N^0 d'ordre....../FT/UMBB/2023

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University M'Hamed Bougara of Boumerdes

Faculty of Technology

PhD DISSERTATION

PRESENTED BY:

Sylia MEKHMOUKH

In Partial Fulfilment Of The Requirements For The Degree Of

DOCTORATE

Field: TELECOMMUNICATIONS
Option: Networks and Telecommunications

Optimization of Wireless Mesh Networks Planning

JURY MEMBERS:

Pr. M'hamed HAMADOUCHE	Professor	UMBB	President
Dr. Yassine MERAIHI	MCA	UMBB	Supervisor
Pr. Dalila ACHELI	Professor	UMBB	Co-Supervisor
Dr. Amir BENZAOUI	MCA	University of Skikda	Examiner
Dr. Karim BAICHE	MCA	UMBB	Examiner
Dr. Hamza AKROUM	MCA	UMBB	Examiner

Academic Year: 2022/2023

ACKNOWLEDGEMENTS

In the Name of Allah, the Most Gracious, the Most Merciful. First and foremost, I would like to thank Allah for guiding me through all the difficulties and challenges. Thank you for the incomparable gift of salvation and the eternal hope you offer all who would place their trust in you.

I would like to extend my sincere gratitude to my supervisor, Yassine MERAIHI, for making this work possible. His guidance and advice carried me through every stage of writing this thesis. I would also like to thank Professor Dalila ACHELI for co-directing this work.

I am deeply grateful to Amar RAMDANE-CHERIF, Professor at the University of Versailles Saint Quentin in France, for his invaluable help and advice despite his many commitments. I would also like to thank Asma BENMASSAOUD GABIS for her encouragement and support with completing the necessary paperwork.

I would also like to thank the members of the thesis jury, particularly Professor M'hamed HAMADOUCHE who served as the President, and doctors Amir BENZAOUI, Karim BAICHE, and Hamza AKROUM, for evaluating my thesis and providing valuable comments and suggestions.

I wish to express my sincere gratitude to my colleagues at the university of Boumerdes, Selma YAHIA and Amilia AIT SAADI for their great help to do all paper works and and their emotional support throughout my journey.

I would like to give special thanks to my family, my parents, my husband, and my children, Wassim and Melissa, for their continuous support and understanding throughout my research and writing.

Finally, I would like to dedicate this thesis to my beloved mother, who passed away during my studies. This work is a tribute to her memory and a testament to her enduring influence in my life. I miss her deeply and hope that she would be proud of my accomplishments.

ملخص

يتمثل التحدي الرئيسي في الشبكات اللاسلكية المعشقة (WMNs) في مشكلة النشر التي لها تأثير كبير على أداء هذه الشبكات (التغطية، الاتصال، الحمل التوازن والإنتاجية) والتكلفة والقدرة على تلبية متطلبات جودة الخدمة. في سياق وضع أجهزة التوجيه المعشقة، يتأثر أداء الشبكة بعدد ومواقع أجهزة التوجيه المعشّقة، ونطاق الإرسال لكل جهاز توجيه شبكي، وعدد العملاء الذين تمت تغطيتهم من طرف كل جهاز توجيه شبكي وحجم منطقة النشر وعدد العملاء الشبكيين وأوزانهم. مشكلة وضع أجهزة التوجيه المعشقة هي مشكلة NP-hard، تم حلها بنجاح باستخدام طرق meta-heuristic المختلفة مع وقت تنفيذ معقول. في هذا العمل، نقترح ثلاث طرق لحل مشكلة تنصيب الموجهات الشبكية. الطريقة الأولى هي نسخة محسنة من تحسين لهب العثة (MFO)، تسمى ECLO-MFO، بناءً على تكامل ثلاث استراتيجيات بما في ذلك استراتيجية توزيع ضربية الطيران (LFD) والخريطة الفوضوية وتقنية التعلم القائم على المعارضة (OBL) لتحسين الأداء الأمثل لـ MFO. الطريقة الثانية هي طريقة هجينة، تسمى ASO-SA، تعتمد على مزيج من قدرة البحث الكلية لـمُحسِّن الثعبان المتكيف (ASO) مع إمكانية البحث المحلي للتليين المحاكي (SA). يعتمد ASO على دمج آلية OBL المعممة (GOBL) في مرحلة استكشاف SO. أخيرًا، تم اقتراح خوارزمية تحسين الحوت الثنائي (BWOA) لحل مشكلة تخطيط الهيكل في شبكات WMNs. تم ادخال وتحليل ثمانية وظائف نقل مقسمة إلى عائلتين: شكل S وV للحصول على نسخة ثنائية من WOA.

