}',\ T_{LCi}')$, as shown in Figure 2.5., where, $r' = r_w + r$ and $l' = l_w + l$ and corresponding delay generated as in (2.9) and (2.10).

$$T'_{RCi} = (r + r_w/2)c_w + T_{RCi} \tag{2.9}$$

$$T'_{LCi} = (l + l_w/2)c_w + T_{LCi} \tag{2.10}$$

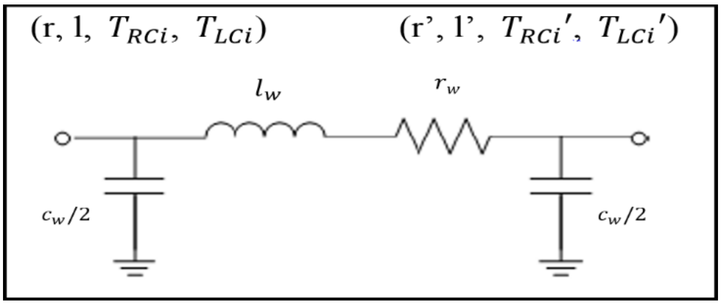
(r, 1, T_{RCi}, T_{LCi})
(r', 1', T_{RCi}', T_{LCi}')
l_w
r_w
$c_w/2$
$c_w/2$

Figure 2.5. Wire only model

(ii) Estimating the delay from source to sink with a buffer:

It's remarkably close to the original prototype. In Figure 2.6, it is seen that the tuple for the next vertex $(r', l',\ T_{RCi}',\ T_{LCi}')$ depends on the tuple for the previous vertex $(r, l,\ T_{RCi},\ T_{LCi})$, the buffer parameter $(c_b\ , d_b)$ and the current segment wire parameters $(c_\omega,\ r_\omega,\ l_\omega)$, where $r' = r_b,\ l' = 0$. And the corresponding delay are formulated as in (2.11) and (2.12).

$$T'_{RCi} = r(c_w+c_b) + r_w(\tfrac{c_w}{2}+c_b) + d_b + T_{RCi} \tag{2.11}$$

$$T'_{LCi} = l(c_w+c_b) + l_w(c_w/2+c_b) + T_{LCi} \tag{2.12}$$

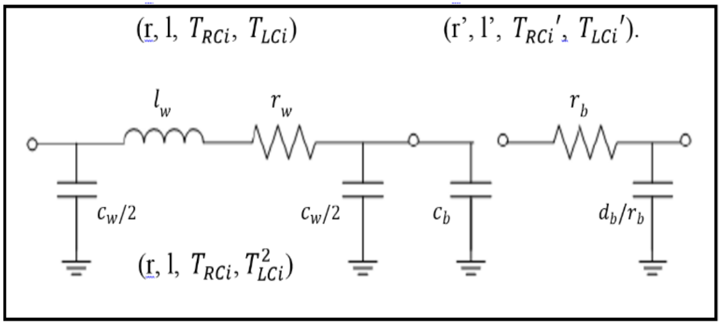
(r, l, T_{RCi}, T_{LCi})
(r', l', T_{RCi}', T_{LCi}').
l_w
r_w
r_b
$c_w/2$
$c_w/2$
c_b
d_b/r_b
(r, l, T_{RCi}, T_{LCi}^2)

Figure 2.6. Buffer-terminated model

2.1.3.4. Minimum Spanning Tree

A spanning tree is a set of edges in a graph that avoids creating any cycles while still linking all of the vertices. There can be many spanning trees for the same graph. The minimum spanning tree (MST) or minimum weight spanning tree is the spanning tree with a weight that is smaller than the weight of every other spanning tree in the graph, where the weight may be the distance between two neighbouring points that are united by the edge. A multi-super-tidal structure is depicted in Figure 2.7.

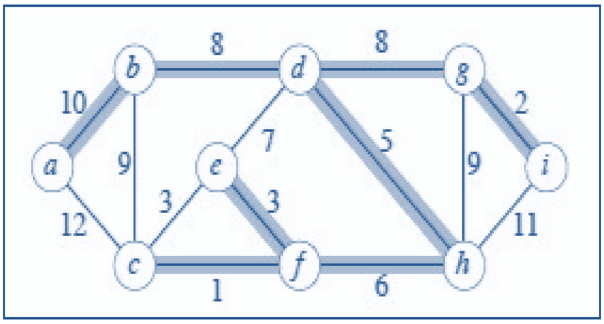

Figure 2.7. Weighted graph of MST. Thick edges represent edges of the tree

A Czech physicist named Otakar Boruvka developed the first technique in 1926 [2.24] to find the shortest path between two nodes in a graph. Prim's algorithm and Kruskal's algorithm are now the most popular methods for calculating MST. The associated decision problems, such as checking if an edge is in the MST or if the minimum total weight exceeds a threshold, can be solved in polynomial time using these greedy algorithms. Reverse-delete algorithm is another greedy algorithm; it is the inverse of Kruskal's algorithm but is not commonly utilised. Due to the linear relationship between wirelength and wiring area produced by VLSI minimum-spacing design guidelines, it is necessary to optimise the total interconnect length of the circuit interconnections in order to lower the overall area of the chip [2.25]. A set, S, of n pins that serve as signal network terminals for interconnection. The optimal solution is a MST over S is $(MST\ (S))$ [2.25] if all the wires point-to-point without any intermediary junctions other than the points specified are linked, as shown in Figure 2.8. Here, the branches of the tree are joined in a straight line. In this case, the endpoints of the net are represented by hollow circles.

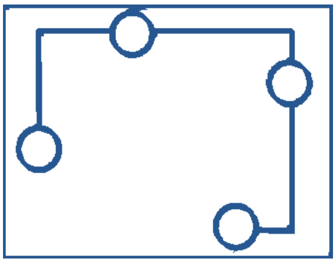

Figure 2.8. A MST structure connected in rectilinear manner

2.1.3.5. Rectilinear Steiner Minimum Tree

Rectilinear distance is used in place of Euclidean distance in the RSMT, also known as the Minimum Rectilinear Steiner Tree (MRST). It can be characterized as follows: For given a set S of n points, it is required to find a set ST of Steiner points that minimises the MST cost over all possible connections.

The MST cost is calculated by (2.1). This is because in VLSI circuits during physical design automation, the nets are connected by means of wires which run only in rectilinear direction i.e., normal to each other. Hence, the total length of the wire is the sum of the lengths of the vertical and horizontal segments [2.2]. A global router's job is to determine the steiner tree (a tree with the original set of points S and the steiner point set ST) of the cheapest chip that connects the necessary nodes on [2.26]. The MRST for these four sites is shown in Figure 2.9. Hollow circles indicate the original or terminal point set P, and solid dots indicate steiner points.

As per Hwang's theorem [2.27] for a set S of n points, the cost of an MST will be greater than that of an MRST. So, it is clear that for a set of points, MRST will give shorter wirelength than MST as shown in Figure 2.10. Thus, the total wirelength is determined from the RSMT or MRST structure which is formulated with the addition of some steiner points. Determination of this steiner points is done by routing optimization algorithms during physical design automation. Wire length cost is generated using (2.1) as wirelength MST or RSMT as in (2.13).

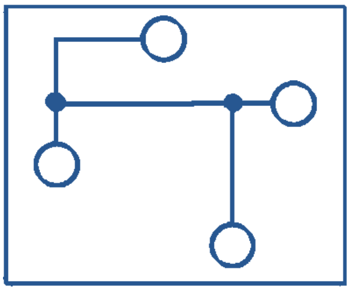

Figure 2.9. An RSMT for a set of four points

$$Cost = |x_1 - x_2| + |y_1 - y_2| \tag{2.13}$$

The relationship between MST and MRST can be stated by Hwang's theorem as follows:

$$\frac{\text{Wirelength (MST)}}{\text{Wirelength (MRST)}} = \frac{3}{2} \tag{2.14}$$

A strong motivation is given by (2.14) for constructing an MRST by an MST-based approximation algorithm.

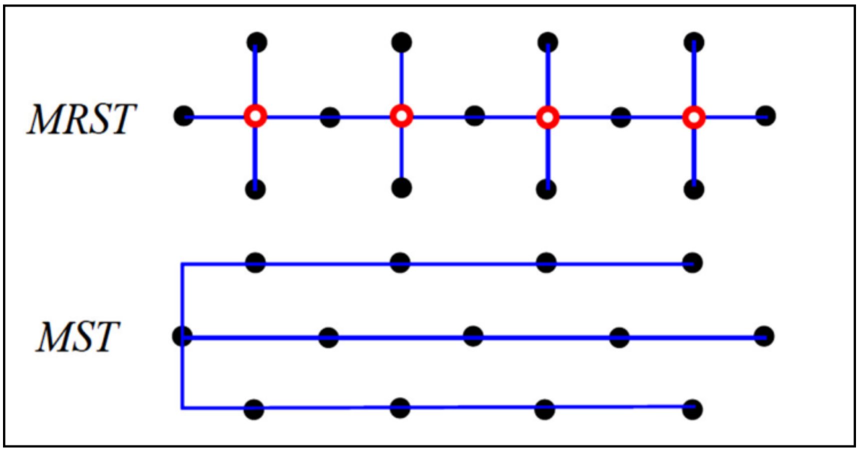

Figure 2.10. Comparison of wirelength between MRST and MST

2.2. Wireless Sensor Network: Overview

The sensor nodes in a wireless sensor network (WSN) are extremely many and dispersed over the network; these nodes are autonomous and very small. Motes is the common name for them. With its limited processing and handling capacities, a WSN typically consists of a large number of geographically dispersed, unobtrusive, battery-powered, embedded devices that are arranged to steadily gather, process, and pass information on to the clients. Motes are the network's tiny personal computers. [2.28]-[2.29]. Sensing, processing, communication, and power units are all included in wireless sensor nodes. Each node may independently collect data, sense its surroundings, process that data, and communicate with other nodes. The neighbouring sensor nodes collect and share data based on input from the sensing unit, there after the processing unit's computation of the restricted permutations of the detected data, and followed by the communication unit's transmission of the processed data is the primary work of a WSN structure.

2.2.1. Components of WSN

The sensor node, relay node, actor node, cluster head, gateway, and base station are the WSN system's components as shown in Figure 2.11.

Sensor node: Node that serves as a sensor and is capable of collecting, processing, and transmitting data.

Rely node: This node in the middle is a relay that links to the node to its left. In order to make networks more reliable, it is implemented. Unlike most other field devices, a depend node does not have any sensors or controls for the 7 processes; therefore, it is unable to exchange data with it. A unique single node CPU can run at speeds of up to 8 MHz with 8 KB of RAM, 128 KB of flash memory, and preferably 916 MHz of radio frequency. An actor is a specialised node that makes a choice according to the application's requirements. These nodes often include greater computing power, faster data transfer rates, and longer battery life than their lower-end counterparts. With 16 KB of RAM, 128 KB of flash memory, and ideally 916 MHz of radio frequency, a unique actor node has a throughput of about 8 MHz.

High-bandwidth sensing nodes perform the data aggregation and clustering duties of WSN cluster heads. Several applications and infrastructure needs will necessitate a variety of cluster heads. A separate cluster head processor operates at 2.4

GHz if possible and features 512 KB of RAM, 4 MB of flash memory, and a clock speed of 4-8 MHz. Each of the other nodes in the sensor network relies on this one as though it were rock solid, risk-free, and reliable.

Gateway: The role of a gateway is to facilitate communication between sensor networks and other networks. In comparison to the sensor node and the cluster head, the gateway node has superior programme and data memory, CPU power, transceiver range, and memory expansion options thanks to external storage. Typical specifications for a gateway processor include 16 MHz of processing speed, 512 KB of random-access memory (RAM), 32 MB of flash memory, and a radio frequency of 2.4 GHz.

Base station: This exceptional class of nodes has strong processing power and computational energy.

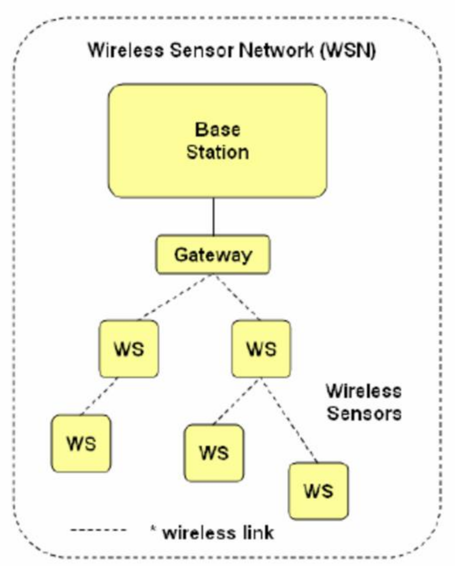

Figure 2.11. Components of WSN [2.36]

2.2.2. WSN Routing Protocols

The routing protocol is a method that determines the best route for sending data from its origin to its destination. While the route must be selected based on factors such as network type, channel conditions, and performance metrics, the process is fraught with potential pitfalls [2.30]. Data collected by sensor nodes in a WSN is typically transmitted to the base station that connects the WSN to other networks (like the internet), where it is processed and action is taken based on the results. Due to the proximity of the base station and the motes (sensor nodes), single-hop communication is achievable in very small sensor networks. Several sensor nodes are positioned too far from the sink node for single-hop communication to be practical in most WSN applications, where the coverage area is substantially bigger and thousands of nodes must be put (gateway). Hence, they are unable to establish a direct connection with the base station. In contrast to indirect communication, which involves relaying messages between multiple nodes, direct communication involves only one relay node. The sensor nodes in a multi-hop communication network not only generate and disseminate content, but also provide a connection for other sensor nodes to reach the base station. The primary responsibility of the network layer is routing, or the selection of a path between two nodes [2.31].

2.2.2.1. WSN routing protocol classification

The routing protocols define the channels through which data will be transmitted from one node to another. The WSN routing protocols can be categorized in a variety of ways [2.32] - [2.33]. Figure 2.12. depicts the fundamental classification of routing protocols.

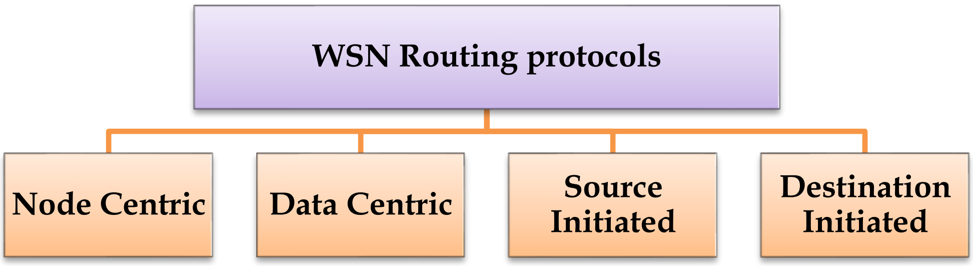

Figure 2.12. Classification of WSN Routing protocols

Some classification of routing protocols:

Node centric: Since wireless sensor networks are not designed for this type of communication, node centric protocols require the target node to be identified by a set of numeric identifiers. Low Energy Adaptive Clustering Hierarchy (LEACH), for instance.

Low energy adaptive clustering hierarchy: The LEACH [2.34] routing protocol is used to build up the cluster so that all of the sensor nodes in the network receive the same amount of power. The LEACH protocol organises a network of sensors into various clusters. A cluster's routing information is stored on a single node, which also acts as the cluster's head. The LEACH protocol employs a random selection process to determine the cluster leader from among the participating nodes. Because the battery of any one node is not overworked by being made the cluster head permanently, this protocol allows for greater longevity.

Data-centric: In most wireless sensor networks, the information or data detected is far more crucial than the node itself. Hence, the primary focus of data centric routing techniques is not data collection from specific nodes, but rather the transmission of information as indicated by specific qualities. Since the sink node in data-centric routing issues queries to certain locations in order to collect data of specific quality, it is necessary to utilise an attribute-based naming scheme in order to identify these characteristics in the data.

Sensor Protocol for Information via Negotiation: Sensor Protocol for Information via Negotiation (SPIN) protocol is intended to be used in place of existing protocols to address deficiencies like gossiping and overflow [2.35]. The fundamental tenet is that sharing data that a node senses may need more resources than sharing meta-data, which is merely a description of the perceived data. Each node's resource management keeps an eye on its resources and modifies their functionality as necessary. Three different types of messages (ADV, REQ, and DATA) are used in SPIN. This node sent out an ADV packet to notify all the other nodes in the network that it was now able to share information. This ADV message from an advertising node contains information about the data it conveys. Individual nodes that are interested in the information that the advertising node is looking for can send it a REQ message. After receiving the REQ message, the advertising node will transfer data to the requested node. This process happens when the node generates and sends an ADV message after receiving data.

Destination initiated: Nodes who have a stake in the information that the advertising node is looking for can send it a REQ message. The REQ message is received, and the advertising node then relays its message to the requested recipient. As a node receives information, it generates and sends an ADV message.

Directed diffusion: Nodes who have a stake in the information that the advertising node is looking for can send it a REQ message. The REQ message is received, and the advertising node then relays its message to the requested recipient. Following the receipt of data, the node produces and transmits an ADV message.

Source initiated: In these protocols, the route is built from the generator side to the endpoint when a source node announces that it has data to share. SPIN is one of these methods.

2.2.2.2. Categories of WSN routing protocols

There are two methods that are utilized to transfer data in sensor networks. The first is known as flooding, while the second is gossiping protocol. It is not necessary to maintain the topology or utilize any routing algorithms. When a sensor node receives a packet of information, the flooding protocol causes that information to be sent to all of the nodes in the immediate vicinity. The broadcasting activity will continue until the packet has successfully reached its destination and the packet must have travelled more than hops [2.36].

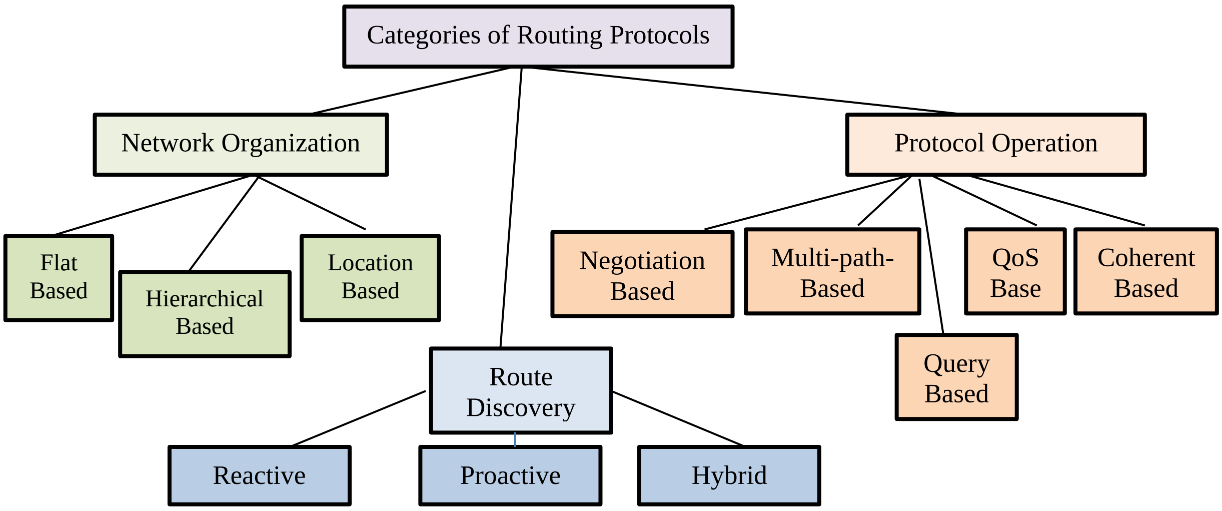

Figure 2.13. Categories of WSN Routing protocols

Flooding has several benefits, but its main benefits are simplicity and convenience of use. The negatives include implosion, redundancy, and resource myopia. The gossiping protocol is a slightly more sophisticated variation of the flooding technique. The receiving sensor node in a gossiping protocol will forward the data packet to an arbitrary neighbour. The sensing nodes pick another node at random and relay data to it on the following turn. This procedure is repeated repeatedly. In contrast to floods, the gossiping protocol does not use broadcasting. Implosion problems can be readily averted in this method. However, this increases delay. Figure 2.13 shows the major categories of routing protocols.

2.2.2.3. Routing protocols focused on route discovery

2.2.2.3.1. Reactive protocols

Reactive routing protocols are only activated when a node needs to communicate data to another node; they do not preserve the entire network topology. Thus, when queries are launched, routes are generated as needed. The following are the reactive routing protocols that are most frequently used:

On-Demand Ad-hoc Distance Vector: After receiving a request, the Ad-hoc On-demand Distance Vector (AODV) protocol will calculate the best route to take. AODV is designed for mobile networks without infrastructure. For the construction of routes among network nodes, it uses the on-demand routing mechanism. When a source node wants to direct data packets, a path is built by itself, and the path is maintained for however long the source node requires. This is why the term "on demand" is used to describe it. AODV routing supports Unicast, Multicast, and Broadcast communication. The wireless ad-hoc network's AODV routing technology distributes packets among mobile nodes. When a mobile node needs to send a packet to a node that doesn't have an open connection, AODV can help it do so by using a neighbouring node. Routing table content is periodically exchanged between neighbouring nodes and set up for unforeseen modifications [2.37]. In order to send data, AODV finds the shortest route that avoids loops in the network's routing table. If the planned route has inaccuracies or deviations, the AODV knows how to calculate a new one so that all messages can be sent without interruption.

Dynamic Source Routing: Wireless sensor networks employ the Dynamic Source Routing (DSR) protocol, which was created in 1996. Reactive or on-demand dynamic source routing are also possible. It uses source routing rather than routing tables, as its name implies. Route maintenance and route discovery are the two components of routing in DSR. The initiating node will initiate a phase of route discovery by exchanging route request and Route Reply (RREP) messages. With DSR, only the final destination node sends an RREP message to the sending host, but in AODV, all intermediate nodes participate in the RREP exchange. The goal of the next stage route maintenance is to prevent RREP message flooding and is used to shorten the distance between nodes [2.38] - [2.39].

2.2.2.3.2. Proactive Protocols

Table-driven routing protocols are also commonly used because they are responsible for keeping track of the network's overall routing tables as packets are forwarded from node to node. Even when there is no traffic, the predetermined paths must be followed. The most popular algorithm is as follows:

Optimized Link State Routing: Proactive routing technology called as Optimal Link State Routing (OLSR) employs a table-focused approach. The excessive overhead in OLSR is its primary drawback. To make up for this lag, multipoint relays (MPRs) are employed to deal with the massive overhead they cause. Each node uses three adjutant nodes called MPRs to transmit data. Since each node delivers control information alternately, persistent control information is not necessary [2.38] - [2.39].

2.2.2.3.3. Hybrid Routing Protocols

By ignoring their shortcomings, hybrid routing methods combine the advantages of proactive and reactive routing strategies.

2.2.2.4. Protocols for operation-based routing

In order to improve network speed, load balancing, and delay, multi-path routing systems provide many paths. The multiple routing system provides a backup route in case the primary one fails. There is greater value in a network with multiple paths than in a dense one. As periodic messages must be delivered to maintain the paths at regular intervals, using multiple path routing is not more energy efficient. These are the multipath routing protocols: [2.38]

- Multiple paths and speeds (MMSPEED)
- Sensor protocols that exchange information (SPIN)

2.2.2.4.1. Query-based routing protocol

Receiver-initiated routing protocols predominate. Sensor nodes will only respond to destination node questions. As the destination node requests information via the network, the target node recognises it and sends it back to the initial requester: [2.38]

- Directed diffusion (DD)
- SPIN
- COUGAR

2.2.2.4.2. Protocols for routing based on negotiations

To reduce unnecessary data duplication, these protocols have sensor nodes communicate with one another and share data about their local resources with their neighbours. Thereafter, decisions are made about what data should be transmitted based on the outcomes of negotiations, which include: [2 .38]

- Sequential assignment routing (SAR)
- Sensor protocols for information via negotiation (SPAN)
- Directed Diffusion (DD)

2.2.2.4.3. QoS based routing protocols

Better service quality can be achieved by employing these protocols. QoS-aware protocols seek to optimise throughput, data delivery, energy consumption, and latency while maintaining the quality metrics often associated with high-quality QoS from source to destination. To cite a few examples: [2.40.]

- Sequential assignment routing (SAR)
- Multi path and Multi SPEED (MMSPEED)
- Speed

2.2.2.4.4. Coherent data processing routing protocol

The nodes in the coherent data processing routing protocol do minimal processing on the data before sending it on to other sensor nodes or aggregators (time stamping, data compression, etc.). The aggregator receives data from multiple source nodes, combines it, and then sends the resultant data to the sink node.

2.2.2.5. Routing challenges in WSNs

Designing routing protocols for WSNs is challenging because they are different from wireless infrastructure-less networks in a number of important aspects. There are various kinds of routing difficulties in wireless sensor networks. The following are a few significant difficulties:

- Allocating a universal IDs scheme for a large number of sensor nodes is almost impossible. Therefore, wireless sensor motes are not capable of utilising traditional IP-based protocols. It is necessary for detected data to flow from a variety of sources to a particular base station. However, conventional communication networks do not experience this.
- The majority of the time, the created data traffic has a lot of redundancy. Because several sensing nodes can produce identical data at the same time. Therefore, it is crucial to take advantage of this redundancy using the routing protocols and to make the best use of the bandwidth and energy that are available.
- In addition, wireless motes are severely constrained in terms of transmission power, bandwidth, storage capacity, and on-board power. A variety of new routing techniques have been proposed as a result of these differences in order to address the routing issues in wireless sensor networks.

2.2.3. Design challenges in WSNs

The lack of available resources, such as power, data transfer capacity, and storage space, presents substantial challenges to the design of wireless sensor networks. A network engineer should satisfy the following requirements while creating new routing protocols.

2.2.3.1. Energy efficiency

Most wireless sensor networks run on batteries. In these sensor networks, energy constraint is a significant problem, particularly in hostile conditions like a battlefield. When battery levels drop below a certain battery threshold level, sensor node performance suffers. When creating sensor networks, energy is a major challenge for designers. In wireless sensor networks, the number of motes is practically infinite. The network's unreliable power source means that node batteries must be carefully managed. Because of this, the routing protocol must minimise power consumption. [2.41].

2.2.3.2. Complexity

The intricacy of a routing protocol can have an effect on the efficiency of the whole wireless network. This is because there is a lack of personnel with hardware competence, and because there are strict energy constraints on wireless sensor networks.

2.2.3.3. Scalability

These days, hundreds or even thousands of sensors can be supported by a single wireless sensor network due to the steadily decreasing cost of sensors. Because of this, it is essential that the routing protocol supports growth in the network. Routing protocol shouldn’t interfere with this if additional nodes are ever added to the network.

2.2.3.4. Delay

Some applications, like temperature sensors or alarm monitoring, demand an immediate response or one without a significant delay. Therefore, the routing protocol ought to have a low delay. In the aforementioned WSN applications, it is necessary to communicate the sensed data in the shortest amount of time possible.

2.2.3.5. Robustness

Wi-fi enabled sensor networks are commonly employed in critical and potentially dangerous settings. Occasionally, a sensor node in a wireless sensor network will die or disconnect. This means that the routing protocol must be resilient in a wide range of conditions, including those that are challenging and lossy. The operation of the routing protocol should also be perfect [2.42].

2.2.3.6. Data transmission

Depending on the use case, wireless sensor networks may send data in one of four ways: query-driven, event-driven, continuous type, or hybrid type. Until the sink initiates a query or an event occurs, nodes in the query-driven paradigm and the event-driven model do not begin sending data. The information is broadcast in a continuous stream at regular intervals. The efficiency of the routing protocol is affected by both the scope of the network and the nature of the transmission medium. Thus, a high-quality transmission medium immediately enhances the performance of a network [2.37].

2.2.3.7. Sensor location

Correctly locating the sensor nodes is a significant difficulty for designers of wireless sensor networks. The majority of routing systems use a localization approach to learn about their locations. There are some situations where GPS receivers are employed.

2.2.4. Clustering in WSNs

In WSN, routing is a crucial operation to take into account. It may be essential for one sensor node to use another sensor node in order to forward a packet to its destination, which is often the base station, due to each node’s restricted transmission range (BS). Due to energy and transmission range constraints, determining and maintaining routes in WSNs is a challenging issue. Routing protocols suggested in the literature for WSNs use well-known routing principles, including clustering, to reduce energy usage. The goal of WSN clustering protocols is to organise sensor nodes into clusters and select a Cluster Head (CH) for each cluster. The CHs can combine the data sent from the cluster members and deliver them straight to the BS in order to create an energy-efficient WSN. A clustering protocol primarily consists of two layers. While the second layer is in charge of sending the data to the BS, the first layer is utilised to choose the best possible set of CHs. The generalised perspective of WSNs is seen in Figure 2.14 and includes a BS, CHs, and sensor nodes (devices) placed throughout a region.

In a WSN, the clustering protocol should take into account the restrictions of the sensor nodes in addition to facilitating data transfer. Additionally, it should adhere to WSN standards for scalability, data delivery dependability, and energy efficiency.

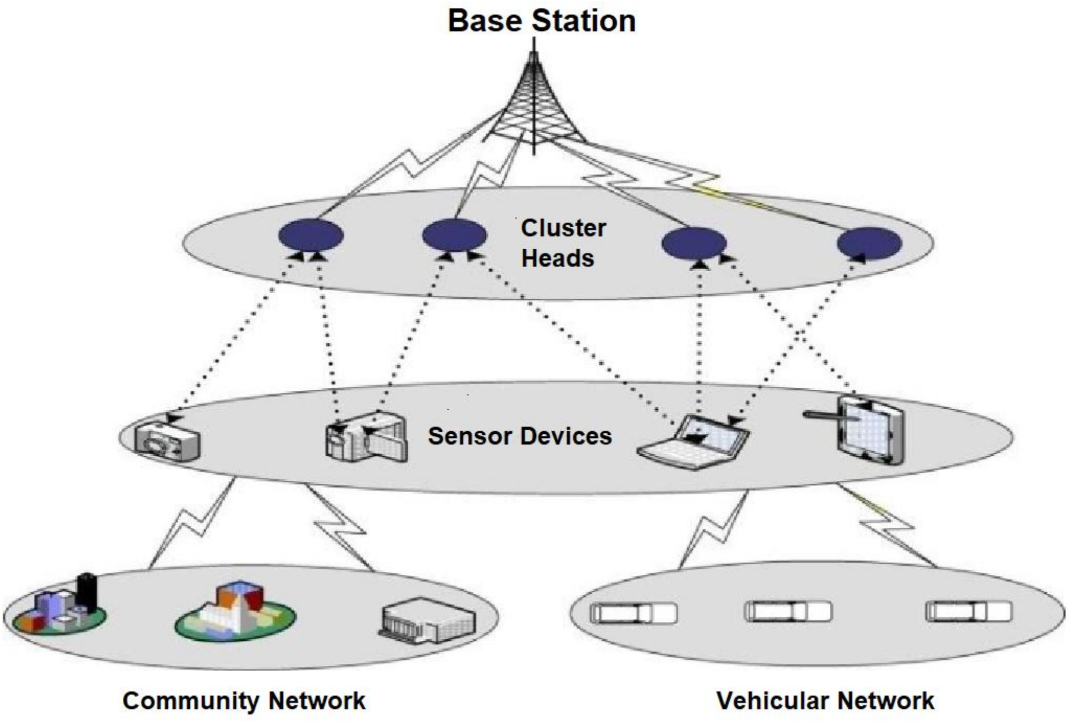

Figure 2.14. Structure of a Clustered WSN [2.43]

2.2.5. WSN & IoT

A physical quantity (such as ambient temperature, fan speed, etc.) can be remotely controlled through the internet depending on how many devices are connected to the internet. IoT is all that this is. The architecture of WSNs consists of a sink node and sensor nodes. Sensing the physical amount and transmitting the perceived data are the two tasks that the sensor node must complete. It must therefore do two tasks: data generation and data transmission. IoT operates at a more advanced level, integrating cloud computing, WSNs, the internet, and any physical thing that is connected to it. It is possible to treat WSNs as a subset of the WSNs depicted in Figure 2.15.

Different methods and tactics can be used to connect WSN applications and low-power sensing nodes to the Internet [2.45]. The most widely used integration techniques are cloud-based integration approaches [2.46] - [2.47], front-end proxy integration approaches [2.48] architecture frameworks [2.49] and integration using established Internet communication protocols [2.50] - [2.51].

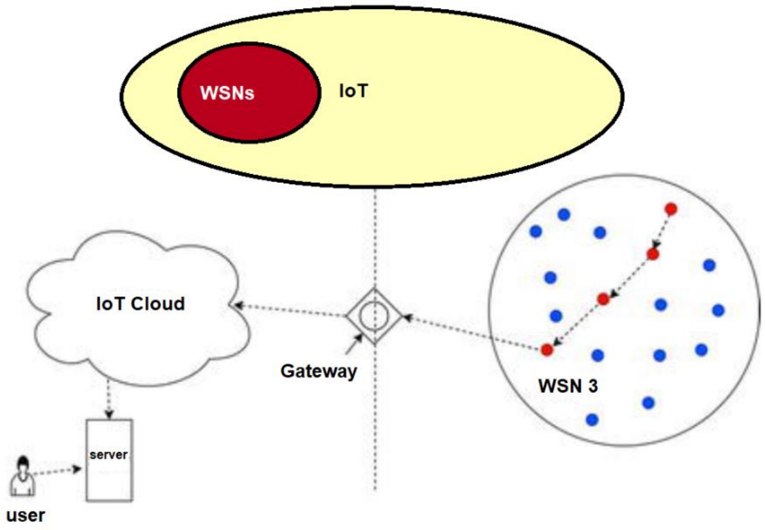

Figure 2.15. WSN and IoT inter relation [2.44]

2.2.6. WSN Application

WSNs are presently being used in a wide range of applications [2.52] including the military, home, and industrial. The following possible WSNs usage are briefly introduced.

System for Monitoring the Environment: Applications for WSN have usually included environmental monitoring heavily [2.53]. Systems for monitoring and controlling the environment include those that regulate temperature, humidity, light, and pressure. Applications for environmental monitoring in agriculture, habitats, indoor spaces, greenhouses, climate change, and forests have expanded quickly. Applications for environment monitoring are the subject of various studies [2.54]. Scalability, coverage, and energy efficiency are the three major criteria for environmental monitoring applications. The size of monitored locations might range from hundreds to thousands of hectares, hence the number of nodes distributed varies. Scalability is a crucial consideration when creating protocols to sustain a high number of nodes and provide complete coverage of the regulated region [2.55]. The procedures suggested in this work are appropriate for applications involving environment monitoring.

Human Body Tracking: The field of wireless healthcare systems has seen an increase in research attention in recent years. The concept of new wireless technology-driven human body monitoring was inspired by the ageing population, people who need ongoing health monitoring, and rising health care expenditures. A wide range of assisted living applications, including human physical and biological control and performance monitoring for health care, e-fitness, urgent detection, emotional recognition for social media, safety, and highly interactive games, have the potential to be made possible by Wireless Body Sensor Networks (WBSNs) [2.56]. Researchers have made several attempts to employ WBSNs for monitoring the human body. Using a network of wireless sensors that can be implanted into the bodily tissue or attached to the surface of the body, the human body is monitored. Small and clever medical sensors that can be carried or installed in the human body have been made possible by recent technological advancements. The data is collected by the sensors and sent to the centre for aggregation and analysis. Applications for health monitoring must be highly reliable because they deal with human lives. Another crucial need to guarantee the system's long-term operation is the network's energy efficiency [2.57].

Intelligent Structures: Building automation has lately modified WSNs to address the escalating cost of energy and the expanding green movement. Using smart sensor nodes, buildings may optimise their energy usage, enhance security and safety, and lower operating costs. In the literature, a number of WSN-based smart building management systems have been presented [2.58]. Different kinds of sensor nodes sensing variables including temperature, humidity, light, and suffocating smoke make up the WSNs utilised in smart building management systems. Actuators, gateways, servers, communication and application software on various levels, as well as other home appliances, may also be included in the systems [2.59]. To communicate across entire buildings, smart building management systems need to leverage multi-hop technology. To fulfil this requirement, specific data or hierarchical protocols can be utilised. Another crucial prerequisite for such systems is the energy efficiency of the network [2.60].

2.3. Metaheuristics and Swarm Intelligence

The fields of operations research and computer science have been working to find solutions to difficult, significant problems in the real world. Finding a workable solution and refining it to converge on the ideal global answer are both important for tackling a large-scale problem. Unfortunately, due to limited resources and the complexity of the majority of optimization issues, finding precise solutions is challenging. To fully profit from metaheuristic, it is essential to comprehend computational complexity. There are only a few algorithms that are known to converge to optimality in a reasonable amount of time according to computational complexity theory (polynomial time-algorithms). But many issues in the actual world are *NP*-hard (non-deterministic). In other words, if there is a solution to the issue it can be quickly proved. Is $P \neq NP$? The search for polynomial-time solutions to *NP*-complete issues is still ongoing. The most challenging *NP* issues are *NP*-complete ones. All *NP*-complete issues are p-time solvable if any *NP*-complete problem is *P*-time solvable.

2.3.1. Metaheuristics

A metaheuristic algorithm is a search method created to locate a suitable answer to an optimization issue that is complex and challenging to solve. In this real-world of scarce resources, it is crucial to develop a close to ideal solution based on faulty or insufficient knowledge (e.g., computational power and time) [2.61]. One of the most significant developments in operations research over the past 20 years has been the development of metaheuristics for resolving these optimization issues.

There are issues that demand consideration to create superior solutions to the already used conventional methods. In order to handle non-linear non-convex optimization problems, different metaheuristic methods are detailed by writers in very considerable detail. It is difficult to tackle specific *NP*-hard issues in combinatorial optimization (i.e., in reasonable run time). So, compared to optimization algorithms, iterative techniques, and straightforward greedy heuristics, metaheuristics can frequently produce effective solutions with less computational work. Using an optimization technique to achieve global optimality is impractical for a variety of problems. When stochastic random variables are incorporated in the objective or constraints, for instance, an optimization issue might get complicated. Thus, it is challenging to implement a comprehensive probabilistic programme utilising robust optimization methods or stochastic programming. Many fields can benefit greatly from the use of metaheuristics. Multi-objective functions with non-linear restrictions are the fundamental building blocks of many optimization problems. For instance, the majority of engineering optimization issues require solutions to multi-objective issues since they are highly non-linear. Metaheuristics are generally more effective than optimization algorithms, repetitive techniques, or basic heuristics at finding good solutions in combinatorial optimization because they search across a much larger range of feasible alternatives. [2.62] Metaheuristics are crucial in addressing real-world issues that are challenging to resolve with traditional optimization techniques. The majority of metaheuristics share the following characteristics: [2.62]

- Strategies that direct the search process are known as metaheuristics.
- Finding solutions that are close to ideal requires effective search space exploration.
- Metaheuristic algorithms use a variety of methods, from straightforward local search techniques to intricate learning procedures.
- Metaheuristic algorithms are typically non-deterministic and approximate.
- Metaheuristics are general solutions to problems.

Based on how they operate in the search space, metaheuristic algorithms are categorised [2.62] into categories including nature-inspired versus non-nature-inspired, population-based versus single point search, dynamic versus static objective methods, one versus different neighbourhood structures, and memory consumption against memory-less approaches. This essay does not intend to contrast search and optimization methods. Therefore, it is crucial to consider if traditional search techniques satisfy robustness criteria. In order to utilise and explore the solution space, metaheuristics are appropriate.

2.3.2. Swarm Intelligence

Swarm intelligence (SI) is simply the aggregate conduct of distributed, self-organized structures, natural or simulated. The idea of Swarm Intelligence fundamentally functions on artificial intelligence. Swarm Intelligence was first presented with regards to cellular robotic systems [2.63].

Swarm intelligence is the developing collaborative intellect of bunch of simple independent agents. Here, an autonomous agent is a subsystem that associates with its current circumstance, which likely comprises of many different agents, yet it acts somewhat freely from any remaining agents. The autonomous agent doesn't obey orders from a pioneer, or some worldwide arrangement.

Swarm Intelligence structures are generally comprised of a population of collaborating locally with each other and with their atmosphere. The basic specialists keep extremely basic guidelines, and in spite of the fact that there is no centralized control structure directing how individual specialists ought to act, neighbourhood, and to some extent arbitrary, connections between such specialists lead to the development of "intelligent" worldwide conduct, obscure to the singular specialists.

Swarm conduct should be visible in bird flocks, fish schools, microbial intelligence, just as in bugs like ant and bees [2.64]. Several optimization algorithms are based on swarm intelligence are observed and formed by researchers and are employed in several domains. Some of the robust metaheuristics and swarm-based optimization concept and techniques used in this literature are given below.

2.3.3. Particle Swarm Optimization

The first widespread theory of particle swarm optimization was by Eberhart and Kennedy in 1995 [2.65]. They devised a way to organize the movement of particles in a search area that investigated potential solutions. Their perspective was built on the ideas of 'colony' and 'evolution' as well as fundamental ways stimulated by the duplication of hunting birds. The idea of this procedure was that when a bird school seeks food in an unfamiliar area, each bird's moving path is inspired by both the global optimist knowledge of a place and the knowledge of the place of the individual [2.66] - [2.67].

PSO is a multi-specialist comparable pursuit strategy that connects with joins an iterative technique to acquire the best arrangement in a complex search space. The PSO model is built on a swarm of n particles, each of which searches for a better solution to a problem in D dimensions by updating its position in light of its own and the group's past explorations.

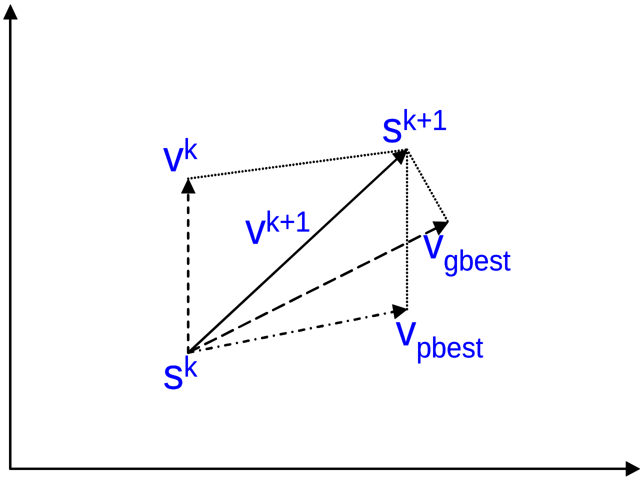

Figure 2.16. Concept of modification of a searching point by PSO

Where S_k, S_{k+1} are the current searching point and modified searching point. V_k, V_{k+1} are the current velocity and modified velocity. V_{pbest} is velocity based on particle’s best and $\mathrm{V_{gbest}}$ is velocity based on global best. A set of location or searching point and velocity vectors, $X_i = \{X_1, X_2, X_3, \dots \dots, X_n\}$ and $V_i = V_1, V_2, V_3, \dots, V_n\}$ are provided to each agent beforehand. Every so often, the criteria in (2.15) and (2.16), which are typical of PSO [2.68], are updated to produce new vectors.

$$V_{i,t+1} = V_{i,t} + c_1 r_1 (p_{best} - X_{i,t}) + c_2 r_2 (g_{best} - X_{i,t}) \tag{2.15}$$

$$X_{i,t+1} = X_{i,t} + V_{i,t+i} \tag{2.16}$$

Thus, the effects of individual and group information for a typical particle are controlled by the constants c_1 and c_2 respectively. Random integers with uniform distributions [2.69] are used for the r_1 and r_2 variables. Every particle is assigned a new position at random, and the fitness value is continually boosted by the $pbest$ value (the best position value of the individual) and the $gbest$ value (the best position value of the entire swarm) in the hopes of achieving the best possible outcome. At [2.69] - [2.71], c_1 and c_2 begin to affect the data of each individual particle and the data of the group as a whole. The ranges of r_1 and r_2 are arbitrarily set to [0, 1]. [2.72 - 2.73]

Particle momentum was ranked using the inertia weight (w), which was adapted [2.74] through the development of PSO-W as a replacement for V_{max}. In order for the algorithm to assemble more effectively, it is proposed to establish some sort of hierarchy in the swarm's exploratory and exploitation abilities. For this reason, we must rewrite (2.15) as (2.17).

$$V_{i,t+1} = wV_{i,t} + c_1r_1\left(p_{best} - X_{i,t}\right) + c_2r_2\left(g_{best} - X_{i,t}\right) \tag{2.17}$$

Traditionally, the size of the search space is attributed to the inertia weight w. It is essential to have a large value of w for composite high dimensional issue spaces, but a small value of w is sufficient for low dimensions search spaces [2.75], [2.76]. If s is the population size, D is the Dimension size, and R is the relative worth of consistent results consistent with [0,1], (2.18) allows for a wide range of inertia weights to be achieved.

$$w = \left[3 - exp\left(-\frac{s}{200}\right) + \left(\frac{R}{8}D\right)^2\right]^{-1} \tag{2.18}$$

In order to improve its performance, the PSO algorithm is justified to use a new parameter, known as the constriction factor given in (2.20). This was first introduced by Clerc [2.77] and has shown to be quite useful in accommodating the exploration and exploitation trade-off, guaranteeing a fruitful algorithmic coincidence. With this new form of the equation, (2.15) is more accurately written as (2.19).

$$V_{i,t+1} = \chi[V_{i,t} + \Phi_1(p_{best} - X_{i,t}) + \Phi_2(g_{best} - X_{i,t}\ ) \tag{2.19}$$

$$\chi = \frac{2}{\left[2-\phi-\sqrt{\phi^2-4\phi}\right]} \tag{2.20}$$

Here, $\phi = \phi_1 + \phi_2$, $\phi_1 = c_1r_1$ and $\phi_2 = c_2r_2$. Typically smearing the value of ϕ as 4.1 [2.78]

2.3.4. Ant Colony Optimization

The primary illustration of a fruitful swarm intelligence model is Ant Colony Optimization (ACO) [2.79]. ACO draws motivation from the social conduct of ant colonies. It is a characteristic observation that a gathering of 'practically blind' insects can mutually work out to find the shortest course between their food and their home with no visual data. The ACO heuristic is influenced by the rummaging conduct of a genuine ant colony in tracking down the shortcut between their home and the food. This is accomplished by a stored and aggregated chemical substance called 'pheromone' by the proceeding ant which advances towards the food. In it looking through the ant, it utilizes its own insight into where the smell of the food comes from (as heuristic data) and the other ants' choice of the way toward the food (as pheromone data).

Ants typically forage randomly in the wild, then return home while leaving pheromone trails indicating where they've been after spotting potential sources of food. If several ants discover such a way (pheromone trail), it's likely that they won't just keep wandering aimlessly but instead will stick to the trail, coming back to add to it in the hopes that they will eventually find food by taking the quickest route. In any case, the pheromone starts to evaporate after a while. The longer it takes for an ant to travel down the trail and back, the longer the pheromone requires to evaporate (and the track to turn out to be less notable). As compared more ants will go to the smaller track (can be depicted as a constructive response loop) and accordingly the pheromone concentration stays high for a more extended time frame [2.80] - [2.81].

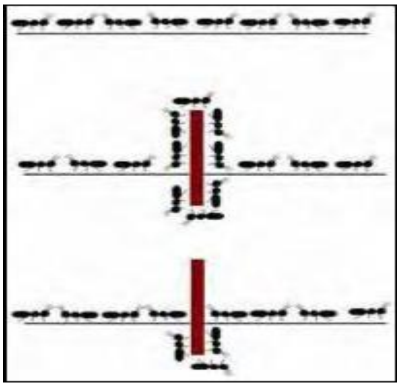

Figure 2.17. Ants finding the shortest path around an obstacle

ACO was first presented utilizing the Traveling Salesman Problem (TSP) [2.82]. Beginning from its start node, an ant repetitively proceeds starting with one node then onto the next. While being at a node, an ant decides to move to a node not visited at time t with a probability stated by (2.21).

$$p_{i,j}^{k}(t) = \frac{[\tau_{i,j}\,(t)]^{\alpha}[\eta_{i,j}\,(t)]^{\beta}}{\sum_{I \in N^{K_i}}[\tau_{i,j}\,(t)]^{\alpha}[\eta_{i,j}\,(t)]^{\beta}} \quad j \in \; N^{K_i} \tag{2.21}$$

Where, N^{k_i} is a practical neighbourhood of the *ant$_k$* , i.e., a collection of cities that have never been visited by *ant$_k$;* $\tau_{i,j}$ *(t)* the amount of pheromone in the edge *(i, j)* at that time *t*, the weight of the pheromone is *a ;* $\eta_{i,j}\,(t)$ is the first obtainable heuristic data on the edge *(i, j)* at that time *t*, β is the weight of heuristic data. Two parameters also decide the related effect of the pheromone trail and heuristic data. $\tau_{i,j}$ *(t)* controlled by

$$\tau_{i,j}(t) = \rho\tau_{i,j}(t-1) + \sum_{k=1}^{n} \Delta\tau^{K_{i,j}}(t) \;\; \forall \;(i,j) \tag{2.22}$$

where, ρ is the evaporation rate of the pheromone trail with $< 0 < \rho < 1$ and n is the no. of ant. $\Delta\tau^{K_{i,j}}(t)$ is the amount of pheromone emitted by *k$_{th}$* ant also given by

$$\Delta\tau^{K_{i,j}}(t) = \begin{cases} \frac{Q}{L_k(t)} & if\ edge\ (i,j)\ selected\ by\ ant_k \\ 0 & otherwise \end{cases} \tag{2.23}$$

where, Q is the pheromone updating constant and $L_k(t)$ is the kth ant tour cost of TSP.

2.3.5. Invasive Weed Optimization

The aggressiveness of weeds motivated the development of the numerical stochastic technique known as Invasive Weed Optimization (IWO) [2.83] - [2.87]. Because of its tenacious and annoying characteristics, weed spreads unintentionally and poses a serious threat to more desirable, cultivated flora. As a result, many people believe that "the weed always wins" when it comes to weeds [2.83]. The features imply that weeds adapt to their surroundings, changing their behaviour and becoming fitter. Seeding, development, and the competition within the weed colony are three key features of weed colonization. Like most of the algorithms, the IWO algorithm has been inspired from the colonization phenomenon of the weeds. It is comprised of the following stages:

Initialization: At the first stage, designated as N_0, a constant number of weeds are cultivated. Serving as the initial growth of the weed population, d being the problem dimension, the maximum seed usage is indicated by s_{max} , n is the index of nonlinear modulation. $iter_{max}$ is the maximum number of iterations that can happen. σ_{final} is the final standard deviation and $x_{initial}$ is the amount of space needed for the first solution.

Reproduction: Weeds can reproduce either sexually or asexually, depending on the type of plant. When the pollen fertilizes a seed made by the plant, it turns into an egg. This is called sexual reproduction. A member of the population can make seeds based on its own fitness and the colony's lowest and highest fitness. As shown in Figure 2.18, the number of seeds a plant can make increases linearly from the minimum to the maximum as the plant's fitness goes up.

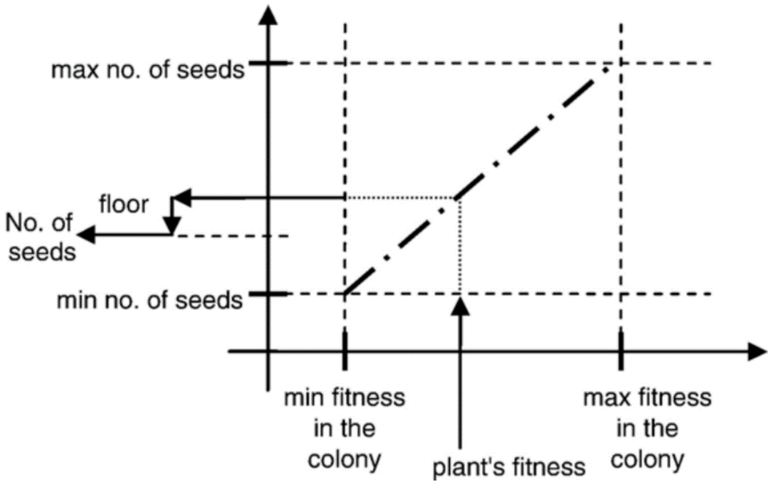

Figure 2.18. Seed production in weed colonization [2.83]

Individuals who are more physically fit are more likely to colonize than those who are less fit [2.83]. Even the less fit individuals often reproduce and add to their optimization. The approach mentioned below allows even the most inaccessible persons to contribute their useful share as given by (2.24)

$$S_{plant} = S_{min} + ceil\left[f_{plant} \times \frac{S_{max} - S_{min}}{f_{max} - f_{min}}\right] \tag{2.24}$$

Minimum and maximum values for the objective function during a given iteration are denoted by S_{min} and S_{max} , whereas f_{min} and f_{max} correspond to the minimum and maximum seed yields achievable during that iteration. The objective function value f_{plant} indicates how many seeds should be produced by a plant with a specific genotype S_{plant}.

Spatial dispersal: Spatial dispersion is the driving force behind the IWO algorithm's unpredictable and adaptable behaviour. Using a normal random distribution, as illustrated in (2.25), the generated seeds are scattered across the search space, with the parent weed's position as the mean and a fluctuating standard deviation (2.26). This procedure ensures that the seeds will not wander far from the mother plant.

$$Y = f\left(\frac{x}{\mu,\sigma}\right) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}} \tag{2.25}$$

However, according to (2.26) with every iteration the value of the standard deviation changes from $\sigma_{initial}$ to σ_{final}.

$$\sigma_{iter} = \left(\frac{iter_{max} - iter}{iter_{max}}\right)^n \left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \tag{2.26}$$

A decrease in from its beginning to its end value is shown in Figure. 2.19 for a range of values of n for non-linear modulation. As n grows larger, the rate of decline quickens.

Competitive Exclusion: After the weed population has reached its peak number of plants which includes seeds and unfit plants as well, a competitive exclusion is applied which allows all the weeds to create and spread the seeds in their specific search region. In order to reach the maximum permitted population in a colony, weeds with poorer fitness are eradicated. Eventually, only the most well-suited plants will survive once the procedure has been iterated as many times as possible.

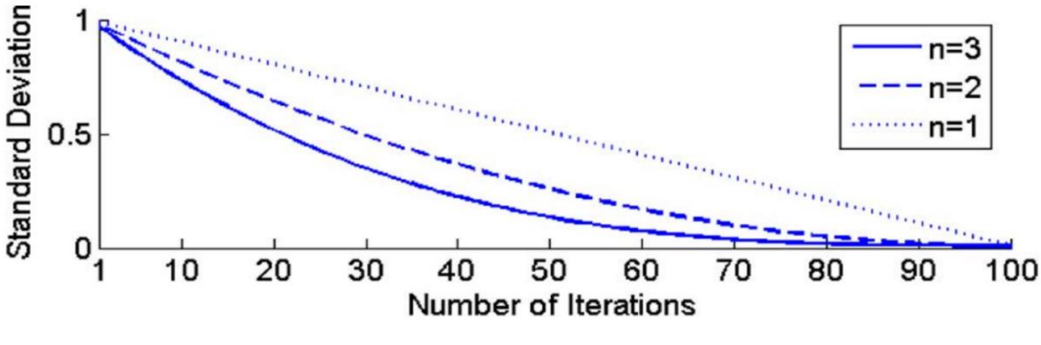

Figure 2.19. Standard deviation vs no. of iterations [2.83]

Termination Condition: The algorithm stops when iterating variable has reached its ceiling value.

2.3.6. Physarum Optimization

Physarum is composed of a series of protoplasmic veins that act as the transport system for the various nutrients as well as the chemical signals, during the vegetative state [2.88]. The study of this unicelled ameboid for 50 long years has paved the way for discovery that this flow is driven by protoplasmic flow. This flow is a derivative of the hydraulic pressure caused due to the synchronous beating of the actin-myosin fibers. The vein network has evolved to the following specifications [2.89]:

- When the food source is absent this network has a tendency to disappear
- The longer veins die out when multiple veins are connected to food sources.

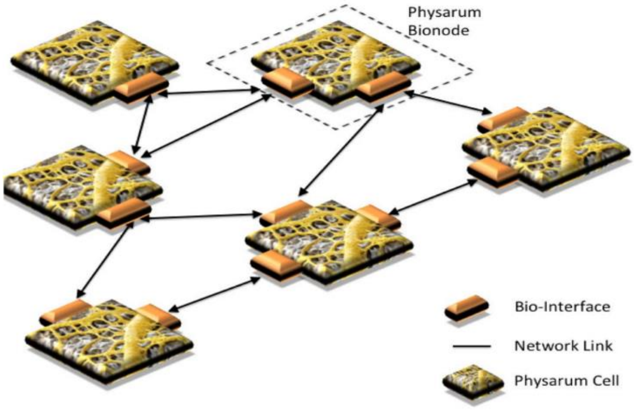

Figure 2.20. Physarum BioNetwork [2.90]

The cells in the ameboid physarum communicate through each other with the help of the protoplasmic vein network. This is the Physarum BioNetwork as shown in the above Figure 2.20. This network helps the bio interface to respond artificially to the changing levels of nutrients, food and stimuli. This reply is the rhythmic change of the actin-myosin fibre length. This changing in the fibre length is mapped and studied by sending impulses through the vein network. The nuclei in the physarum cells are connected to each other virtually via the vein network link. These connections form the bio interface. Each cell after receiving the stimuli transmits it to the other cell after conducting it through its internal vein network. This stimuli reception and transmission generate shortest path between the food source [2.91] thereby create an

MST. The procedure for this is shown in Figure 2.21. It is an example of a 4-node system with 3 external or steiner node.

To check the change in size of the actin-myocin fibre, feedback is artificially fed to the nodes. It is shown in Figure 2.21(a).

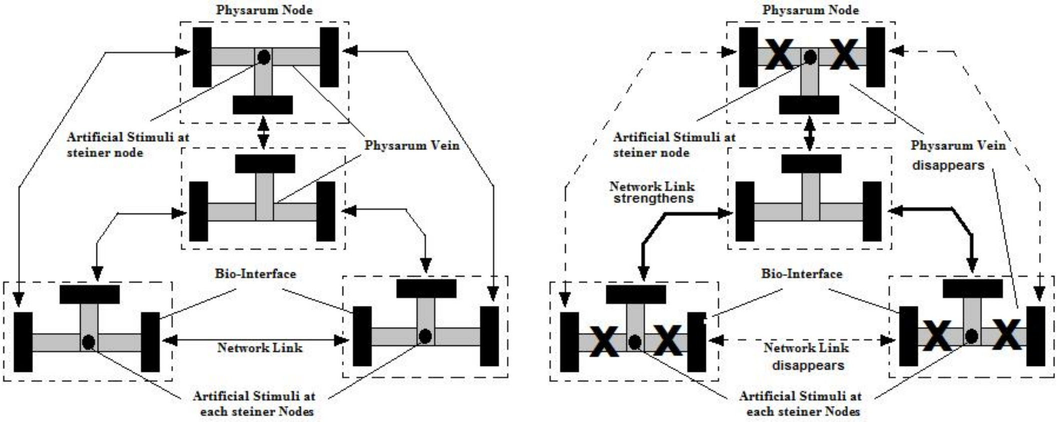
Physarum Node
Artificial Stimuli at steiner node
Physarum Vein
Bio-Interface
Network Link
Artificial Stimuli at each steiner Nodes
Physarum Node
Artificial Stimuli at steiner node
Network Link strengthens
Physarum Vein disappears
Bio-Interface
Network Link disappears
Artificial Stimuli at each steiner Nodes

Figure 2.21. (a) Four node Physarum BioNetwork (b) MST in Physarum BioNetwork

To study how the cells revise their physical nature when encountered with the protoplasmic flow, a flow is generated by forcefully through the vein network [2.92]. The viens are closely monitored as to which ones receive and transmit this flow resulting in the formation of the minimum steiner tree as shown in Figure 2.21(b). For a set of nodes v a pair of nodes v_i and v_j, is selected. Their connecting edge is e_{ij} and $l_{ij} = L(e_{ij})$ is the length of the edge. Let p_i and p_j be the pressure at v_i and v_j respectively. The protoplasmic flux is given by (2.27).

$$q_{ij} = \frac{d_{ij}}{l_{ij}}(p_i(t) - p_j(t)). \tag{2.27}$$

where, $d_{ij} = \frac{\pi r_{ij}^4}{8\xi}$ is a value of the conductivity of the tube, ξ is the fluid viscosity. A source of food v_s is chosen randomly along with another food source appointed as sink v_e, at each iteration. The presence of the gradient between the source and sink enables the protoplasmic flow, such that $\sum_j q_{ij} = I_0$ where I_0 is the positive flux entering the source node along with the sink v_e, whose respective flux is $\sum_j q_{ij} = -I_0$. The discussed theory is denoted as in (2.28).

$$\sum_j q_{ij} = \sum_j \frac{d_{ij}}{l_{ij}}(p_i - p_j\,) = \begin{cases} I_0, & if\ v_i = v_s \\ -I_0, & if\ v_i = v_e \\ 0, & otherwise \end{cases} \tag{2.28}$$

where, q_{ij} is protoplasmic flux, d_{ij} is conductivity of the tube. v_i & v_j are random nodes and v_s & v_e are source and sink.

The values of p_i and every q_{ij} can be obtained from the above equation on the basic control parameter of $p_a = 0$. The vein network spread over the physarum constricts to enable the protoplasmic flow. Synonymously in the chip the d_{ij} is being changed after every iteration. When this value reaches a certain threshold that is lower than a minimum it gets eliminated from the solution space along with the nodes that it connects. The associated points or the food sources from the remaining edges whose d_{ij} is more or equal to the given threshold form the minimum path.

2.3.7. Directed Artificial Bat Algorithm

The Directed Artificial Bat Algorithm (DABA) emphasizes on creating artificial agents that have the characteristics of bats. These bats are a part of swarms so they have to traverse the topography to find out and locate targets that are feasible [2.93] - [2.94]. And they do so by producing high-frequency ultrasonic sound waves. Evidently, the initial position of a bat represents the direction of waves and a solution. These initial amounts of every bat are set in the first iteration of the initiation step. Each bat thus looks for a better solution within a specific directional breadth. The present iterative bat searches locally for neighbourhood solutions. However, wave that is transmitted has a directed search scope, and it also has a better fitness value.

The DABA is a metaheuristic algorithm that is swarm intelligence-based. It makes use of ultrasonic waves for the purpose of echolocation, that is, for sensing distance and for flying randomly to search for a prey. The wave which is emitted have a wave speed given by the relation $V = f\,\lambda$, where frequency is f and wavelength is λ. Both the wavelength and frequency of the bat are dynamic in nature as according to the behaviour when a prey is in vicinity as given below.

- Wavelength value is increased and frequency is declined when no prey is in vicinity.
- Wavelength is declined and the frequency is increased to attack the prey accurately.

The search space consists of n bats, and their positions x_i and velocities v_i are being updated. Now, each bat's position $x_i = \left[x_1, x_2, \dots\dots x_{id,}\right]$ is determined and evaluated by a fitness function $f(x_i)$. Here the fitness of each particle is calculated with the help of frequency f_i which is multiplied with the directed bat's wavelength λ. The pulse increase factor r and the loudness A_0 manipulates the pulse frequency. The loudness is varied in the range of positive A_0 to A_{min} constant value. The right updating of the amplitude A_i and the pulse rate r_i balances the tendencies of exploration and exploitation of each. Once a bat finds a solution, the pulse emission rate rises with the decrease of amplitude level, in order to attain more accuracy. The updated velocities v_i and locations x_i are as follows:

$$F_i = f_{min} + (f_{max} - f_{min})\,\beta \tag{2.29}$$

$$V_{it} = v_{it} - 1 + (x_{it} - 1 - x^*)f_i \tag{2.30}$$

$$X_{it} = x_{it} - 1 + v_{it} \tag{2.31}$$

Here β = random vector that is drawn from uniform distribution [0, 1]. During the iteration, the pulse emission rates and the amplitude undergoes particular changes, and among n bats within the population, the current best solution x^* can be attained. The frequency range of f_{min} and f_{max} is 20kHz to 500kHz and wavelength range of λ_{min} and λ_{max} is from 0.7mm to 17mm. When the best solutions are achieved, a new solution for each bat is generated among them using random walk by the following in (2.32)

$$x_{new} = x_{old} + QA_t \tag{2.32}$$

where, A_t is the average loudness of the bats at time t and ϙ $\in [-1, 1]$ is a random number. Also,

$$A_{t+1\,i} = \alpha\, A_{t\,i} \tag{2.33}$$

$$r_{t+1\,i} = r_{0\,i}\,[1 - exp(-\gamma t)] \tag{2.34}$$

where α and γ are constants and $\alpha = \gamma = 0.9$, $A_{0\,i}$ to A_{max} is [1, 2] and $r \in [0,1]$.

References

2.1 S. M. Sait, and Habib Youssef, *VLSI Physical Design Automation: Theory and Practice*. World Scientific Publishing, 1999.

2.2 N. Sherwani, *Algorithms for VLSI Physical Design Automation*. Norwell, MA, US: Kluwer Academic Publishers, 1993.

2.3 N. A. Sherwani, S. Bhingarde, and A. Panyam, *Routing in the Third Dimension: From VLSI Chips to MCMs*. Piscataway, NJ, USA: Wiley-IEEE Press, 1995.

2.4 J. Lienig, I. L. Markov, and J. Hu, *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Springer Publishing Company, 2011.

2.5 Y. J. Chang, Y. T. Lee, J. R. Gao, P. C. Wu, and T. C. Wang, ″NTHU-route 2.0: a robust global router for modern designs," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions*, vol. 29, no. 12, pp. 1931-1944, November 2010.

2.6 I. M. Liu, T. L. Chou, A. Aziz, and D. F. Wong, "Zero-skew clock tree construction by simultaneous routing, wire sizing and buffer insertion," *in Proceedings of the 2000 international symposium on Physical design,* San Diego California, USA, 2000, pp. 33-38.

2.7 J. P. Cohoon, and P. L. Heck, "BEAVER: A computational-geometry-based tool for switchbox routing," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions,* vol. 7, no. 6, pp. 684-697, June 1988.

2.8 V. Granville, M. Krivanek, and J. P. Rasson, "Simulated annealing: a proof of convergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 652-656, June 1994.

2.9 "ISPD 2007 Global Routing Contest," Sigda, http://www.sigda.org/ispd2007 /contest.html (accessed Jul. 18, 2018)

2.10 M. Cho, K. Lu, K. Yuan, and D. Z. Pan, "Boxrouter 2.0: architecture and implementation of a hybrid and robust global router," *in Proceedings of*

IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, 2007, pp. 503–508.

2.11 M. D. Moffitt, "Maizerouter: engineering an effective global router," *in Proceedings of Asia and South Pacific Design Automation Conference*, Seoul, Korea, 2008, pp. 226–231.

2.12 M. M. Ozdal, and M. D. F. Wong, "Archer: a history-driven global routing algorithm," *in Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2007, pp. 488–495.

2.13 Y. Xu, "3D Global Router: A Study to Optimize Congestion, Wirelength and Via for Circuit Layout," Ph.D. dissertation, *Dept. Elect. and Comp. Engg.*, Iowa State University, 2012.

2.14 L. McMurchie, and C. Ebeling, "Pathfinder: a negotiation-based performance-driven router for FPGAs," *in Proceedings of International Symposium on Field-Programmable Gate Arrays*, Monterey, CA, 1995, pp. 111– 117.

2.15 J. R. Gao, P. C. Wu, and T. C. Wang, "A new global router for modern designs," *in Proceedings of Asia and South Pacific Design Automation Conference*, Seoul, Korea, 2008, pp. 232–237.

2.16 D. Roy, P. Ghosal, and S. P. Mohanty, "FuzzRoute: A Method for Thermally Efficient Congestion Free Global Routing in 3D ICs," *in Proceedings of* 2014 *IEEE Computer Society Annual Symposium on VLSI*, Tampa, FL, USA, 2014, pp. 71-76.

2.17 M. Pan, Y. Xu, Y. Zhang, and C. Chu, "FastRoute: an efficient and high-quality global router," *VLSI Design*, vol 2012, pp 14, Aug., 2012.

2.18 Y. J. Chang, Y. T. Lee, and T. C. Wang, "NTHU-Route 2.0: A fast and stable global router," *in Proceedings of ICCAD*, San Jose, CA, USA, 2008, pp. 338-343.

2.19 M. Hanan, "On Steiner's problem with rectilinear distance," *SIAM Journal on Applied Mathematics*, vol. 14, no. 2, pp. 255-265, March 1966.

2.20 W. C. Elmore, "The Transient Response of Damped Linear Networks," *Applied Physics*, vol. 19, no. 1, pp. 55 - 63, January 1948.

2.21 Y. I. Ismail, E.G. Friedman, and J. L. Neves, "Equivalent Elmore delay for RLC trees," *in Proceedings of the 36th Design Automation Conference (DAC'99)*, New Orleans, LA, USA, 1999. pp.715-720.

2.22 Z. M. Yusof, M. K. Hani, N. S. Husin, and M. N. Marsono, "Iterative RLC Models for Interconnect Delay Optimization in VLSI Routing Algorithm," *in Proceedings of 2008 Student Conference on Research and Development (SCOReD 2008)*, Johor, Malaysia, 2008, pp 270.1-4.

2.23 N. S. Husin, M. K. Hani, "Optimal routing algorithm for minimizing interconnect delay in VLSI layout design," *in Proceedings of International Conference on Robotics, Vision, Information and Signal Processing (ROVISP),* 2007, Penang, pp. 345–349.

2.24 N. Jaroslav, E. Milková, and H. Nešetřilová, "Otakar Borůvka on minimum spanning tree problem translation of both the 1926 papers, comments, history," *Discrete Mathematics*, vol. 233, no. 1, pp. 3-36, April 2001.

2.25 G. Robins, and A. Zelikovsky, "*Minimum steiner tree construction: The Handbook of Algorithms for VLSI Phys. Design Automation*," Auerbach Publications, pp. 487-508, 2009.

2.26 F. K. Hwang, D. S. Richards, and P. Winter, "*Rectilinear steiner problem in The Steiner tree problem",* North Holland: Elsevier Science Publishers B.V. ,1992.

2.27 F. K. Hwang, "On Steiner minimal trees with rectilinear distance," *SIAM journal on Applied Mathematics*, vol. 30, no. 1, pp.104-114, Jan. 1976.

2.28 K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications.* Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 2007.

2.29 V. Potdar, A. Sharif, and E. Chang, "Wireless Sensor Networks: A Survey," *in Proceedings of* International *Conference on Advanced Information Networking and Applications Workshops, WAINA '09*, Bradford, UK, 2009, pp.636- 641.

2.30 H. Karl, and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. London, New York, USA: John Wiley & Sons, 2005.

2.31 K. Akkaya, M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad Hoc Network*, vol. 3, no. 3, pp. 325–349, May 2005.

2.32 H. Karl, A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Chichester, West Sussex, UK: John Wiley & Sons, 2005.

2.33 P. Tyagi, Ms. S. Jain, "Comparative Study of Routing Protocols in Wireless Sensor Network," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 9, September, 2012.

2.34 N. G. Palan, B. V. Barbadekar, and S. Patil, "Low energy adaptive clustering hierarchy (LEACH) protocol: A retrospective analysis," *in Proceedings of International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India, 2017, pp. 1-12.

2.35 L. Jing, F. Liu, and Yuling Li, "Energy Saving Routing Algorithm based on SPIN Protocol in WSN," *in Proceedings of International Conference on Image Analysis and Signal Processing*, Hubei, China, 2011, pp. 416-419.

2.36 B. Carter, and R. K. Ragade, "Message Transformation Services for Wireless Sensor" *Networks, in Proceedings of International Conference on Wireless Networks*, Las Vegas, Nevada, USA, 2006.

2.37 S. Ehsan, and B. Hamdaoui, "A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 265-278, January 2012.

2.38 N. Shabbir, R. Nawaz, M. N. Iqbal, and J. Zafar, "Routing protocols for a small scale WLAN based wireless sensor networks," *in Proceedings of* 9*th International Conference on Sensing Technologies*, Auckland, New Zealand, 2015, pp 412-415.

2.39 B. A. Bakr, L. Lilien, "LEACH-SM: A protocol for extending wireless sensor network lifetime by management of spare nodes," *in Proceedings of International Symposium on Modelling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), IEEE*, New Jersey, Princeton, 2011, pp 375.

2.40 M. Asif, S. Khan, R. Ahmad, M. Sohail, D. Singh, "Quality of Service of Routing Protocols in Wireless Sensor Networks: A Review," *IEEE Access*, vol. 5, pp. 1846-1871, Jan., 2017.

2.41 G. Yin, G. Yang, W. Yang, B. Zhang, and W Jin, "An energy-efficient routing algorithm for wireless," *in Proceedings of International Conference on Internet Computing in Science and Engineering (ICICSE'08),* Harbin, China, 2008 , pp. 181-186.

2.42 Z. Zang, J. D. Qi, Y. J. Cao, "A robust routing protocol in wireless sensor networks," *in Proceedings of IET International Conference on Wireless Sensor Network*, Beijing, China, 2010, pp. 276-279.

2.43 F. L. Lewis, *Wireless Sensor Networks*, New York, John Wiley & Sons, 2005.

2.44 B. Kundaliya, *Challenges of WSNs in IoT: Wireless Sensor Networks - Design, Deployment and Applications*. 2021.

2.45 J. Granjal, E. Monteiro, and J. S. Silva, "Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey," *Ad Hoc Networks*, vol. 24, no. 1, pp. 264–287, January 2015.

2.46 H. Kumarage, I. Khalil, A. Alabdulatif, Z. Tari, and X. Yi, "Secure data analytics for cloud-integrated internet of things applications," *IEEE Cloud Computing*, vol. 3, no. 2, pp. 46–56, May 2016.

2.47 A. Botta, W. D. Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, March, 2016.

2.48 M. Khan, B. N. Silva, and K. Han, "A web of things-based emerging sensor network architecture for smart control systems," *Sensors*, vol. 17, no. 2, February, 2017.

2.49 K. C. Serdaroglu, and S. Baydere, "WiSEGATE: Wireless Sensor Network gateway framework for internet of things," *Wireless Networks*, vol. 22, no. 5, pp. 1475–1491, July 2016.

2.50 V. C. Thang, and N. V. Tao, "A performance evaluation of improved IPv6 routing protocol for wireless sensor networks," *International Journal of Intelligent Systems and Applications*, vol. 8, no. 12, pp. 18–25, December 2016.

2.51 S. Raza, L. Seitz, D. Sitenkov, and G. Selander, "S3K: Scalable security with symmetric keys - DTLS key establishment for the internet of things," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 3, pp. 1270–1280, January 2016.

2.52 D. Puccinelli, and M. Haenggi, "Wireless Sensor Networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol. 5, no. 3, pp. 19–31, September 2005.

2.53 M. F. Othman, and K. Shazali, "Wireless Sensor Network applications: A study in environment monitoring system," *Procedia Engineering*, vol. 41, pp. 1204 – 1210, December 2012.

2.54 M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, March 2013.

2.55 T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in Wireless Sensor Networks: A top-down survey," *Computer Networks*, vol. 67, pp. 104 – 122, July 2014.

2.56 F. Aiello, F. Bellifemine, G. Fortino, S. Galzarano, and R. Gravina, "An agent-based signal processing in-node environment for real-time human activity monitoring based on wireless body sensor networks," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 7, pp. 1147–1161, October 2011.

2.57 M. Souil, and A. Bouabdallah, "On QoS provisioning in context-aware wireless sensor networks for healthcare," *in Proceedings of 20th International Conference on Computer Communications and Networks*, Lahaina, HI, USA, 2011, pp. 1–6.

2.58 N. Suryadevara, S. Mukhopadhyay, S. Kelly, and S. Gill, "WSN-based smart sensors and actuator for power management in intelligent buildings," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 564–571, April 2015.

2.59 A. I. Dounis, "Artificial intelligence for energy conservation in buildings," *Advances in Building Energy Research*, vol. 4, no. 1, pp. 267–299, January 2010.

2.60 K. Jaafar, and M. Watfa, "Sensor networks in future smart rotating buildings," *in Proceedings of* IEEE *Consumer Communications and Networking Conference*, Las Vegas, NV, USA, 2013, pp. 962–967.

2.61 F. Glover, and G. Kochenberger, *Handbook of metaheuristics: International Series in Operations Research & Management Science*. Springer, 2003.

2.62 C. Blum, and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, September 2003.

2.63 G. Beni, and J. Wang, "Swarm Intelligence in Cellular Robotic Systems," *in Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, Berlin, Heidelberg, 1993, pp. 703–712.

2.64 J. Silberholz, B. Golden, S. Gupta, X. Wang, M. Gendreau, J. Y. Potvin, "*Computational Comparison of Metaheuristics: Handbook of Metaheuristics*", Volume 272, Springer, Cham. , 2019, pp 581-604 .

2.65 J. Kennedy, and R. Eberhart, "Particle swarm optimization," *in Proceedings of Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948.

2.66 A. RezaeeJordehi, and J. Jasni, "Parameter selection in particle swarm optimisation: a survey," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, no. 4, pp. 527-542, June 2013.

2.67 M. A. Khanesar, M. Teshnehlab, and M. A. Shoorehdeli, "A novel binary particle swarm optimization," *in Proceedings of Control & Automation, Mediterranean Conference (MED'07)*, Athens, Greece, 2007, pp. 1-6.

2.68 R. C. Eberhart, and Y. Shi, "Particle swarm optimization: developments, applications and resources," *in Proceedings of* Evolutionary *Computation*, Seoul, Korea, 2001, pp. 81-86.

2.69 B. Bhushan, and S. S. Pillai, "Particle Swarm Optimization and Firefly Algorithm: Performance analysis," *in Proceedings of* IACC, Ghaziabad, India, 2013, pp. 746-751.

2.70 W. Xie, C. M. Ji, and X. W. Li, "Particle Swarm Optimization Based on Cultural Algorithm for Short-Term Optimal Operation of Cascade Hydropower Stations," *in Proceedings of* ICNC '*09*, Tianjian, China, 2009, pp. 289-293.

2.71 Y. M. Chen, and C. T. Lin, "A particle swarm optimization approach to optimize component placement in printed circuit board assembly," *The International Journal of Advanced Manufacturing Technology*, vol. 35, no. 5-6, pp. 610-620, November 2007.

2.72 J. Kennedy, and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *in Proceedings of Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, Orlando, FL, USA, 1997, pp. 4104-4108.

2.73 F. Zhao, Q. Zhang, and Y. Yang, "A Scheduling Holon Modeling Method with Petri Net and its Optimization with a Novel PSO-GA Algorithm," *in Proceedings of CSCWD '06*, Nanjing, China, 2006, pp. 1-6.

2.74 Y. Shi, and R. C. Eberhart, "Empirical study of particle swarm optimization," *in Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, Washington, DC, USA, 1999, pp 1945-1950.

2.75 T. Niknam, "A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem," *Applied Energy*, vol. 87, no. 1, pp. 327-339, January 2010.

2.76 Y. Valle, G. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Trans. on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, March 2008.

2.77 Y. D. Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, R. G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and

Applications in Power Systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171 – 195, March 2008.

2.78 R. C. Eberhart, and Y. Shi, "Particle swarm optimization: developments, applications and resources," *in Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea, 2001, pp. 81-86.

2.79 M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 26, no. 1, pp. 29-41, February 1996.

2.80 M. Dorigo, M. Birattari, and C. Blum, "*Ant colony optimization and swarm intelligence,*" Springer Verlag, 2004, Vol. 5217, no. 8, pp. 767–771.

2.81 M. Birattari, P. Pellegrini, and M. Dorigo, "On the Invariance of Ant Colony Optimization," *IEEE Transactions on Evolutionary Computation,* vol. 11, no. 6, pp. 732–742, November 2007.

2.82 M. Dorigo, and L. M. Gambardella, "Learning Approach to the Traveling Salesman Problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53 - 66, April 1997.

2.83 R. Mehrabian, C. Lucas, and S. Mohagheghi, "A novel Numerical Optimization Algorithm Inspired from Weed Colonization," *Ecological Informatics*, vol. 1, no. 4, pp 355-366, December 2006.

2.84 H. S. Rad, and C. Lucas, "A Recommender System based on Invasive Weed Optimization Algorithm," *in Proceedings of IEEE Congress on Evolutionary Computing,* Singapore, 2007, pp. 4297-4304.

2.85 A. Rahimi, P. Ramezani, M. Ahangaran, and T. Kashkooli, "An Improved Artificial Weed Colony for Continuous Optimization," *in Proceedings of UKSim 5th European Symposium on Computer Modeling and Simulation*, Madrid, Spain, 2011, pp. 1-5.

2.86 K. Affolter, T.Hanne, D. Schweizer, and R. Dornberger, "Index Tracking with Invasive Weed Optimization," *in Proceedings of International Conference on Soft Computing & Machine Intelligence*, New Delhi, India, 2014, pp. 110-114.

2.87 Z. Ren, S. Huang, C. Sun, and Y. Liang, "Improved Invasive weed optimization based on clustering strategy," *in Proceedings of Chinese Control and Decision Conference*, Yinchuan, China, 2016, pp. 1001–1005.

2.88 A. Tero, R. Kobayashi, and T. Nakagaki, "A mathematical model for adaptive transport network in path finding by true slime mold," *Journal of Theoretical Biology, Elsivier*, vol. 244, no. 4, pp. 553–564, February 2007.

2.89 S. Tsuda, K. P. Zaunerand, and Y. P. Gunji, "Robot control with biological cells," *Biosystems*, vol. 8, no. 2-3, pp. 215–223, February 2007.

2.90 M. Caleffi, F. AkyildizI, and I. Paura, "On the Solution of the steiner Tree NP-Hard problem via Physarum BioNetwork," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1092-1096, August 2015.

2.91 T. Miyaji, and I. Ohnishi, "Physarum can solve the shortest path problem on Riemannian surface mathematically rigorously," *International Journal of Pure and Applied Mathematics*, vol. 47, no. 3, pp. 353–369, January 2008.

2.92 A. Tero, T. Nakagaki, K. Toyabe, K. Yumiki, and R. Kobayashi, "A method inspired by Physarum for solving the Steiner problem," *International Journal of Unconventional Computing*, vol. 6, no. 2, pp. 109–123, January 2010.

2.93 A. Rekaby, "Directed Artificial Bat Algorithm (DABA)-A New Bio-Inspired Algorithm," *in Proceedings of International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Mysore, India, 2013, pp 1241-1246..

2.94 X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," *Nature Inspired Cooperative Strategies for Optimization, Studies in Computational Intelligence*, vol. 284, Berlin, Heidelberg, Springer, 2010, pp. 65–74.

2.95 R. Dwivedi, H. K. Sethi, M. Rohilla, V. Garg, and Y. Nagpal, "Edge detection using bat algorithm," *in Proceedings of 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom),* New Delhi, India, 2016, pp. 2605-2608.

CHAPTER 3

VLSI ROUTING OPTIMIZATION BASED ON PARTICLE SWARM OPTIMIZATION

3.1. Introduction

Expressing advancement as a standard peculiarity is a touch of embellishment, which incorporates monetary improvement to designing technique just as occupation booking to asset assignment. The objective of any improvement is to achieve the same or similar outcomes under constrained settings while either reducing or amplifying certain parameters. In the context of a VLSI physical design, there are a number of parameters that must be refined.

Every 18 months, the number of transistors in a device doubles, according to Moore's Law [3.1]. Meanwhile, transistor sizes are basically shrinking by the current ages as a result of massive growth in technologies, and more transistors are attaining adapted in that one chip location than before employing cutting-edge assembly processes. Interconnect length has so been greatly increased. Previously, updating the gate delay was sufficient, but since the establishment of the 130nm technological node, the connection delay has become even more apparent.

The goal of the VLSI physical strategy is to improve the device preparations and interconnection architectures among these devices prior to their expected presentation. The routing phase of VLSI physical design is primarily characterised by Global Routing and Detailed Routing, and it is at this phase that the wire length estimation of interconnects is measured. Circuit interconnections with the shortest possible wavelength and the lowest possible interconnect time are mandatory in the global routing phase. A sequential approach where VLSI nets are sequenced according on their criticality and the practical router's hire improvement point helps alleviate some of the world's routing concerns. Rerouting after severing obstructive wires [3.2] and the "shove-aside" operation [3.3], as well as offering a concurrent approach where the parallel integer programming concept is tested to improve global routing within constraints.

The Routing problem in VLSI physical design can also be planned in classical Graph Theory, where the wirelength minimization of connected VLSI circuit block depends on solving the RSMT [3.4], a prominent NP-Complete problem in Graph Theory. Such *NP*-complete problems are well suited for a branch of AI called Swarm Intelligence. Swarms share information and resources to keep going no matter what. Individually, these social experts have been shown to have restricted capabilities; together, however, they are well-suited for accomplishing a goal, and do it rather insightfully, given their reality. Scientists and experts felt compelled to replicate the successes of these natural multi-systems in their own work.

As the length of the interconnects has a significant effect on the delay of the circuit's devices, engineering geometry has become crucial to achieving sub-micron levels of reflection, circuit proficiency, and dependability. Algorithms are often helpful in bringing out improvement, as routing delay or interconnect length depends on aspects. In the past, reducing wire length was the best strategy for reducing routing latency and increasing circuit throughput. Nevertheless, it is certain that the routing way in addition to buffer insertion fundamentally upgrades this methodology. Delay enhancement is rudimentary to achieve the timing conclusion of a top-notch VLSI format.

Zhou et al. [3.4] projected the methodology of coordinated buffer incorporation while searching for the ideal routing path. Although it does find the shortest path from source to sink, this method is more involved than the rudimentary method of embedding buffers. Because of its early detection of and ability to avoid cushion obstructions, it outperforms more traditional approaches.

The Elmore wire model [3.5] provides precision for estimating delay in worldwide interconnect routing strategies. However, this eliminates the effects of inductance, which have been shown to cause a 35% time-delay miscalculation in nanometre VLSI innovation [3.6]. Reflective submicron technology has allowed the VLSI configuration to advance to the point where Friedman et al. [3.7] can predict a high-order RLC delay model. Patches are made to the RLC model in [3.8] to facilitate the sink-to-source and source-to-sink iterative defer figuring by Md. Yusof et al. [3.9], which are implemented in the two-way RLC based retribution as in the S-RABILA calculation [3.10].

In 1989, G Beni and J Wang [3.11] first used the term "swarm insight" to describe a new way of looking at problems in international development. The methods used there are heuristics or meta-heuristics. For the purpose of "discovery by experimentation," heuristics bring people together. If one trying to solve a difficult development problem (such as an *NP*-complete problem) in a reasonable amount of time, these methods will be very helpful. Particle Swarm Optimization is one widely used meta-heuristics calculation [3.12].

In this chapter, it is learned how to use algorithms based on PSO variants and hybrids to optimize wirelength and decrease interconnection delay in VLSI circuits. In the first section, two modification is done in weighted (PSO-W) algorithms [3.13]. The Self-Adaptive Acceleration Coefficient (PSO-SAAC) is a method for improving the properties of a searching procedure designed with a high rate of effective merging by readjusting the acceleration coefficients through a flexible system of local inquiry and global search. By incorporating a component related to a mutation factor into the position update trademark condition of PSO-W, an additional new strategy is implemented that brings together the intuition of the hereditary calculation with PSO-W generating PSO-Mutation (PSO-MU). In addition to the previously mentioned PSO alternatives, PSO with constriction factor (PSO-C) [3.14] is evaluated with these calculations, on a common ground for boosting VLSI global steering. The chapter also details the modified PSO-C algorithm in the second section for minimizing routing delays. CPSO-MU determines the minimally invasive placement of the wire, and followed by it examines the optimal region along the wire in which to embed the buffer.

3.2. Literature Review

VLSI Routing is well-thought-out to be the most critical step in the complete course of physical designing. The standard method for managing directing arranges the routing into two unique stages, the first being the phase of global routing followed by the second detailed routing. The ultimate objective of global routing is to shorten the complete flood and the general wire length other than lessening the execution time. The issue of global routing is regularly inspected as a diagram issue. The routing segments and their affiliations, limits, and interconnections are demonstrated as diagrams and in any case, the arrangement style solidly impacts the chart models used. To discover an ideal relationship among the given terminal nodes, different spanning tree algorithms are used yet the resulting cost of these RSMT acquired because of these algorithms are awful and thus, to diminish the general way cost of the routing ways, other than the terminal nodes, a bunch of additional places, called the steiner nodes are likewise considered which at last aides in additional decreasing the general expense of the MST and bring about the RSMT [3.15]. In order to reduce the cost of the routing path acquired with least spreading over algorithms, steiner nodes were suggested [3.16] for use in VLSI routing. Steiner nodes are not a piece of the first arrangement of vertices; however, they essentially limit the general way length producing Minimum Steiner Tree.

J. P. Cohoon et al. [3.17] reported the issue of observing the Rectilinear Steiner Tree (RST) with minimum length that interfaces a bunch of focuses on the limit of a rectangle. An algorithm shown for linear time problem.

Jan-ming Ho et al. [3.18] examined another way to deal with developing the RST of a given arrangement of focuses in the plane, beginning from a base traversing tree. The principle thought in methodology is to decide L-molded designs for the edges of the MST, to amplify the covers between the formats, subsequently limiting the expense (i.e., wire length) of the subsequent RST. A straight time calculation portrayed for building a RST from a MST, to such an extent that the RST is ideal under the limitation that the format of each edge of the MST is a L-shape. The RST's delivered by this calculation have 8-33% lower cost than the MST, with the normal expense improvement, over countless arbitrary point sets, being around 9%. The running season of the calculation on an IBM 3090 processor is under 0.01 seconds for point sets with cardinality 10.

An RST for a given arrangement of focuses is developed by Chao Ting-Hai et al. [3.19]. The suggested calculation outflanks most different calculations of its group by the way that the normal expense improvement over the rectilinear least traversing tree is 10.4%, and its time intricacy is $O\ (n^2 \log n)$.

Xianlong Hong et al. in 1993 [3.20] reported two execution driven steiner tree calculations for worldwide directing which consider the minimization of planning delay during the tree development as the objective. One calculation depends on nonlinear advancement technique, one more uses heuristic way to deal with guide the development of steiner tree. Another planning model is set up which incorporates both length and basic way among source and sink in defer definition, and an upper destined for timing postpone is deducted and used to direct the two calculations.

K. D. Boese et al. [3.21] gave another hypothetical system to developing steiner directing trees with least Elmore delay. Vitally hypothetical outcome is a speculation of Hanan's hypothesis which restricted the quantity of potential areas of steiner hubs in an ideal defer RST. Another hypothetical outcome builds up another decay hypothesis for developing ideal defer Steiner trees. A branch-and-bound strategy is fostered, called BB-SORT-C, which precisely limits any direct mix of Elmore sink delays; BB-SORT-C is down to earth for steering little nets and for delimiting the space of attainable steering arrangements concerning Elmore delay.

A calculation for the development of deferral limited multicast trees depends on the "least expensive addition" heuristic, a notable answer for the ordinary steiner tree issue in charts as reported by R. Novak et al. [3.22]. A least expensive inclusion heuristic adjusted for the requirement rendition of the Steiner tree issue by presenting a strategy for imperfect compelled briefest way calculation. The suggested steiner tree calculation utilizes the table passing instrument, rather than the much of the time utilized "waving" strategy.

S. Areibi et al. in 2007 [3.23] focused on global routing problem. The fundamental point is to foster a productive K Rectilinear Steiner Trees (K-RST) calculation. A K-RST routine is created to produce a bunch of RST for each net. The K-RST utilizes nearby tree fragment changes to guarantee that there is no duplication of steiner trees for a net. The most limited tree for a net is overall 11% more limited than that of the negligible crossing tree, which prompts region reserve funds.

Yu Hu et al. [3.24] presented a functional heuristic for RSMT development dependent on subterranean insect province streamlining ACO. This calculation was executed on a Sun workstation with Unix working framework and the outcomes have been contrasted and the GeoSteiner 3.1 and a new work utilizing grouped covetous triple development. Test results show that the calculation, named ACO-Steiner, can get an exceptionally short run time and keep the elite exhibition.

A two-stage calculation is insinuated by G. Grewal et al [3.25] for rapidly developing an assorted pool of Steiner trees for directing multi-terminal nets. In the principal stage, an original productive calculation, called Shrubbery, is utilized to develop excellent steiner trees to enter the pool. To guarantee assortment among pool individuals, a drawn-out memory and edge-weight irritation system is utilized to differentiate the hunt when looking for new arrangements. Nearby inquiry is utilized in the subsequent stage, to additionally work on the nature of trees in the pool. Computational analyses performed on north of 800 regularly utilized benchmarks show that the suggested calculation can create pools of ideal (or close ideal) trees in a tiny measure of time.

C. Chu et al. presented [3.26] an extremely quick and precise RSMT calculation called quick query table assessment (FLUTE). Woodwind depends on a precomputed query table to make RSMT development exceptionally quick and extremely exact for low-degree1 nets. For serious level nets, a net-breaking strategy is suggested to lessen the net size until the table can be utilized. A plan is likewise introduced to permit clients to control the tradeoff among exactness and runtime. Woodwind is ideal for low-degree nets (up to degree 9 in the present execution) is still extremely precise for nets up to degree 100. In this manner, it is especially appropriate for extremely enormous scope joining applications in which most nets have a level of 30 or less. A north of 18 modern circuits in the ISPD98 benchmark suite showed, FLUTE with default exactness is more precise than the Batched 1-Steiner heuristic and is nearly pretty much as quick as an exceptionally productive execution of Prim's rectilinear least crossing tree calculation.

Hiroshi Totsukawa et al. reported [3.27] a genetic algorithm where the three-dimensional rectilinear Steiner tree with limited number of twists is acquired by supplanting each edge of the given Euclidean traversing tree by the sections which are corresponding to the X-hub, the Y-hub, or the Z-pivot. In the suggested strategy, the calculation can stay away from hindrances deftly by utilizing, probably, three curves to supplant one edge of the Euclidean crossing tree. For the wellness esteem, a straight amount of the wire length and breadth of the RST is utilized. In the trial results, it is shown that suggested equal hereditary calculation can stay away from snags, and get the three-dimensional rectilinear Steiner tree with limited number of curves.

X. Ma et al. [3.28] suggested a molecule swarm streamlining for taking care of steiner tree issue. In the calculation a tree structure portrayal is utilized to encode a molecule. To understand the transmission of tree structure data a clever strategy for particles flying in search space is suggested. X. Ma et al. additionally present the r-neighborhood ring geography of particles to improve the capacity of nearby and worldwide pursuit of PSO calculation, and a molecule change strategy to keep the variety of molecule populace. Thorough recreation tests are done on various issues and diverse organization geographies. The outcomes show that the suggested calculation has great scanning execution for observing ideal steiner tree.

Lie Genggeng et al. [3.29] presented a RSMT calculation dependent on discrete PSO (DPSO), to be specific BRRA_DPSO, to limit the wiring length and diminish the quantity of curves, which is useful for through decrease and dependability increase in the steering stage. Genetic Algorithm (GA) suggested to tackle the issue of the sluggish assembly pace of PSO utilized for a high-layered space improvement, a self-adjusting procedure that can change the learning variables, and consolidate with the hybrid and transformation administrators. The outcomes show that the suggested calculation can proficiently furnish the arrangement of RSMT issue with great quality and meet more quickly than GA. Additionally, the calculation can likewise lessen the quantity of twists.

There is a BOB-RSMT-based algorithm that was developed by Y. Zhang et al. [3.30]. Using buffering-attention, the suggested calculation moves starting tree structures steadily and efficiently to meet slew constraints while keeping wire length to a minimum. It can deal with complex squares including rectilinear shapes. The investigations on different benchmarks exhibit extremely encouraging outcomes. When compared to the conventional Obstacle Avoiding RSMT (OA-RSMT) calculations, the outside block wire length and the absolute wire length are reduced by making strategic use of over-the-block steering assets. RSMT likewise decreases the repeater count or region expected to fulfil slew limitations, which is vital for present day configuration conclusion.

N. Maharjan et al. [3.31] elaborated another technique for addressing the MST in an undirected diagram. This strategy depends on the conduct of genuine subterranean insects and takes care of the issue in lesser opportunity to create the Minimal Steiner tree. The overall idea of observing a base way creates from the deviation of the subterranean insects' way which is impacted by other subterranean insect settlements or the ways made by subterranean insects of different provinces when they move starting with one state then onto the next and consequently subsequently framing the Steiner tree. The way so shaped is viewed as the base distance way between every one of the provinces in question.

N. R Latha et al. [3.32] worked-on Woodwind (Fast Look-Up table) based methodology. Woodwind (Fast Look-Up table) based methodology introduced a quick and precise RSMT development for both more modest and more significant level nets. The model lessens the time intricacy for RSMT development for more modest nets, but for bigger nets there exists memory overhead. Since woodwind-based model did not consider the memory necessity in developing RSMT, the suggested work presents a Memory-Enhanced RSMT (MORSMT) development to address the memory overhead for bigger nets. Tests are led to assess the presentation of the suggested approach over a current model for differed benchmarks as far as calculation time, memory overhead, and wire length.

Gengjie Chen et al. [3.33] reported a steiner SLT development technique called Steiner SLT (SALT), which is productive and has the most impenetrable bound over all the cutting-edge general-diagram SLT calculations. Applying SALT to Manhattan space offers a smooth tradeoff between rectilinear steiner least tree and rectilinear steiner least arborescence for VLSI directing. The adaption additionally decreases the time intricacy from $O\ (n^2)$ to $O\ (n\ logn)$.

The above reported works on RMST problem is solved using different approaches in different domain.

The routing problem in VLSI chip is mapped as RSMT problem where researcher suggested efficient method with the usage of heuristic and meta heuristics method. Swarm based method [3.34] - [3.36] became prominence in VLSI problems as found in literature.

T. Arora et al. applies [3.37] ACO to the NP-hard issue of tracking down ideal courses with least capacitance for interconnect directing on VLSI chips. The limitations on interconnect steering are utilized by insects as heuristics which guide their pursuit cycle ACO calculations executed on both Manhattan and non-Manhattan directing models. The outcomes are contrasted and a few best-in-class scholarly switches. The ACO steering calculation had the option to acquire a general improvement of 8% as far as wire-length, 7% as far as vias and capacitance. Running occasions were longer than those switches, however basically the same as the other switch which can course all wires on all benchmark chips.