الكلمات الرئيسية: الشبكات اللاسلكية المعشقة، مشكلة وضع الموجهات المعشقة، الاستدلال الفوقي، تحسين لهب العثة MFO، محسن الثعبان SO، الحوت الثنائيBWOA.

ABSTRACT

The main challenge in Wireless Mesh Networks (WMNs) is the deployment issue that has a significant impact on the performance of such networks (coverage, connectivity, load balancing, and throughput), cost, and the capacity to satisfy the Quality of service requirements. In the context of mesh routers placement, the performance of the network is influenced by the number and locations of mesh router, the transmission range of each mesh router, the number of covered clients per mesh router, the size of the deployment area, and the number and weights of mesh clients. The mesh routers placement problem is an NP-hard optimization problem, successfully solved using meta-heuristic optimization approaches with reasonable time execution. In this work, we propose three approaches for solving the mesh routers placement problem. The first approach is an improved version of Moth Flame Optimization (MFO), called ECLO-MFO, based on the integration of three strategies including: the Lévy Flight Distribution (LFD) strategy, chaotic map, and the Opposition Based-Learning (OBL) technique to enhance the optimization performance of MFO. The second approach is a hybrid approach, called ASO-SA, based on the combination of global search capability of an Adaptive Snake Optimizer (ASO) with the local search capability of Simulated Annealing (SA). ASO is based on the integration of the Generalized OBL (GOBL) mechanism into the exploration phase of SO. Finally a Binary Whale Optimization Algorithm (BWOA) is suggested to solve the topology planning problem in WMNs. Eight transfer functions divided into two families such as S-shaped and V-shaped are introduced and analyzed to obtain a binary version of WOA.

Keywords: Wireless Mesh Networks WMNs, the mesh routers placement problem, meta-heuristics, Moth Flame Optimization MFO, Snake Optimizer SO, Binary Whale Optimization Algorithm BWOA.

RÉSUMÉ

Le principal défi des réseaux maillés sans fil (WMNs- Wireless Mesh Networks en anglais) est le problème de déploiement qui a un impact significatif sur les performances de tels reseaux (coverture, connectivité, equilibrage de charge, débit), le cout, et la capacité à satisfaire les exigences de qualité de service. Dans le contexte du placement des routeurs maillés, les performance du réseau sont influencées par le nombre et l'emplacement des routeurs maillés, la transmission de chaque routeur maillé, la taille de la zone de déploiement, le nombre et les poids des clients maillés. Le problème de placement des routeurs maillés est un problème d'optimisation NP-difficile, résolu avec succès en utilisant des approches d'optimisation méta-heuristiques avec un temps d'exécution raisonnable. Dans ce travail, nous proposons trois approches pour résoudre le problème de placement des routeurs maillés. La première approche est une version améliorée de optimiseur de la flamme papillon (MFO- Moth Flame Optimization en anglais), appelée ECLO-MFO, basée sur l'intégration de trois stratégies dont: la distribution du prelevement volant (LFD- Lévy Flight Distribution en anglais), chaotic map, et la technique de l'apprentissage basé sur l'opposition (OBL- l'Opposition Based-Learning en anglais) pour améliorer les performances d'optimisation de MFO. La deuxième approche est une approche hybride, appelée ASO-SA, basée sur la combinaison de la capacité de recherche globale de l'optimiseur du serpent adaptive (ASO) avec la capacité de recherche locale du recuit simulé (SA). ASO repose sur l'intégration du mécanisme OBL generalisé (GOBL) dans la phase d'exploration de SO. Enfin, un algorithme d'optimisation de baleine binaire (BWOA) est proposé pour résoudre le problème de planification de la topologie dans les WMN. Huit fonctions de transfert divisées en deux familles telles que S-shaped et V-shaped sont introduites et analysées pour obtenir une version binaire de WOA.

Mots clés: Réseaux radio maillées WMNs, le probléme de placement des routeurs maillées, metaheuristiques, optimisation de la flame papillon MFO, optimiseur de serpent SO, algorithme d'optimisation des baleines binaire BWOA

CONTENTS

LIST OF ACRONYMS **IX**

General Introduction **1**

1 An overview of Wireless Mesh networks **3**

1.1 Introduction . . . 3
1.2 Wireless Mesh Networks . . . 3
1.3 Components of WMNs . . . 4
1.4 WMN classification . . . 5
1.4.1 Classification according to the type of mesh . . . 5
1.4.2 Classification according to the architecture . . . 6
1.5 Wireless Mesh Networks Characteristics . . . 8
1.6 WMNs advantages . . . 9
1.7 Academic Supporting and Industry Standards . . . 10
1.7.1 IEEE 802.11s Mesh WLAN . . . 10
1.7.2 IEEE 802.15.1 Bluetooth . . . 11
1.7.3 IEEE 802.15.4 Zigbee . . . 11
1.7.4 IEEE 802.16j WiMAX . . . 11
1.8 Transmission techniques in WMNs . . . 11
1.8.1 Orthogonal Frequency Division Multiplexing (OFDM) . . . 11
1.8.2 Multiple Input-Multiple Output (MIMO) . . . 12
1.9 Medium access protocols . . . 13
1.9.1 Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) . . . 13
1.9.2 Time Division Multiple Access (TDMA) . . . 14
1.9.3 Hybrid CSMA/TDMA Media Access Control (MAC) protocol . . . 14
1.10 Applications scenarios . . . 14
1.10.1 Broadband home networking . . . 14