T. Arora et al. [3.38] applies ACO to the *NP*-hard issue of tracking down ideal courses for interconnect directing on VLSI chips. The imperatives on interconnect directing are utilized by subterranean insects as heuristics which guide their pursuit interaction. It is observed that ACO calculations had the option to effectively fuse numerous requirements and course interconnects on set-up of benchmark chips. On a normal, the calculation steered with absolute wire length 5.5% not exactly other set up directing calculations.

S. Manna et al. [3.39] portrays the worldwide routing in VLSI utilizing Differential Evolution (DE) based enhancement method to discover the base directing wire length. This paper proposes an original way to deal with apply of Differential Evolution calculation for taking care of the streamlining issues in discrete spaces.

The DE calculation [3.40] has shown great application impact in taking care of different *NP*-hard issues. Hence, in light of the possibility of DE calculation, H. Wu et al. [3.41] proposes a RSMT development calculation for tackling this issue.

P. Bhattacharya et al. [3.42] works on algorithm in light of Artificial Bee Colony (ABC) for working out the wire length in worldwide steering and the upgraded outcomes are thought about against the standard switch NTHU 2.0. NTHU-Route 2.0 is a fast and vigorous worldwide switch which addresses all ISPD benchmarks keeping up with excellent quality.

H. Zhang et al. [3.43] presents a methodology that applies the ABC calculation to the Two-Terminals-Net-Routing (TTNR) issue in VLSI actual plan and contrasts its exhibition and the labyrinth calculation variation known as the best in class worldwide directing calculation.

A. Khan et al. [3.44] suggested a plan for global routing in view of contemporary ST calculations: Firefly Algorithm (FA), and ABC calculation and have thought about the presentation of the two. FA produces better improvement brings about correlation than ABC despite it ends up very costly, computationally.

M. N. Ayob et al. [3.45] investigates the utilization of FA in routing of VLSI. The area of doglegs is utilized to display the firefly that addresses the directing arrangement. The suggested approach is then contrasted and past writing for benchmarking. The outcome shows that it has a decent potential in VLSI and can be additionally stretched out in future.

Researchers also suggested the use of Honey Bee algorithm [3.46] - [3.48] for routing optimization in wireless network which can be efficiently used in VLSI routing optimization.

G. Chen et al. [3.49] reported a decision algorithm in light of Particle Swarm Optimization (PSO) to get a possible floor arranging in VLSI circuit actual situation. The PSO was applied with whole number coding in view of module number and another suggested worth of speed increase coefficients for ideal arrangement. The calculation can stay away from nearby least and performs well in union.

H. Zhou et al. [3.50] think about limitations on buffer location and address the synchronous maze routing and support inclusion issue. Given a square situation characterizing support area limitations and a couple of pins (a source and a sink), a polynomial time accurate calculation used to observe a cradled course from the source to the sink with least Elmore delay.

B Kantha et al. [3.51] suggested usage of PSO in VLSI sensor device and N B Singh et al. [3.52] - [3.54] reported employ of PSO in VLSI nano device. A. Khan [3.55], [3.56] insinuated PSO in solving VLSI global optimization.

G. Liu et al. [3.57] works on issues of RSMT. It limits the wiring length and number of bends. For taking care of slow slow convergence rate of PSO Genetic Algorithm suggested. The results proved that suggested calculation can effectively give the arrangement of the RSMT issue with great quality and combine more quickly than GA. Besides, the calculation can likewise decrease the quantity of twists. An algorithm [3.58] based on parallel neural network is likewise reported.

H. Wang et al. [3.59] reported an algorithm on PSO that is surrogate-assisted. In the reported calculation, a worldwide model administration system is created, which looks for something good and most unsure arrangements as indicated by a substitute troupe utilizing a PSO calculation and assesses these arrangements utilizing the costly genuine capacity.

L. Zhao et al. [3.60] suggested another PSO with dynamic processing of constraints. In the first place, requirements partitioned into three cases and plan another PSO based hybrid administrator to progressively manage limitations; second, a typical dissemination is added into PSO to function as change administrator, which will upgrade the variety of the multitude and explore the pursuit course.

3.3. Wirelength minimization of VLSI circuits using variants and hybrid of Particle Swarm Optimization

PSO is an iterative method that uses a group of specialists to find the optimal solution to a problem in a complex search space. An n-agent team is given the option of dividing up the search space into d dimensions. Each agent has its own unique set of position and velocity vectors, $X_i = \{X_1, X_2, \dots \dots, X_n\}$ and $V_i = \{V_1, V_2, \dots \dots, V_n\}$ respectively. These vectors are periodically updated so that they continue to satisfy the PSO criterion conditions as given in (2.15) and (2.16), re written as given in (3.1) and (3.2).

$$V_{i,t+1} = V_{i,t} + c_1 r_1 (p_{best} - X_{i,t}) + c_2 r_2 (g_{best} - X_{i,t}) \tag{3.1}$$

$$X_{i,t+1} = X_{i,t} + V_{i,t+i} \tag{3.2}$$

Here, the effects of individual information on a typical particle and, similarly, the effects of group information on a particle population are controlled by the constants c_1 and c_2. Random numbers with uniform distributions are used for r_1 and r_2 [3.61]. Particles are assigned positions arbitrarily and kept on being encouraged by the fitness value biased by the p_{best} value (best position value of the individual) and the g_{best} value (best position value of the entire swarm) in the hopes of achieving the best possible outcome.

3.3.1. Modification PSO algorithm parameters and variants

The velocity of the particles is an important parameter of the PSO algorithm because it determines the swarm's step size at each iteration. Particles change their speed and direction in the problem space in every direction at every time step. If the particle is moving at a very high rate, its assessment attribute will be quite high, and it may swerve and empty the search space's fringes very quickly. In contrast, when velocities are low, particle crusades are limited to a narrow front and take place within the bounds of a local maximum. As a result, a parameter V_{max}, assumed to be $V_{max} = \frac{(X_{max}-X_{min})}{k}$, must be placed to preserve a balance between exploration and exploitation. Value 2 [3.62] is chosen as the empirical value of k.

Particle Momentum was ranked using the inertia weight (w), which is adapted in [3.63] developing PSO-W as a substitute for V_{max}. In order for the algorithm to assemble more effectively, it is implemented that the swarm's exploration and exploitation skills be ordered. Therefore, it is rewritten (3.1) as (3.3).

$$V_{i,t+1} = wV_{i,t} + c_1 r_1 (p_{best} - X_{i,t}) + c_2 r_2 (g_{best} - X_{i,t}) \tag{3.3}$$

According to conventional wisdom, the size of the search space is determined by the inertia weight w. For composite high-dimensional problem spaces, a large value of w is crucial, while a small value is sufficient for low-dimensional search spaces.

The inertia weight can be changed by a factor of (3.4), where s represents the population size, D the size of the dimension, and R the comparative worth of consistent results that are consistent with [0,1].

$$w = \left[3 - exp\left(-\frac{s}{200}\right) + \left(\frac{R}{8} \times D\right)^2\right]^{-1} \tag{3.4}$$

A new parameter χ, known as the constriction factor described in, is rationalised to replace the inertia weight w and the maximum velocity V_{max} within the PSO algorithm (3.6). This was pioneered by Clerc [3.64] and has shown to be of great importance in adjusting the exploration and exploitation trade-off and ensuring a productive coincidence of algorithm. The equation (3.1) is adjusted as (3.5).

$$V_{i,t+1} = \chi[V_{i,t} + \Phi_1(p_{best} - X_{i,t}) + \Phi_2(g_{best} - X_{i,t})] \quad (3.5)$$

$$\chi = \frac{2}{\left[2-\phi-\sqrt{\phi^2-4\phi}\right]} \quad (3.6)$$

For this situation, it is denoted as $\phi = \phi_1 + \phi_2$, $\phi_1 = c_1 r_1$ and $\phi_2 = c_2 r_2$. Typically smearing the value of ϕ as 4.1 the control of χ outcomes to 0.729. It follows that the particles promptly adjust their trajectory manipulated by p_{best} and g_{best} with guaranteed union, as indicated by the formula $\chi w = 0.729\ w < w$. To get to $(p_{best} - X_{i,t})$ and $(g_{best} - X_{i,t})$, it is multiplied by $2 * 0.729 = 1.458$ [3.65]. These standards are typically favoured because they improve equilibrium and cohesion. The acceleration coefficient as a PSO parameter is defined where both c_1 and c_2 can typically take the value 2 [3.66]. To achieve a superior result in the belvedere of track minimization of VLSI global routing, there are two distinct methods by which to examine the evenness relating to these parameters.

3.3.1.1. Self-Tuned

The weighted PSO (PSO-W) in (2.17) is modified where acceleration coefficients are given by (3.7).

$$c_1 = c_2 = c_{max} - [(c_{max} - c_{min}) * iter]/max\_iter \quad (3.7)$$

where, c_{max} and c_{min} are initial and initial and final coefficient respectively. $'iter'$ is the current iteration number and $1.49 < (c_1 = c_2) < 2$. This Self-Tuned PSO (PSO-ST) [3.67] algorithm linearly reduces the acceleration coefficients c_1 and c_2 over the range of time steps 2 to 1.49. The algorithm is initialised with accurate data, specifically $c_1 = c_2 = 2$. This modification of linear decline can preserve the swarm's exploration and exploitation capacity efficiently for speeds increase, and it can also provide a fast junction to the algorithm. It turns out that this method can provide the best possible results with a very high conjunction rate.

3.3.1.2. Self-Adaptation

Further Acceleration constant of PSO-W is modified with new mechanism where the algorithm modifies the trade-off between global exploration and local exploitation by mixing the two acceleration relentless factors c_1 and c_2 in a way that maximises the former's influence. Starting out with the swarm's highest exploration and lowest exploitation abilities, the algorithm gradually refines these characteristics over time. Thus, the swarm's particles are capable of consistently dispersing across the search space, as envisioned by the social module of the velocity vector in the initial experimentation phase. In the next step of the experiment, the swarm's perceptual component surpasses its social component, and they arrive at the local search course based on the evaluated results of the global search process with the goal of finding the unique local optima. This self-adaptive mechanism is given by (3.8) and (3.9).

$$c_1 = c_2 = c_{max} - [(c_{max} - c_{min}) \times iter\,]/max\_iter \tag{3.8}$$

$$c_1 = c_2 = c_{max} + [(c_{max} - c_{min}) \times iter\,]/max\_iter \tag{3.9}$$

where $1.35 < (c_1 = c_2) < 2.45$. The algorithm named as Self-adaptive PSO (PSO-SAAC) can effectively generate the most notable g_{best} value throughout the entire searching process, hence increasing the optimisation rate.

3.3.1.3. PSO-Mutation

In the section of PSO [3.86] devoted to presenting the opinion of the Genetic Algorithm, a new algorithm is provided. After a predetermined amount of time has passed, the algorithm makes good on its initial promise by picking swarms from the currently active generation. The swarms with the highest fitness probability are chosen, with a probability of selection factor equal to $\frac{f_j}{\sum_{j=1}^{N} f_j}$,where N is the total population size. Later, a mutant is created by isolating the high fitness component from the selected pool. As a result of this mutation in PSO [3.67], a new generation of swarms is evolved, each with a greater understanding of the high fitness attribute produced in the position vector (3.2). In (3.10), the following is the recommended position vector and is named as PSO-Mutation (PSO-MU).

$$X_{i,t+1} = (\psi \times X_{i,t} + \xi\,) + V_{i,t+1} \tag{3.10}$$

where, ψ is the randomization factor and ξ is the mutant fitness factor.

3.3.2. Grid graph model and Encoding

A VLSI Physical plan's global routing issue test has two aims. First in restricting power distribution and speeding up motion among segments or squares in VLSI architecture by reducing the total wire length of connecting terminals or squares. The Global routing issue can be officially described where $N = \{N_1, N_2, N_3 \dots N_m\}$ is the arrangement of nets representing interconnections between blocks in the VLSI design and D_i is the assessed wirelength of net N_I, $1 < I < m$. The general entire wirelength $\sum_{i=1}^{m} D_i$ can be limited by communicating issue capacity. Global routing plan is completed by planning VLSI design in Grid Graph model. The network diagram model is used to calculate the above. Figure 3.1 shows a matrix diagram of a directing area design, $G = (V, E)$. Vertex v_i and the edge e_{ij} linking the two adjacent vertices v_i and v_j are suggested by every phone addressing directing region between blocks as unfilled regions. The nodes and edges resemble VLSI routing techniques, and the vertices appear like VLSI blocks.

The VLSI routing challenge for a multi-terminal net must be expressed as the problem of acquiring an RSMT from a graph. The minimum spanning tree of interconnected terminal nodes is calculated using graph computations to estimate the base cost of interconnected length. With the presentation of arbitrary steiner nodes with terminal nodes of multi-terminal VLSI format, the expenditure or general wire length is further reduced, generating the graph's basic steiner tree cost (length). The cost or length can be reduced by the position and number of steiner nodes. With several terminal hubs, selecting the number of steiner nodes and their desired placement becomes computationally difficult, so the PSO calculation is used to choose the possible number and create this arbitrary situation to increase the steiner cost.

The method generated various swarm sizes of z particles and put them in a $n \times n$ diagram. Each swarm has $(p - 2)$ arbitrarily produced Steiner focuses drawn from Steiner set S with $(n\,||2-p)$ points, where p is the number of terminal nodes with assigned vertex $V_{ij} = \{1,2,3,\dots r\}$ and Steiner subset $Q_j \subseteq S$, where $j = \{1, 2, 3\dots\dots, z\}$. 100×100 pursue space is used. The issue matrix addresses 1 the characterised destination nodes or terminal nodes. To reduce computational complexity, rows and columns lacking destination nodes are removed, creating the decreased network Steiner focuses are provided randomly in the issue space, meaning 1. Figure 3.2 shows the reduced network and steiner lattice.

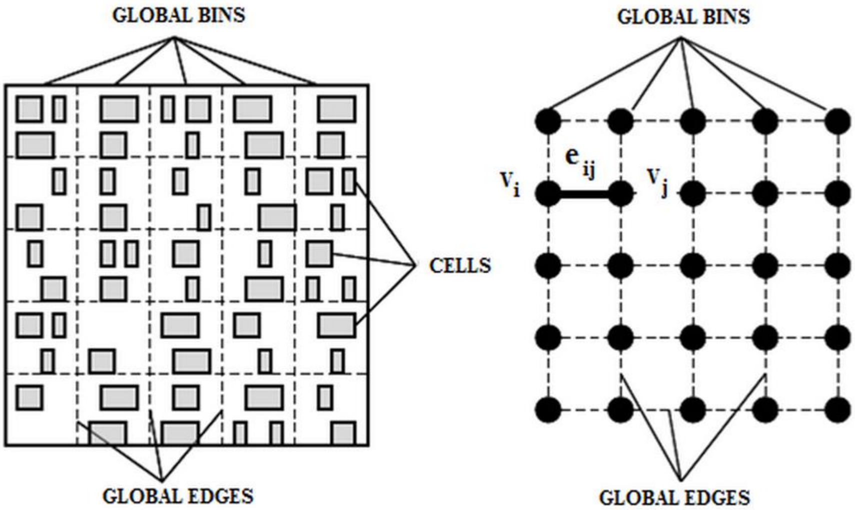

Figure 3.1. Grid Graph showing routing regions

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad X = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \boxed{1} & 0 & 0 \\ 0 & \boxed{1} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 & \boxed{1} \\ 0 & 0 & 0 & \boxed{1} & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 1 & 0 \end{bmatrix}$$

Figure 3.2. Reduced Graph Matrix and Steiner Matrix

3.3.3. Flowchart and algorithm of PSO variants

PSO planning uses steiner networks as particles. Figure 3.2 shows a particle with destination nodes. Wellness F_i for each particle seed is calculated by calculating RSMT or MRST cost using objective capacity $MST\ (G_i)$ and the least $MST\ (G_i)$. PSO parameters and most iterations are static. PSO velocity conditions with acceleration coefficient tuned in straight declining or self-versatile mode and the relating position condition in conventional mode or with transformation factor are used to assess p_{best}

and g_{best} values. The advanced MRST cost is the best value created or best multitude particle. The advanced MRST cost at the end of the PSO computation is the base length of the interconnected terminal nodes in the VLSI framework, allowing the least wire length routing method. Figure. 3.3 shows the PSO stream outline and variations.

Algorithm:

```
Step 1.  Search space, terminal nodes, swarm size and no. of iterations are initialized.
Step 2.  Generate an initial population of particles $X_i = \{X_1, X_3, \dots \dots, X_n\}$
Step 3.  Fitness is calculated as $f(X_i)$ and $MIN(f(X_i))$
Step 4.  Acceleration constant, $c_1$ and $c_2$ are evaluate according to any of the variants mentioned in this chapter.
Step 5.  Evaluate Inertia Weight $w$ as in (3.4) or evaluate constriction factor $\chi$ as in (3.6)
Step 6.  Set $p_{best}$= $f(X_i)$ and $g_{best}$= $MIN(f(X_i))$
Step 7.  for $i = 1: n$ (for particles)
Step 8.  Calculate particle velocity $V_{i,t+1}$ according to the velocity equation as in (3.3) or (3.5)
Step 9.  Update the particle position $X_{i,t+1}$ in accordance to position equation as in (3.2)
Step 10. Or update the particle position as in (3.10)
Step 11. Evaluate $f(X_i)$ and $MIN(f(X_i))$
Step 12. Update $p_{best}$ and $g_{best}$
Step 13. end for $n$
Step 14. $t = t + 1$
Step 15. end while
Step 16. Post processing the results and visualization
```

Flowchart:

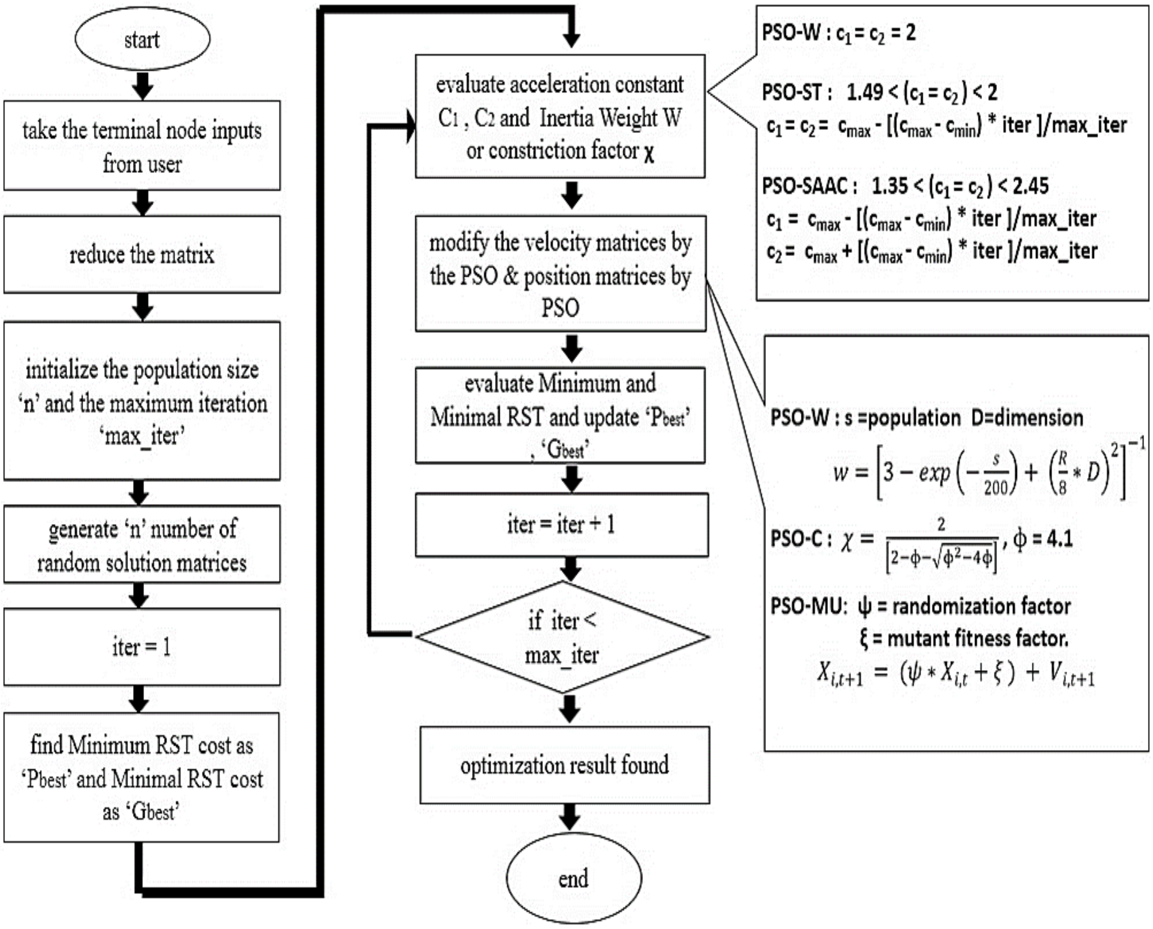
start
take the terminal node inputs from user
reduce the matrix
initialize the population size 'n' and the maximum iteration 'max_iter'
generate 'n' number of random solution matrices
iter = 1
find Minimum RST cost as 'Pbest' and Minimal RST cost as 'Gbest'
evaluate acceleration constant C1 , C2 and Inertia Weight W or constriction factor χ
modify the velocity matrices by the PSO & position matrices by PSO
evaluate Minimum and Minimal RST and update 'Pbest' , 'Gbest'
iter = iter + 1
if iter < max_iter
optimization result found
end
PSO-W : $c_1 = c_2 = 2$
PSO-ST : $1.49 < (c_1 = c_2) < 2$
$c_1 = c_2 = c_{max} - [(c_{max} - c_{min}) * iter]/max\_iter$
PSO-SAAC : $1.35 < (c_1 = c_2) < 2.45$
$c_1 = c_{max} - [(c_{max} - c_{min}) * iter]/max\_iter$
$c_2 = c_{max} + [(c_{max} - c_{min}) * iter]/max\_iter$
PSO-W : s =population D=dimension
$w = \left[3 - exp\left(-\frac{s}{200}\right) + \left(\frac{R}{8} * D\right)^2\right]^{-1}$
PSO-C : $\chi = \frac{2}{\left[2-\phi-\sqrt{\phi^2-4\phi}\right]}, \phi = 4.1$
PSO-MU: ψ = randomization factor
ξ = mutant fitness factor.
$X_{i,t+1} = (\psi * X_{i,t} + \xi) + V_{i,t+1}$

Figure 3.3. Flow chart of Variants of PSO

3.3.4. Experimental Procedure

A fixed two-dimensional 100 × 100 search space as height × width is used to randomly generate two coordinate sets of *15* terminal nodes based on the various distribution topologies of terminal nodes in a VLSI system of micro-metre or nano-metre. Figure 3.4 and Figure 3.5 graphically represent the coordinate sets for a nearly Uniform distribution and a Bivariate distribution, respectively.

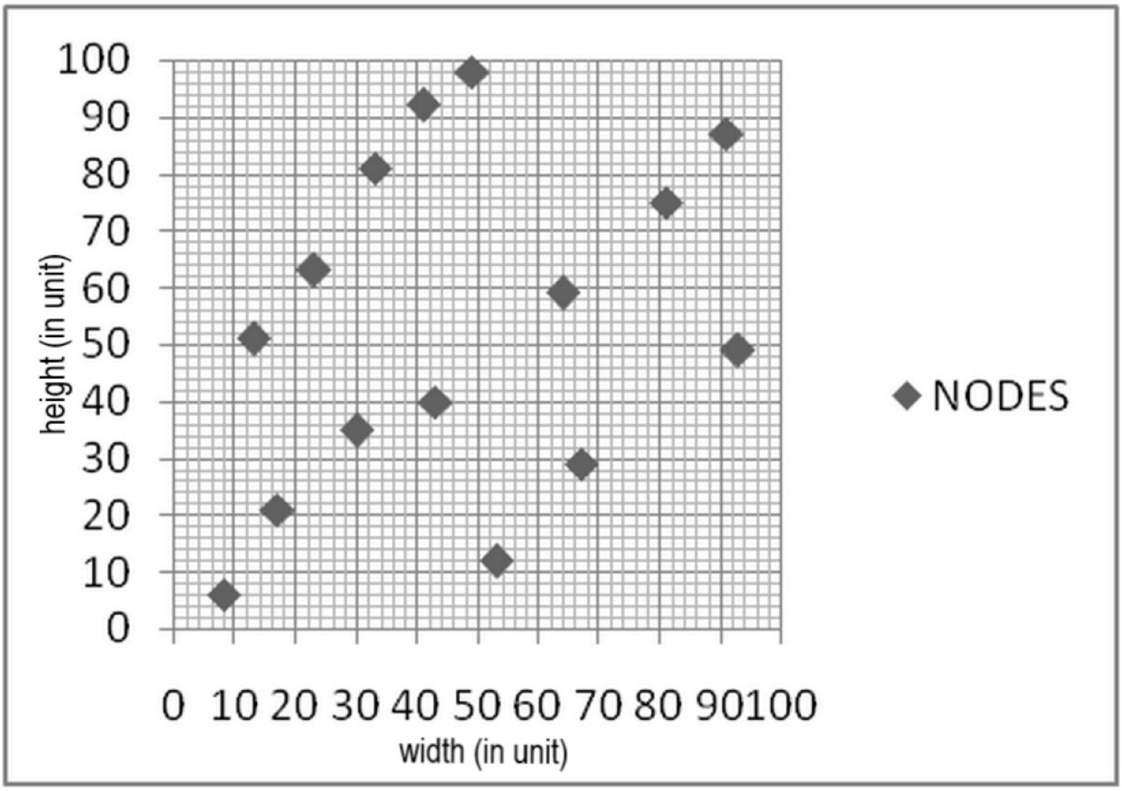

Figure 3.4. Uniform distribution of terminal nodes

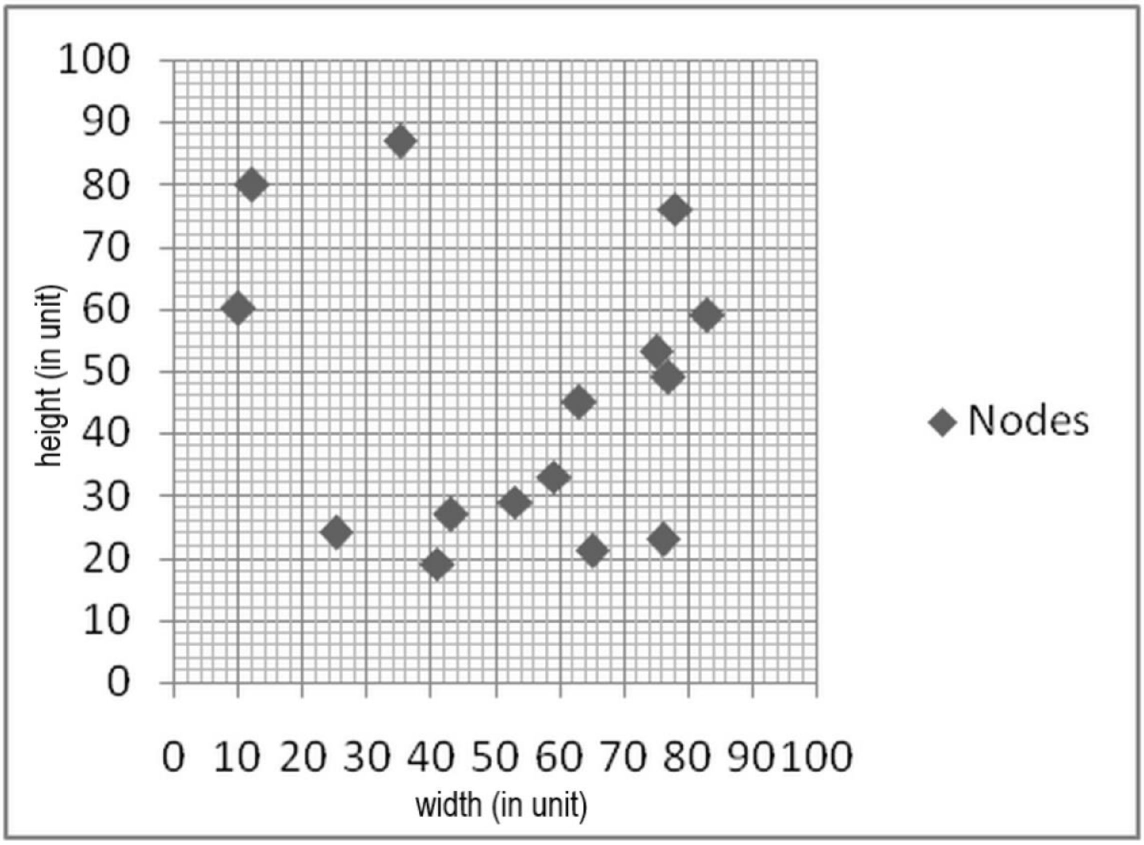

Figure 3.5. Bivariate distribution of terminal nodes

3.3.5. Experimental Results and Discussions:

For the two aforementioned coordinate sets, separate research is conducted on not only PSO-W but also on the modified algorithms PSO-ST [3.67] and PSO-SAAC to relate the terminal point for each set and to refund the base expense of interconnection equitably. Recorded are the initial VLSI global interconnection cost, the average cost in micro metre (μm) of VLSI worldwide interconnection across all *25* iterations of the algorithms, and the typical irregularities that arise due to these factors. Table 3.1 summarises the results of comparative investigations of PSO-SAAC, including the repercussions of both normal g_{best} and least g_{best}.

Table 3.1. Performance Comparison PSO-ST and PSO-SAAC

Test case	g_{best} value	PSO-W	PSO-ST	PSO-SAAC
SET 1	Average	354.5	348.4	341.7
	Minimum	350	343	338
SET 2	Average	256.7	254.9	257.3
	Minimum	253	253	255

PSO-SAAC outperforms the other two methods when it comes to maintaining a nearly uniform distribution of terminal hubs across the VLSI design. As shown in Figure 3.6, SET 1's global base interconnection cost of *'338'* provides an incentive for PSO-SAAC. As can be shown in Table 3.1, PSO-self-tuned ST's acceleration consistently controlling component is superior to the other two algorithms when it comes to bivariate appropriation of terminal nodes in VLSI design. As shown in Figure 3.7, PSO-ST generates the lowest interconnection cost of *253* under the assumption of uniform allocation. It is also seen that the typical interconnection cost of the VLSI worldwide best parameter grows further thanks to the speed increase consistent tuning component of PSO. Consequently, it can be safely specified that the RSMT, made by connecting the terminal nodes, is reduced in cost by PSO-SAAC for nearly uniform distribution and by PSO-ST for enhanced random bivariate distributions. As a result, the VLSI connection length can be reduced to an infinitely small value, and the RSMT problem of graphs can be solved effectively. From Table 3.2, it is also seen that the standard deviation of PSO-SAAC is lowest for SET 2, whereas the standard deviation of PSO-ST is lowest for SET 1. This implies that the self-tuned mechanism of PSO is more trustworthy in the case of extremely random distribution of terminal nodes in the prescribed search space, while the self-adaptive mechanism of PSO safeguards more steadiness for relatively uniform and less arbitrary distribution.

Table 3.2. Comparative studies of PSO-SAAC over SD

Test case	PSO-W	PSO-ST	PSO-SAAC
SET 1	7.77	0.71	5.41
SET 2	1.94	1.88	4.12

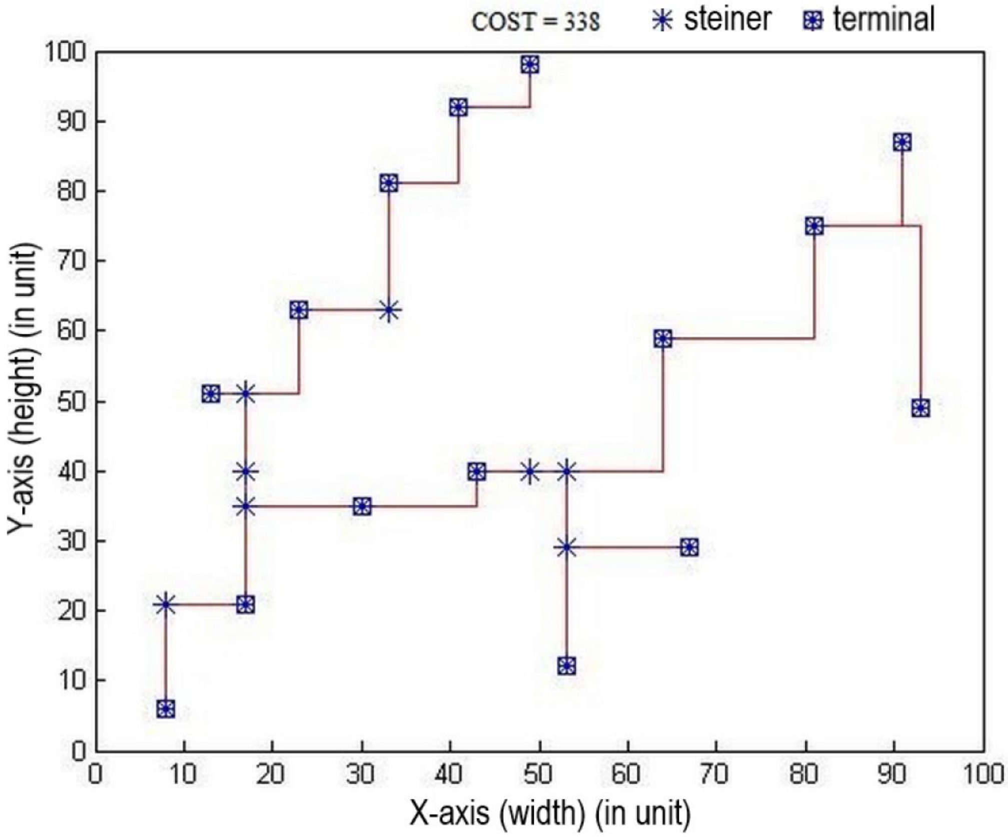

Figure 3.6. SET 1: wirelength 'cost' obtained for PSO-SACC

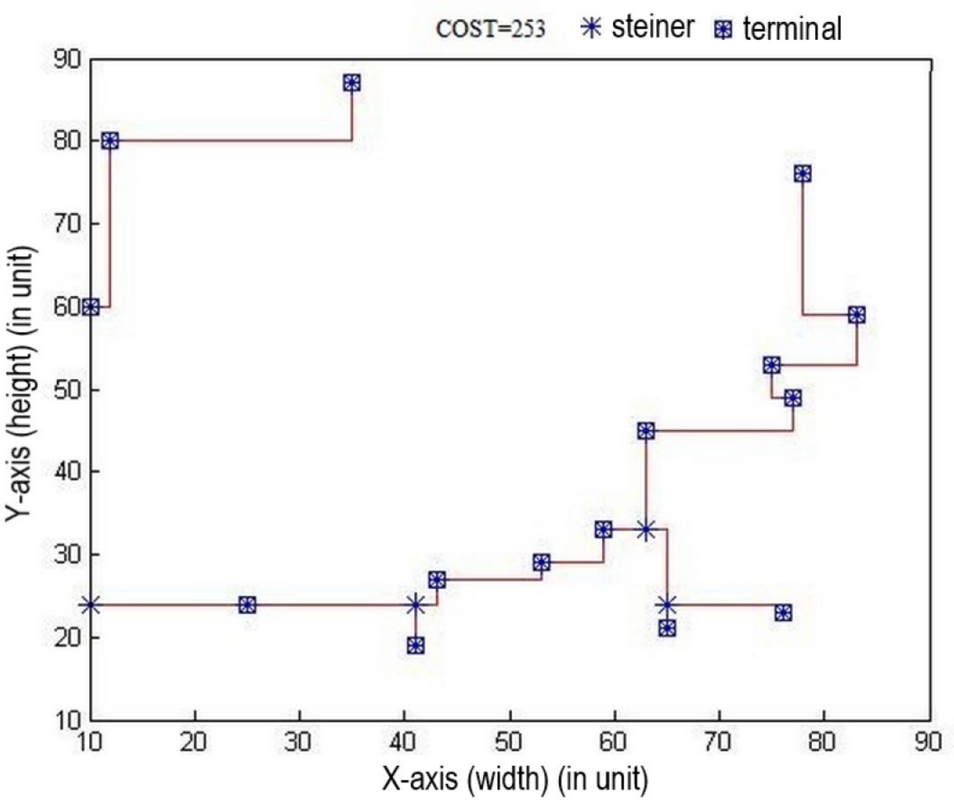

Figure 3.7. SET 2: wirelength 'cost' obtained for PSO-ST

Weighted PSO (PSO-W), PSO with a constriction factor (PSO-C), and PSO with a mutation algorithm (PSO-MU) are all used in the trials, in that order, for all pairs of measured coordinates. Table 3.3 summarises the results of all algorithms in terms of least interconnect cost, typical cost, and average implementation time.

Table 3.3. Comparative studies of PSO-MU

Test case	g_{best} value	PSO-W	PSO-C	PSO-MU
SET 1	Average (in unit)	354.5	350.4	336.8
	Minimum (in unit)	350	345	329
	System time (in Sec)	52.825	101.51	85.48
SET 2	Average (in unit)	256.7	256	250.4
	Minimum (in unit)	253	254	248
	System time (in Sec)	49.05	86.01	66.96

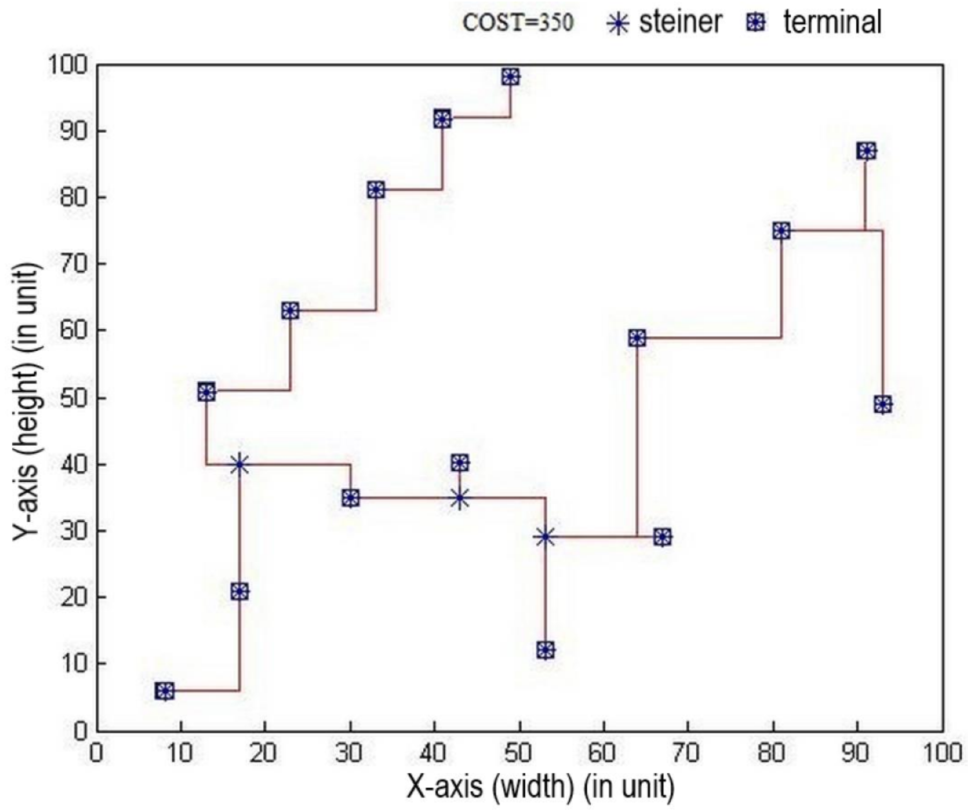

Figure 3.8. SET 1: wirelength 'cost' obtained for PSO-W

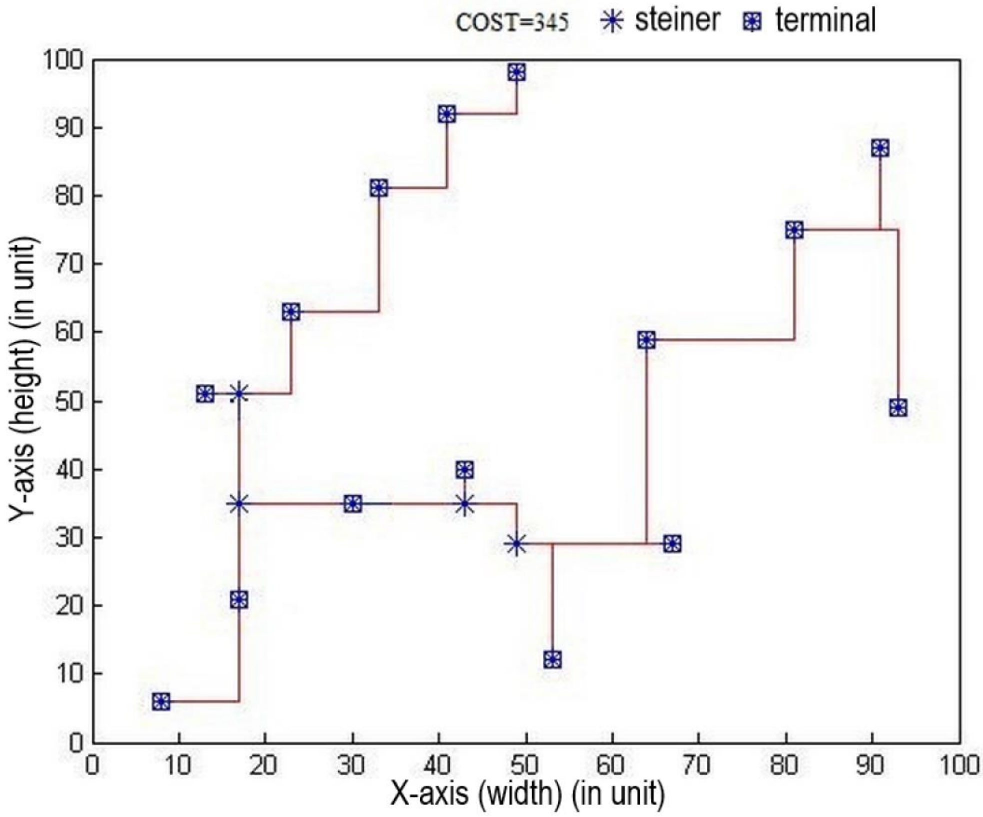

Figure 3.9. SET 1: wirelength 'cost' obtained for PSO-C

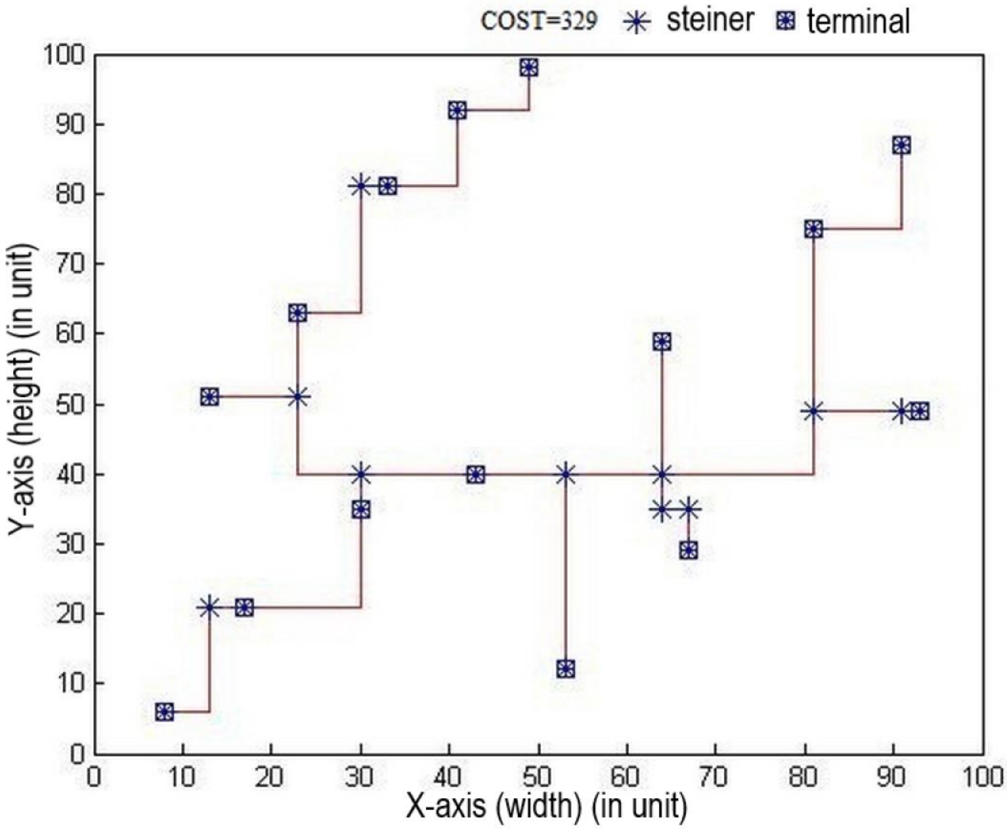

Figure 3.10. SET 1: wirelength 'cost' obtained for PSO-MU

In Figure 3.8, Figure 3.9, and Figure 3.10, it is seen that for generating RSMT, all of the aforementioned methods generated for the aforementioned VLSI topologies with the lowest possible interconnect costs. It reveals that PSO-MU can generate the minimal mean value and the lowest possible global best value in both sets of coordinates. In comparison to PSO-W and PSO-C, the presented algorithm, PSO-MU, ensures efficient VLSI global routing cost minimization and better convergence.

According to Table 3.3, the PSO-MU algorithm takes significantly longer to run than PSO-W, with its g_{best} value initially being determined to be *'329'* for Coordinate Set 1. It is estimated that the PSO-C algorithm will take 101.51 more milliseconds to complete its initial run than the PSO-MU algorithm will take. From this, it is inferred that the PSO-MU algorithm is superior than the standard PSO-W and PSO-C presentations in the context of VLSI global routing, while being slower and using a larger Timing budget than the PSO-C approach.

Table 3.4. Comparative studies of PSO variants over SD

Test case	PSO-W	PSO-C	PSO-MU
SET 1	7.77	0.71	5.65
SET 2	1.94	1.88	3.83

Standard deviation (SD) values are computed for all methods on both coordinate sets and published in Table 3.4 so that the consistency of these algorithms may be examined. The PSO-C SD value is calculated to be *'0.71'* and *'2.25'* for the two different coordinate systems, respectively. Despite the trade-off in system execution time, the technique is resistant against distribution difficulties of the search space in VLSI layout because these values are substantially lower than any other SD values of PSO-W and PSO-MU. Inferring from this, PSO-C demonstrates algorithm stability over all different distribution topologies of the terminal nodes in the aforementioned VLSI layout, while producing greater value of global routing interconnection cost and system execution time compared to PSO-W and PSO-MU.

3.4. Modified Constricted PSO Algorithm based Interconnect delay minimization of VLSI circuits using iterative RLC delay model

Submicron precision in the manufacturing process allows for greater resistance and a smaller wire diameter. Wire resistance must be accounted for since it significantly increases the connection latency. As a result, the lumped delay model has become defunct [3.7]. The RLC interconnect model of [3.8] is enhanced by the BPSO method suggested by Md. Yusof et al [3.9], which is shown to have practical applicability in S-concurrent RABILA's routing and buffer placement algorithm. The path in VLSI design represents the many connections between the source and the destination nodes. Although the whole-time lag from the origin to the destination is what is being dealt with here. The equation for path p is derived from all the segments or nodes as in (3.11)

$$p = (\sum_{n=1}^{m} |e_n - e_{n+2}|) + |x_2 - e_{m-1}| + |x_1 - e_1| \tag{3.11}$$

where, m = maximal dog-leg, e_n= the node's position in the grid graph, x_1, x_2 = abscissa of the source and the sink respectively (co-ordinates of the x-axis on the graph).

The work's anticipated algorithm, mutation-based constricted PSO (CPSO-MU), is used. The projection CPSO-MU position vector as defined by (3.10) is characterised by (3.5). The first stage entails locating the shortest augmented path. Buffer and wire obstacles serve as inputs, and the resulting enhanced path serves as an output. The second stage of the algorithm aims to locate the wires and buffers with the least amount of delay possible. The output is the augmented path with the wires and buffers in their proper locations, whereas the inputs are the augmented path, wire types, and buffer types.

3.4.1. Generation of shortest path modelling by CPSO-MU algorithm and combination of wire and buffer placement:

Each particle in a VLSI routing map has seven position vectors, labelled e_1 through e_7 in Figure 3.11. Where even represents a location in a network diagram evaluated along the $x - pivot$, at the beginning, and odd addresses a district in a matrix diagram evaluated along the $y - hub$, beginning at the beginning, the e_n in the lattice chart is referred to as the hub's area. If these consecutive nodes are connected, they will form a complete route from origin to destination. For the purposes of the PSO calculation, a route in this Grid graph can be thought of as a particle, s, and its state will be reset to its initial state after a certain number of cycles, as shown in the algorithm part.

Using the 8 dogleg, shown in Figure 3.11, the source-to-sink position vector (e_4 , e_3) will be arranged to handle the situation at point A, and the absolute steering region will look like this as in (3.12).

$$source \rightarrow e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7 \rightarrow sink \tag{3.12}$$

Besides, if planned in a row-column coordinate form, the spot of an entire path is:

$$(y_1, x_1) \to (y_1, e_1) \to (e_2, e_1) \to (e_2, e_3) \to (e_4, e_3) \to (e_4, e_5) \to (e_6, e_5) \to (e_6, e_7) \to (y_2, e_7) \to (y_2, x_2) \qquad (3.13)$$

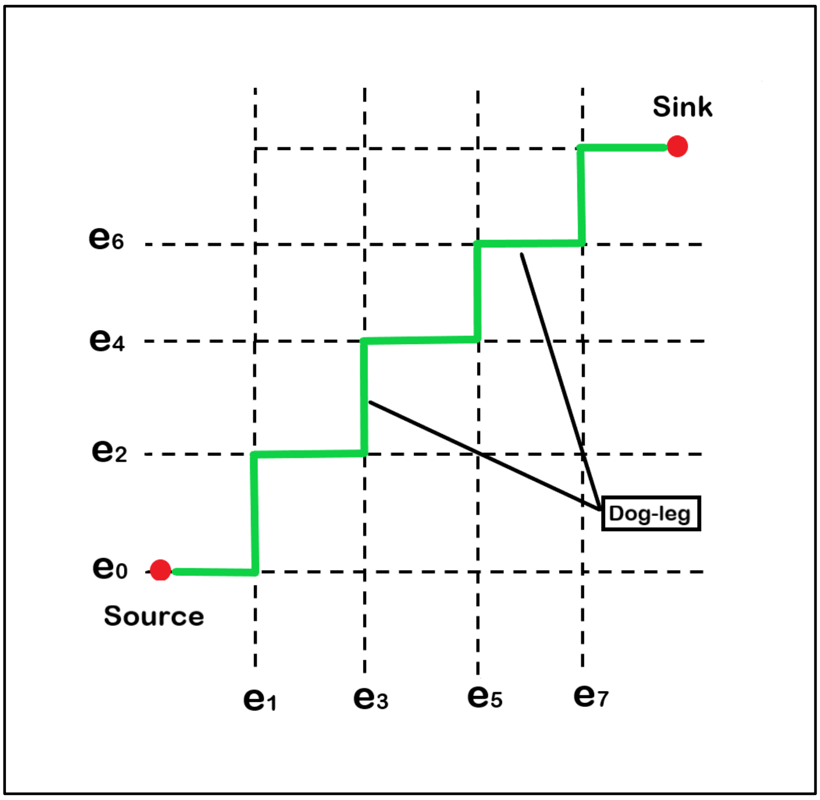

Figure 3.11. VLSI routing Map

Where $((y_1, x_1)$ and $((y_2, x_2)$ represent the source and sink facilities, respectively. It is plain to see that the VLSI lattice diagram's vector situation corresponds to the location of a particle, and that this vector is then applied to each of the particles' velocities. As a result, a particle I is presented with the given position and velocity (3.14).

$$s_i = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{bmatrix} \qquad v_i = \begin{bmatrix} v_{e1} \\ v_{e2} \\ v_{e3} \\ v_{e4} \\ v_{e5} \\ v_{e6} \\ v_{e7} \end{bmatrix} \qquad (3.14)$$

Particle fitness can be calculated under the following conditions generating path value as in (3.11). Next objective, on the generated path, is to find the most optimal spot along the wire to embed the buffer.

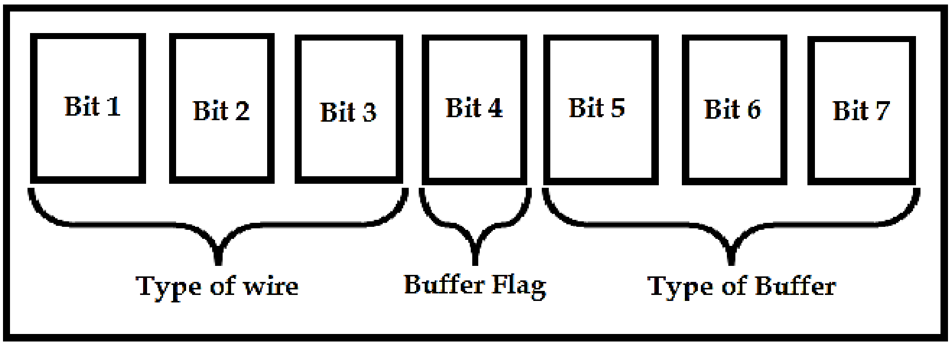

Figure 3.12. Representation for buffer placement in binary

In the next stage, CPSO-MU keeps looking for the best place along the wire to embed the buffer in order to acquire a negligible lag in VLSI guiding. In accordance with the framework,

$$s = [w_1\ w_2 w_3 \dots w_n]^T \tag{3.15}$$

where, w_n conveys the wiring type (whether a buffer is present) at a given location (m) along the length. For the case study, seven pieces were used for each w, three of which were used to buffer the different types of wire used, from the blend *000* as *Type-0* to the blend *111* as *Type-7*. The fourth bit indicates buffer use, and the remaining three bits designate the specific buffer types in play. A hindered graph in Figure 3.12 is used to illustrate this point. A buffer presence or absence can be indicated by the values *1* and *0*, respectively. Consequently, it is important to note that the last three bits of the buffer flag (*4th* piece) are disregarded regardless of their value if the buffer flag itself is *0*. This value indicates the types of buffers being used.

3.4.2. Algorithm of CPSO-MU:

Step 1. Grid size and Terminal nodes source, destination is distinct. Swarm size and Max-iterations are defined

Step 2. Generate an initial population of particles $X_i = \{X_1, X_2, \ldots., X_n\}$

Step 3. Calculation of $f(X_i)$ and $MIN(f(X_i))$

Until max iteration is reached step 4 to step 11 is continued

Step 4. Evaluate φ_1 and φ_2.

Step 5. Evaluate Constriction Factor (χ) according to (3.6)

Step 6. Set $p_{best=}$ $f(X_i)$ and g_{best}= $MIN(\, f(X_i))$

For all n particles

Step 7. Calculate V_{i+1} , particle velocity according to the velocity equation (3.5)

Step 8. Update the particle position as presented in (3.10)

Step 9. Evaluate $f(X_i)$ and $MIN\big(\, f(X_i)\big)$

Step 10. Update p_{best} and g_{best}

Step 11. Till n particles are taken into account start with step 8

Step 12. Increase the iteration counter

Step 13. Post processing the results and visualization

Step 14. Optimized result of $MIN\big(\, f(X_i)\big)$ or optimal VLSI routing path.

3.4.3. Experimental Procedure:

The direction of the presented algorithms is distinguished on a *32*-by-*32*-square benchmark test graph [3.69]. As shown in Figure 3.13, the test diagram depicts a VLSI circuit with a grid-graph, the location of the source and the sink, the area covered by wire deterrent, and the area covered by buffer hindrance. Table 3.5 depicts the boundaries related to the contextual investigation. The six different types of wires and cushions used in this case study are shown in Table 3.6 and Table 3.7, respectively.

The simulation was tested on a desktop computer with 2 GB of RAM and an Intel Core 2 Duo processor running at 1.8 GHz, and the code was written using MATLAB.

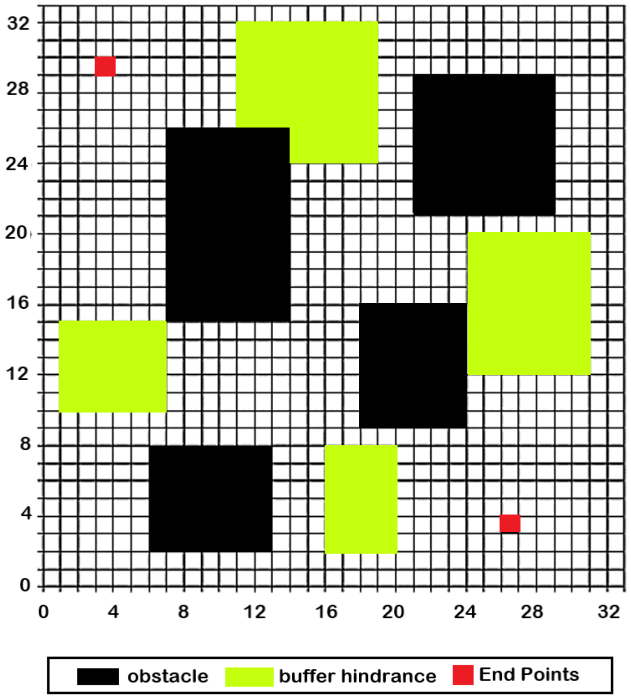

Figure 3.13. 32 × 32 Grid test graph with buffer hindrance and obstacles [3.69]

Table 3.5. Parameters in the case study

Parameter	Value
Source Location	(3,3)
Sink Location	(29,26)
Number of buffer	4
Number of obstacle	4
Maximum dog-leg used	8
Resistance at source	140 Ω
Load Capacitance	0.002 pF

Table 3.6. Wire Library Parameters

Category	Wire ($z = w$)					
Type	W_0	W_1	W_2	W_3	W_4	W_5
r_z (Ω)	58	2.037	1.358	44.9	22.45	11.22
c_z (pF)	42	28.69	36.67	16.67	21.87	31.84
l_b (nH)	0.106	0.106	0.106	0.106	0.106	0.106

Table 3.7. Parameters in Buffer Library

Category	**Buffer (b = w)**					
Type	**B_0**	**B_1**	**B_2**	**B_3**	**B_4**	**B_5**
r_z (Ω)	140	180	162	145.8	131.2	118.1
c_z (pF)	2	22	24.2	26.6	29.3	32.3
d_b (ps)	40	15	22.5	33.8	50.6	75.9

In contrast to the traditional method dependent on BPSO to limit the total wire-length of the VLSI circuit in the first step and VLSI interconnect delay in the subsequent improvement, the projected algorithm taking CPSO-MU into consideration expects to outfit ideal routing game plans with buffer expansion. In Table 3.8, the similarities and differences between the control boundaries of the traditional approach and the CPSO-MU are observed.

Table 3.8. Control Parameters for Algorithms Initialization

Parameters	**BPSO**	**CPSO-MU**
No. of Agents	50	50
Max Iterations	500	500
No. of Computations	10	10
Inertia weight 'w'	0.9 → 0.4	–
Cognitive component 'ϕ_1'	2	1.49 → 2.61
Social component 'ϕ_2'	2	2.61 → 1.49
Randomization factor 'ψ'	–	2 → 1.49
'r_1', 'r_2'	[0,1]	[0,1]
'r_3'	–	[0,1]

3.4.4. Experimental Results and Discussions:

Table 3.9 records the final results of the re-accreditation of the algorithms, showing that the predicted calculation successfully delivers the interconnect delay of '*371.44*' ps for the test graph, similar to the previous BPSO, while attaining global combination with the smallest amount of emphasis at '*343*,' which is endlessly improved when diverged from the earlier methodology. It also demonstrates that the presented CPSO-MU is consistent with the earlier BPSO in producing the best response for a general wire-length of *'49'* components, as shown in Figure 3.14, while using a relatively low number of cycles.

Table 3.9. Comparative studies on CPSO-MU with BPSO

Parameters	**BPSO [3.69]**	**CPSO-MU**
Overall wirelength/ Shortest path (in unit)	49	49
Least no. of iterations for Global Convergence	351	343
Average no. of iterations for Global Convergence	439.1	391.5
VLSI Interconnect Least Delay obtained	371.44 ps	371.44 ps
VLSI Interconnect Average Delay obtained	394.62 ps	391.73ps

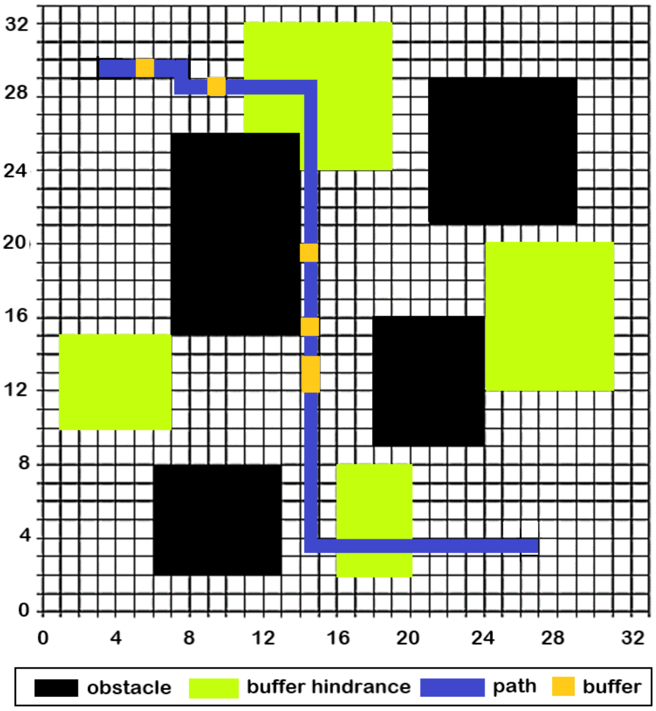

Figure 3.14. Placement of buffers with optimal path

Examinations of the current proving ground are carried out with the end goal in mind to check the consistency of these calculations to see how little they veer from their specific outclasses when running various occasions; this is inferred from the revolutionary idea of these algorithms that cause unmistakable expenses, i.e. least wire-length at different places and times. The average standard deviation from *30* iterations of each algorithm is shown in Table 3.10.

Table 3.10. SD comparison of BPSO and CPSO-MU

Standard Deviation	BPSO	CPSO-MU
VLSI Interconnect Delay	16.55	11.23
Iterations for Global convergence	52.33	39.5

As can be seen in Table 3.10, the presented SD of CPSO- MU is significantly lower than that of the earlier BPSO approach, guaranteeing more predictable execution, due to PSO's hand-tuned boundaries, which prevent it from wandering into neighbouring optimum solutions and thus slowing down the rate of combination. The transformation allows CPSO to provide formidable rival swarm particles to a state-of-the-art CPSO-MU, which in turn fights the BPSO algorithm and provides the shortest possible interconnect delay by finding the best route to address the buffer expansion problem in the shortest possible amount of time.

References

3.1 G. E. Moore, “Cramming more components onto integrated circuits,” *IEEE*, vol. 86, no. 1, pp. 82-85, January 1998.

3.2 H. Bollinger, “A mature DA system for PC layout,” *in Proceedings of first International Printed Circuit Conference, IEEE Computer Society Press*, Lo Alamitos, CA, 1979, pp. 85-99.

3.3 W. A. Dees, and P.G. Karger, “Automated rip-up and reroute techniques,” *in Proceedings of Design Automation Conference, IEEE Press*, Piscataway, NJ, 1982, pp 432-439.

3.4 J. M. Ho, G. Vijayan, and C. K. Wong, "New algorithms for the rectilinear Steiner tree problem, " *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 2, pp. 185-193, February 1990.

3.5 H. Zhou, D. Wong, I.-M. Liu, and A. Aziz, "Simultaneous routing and buffer insertion with restrictions on buffer locations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 7, pp. 819–824, July 2000.

3.6 W. C. Elmore, "The Transient Response of Damped Linear Networks," *Applied Physics,* vol. 19, pp. 55 - 63, January 1948.

3.7 Y. I. Ismail, E.G. Friedman, and J. L. Neves, "Equivalent Elmore delay for RLC trees," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol 19, no 1, 2000, pp 83-97, 1999

3.8 M. K. Hani, and N. S. Husin, "An Optimization Algorithm Based on Grid-Graphs for Minimizing Interconnect Delay in VLSI Layout Design, " *Malaysian Journal of Computer Science*, vol. 22, no. 1, pp. 19 – 33, 2009.

3.9 Z. M. Yusof, M. K. Hani, N. S. Husin, and M. N. Marsono, "Iterative RLC Models for Interconnect Delay Optimization in VLSI Routing Algorithm," *in Proceedings of Student Conference on Research and Development (SCOReD),* Johor, Malaysia, 2008, pp 270.1-4.

3.10 N. S. Husin, and M. K. Hani, "Optimal routing algorithm for minimizing interconnect delay in VLSI layout design," *in Proceedings of International Conference on Robotics, Vision, Information and Signal Processing (ROVISP),* 2007, pp. 345–349.

3.11 G. Beni, and J. Wang, "Swarm Intelligence in Cellular Robotic Systems, Robots and Biological Systems: Towards a New Bionics?," *in Proceedings of NATO ASI Series*, Springer, Berlin, Heidelberg, 1993, pp. 703-712.

3.12 R. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory," *in Proceedings of Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39-43.

3.13 Y. Shi, and R. C. Eberhart, "Empirical study of particle swarm optimization," *in Proceedings of the 1999 Congress on Evolutionary Computation*, Washington, DC, U.S.A., 1999, pp. 1945-1950.

3.14 M. Clerc, and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation,* vol 6, no. 6, pp. 58 – 73, February 2002.

3.15 F.K. Hwang, "On Steiner Minimal Trees with Rectilinear Distance," *SIAM Journal of Applied Mathematics,* vol. 30, no. 1, pp. 104-114, Jan. 1979.

3.16 M. Hanan, "On Steiner's problem with rectilinear distance," *SIAM Journal of Applied Mathematics*, vol. 14, no. 2, pp. 255-265, March 1966.

3.17 J. P. Cohoon, D. S. Richards, and J. S. Salowe, "A linear-time Steiner tree routing algorithm for terminals on the boundary of a rectangle," *in Proceedings of IEEE International Conference on Computer-Aided Design (ICCAD-89) Digest of Technical Papers*, Santa Clara, CA, USA, 1988, pp. 402-405.

3.18 J. M. Ho, G. Vijayan, and C. K. Wong, "A New Approach to the Rectilinear Steiner Tree Problem," *in Proceedings of 26th ACM/IEEE Design Automation*, Las Vegas, NV, USA, 1989, pp. 161-166.

3.19 C. T. Hai, and H. Y. Chin, "Rectilinear Steiner tree construction by local and global refinement," *in Proceedings of 1990 IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*, Santa Clara, CA, USA, 1994, pp. 432-435.

3.20 X. Hong, T. Xue, E. S. Kuh, C. K. Cheng, and J. Huang, "Performance-Driven Steiner Tree Algorithms for Global Routing," *in Proceedings of 30th ACM/IEEE Design Automation*, Dallas, TX, USA, 1993, pp. 177-181.

3.21 K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, "Rectilinear Steiner Trees with Minimum Elmore Delay," *in Proceedings of 31st Design Automation*, San Diego, CA, USA, 1994, pp. 381-386.

3.22 R. Novak, and J. Rugelj, "Distribution of constrained Steiner tree computation in shortest-delay networks," *in Proceedings of 8th Mediterranean Electrotechnical Conference on Industrial Applications in Power Systems, Computer Science and Telecommunications (MELECON 96)*, Bari, Italy, 1996, pp. 959-962.

3.23 S. Areibi, M. Xie, and A. Vannelli, "An efficient rectilinear Steiner tree algorithm for VLSI global routing," *in Proceedings of Canadian Conference on Electrical and Computer Engineering*, Toronto, Canada, 2001, pp. 1067-1072.

3.24 Y. Hu, T. Jing, X. Hong, Z. Feng, X. Hu, and G. Yan, "An efficient rectilinear Steiner minimum tree algorithm based on ant colony optimization," *in Proceedings of International Conference on Communications, Circuits and Systems*, Chengdu, China, 2004, pp. 1276-1280.

3.25 G. Grewal, and M. Xu, "An efficient graph-based steiner tree heuristic for the global routing of macro cells," *Canadian Journal of Electrical and Computer Engineering*, vol. 31, no. 4, pp. 211-219, December 2006.

3.26 C. Chu, and Y. Wong, "FLUTE: Fast Lookup Table Based Rectilinear Steiner Minimal Tree Algorithm for VLSI Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 70-83, Jan. 2008.

3.27 H. Totsukawa, H. Senou, and M. Ohmura, "A parallel genetic algorithm for 3-D rectilinear Steiner tree with bounded number of bends," *in Proceedings of 51st Midwest Symposium on Circuits and Systems*, Knoxville, TN, USA, 2008, pp. 89-92.

3.28 X. Ma, and Q. Liu, "A particle swarm optimization for steiner tree problem," *in Proceedings of* Sixth *International Conference on Natural Computation*, Yantai, China, 2010, pp. 2561-2565.

3.29 G. Liu, G. Chen, W. Guo, and Z. Chen, "DPSO-based Rectilinear Steiner Minimal Tree construction considering bend reduction," *in Proceedings of Seventh International Conference on Natural Computation*, Shanghai, China, 2011, pp. 1161-1165.

3.30 Y. Zhang, A. Chakraborty, S. Chowdhury, and D. Z. Pan, "Reclaiming over-the-IP-block routing resources with buffering-aware rectilinear Steiner minimum tree construction," *in Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, USA, 2012, pp. 137-143.

3.31 N. Maharjan, A. Shrestha, A. Tamrakar, and S. P. Panday, "Solving Minimum Steiner Tree based on behavior of ants," *in Proceedings of Third Asian Himalayas International Conference on Internet*, Kathmundu, Nepal, 2012, pp. 1-4.

3.32 N. R. Latha, and G. R. Prasad, "Wirelength and memory optimized rectilinear steiner minimum tree routing," *in Proceedings of 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, 2017, pp. 1493-1497.

3.33 G. Chen, and E. F. Y. Young, "SALT: Provably Good Routing Topology by a Novel Steiner Shallow-Light Tree Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 6, pp. 1217-1230, June 2020.

3.34 N. B. Singh, S. Deb, and S. K. Sarkar, "Swarm Approach towards Better Quantum Well Nanostructure Modelling," *in Proceedings* of *International Conference & Workshop on Emerging Trends in Technology*, Mumbai, India, 2010, pp. 667–669.

3.35 T. Chakrabarti, S. Manna, U. Sharma, T. Chakrabarti, and S. K. Sarkar, "Design of Intelligent Maximum Power Point Tracking (MPPT) technique based on Swarm Intelligence based Algorithms," *in Proceedings* of *International Conference on Power and Advanced Control Engineering (ICPACE),* Bengaluru, India, 2015, pp 173-177.

3.36 S. Deb, N. B. Sing, S. Sarkar, and S. K. Sarkar, "Particle Swarm Approach for Better Modelling of Quantum well Nano-structure," *in Proceedings* of *3rd National Conference on Recent Trends in Electronics & Communication*, Bangalore, India, 2009, pp 138-142.

3.37 T. Arora, and M. E. Moses, ''Ant Colony Optimization for power efficient routing in manhattan and non-manhattan VLSI architectures,'' *in Proceedings* of *IEEE Swarm Intell. Symp.*, Nashville, TN, USA, 2009, pp. 137-144.

3.38 T. Arora, and M. Moses, "Using ant colony optimization for routing in VLSI chips," *in Proceedings* of *AIP*, Nashville, TN, USA, 2009, pp. 145-156.

3.39 S. Manna, T. Chakrabarti, U. Sharma, and S. K. Sarkar, ''Efficient VLSI routing optimization employing discrete differential evolution technique,'' *in*

Proceedings of *IEEE 2nd Int. Conf. Recent Trends Inf. Syst. (ReTIS)*, Kolkata, India, 2015, pp. 461 464.

3.40 S. Manna, U. Sharma, T. Chakrabarti, and S. K. Sarkar, "Efficient VLSI routing Optimization employing Discrete Differential Evolution Technique," *in Proceedings of ReTIS*, Kolkata, India, 2015, pp. 461 - 464.

3.41 H. Wu, S. Xu, Z. Zhuang, and G. Liu, ''X-architecture Steiner minimal tree construction based on discrete differential evolution,'' *in Proceedings of ICNC-FSKD ICHSA*, 2019, pp. 433-442.

3.42 P. Bhattacharya, A. Khan, S. K. Sarkar, and S. Sarkar, ''An Artificial Bee Colony optimization based global routing technique,'' *in Proceedings of Int. Conf. Control, Instrum., Energy Commun. (CIEC)*, Calcutta, India, 2014, pp. 621-625.

3.43 H. Zhang, and D. Ye, ''An Artificial Bee Colony algorithm approach for routing in VLSI,'' *in Proceedings of* Int. *Conf. Swarm Intell. Springer*, Fuzhou, China, 2012, pp. 334-341.

3.44 A. Khan, P. Bhattacharya, and S. K. Sarkar, ''A swarm based global routing optimization scheme,'' *in Proceedings of* Int. *Conf. Adv. Electr. Eng. (ICAEE*), Vellore, India, 2014, pp. 1-4.

3.45 M. N. Ayob, F. Hassan, A. H. Ismail, H. H. Basri, M. S. Azmi, and A. F. Z. Abidin, ''A swarm algorithm approach for routing in VLSI,'' *in Proceedings of Int. Symp. Comput. Appl. Ind. Electron. (ISCAIE)*, 2012, pp. 43-47.

3.46 R. R. Sahoo, M. Singh, B. M. Sahoo, K. Majumder, S. Ray, and S. K. Sarkar, "A Light Weight Trust based Secure and Energy Efficient Clustering in Wireless Sensor Network: Honey Bee Mating Intelligence Approach", *in Proceedings of International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA*), Kalyani, India, 2013, pp. 515-523.

3.47 R. R. Sahoo, M. Singh, B. M. Sahoo, K. Majumder,S. Ray, and S. K. Sarkar, "A Light Weight Trust Based Secure and Energy Efficient Clustering in Wireless Sensor Network: Honey Bee Mating Intelligence Approach," *in Proceedings of International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA*), Kalyani, India, 2013, pp. 515-523.

3.48 R. R. Sahoo, and S. K. Sarkar, "A light weight Trust Based secure and Energy Efficient Clustering in wireless sensor Networks: Honey Bee Mating Intelligent Approach," *Procedia Technology*, vol 11, pp. 515-523, Dec. 2013.

3.49 G. Chen, W. Guo, and Y. Chen, ''A PSO-based intelligent decision algorithm for VLSI floor planning,'' *Soft Comput.*, vol. 14, no. 12, pp. 1329-1337, Oct. 2010.

3.50 H. Zhou, D. Wong, I. M. Liu, and A. Aziz, ''Simultaneous routing and buffer insertion with restrictions on buffer locations,'' *IEEE Trans. Computer Aided Design Integrated Circuits Syst.*, vol. 19, no. 7, pp. 819-824, Jul. 2000.

3.51 B. Kantha, and S. K. Sarkar, "Comparative Study Analysis of Particle Swarm Optimization and Genetic Algorithm for the Optimization of Parameter of MEMS based Micro-heater," *Computational and Theoretical Nanoscience*, ASP, vol. 12, no. 8, pp. 1-6, August 2015.

3.52 S. Deb, N. B. Singh, D. Das, and S. K. Sarkar, "Particle swarm approach for parameter optimization of quantum well nano structure," *Expert Systems with Applications Elsevier*, vol. 38, no. 10, pp. 12999-13004, September 2011.

3.53 N. B. Singh, S. Deb, A. Ghosh, S. K. Sarkar, and S. K. Sarkar, "Application of Particle swarm Optimization to predict optimized system parameters for Delta Doped GaAs quantum wells," *in Proceedings of Intl. Conf. on Contemporary Computing IC3* Springer CCIS, 2009, pp.349-357.

3.54 N. B. Singh, S. Deb, G. P. Mishra, S. K. Sarkar, and S. K. Sarkar, "Application of Particle Swarm Optimization Algorithm for better Nano-devices," *in Proceedings of 2nd Int. Conf. on Contemporary Computing (IC3)*, Noida, India, 2009, pp. 349–357.

3.55 A. Khan, P. Bhattacharya, and S. K. Sarkar, "A Swarm Based Global Routing Optimization Scheme," *in Proceedings of* International *Conference on Advances in Electrical Engineering (ICAEE)*, Vellore, India, 2014, pp. 189–197.

3.56 A. Khan, S. Laha, and S. K. Sarkar, "A Novel Particle Swarm Optimization approach for VLSI Routing," *in Proceedings of IEEE International Advance Computing Conf.*, Ghaziabad, India, 2013, pp 258-262.

3.57 G. Liu, G. Chen, W. Guo, and Z. Chen, ′′DPSO-based rectilinear Steiner minimal tree construction considering bend reduction,′′ *in Proceedings of 7th Int. Conf. Natural Comput.*, Shanghai, China, 2011, pp. 1161-1165.

3.58 S. K. Sarkar, A. Ghosh, M. A. Goutham, A. S. Rani, and D. Samanta, "An efficient technique of integrating parallel neural networks for faster and power efficient nanodevices for ultra-dense VLSI circuits," *in Proceedings of 14thInternational Workshop on the Physics of Semiconductor Devices (IWPSD)*, Mumbai, India, 2007, pp 232-235.

3.59 H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664-2677, Sep. 2017.

3.60 L. Zhao, and J. Wei, "A new bi-level PSO algorithm based on dynamic constraint processing and approximate navigation," *in Proceedings of IEEE Congr. Evol. Comput. (CEC)*, Wellington, New Zealand, 2019, pp. 1659-1663.

3.61 A. Carlisle, and G. Dozier, "An Off-The-Shelf PSO," *in Proceedings of the Workshop on Particle Swarm Optimization,* Indianapolis, IN, USA, 2001, pp. 1-6.

3.62 M. Clerc, and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58 - 73, February 2002.

3.63 Y. Shi, and R. C. Eberhart, "Empirical study of particle swarm optimization," *in Proceedings of* the *1999 Congress on Evolutionary Computation*, Washington, DC, USA, 1999, pp. 1945-1950.

3.64 Y. D. Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171 – 195, April 2008.

3.65 R. C. Eberhart, and Y. Shi, "Particle swarm optimization: developments, applications and resources," *in Proceedings of* the *Congress on Evolutionary Computation*, Seoul, South Korea, 2001, pp. 81-86.

3.66 A. R. Jordehi, and J. Jasni, "Parameter selection in particle swarm optimization: a survey," *Experimental and Theoretical Artificial Intelligence*, vol. 25, no. 4, pp. 527-542, Jun. 2013.

3.67 S. Ghosh, S. Nath, and S. K. Sarkar, "PSO Algorithm with Self Tuned Parameter for Efficient Routing in VLSI Design," *in Proceedings of International Conference on Futuristic Trends in Computing and Communication*, Chennai, Tamil Nadu, India, 2015, pp. 60-63.

3.68 R. Hassan, B. Cohanim, O. D. Weck, and G. A. Venter, "Comparison of Particle Swarm Optimization and the Genetic Algorithm," *in Proceedings of 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials,* Austin, Texax, 2015, pp. 1-13.

3.69 Z. Md Yusof, T. Z. Hong, A. F. Z. Abidin, Md N. A. Salam, A. Adam, K. Khalil, J. A. A. Mukred, N. Sk. Husin, and Z. Ibrahim, "A Two-Step Binary Particle Swarm Optimization Approach for Routing in VLSI with Iterative RLC Delay Model," *in Proceedings of Third International Conference on Computational Intelligence, Modelling & Simulation*, Langkawi, Malaysia, 2011, pp. 63-67.

CHAPTER 4

WIRELENGTH MINIMIZATION OF VLSI CIRCUITS BASED ON INVASIVE WEED OPTIMIZATION

4.1. Introduction

Integrated Circuit designing and modeling involves compressing and compiling a few hundred thousand transistors into a singular chip with the progress of the nanotechnology. As a result, processing speed has grown, but circuit complexity has increased in the routing of signals in the design of VLSI. Interconnection routing latency is directly proportional to the wire length, so its optimality is critical. Due to the complexity of the problem, an algorithm is typically used to determine the optimal routing path for minimizing wire length. An indicator of better circuit performance is shorter wires. As the number of components in an integrated circuit grows, the complexity of the transmission of internal signals between them typically requires a global routing phase before more fine-grained routing can be performed. The electronic circuits are routed globally as a result it partitions the regions of routing which leads to the optimization of the wire length. This wire length connects the various components without affecting the throughput and power processing of the IC and also reduces the total power leakage in the circuits. Graph theory has been extensively required for global routing because it graphically maps the links between different components and apply certain algorithms and multiple software to find the best wirelength from various VLSI circuits.

The difficulties of worldwide routing can be alleviated to some extent by employing a sequential approach, whereby VLSI nets are sequenced according to the importance of their components, and wherein improvement phases are implemented in working routers. 'Shive-aside' techniques [4.1] and concurrent ways are introduced where integer programming principles are done parallel. These principles have been tested with various limitations are used to improve the various rerouting methods which are globally routed after pulling interfering wires which [4.2]. Some of the efficiently globally working routers that provide great connectivity in electrical circuits include Box Router [4.3], Fast Route2.0 [4.4], NTHU-Route [4.5], and Maize Router [4.6].

The Steiner Tree problem is crucial in attaining the aforementioned goal because solving it costs less than solving the minimum spanning tree for the same graph, resulting in a shorter wire length of the circuit. The Problem of the Steiner Tree is a member of NP class because the given graph is optimized by selecting the Steiner nodes which are needed and the optimal placements could not result in polynomial time. Experts like S. Rajagopalan and Vazirani used Integer Linear Programming optimization to apply bi-directed cut relaxation [4.7]. An iterated *1*-Steiner approach concluded that the graph with n points has *n-2* Steiner points. From the start of developing and improving various algorithms the use of heuristics is very crucial to check for their productivity in providing the best solution for the Steiner Tree problem. However, these algorithms are held back in their productivity due to time and space complexity and various other factors. Simulated Annealing [4.8] has been successfully used to solve challenging combinatorial optimization problems, but the approach's feasibility gets shorthanded because the new graphs require very fine-tuned parameters which needs to be controlled to perfection. Genetic algorithm also gained importance in around 2003 [4.9]- [4.10], where researchers introduced this algorithm in wireless network and delay optimization in VLSI application.

Researchers subsequently shifted their focus on a new produce of meta-heuristics that perform better in given paradigms of Global VLSI routing optimization that are inspired by natural phenomena and behavior of live beings. The initialization of the algorithms starts with the primarily solutions which are random and are updated iteratively preceding one to converge toward optimal solutions. The optimization of ACO [4.11] is one such method that simulates the behavior of foraging food that is exhibited by colonial ants, which selects a path chosen in tandem with the majority of the population as the best path to food. A R Sardar et al. [4.12] introduced ACO for better quality of service in Mobile Adhoc network. A. Khan et al. [4.13] worked on routing optimization using Firefly algorithm which is bio-inspired algorithm imitate the flashing behavior of fireflies. Another ABC algorithm simulate the foraging behavior of honeybees is found to an effective optimization algorithm in routing optimization as reported by P. Bhattacharya et al. [4.14]. R.R. Sahoo et al. reported Honey bee mating intelligence approach in wireless Sensor network. [4.15]

In 2009, PSO reported by some researchers in VLSI [4.16]- [4.17]. A researcher named Dong et al. [4.18] put PSO in action for finding the solution of the Routing problem in VLSI system. PSO, a strong meta-heuristics algorithm, also uses approximation techniques to update the starting solution, simulating the communal behavior of the flocking of birds or the schooling of fish. Such algorithms have trouble selecting the best regulating parameters, necessitating manual tuning to obtain maximal optimization. A. Khan et. al. [4.19] reported use of this algorithm in VLSI optimization.

Khan et al. [4.20] in 2014 works on Swarm Intelligence drawing inspiration from the collective behaviors of insect swarms such as Firefly Algorithm, Artificial Bee colony and PSO in the realm of VLSI design optimization and made comparative studies.

In 2015, Liu et al. [4.21] introduces a particle swarm optimization-based algorithm to pioneers a novel approach for constructing obstacle-avoiding preferred direction

X-architecture Steiner trees, advancing chip design theory for non-Manhattan architectures.

G. Chen and E.F.Y. Young [4.22] in 2020 presents a novel Steiner Shallow-Light Tree (SALT) construction method for weighted undirected graphs, offering an efficient solution that closely approximates both the shortest distances from a root to other vertices and the minimum tree weight.

In 2020 Chen et al. [4.23] explores VLSI routing complexities and challenges, advocating for Swarm Intelligence (SI) techniques as effective solutions. It reviews five common SI methods applied to VLSI routing, addressing classical problems like Steiner tree construction and global routing

New and more advanced optimized algorithms are required to tackle the downside of the above discussed algorithms.

This chapter presents the detailed understanding of a newer Invasive Weed Optimization based meta-heuristic techniques which is similar to weed colonization behavior, the detailed study is given in the literature review section. Weeds use an intriguing adaptation to expand their colony: they remove weeds from their main population which are not fit because they have a heavy reproduce. The method employs spatial dispersion method for seeds across the field. Now with higher yielding weeds, the weeds have a better chance of surviving. This means that better techniques for optimizing the Steiner problem to minimize wire length in VLSI Global Routing circuits will eventually replace the current algorithms. A new hybridization method for PSO is being studied to increase global optimization by combining both algorithms' efficiency.

4.2. Literature Review

Mehrabian et al. [4.24] for the first time worked-on Colonization of invasive weeds from agriculture sector in the year 2006.

H.S. Rad. et al. [4.25] introduced IWO to a device an optimization algorithm for a recommender system domain which enhanced the quality of recommendation for gaining the effectiveness of the prioritized based user profile based on some characteristics like sex, age, rating. Clients' interests accomplished by considering these ratings in the profile matching stage.

M. R. Ghalenoei et al. [4.26] reported technique of DIWO technique from weed colonization. This technique used for UAVs (unmanned aerial vehicles) task assignments. DIWO performance was compared with time-cost with trade-off. Results obtained with simulations of Monte Carlo with better performance from DIWO and was having less computational time.

An algorithm of hybrid optimization that is based on IWO and PSO was reported by H. Hajimirsadeghi and C. Lucas [4.27]. Effectiveness of these algorithms checked for converging speed and getting optimal solution. Simulated results proved that the method is fast with optimal solution which can be used for global optimization method.

C. Veenhuis in 2010 [4.28] worked on concept of binary IWO. BinIWO considers weeds and seeds as bitstrings. Reproduction process determines the young weeds in a precise normal distribution in the place of bitstrings. The normal distribution process is taking place at bit numbers where the reported technique works with less difference and bits.

M. Ahangaran et al. [4.29] introduced a system applied with the IWO to make a decent harmony among broadening and escalation parts of the calculation during all ages. In the first IWO, the standard deviation of typical arbitrary capacity is equivalent for all weeds in the settlement. In the method another capacity was applied to allot distinctive standard deviation to each weed in the settlement in every age. Weeds with better wellness gain lower standard deviation to circulate their seeds in area of themselves as well as the other way around weeds with more regrettable wellness get a better-quality deviation to spread their weeds far away from their present positions.

I.D. Falco et al. [4.30] reported on a migration model called biological invasion method which is executed through a multistage interaction including attacking subpopulations and their opposition with local people. Such an overall methodology is utilized inside a venturing stone model taking on differential Evolution as the neighbourhood calculation. The resultant algorithm is assessed on a wide arrangement of traditional test capacities against an enormous number of successive and other disseminated forms of Differential Evolution accessible in writing. The discoveries show that, in a large portion of the cases, the method can accomplish better execution as far as both arrangement quality and combination rate.

Another group of researchers K. Suresh et al. [4.31] worked on extended IWO. This extended IWO used to diminish the seed populations variance. It utilizes the idea of fuzzy logic for picking the best greatest number of populations individuals. The execution of the multi-objective IWO is differentiated over a test-suite containing seven unconstrained (bound obliged) and five general constrained multi-objective issues taken from the IEEE CEC 2009 contention and one of a kind gathering on multi-objective smoothing out estimations. IWO can appear as a very uplifting candidate metaheuristic in the space of multi-objective headway.

A hybrid with optimal solution by joining the distinction vector-based change plan of DE into the fundamental construction of an environmentally motivated metaheuristic IWO was reported by Maity et al. [4.32] for settling a genuine boundary and single-objective improvement issues. The collaboration of DE and IWO is made in such a manner, so as not to force any computational weight on the principal calculation. Through insightful and observational investigations, it was represented that the subsequent DIWO calculation might have a higher populace change over ages than both of its predecessors – DE and IWO.

C Sur et al. [4.33] in 2013 broadened the IWO utility combinatorial enhancement for way search and anticipating vehicle directing from a source to objective in a graph. The issue can be considered to be a multimodal improvement issue where determination of a specific arrangement of multimodal arrangements would be best arrangement. The traditional IWO is altered to suit the chart-based circumstance and rolled out fundamental improvement in suggestions to adapt up to the diagram boundaries. The intermingling pace of the Discrete Invasive Weed Optimization (DIWO) Algorithm is being contrasted on a street diagram model for course enhancement for vehicles regarding multi-objective of voyaging.

A hybrid IWO-PSO algorithm for pattern synthesis of conformal phased arrays was reported by Y. Bai et al. [4.34]. The interaction between the dispersion method of the IWO algorithm and the evolution velocity of the PSO algorithm is studied. In addition, the exploitation and exploration balancing respectively in the local and global of the hybrid IWO-PSO algorithm is shown. As an example, a 3 × 9 cylindrical conformal array is manufactured to extract the patterns and investigated to confirm the suggested algorithm.

Another group of researchers D. Das et al. [4.35] insinuated multi-objective mixture based on IWO and PSO. This is based on Soft decision fusion approach for streamlining the global decision and weight coefficient vector allocated to each cognitive users to augment the identification likelihood and large likelihood of error. And it also reduces the false alarm occurrence. The strategy beats the nondominated sorting genetic algorithm, MOPSO (multi-objective particle swarm optimization) and nondominated sorting IWO in the terms of location exactness and nondominated arrangements.

S. K. Mahto and A. Choubey [4.36] in 2016 worked on algorithm that is executed to examine the array by considering the array position. Pattern synthesis have been used here. A Precise strategy is utilized to detail the objective of the function. Three instances of example combination are considered to delineate the adequacy of the algorithm. The simulated results show the further developed presentation of cross breed IWO.

Y. R. Naidu and A. K. Ojha [4.37] reported a model of IWO that is multi-working. In the model, multiple populations are carried out to handle multiple objectives. IWO along with STS (space transformation search) are combined, where the algorithm run for each population to optimize the corresponding objective. In addition, an archive and local search are also integrated to save all the nondominated solutions, and to improve the archive solutions, respectively.

X. Yue. et al. [4.38] suggested an algorithm with IWGOA (Invasive weed grasshopper). The IWO strategy alongside irregular walk system is implanted to the computation. The system is suggested to control the developments of the grasshoppers in a superior manner. In the IWGOA calculation, the positions refreshing sorts and steps are affected by the iterative numbers and objective capacity esteems.

Toso et al. [4.39] presents a synthesis method for designing large planar array antennas with phase-only control. It utilizes Zernike polynomials as global basis functions for phase, reducing optimization variables compared to the array size. Invasive weed optimization (IWO) optimizes polynomial coefficients to address nonlinearity and local trapping issues common in phase-only problems.

Further, Pradhan et al. [4.40] introduces a hybrid MPPT algorithm, combining modified IWO and PSO techniques, to enhance photovoltaic(PV) power generation under changing conditions. It focuses on a standalone PV-based hybrid system, incorporating PV array, battery, electrolyser, fuel cell, and load, with a coordinated power management strategy.

4.3. Wirelength Minimization of VLSI Circuits using Invasive Weed Optimization Algorithm

MRST problem is solved using IWO feature. Simple IWO does not provide as good results as the above developed approach which is derived from IWO. This Enhanced IWO (EIWO) helps to disperse the seeds in a better process. This approach or EIWO minimizes the problems of Steiner explosion when the Steiner points are placed in the MRST. As we have competitive exclusion to remove fewer fit weeds from the population similarly EIWO sets up the tree in such a way that we only have the best position of the nodes or Steiner vertices so as to achieve the lowest wirelength. Figure 4.1 depicts the flowchart showing it.

The weeds of the present population in every iteration are synonymous to the vertices of the VLSI chip. Similarly, in the Hanan grid the field the weed population farm is synonymous to the VLSI layout of the grid. The goal of this approach is to minimize $\sum cost\ (e)\ x_e \forall\ e\ \in\ E$. In this case, $x_e\ \in\ \{0,1\}$, where E is the edge set of rectilinear graphs $G\ (V, E)$. At the point in the VLSI chip where the wirelength is shortest, costs are at their lowest.

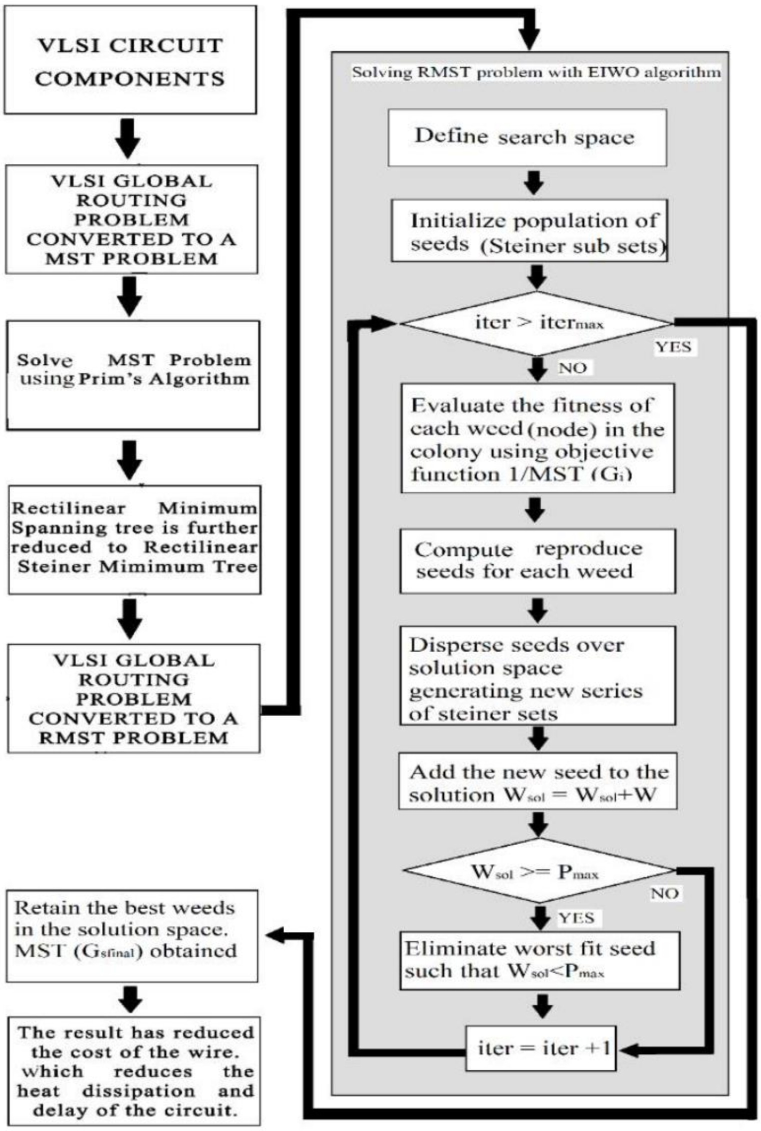
VLSI CIRCUIT COMPONENTS
VLSI GLOBAL ROUTING PROBLEM CONVERTED TO A MST PROBLEM
Solve MST Problem using Prim's Algorithm
Rectilinear Minimum Spanning tree is further reduced to Rectilinear Steiner Mimimum Tree
VLSI GLOBAL ROUTING PROBLEM CONVERTED TO A RMST PROBLEM
Retain the best weeds in the solution space. MST (Gsfinal) obtained
The result has reduced the cost of the wire. which reduces the heat dissipation and delay of the circuit.
Solving RMST problem with EIWO algorithm
Define search space
Initialize population of seeds (Steiner sub sets)
iter > itermax
YES
NO
Evaluate the fitness of each weed (node) in the colony using objective function 1/MST (Gi)
Compute reproduce seeds for each weed
Disperse seeds over solution space generating new series of steiner sets
Add the new seed to the solution Wsol = Wsol+W
Wsol >= Pmax
NO
YES
Eliminate worst fit seed such that Wsol<Pmax
iter = iter +1

Figure 4.1. Diagrammatic representation of the procedure and logic flow.

4.3.1. Algorithm based on IWO

Step 1 $n(R)$ = The algorithm begins with a set number of terminal points, denoted by p. The start and end value for p is predefined. These two points demarcate the specific space around the plant where seeds will be distributed. The maximum number of weeds allowed in the colony is denoted by p_{max} .

Step 2 A steiner set is created due to Step 1 which is filled with $(N^2 - p)$ number of points. The search space created with these points have $N \times N$ dimensions. This space has x number of seeds, with every moving iteration $p - 2$ steiner points are created in this space randomly and each p number of weeds has equal fitness.

Step 3 Steiner sub-sets are generated at random and used as x seeds, $S_i \subseteq S$, where $i = \{1, 2, 3, \dots, ,, x\}$. The terminal points synonymous to each of the seeds are uniformly and spatially distributed. This distribution follows the below probability distribution with (4.1).

$$f(x) = \{\frac{1}{b-a} \ \forall\, a \leq x \leq b \qquad 0 \ \ \forall\, x < a \; and \; x > b \tag{4.1}$$

Step 4 For the x number of seeds initially chosen for germination the fitness is w_i for each weed is calculated using function $1/\ MST\ (G_i)$.

Step 5 The condition for iteration is set.

Step 6 Seed production for the chosen weeds is optimized according to their reproductive fitness using the IWO algorithm, with the formula $S_{plant} = S_{min} + \ ceil\left[f_{plant} \times \frac{S_{max} - S_{min}}{f_{max} - f_{min}}\right]$ as in (2.24) , where S_{min} and S_{max} are the ranges of possible seed yields, f_{min} and f_{max} are the minimum and maximum objective function values for the current iteration, and S_{plant} is the target seed yield for the current plant having an objective function value of f_{plant} .