1.10.2 Enterprise networking . . . 15
1.10.3 Metropolitan area networking . . . 16
1.10.4 Transportation systems . . . 16
1.10.5 Building automation . . . 16
1.10.6 Health and medical sciences . . . 17
1.10.7 Safety and surveillance systems . . . 17
1.11 Problems and challenges in WMNs . . . 18
1.11.1 Physical layer . . . 18
1.11.2 Medium access layer . . . 19
1.11.3 Transport layer . . . 19
1.11.4 Network layer . . . 19
1.11.5 Application layer . . . 20
1.11.6 Topological and Deployment . . . 20
1.12 Conclusion . . . 21

2 Optimization methods 23
2.1 Introduction . . . 23
2.2 Optimization problem . . . 23
2.3 Notions and concepts relating to optimization . . . 24
2.4 Classification of optimization problems . . . 26
2.4.1 Classification according to the complexity of optimization problems . 26
2.4.2 Classification according to the nature of the search domain . . . 28
2.4.3 Classification according to the objective function type . . . 28
2.4.4 Classification according to the constraints . . . 29
2.4.5 Classification regarding the number of optimums . . . 30
2.5 Optimization methods . . . 30
2.5.1 Exact methods . . . 30
2.5.2 Approached methods . . . 31
2.6 Conclusion . . . 42

3 The mesh routers placement problem in WMNs based on swarm intelligence metha-heuristics 43
3.1 Introduction . . . 43
3.2 Related works . . . 43
3.3 Mesh router nodes placement problem formulation . . . 44
3.3.1 System model . . . 45
3.3.2 Problem Formulation . . . 45
3.4 Enhanced Moth Flame Optimization . . . 47
3.4.1 Moth–Flame Optimization Algorithm . . . 48
3.4.2 Lévy Flight Distribution . . . 51

3.4.3 Chaotic map 51
3.4.4 Opposition-based learning 52
3.4.5 The proposed ECLO-MFO for solving the mesh router nodes placement problem 53
3.5 Simulation results and analysis 56
3.5.1 Effect of varying the λ value 56
3.5.2 Effect of varying the number of MCs 57
3.5.3 Effect of varying the number of MRs 60
3.5.4 Effect of varying coverage radius values 62
3.5.5 Convergence Analysis 65
3.6 Conclusion 67

4 A hybrid approach for the mesh routers placement problem with service priority 68
4.1 Introduction 68
4.2 Related works 68
4.3 The mesh routers placement problem formulation 70
4.4 Preliminaries 71
4.4.1 Snake Optimizer 71
4.4.2 Generalized opposition based-learning 76
4.4.3 Simulated Annealing (SA) 76
4.5 Hybrid Adaptive Snake Optimizer with Simulated Annealing (ASO-SA) for the mesh routers placement problem 77
4.6 Simulation results 78
4.6.1 Effect of varying the number of MRs 79
4.6.2 Effect of varying the number of MCs 81
4.6.3 Effect of varying the coverage radius values 81
4.6.4 Convergence analysis 83
4.7 Conclusion 84

5 Binary Whale Optimization Algorithm for topology planning problem in WMNs 85
5.1 Introduction 85
5.2 Related works 85
5.3 The system model and problem formulation 86
5.4 Whale Optimization Algorithm (WOA) 88
5.5 Binary Whale Optimization Algorithm (BWOA) 89
5.6 Simulation results 91
5.6.1 Effect of varying the number of MCs 93
5.6.2 Effect of varying the coverage radius values 94

5.7 Conclusion 96

LIST OF FIGURES

1.1 WMN architecture 4
1.2 Full mesh 5
1.3 Partial mesh 6
1.4 Infrastructure/Backbone WMNs 7
1.5 Client WMNs 7
1.6 Hybrid architecture 8
1.7 Sub-networks created using OFDM technique 12
1.8 MIMO communication technique 12
1.9 WMNs for broadband home networking 15
1.10 WMNs for Enterprise networking 15
1.11 WMNs for metropolitan area network 16
1.12 WMNs for building automation 17
1.13 Ideal application of WMNs in hospital environment 18
1.14 WMNs for surveillance systems 18