Step 7 As stated by the normal distribution function, $Y = f\left(\frac{x}{\mu,\sigma}\right) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}$ as in (2.25) where the mean position is set to the location of the parent weed, these new seeds are dispersed along with their parent weeds. It can be calculated where the replicated seeds should be dispersed as follows:

$$S_x = x(w_i) + n\sigma \tag{4.2}$$

In this case, $n \in R$, and $\mu_x = x(w_i)$ and $\mu_y = y(w_i)$. It is this function that determines the new Steiner sub-set by modifying the coordinates of the Steiner components as given in (4.3).

$$S_{new} = \{x(w_i) + n_1\sigma,\ x(w_i) + n_2\sigma, \dots + x(w_i) + n_{p-2}\sigma\} \tag{4.3}$$

where $S_{new} \subseteq S$.

Step 8 By dividing the mass of a weed by its mean squared temperature, or $1/MST\ (G_{Snew})$, where G_{Snew} is the modified graph, one can determine the probability that a given weed will produce viable offspring.

Step 9 The new seeds are combined with the existing W_{sol} solutions and the seeds are ranked according to their fitness.

Step 10 To get rid of all of the undesirable seeds (those with a high MST), it is advised to use Peirce's method [4.41] in the competitive exclusion procedure (G_{Snew}) . Once the population reaches or equals the colony's maximum permissible weeds, $W_{sol} + W \geq p_{max}$, the fitness mean (W_m) for all weeds is determined. As implemented in, an weed whose viability digresses from the point F_m by more or less than the accepted value (r_a) is deleted from the solution space (4.4).

$$R = \frac{|W_i - W_m|}{r_a} \tag{4.4}$$

The weed with fitness W_i is eradicated if $R > 1$. As a result, the algorithm's permitted range (r_a) is a required terminating parameter. Its tweaking is simple in order to ensure the presented algorithm's convergence.

Step 11 The dispersed seeds germinate close to the plant which reproduced them. This is mathematically proven by the value of standard deviation which deviates from the initial to final values w.r.t (4.5) as in (2.26).

$$\sigma_{iter} = \left(\frac{iter_{max} - iter}{iter_{max}}\right)^n \left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \tag{4.5}$$

With all changing iteration unfit weeds are disposed and only viable weeds are retained in the solution space.

Step 12 In the event that the terminating condition is met, the Steiner set $\mathrm{S}_{\mathrm{final}}$ representing the nearest perfect solution is obtained, and the minimum length wire solution MST (G_{Sfinal}) is obtained.

4.3.2. IWO-PSO Hybrid Algorithm

The PSO removes the nonviable assets from the starting population so PSO is integrated with the thought about EIWO to generate a hybrid algorithm. The goal is to offer a collection of better Steiner locations to the EIWO algorithm than random creation. PSO is run on the same set of terminals to retrieve the Steiner points. The candidate swarms (Steiner matrices) that make up EIWO's initial population are fine-tuned in this step.

VLSI layout on a two-dimensional Hanan grid $G\ (V\ , E)$ simulates the problem space with p terminal vertices.

Step 1 At the start z particles are assimilated randomly to form the swarm (4.1). This population is put in the solution space of $n \times n$ dimensional grid. This population has $p - 2$ Steiner points randomly collected from the set S with (n^2-p) points resulting in the formation of Steiner subset $Q_j \subseteq \mathrm{S}$, where $j = \{1,2,3, \ldots\ldots, z\}$

Step 2 The viability of each asset or seed F_i is measured with the function $1/MST(G_i\ )$ and also $1/MIN(MST(G_i))$ is also measured.

Step 3 The values of c_1 and c_2 is set to 2, $p_{best\ i} = F_i$ and $g_{best} = MAX\ (F_i)$ and the iteration is set to k times.

Step 4 Using weighted PSO velocity equation (2.17) and PSO position equation (2.16), the inertia weight w is measured and is incorporated to the position and velocity of each asset.

Step 5 To reevaluate the p_{best} and g_{best} . Afterk iterations, swarms with high fitness are selected, and a new modified set X is formed from Q_j, where $X \in S$.

Step 6 The initial population starts with p number of weeds. These number of weeds are synonymous to the terminal vertices in the EIWO algorithm. Two things are set in the initial iteration the viability of the weeds is set to the same value and number of seeds reproduced is x set with the PSO algorithm. Every seed has a fixed number of Steiner points associated to it which is $p - 2$. They are also associated with a randomly generated Steiner sub-set, S_i

where $S_i \subseteq X$, where $i = \{1,2,3, \ldots ., x\}$. The probability distribution function uniformly distributes the associated steiner points for every seed as $Y = f\left(\frac{x}{\mu, \sigma}\right) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}$.

Step 7 Starting at stage 4 of the presented EIWO method and holding the values of $\sigma_{initial}$ and σ_{final} constant, this algorithm performs a series of iterations.

4.3.3. Experimental Procedure

The experiment's test space is generated by randomly creating value sets of $10, 20, 50, 100$ and 500 terminal nodes in a grid of 1000 by 1000 nodes. The Intel i7 quad processor running at 2.2 GHz, along with 8 GB of RAM and a turbo boost, was used to run all the experiments. The optimizer algorithms listed in Table 4.1 were implemented after the governing conditions were established.

Table 4.1. Algorithm initialization with control parameters

Parameters	PSO-W	EIWO	EIWO-PSO
iter$_{max}$	75	75	75
c$_1$	2		2
c$_2$	2		2
w	0.05		0.05
swarms (z)	150		150
max_weed		150	150
$\sigma_{initial}$		1	1
σ_{final}		0.01	0.01
n (non-linear modulation index)		3	3

4.3.4. Result and Discussions:

In this results and discussions sub-section, a detailed insight of performance EIWO algorithms and EIWO-PSO hybrid algorithm is presented. According to Table 4.2, the EIWO and EIWO-PSO methods outperform the PSO-W procedure when applied to the Minimum Spanning Tree metric. The results reveal that the MRST created by the presented algorithm EIWO outperforms the MST method when juxtaposed to the PSO-W algorithm. On comparison to the other algorithms, EIWO-PSO hybrid provides the least valued MRST cost which is independent to the set of values in the experiment space. The output illustrates how IWO's features of guaranteeing the existence of the most viable assets helps in lifting the standard of optimization and in turn reducing the wire length of related nodes, in contrast to the PSO-W which relies on the probability function there is no room for.

Table 4.2. Comparison on Wirelength cost (MRST) vs MST

NODES	MST (in unit)	MRST (in unit)		
		PSO-W	EIWO	EIWO-PSO Hybrid
10	2219	2198	2198	2198
20	4034	3815	3765	3701
50	5988	5716	5606	5549
100	8766	7942	7889	7830

The results of EIWO with a specific wire length of ‘3765’ are depicted in Figures 4.2, while the results of EIWO-PSO with a specific wire length of ‘3701’ are depicted in Figure 4.3. Both examples use 20 terminal nodes. The minimum routing wirelength (MRST value) for 50 terminal nodes generated by EIWO and EIWO-PSO hybrid are depicted in Figures 4.4 and Figures 4.5, respectively. This exhibits that hybrid model of EIWO-PSO outperforms EIWO and PSO-W as individuals.

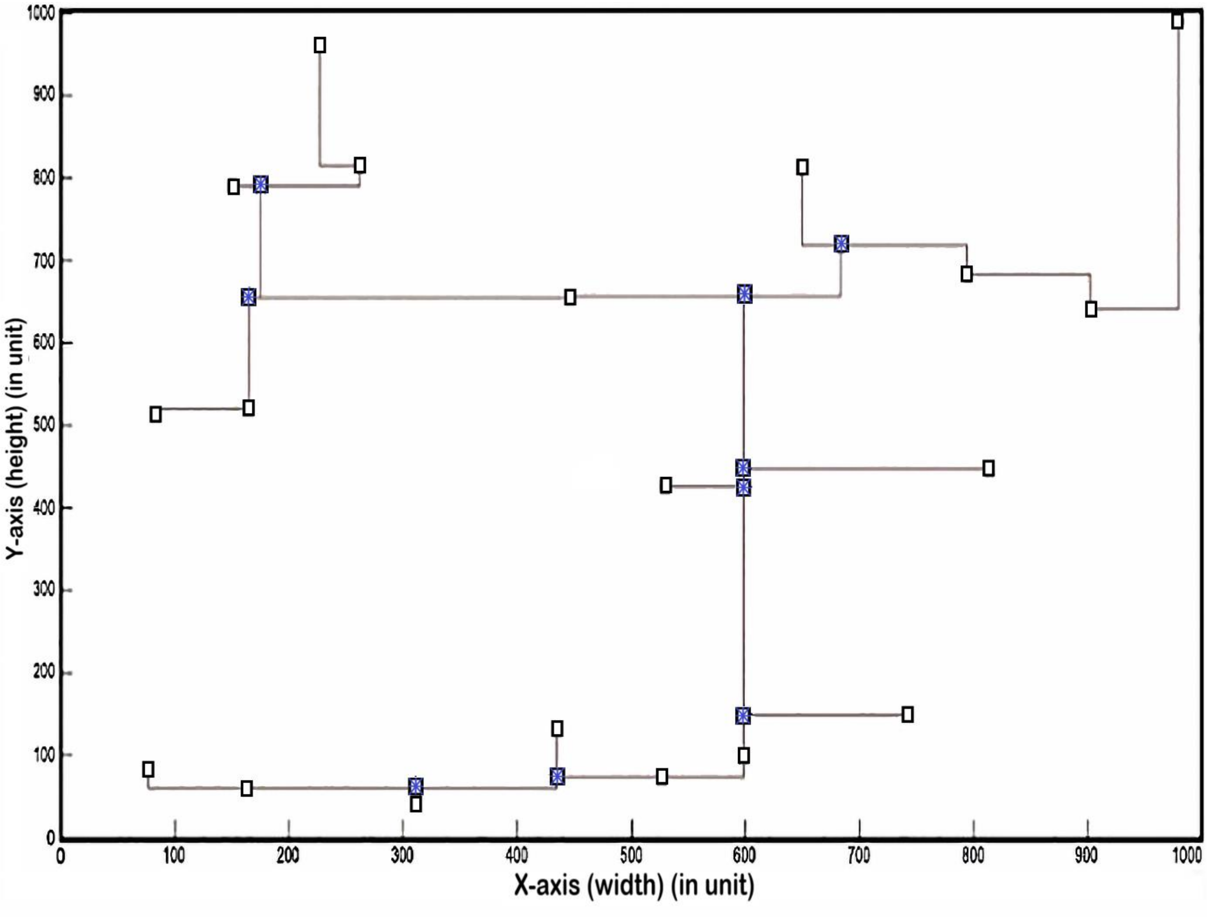

Figure 4.2. 20-node MRST using EIWO

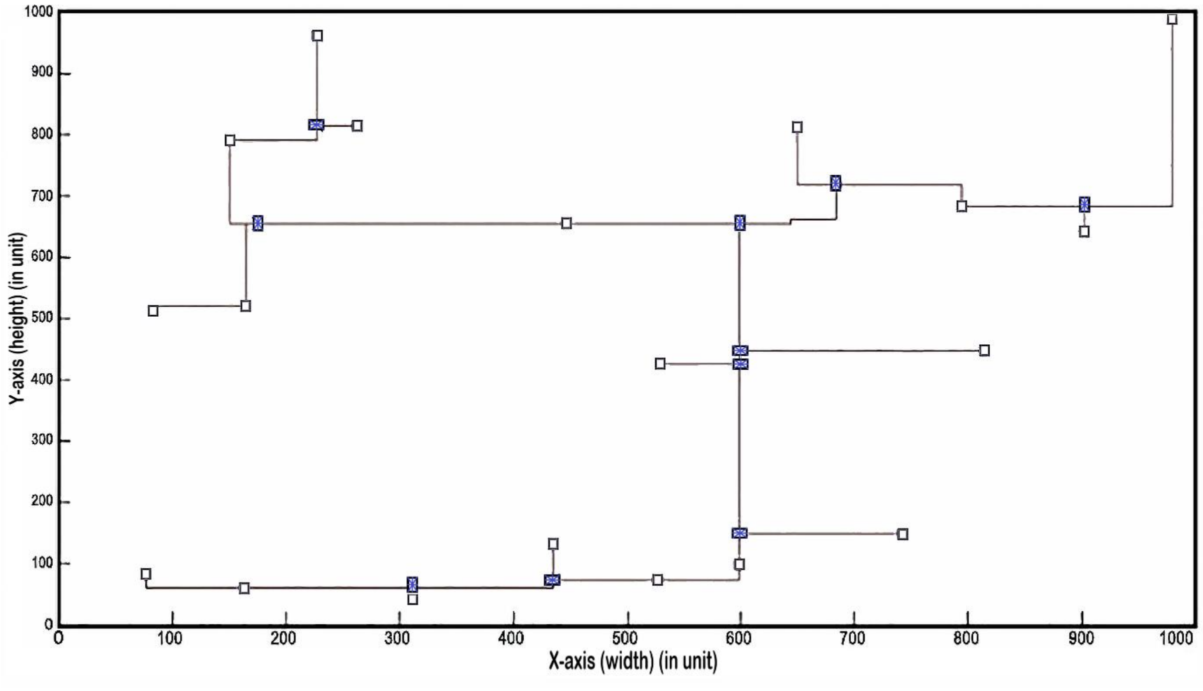

Figure 4.3. A 20-node MRST using EIWO-PSO

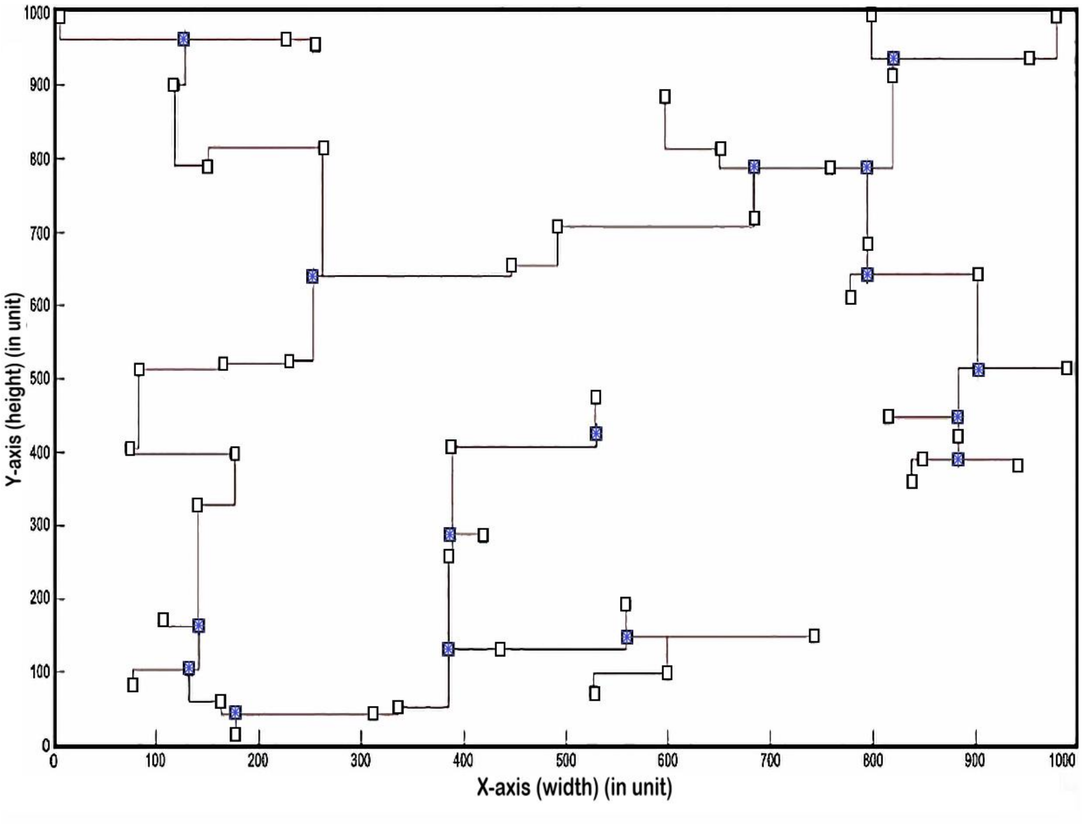

Figure 4.4. 50-node MRST achieved with EIWO at a value of '5606'

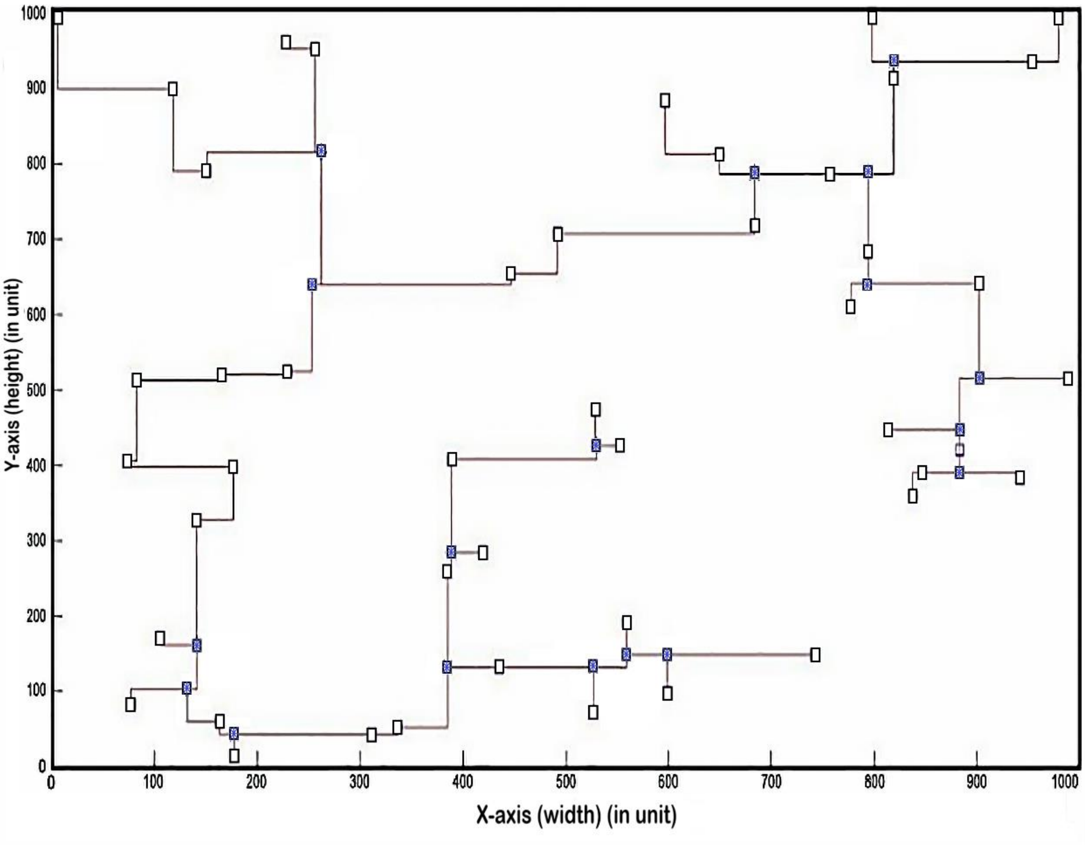

Figure 4.5. 50-node MRST achieved with EIWO-PSO Hybrid at a value of '5549'

The average system run time (in secs) of the aforementioned algorithms as measured through experimental iteration is displayed in Table 4.3. The results show that PSO-W outperforms EIWO and EIWO-PSO because it eliminates less desirable group members with no negative side effects, whereas EIWO and EIWO-PSO require more time to run due to the use of subroutines.

Table 4.3. Standardized CPU execution time for the algorithms

NODES	PSO-W	EIWO	EIWO-PSO Hybrid
10	4.17	4.26	9.14
20	9.27	7.94	19.03
50	27.2	32.32	78.78
100	93.7	91.4	201.71

Although EIWO-PSO is a difficult procedure, the blended difficulties of PSO-W and EIWO leads to a higher time complexity, as shown in Table 4.3 EIWO-PSO produces the least possible wire-length of the interconnected end nodes.

Table 4.4 compares the suggested approach to the Geosteiner-5.0.1 benchmark [4.42], which is similar to MRST algorithm with a higher space complexity. It shows that the suggested EIWO algorithm and the EIWO-PSO Hybrid algorithm, performs identically to the benchmark algorithm for 10 terminal nodes, with the lowest minimal wire length cost of '2198'. For this value sets with more terminal nodes, EIWO-PSO Hybrid is found to be the best, whereas PSO-W is found to be the worst, with the most divergence from the benchmark value. As a result, the RMST problem of graphs and, as a result, the wire-length of connecting terminal nodes can be efficiently handled. Therefore, it is reasonable to assume that the presented algorithms can efficiently handle the MRST problem of graphs, significantly shortening the wire-length between the connecting terminal nodes.

Table 4.4. MRST comparison against Geosteiner-5.0.1

NODES	PSO-W (in unit)	EIWO (in unit)	EIWO-PSO Hybrid (in unit)	Geosteiner-5.0.1 (in unit)
10	2198	2198	2198	2198
20	3815	3765	3721	3642
50	5716	5606	5549	5365
100	7942	7899	7830	7515

The standard deviations obtained as the output after performing every method 30 times are shown in Table 4.5 to ensure consistency. This data shows that the presented EIWO provides more stable performance than the PSO-W and hybrid EIWO-PSO algorithms across all value groups in the experimental setting.

Table 4.5. Estimates of the MRST's Standard Deviation

	NODES	PSO-W	EIWO	EIWO-PSO Hybrid
SD	10	5.34	0.84	1.05
	20	25.48	14.07	16.89
	50	23.07	11.619	15.837
	100	32.59	19.35	20.27

Evolutionary algorithms must converge or stabilize unstable populations. From Figure 4.6, it is obvious that as the population grows, the MRST costs of both algorithms reduce, which are initially steep but gradually become less steep over time, and eventually begin to get straight. It demonstrates that at a certain rise in population, they stabilize and achieve ideal results. The EIWO-PSO hybrid method has a greater efficiency but a slower convergence rate.

As seen in Figure 4.7, the decline in EIWO is quite steep as the number of iterations increases, but it flattens out at a particular point. This particular point is the convergence point where the algorithms best case performance at a fixed time constrain can be found. The EIWO-PSO combined procedure starts with a steep descent, however there is an area of very slow and almost no descent before converging and smoothing out, which is owing to the method's borderline inconsistency. Table 4.6 depicts that for large instance data sets, EIWO as well as EIWO-PSO algorithms have better efficiency than PWO-W.

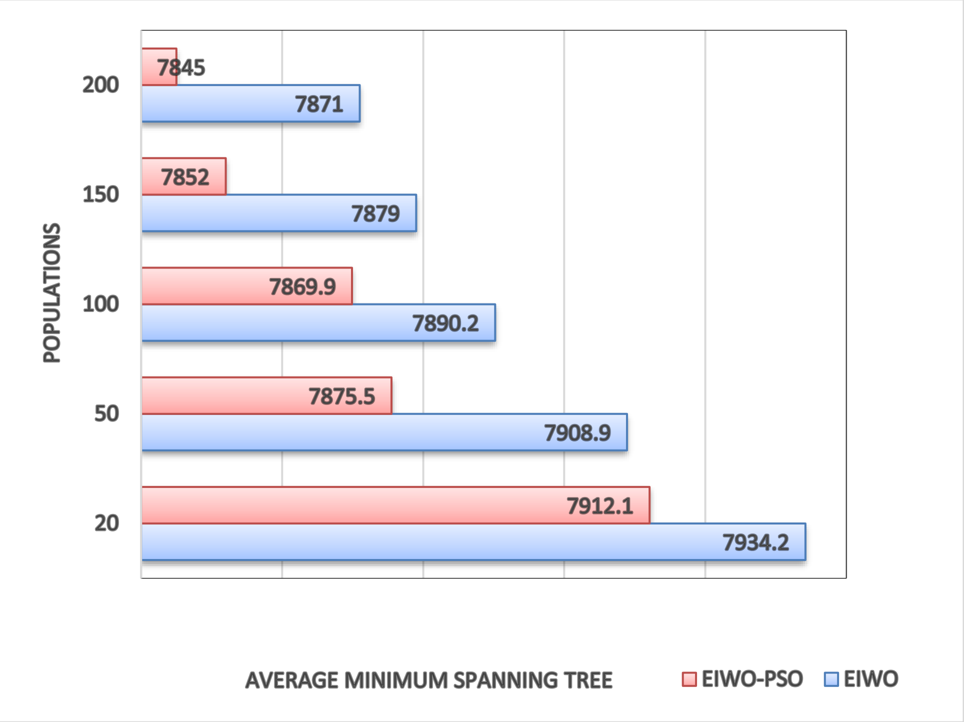

Figure 4.6. EIWO vs EIWO-PSO Hybrid population growth

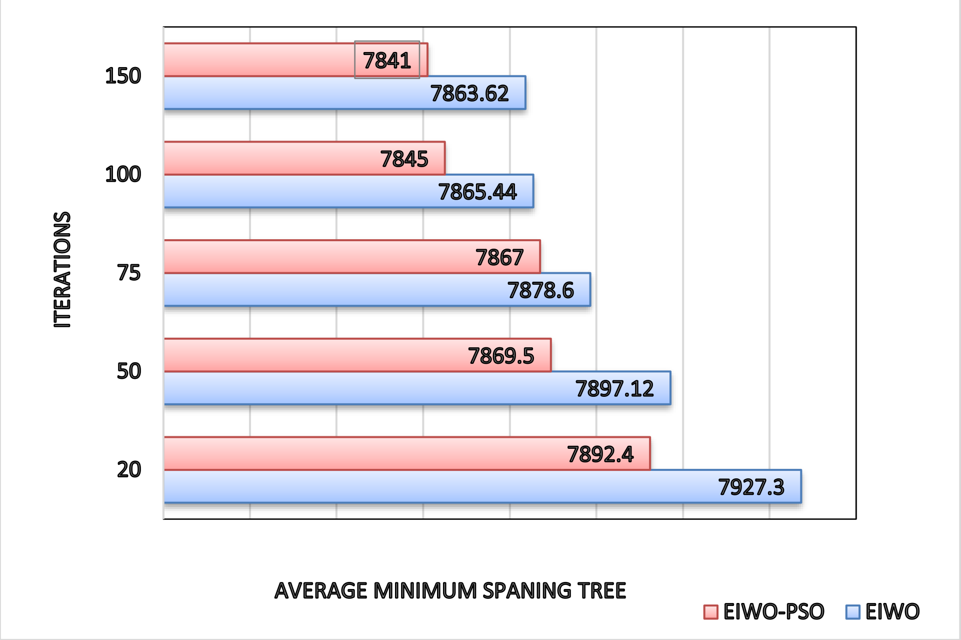

Figure 4.7 EIWO-PSO Hybrid vs. increment iterations

Figure 4.8 shows that EIWO-PSO hybrid reduces wire length cost to '18014' for *500* terminal nodes compared to '*18405*'. Figure 4.9 shows *1000* end nodes EIWO-generated MRST value of '*25652*'.

Table 4.6. Comparing large VLSI cases

NODES	MST (in unit)	MRST (in unit)		
		PSO-W	EIWO	EIWO-PSO Hybrid
500	18435	18405	18278	18014
1000	25798	25787	25652	25395

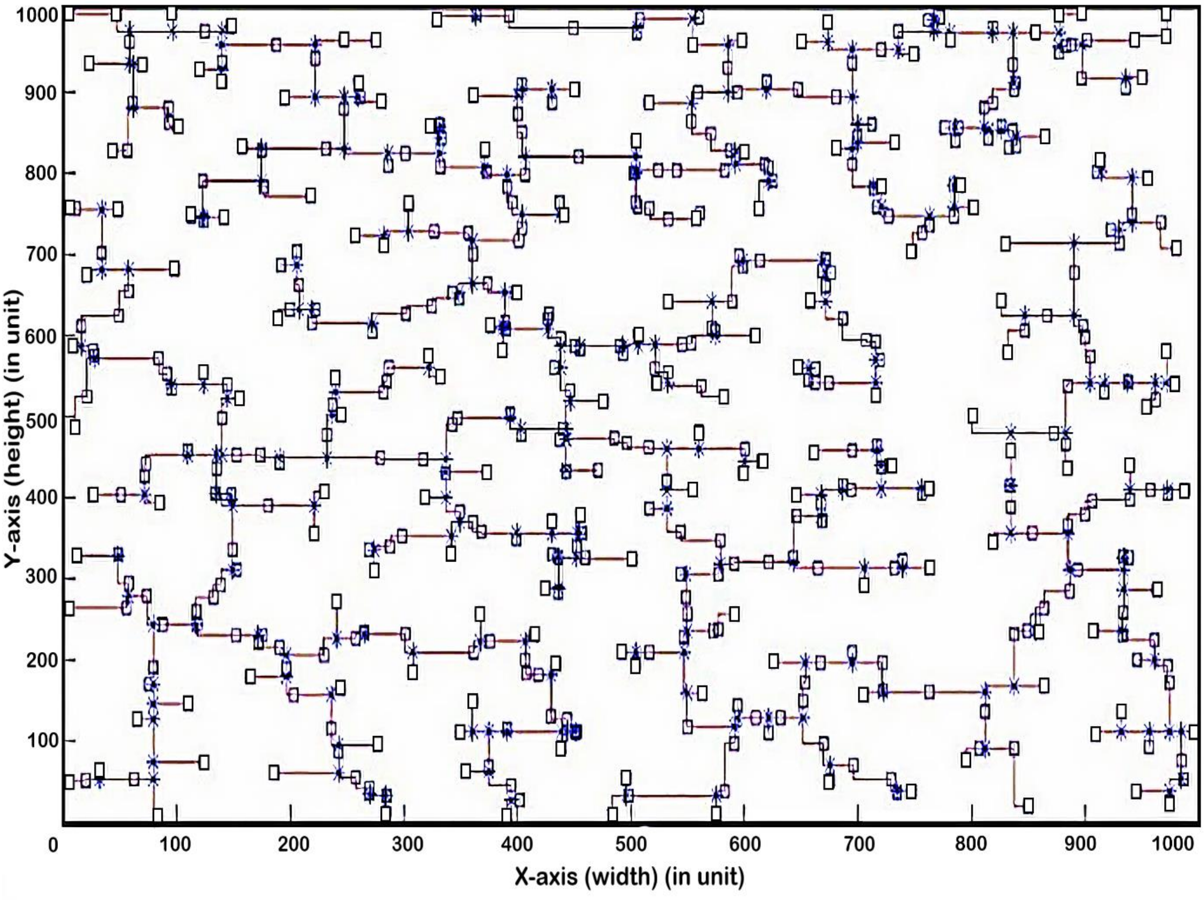

Figure 4.8. 500 terminal node EIWO-PSO hybrid result

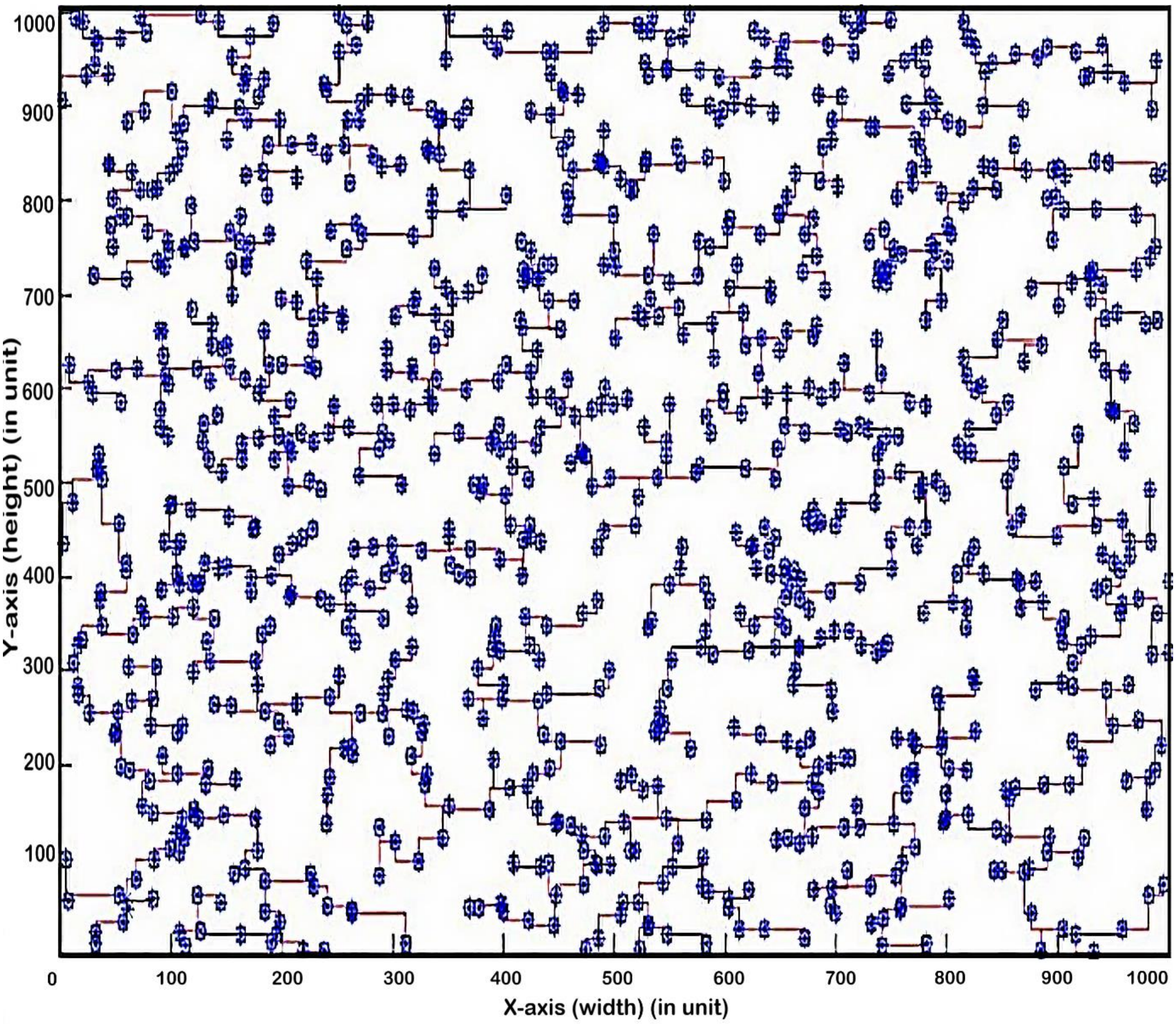

Figure 4.9. 1000 terminal node EIWO result.

Table 4.7: EIWO-PSO vs. ISPD'98 benchmark.

Benchmark	Wirelength in [4.44] (in unit)	Wirelength in EIWO-PSO (in unit)
ibm01	62815	62785
ibm03	134511	134304
ibm05	254512	254489
ibm07	353078	353162
ibm10	588269	588328

The discussed hybrid algorithm is juxtaposed to certain typical benchmarks for VLSI global routing, such as ISPD'98 [4.43], and the wirelength obtained is compared to [4.44], as shown in Table 4.7. The findings clearly demonstrate the presented method's comparable efficiency, which varies roughly *7%* from the standard value, and in certain situations, the suggested algorithm demonstrates its superiority.

The suggested EIWO method and the PSO algorithm have a balance between the time and performance. In order to maintain the exchange, the EIWO with high runtime complexity outperforms the PSO-W. The presented EIWO-PSO Hybrid algorithm outperforms PSO-W and EIWO at path minimization for VLSI circuits, but at the expense of computational time.

References

4.1 W. A. Dees, and P. G. Karger, "Automated rip-up and reroute techniques," *in Proceedings of Design Automation Conference*, Las Vegas, NV, USA, June 1982, pp. 432-439.

4.2 W. A. Dees, and P. G. Karger, "Automated rip-up and reroute techniques," *in Proceedings of Design Automation Conference*, Las Vegas, NV, USA, June 1982, pp. 432-439.

4.3 M. Cho, and D. Z. Pan, "Box Router: A new global router based on box expansion and progressive ILP," *in Proceedings of. DAC*, San Francisco, CA, USA, 2006, pp. 373–378.

4.4 M. Pan, and C. Chu, "Fast Route 2.0: A high-quality and efficient global router," *in Proceedings of ASP-DAC*, Yokohama, Japan, 2007, pp. 250–255.

4.5 Y-J. Chang, Y-T. Lee, and T-C. Wang, "NTHU-Route 2.0: A fast and stable global router," *in Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, USA, 2008, pp. 338-343.

4.6 M. D. Moffitt, "Maizerouter: engineering an effective global router," *in Proceedings of Asia and South Pacific Design Automation Conference*, Seoul, Korea, 2008, pp. 226–231.

4.7 S. Rajagopalan, and V. V. Vazirani, "On bidirected cut relaxation for the metric Steiner tree problem," *in Proceedings of Tenth annual ACM-SIAM symposium on Discrete algorithms*, Baltimore, Maryland, 1999, pp. 742-571.

4.8 S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220 no. 4598, pp 671-680, May, 1983.

4.9 A. K. Biswas, A. Kumar, N. Das, and S.K. Sarkar, "Genetic Algorithm Based Delay Optimization in Computer Network to Initiate Efficient Dynamic Routing for Congestion Control," *in Proceedings of conference on Horizons of Telecommunication*, Calcutta, India, 2003.

4.10 M. K. Naskar, B. Mukherjee, J. N. Mazumder, and S.K Sarkar, **"**A Genetic Approach for Design Protect ion in WDM Optical Networks," *in Proceeding of conference on Photonics-2002,* BARC, Bombay, India, 2002.

4.11 M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system optimization by a colony of co-operating agents," *In IEEE Transaction on Systems, Man and Cybernetics-B*, vol. 26 no. 1, pp. 1-3, Feb., 1996.

4.12 A. R. Sardar, M. Singh, R. R. Sahoo, K. Majumder, J. K. Sing, and S. K. Sarkar, "An Efficient Ant Colony Based Routing algorithm for Better Quality of Services in MANET," *in Proceedings of the 48th Annual Convention of CSI,* 2014, pp: 233-238.

4.13 A. Khan, P. Bhattacharya and S. K. Sarkar, "A swarm based global routing optimization scheme," *in Proceedings of International Conference on Advances in Electrical Engineering*, Vellore, India, 2014, pp. 1-4.

4.14 P. Bhattacharya, A. Khan, S. K. Sarkar and S. Sarkar, "An artificial bee colony optimization based global routing technique," *in Proceedings of The 2014 International Conference on Control, Instrumentation, Energy and Communication (CIEC),* Calcutta, India , 2014, pp. 621-625.

4.15 R. R. Sahoo, M. Singh, B. M. Sahoo, K. Majumder, S. Ray, and S. K. Sarkar, "A Light Weight Trust based Secure and Energy Efficient Clustering in Wireless

Sensor Network: Honey Bee Mating Intelligence Approach," *Elsevier, Procedia Technology,* vol. 10, pp. 515-523, 2013.

4.16 N. B. Singh, S. Deb, G.P. Mishra, and S. K. Sarkar, "Application of Particle Swarm Optimization Algorithm for better Nano-devices," *in Proceedings of 2nd Int. Conf. on Contemporary Computing (IC3),* Noida, India, 2009, pp 349–357.

4.17 S. Deb, N. B. Sing, S. Sarkar, and S. K. Sarkar, "Particle Swarm Approach for Better Modelling of Quantum well Nano-structure," *Expert system with application*, Science Direct, vol. 38, no. 10, pp. 12999-13004, Sept., 2011.

4.18 C. Dong, G. Wang, Z. Chen, S. Sun, and D. Wang, "A VLSI routing algorithm based on improved DPSO," *in Proceedings of IEEE International Conference on Intelligent Computing and Intelligent Systems*, Shanghai, China, 2009, pp. 802-805.

4.19 A. Khan, S. Laha, and S. K. Sarkar, "A novel particle swarm optimization approach for VLSI routing," *in Proceedings of 3rd IEEE International Advance Computing Conference*, Ghaziabad, India, 2013, pp. 258-262.

4.20 A. Khan, P. Bhattacharya and S. K. Sarkar, "A swarm based global routing optimization scheme, " *in Proceedings of International Conference on Advances in Electrical Engineering (ICAEE)*, Vellore, India, 2014, pp. 1-4

4.21 G. Liu, X. Huang, W. Guo, Y. Niu and G. Chen, "Multilayer Obstacle-Avoiding X-Architecture Steiner Minimal Tree Construction Based on Particle Swarm Optimization, " *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 1003-1016, May 2015

4.22 G. Chen and E. F. Y. Young, "SALT: Provably Good Routing Topology by a Novel Steiner Shallow-Light Tree Algorithm, " *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 6, pp. 1217-1230, June 2020

4.23 X. Chen, G. Liu, N. Xiong, Y. Su and G. Chen, "A Survey of Swarm Intelligence Techniques in VLSI Routing Problems," in IEEE Access, vol. 8, pp. 26266-26292, 2020

4.24 A. R. Mehrabian, and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *In Ecological Informatics*, vol. 1, no. 4, pp. 355–366, Dec., 2006.

4.25 H. S. Rad, and C. Lucas, "A recommender system based on invasive weed optimization algorithm," i*n Proceedings of IEEE Congress on Evolutionary Computation*, Singapore, 2007, pp. 4297–4304.

4.26 M. R. Ghalenoei, H. Hajimirsadeghi, and C. Lucas, "Discrete invasive weed optimization algorithm: Application to cooperative multiple task assignment of UAVs," *in Proceedings of 48th IEEE Conference on Decision and Control and 28th Chinese Control*, Shanghai, 2009, pp. 1665–1670.

4.27 H. Hajimirsadeghi, and C. Lucas, "A hybrid IWO/PSO algorithm for fast and global optimization," *in Proceedings of IEEE International Conference on Smart Technologies*, St. Petersburg, Russia, 2009, pp. 1964-1971.

4.28 C. Veenhuis, "Binary Invasive Weed Optimization," *in Proceedings of Second World Congress on Nature and Biologically Inspired Computing*, Kitakyushu, Japan, 2010, pp. 449-454.

4.29 A. Rahimi, M. Ahangaran, P. Ramezani, and T. Kashkooli, "An Improved Artificial Weed Colony for Continuous Optimization," *in Proceedings of UKSim 5th European Symposium on Computer Modeling and Simulation*, Madrid, Spain, 2010, pp. 1-5.

4.30 I.D. Falco, A.D. Cioppa, D. Maisto, U. Scafuri, and E. Tarantino, "Biological invasion–inspired migration in distributed evolutionary algorithms," *In Information Sciences,* vol. 207, pp. 50–65, Nov., 2012.

4.31 D. Kundu, K. Suresh, S. Ghosh, S. Das, B.K. Panigrahi, and S. Das, "Multi-objective optimization with artificial weed colonies," *In Information Sciences*, vol. 181, no. 12, pp. 2441–2454, June, 2012.

4.32 A. Basak, D. Maity, and S. Das, "A differential invasive weed optimization algorithm for improved global numerical optimization", *In Applied Mathematics and Computation*, vol. 219, no. 12, pp. 6645–6668, Feb. 2013.

4.33 C. Sur, and A. Shukla, "Discrete invasive weed optimization algorithm for graph based combinatorial road network management problem," *in Proceedings of IEEE International Symposium on Computational and Business Intelligence*, New Delhi, India, 2013, pp. 254-257.

4.34 Y. Bai, S. Xiao, C. Liu, and B. Wang, "A Hybrid IWO/PSO Algorithm for Pattern Synthesis of Conformal Phased Arrays," *In IEEE Transactions on Antennas and Propagation*, vol. 61, no. 4, pp. 2328-2332, April, 2013.

4.35 D. Das, and S. Das, "A cooperative spectrum sensing scheme using multi objective hybrid IWO/PSO algorithm in cognitive radio networks," *In proceeding of International Conference on Issues and Challenges in Intelligent Computing Techniques*, Ghaziabad, India, 2014, pp. 225-230.

4.36 S. K. Mahto, and A. Choubey, "A Novel Hybrid IWO/WDO Algorithm for Interference Minimization of Uniformly Excited Linear Sparse Array by Position-Only Control," *In IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 250-254, 2016.

4.37 Y. R. Naidu, and A. K. Ojha, "Solving Multi objective Optimization Problems Using Hybrid Cooperative Invasive Weed Optimization with Multiple Populations," *In IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 6, pp. 821-832, June, 2018.

4.38 X. Yue, H. Zhang, and H. Yu, "A Hybrid Grasshopper Optimization Algorithm with Invasive Weed for Global Optimization," *In IEEE Access*, vol. 8, pp. 5928-5960, Sept., 2020.

4.39 P. Angeletti, L. Berretti, S. Maddio, G. Pelosi, S. Selleri and G. Toso, "Phase-Only Synthesis for Large Planar Arrays via Zernike Polynomials and Invasive Weed Optimization," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 3, pp. 1954-1964, March 2022.

4.40 C. Pradhan, M. K. Senapati, S. G. Malla, P. K. Nayak and T. Gjengedal, "Coordinated Power Management and Control of Standalone PV-Hybrid

System with Modified IWO-Based MPPT," IEEE *Systems Journal*, vol. 15, no. 3, pp. 3585-3596, Sept. 2021.

4.41 S. M. Ross, "Peirce's Criteria for the elimination of suspect experimental data," *Journal of Engineering*, vol. 20, no. 2, pp 1-12, Sept., 2003.

4.42 D. M. Warme, P. Winter, and M. Zacharisen, "The GeoSteiner 5.0.1 package", http://www.geosteiner.net/geosteiner-5.0.1.tar.gz. (accessed April. 18, 2019)

4.43 C. J. Alpert, "The ISPD98 Circuit Benchmark Suite," *in Proceedings of International Symposium on Physical Design (ISPD '98)*, Monterey, CA, April 1998, pp.80-85.

4.44 R. Sardar, R. Mondal, and T. Samanta, "Geometry Independent Wirelength Estimation Method in VLSI Routing," *in Proceedings of 26th International Conference on VLSI Design and the 12th International Conference on Embedded Systems*, Pune, India, 2013, pp 257-261.

CHAPTER 5

VLSI GLOBAL ROUTING OPTIMIZATION USING PHYSARUM BIONETWORK

5.1. Introduction

Advent of the nanometric age development and efficiency has put an adverse impact on the production methods of VLSI [5.1] - [5.2]. Due to this impact the chip size has declined resulting into the implementation of thousands of transistors in a singular chip. Due to this increased number of parts in a small chip space the intricacies of the connection increases and this leads to increased power loss. To resolve this issue on one hath and to design the routing network of the chip more efficiently on another hand thus The Global routing [5.3] - [5.4] of VLSI chips became a puzzle for all the researchers. The puzzle involves the reduction of the wire length without disclosing the original geometric layout. To solve and project this puzzle in real world has another set of issues related to it. To portray it is required mapping in a graph, this leads to the RSMT Problem [5.5] - [5.6]. This problem is a NP hard problem [5.7] and cannot be solved in polynomial time and it directs to the issue of great time complexity. A profound researcher named Hanan has provided a lot of theorems and their related lemmas to solve this polynomial time issue of RMST [5.8]. These solutions lead to reduced search area; however, the issue of combinatorial explosion could not be tackled due to the tradeoff for search space. Several algorithms have taken account of heuristic techniques [5.9] to check for better and efficient solutions to this tree problem, however even the space and time complexities still posse a problem. Due to these drawbacks researchers started developing new generation of algorithms under the branch of metaheuristics [5.10] - [5.12]. These algorithms were inspired from swarm intelligence i.e., the natural lifestyle of living beings like ants, weeds, etc. These new generation algorithms have mainly two steps, first it defines a random search space and secondly it iterates until the new solution is better than the previous solution until the optimal solution is reached. The ACO [5.13] – [5.16] and PSO are two popular algorithms in this field. Being an efficient and reliable algorithm the PSO method [5.17] – [5.21] relays the behavioral characteristics of swarms and follows the above-mentioned steps. The only drawback of the PSO algorithm is that when the population booms in the solution space it converges before it is supposed to and the optimal solution is not achieved. Hence to settle this drawback the control parameters are closely monitored. When the regular computational algorithms fail to provide optimal output, researchers turn to the behavioral patterns of the natural world in order to develop new algorithms. This field is known as Bio-inspired optimization [5.22] - [5.26]. Bio-inspired optimization works based on the interaction between the organisms, it has shown great promise and has a lot of application. R.R. Shaoo et al. [5.27] employed bio inspired approach in wireless sensor network in 2015. Bio-sensor [5.28] and application also find prominence in recent times.

To deal with the Global routing issue in VLSI circuit another new computational method using Bio-inspired optimization is implemented. This Bio-inspired optimization has been taken up from the study of single celled, amoeboid organism

Physarum polycephalum slime mold. Physarum polycephalum is made up of a network of tubes via which nutrients, neural signals and body mass is transported. This process is conducted over the shortest path hence this method is adopted in solving the issue of the RSMT to efficiently design the global routing.

Furthermore, new hybridization algorithms for PSO with Physarum are being investigated, in which the efficiencies of both algorithms are combined in a novel way to improve global optimization.

5.2. Literature Review

T. Nakagaki et al. [5.29] in 2000 took a developing tip of a suitable size from a huge plasmodium in a 25235 cm culture box and isolated it into little pieces. At that point, a labyrinth made by cutting a plastic film and putting it on an agar surface. The plasmodial pieces spread and combine to frame a solitary life form that filled the labyrinth. Then it was staying away from the dry surface of the plastic film.

T. Nakagaki et al. [5.30] in 2001 showed their work on the essential characters of the cylinder morphogenesis by applying an outer incitement through adding supplements to the organic entity. The important conditions under which the plasmodium can follow the briefest not really settled.

T. Nakagaki et al. [5.31] provide the plasmodium with another kind of undertaking including enhancement conduct. Two separate functions are introduced to the creature, which is enlightened by an inhomogeneous light field. Since the plasmodium is photophobic, tubes associating the functions avert following the basic briefest ways, however structure as indicated by the light inhomogeneity. A report on the conduct of the creature under these conditions made to examine its physiological importance. A numerical model suggested for the cell elements what's more present a computational calculation for its concern addressing.

A. Tero et al. [5.32] depicts how the organization of cylinders grows and contracts depending on the transition of protoplasmic streaming, and replicates test perceptions of the conduct of the life form. The calculation is dependent on physarum is basic and incredible.

J. Garcia et al. [5.33] gives another way to deal with issues related with combinatorial. It utilizes a natural similarity roused by the execution of viruses. The replication instrument, just as the hosts' contamination processes is utilized to produce a metaheuristic that permits the obtention of significant outcomes. The viral framework (VS) hypothetical setting is portrayed and it is applied to a library of medium-to-huge measured instances of the Steiner issue for which the ideal arrangement is known. The strategy is contrasted and the metaheuristics that have given the best outcomes to the Steiner issue. The VS gives preferable arrangements over hereditary algorithm sand certain unthinkable hunt draws near. For the most modern unthinkable pursuit draws near (the best metaheuristic approximations to the Steiner issue arrangement) VS gives arrangements of comparable quality.

S. Takagi et al. [5.34] in 2010 took advantage of the Physarum polycephalum to foster an organically propelled model for versatile organization improvement. Physarum is a huge, single-celled amoeboid life form that scrounges for patchily circulated food sources. The singular plasmodium at first investigates with a generally adjoining scrounging edge to amplify the region looked. Nonetheless, behind the edge, this is settled into a rounded organization connecting the found food sources through direct associations, extra middle of the road intersections (Steiner focuses) that diminish the general length of the associating organization, furthermore the arrangement of infrequent cross-connects that further develop in general vehicle proficiency and versatility.

Computational and numerical techniques are broadly used to investigate and display organic frameworks by Y. Afek et al. [5.35]. An illustration of the opposite of this technique, where a natural cycle is utilized to determine an answer for a long-standing computational issue is reported. It is accepted an assortment of indistinguishable processors set at hubs of a subjective simultaneous correspondence organization. Hubs can just transmission the slightest bit messages. A message communicated by a hub arrives at every one of its neighbours' that are as yet dynamic in the calculation. The model is proper for radio organizations with impact location.

L. Liu et al. [5.36] utilizing bits of knowledge from natural cycles could assist with planning new enhancement strategies for long-standing computational issues. This work takes advantage of a cell registering model in the physarum polycephalum to tackle the insignificant openness way issue which is a crucial issue comparing to the most pessimistic scenario inclusion in remote sensor organizations. It initially details the insignificant openness way issue, and afterward convert it into the most limited way issue by discretizing the observing field to a huge scope weighted matrix. Motivated by the way tracking down capacity of physarum. Then another advancement calculation fostered as the physarum improvement, for tackling the most limited way issue. The suggested calculation is with low-intricacy and high-parallelism. Additionally, the center component of physarum improvement in the work is likewise useful for planning new chart calculations and improving directing conventions and geography control in self-coordinated organizations.

K. Mehlhorn et al. [5.37] gives an idea on Physarum polycephalum. It is a slime form that is clearly ready to tackle most brief way issues. A numerical model has been suggested in year 2007 to depict the input instrument utilized by the slime form to adjust its cylindrical channels while rummaging two food sources s_0 and s_1 by Tero et al. It was demonstrated that, under this model, the mass of the shape will ultimately join to the most limited s_0 to s_1 way of the organization that the form lies on, freely of the design of the organization or of the underlying mass conveyance.

H. Zhang, and L. Liu [5.38] in 2014 solved energy proficiency issue in remote multi-hop networks addressed by an Advanced Distributed physarum optimization calculation. An ideal energy-proficient tree issue dependent on the cell processing model in the ooze shape physarum polycephalum is figured. At that point, union rate and precision by the clever transition terminal choosing strategy improved and erasing edge system, and compromise iterative expense and strength by consolidating the focal and appropriated iterative calculation. The simulated results exhibits that the energy-proficient tree developed by the calculation accomplishes the preferred presentation over Directed Diffusion convention, and the comparable presentation to Loss-Contracting calculation in less building overhead.

L. Liu et al. [5.39] takes advantage of model on cellular computing in the physarum polycephalum to tackle the Steiner tree issue which is a significant NP-hard issue in different applications, particularly in network plan. Roused by the way finding and organization arrangement capacity of physarum, another advancement calculation fostered, named as the physarum improvement, with low-intricacy and high-parallelism. To approve and assess the models and calculation, physarum enhancement to the insignificant openness issue applied which is a principal issue comparing to the most pessimistic scenario inclusion in remote sensor organizations. Additionally, the center component of physarum advancement likewise may give a valuable beginning stage to foster some viable dispersed calculations for network plan.

Y. Sun et al. [5.40] reported two new physarum-enlivened calculations to address Node Weighted Steiner Tree Problem. Since all the current benchmark cases have a vacant terminal set, new benchmark cases with non-void terminal sets are produced to cover the lack of existing benchmark examples. Both calculations are contrasted. Moreover, an adjusted Dijkstra's calculation is suggested to give the ideal answers for a piece of these benchmark examples where there are two terminals and the hub loads are negative. Simulated results show that previously suggested calculation can observe the ideal answers for NWSTP with two terminals in diagrams with negative hub loads, and second suggested calculation can observe close estimated answers with various terminals in any hub weighted chart.

C. Gao et al. [5.41] reported the computational capacity and feedback component directly following scavenging interaction of physarum, which is a huge one-celled critter like cell comprising of a dendritic organization of cylinder like pseudopodia, an overall physarum-based computational system for local area discovery suggested. In light of the system, the in between local area edges can be recognized from the intra-local area edges in an organization and the positive input of tackling process in a calculation can be additionally upgraded, which are utilized to work on the effectiveness of unique improvement based and heuristic-based local area discovery calculations, separately. Bench mark datasets have been utilized to assess the proficiency of the suggested computational system. Tests show that the calculations streamlined by physarum-roused computational structure perform better compared to the first ones, as far as exactness and computational expense. Besides, a computational intricacy examination confirms the structure of suggested system.

Y. Liu et al. [5.42] report vehicle organizations taken as one of the most discussed issues in the space of computational knowledge. Some nature-propelled calculations have shown amazing capacities in the versatile organization development. The work to plan a physarum scavenging stage for building transport organizations. In particular, the customary physarum scavenging model is adjusted to develop transport networks in China. To draw near to the genuine situation, functional information is gathered to construct the climate of the physarum scrounging model and the design of genuine vehicle organizations. A few estimations in the area of complicated organizations, for example, normal way length, network effectiveness, geography strength, and utilitarian vigour, are utilized for execution examination. The trial results show that physarum scrounging models dominate in building profoundly productive and strong organizations, which can be used for coordinating the plan of transport networks in reality.

5.3. Wirelength Minimization of VLSI circuits based on Physarum Bio Network

The major concern of VLSI Global routing is wirelength minimization of the interconnected nodes in a VLSI layout which is mapped to graph theory generating a RSMT problem. A grid graph is executed for the algorithm based on Physarum BioNetwork. $G(V,E)$ is a graph representing a solution space, where V are the endpoints and E are the global routing paths. R is the set of terminal vertices which are to be connected in the VLSI layout and $V-R$ is the set of steiner vertices.

In the Physarum model for VLSI, the flux changes along with the time due to its time variant nature from (2.27) shown in (5.1)

$$q_{ij}^{\sigma} = \frac{d_{ij}^{\sigma}}{l_{ij}} (p_i^{\sigma}(t) - p_j^{\sigma}(t)) \tag{5.1}$$

Here q_{ij} is the flux induced by the vertex $v \in S$ and S is the set of consisting of vertices other than the terminal nodes, i.e. $S = V - R$.

For a physarum cell v_i , the algebraic sum of the fluxes (2.28) are given by (5.2)

$$\sum_{i=1,j\neq1} q_{ij}^{\sigma} = \sum_{i=1,j\neq1} \frac{d_{ij}^{\sigma}}{l_{ij}} \left(p_i^{\sigma}(t) - p_j^{\sigma}(t)\right) = \begin{cases} f_0, & if\ v_i \in S \wedge i = \sigma \\ -f_0 & if\ v_i \in S \wedge i \neq \sigma \\ 0, & if\ v_i \in V \setminus S \end{cases} \tag{5.2}$$

To put d_{ij} value as 1 other than 0, d_{ij} is updated and thereby the the pressure value p_i^{σ} is also updated. The value of d_{ij} is constantly reduced at each time step as the value of d_{ij} is represented as in (5.3)

$$\frac{d(d_{ij})}{dt} = f\left(|q_{ij}|\right) - \alpha d_{ij} \tag{5.3}$$

This value gets eliminated from the solution space along with its respective connected nodes, at the time when it touches a particular threshold or limit, lower than that of the minimal. The associated points or the food sources from the edges left with d_{ij} more or equal to the corresponding threshold are chosen to be applied in a separate graph along with the terminal nodes. The RSMT is calculated, obtaining the overall minimum wirelength of the interconnected terminal nodes such that VLSI global routing optimization is attained.

5.3.1. RSMT Algorithm based on Physarum BioNetwork

```
procedure Physarum ($graph(N \times N), t$)
    Put the terminals $t$ in the graph
    for $iter = 1$ to $steiner$ do
        Create a Steiner point $st\ (x, y)$
        if $st \neq terminal$ and $st$ does not lie on path then
            Put the Steiner point on the graph
        else
            $collision = collision + 1$
    $steiner = steiner - collision$
    for $j = nodes + 1$ to $nodes + steiner$ do
        Calculate the pressure and initialize all vertices
    for $h = 1$ to $nodes + steiner$ do
```

```
        for $g = 1$ to $nodes$ do
            $pdif\ f(h,g) = p(h) - p(g)$
    for $j = 1$ to $nodes + steiner$ do
        for $k = 1$ to $nodes + steiner$ do
            if $j \neq k$ then
                Calculate the $d(j,k)$
    if $d(j,k) \geq threshhold$
        $d(j,k) = q_{jk} - d_{jk}$
    else if $d(j,k) < threshold$
        Eliminate $edge(j,k)$
```

5.3.2. Physarum BioNetwork-PSO Hybrid

The amoeboid bio network is used to remove the unnecessary edges of the RSMT optimizing it. However, when the solution space is large iterating through all the edges to eliminate the unnecessary ones takes a lot of iterations hence increasing time complexity and reducing the convergence rate.

The PSO algorithm is devoid of this drawback because it works by updating the previous velocity and position in the next iteration. This change is based on the swarm's behavioral characteristics. In this algorithm the viability of the Steiner points and then updates the Steiner set leading to a higher rate of convergence. However, the only drawback for this algorithm is that it depends on probability. This results in a lot of issue while designing the global and local routing and leads to premature convergence.

PSO algorithm is gets more efficient when the Physarum BioNetwork is implemented in it. Physarum polycephalum is made up of a network of tubes via which nutrients, neural signals and body mass is transported. This process is conducted over the shortest path hence this method is adopted in solving the issue of the RSMT to efficiently design the global routing. This optimization is dependent on how the bio network is used in the discussed hybridization. This hybridization leads to the production of viable population where they are used to create the initial set hence ensuring better output and the efficiency of the algorithm improves.

$G(V,E)$ is a graph representing a solution space, where V are the endpoints and E are the global routing paths. Pre-algorithm settings include the swarm's size and the maximum number of iterations. The initial population of size p is set and the Steiner set where $n(p) = n - 2$, where n being the terminal nodes.

5.3.3. Algorithm based on Physarum-PSO Hybrid

Step 1 Random swarms of size z are initialized and each $Q_j \epsilon z$ where $n(Q_j) = t - 2$ where t is the number if terminal points.

Step 2 Fitness of each particle is evaluated $1/MST(Q_j)$ and the respective p_{best} and g_{best} values are calculated.

Step 3 Particle's (p) velocity (V) and position (S) are computed using (2.17), (2.18) and (2.16)

Step 4 Evaluate $\mathrm{p_{best}}$ and $\mathrm{g_{best}}$ after each iteration

Step 5 Select the particle $\mathrm{Q_j}'$ with minimum $\mathrm{g_{best}}$

Step 6 Physarum algorithm begins with $\mathrm{Q_j}' = \{\mathrm{p_1}, \mathrm{p_2}, \dots \dots., \mathrm{p_s}\}$ where $\mathrm{n}(\mathrm{Q_j}') \leq \mathrm{t} - 2$

Step 7 For each $\mathrm{p_i} \in \mathrm{Q_j}$ calculate flux $\mathrm{q_{ij}}$ using (5.1)

Step 8 Calculate $\sum \mathrm{q_{ij}}$ using (5.2)

Step 9 Calculate new $\mathrm{d_{ij}}'$ using (5.3)

Step 10 When $\mathrm{d_{ij}} <$ threshhold

Step 11 Eliminate the edge $\mathrm{l_{ij}}$

5.3.4. Mathematical Implication of the Algorithm

A graph of R number of terminal nodes and V total nodes are set up. The generation of R random terminals, in a graph consisting of V nodes is considered $'p_i'$ at $'v_i'$. The idea of the algorithm remains same i.e.; to reduce the interconnecting length of the nodes, so the following rules are followed in (5.4), (5.5) and (5.6).

$$d(a,b) \geq 0 \, for \, all \, a \neq b \tag{5.4}$$

$$d(a,b) = d(b,a) \tag{5.5}$$

$$d(a,b) \leq d(a,c) + d(c,b) \tag{5.6}$$

It is observed from (5.6) that the amoeboid cells which occupy the same position as that of the edges or Steiner points leads to an increased flux density which results in longer graph length. So, to avoid this scenario amoeboid cells are placed in separate location and hence the edge length also decreases. This scenario is looked into by beginning with a first order equation $ax + by + c = 0$ and with just 1 terminal. The nodes are placed along the line $y = 0$.

The nodes will found at $P = (x_1, f(x_1))$, $Q = (x_2, f(x_2))$, $R = (x_3, f(x_3))$ along the equation and $P' = (x_1, 0)$, $Q' = (x_2, 0) \, and \, R' = (x_3, 0)$ along the X-axis as shown in Figure 5.1.

Now, taking determinant $D = |x_1 \, y_{11} \, x_2 \, x_3 \, y_{21} \;\; y_{31}| = 0$, area connecting the three nodes is 0 and thus, nodes N_1, N_{21} & N_3 are in a same line. This shows that the Steiner points analogous to the physarum cells if placed along the X-axis the least length would be $L_{min} = (\sum_k^3 |x_k| + |fx_k|) - x_2$. Now if they are put away from the nodes the edge length might decrease as shown in the below graph.

In this case, placing a cell or the Steiner point as shown in Figure 5.2. at X$(x_2, f(x_3))$: the minimum length would be

$$L'_{min} = x_1 + f(x_1) + x_2 + f(x_3) + (x_3 - x_2) \tag{5.7}$$

$$\text{Therefore, } L'_{min} - L_{min} = f(x_2) + (x_3 - x_2) \tag{5.8}$$

$$\text{Therefore, } L_{min} > L'_{min} \tag{5.9}$$

$$\text{Radius of Curvature of } y = f(x) = \frac{(1+(\frac{dy}{dx})^2)^{1.5}}{|\frac{d^2y}{dx^2}|} \tag{5.10}$$

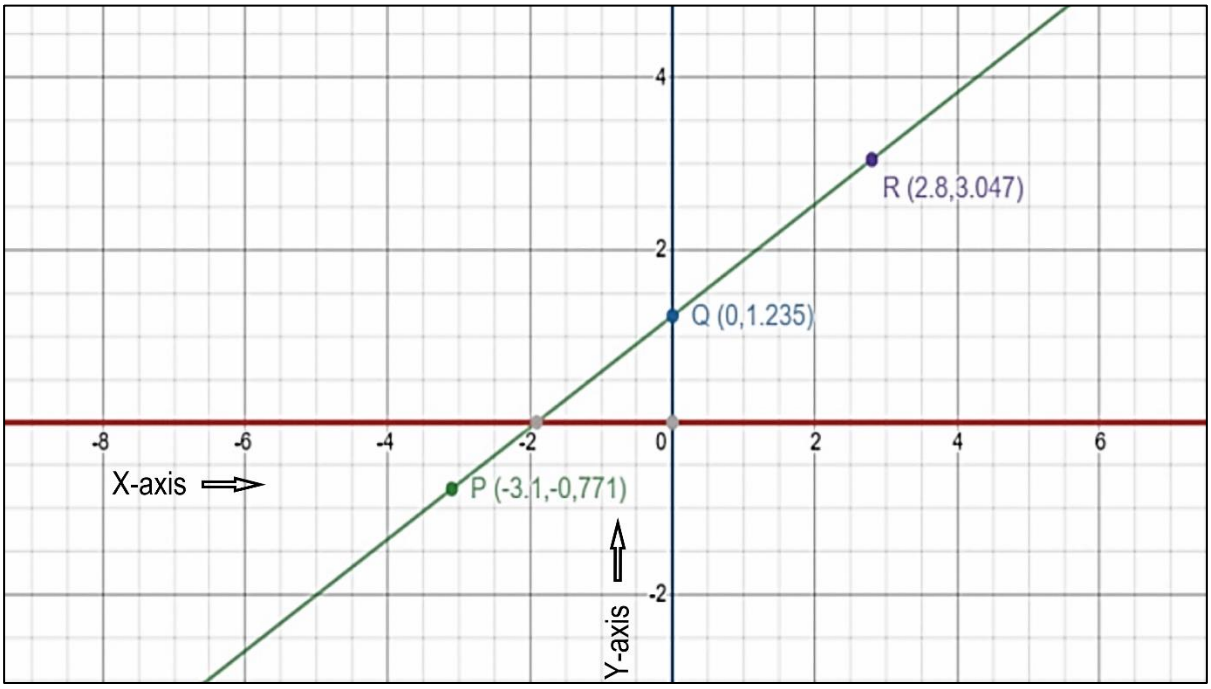

Figure 5.1. Graph 1 representing the placement of nodes along a linear curve

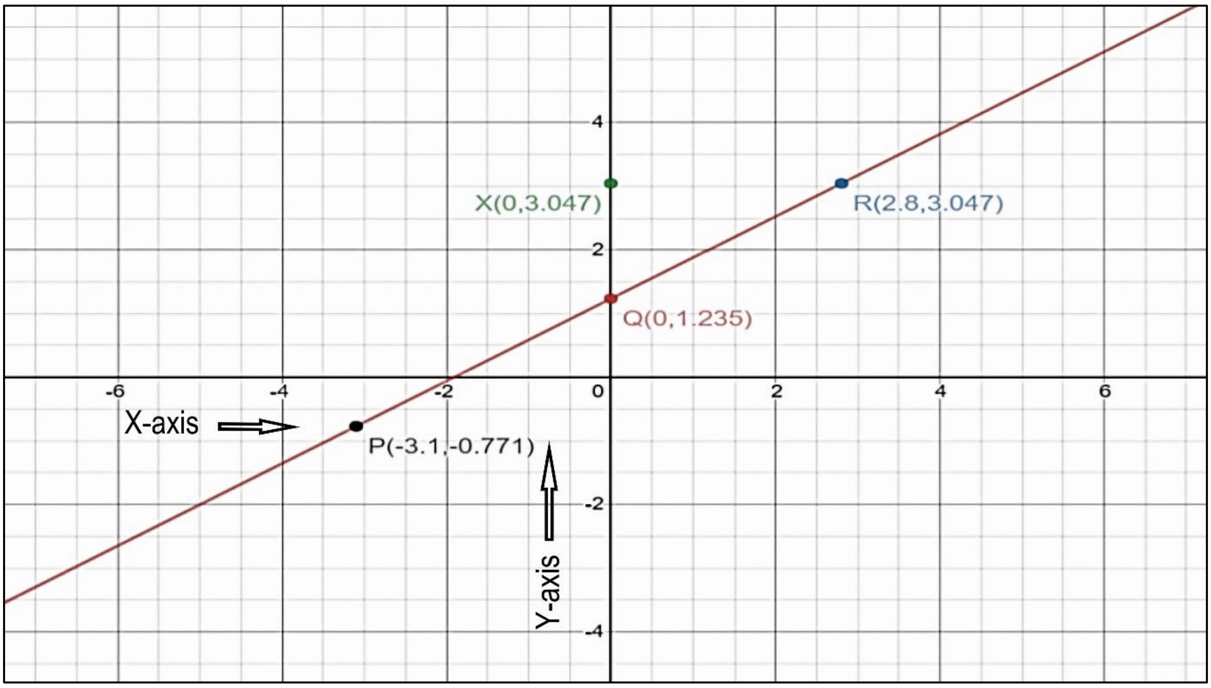

Figure 5.2. Graph 2 representing the position of a single Steiner point w.r.t. the nodes of Graph 1

Now, $ax + by + c = 0$

⇨ $y = -\frac{a}{b}x - \frac{c}{b}$

⇨ $\frac{dy}{dx} = -\frac{a}{b}$

⇨ $\frac{d^2x}{dx^2} = 0$

⇨ $\left|\frac{d^2x}{dx^2}\right| = 0$ (5.11)

Therefore, radius of curvature = ∞

The results are suitable even when the graphs are curved if the Steiner points are plotted along X-axis. If the order of the curve or the radius of curvature increases the number of terminals increases as well. Hence it is shown that this algorithm is viable for any distribution of terminal nodes. To check this work for sure a second-degree equation with 1 terminal node is considered. Let the equation be $ax^2 + bx + c = 0$ and let the nodes be placed along the line $y = 0$ and the equation $ax^2 + bx + c = 0$.

The coordinates of the Steiner nodes will be $P = \left(x_1, f(x_1)\right)$, $Q = \left(x_2, f(x_2)\right), R = \left(x_3, f(x_3)\right)$ along the equation and $P\,' = (x_1, 0)$, $Q' = (x_2, 0)\ and\ R' = (x_3, 0)$ along the X-axis as shown in Figure 5.3.

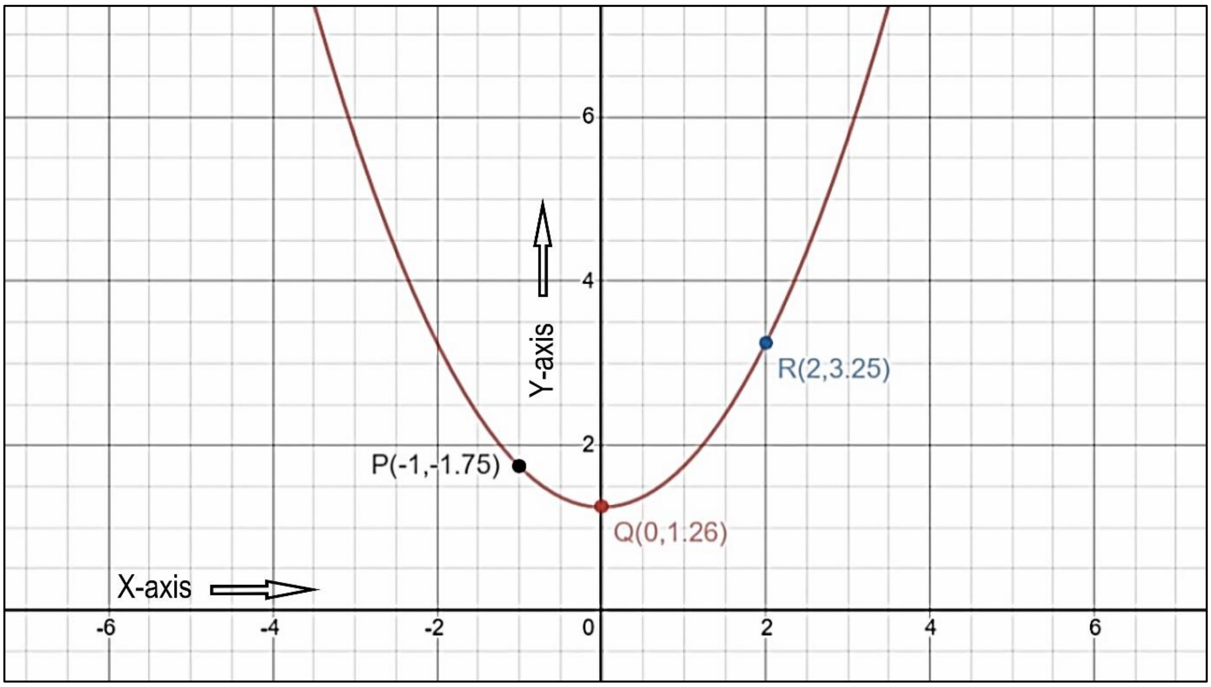

Figure 5.3. Graph 3 representing the placement of nodes along a curve of second degree

In this case as shown in Figure 5.4, placing a cell or the Steiner point at $X(x_2, f(x_3))$: the minimum length is

$$L'_{min} = x_1 + f(x_1) + x_2 + f(x_3) + (x_3 - x_2) \tag{5.12}$$

$$\text{Therefore, } L'_{min} - L_{min} = f(x_2) \tag{5.13}$$

$$\text{Therefore, } L_{min} > L'_{min} \tag{5.14}$$

Thus, this algorithm stands validated with mathematical experimentation.

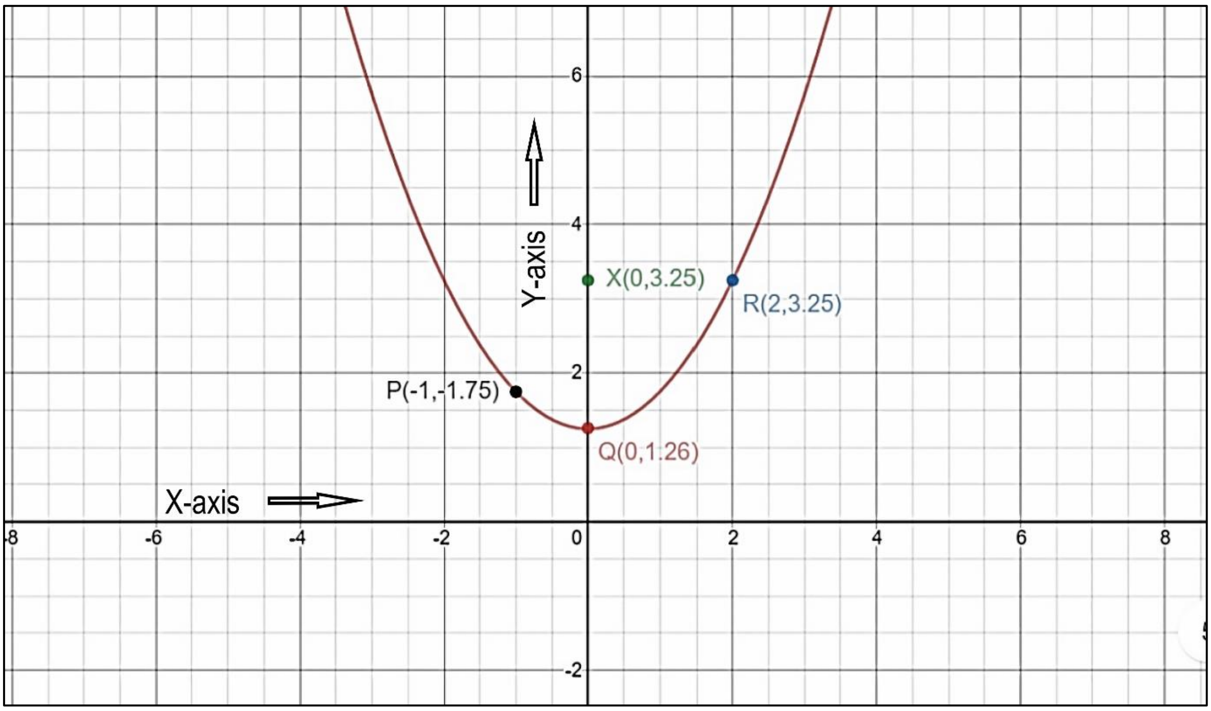

Figure 5.4. Graph 4 representing the position of a single steiner point w.r.t. the nodes of graph 3

5.3.5. Experimental Procedure

A two-dimensional search space of measurements 500 × 500 each is set up where the simulations are run designed on the discussed hybridized algorithms of Physarum (APO) and PSO against the swarm optimization. Two different random co-ordinate sets are created for 15 and 30 terminal nodes based on the distribution pattern in the solution space. Uniform distribution and non-uniform distribution are used for Set 1 and Set 2 respectively. There are 5 set of parameters on which the algorithms are compared.

5.3.6. Results and Discussions

In this results and discussions sub-section, a detailed insight of performance Physarum-PSO algorithms is presented. Table 5.1 shows the records of the output from the algorithms that were run 25 times. The output data exhibits that the discussed algorithms are more efficient than the PSO-W when the number of nodes is increased. However, when the test data is small PSO provides better output, in contrary to it Advance Physarum Optimization (APO) algorithm and APO-PSO hybrid are more efficient in finding the least distance path among nodes. Irrespective of altering the parameters of distribution of nodes, deviation among the minimum and mean length of wire both the presented algorithms provide output in the acceptable range whereas the PSO-W hampers with the consistency of its results**.** Considering 30 nodes, Figure 5.5 shows the minimum wire-length generated by the APO algorithm. and Figure 5.6 show lowest minimum cost generated by APO-PSO hybrid algorithm.

Table 5.1: Comparison of Algorithms for overall interconnected wirelength

Test		**RSMT Cost (in unit)**					
		PSO-W		**APO**		**APO-PSO**	
		Min	**Mean**	**Min**	**Mean**	**Min**	**Mean**
15 nodes	**Set 1**	455	489.2	455	460.4	455	459.1
	Set 2	437	442.5	439	442.2	437	441.5
30 nodes	**Set 1**	2510	2514	2450	2452.3	2441	2446.9
	Set 2	3515	3520.1	3435	3438.6	3432	3437.1

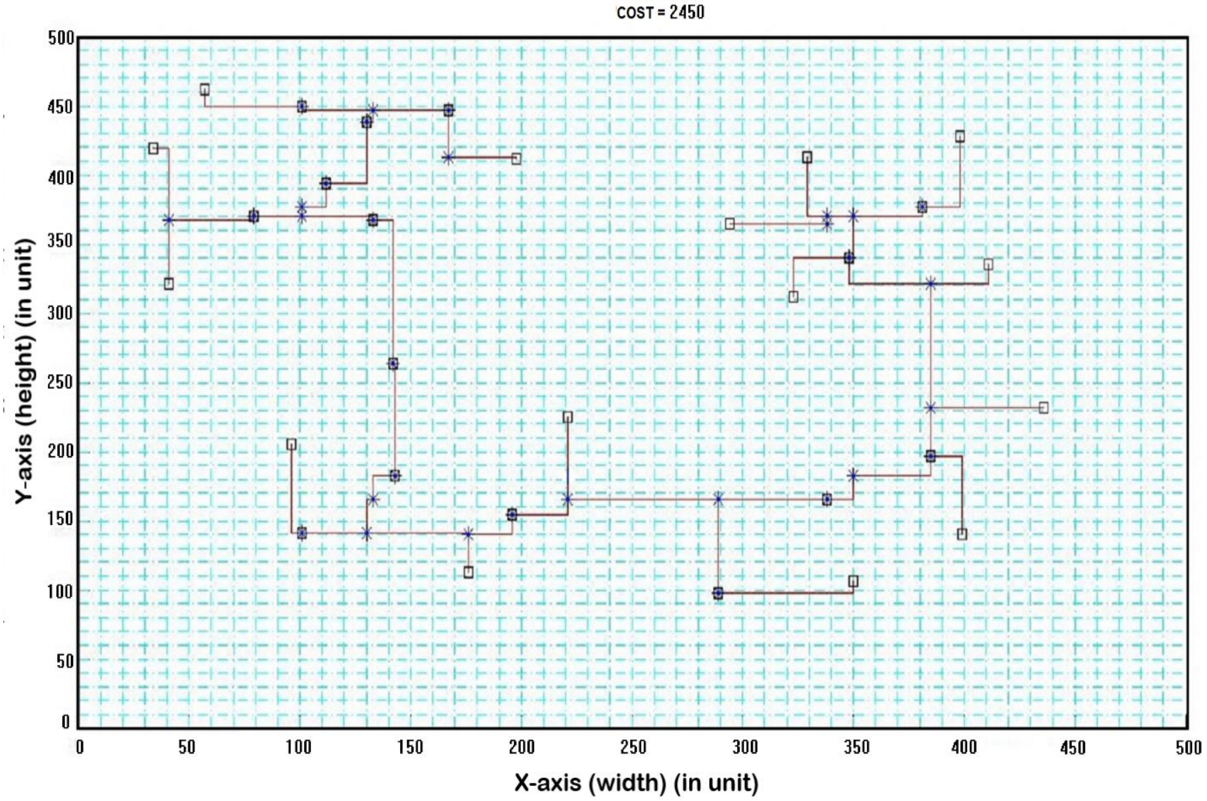

Figure 5.5. Minimum wirelength by APO for Set 1 with 30 nodes

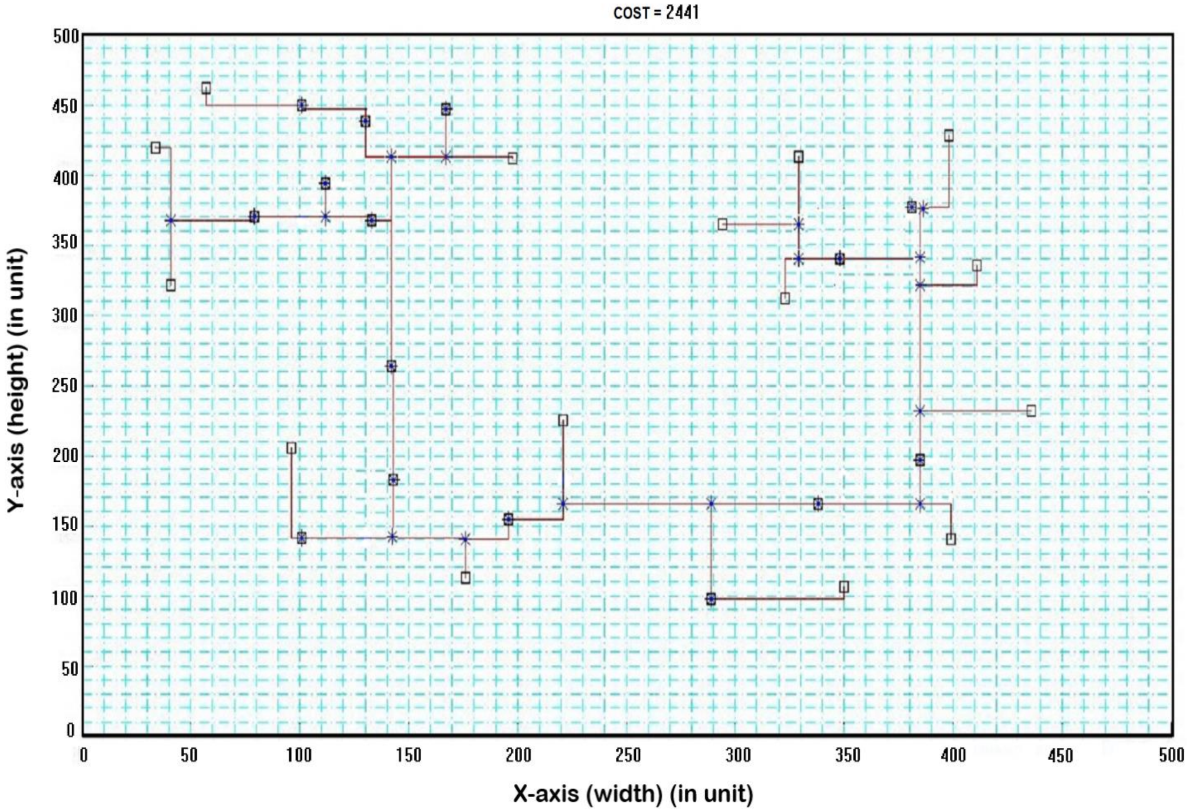

Figure 5.6. Minimum wirelength by APO-PSO for Set 1 with 30 nodes

Table 5.2 depicts the computational complexity of the algorithms during execution. The tests are carried out in a computer with specifications of 8 GB DDR3 RAM, Intel Core *i3* processor with 2.4 GHz speed. With increase in the number of nodes the average runtime of PSO-W increases non-linearly. Due to the physarum optimization the APO algorithm runs more efficiently due to the elimination of less viable routes synonymously the Steiner nodes gets removed. Due to this optimization unnecessary paths gets removed and time complexity gets optimized. However, when the number of nodes get large the complexity increases non-linearly. The table shows that implemented APO consume least average runtime. The hybridization of APO-PSO results in better convergence but more execution time.

Table 5.2. Average algorithm runtime comparison

Test (in seconds)		**PSO-W**	**APO**	**APO-PSO**
15 Nodes	**Set 1**	4.91	3.89	5.76
	Set 2	4.35	3.42	5.39
30 Nodes	**Set 1**	8.26	5.599	11.93
	Set 2	10.80	6.506	15.36

Table 5.3. Standard Deviation comparison obtained from RSMT

Test		**PSO-W**	**APO**	**APO-PSO**
15 Nodes	Set 1	15.44	15.21	11.71
	Set 2	14.13	13.65	10.23
30 Nodes	Set 1	33.809	23.744	18.92
	Set 2	42.48	35.145	31.56

In Table 5.3, when the test set is small in size the PSO and the hybridized algorithm cannot be distinguished as both the test result in terms of convergence is same. However, when the test set is large the later beats the PSO algorithm in time. The PSO algorithm provides the least Standard Deviation (SD) in less time than the physarum algorithm. Now on the parameter of solution space and distribution function involved the physarum algorithm is more efficient than the PSO algorithm.

Table 5.4. Comparison with Geosteiner -5.0.1

Test		**RSMT cost (in unit)**			
		Geosteiner 5.0.1	**PSO-W**	**APO**	**APO-PSO**
15 Nodes	**Set 1**	455	455	455	455
	Set 2	437	437	439	437
30 Nodes	**Set 1**	2357	2510	2450	2441.3
	Set 2	3345	3515	3435	3431.7

Taking into account the benchmark of Geosteiner -5.0.1 of researchers [5.43] same parameters were set up for the experiment of 15 and 30 node data set. All the three discussed algorithms provided the least value of *'455'* which is the benchmark for 15 node data set. Table 5.4 shows the output data against the benchmark. It indicates that APO-PSO Hybrid creates the best while PSO-W generates the worst, proving that RMST graph problem can be solved efficiently utilizing APO-PSO hybrid to minimize wire length.

A random data set of 200 terminal nodes is used to test the presented techniques in a large-scale setting. For 200 terminal nodes, the minimal wirelength cost created by APO-PSO is *'11762'*, compared to *'11854'* generated by APO as shown in Figure 5.7.

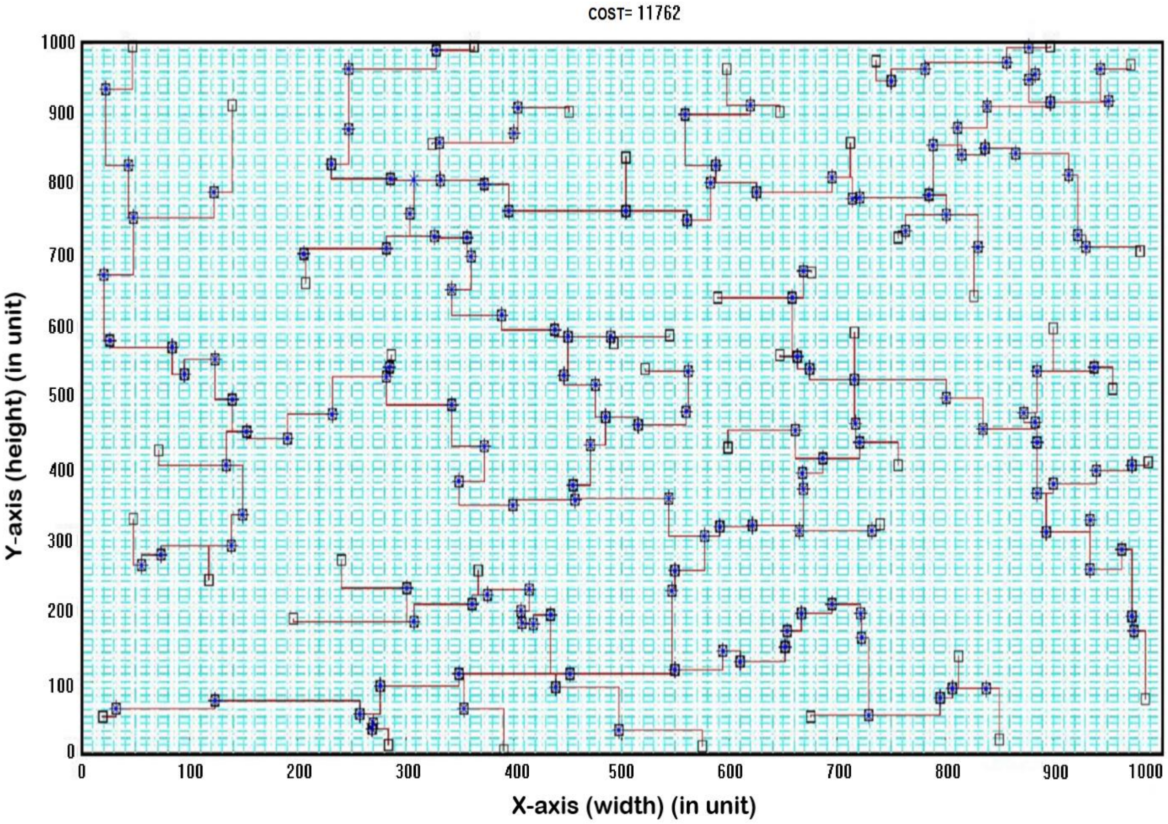

Figure 5.7. APO-PSO result for 200 terminal nodes

The performance of the presented algorithms is measured using some benchmark ISPD'98 [5.44] for global routing and the estimated wirelength is compared against the existing literature [5.45] as tabulated in Table 5.5. It is seen from the results that the discussed algorithm deviates around *5%* where it beats the algorithm in some benchmarks.

Table 5.5. Comparison on Benchmark circuits

Benchmark	Wirelength in [5.45] (in unit)	Wirelength in APO-PSO (in unit)
ibm01	62815	62742
ibm03	134511	134496
ibm05	254512	254489
ibm07	353078	353101
ibm10	588269	588350

The APO-PSO algorithm exhibit superior performance than the conventional PSO algorithm. Experiments show that the APO-PSO hybrid generates the smallest wirelength of interconnected nodes, which is competitive with the established standards. On comparing the APO-PSO hybrid algorithm to APO, it is noticed a little amount of inconsistency; nevertheless, this is far less noticeable than in PSO-W. Nonetheless, this approach can be employed more effectively by running many copies of the same set, hence boosting the likelihood to its peak in obtaining the least optimal wirelength of the VLSI circuit, because of its great ability to lower the wire length size of VLSI circuits.

References

5.1 J. Wu, Y. L. Shen, K. Reinhardt, H. Szu, and B. Dong. "A Nanotechnology Enhancement to Moores Law," *Applied Computational Intelligence and Soft Computing*, vol. 2013, no. 2, pp. 2-15, Jan., 2013.

5.2 W. R. Heller, "Contrasts in Physical Design between LSI and VLSI," *in Proceedings of 18th Design Automation Conference*, Nashville Tennessee USA, 1981, pp. 676-683.

5.3 L. Behjat, "*New modeling and optimization techniques for the global routing problem,*" Ph.D. Thesis, Computer Science Dept., University of Waterloo, Canada, 2002.

5.4 N. Sherwani, "*Algorithms for VLSI physical design automation,*" Dordrecht, Netherland, Kluwer Academic Publishers, 1999.

5.5 J. Ho, G. Vijayan, and C. K. Wong, "New algorithms for the rectilinear Steiner tree problem," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 2, pp. 185-193, Feb. 1990.

5.6 K. Sakai, K. Tsuji, and T. Matsumoto, "An efficient approximation algorithm for the Steiner tree problem in rectilinear graphs," *in Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, Portland, USA, 1989, pp. 339-342.

5.7 M. R. Garey, and D. S. Johnson, "The Rectilinear Steiner Tree Problem is NP-Complete," *SIAM Journal on Applied Mathematics*, vol. 32, no. 4, pp. 826–834, June 1977.

5.8 S. Rajagopalan, and V. V. Vazirani, "On bidirected cut relaxation for the metric Steiner tree problem", *in Proceedings of Tenth annual ACM-SIAM symposium on Discrete algorithms*, 1999, pp. 742-571.

5.9 D. Du, Y. Zhang, and Q. Feng, "On better heuristic for Euclidean Steiner minimum trees," *in Proceedings 32nd Annual Symposium of Foundations of Computer Science,* Washington DC, USA, 1991, pp. 431-439.

5.10 C. Blum, and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison", *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268-308, Sept., 2003.

5.11 F. Glover, and G. A. Kochenberger, "*Handbook of metaheuristics,*" Dordrecht, Netherlands, Kluwer Academic Publishers, Jan. 2003.

5.12 I. H. Osman, and J. P. Kelly, "*Meta-heuristics: Theory and applications,*" Norwell, Massachusetts, USA, Kluwer Academic Publishers, vol. 48, 1997.

5.13 R. Beckers, J. L. Deneubourg, and S. Goss, "Trails and U-turns in the selection of the shortest path by the ant Lasius Niger," *Journal of Theoretical Biology,* vol. 159, no. 4, pp. 397–415, Dec., 1992.

5.14 M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system optimization by a colony of co-operating agents," *IEEE Transaction on Systems, Man and Cybernetics-B*, vol. 26, no. 1, pp. 1-3, Feb., 1996.

5.15 A. R. Sardar, M. Singh, R. R. Sahoo, K. Majumder, J. K. Sing, and S. K. Sarkar, "An Efficient Ant Colony Based Routing algorithm for Better Quality of Services in MANET," *in Proceedings of the 48th Annual Convention of CSI*, Visakhapatnam, India 2013, vol 1, pp: 233-238.

5.16 N. Maharjan, A. Shrestha, A. Tamrakar, and S. P. Panday, "Solving Minimum Steiner Tree based on behavior of ants," *in Proceedings of Third Asian Himalayas International Conference on Internet*, Kathmundu, Nepal, 2012, pp. 1-4.

5.17 Y. H. Shi, and R. C. Eberhart, "Empirical study of particle swarm optimization," *In Proceedings of IEEE Congress on Evolutionary Computation,* Washington-DC, USA, 1999, pp. 1945-1950.

5.18 M. Clerc, and J. Kennedy, "The Particle Swarm: Explosion, Stability, and Convergence in a Multi-dimensional Complex Space," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 1, pp. 58–73, February, 2002.

5.19 N. B. Singh, S. Deb, G.P. Mishra, and S. K. Sarkar, "Application of Particle Swarm Optimization Algorithm for better Nano-devices" *in Proceedings of 2nd Int. Conf. on Contemporary Computing (IC3),* Noida, India,2009, pp 349-357.

5.20 C. Dong, G. Wang, Z. Chen, S. Sun and D. Wang, "A VLSI routing algorithm based on improved DPSO," *In Proceedings of IEEE International Conference on Intelligent Computing and Intelligent Systems*, Shanghai, China, 2009, pp. 802-805.

5.21 A. Khan, S. Laha, and S. K. Sarkar, "A novel particle swarm optimization approach for VLSI routing," *In Proceedings of 3rd IEEE International Advance Computing Conference*, Ghaziabad, India, 2013, pp. 258-262.

5.22 D. Simon, "Biogeography-Based Optimization," *IEEE transactions on evolutionary computation,* vol. 12, no. 6, pp. 702-713, Dec., 2008.

5.23 Junchen, and M. Mahfouf, "Artificial Immune Systems a Bio-Inspired Optimization Techniques and its Engineering Application," *IGI Global (2009),* vol. 193, pp. 22--48, July 2009.

5.24 K. L. Kroeker, "Biology-inspired networking," *Comm. ACM*, vol. 54, no. 6, pp. 11-13, November 2011.

5.25 W. Marwan, "Amoeba-inspired network design," *Science*, vol. 327, pp. 419-420, January 2010.

5.26 B. Biswas, S. Bhowmick, M. Biswas, A. Dey, S. Roy, and S. K. Sarkar, "Simulation of Capacitive Multilayered Bio-Sensor for Sensing Biomolecules," *in Proceedings of International Conference On Electrical, Electronics, Computers, Communication, Mechanical And Computing (EECCMC- 2018)*, Nagpur, India, 2018, pp. 701–703.

5.27 R. R. Sahoo, A. R. Sardar, M. Singh, S. Ray, and S. K. Sarkar, "A Bio Inspired and Trust Based Approach for Clustering in WSN," *Natural Computing, Springer*, vol. 15, no. 3, pp-423-434, September 2016.

5.28 T. Nakagaki, H. Yamada, and A. T'oth, "Maze-solving by an amoeboid organism," *Nature*, vol. 407, pp. 470, September 2000.