2.1 A visualisation of global and local optimum 25
2.2 Classification of optimization problems 26
2.3 Complexity of problems 27
2.4 Pareto-optimal solutions 29
2.5 Classification of population-based meta-heuristics 36
2.6 Movement strategy of a particle 38

3.1 An example of WMN instance 47
3.2 Plot of the two dimensional variables from Lévy flights distribution started from the origin (marked with a bold point) and plotted 200 points with a=1.4 51
3.3 The chaotic dynamics of Sine map 52
3.4 The effect of combining sine map with lévy flight 53
3.5 The flowchart of: (a) MFO, (b) ECLO-MFO 54

3.6 Effect of varying λ values on Fitness f. 57
3.7 Effect of varying the number of MCs on: (a) Coverage Ψ (b) Connectivity Φ (c) Fitness f. 59
3.8 Effect of varying the number of MRs on: (a) Coverage ψ (b) Connectivity Φ (c) Fitness f. 61
3.9 Effect of varying coverage radius values on: (a) Coverage Ψ (b) Connectivity Φ (c) Fitness f. 63
3.10 Example of algorithms convergence using: (a) $Instance_1$ (b) $Instance_2$ (c) $Instance_3$ (d) $Instance_4$ (e) $Instance_5$ 66

4.1 Placement obtained using: (a) ASO-SA (b) SO-SA (c) ASO (d) SO (e) SA . 80
4.2 Fitness under various number of mesh routers 81
4.3 Fitness under various number of mesh clients 82
4.4 Fitness under various coverage radius values 82
4.5 Example of algorithms convergence using: (a) $Instance_1$ (b) $Instance_2$ (c) $Instance_3$ (d) $Instance_4$ 83

5.1 S-shaped family of transfer functions. 90
5.2 V-shaped family of transfer functions. 91
5.3 Obtained placements using: (a) BWOAS1 (b) BWOAS2 (c) BWOAS3 (d) BWOAS4 (e) BWOAV1 (f) BWOAV2 (g) BWOAV3 (h) BWOAV4 92
5.4 Mean number of mesh routers under various number of mesh clients 93
5.5 Example of algorithms convergence using: (a) 20 MCs (b) 40MCs (c) 60 MCs (d) 80 MCs (e) 100 MCs 95
5.6 Mean number of mesh routers under various coverage radius values 96
5.7 Example of algorithms convergence using various coverage radius values: (a) 300m (b) 350m (c) 400m (d) 450m (e) 500m 97

LIST OF TABLES

3.1 Summary of some existing WMNs nodes placement representative works . . 45
3.2 Representation of the solution of deploying eight mesh routers 47
3.3 Parameters values considered in our simulations 56
3.4 Fitness f for varied λ values 57
3.5 Coverage Ψ, connectivity ϕ, and fitness f for varied number of mesh clients . 58
3.6 Coverage ψ, connectivity Φ, fitness f under various number of mesh routers . 60
3.7 Coverage Ψ, connectivity Φ, and fitness f for varied coverage radius values . 64
3.8 Network instances taken into account in the convergence analysis 65
3.9 Convergence analysis between ECLO-MFO, MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA algorithms 65

4.1 Summary of hybrid approaches employed to solve the mesh routers placement problem in WMNs 69
4.2 Parameters values considered in our simulations 78
4.3 Algorithms parameters 79
4.4 Fitness under various number of mesh routers 81
4.5 Fitness under various number of mesh clients 81
4.6 Fitness under various coverage radius values 82
4.7 Network instances considered in convergence analysis 83
4.8 Convergence analysis between ASO-SA, SO-SA, ASO, SO, and SA algorithms 84

5.1 S-shaped family of transfer functions 90
5.2 V-shaped family of transfer functions 91
5.3 Parameters values considered in our simulations 93
5.4 Maximum, minimum, and mean number of mesh routers under various number of mesh clients 94
5.5 Maximum, minimum, and mean number of mesh routers under various coverage radius values 96

LIST OF ALGORITHMS

1 The pseudo-code of Descent Method 32
2 The pseudo-code of Simulated Annealing algorithm 33
3 The pseudo-code of Tabu Search 33
4 The pseudo-code of Variable Neighborhood Search 34
5 The pseudo-code of population-based meta-heuristics 35
6 The pseudo-code of evolutionary algorithms 35
7 The pseudo-code of Particle Swarm Optimization 38
8 The pseudo-code of Firefly Algorithm (FA) 41
9 The pseudo-code of MFO Algorithm 50
10 The pseudo-code of the proposed ECLO-MFO for the WMN-MRNP problem 55
11 The pseudo-code of SO algorithm 75
12 The pseudo-code of Simulated Annealing algorithm 77
13 The WOA Algorithm 89

LIST OF ACRONYMS

ABC Ant Bee Colony

AIS Artificial Immune Systems

APOA Artificial Plant Optimization Algorithm

AO Aquila Optimizer

AVO African Vultures Optimization

ACO Ant Colony Optimization

AP Access Point

ACK Acknowledgement

AODV Ad hoc On Demand Distance Vector

ASO Adaptive Snake Optimizer

BA Bat Algorithm

BA Biological-based Algorithms

BAC Building Access Control

BBBC Bing-Bang Big-Crunch

BFO Bacterial Foraging Optimization

BWOA Binary Whale Optimization Algorithm

CFO Central Force Optimization

FWA Fire Work Algorithm

CS Cuckoo Search

COA Coyote Optimization Algorithm

CSMA/CA Carrier Sense Multiple Access/ Collision Avoidance

CTS Clear To Send

CRO Chemical Reaction Optimization

CM Constriction Method

CHIO Coronavirus Herd Immunity Optimizer

DIFS Distributed Inter Frame Space

DM Descent Method

DA Dragonfly Algorithm

DF Differential Evolution

DCA Dentritic Cell Algorithm

EM Electromagnetism-like Mechanism

EA Evolutionary algorithms

EP Evolutionary Programming

EFO Electromagnetic Field Optimization

ES Evolutionary Strategy

FA Firefly Algorithm

FPrIM Fixed Protocol Interference Model

FGIA Foot Game Inspired Algorithm

GA Genetic Algorithm

GP Genetic Programming

GOBL Generalized Opposition Based-Learning

GWO Grey Wolf Optimization

GROM Golden Ratio Optimization Method

GSA Gravitational Search Algorithm

HA Human-based Algorithms

HS Harmony Search

HBA Honey Badger Algorithm

HGSO Henry Gas Solubility Optimization

HSS Hyper-Spherical Search

ILS Iterated Local Search

ICA Imperialist Competitive Algorithm

KHA Krill Herd Algorithm

LFD Lévy Flight Distribution

LDIWM Linear Decreasing Inertia Weight

MC Mesh Client

MR Mesh Router

MG Mesh Gateway

MFO Moth Flame Optimization

MAC Medium Access Layer

MANET Mobile Ad hoc Network

MIMO Multiple Input-Multiple Output

MLP Mixed Linear Programming

MFO Moth Flame Optimization

MA Math-based Algorithms

MVO Multi-Verse Optimizer

MCSS Magnetic Charged System Search

MST Minimum Spanning Tree

OBL Opposition Based-Learning

OFDM Orthogonal Frequency Division Multiplexing

OIO Optics Inspired Optimization

PA physical-based Algorithms

PDA Personal Digital Assistant

PA Plant-based Algorithms

PFA Flower pollination Algorithm

PAN Personal Area Network

PrIM Protocol Interference Model

PSO Particle Swarm Optimization

RD Research and Development

RTT Round Trip Time

RTP Real Time Protocol

RCP Rate Control Protocol

RTS Ready To Send

RIWM Random Inertia Weight Method

RDVM Ration of Decrement of Vmax Method

RMO Root Mass Optimization

RMO Radial Movement Optimization

SIA Swarm Intelligence Algorithms

SSA Salp Swarm Algorithm

SO Snake Optimizer

SA Simulated Annealing

SFS Stochastic Fractal Search

SCA Sine Cosine Algorithm

SIO Sonar Inspired Optimization

SGuA Salping Growing up Algorithm

TDMA Time Division Multiple Access

TCP Transmission Control Protocol

TS Tabu Search

TEO Thermal Exchange Optimization

TLBA Teaching Learning-Based Algorithm

VoIP Voice over Internet Protocol

VNS Variable Neighborhood Search

VPSA Vibrating Particles system Algorithm

WSO White Shark Optimization

WMN Wireless Mesh Network

WOA Whale Optimization Algorithm

WLAN Wireless Local Area Network

Wi-Fi Wireless Fidelity

WiMAX Worldwide Interoperability for Microwave Access

WPAN Wireless Personal Area Network

WSMNs Wireless Sensor Mesh Networks

WOA Whale Optimization Algorithm

WEO Water Evaporation Optimization

WMEN Wireless Mesh Enterprise Network

GENERAL INTRODUCTION

The development of economical and effective wireless networks is currently an imperative necessity due to the astounding success of wireless technologies and the expansion of Internet services. To this end, Wireless Mesh Networks (WMNs) are promising networks for providing high-speed internet access for both network providers and their customers. WMN technology has received a lot of attention from the community of researchers and academics due to its easy implementation, dynamic self-organization, self-configuration, and self-adaptive nature. It is currently used in many applications such as broadband home networking, education, healthcare, corporate networks, industrial automation, disaster management, military, and rescue operations. It provides connectivity to various types of networks such as Wireless Fidelity (Wi-Fi), Cellular, Worldwide Interoperability for Microwave Access (WiMAX), and Sensor. In a WMN, a mobile client (MC) can access the Internet through a wireless backbone formed by wireless Mesh Routers (MRs) which are interconnected in a multi-hop fashion while some MRs known as Mesh Gateways (MGs) act as the communication bridges between the wireless backbone and the Internet.