5.29 T. Nakagaki, H. Yamada, and A. T'oth, "Path finding by tube morphogenesis in an amoeboid organism," *Biophysical Chemistry*, vol. 92, pp. 47–52, Aug. 2001.

5.30 T. Nakagaki, M. Iima, T. Ueda, Y. Nishiura, T. Saigusa, A. Tero, R. Kobayashi, and K. Showalter, "Minimum-risk path finding by an adaptive amoebal network," *Physical Review Letter*, vol. 99, no. 068104, pp. 99-06, September 2007.

5.31 A. Tero, K. Yumiki, R. Kobayashi, T. Saigusa, and T. Nakagaki, "Flow-network adaptation in physarum amoebae," *Theory in Bioscience*, vol. 127, no. 2, pp. 89–94, Jun. 2008.

5.32 P. Cortes, J. Garcia, J. Munuzuri, and L. Onieva, "Viral systems: A new bio-inspired optimisation approach," *Computers & Operations Research*, vol. 35, no. 9, pp. 2840–2860, September 2008.

5.33 A. Tero, S. Takagi, T. Saigusa, K. Ito, D. Bebber, M. Fricker, K. Yumiki, R. Kobayashi, and T. Nakagaki, "Rules for biologically inspired adaptive network design," *Science*, vol. 327, no. 5964 pp. 439–442, Jan. 2010.

5.34 Y. Afek, N. Alon, O. Barad, E. Hornstein, N. Barkai, and Z. Joseph, "A biological solution to a fundamental distributed computing problem," *Science*, vol. 331, no. 6014, pp. 183–185, Jan. 2011.

5.35 L. Liu, Y. Song, H. Ma, and X. Zhang, "Physarum optimization: A biology-inspired algorithm for minimal exposure path problem in wireless sensor networks," *in Proceedings of IEEE International Conference on Computer Communications*, Munich, Germany, 2012pp. 1296–1304.

5.36 V. Bonifaci, K. Mehlhorn, and G. Varma, "Physarum can compute shortest paths," *Journal of Theoretical Biology*, vol 309, pp. 121–133, January 2012.

5.37 H. Zhang, and L. Liu, "An advanced distributed physarum optimization algorithm for minimal energy-efficient tree in wireless multi-hop networks," *in*

Proceedings of IEEE Wireless Communications and Networking Conference, Istanbul, Turkey, 2014, pp. 2360-2365.

5.38 L. Liu, Y. Song, H. Zhang, H. Ma, and A. V. Vasilakos, "Physarum Optimization: A Biology-Inspired Algorithm for the Steiner Tree Problem in Networks," *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 818-831, March 2015.

5.39 Y. Sun, and S. Halgamuge, "Fast algorithms inspired by Physarum polycephalum for node weighted steiner tree problem with multiple terminals," *in Proceedings of 2016 IEEE Congress on Evolutionary Computation*, Vancouver, Canada, 2016, pp. 3254-3260.

5.40 C. Gao, M. Liang, X. Li, Z. Zhang, Z. Wang, and Z. Zhou, "Network Community Detection Based on the Physarum-Inspired Computational Framework," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1916-1928, Nov. 2018.

5.41 Y. Liu, C. Gao, and Z. Zhang, "Simulating Transport Networks with a Physarum Foraging Model," *IEEE Access*, vol. 7, pp. 23725-23739, Feb. 2019

5.42 D. M. Warme, P. Winter, and M. Zacharisen, "The GeoSteiner 5.0.1 package", http://www.geosteiner.net/geosteiner-5.0.1.tar.gz (accessed September 13, 2019)

5.43 C. J. Alpert, "The ISPD98 Circuit Benchmark Suite," *in Proceedings of International Symposium on Physical Design (ISPD '98),* Monterey, CA, 1998, pp. 80-85.

5.44 R Sardar, R Mondal, and T Samanta, "Geometry Independent Wirelength Estimation Method in VLSI Routing", *in Proceedings of 26th International Conference on VLSI Design and the 12th International Conference on Embedded Systems,* Pune, India, 2013, pp 257-261.

CHAPTER 6

OPTIMIZATION IN CLUSTER BASED WSN & IOT ENVIRONMENT USING SWARM INTELLIGENCE

6.1. Introduction

A set of viable nodes are spread over an area which communicate with each other to obtain the necessary information required forms the Wireless Sensor Network. The fields of disaster prediction and analysis, environmental study and health system has implemented varied implication of WSNs. This is because of the fact that WSNs are highly adaptive, obtains better result in less time and are reliable [6.1]. As routing algorithms are embedded in WSNs their output is more reliable and efficient and enables proper continues stream of data [6.2].

The nodes spread over the solution space are places together and this process is called clustering. It is implemented to deal with the issues of very large scaling, limited power supplies and reducing the power intake. Now the two variety of nodes [6.3] - [6.6] namely the Head node and the Sensor nodes are divided into from the cluster. To control the sensor nodes and to develop a line of communication between them the Head nodes come into play.

As with the advancement of technology the algorithms have become more efficient, this has made clustering [6.7] - [6.8] to be efficiently used to transfer data between nodes. The data can either be delivered to the Base Station(s) in the given described topology or it is intercepted by a neighboring node of the cluster or by another cluster in an Ad-Hoc topology. The 802.11 standards [6.9] is divided into three separate sections to establish a line of information exchange. The 802.11n is a modern and efficient protocol however it utilises a lot of power and bandwidth, hence is limited in WSNs. The 801.11a is old and is not viable enough to run the communication lines, so the 802.11b is the only viable option.

Due to the varies size of the solution space and number of initial nodes in them meta heuristics is the apt way to find a viable solution. This genre of algorithm imitates the nature to search over a solution space and provide the best possible solution.

ACO or Ant colony optimisation [6.10] - [6.11] follows ants which looks for food leaving behind pheromone trail for other ants to follow. So, if a path has more amount of this pheromone that means more ants have followed this path and other path having less pheromone indicates a shorter route. This process of determining the shortest path has proven to be very efficient for routing in WSN however it falters when nodes get mobile.

The PSO algorithm [6.12] - [6.16] is an efficient algorithm that employs particles in the search space to develop a viable solution**.** Its location determines the Candidate Solution for the respective particle**.** Every node tries to obtain a better candidate position hence the cluster as a whole strives toward a more efficient solution. At first the positions are opted by a random function. After their positions being initialised, they strive for opting a better one which leads to a more efficient solution. This acted as a necessary evil because this dynamic nature helps opt out the optimised solution however this unguided random function limits the output for varied topography.

Meta-heuristics have found its varied application in WSN. This amalgamation results in viable output to existing real life problem which exist because WSN is dynamic in nature. Data transmission have become more efficient after meta-heuristics are being incorporated in the LEACH [6.17] protocols. ACO implemented clustering algorithms has shown better output with WSN however the limitations faced by it is because of the dynamic nature of WSNs. LEACH and OEERP [6.14] are incorporated in PSO based clustering. This works efficiently within the cluster to provide viable output however when it is employed for inter cluster communications it cannot provide viable output.

To offer more optimal usage conditions in the WSN environment, by communicating technologically communicable products with each other, important opportunities were unearthed. And this occurred with the rise of WSN technology and such products.

The Electronics and Telecommunications Technology industries used these opportunities to develop the Internet of Thing, a networking-enabled and computational platform. The industries created this to provide for an exclusively-connected breed of devices capable of parameterized awareness. The Internet of Things, or IoT, [6.18] - [6.20] provides a platform in which the IoT-enabled devices can communicate with other devices. With the development of an access methodology, it also makes the mass control, deployment, and monitoring of those devices possible and the unique infrastructure of IoT interfaces to this methodology. This prototype is an assembled network of physical objects, namely "Things," which is embedded with sensors, electronics, and connectivity. It achieves service and value by communicating with the operator, manufacturer, and other devices with the help of data pockets and by advanced network protocols. The realization of pedestrian mass, the applications industries use, and even governments can dawn with the interconnection of billions of such devices to the internet. This can bring in a set of proper usage metrics and vital optimizations never been seen before if IoT can prove this prototype.

A sheer infrastructure needs a storage solution that is viable and reliable, and when required, people can access it with proper permissions. For this, the only appropriate solution is Cloud Computing, a software-defined model. Cloud Computing [6.21] - [6.22] works alongside resources that are geographically distributed, and it enables on-demand and favorable usage of a shared pool of storage resources or configurable computing. These geo-replicable resources and the distributed topology makes combining cloud services with the IoT infrastructure highly serviceable. The user's indulgence in having access to these on-demand services is an evident benefit of this integration.

To fetch and store the necessary information, a large number of devices must access the cloud service. Therefore, it can be hypothesized that this will result in a huge escalation in the coinciding wastage of leased bandwidth as well as the usage. To tackle such issues, a new platform delivering services and web applications to end-users by extending the cloud platform was developed. This was termed Fog Computing [6.23], a wireless load distributor. It distributes in the IoT model, a computational load of devices. This Fog Computing offers a useful load balancing technique [6.24]. The distributed operation blends with the IoT infrastructure and this technique is solicited by the nature of the operation. The proximity that Fog Computing maintains to the end-users as well as its support for mobility forms the distinct Fog characteristics. The IoT applications that require predictable, real-time latency is supported by it. Complexities may occur within the infrastructure due to the IoT field's ever-evolving adoption.

This chapter presents a modern efficient process inspired from PSO and ACO with levy flight to enhance the cluster creation along with its intercommunication. The specific Constricted approach of PSO creates a limiting factor which keeps the velocity of the particles in check unless the cluster is generated. ACO with a levy flight feature enables to generate a viable route both among the nodes and the clusters. In the next section of the chapter deploys a new hybrid algorithm that is built on Constricted PSO and DABA, that depends on the collective performance of decentralized agents within the IoT environment. This algorithm helps in efficient data routing as well as management of IoT devices. It is also used for the Dynamic Graph Partitioning algorithm that helps in balancing loads within Fog Servers.

6.2. Literature Review

A scheme for higher-level nodes' Self-Organization Management Protocols has been suggested [6.25] to compete with multi-hop form hierarchical clusters, introduces the "20/80 Rule" for determining the ratio of headers to member nodes, and develops a new cluster-based routing protocol that integrates inter-cluster on-demand routing and intra-cluster table-driven routing for further consumer applications in Sensor Networks.

Chi-Tsun et al. presented [6.26] a decentralized clustering algorithm based on social insect colonies mainly to increase the network lifetime. The results of this approach are compared with Low-Energy Adaptive Clustering Hierarchy, Power-Efficient Gathering in Sensor Information Scheme and Power Efficient Data Gathering and Aggregation Protocol in this paper.

Liao et al. suggested an optimal configuration for load-balanced clustering in WSN with distributed self-organization [6.27]. The goal of this research is to investigate a Balanced Clustering Algorithm with Distributed Self-Organization for Wireless Sensor Networks that can handle stochastic sensor node distribution.

The suggested clustering scheme, Improved WCA, in the research article [6.28], when compared to existing algorithms, can efficiently optimize the cluster head load, improve the clustering algorithm can improve the premise, in the stability of the clusters reduces the overhead while maintaining the characteristics of WCA Algorithm.

Yadav et al. [6.29] focused on optimum clustering for UWSNs using any of the acoustic, free-space optical (FSO), or electromagnetic (EM) wave-based communication protocols in this research. A sensor node energy dissipation model for FSO and EM wave-based communication was presented, and it was compared to current energy dissipation models for acoustic-based communication, which shows this algorithm has low energy consumption and better optimal clustering.

Dargie et al. suggested a method [6.30] based on computing a binary adjacency matrix that represents the neighborhood of a certain node. For stationary nodes, this method deterministically identifies the cluster heads and associated child nodes for each round, obviating the necessity (and expense) for nodes to announce their candidacy for cluster heads and, after the cluster heads are elected, to declare their membership.

Kulkarni introduced PSO in WSN [6.31] and discusses how PSO can be used to address clustering issues in WSN. The main objective of this work is to give a taste of PSO in WSN for researchers to work on in the future.

Ali et al. in 2021 [6.32] developed a cluster head selection algorithm in WSNs, incorporating rank-based clustering which reduces energy consumption and enhances network lifetime dynamically by considering residual energy and communication distance. It outperforms PSO by approximately 25%.

Sahoo et al. [6.33] in 2021 explored meta-heuristic approaches like genetic algorithms and whale optimization (WOA) are emerging as efficient clustering methods to decrease energy consumption and extend WSN lifespan. He observed a comparative analysis where WOA surpassed that of differential evolution, GA, particle swarm, and grey wolf optimization.

Chen et al. [6.34] in 2022 introduced a new distributed 2-hop cluster-routing protocol, formed clusters within a 2-hop range and selecting energy-efficient cluster heads where multiple chains connect cluster heads closer to the base station, optimizing both intra-cluster and inter-cluster communication for prolonged network lifespan.

Savazzi et al. suggested efficient and accurate methods [6.35] and tools for the optimization of wireless cloud networks and the prediction of virtual coverage using a stochastic model in the industrial domain to identify the imperfectly positioned nodes and inaccuracies in *3-D* layout. They also decrease the loss over short range communication between clusters of devices in the cloud improving reliability on cloud networks composed of weakly connected clusters of devices by taking account of the relay deployment problem.

W. Yichuan et al. [6.36] reported a novel model for the game theory that results in a significant reduction in threats from DDoS attacks over cloud networks. In the reported method they show the attacker tends to occupy as much bandwidth resources as possible by minimizing IoT attack devices while the defender tries to minimize the rate of false alarm considering both the attacker and the defender are rational and strategically dynamic.

Siddiqui et al. [6.37] worked on a unique tool called Smart Meter to develop an effective filter to limit the disorder elements within a range of energy consumption. The reported work to limit the energy consumption of disorder devices in a cloud network and cloud of things find a way to drastically reduce the overall energy consumption and hence increase the lifespan of the devices in the cloud network.

Sajid et al. highlighted the security issues in the industrial SCADA system [6.38] in the IoT-cloud environment. They also reported that the future techniques such as cloud computing integration of complex architecture with the technologies of IoT, Mobile Wireless Sensor Networks are open to public revealing security issues. They emphasize the issues that belong to the security of those future technologies through this paper.

Pawlick et al. conceived a novel concept of trust or strategic trust [6.39] at large using game theory to provide a safe room to the administration of the cloud network. The trust signals are analyzed by the devices in a cloud that is already connected with a vulnerable cloud network. The accuracy of the trust signal passed among devices of safe cloud infrastructure will ensure the overall safety of the cloud network.

Nandan et al. [6.40] in 2022 explored IoT-enabled WSNs which demand energy-efficient solutions due to limited battery power. Researchers employed a genetic algorithm for cluster head election, optimizing node density, distance, energy, and capability which integrates movable sinks to reduce communication distance and dynamic sensing range adjustment to minimize energy consumption.

Ali et al. [6.41] in 2022 conducted a comprehensive survey to analyze data collection proposals, revealing consistent trends and identifying nine novel contributions predominantly in models and algorithms in IoT, WSN, and Sensor Cloud.

M. Chiang et al. [6.42] discussed the use of fog computing in the future technologies of electronic automobiles and the challenges related to providing an efficient and agile architecture packed with a software application responding enough to encounter fundamental problems in electronic automobile design.

Tang et al. [6.43] reported fog computing architecture in big data analysis for smart cities in 2017. They also suggested the concept of edge computation in a network under the shade of fog architecture to satisfy the low latency in the network designed for smart cities. The suggested architecture is responsive to neighborhood-wide, community-wide, and city-wide levels with high accuracy.

An effective and efficient architecture to analyze the large-scale data used in smart city applications were delivered by J. He. et al. [6.44]. They utilize dedicated multi-tier fog infrastructures for computing purposes built up with ad-hoc facilities in order to get high accuracy and low latency for large-scale data analysis.

Y. A. Chen et al. [6.45] reported the use of fog task model to develop a unique way for load balancing without the knowledge of the schedule of individual tasks making the schedule problem for server level instead of the device level. They develop a mobility prediction algorithm at first to support all the later works and to achieve a smaller number of missing deadlines and minimum runtime for connected car systems based on fog computing infrastructure.

E. Batista et al. reported [6.46] a new approach of network programmability over the fog infrastructure for IoT devices in 2018 to help overcome the problems related to load balancing in Fog infrastructure. The approach using Software-Defined Networks optimizes the set of sensors in IoT management services by reducing the network link failures in IoT gateways and improves the overall efficiency and hardware stability of the Fog infrastructure in IoT devices such as Mobile, Desktop, Web etc.

R. Beraldi et al. [6.47] presented a novel protocol to achieve load balancing in fog computing infrastructure in 2019. The reported method is a centralized protocol to select a node having the least load at a point from a set of randomly selected nodes to assign the jobs waiting in the queue. This solution strategically chooses the worker for a particular job instead of using a random selection process that may increase the overhead inside the fog infrastructure on the failure of job assignment.

Maswood et al. developed [6.48] a unique approach of load balancing at the network and the server level considering an integration of the advantages of both fog and cloud environments. The reported work helps not only minimize the resource cost by load balancing in fog servers but also reduce bandwidth costs in CPU and path establishment from cluster point to server using Mixed-Integer Linear Programming.

A. J. Kadhim elaborated [6.49] a proactive load balancing technique that helps in reduction of mitigated tasks assigned to cloud servers significantly by using software defined network in fog computing infrastructure supported by parked vehicles. The suggested work supports the load balancing process by utilizing the parked vehicles as available fog computing nodes in the infrastructure and prioritizing deadline-specific tasks.

Angelin et al. [6.50] in 2022 Integrated fog-cloud-IoT architecture to optimizes WSN's ecological monitoring potential where edge computing augmented network efficiency, overcoming distributed computing drawbacks.

Dorigo et al. [6.51] stated ACO as a completely new type of meta-heuristic. The ACO meta-heuristic was suggested with the possibility that it may help in the ongoing research in this growing field, along with the development of new applications with ease.

In 2003, K. M. Sim et al. [6.52] promoted an idea of balancing the load in the network by going through the difference regarding routing information, routing overhead and adaptivity between ACO and traditional routing algorithms. The issues regarding stagnation in ACO algorithms were discussed, and analyzed over the state-of-the-art approaches on mitigation of the stagnation.

A novel ACO algorithm (Three Pheromones ACO, TPACO), where three types of pheromones instead of one type are used to find the solution with an efficiency, was reported by Lee et al. [6.53]. An ant covers the sets of fewer sensors with the help of local pheromone, one of the three pheromones whereas other two global pheromones optimize the number of required active sensors per Point of Interest, and form a sensor set equal to the number of active sensors selected by an ant using former pheromone. Probabilistic sensor detection models and heterogeneous sensors in continuous space techniques are also reported in this paper to solve the EEC problem.

John et al. [6.54] demonstrated one of the nature-inspired algorithms, ACO for clustering in MANET in 2014. ACO is used to find shortest path in routing problems with other solutions of complex dynamic problems e.g. Traveling Salesman Problem, Scheduling, Network Model Problem etc.

A noble approach using ACO algorithm [6.55] to find an optimal route approaching better performance and faster convergence in WSNs to transmit data is reported by Sharmin et al. with the help of remaining energy and the mobility of the nodes.

Sharmin et al. [6.56] in 2021 suggested a secure bio-inspired WSN routing protocol, integrating trust evaluation and ACO for energy-efficient, secure paths in IoT environments.

A study [6.57] described a metaheuristic method based on evolutionary computation and swarm intelligence notions, as well as the basics of microbat echolocation. The goal is to solve mono- and multi-objective optimization issues with brushless DC wheel motors.

A. Rekaby et al introduced [6.58] the "Directed Artificial Bat Algorithm" (DABA) is a novel bio-inspired swarm algorithm in this study. The echolocation behavior of bats inspired this method. The DABA algorithm employs the same approach to tackle optimization issues. DABA is a computer method that provides a computational representation of actual bat hunting principles.

Afrabandpey suggested a fundamental idea [6.59] is to employ chaotic sequences to initialize virtual bats' parameters rather than random initialization. The influence of chaotic sequences on the Bat algorithm's convergence behavior was investigated in this work.

In their work [6.60], Y. Saji et al. suggested a novel adaptation of the Bat Algorithm to solve the Travelling Salesman Problem (TSP) which is known as an NP-Hard Problem. This is the first adaptation of the BA to solve a discrete problem such as the TSP problem.

The literature in [6.61] suggested to solve the WSN with weighted mesh clients and integrate it with three local search schemes using the bat-inspired algorithm. A dynamic probabilistic local search method is applied to choose the optimal search model dynamically during the algorithmic process.

In his work [6.62], G. Wang et al. demonstrated a multi-swarm bat algorithm (MBA) for global optimization, in which different swarms with their own parameter settings explore the specified territory concurrently in the MBA approach, and they can communicate information via the immigration operator. This arrangement can achieve a nice balance between global and local search in this way. The finest individuals from each swarm are then gathered by a selection operator to form the elite swarm

Parija and Sahu [6.63] dealt with an optimal design for cellular network to minimize the total location management cost using the Bat Algorithm. The Bat method is developed in this research for tackling the complicated issue of minimizing the overall mobile location management cost.

In the research article [6.64], Ambulance vehicles are routed using the Bat Algorithm to create a network of quick response time in case of an accident through the shortest path. The position of the accident and the accident is fed through the Bat

Algorithm, Roads are simulated as the nodes, and the Bat Algorithm gives the shortest path to the accident place as the output.

Pitchaimanickam in 2021 [6.65] integrated LEACH-C with bat-inspired echolocation techniques to balance exploration and exploitation, selecting cluster heads based on residual energy and node distance, resulting in prolonged lifetime and reduced energy consumption in Wireless sensor networks.

Wen et al. [6.66] in 2022 employed cellular grids to organize the monitoring area, construct a bipartite graph model, and utilize the vampire bat algorithm for matching where improved virtual force optimization enhanced coverage and reduced moving distance.

6.3. Clustering optimization using Constricted PSO and ACO with Levy Flight

6.3.1 ACO based Clustering in WSN

Ant Colony Optimization is created based on the activities of ants to find and gather food through pheromone trails. The area in which the ants search for food is synonymous to the given search space. The ants are synonymous to the software agents which strive to provide the viable solution. The foraging ants leave behind pheromones so that other ants can follow the path taken. The more the ants walk on a certain path the more volatile pheromone present in the path leading to an optimal solution.

Based on the given algorithm [6.10] a group of m ants or agents are employed to create a solution from a given search space components, $C = \{ C_{\mathrm{ij}}\},\ I = \{1, \dots, n\},\ j = \{1, \dots, |D\mathrm{i}|\ \},\ C = \{ C_{ij}\},\ I = \{1, \dots, n\},\ j = \{1, \dots, |D\mathrm{i}|\ \}$. An empty solution set $s^p = \emptyset$ is initially considered to begin with the framing of a solution set. Then at every iteration s^p is updated with the current viable solution from the neighboring nodes $N\ (s^p) \subseteq C$. This procedure is taken as path creation of the graph $G_{\mathrm{C}}\ (V, E)$. These paths generated in the graph G_{C} are created as we go along with the solution, it also creates the set N (s^{p}) based on the partial solutions. The selection of a partial solution from N (s^{p}) is done using a probability at each iteration where the rule given in (6.1).

$$\mathrm{p}_{(c_{ij}/s^{p})} = \frac{\tau_{\mathrm{ij}}^{\alpha}.\eta_{\mathrm{ij}}^{\beta}}{\sum \tau_{\mathrm{ij}}^{\alpha}.\eta_{\mathrm{ij}}^{\beta}}\ , \forall\ c_{ij} \epsilon\ N \tag{6.1}$$

where, the pheromone value of t_{ij} and the heuristic value of η_{ij} are related with the component c_{ij} , α and β are positive real parameters which defines the relative importance of pheromone with respect to heuristic data. This pheromone value is very crucial to the determination of the best solution path as its value is directly proportional to the efficiency index of the path. This is obtained in (6.1) by diminishing all the pheromone values through pheromone evaporation, and in (6.2) by increasing the pheromone levels associated with a chosen set of good solutions.

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho \sum_{i=1}^{m} F(s), s \in s_{upd} | c_{ij} \in s \tag{6.2}$$

where S_{upd} is the set of solutions that are used for the update, $\rho \in [0,1]$ is a parameter called evaporation rate, and $F : S \rightarrow R + 0$ is a function such that $f(s) < f(s') \Rightarrow F(s) \geq F(s'), \forall s \neq s' \in S$. F(·) commonly referred to as the fitness function.

As evident from the equations, ACO [6.67] is an iterative algorithm that gives out the time required to sketch out the best possible route. Hence it is derived that the number of iterations is directly proportional to the viability of the output. The only drawback arises is when time and number of agents to scourge the solution space is limited along with the dynamic nature of the nodes, it falters to provide a viable output.

6.3.2 PSO Based Clustering in WSN

To deal both discrete and continuous data set the PSO or particle swarm optimizations is used. It is a data quantity determined algorithm [6.12] that enables particles to scourge the solution space to find a viable output. The coordinates of these agents demarcate the solutions of the optimizations issue. To further optimize the solution the velocity of the particles is updated with respect to position.

The algorithm starts by deploying particles using a random function in the designated solution space $\Theta' \subseteq \Theta$. The particle velocities are stored with Θ' or *zero* or by using a random function. This helps to keep the particles within the search space. Within the main loop the particle coordinate and their velocity are updated using (6.3) and (6.4) until a limiting condition is met.

$$\vec{v}^{\,t_1+1} = w\vec{v}^{\,t_1} + \varphi_1\vec{U}^{\,t_1}(\vec{b}^{\,t_1} - \vec{x}^{\,t_1}) + \varphi_2\vec{U}^{\,t_2}(\vec{l}^{\,t_1} - \vec{x}^{\,t_1}) \qquad (6.3)$$

$$\vec{x}^{\,t_1+1} = \vec{x}^{\,t_1} + \vec{v}^{\,t_1+1} \qquad (6.4)$$

where w represents the inertia weight, φ_1 and φ_2 are acceleration coefficients and $\vec{U}^{\,t_1}$ and $\vec{U}^{\,t_2}$ are two $n \times n$ diagonal matrices where the main diagonal elements are selected using a uniform random function within the interval $[0,1]$. With each iteration these matrices are recreated and updated. The vector $\vec{l}^{\,t_1}$ is denoted as the neighborhood best, and $\vec{x}^{\,t_1+1}$ is the best coordinate obtained by any particle within the vicinity of the particle p_i, that being, $f(\vec{l}^{\,t_1}) \leq f(\vec{b}^{\,t_1})\, \forall p_j \in N_i$. Considering the values of w, φ_1 and φ_2 suitably kept under parameter, the particle's velocities do not increase to infinity and best fitness solution can be achieved.

The updation algorithm of PSO [6.13], [6.14] enables the intercommunication between the clusters. This algorithm also updates the particle velocities for every iteration. This procedure is synonymous to the WSNs dynamic movements and this algorithm efficiently provides an output. However, the PSO fails miserably when number of clusters increase. During the starting phase of PSO the particle coordinates are randomly assigned. The dynamic nature, and distances among the clusters of WSNs, the change in coordinate vector for approaching optimization does not lead to the target cluster therefore a more guided approach is necessary for the multi node optimizations.

6.3.3. Cluster formation and Data routing using C- PSO and ACO-Levy Flight

In WSNs where multi clusters are involved the existing algorithms do not perform as expected. Now these drawbacks if mended would result to better viable outputs. So, to improve the already existing algorithm the implemented PSO-C is implemented which creates cluster during the learning phase, which helps us to generate clusters with better densities. As WSNs are dynamic in nature with their low power supply and fast routing conditions are necessary. These requirements lead to the use of ACO with Levy Flight (ACO-LF). This revised algorithm mimics a Random Walk tendency which reaches to a viable solution more efficiently. In intra-cluster optimization the algorithm has sorted to PSO-C and for inter-cluster optimization ACO-LF is used.

6.3.3.1 Constricted PSO for Cluster Formation in WSN

Constricted Particle Swarm Optimization [6.15] comes into play when the clusters become highly dynamic in nature. When these limiting factors are applied the expansion of cluster creation width decreases. It also reduces the distances between the nodes within the cluster. It thus frees up space for inclusion of more nodes into the said cluster thereby improving and increasing the intercommunication within the cluster. This limiting factor also minimizes swarm explosion. It directly relies on the search results of the previous iteration. χ is the constriction factor, as in (6.5),

$$\chi = \frac{2}{\left[2- \varphi - \sqrt{\varphi^2 - 4\varphi}\right]}, \tag{6.5}$$

where, $\varphi = \varphi_1 + \varphi_2$, $\varphi > 4$, φ being the acceleration coefficients. The constriction factor is put into the velocity equation of conventional PSO (6.3) and changes to (6.6).

$$\vec{v}^{\,t_1+1} = \chi[\vec{v}^{\,t_1} + \varphi_1 \vec{U}^{\,t_1} \left(\vec{b}^{\,t_1} - \vec{x}^{\,t_1}\right) + \varphi_2 \vec{U}^{\,t_2} \left(\vec{l}^{\,t_1} - \vec{x}^{\,t_1}\right)] \tag{6.6}$$

In most cases the value of φ is around the value of 4.1, resulting χ at 7.29. The previous velocities are multiplied by 0.729. [3.65] For stabilising the algorithm these values are only taken while implementing constriction in PSO. The results of the routing are stored in a knowledge table and are utilized to keep the fitness of the agents to point.

6.3.3.2. Constricted PSO Algorithm for Cluster Formation

Step1: Nodes (Distributed Randomly), velocities initialized.

Step2: Calculate the Fitness Function for each nodes, defined by the velocity vector in (6.3)

Step3: Select the new nodes from the initial nodes, based on the Fitness values, computed in the previous step (Current Velocity is defined by the rate at which the particle's velocity has changed)

Step4: New Velocity is evaluated by the Constricted Value as in (6.6)

Step5: Update the new positions by (6.4)

Setp6: If (new fitness value > old fitness value)

Select the node accordingly.

Step7: Select the Local Best node; goto Step2

Step9: The Local Best are used to create the clusters.

Step10: The Global Best of each cluster is selected as the Cluster Head.

6.3.3.3. Levy Flight Based ACO Routing Optimizations in WSN

The ACO-LF has the same features as that of the ACO with a small change in the movement patterns of the ants. To meet the needs of the dynamic topography of WSNs, ACO-LF uses random walk to produce viable output. The value of ρ remains constant in ACO [6.68] however the ants implement the levy distribution to produce efficient results. In random walk the velocities of u and v follow the normal distribution. It is defined by $u \sim N\ (0, \sigma^2)$, $v \sim N(0, \delta v)$, and is combined to produce the Levy flight equation as $L(s) \sim \frac{u}{|v|^{\frac{1}{\beta}}}$. By substituting it in (6.2) the pheromone equation is modified as in (6.7).

$$(t + 1) = (1 - \rho_{i,}) \times \tau_{ij}(t) + \Delta\tau_{ij}(t),\ \rho_{i,j} \sim levy(\beta) \qquad (6.7)$$

where, $levy(\beta) \sim \frac{u}{|v|^{\frac{1}{\beta}}}(\tau_{ij}(t+1) - \tau_{ij}(t)$ and $u \sim N(0, \sigma_{2u})$, $v \sim N(0,1)$

$$\sigma_u = \frac{\tau(1+\beta)\sin(\beta\pi/2)^{1/\beta}}{\tau\left(\frac{1+\beta}{2}\right)\beta 2^{(\beta-1)/2}} \qquad (6.8)$$

6.3.3.4 Levy Flight Based ACO Routing Algorithm

Step1: Initialization of Hello Packets

Step2: Set the initial path of the traversal, to any of Cluster Heads, and set it as the start node

Step3: Initialize the traversal paths.

Step4: If (current node = destination node) Add node to the path

Step5: Else if, the node has a routing table:

Step6: Check if the routing table contains destination node

Step7: Set the current node to the destination node; goes to Step 4

Step8: Sort the node distribution w.r.t the pheromone concentration, defined by β, u and σ_u as in (6.7).

Step9: Select the node based on the pheromone concentration

Step10: Initiate the Random Walk; goes to Step 4

6.3.4. Simulation Setup

The experiments to prove the said algorithm is performed using the INET Framework in OMNET++ 5.0. To implement real life cases very small changes are made in the .ini file. The properties required for the smooth working of the WSN networks are laid down in the simulation area in Table 6.1. To ensure the proper simulation for this algorithm to function the iterations are run for 3400s. Given below are representations of the WSN and clustering in the Cmdenv of the OMNET++ 5.0.

Table 6.1. Simulation Attributes for execution

Parameters	Value	Parameters	Value
Topology	1200 m x 1200 m	Mobility Model	Random Walk
Number of Nodes	600	Maximum Channel Power	2mW
Data Packet Size	4 bytes	Radio Bitrate	1000kbps
Control Packet Size	100 bits	Simulation Time	3400s
Initial Energy Per Node	2.1 J	Simulation Style	Cmdenv-fast-mode

6.3.5. Result and Discussions

The pre clustering and the post clustering results are depicted in Figure 6.1 and Figure 6.2. respectively.

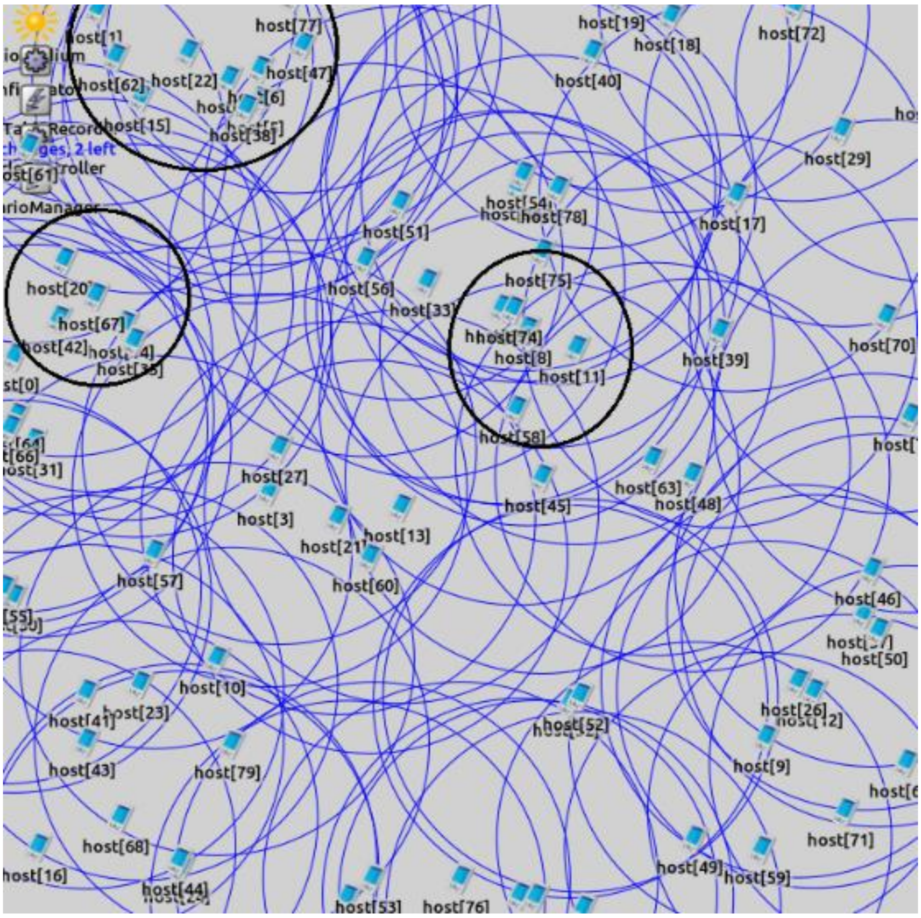

Figure 6.1. Pre-Clustering Scenario.

Figure 6.2 depicts the algorithm run closer to the finishing point of the complete run from inutilization of PSO-C. As the data is not discrete hence cannot be put down into charts. Therefore, the continuous data sets are plotted against given parameters.

In Figure. 6.3 for the unguided clustering and the guided PSO-Constricted Clustering, number of dead nodes plotted versus time. Because the PSO-C based clustering is constrained to a specific dimensional search space and has a skill set to search across, it produces fewer non-functional nodes. This aids in increasing the cluster's concentration and, as a result, enhancing the relay properties for which they were placed in the first place.

The Table 6.2 shows that Hybridised AODV out performs the other algorithms in all parameters of Loss percentage, packet delivery ration in varied topology.

Table 6.2. Comparison over loss percentage, packet received and packet delivery ratio.

Loss Percentage				
Nodes	Topology (m)	Regular AODV (%)	ACO_ AODV (%)	Hybridized_ AODV (%)
20	600 x 600	78.26	72.14	69.13
40	1200 x 1200	88.61	87.57	83.27
60	1800 x 1800	97.64	93.36	89.22
80	2000 x 2000	98.00	96.07	92.12
Packets Received				
Nodes	Topology (m)	Regular AODV	ACO_ AODV	Hybridized_ AODV
20	600 x 600	740	946	1049
40	1200 x 1200	374	422	569
60	1800 x 1800	99	254	367
80	2000 x 2000	68	133	266

Packet Delivery Ratio				
Nodes	Topology (m)	Regular AODV	ACO_ AODV	Hybridized_ AODV
20	600 x 600	0.22	0.27	0.31
40	1200 x 1200	0.11	0.12	0.17
60	1800 x 1800	0.03	0.07	0.11
80	2000 x 2000	0.02	0.04	0.08

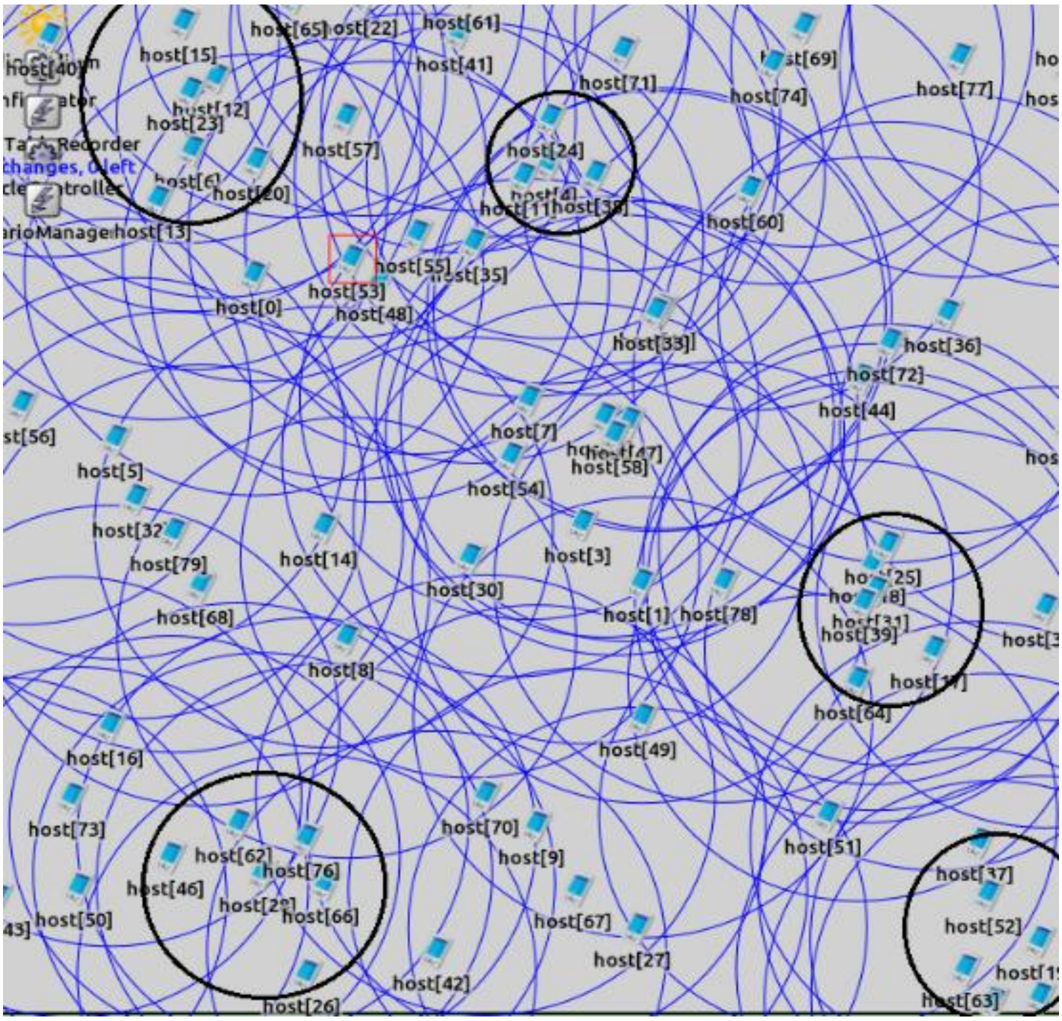

Figure 6.2. Clustering using PSO-C algorithm.

Although both algorithms indisputably converge near the end of the assessment, the performance level of the PSO-C approach is superior in a given length of time for the simulation, which can be demonstrated for better data collecting.

When compared to unguided routing, ACO-based routing, and ACO-LF-based data routing, Figure 6.4 shows the packet delivery ratio. Unguided routing, it suffices to say, fails to perform and is unable to reach the 85 percent dense cluster structures for data routing. The ACO and ACO-LF both break beyond the 80% barrier, however the ACO comes up short due to its repetitive drawback. ACO requires numerous iterations to produce a promising outcome, but it loses to the Random Walk-inspired ACO-LF in the situation of a decaying power source. This outperforms the ACO in terms of searching the complete search area (the topography) for its goal, and hence produces better results when compared to its basic equivalent.

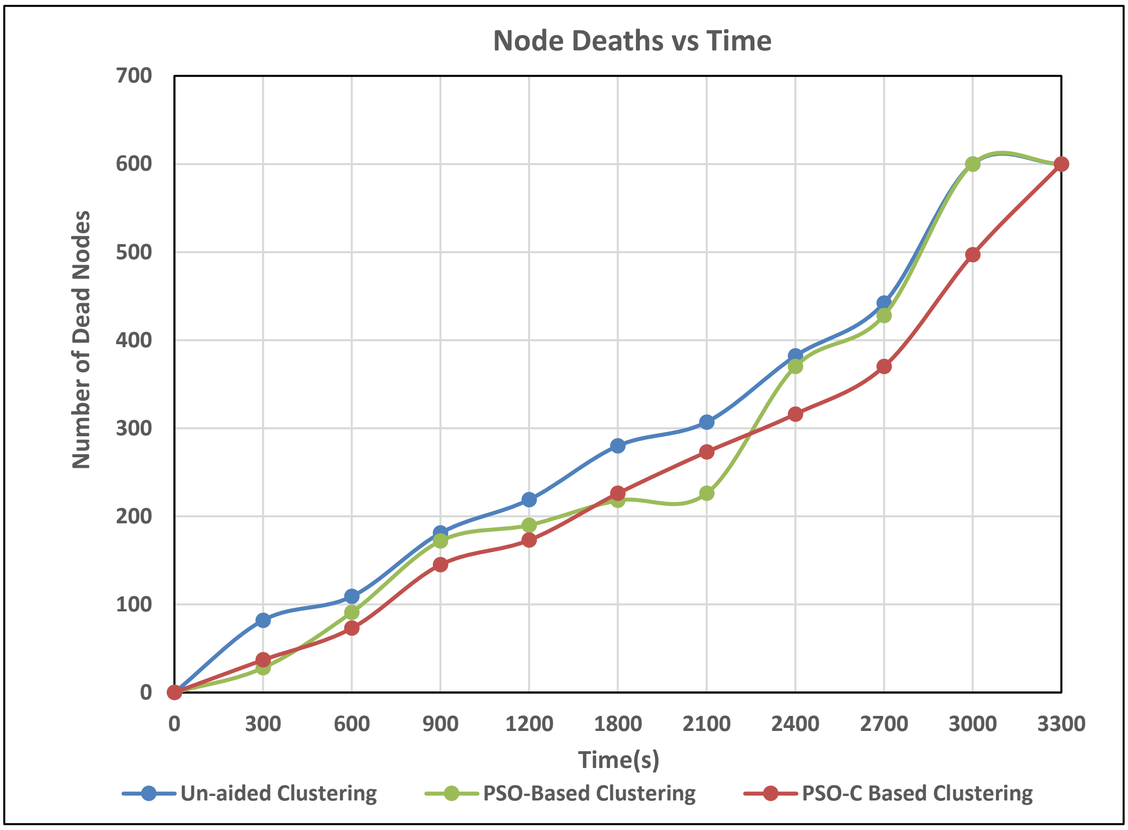

Figure 6.3. Comparison of the algorithms during Cluster formation

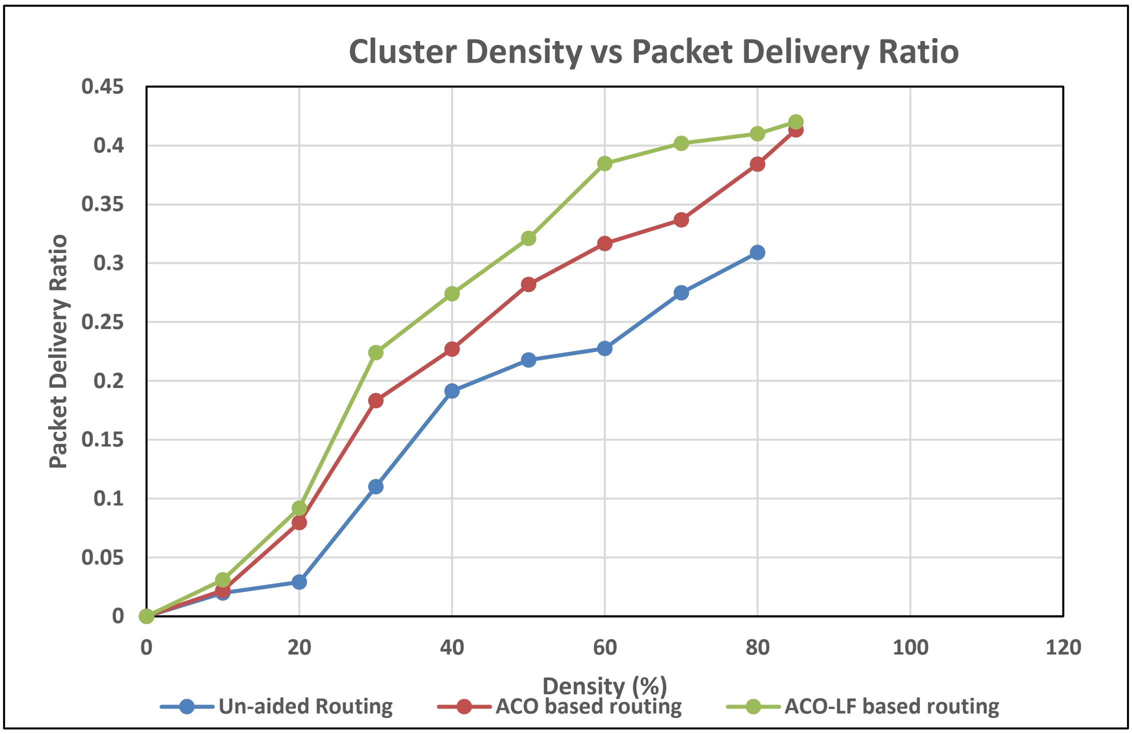

Figure 6.4. Comparison of the algorithms during node Cluster connectivity

6.4. Routing Optimization and Workload Balancing in IoT Environment

6.4.1. IoT Environment

WSNs have become the more potent solution at implementing IoT due to the rapid growth of the sensor technology The WSN communication link with the global network can be separated into three different parts [6.18]. The simplest way lies in creating clusters of WSNs, a single network gateway and therefore it provides access to the global network for transfer of data and on demand access with the only drawback being a little bit of network congestion which depends on the number of concurrent nodes using the same default gateway. Another way of implementing this is using a hybrid infrastructure, which encompasses the overplayed network infrastructure while maintaining its independence and hence allowing for the network connection with the help of a few dual sensor nodes which are capable of expanding the network signals around their neighbours. The last procedure requires the usage of the WLAN structure. In this the WSNs form a rather dese 802.1x access point networks which enable more than one sensor nodes to join and have a news delay-less access to the global network with just one hop.

The biggest drawback that can be found regarding the first technology is that it uses a single-access network gateway. In a situation when there is huge traffic and the resources fail to comply, the access to the global connection will be delayed which will result in heavy loss for the entire infrastructure. The second and the third method of doing this is way better than this one. Due to the presence of multiple external network access routes and gateways, when a single gateway crashes, it is easy to balance the load and hence due to this the network never really crashes.

Requirements for the types of network access provider, depends mostly on the deployment scenarios and the type of IoT enabled devices that are being used. The initial approach solely relies and is heavily modelled, based on the star topography distribution scenarios, whereas the latter two methodologies revolve around the mesh topologies, which is, in fact mainly used in WSN scenarios. Nonetheless, the secondary and tertiary device integration techniques are able to support merely the static network configurations and there is minimal area of change. But it is also to consider that, every newly introduced device waiting to have an access the public network needs time-consuming Routing Information Protocol reprogramming. Therefore, it is quite understandable, that such existing methodologies are not feasible for complete IoT integration, because of the stated obvious loopholes. Contemporarily, more feasible approaches are widely used to develop more modern distributed networks where the IoT nodes are but a connected collection of complete duplex enabled network devices.