The design of the network architecture is a fundamental issue for a WMN and is critical in determining the network performance and providing Quality of Service requirements. In fact, the bad positioning of mesh nodes (MR and/or MG) causes many interferences and congestion resulting in low throughput, considerable packet loss, and high delays. To cope with these drawbacks, network operators must adopt efficient optimization methods for WMNs nodes placement. WMNs nodes placement is known to be an NP-hard problem. So approximate optimization algorithms (i.e. heuristic, meta-heuristic, and hybrid algorithms) have been presented as successful optimization algorithms to solve them (obtain ideal solutions) within reasonable time. More details about used approaches for the mesh nodes placement problem can be found in [1]

This thesis is based on three major contributions. In the first contribution, we suggested an improved version of Moth Flame Optimization (MFO), named ECLO-MFO, for solving the mesh routers placement problem. ECLO-MFO is based on the integration of three strategies including: the Lévy Flight Distribution (LFD) strategy, chaotic map, and the Opposition Based-Learning (OBL) technique to enhance the optimization performance of MFO.

In the second contribution, we have proposed a hybrid approach called ASO-SA, based on the combination of the global search capability of an Adaptive Snake Optimizer (ASO) with the local search capability of Simulated Annealing (SA). ASO is based on the integration of the Generalized OBL (GOBL) mechanism for improving the exploration phase of SO algorithm.

In the last contribution, we have suggested a Binary Whale Optimization Algorithm (BWOA) for solving the topology planning problem in WMNs.

Our thesis is organized into five chapters:

In the **first chapter**, we address wireless mesh networks, their components, architectures, characteristics, advantages, and applications. In addition, we present some problems and challenges in WMNs.

In the **second chapter**, we present optimization problem, the notions and concepts related to optimization, the classification of optimization problems regarding several criteria: complexity of the problems, the nature of the problems, the number of optima, the type of the objective function and the number of constraints. Again, we present the optimization methods where we describe the basic principle of some algorithms used for solving complex problems.

In the **third chapter**, we describe the improved MFO approach for solving the mesh router nodes placement problem in WMNs, and we evaluate their performance and compare its characteristics with other optimization methods.

In the **fourth chapter**, we present the hybrid approach, called ASO-SA, based on the combination of ASO and SA, and we evaluate the performance of the proposed approach and compare its characteristics with SO and SA algorithms.

In the **fifth chapter**, we suggest a Binary WOA to solve the topology planning problem in WMNs, and we introduce and analyze S-shaped and V-shaped to obtain a binary WOA.

Finally, we summarize our findings and discuss ideas on how to extend this research.

CHAPTER 1
AN OVERVIEW OF WIRELESS MESH NETWORKS

1.1 Introduction

Wireless mesh networks (WMNs) represent a new generation of multi-hop wireless networks, used to connect various wireless devices by building an unwired mesh [2, 3]. They seek to offer mobile and fixed consumers high-speed internet access everywhere, anytime, and are gaining more and more real interest from the Research and Development (RD) community, network operators, and service providers [3, 4]. Self-configuration, self-organization, and the capacity to autonomously create and sustain connectivity between nodes in locations and surroundings devoid of the internet are all traits that WMNs display [3]. These traits offer a number of benefits, including low implementation costs, simple network management, robustness, etc.

This chapter addresses WMNs, their components, architectures, characteristics, and advantages. In addition, the applications scenarios of WMNs are explored. Finally, we give the problems and challenges in WMNs.

1.2 Wireless Mesh Networks

A wireless mesh network is a specific kind of a wireless network. It offers a possible solution to problems that arise regularly in WLAN and cellular networks. Cellular and WLAN have a small range of connectivity, which is their biggest drawback. These systems have a low data transfer rate and are highly expensive. Wireless mesh networks, on the other hand, are less expensive and offer quicker data transfer rates.

In WMN, communications between two nodes can be supported by several intermediate nodes called mesh routers. There are three categories of wireless routers which together form a relatively static infrastructure in the WMN, including Mesh Router (MR), Mesh Gateway (MG), and Mesh Client (MC). MC connects to the internet through MRs, MRs relay traffic

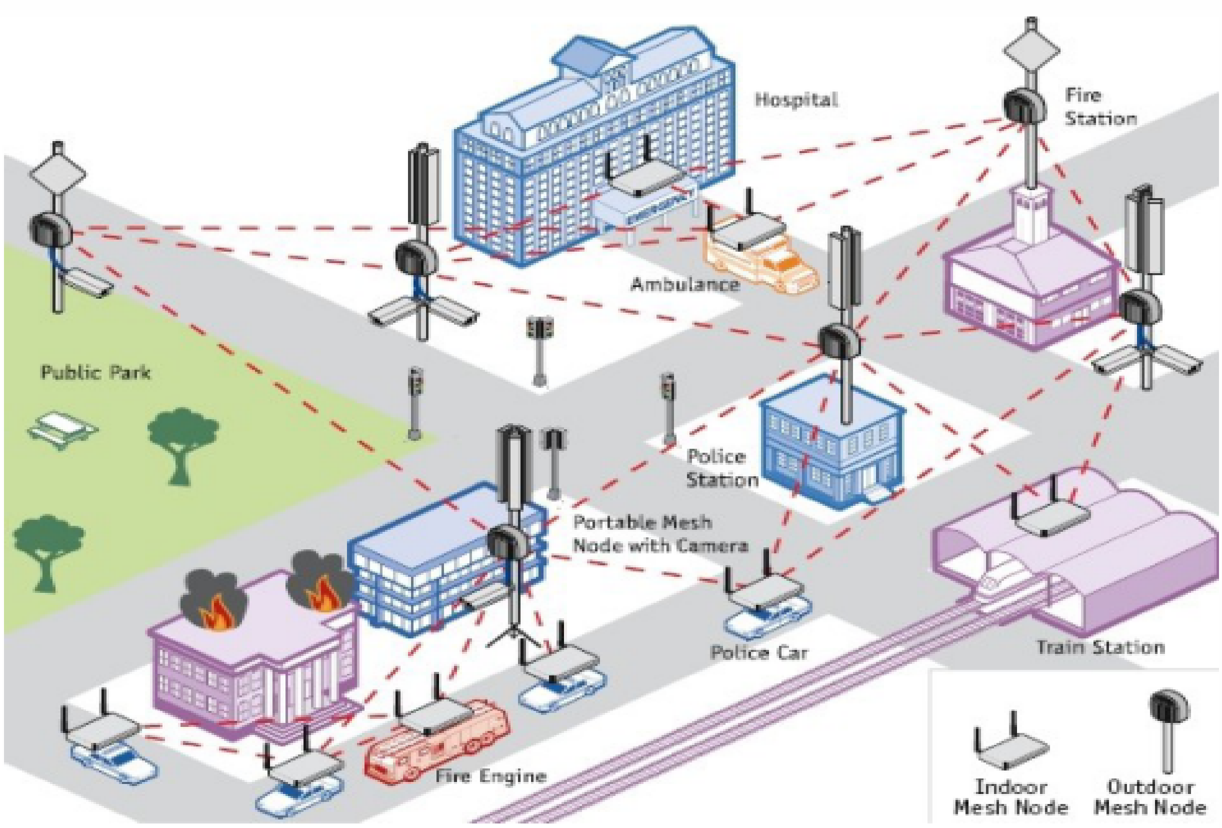


Figure 1.1: WMN architecture

to and from MGs which are connected to internet infrastructure. The typical architecture of such networks is illustrated in Figure 1.1 [5].

1.3 Components of WMNs

As cited earlier, WMNs consists of three types of nodes such as: Mesh Client (MC), Mesh Router (MR), and Mesh Gateway (MG).

Mesh clients are the end-user devices, such as laptops, Personal Digital Assistants (PDAs), smart phones, etc., that can connect to the network and use services like Voice over Internet Protocol (VoIP), games, location-based services, etc. We assume that these devices are portable, have limited power, may perform routing, and may or may not always be connected to the network.

Mesh routers have the capability of routing the traffic in the network. The routers' mobility is restricted, but they offer reliable characteristics. For multi-hop strategy, the power consumption of mesh routers is low. In order to provide scalability in a multi-hop mesh environment, a mesh router's Medium Access Control (MAC) protocol also supports multiple channels and multiple interfaces.

Mesh gateways are mesh routers that have direct access to the Internet via wired infrastructure. The mesh gateways in WMNs are expensive because they require several interfaces to connect to both wired and wireless networks. As a result, the network has a limited number of WMN gateways. Additionally, the locations of mesh gateways significantly affect network performance.

1.4 WMN classification

1.4.1 Classification according to the type of mesh

Based on the type of the mesh, WMNs are classified into two types:

- Full mesh
- Partial mesh

Full mesh

In this type of network, all nodes are within reach of all, where each node in the network is connected with all other nodes, so if the network has N nodes, then each node contains $N-1$ direct connections (one-hop neighbors) as illustrated in Figure 1.2 [6]. Communication between nodes is done directly, which optimizes quality of service, throughput, latency, and bandwidth. In other hand, in addition to the problem of interference, and exposed node, the network coverage area cannot exceed the coverage area of a node.