The Cloud Infrastructures [6.21] are primarily designed to provide consumers with on-demand services only when a produced process from a user request requires them. Cloud devices can allocate and use resources on-demand that are required, such as server nodes or elastic storage, and thus fulfil a specific task while de-allocating unused resources to a central heap. Because of the nature of Cloud Infrastructures, scaling up or down the operating infrastructure on-demand to meet work requirements is relatively simple and cost effective. Due to the distributed nature of the computing nodes, data fragmentation processes can be dispersed among geographically replicated, distributed nodes enabling parallel process execution.

The simultaneous connections between the respective lowest base level nodes and the supplied cloud servers in IoT are taxing on both the nodes' intrinsic resources and the system's overall bandwidth consumption. As a result of these advances, the Fog Servers, a new iteration level that sits between the lowest base level and the highest cloud level, were quickly created. The primary serfdom of this level is geared toward load management through packet filtering and the establishment of external connections to the cloud for data transfer. A generic Fog Server functions by associating with any IoT capable base devices in its reception proximity, allowing the node to be assigned a priority queue position, which, depending on the resource regulation algorithm in place, will assign the device a proper gateway address, allowing the node to authorize an uplink with the Cloud Domain. This serves two significant purposes. To begin with, bandwidth wastage is significantly reduced due to the reduction in the simultaneous construction of uplinks with the Cloud Do-main. Furthermore, because the total number of valid devices establishing an uplink with the Cloud Domain is dramatically decreased, it helps to reduce total power consumption at the IoT-enabled device level.

When two or more machines or clusters [6.7] communicate via Fog servers in an IoT context, the connections involved in the resource allocation table can be represented in graph. Isomorphic graphs are generated when the connections remain identical after reassigning subjugated connections, resulting in balanced and improved network fluidity when modelling isomorphism inside machine networks. The graph partitioning [6.8], [6.69] problem is stated in the form of a graph $G = (V, E)$ with V vertices and E edges, where G is partitioned into smaller entities with particular properties. Good partitions can be defined as those with the fewest possible edges between the separated components. If graph G_1 is isomorphic to G_2, it implies that there is a matching edge-preserving vertex. Mathematically, bijection $f: V_1 \rightarrow V_2, u — v$ in E_1 $iff\ f\ (u) — f\ (v)\ in\ E_2$, where $u — v$ is an edge in G_1 and in G_2 $f(u) — f(v)$ is an edge. This concept of properly configured, complete network-enabled Fog server gave rise to an efficient IoT paradigm, although with certain changes and adaptations, to modify the paradigm depending on the deployment scenarios.

6.4.2. Optimization in IoT Environment

IoT infrastructures are in high demand causing an increase in the load in servers and heightening itself beyond the boundaries. Therefore, data synchronization and connection will get more importance. Such complexities and complications may negatively affect the Cloud and Fog Infrastructures.

The primary problems include the legitimate entrustment of Fog Server nodes and in the complex network mobility of these IoT-based devices, instantaneous manipulation of nodes. Although normal load balancing is adequate, it degrades noticeably when dealing with increased traffic and adverse node positioning changes. Currently, existing infrastructures are viable, despite the fact that they lack proper escalation, require optimization at every level in the IoT infrastructure, and are essential for convenient and efficient management.

The reported IoT infrastructure is divided into three distinct intrinsic sections, in order to make it more constrained and limited. The base portion is responsible for incorporating a number of clusters of IoT-connected devices. This segment has a considerable volume of network overhead, resulting in the absorbance of the highest power frequency. Fog servers, which serve as an intermediary layer between the nodes, IoT clusters, and the Cloud Servers, make up the second level of infrastructure. These Fog servers are mostly used for connection filtering, allowing for precise treatment of the bandwidth that can be obtained. The gateways, which are assigned to each node, establish a link with the Cloud Server and these Fog servers apportions them when a certain node is nearby and recover the priority to approach or reach the resource. The final and highest level includes Cloud Server nodes that are geographically distributed, and whose primary goal is to conserve data enumerations with geographically distributed redundancy and to govern access, based on the public key infrastructure as and when needed.

6.4.2.1 Hybrid metaheuristic approaches in intra IoT Clusters

In each device, the 2.4 GHz receptors permit intra-cluster communications at the base level of the presented framework. The suggested metaheuristic DABA is employed to allow for smooth and drop-free communication between nodes. This is because of the already overcrowded *2.4* GHz spectrum and a large number of nodes. In DABA, n bats, and their positions x_i and velocities v_i are being updated where the fitness of each particle is calculated with the help of frequency (f_i) which is multiplied with the directed bat's wavelength (λ). The pulse increase factor *(r)* and the amplitude decay factor *(A)* manipulates the pulse frequency. The right updating of the amplitude (A_i) and the pulse rate (r_i) balances the tendencies of exploration and exploitation of each. The updated velocities v_i and locations x_i are as follows:

$$F_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{6.8}$$

$$V_{\mathrm{it}} = v_{\mathrm{it}}-1 + (x_{\mathrm{it}} -1 - x^*)f_{\mathrm{i}} \tag{6.9}$$

$$X_{\mathrm{it}} = x_{\mathrm{it}}-1 + v_{\mathrm{it}} \tag{6.10}$$

where β = random vector that is drawn from uniform distribution [0, 1]. During the iteration, the pulse emission rates and the amplitude undergoes particular changes, and among n bats within the population, the current best solution x^*
can be attained.

The DABA employed, in this case, was controlled by a single node, which was also used to choose the targets. By creating a regulated process of broadcasting Hello packets, the virtual mapping of Ad Hoc On Demand Distance Vectoring Protocol is done. Using DABA, the Route Request (RREQ) messages are transmitted equally from the source node across the full mapped topology. Here the Timestamp values collected while traversing between neighborhood cluster nodes help generate the frequency value (*fi*). If a node intercepts an RREQ message, it sends to the source node a Request Reply (RREP) message, which generates a path to destination from the source. A suitable path to the destination is randomly sought utilizing the produced Neighbor table and Encapsulation that is based on Location for probabilistic behavior using DABA's Pulse increase factor and Amplitude decay factor. In the IP header of the RREP message, this probabilistic technique is mapped to the Time-to-live (TTL) field to calculate hop counts.

Through the accurate spread of IoT-capable devices and the maintaining of an adequate cluster density using Normal Distribution, Inter Cluster communication is achieved. The Normal Distribution facilitates in uniformly dispersing the nodes across the infrastructure. This assists in the acquisition of a situation in which IoT devices are hightailed in a city or an urban area. The function $\emptyset(x) = e^{\frac{(-\frac{1}{2}x^2)}{\sqrt{2}}}$ is used to visualize the node positions and modify them, resulting in optimum efficacy. Inter cluster communication intends to dynamically search for cluster heads. It also maintains communication by using a PSO-based routing method with velocity as the critical factor, as Constricted PSO. The suggested routing algorithms perform significant package transfer using swarm intelligence. This employs infrastructure gateways that are assigned to specific devices and adjust to their presumed position.

At the lowest level, the exchanges in surplus packet among IoT nodes in close proximity reduce the overall system's efficiency. As a result, issues with bandwidth waste arise as a result of counterproductive packet flooding, which causes the network to back up, preventing proper data channeling. C-PSO and DABA's hybridization for density control and increased packet delivery mechanism effectively resolve this issue.

DABA algorithm for intra cluster communication in IoT

01 Initialize total generation of base level connected nodes

02 Select an initial bat for iteration, i.e. cluster head

03 Define pulse frequency $f_i = 3$ and wavelength $\lambda = 5$ at x_i
for each cluster head (bat) so, generated by PSO do

04 Calculate mean function of the wave's solution fitness values determined by the TTL of the Hello Packets by employing equations (6.8), (6.9) and (6.10)

05 Calculate wave's average fitness with the mean values of that instance.

06 Select the maximum of the average values.

07 If the maximum average is less than the current cluster head's fitness value

```
    then
        Find out the fitness values of all visited solutions
        Find the maximum fitness value from the newly calculated ones.
        If the new maximum is less than the current bat, then
            If the present node is at the best solution, change the cluster head (recall
            the PSO) endif
            If the present node fitness is not equal the current best fitness, traverse to
            node with the present value endif
            Revert to the initial values with minimizing f and maximizing λ after
            traversal end if
            if the present maximum fitness is more than the fitness of the present
            node, traverse to the node with the maximum average fitness,
            and revert the values to default after traversal endif
        end if
  if the maximum average is more than the current cluster head fitness value
  then
        Revert the value with maximizing f and minimizing λ
        Set new maximum fitness value with this max value
    end if
    Update the Best Solution
end for
```

6.4.2.2 Hybrid metaheuristic approaches in intra IoT Clusters

The velocity updating technique, which is intended to replicate the dynamic motions of IoT-equipped devices, ensures its use in inter-cluster connections and generates an optimum path for data packets. However, it has limitations due to the randomization nature of the process. Despite these advantages, PSO suffers from premature convergence. This is due to its probabilistic structure, as well as the inability to balance cognitive and social searches due to the non-adjusted control parameter inertia weight. During updating, the position vector might not reach the target cluster due to the variable nature and interconnected distances between IoT clusters, and so stays un-optimized. During these multi-node optimizations, a more regulated approach is necessary. Because PSO does not rely on the starting population for convergence, but can occasionally experience premature convergence, a novel technique based on PSO with a constriction factor is developed. For multi-cluster optimizations, it is hybridized with the DABA. The PSO's inertia weight factor is replaced by the constriction factor, ensuring a guided and controlled balance between the global and local search processes. This results in effective multi-cluster optimizations in the IoT paradigm.

C-PSO algorithm for inter cluster communication in IoT.

```
Initialize particle (cluster node) velocity and position distributed with normal
distribution
Evaluate fitness of each particle
Define $\varphi_1 = 2$ and $\varphi_2 = 1.4$
procedure PSO (search space, terminal nodes, swarm size, max-iterations)
    Generate a population of particles in the search space with initial fitness
    value while maximum no. of iterations is not reached do
        Evaluate new acceleration coefficients
        Evaluate constriction factor using equation (6.5)
        for each particle in the cluster do
            Update particle velocity using equation (6.6)
            Update particle position using equation (6.4)
            Calculate the fitness of the particle and use this local best to
            generate a cluster node
        end for
        Find the best fitness in the swarm and use this global best to select the
        cluster head
    end while
    Calculate the fitness for each particle and find the best fitness in the cluster
end procedure
```

6.4.2.3. Load Balancing in Fog Servers

When a zone's network traffic density due to a single Fog server rises during authentic operations, a number of attributes reveal the system's competence. However, because there is no Fog server facility for creating a viable uplink or there is a waiting period in the allocation of resources in a Fog server, most nodes fail to achieve a genuine uplink to these cloud servers. This causes the nodes to come to a standstill for an uncertain amount of time in order to gain access to cloud servers. The Dynamic Graph Partitioning approach is presented to control the pack to perform such obstructions. This is done to counteract this resource allocation issue.

The best repartitioning technique for dynamic graphs in Fog servers considers that the partitions must be updated because variations are unavoidable. Partitioning must be changed quickly in response to graph changes in order to avoid performance degradations. Also, high ascend ability must be guaranteed for these partition optimization decisions.

The Resource Allocation Table was the primary focus of the Dynamic Graph Partitioning Algorithm. This table was present and operated in each Fog Server. The graph, an enumerated list of all node connections in the range of a specific swarm, can be used effectively, thanks to repartitioning. The position of every node was changed and updated from a flat file that contained each node's positional changes to ensure a good allocation and wait cycle.

$G(t) = (V(t), E(t))$ is a dynamic graph that is constructed to outline a graph with edges E and vertices V and transits with respect to time t. This graph is obtained after the reduction of the elements. Now suppose, at time t, $R(t)$ is a collection of partitions that is on V and $R_i(t)$. Here, i is the individual partition and $|R_i| = m$. The divisions are done accordingly such that $m, i, t, R_i(t) = V$ and $R_i(t) \cap R_j(t) = \emptyset$; this is for any $i = j$. From the edge cut set the restricting point vertices, that is, $Ec \subseteq E$, which is a set of edges, is a part of the distinctive divisions. A distributed graph processing system divides the distribution and allocation between compute nodes. At time = 0, each vertex is allocated to a section, and here the graph is supplied with an initial section. The additional vertices in the graph must also be allocated to a section within the IoT-particularized connection set when $t > 0$. In large-scale IoT-based dynamic graph processing systems, hash partitioning is a widely used scenario. If $H(v) \, mod \, k = i$, then a vertex is allocated to a segment $P_i(0)$, where $H(v)$ is the hashing function. The Dynamic Graph Partitioning algorithm optimizes the Fog Servers by balancing the load in concurrence with an IoT infrastructure that is optimized.

6.4.3. Simulation Setup

To offer a good monitoring environment and a comparatively straightforward algorithm deployment, the INET framework mode and the relevant topology setups run the simulations. Several things such as the allotment timings of the network, the propensity of the IoT devices that are base level, node cluster managements, packet transmissions, and monitoring— all are utilized as an SDN simulation with the help of OMNET++ 5.0 network simulator [6.70], [6.71]. This is done for the deployment and appropriate usage of the algorithms. This procedure was performed using a base test case with 600 network-capable nodes. And, the topographic size of 1600m × 1600m to 2200m × 2200m.

With the simulation's known nature, a certain degree of stress is produced on the complete subsystem of nodes. Considering the simulation's Random Walk mobility model and the stress conditions that are imposed, a close-to-realistic scenario has been produced, while portraying a sparse distribution. Table 6.3 below lists the attributes of simulation for the experiments.

The Fog servers are implemented in Python 2.7.7 [6.72]. It also uses the node handovers that occurs during server load balancing, A concurrency is tough to sustain due to the timing discrepancies. And because it cannot be fed through to the python simulations, a record is kept. This record is then used as a sample file by the Fog Server simulation.

Table 6.3. Simulation Attributes for the experiment

Parameter	Values
Topology Size	1600 m × 1600 m to 2200mm x 2200mm
Number of Nodes	600
Data Packet Size	4 bytes
Control Packet Size	100 bits
Initial Energy Per Node	2.1 J
Mobility Model	Random Walk
Maximum Channel Power	2mW
Radio Bitrate	1000kbps
Simulation Time	3400s
Simulation Style	Cmdenv-fast-mode

6.4.4. Results and discussions

By having consistent node connections, the simulation was accomplished for the purpose of evaluation. Traversing of Nodes with DABA are also simulated, as well as two other presented hybrid algorithms: nodes with DABA, C-PSO hybrid and nodes with DABA, PSO hybrid. To achieve a near palpable milieu, each condition is repeated 30 times for a total of 3400 seconds. Table 6.4 contains the results of all algorithms. It shows that the presented algorithm's nodes with both DABA C-PSO hybrid and DABA-PSO hybrid connections outperform the traditional regular connection of nodes and also nodes with DABA in contrast with the packet loss percentage. Both the algorithms that are presented produce minimal loss percentages when the number of nodes in the network is increased. It is also apparent that when the network topology grows more complex, the efficiency of all algorithms reduces, despite the fact that the suggested algorithm DABA C-PSO hybrid yields the smallest loss percentage value when compared to the other algorithms.

Table 6.4. Loss Percentage of packets in IoT environment

Nodes	Topology	Loss Percentage (%)			
		Regular node connections	Nodes with DABA	Nodes with DABA and PSO hybrid	Nodes with DABA and C-PSO hybrid
40	1200 × 1200	88.32	54.24	41.2	39.1
60	1800 × 1800	96.31	77.46	64	61.2
80	2000 × 2000	98.74	86.74	71.8	67.5
100	2200 × 2200	98.8	87.44	72.1	67.9

Table 6.5 shows that when DABA and C-PSO hybrid are used, they outperform their equivalents in terms of packet delivery ratio, resulting in a more dependable IoT environment.

When using the aided and non-aided Fog load balancing, characterizing the total number of nodes, a correct Fog Server allocation out on a predetermined number of existing nodes is devised. Figure 6.5 depicts the outcomes of both modes, which are recorded and shown in a graph.

Table 6.5: Comparison over packet delivery ratio in IoT environment

Nodes	Topology	Packet Delivery Ratio			
		Regular nodes connections	Nodes with DABA	Nodes with DABA and PSO hybrid	Nodes with DABA and C-PSO hybrid
40	1200 × 1200	0.1168	0.4576	0.588	0.605
60	1800 × 1800	0.03689	0.2254	0.36	0.391
80	2000 x 2000	0.0126	0.1326	0.282	0.318
100	2200 × 2200	0.098	0.1198	0.271	0.304

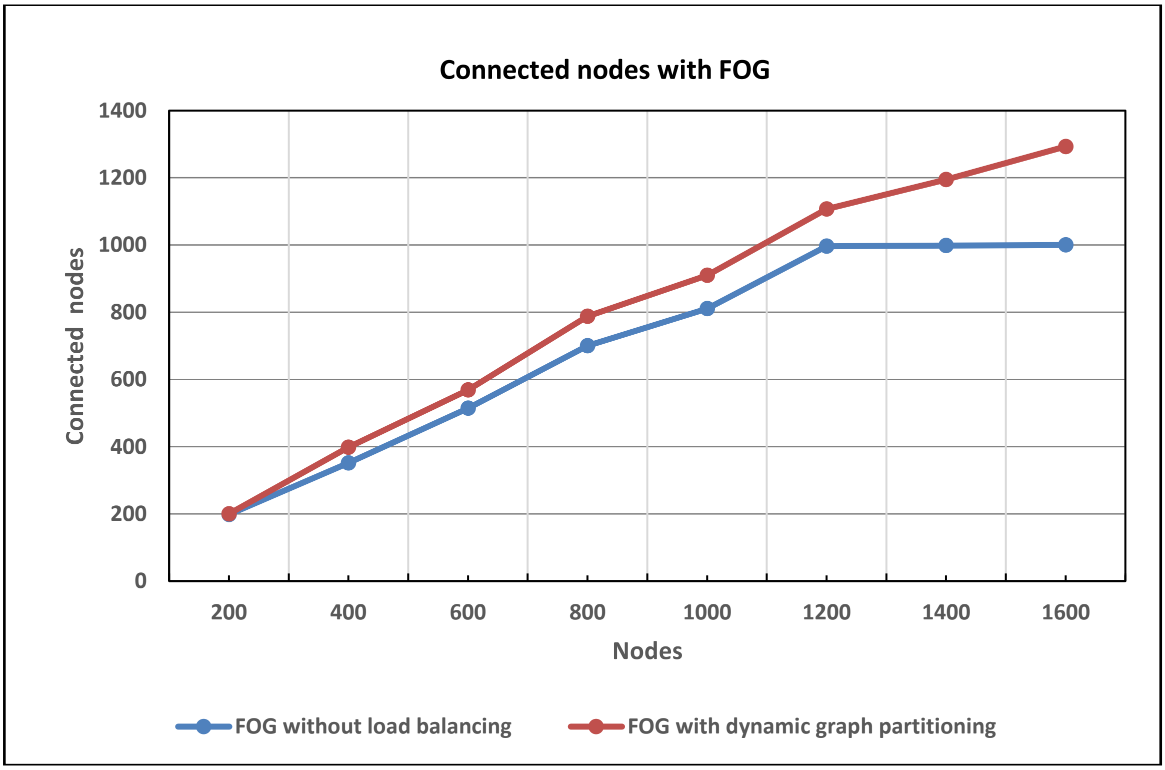

Figure 6.5. Comparison of fog load balancing.

With the placement of the Dynamic Graph Partitioning Algorithm in case of their Load Balancing, the Fog allocations and the node handovers are far enhanced, as can be seen in the graph in the image. Fog allocations are ominously unusual to expedite appropriate data flow in the IoT environment, even in the situation of node constraints or bottlenecks.

References

6.1 S. K. Sarkar, *Wireless Sensor and Ad-Hoc Networks under Diversified Network Scenarios*. Artech House, USA: Artech, 2012.

6.2 T. G. Basavaraju, K. B. Sarckha, S. Gowrishankar, and S. K. Sarkar, "Comparative analysis of routing protocols in wireless Sensor networks," *in Proceedings of National Conference on Ubiquitous Computing (NCUPC)*, Nitte, India, 2008, pp. 16-22.

6.3 G. Degirmenci, J. P. Kharoufeh, and O. A. Prokopyev, "On Optimal Clustering in Mobile Wireless Sensor Networks Under Uncertainty," *Military Operations Research*, vol. 20, no. 2, pp. 19–33, January 2015.

6.4 *B. Mamalis, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "Clustering in wireless sensor networks," in Proceedings of National Conference RFID and sensor networks: architectures, protocols, security and integrations,* 2009, pp. 324-353.

6.5 S. K. Sarkar, T. G. Basavaraju, and C. Puttamadappa, *Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Application*. Boston, MA, USA: CRC Press, Auerbach Publications, 2013.

6.6 S. Gowrishankar, T. G. Basavaraju, and S. K. Sarkar: "Scenario Based Simulation study of Ad Hoc Routing Protocols Behaviour in wireless Sensor Networks," *in Proceedings of* 3rd *International conference on Intelligent System and Networks*, Klawad, India, 2009, pp 527-532.

6.7 H. Meyerhenke, P. Sanders, and C. Schulz, "Partitioning Complex Networks via Size-constrained Clustering," *in Proceedings of the 13th Int. Symp. on Experimental Algorithms*, 2014, pp. 351–363.

6.8 Y. Akhremtsev, P. Sanders, and C. Schulz, "Semi-External Algorithms for Graph Partitioning and Clustering," *in Proceedings of 15th Workshop on Algorithm Engineering and Experimentation (ALENEX)*, Karlsruhe, Germany, 2015, pp. 33–43.

6.9 P. Calhoun, M. Montemurro, and D. Stanley, "*Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11*," RFC 5416, Standard Track, March 2009.

6.10 Dr. T. Karthikeyan, Mr. J. MohanaSundaram, "A Study on Ant Colony Optimization with Association Rule," *IJARCSSE*, vol. 2, no. 5 ,2005.

6.11 H, Fu, "A Novel Clustering Algorithm with Ant Colony Optimization," *in Proceedings of IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Wuhan, China, 2008, pp. 66-69.

6.12 D. P. Rini, S. M. Shamsuddin, and S. S. Yuhaniz, "Particle Swarm Optimization: Technique, System and Challenges", *International Journal of Computer Applications,* vol. 14, no. 1, Sept. 2011.

6.13 J. R. Parvin, and C. Vasanthanayaki, "Particle Swarm Optimization-Based Clustering by Preventing Residual Nodes in Wireless Sensor Networks," *IEEE Sensors*, vol. 15, no. 8, pp. 4264 – 4274, March 2015.

6.14 R. K. Yadav, V. Kumar, and R. Kumar, "A Discrete Particle Swarm Optimization Based Clustering Algorithm for Wireless Sensor Networks," *in Proceedings of 49th Annual Convention of the Computer Society of India*, Hyderabad, India, 2014, pp 137–144.

6.15 Y. Marinakis, M. Marinaki, and N. Matsatsinis, "A Hybrid Clustering Algorithm Based on Multi-swarm Constriction PSO and GRASP," *in Proceedings of 10th International Conference,* Turin, Italy, 2008, pp. 186–195.

6.16 C. Vimalarani, R. Subramanian, and S. N. Srivanandam, "An Enhanced PSO-Based Clustering Energy Optimization Algorithm for Wireless Sensor Network," *The Scientefic World Journal*, vol. 2016, no. 1, pp. 1-11, Jan 2016.

6.17 T. S. Nishanth, Rajesh, A. Bharadwaj, and N. Chakravarti, "Implementation and Comparison of LEACH and Non-LEACH Protocols in Wireless Sensor Networks," *in Proceedings of National Conference*, Bangalore, India. 2009.

6.18 H. G. Sabbagh, "Integration of Wireless Sensor Network (WSN) and Internet of Things (IOT), Investigation of Its Security Challenges and Risks," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 6, pp. 37-40, 2016.

6.19 B. Biswas, S. Bhowmick, M. Biswas, S. Roy, A. Dey, and S. K. Sarkar, "IoT-Based Smart Heart-Health Monitoring System," *in Proceedings of IEEE-International Conference on New Trends in Engineering & Technology (ICNTET),* Chennai, India, 2018.

6.20 Bi. Biswas, A. Dey, B. Goswami, K. Naskar, S. Roy, M. J. Hazarika, S. Bhowmick, A. Janaand, and S. K. Sarkar, "IoT-Based Smart Heart-Health

Monitoring System," *in Proceedings of International Seminar Cum Research Colloquium on MEMS based Sensors and Smart Nanostructured Devices (MSSND-2019)*, Kolkata, India, 2019.

6.21 M. Armbrust, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50-58, April 2010.

6.22 L. M. Vaquero, and F. Cuadrado, "Adaptive Partitioning of Large-Scale Dynamic Graphs," *in Proceedings of the 4th annual Symposium on Cloud Computing*, Madrid, Spain, 2014, pp. 144-153.

6.23 M. Yannuzzi, R. Milito, R. S. Gracia, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog Computing, Cloud Computing, and more Fog Computing," *in Proceedings of IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Athens, Greece, 2014, pp 325-329.

6.24 Y. Akhremtsev, P. Sanders, and C. Schulz, "Semi External Algorithms for Graph Partitioning and Clustering," *in Proceedings of 15th Workshop on Algorithm Engineering and Experimentation (ALENEX)*, Karlsruhe, Germany, 2015, pp. 33–43.

6.25 Y. C. Chang, Z. S. Lin, and J. L. Chen, "Cluster based self-organization management protocols for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 1, pp. 75-80, Feb. 2006.

6.26 C. T. Cheng, C. K. Tse, and F. C. M. Lau, "A Clustering Algorithm for Wireless Sensor Networks Based on Social Insect Colonies," *IEEE Sensors*, vol. 11, no. 3, pp. 711–721, September 2011.

6.27 Y. Liao, H. Qi, and W. Li, "Load-Balanced Clustering Algorithm With Distributed Self-Organization for Wireless Sensor Networks," *IEEE Sensors*, vol. 13, no. 5, pp. 1498-1506, May 2013.

6.28 Y. Hua, and L. Zhimei, "A clustering optimization algorithm based on WCA in MANET," *in Proceedings of IEEE 2015 6th International Conference on*

Computing, Communication and Networking Technologies (ICCCNT), Dallas-Fortworth, TX, USA, 2015, pp. 1–5.

6.29 S. Yadav, and V. Kumar, "Optimal Clustering in Underwater Wireless Sensor Networks: Acoustic, EM and FSO Communication Compliant Technique" *IEEE Access*, vol. 5, pp. 12761-12776, 2017.

6.30 W. Dargie, and J. Wen, "A Simple Clustering Strategy for Wireless Sensor Networks," *IEEE Sensors Letters*, vol. 4, no. 6, pp. 1-4, June 2020.

6.31 R. V. Kulkarni, and G. K. Venayagamoorthy, "Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey," *IEEE Transactions on Systems, Man, and Cybernetic*s, *Part C (Applications and Reviews)*, vol. 41, no. 2, pp. 262-267, March 2011.

6.32 H. Ali, U. U. Tariq, M. Hussain, L. Lu, J. Panneerselvam and X. Zhai, "ARSH-FATI: A Novel Metaheuristic for Cluster Head Selection in Wireless Sensor Networks," in *IEEE Systems Journal*, vol. 15, no. 2, pp. 2386-2397, June 2021.

6.33 B. M. Sahoo, H. M. Pandey and T. Amgoth, "A Whale Optimization (WOA): Meta-Heuristic based energy improvement Clustering in Wireless Sensor Networks," *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2021, pp. 649-654.

6.34 C. Chen, L. -C. Wang and C. -M. Yu, "D2CRP: A Novel Distributed 2-Hop Cluster Routing Protocol for Wireless Sensor Networks," in *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19575-19588, October 2022.

6.35 S. Stefano, R. Vittorio, and S. Umberto, "Wireless Cloud Networks for the Factory of Things: Connectivity Modeling and Layout Design," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 180–195, March 2014.

6.36 W. Yichuan, Z. Yefei, H. Xinhong, J. Wenjiang, and M. Weigang, "Game Strategies for Distributed Denial of Service Defense in the Cloud of Things," *Communications and Information Networks,* vol. 1, no. 4, pp. 143 – 155, December 2016.

6.37 S. I. Farah, L. S. U. Jin, A. Asad, and B. A. Kashif, “Optimizing Lifespan and Energy Consumption by Smart Meters in Green-Cloud-Based Smart Grids,” *IEEE Access*, vol. 5, pp. 20934–20945, September, 2017.

6.38 S. Anam, A. Haider, and S. Kashif, “Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges,” *IEEE Access*, vol. 4, pp. 1375 – 1384, March, 2016.

6.39 J. Pawlick, and Q. Zhu, “Strategic Trust in Cloud-Enabled Cyber-Physical Systems with an Application to Glucose Control,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2906 – 2919, July 2017.

6.40 A. S. Nandan, S. Singh, R. Kumar and N. Kumar, "An Optimized Genetic Algorithm for Cluster Head Election Based on Movable Sinks and Adjustable Sensing Ranges in IoT-Based HWSNs," in *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5027-5039, 1 April1, 2022

6.41 I. Ali, I. Ahmedy, A. Gani, M. U. Munir and M. H. Anisi, "Data Collection in Studies on Internet of Things (IoT), Wireless Sensor Networks (WSNs), and Sensor Cloud (SC): Similarities and Differences," in *IEEE Access*, vol. 10, pp. 33909-33931,2022

6.42 M. Chiang, B. Balasubramanian, and F. Bonomi, “The Role of Fog Computing in the Future of the Automobile,” *in Fog for 5G and IoT*, Wiley, 2016, pp 189 - 210.

6.43 T. Bo, C. Zhen, H. Gerald, P. Shuyi, T. Wei, H. Haibo, and Y. Qing, “Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2140 – 2150, March 2017.

6.44 J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang, "Multitier Fog Computing with Large-Scale IoT Data Analytics for Smart Cities," *IEEE Internet of Things*, vol. 5, no. 2, pp. 677-686, April 2018.

6.45 Y. A. Chen, J. P. Walters, and S. P. Crago, "Load Balancing for Minimizing Deadline Misses and Total Runtime for Connected Car Systems in Fog

Computing," *in Proceedings of IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, Guangzhou, China, 2017, pp. 683-690.

6.46 E. Batista, G. Figueiredo, M. Peixoto, M. Serrano, and C. Prazeres, "Load Balancing in the Fog of Things Platforms Through Software-Defined Networking," *in Proceedings of IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom)*, Halifax, NS, Canada, 2018, pp. 1785-1791.

6.47 R. Beraldi, and H. Alnuweiri, "Exploiting Power-of-Choices for Load Balancing in Fog Computing," *in Proceedings of IEEE International Conference on Fog Computing (ICFC)*, Prague, Czech Republic, 2019, pp. 80-86.

6.48 M. M. Shahriar Maswood, M. R. Rahman, A. G. Alharbi, and D. Medhi, "A Novel Strategy to Achieve Bandwidth Cost Reduction and Load Balancing in a Cooperative Three-Layer Fog-Cloud Computing Environment," *IEEE Access*, vol. 8, pp. 113737-113750, June, 2020.

6.49 A. J. Kadhim, and J. I. Naser, "Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN," *China Communications*, vol. 18, no. 2, pp. 271-289, Feb. 2021.

6.50 P. Anitha Christy Angelin., P. V. Samuel Devakumar., L. H. Al-Farhani, G. V. Kumar, S. Deep and S. Mohammad, "Fog-Cloud-IoT Enabled WSN Architecture," *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, London, United Kingdom, 2022, pp. 588-595.

6.51 M. Dorigo, and G. D. Caro, "Ant colony optimization: a new meta-heuristic," *in Proceedings of the 1999 Congress on Evolutionary Computation*, Washington, DC, USA, 1999, pp. 1470-1477.

6.52 K. M. Sim, and W. H. Sun, "Ant colony optimization for routing and load-balancing: survey and new directions," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 33, no. 5, pp. 560-572, Sept. 2003.

6.53 J. W. Lee, B. S. Choi, and J. J. Lee, "Energy-Efficient Coverage of Wireless Sensor Networks Using Ant Colony Optimization with Three Types of Pheromones," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 419–427, June 2011.

6.54 J. John, and R. Pushpalakshmi, "A reliable optimized clustering in MANET using Ant Colony algorithm," *in Proceedings of International Conference on Communication and Signal Processing*, Melmaruvathur, India, 2014, pp. 051-055.

6.55 S. Afsah, F. Anwar, and S. M. A. Motakabber, "Computer and Communication Engineering (ICCCE) - A Noble Approach of ACO Algorithm for WSN," *in Proceedings of IEEE 2018 7th International Conference on Computer and Communication Engineering (ICCCE),* Kuala Lumpur, Malaysia, 2018, pp. 152–156.

6.56 A. Sharmin, F. Anwar, S. M. A. Motakabber and A. -H. A. Hashim, "Secure ACO-Based Wireless Sensor Network Routing Algorithm for IoT," *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, Kuala Lumpur, Malaysia, 2021, pp. 190-195.

6.57 T. C. Bora, L. d. S. Coelho, and L. Lebensztajn, "Bat-Inspired Optimization Approach for the Brushless DC Wheel Motor Problem," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 947-950, Feb. 2012.

6.58 A. Rekaby, "Directed Artificial Bat Algorithm (DABA) - A new bio-inspired algorithm," *in Proceedings of International Conference on Advances in Computing, Communications and Informatics (ICACCI),* Mysore, India, 2013, pp. 1241-1246.

6.59 H. Afrabandpey, M. Ghaffari, A. Mirzaei, and M. Safayani, "A novel Bat Algorithm based on chaos for optimization tasks," *in Proceedings of Iranian Conference on Intelligent Systems (ICIS)*, Bam, Iran, 2014, pp. 1-6.

6.60 Y. Saji, M. E. Riffi, and B. Ahiod, "Discrete bat-inspired algorithm for traveling salesman problem," *in Proceedings of Second World Conference on Complex Systems (WCCS)*, Agadir, Morocco, 2014, pp. 28-31.

6.61 C. Lin, Y. Li, and D. Deng, "A bat-inspired algorithm for router node placement with weighted clients in wireless mesh networks," *in Proceedings of 9th International Conference on Communications and Networking,* Maoming, China, 2014, pp. 139-143.

6.62 G. Wang, B. Chang, and Z. Zhang, "A multi-swarm bat algorithm for global optimization," *in Proceedings of IEEE Congress on Evolutionary Computation (CEC),* Sendai, Japan, 2015, pp. 480-485.

6.63 S. Parija, and P. K. Sahu, "A metaheuristic bat inspired technique for cellular network optimization," *in Proceedings of 2nd International Conference on Man and Machine Interfacing (MAMI),* Bhubaneswar, India, 2017, pp. 1-6.

6.64 T. D. H. Hussein, M. Frikha, S. Ahmed, and J. Rahebi, "Ambulance Vehicle Routing using BAT Algorithm," *in Proceedings of 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE),* Kuala Lumpur, Malaysia, 2021, pp. 1-5.

6.65 B. Pitchaimanickam, "Bat Inspired Technique for Effective Clustering in Wireless Sensor Networks," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2021, pp. 79-84.

6.66 Q. Wen, X. -Q. Zhao, Y. -P. Cui, Y. -P. Zeng, H. Chang and Y. -J. Fu, "Coverage Enhancement Algorithm for WSNs Based on Vampire Bat and Improved Virtual Force," in *IEEE Sensors Journal*, vol. 22, no. 8, pp. 8245-8256, 15 April15,2022

6.67 V. Selvi, and Dr. R. Umarani, "Comparative Analysis of Ant Colony and Particle Swarm Optimizaton Techniques," *International Journal Computer Application*, vol. 5, no. 4, August 2005.

6.68 H. Wang, and C. Liang, "An Improved Ant Colony Algorithm for Continuous Optimization Based on Levy Flight," *Chemical Engineering Transactions*, vol. 51, August, 2016.

6.69 A. Paul, "Graph Based M2M Optimization in an IoT Environment," *in Proceedings of RACS'13,* Montreal, QC, Canada, 2013, pp. 45-46.

6.70 A. Varga, "The OMNeT++ Discrete Event Simulation System," *in Proceedings of European Simulation Multiconference (ESM2001)*, June, 2001.

6.71 A. Varga, and R. Hornig, "An overview of the OMNeT++ simulation environment," *in Proceedings of the 1st Int'l Conf. on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, Marseille, France, 2008, pp. 1–10.

6.72 "Python 2.7.7 Release." python.org. https://www.python.org /download/releases/2.7.7/ (accessed June 1, 2017).

CHAPTER 7

CONCLUSIONS AND FUTURE SCOPE OF THE WORK

7.1. Conclusions:

Over the course of the last few decades, the world has transitioned from simple electronic circuits and devices to complex intelligent systems capable of making real-time decisions and collaborating with other systems over a network in order to reach a common high-level goal. This constantly connected digital world is increasingly defined by cutting-edge advances in a wide range of fields including circuit design, mathematical optimization, networking and artificial intelligence. VLSI design of electronic circuits is the most crucial and sought-after technology for building efficient and reliable digital systems due to its enormous advantages. Modern VLSI designs demand a wide range of functionalities like small device size, high efficiency, high speed and low power consumption. This imposes a great challenge on VLSI physical design, especially in the arena of VLSI Routing. Routing being a computationally hard problem cannot be solved in polynomial time. Thus, metaheuristic algorithms are alternative way to obtain near satisfactory results. Several Swarm Intelligence techniques ACO, PSO, IWO, Physarum BioNetwork and their variants and hybrids are carried out in this research which have shown great promise in solving the ever-growing problem of routing optimization in VLSI physical design.

With over a decade of research efforts aimed at improving their performance, Wireless Sensor Network have developed from an initial concept to a mature field with numerous supporting protocols. It laid the foundation of key technologies like IoT. WSNs are used to deploy network topologies of varied dimension with the use of conventional protocol. These varied possibilities of its usage are merely because of its platform. Although many issues have been studied in route discovery and clustering phenomena, WSN have design and operational complexity. Meta-Heuristics are constantly being used and are fine tuned to reduce packet loss in various scenarios of WSN. Swarm Intelligence is one of the key factors which mandates the efficiency of a swarm. metaheuristics play an important role in optimizing the behavioural tendencies of a swarm and hence helps us in reducing its complexities. Thus, the dissertation aimed to attain better circuit performance by addressing the routing issues VLSI circuit and WSN.

Chapter 3 first part mainly focuses on variations of the Particle Swarm Optimization method that have been developed to handle the global routing problem in the VLSI environment. Simultaneously, the control of the acceleration constant in PSO for the VLSI routing problem has been verified. Finally, a proportional analysis is performed among the aforementioned algorithms, as well as three PSO variations that have been recognized as good routing algorithms in VLSI design. The results show that PSO-ST performs well in topologically different issue spaces in the VLSI domain, whereas PSO-SAAC performs best in an almost uniform distributed problem space. PSO-SAAC outperforms PSO-W and PSO-ST in terms of virtually uniform terminal node distribution in VLSI layout generating the minimum interconnection global cost value of *'338'* for the random dataset of bivariate distribution of terminal nodes in VLSI layout. As a result, it is acceptable to conclude that PSO-SAAC reduces the cost of RSMT produced by linking the terminal nodes for almost uniform distributions while PSO-ST reduces the cost of RSMT constructed by increasing random bivariate distributions. The performance of PSO-C and PSO-MU has also been observed to be unaffected by the heterogeneous distribution of VLSI global routing issue space. From the same bivariate set g_{best} value of PSO–MU is determined to be *'329,'* indicating that this method takes substantially longer to execute than PSO-W. The PSO-C algorithm has a runtime of 101.51 seconds, whereas the PSO-MU algorithm has a runtime of 85.48 seconds. In the context of VLSI global routing, this means that the PSO-MU algorithm outperforms the traditional PSO-W and PSO-C algorithms while decreasing the Timing budget when compared to the PSO-C algorithm. For the two coordinate sets, the SD value of PSO-C is '*0.71*' and '*2.25*,' respectively. These values are significantly lower than any other SD values of PSO-W and PSO-MU, ensuring robustness while sacrificing system execution time of the algorithm independent of the search space distribution complexity in VLSI designs. The PSO-MU algorithm's performance maintains a balance between optimization and convergence rate. Despite the fact that PSO-MU is stable in a random problem space, PSO-C appears to be the best algorithm in terms of robustness.

The second section elaborates the proposed method that combined Constricted Particle Swarm Optimization with transformation activity algorithms for synchronous buffer inclusion in a creative way and wirelength minimization in VLSI global routing to achieve the circuit's shortest interconnect latency and minimization of delay in VLSI circuits. The suggested technique is compared to the prior approach based on BPSO in order to restrict the general aggregate wirelength of the VLSI circuit in the first step and VLSI interconnect latency in the second step, using CPSO-MU to provide optimal routing arrangements with buffer addition. The suggested CPSO-MU method has been shown to be a capable solution for the benchmark 32 × 32 grid-graph, and it could be useful in the routing of large mechanical circuits with higher levels of complexity. The results of the algorithm's re-enactment show that, like the previous BPSO, the proposed algorithm effectively produces an interconnect delay of '*371.43*' ps for the test graph while achieving global convergence with the fewest number of iterations '*343*,' which is vastly improved when compared to the prior approach. It also shows that the proposed CPSO-MU is comparable to the previous BPSO in terms of producing the best solution for a general wire length of '*49*' pieces with the smallest number of iterations and lower standard deviation for both elements of the test, ensuring more predictable execution.

The prime motive of **chapter 4** section one based on Invasive Weed Optimization is suggested for solving the MRST problem for application in VLSI global routing. The EIWO and EIWO-PSO algorithms are compared to the PSO-W algorithm using MRST Improvement over Minimum Spanning Tree. When compared to the PSO-W method, the suggested algorithm EIWO achieves a greater improvement over the Minimum Spanning Tree. On comparison to the other algorithms, EIWO-PSO hybrid provides the lowest MRST cost regardless of the data set in the test bed. EIWO and EIWO-PSO have higher temporal complexity due to subroutines implementation of the elimination mechanism, whereas PSO-W is substantially faster because it has no overhead due to the elimination of low fitness members. The algorithm is compared to the Geosteiner-5.0.1 benchmark which is a high-space-complexity precise RMST algorithm. The EIWO and EIWO-PSO Hybrid algorithms perform identically to the benchmark, with the lowest minimal wire length cost of '*2198*'. The EIWO-PSO hybrid method has a higher optimization but a slower convergence rate. For large instance data sets, both EIWO and EIWO-PSO algorithms beat PWO-W. For 500 terminal nodes, EIWO-PSO hybrid generates a minimum wire length cost of 18014, compared to '*18405*' for EIWO-PSO hybrid. Further, the hybrid algorithm is compared to a standard VLSI global routing benchmark, ISPD'98, and the wirelength obtained is measured where the findings clearly demonstrate the proposed method's comparable efficiency, which deviates roughly *7%* from the benchmark, and in some situations, the suggested algorithm demonstrates its superiority. The algorithms' superior performance, even with limited resources and large VLSI instances, and comparable performance against standard benchmarks, establishes their robustness, allowing for improved interconnect wire-length optimization of VLSI circuits, allowing devices to act faster and consume less power.

Chapter 5 focuses on new optimization strategies based on biological microorganism behaviour could be very useful in solving computer difficulties. With a larger number of terminal nodes in the search space, the proposed methods outperform PSO-W. The suggested algorithms are more effective in determining the shortest connected rectilinear path between nodes. It's also worth noting that, regardless of the terminal node distribution pattern or the difference between the minimum and mean wire-length value, both the proposed Physarum-based Optimization algorithm and its hybrid produce overall interconnected wirelengths within acceptable ranges, whereas PSO-W causes PSO's consistency to suffer. Due to the randomization nature of PSO-W, the average CPU runtime increases in a non-linear manner as the number of nodes increases. The suggested APO algorithm uses less CPU time because, in Physarum Optimization, the connection where flow is steadily diminishing is deleted at a rapid rate, resulting in the elimination of unneeded steiner points. The algorithms' least optimal wirelength is compared to that of the Geosteiner-5.0.1 benchmark where APO-PSO achieve the shortest minimum wire length of *'455'* for 15 terminal nodes, which is identical to the benchmark. On a large instance random data set of 200 terminal nodes, the proposed techniques are tested. APO-PSO generates a minimum wirelength cost of *'11762'* as opposed to APO's *'11854'*. The suggested algorithms are evaluated using the ISPD'98 global routing benchmark, and the estimated wirelengths are compared. The experimental results show that the suggested method deviates roughly *5%* in various benchmarks, where it outperforms the approach.

Chapter 6 section one dedicated on operating a WSN system on a wide scale in a variety of settings, it is critical to conserve energy while simultaneously maintaining a steady connection among the other nodes. In a WSN, the clustering technique tries to improve node connectivity and increase system efficiency. The addition of a constrained factor to PSO not only allows for faster and denser cluster formation in a high-scale WSN environment, but it also provides smooth convergence and beats its competitors. For both unguided clustering and directed PSO-Constricted Clustering, the number of non-functional nodes is evaluated versus time. Because the PSO-C based clustering is constrained to a certain search domain and has a knowledge base to search across, it produces fewer non-functional nodes. This aids in increasing the cluster's density and, as a result, improving the relay properties for which they were placed in the first place. When compared to unguided routing, ACO-based routing, and ACO-LF-based data routing, the packet delivery ratio is higher. Unguided routing, it goes without saying, fails to perform and is unable to reach the *85 %* dense cluster structures for data routing. The ACO-LF, on the other hand, uses the

Random Walk to develop greater connectivity within topographies, but it does so in a more enhanced method due to its Levy Flight characteristics, which improves the results over the original ACO. The suggested technique demonstrates that it may be used to create and route data for Cluster-based WSN in an efficient and reliable manner. The ACO and ACO-LF both break beyond the 80% barrier, however the ACO loses ground due to its iterative drawback. ACO requires numerous iterations to produce a good outcome, however it loses to the Random Walk inspired ACO- LF in the situation of a decaying power supply. This outperforms the ACO in terms of scouring the entire search space (the topography) for its goal, and hence produces superior results when compared to its basic equivalent. The addition of a constrained factor to PSO not only allows for faster and denser cluster formation in a high-scale WSN environment, but it also provides smooth convergence and beats its competitors.

Section two of Chapter 6 focuses on the connection of thousands of appliances to the internet, combined with the recent penetration of WSN technology, provides a problem in terms of establishing a solid network and promises the creation of an unrestricted IoT infrastructure platform. A clever combination of fog and cloud computing has resulted in a flexible and ascendable IoT platform. The incorporation of both metaheuristics PSO and DABA optimization algorithm as a hybridization process has greatly aided in cumulating node connections and packet routing in congested and mobbed settings, as demonstrated by simulated results. The results of algorithms (Nodes with DABA, Nodes with DABA and PSO hybrid, Nodes with DABA and C-PSO hybrid) show that in terms of packet loss percent-age, both the suggested methods nodes with DABA-PSO hybrid and DABA C-PSO hybrid outperform traditional regular node connections as well as nodes with DABA. Both proposed algorithms produce modest loss percentages when the number of nodes in the network is increased. It is also apparent that when the network topology grows more complex, the efficiency of all algorithms diminishes, despite the fact that the suggested algorithm DABA C-PSO hybrid yields the lowest loss percentage value when compared to the other algorithms. When compared to traditional methods, the dynamic graph partitioning mechanism incorporated in Fog computing has effectively aided in maintaining a flatter handover and resource usage. When they are arranged functionally to properly load balance the intermediate fog servers, the proposed IoT infrastructure, which is super meant by the hybrid algorithm and dynamic graph partitioning, can graft excessively and is competent to handle bottlenecks and network traffic superiorly. Furthermore, the addition of C-PSO, DABA hybrid further improves the overall structure's efficiency by preventing premature convergence of the PSO algorithm in route optimization. With the addition of the dynamic graph partitioning mechanism for load balancing, node handovers and fog allocations are greatly enhanced.

7.2. Future Scope of the Work

The outcomes of the present research dissertation also show that there still exists room for improvement and possible extensions can be aimed as future work.

1. Future work can be done by taking into account a varying distance between the terminal points (indicating a part of the system is more crowded than the other parts which is a quite common scenario in modern VLSI systems) in contrast to the fixed equal distances between terminal points used in this work assuming full availability of the chip area for routing. Hybridized metaheuristic algorithm can also be applied for solving other problems like obstacle avoidance and congestion avoidance along with interconnect wire length reduction.

2. The fog allocations are ominously remarkable to expedite appropriate data flow in the IoT infrastructure, even in the face of node constraints. Significant optimization, combined with the proper application of swarm intelligence at various levels of the IoT infrastructure, can significantly improve the platform's efficacy. To systematize and abridge the IoT, changes might be implemented into the system design. Auxiliary research can be done to develop a more radical and vibrant cloud platform capable of handling and delivering the unique requirements of IoT infrastructure.

3. The goal of finding the most optimal solution possible against the odds of the high computational complexity of perfect optimization of these problems can led

researchers to look at approaches used in fields beyond metaheuristics such as data-intensive algorithms that perform predictive analytics. Machine learning, and data science in general, bring with them the marvel of algorithms that learn from experience, that better their results with each new test case they see – opening new horizons for machine-intelligence and machine-experience based optimization and problem-solving in the future. Neural networks that model the neural pathways of the brain have been successfully used for combinatorial optimization in tandem with adaptive learning, when given adequate hardware resources for large-scale parallel processing of vector operations. New algorithms for machine learning and new architectures of neural networks are currently in development by a large number of research groups across the world and their progress are suitable for any researcher in optimization of algorithms in VLSI circuits and WSNs to look forward to.

Subhrapratim Nath
13.03.23