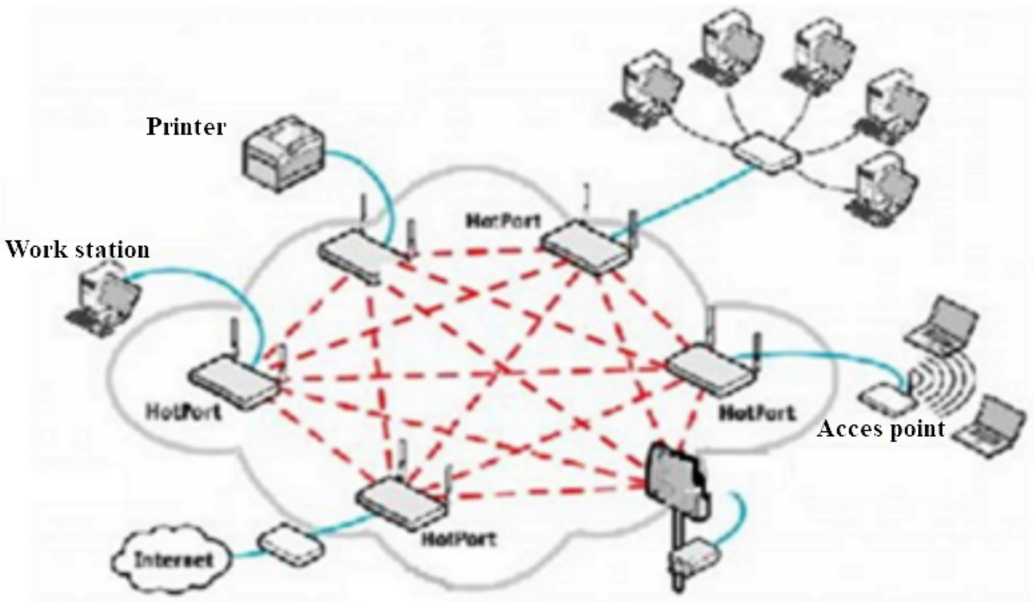


Figure 1.2: Full mesh

Partial mesh

The nodes are connected to each other forming a connected graph as illustrated in Figure 1.3 [6], such that each node can communicate directly with its neighbors and indirectly with the rest of the nodes. In this type o mesh, the coverage area can be extended and the interference problem and exposed node can be reduced.

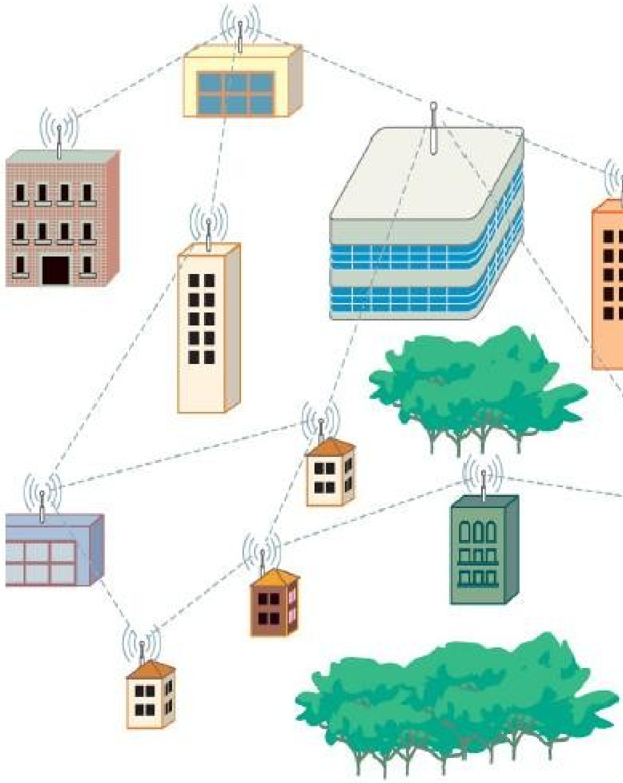


Figure 1.3: Partial mesh

1.4.2 Classification according to the architecture

Based on the function of the nodes in the network, WMNs are classified into three types:

- Infrastructure/Backbone WMNs
- Client WMNs
- Hybrid WMNs

Infrastructure/Backbone WMNs

The backbone network infrastructure is composed of MRs, which also have limited mobility and no energy limitations. MRs can be configured with gateway or bridge capabilities, enabling them to offer an infrastructure for customers to connect them to each other, the Internet, or to other wireless networks. An illustration of this kind of architecture is provided in Figure 1.4

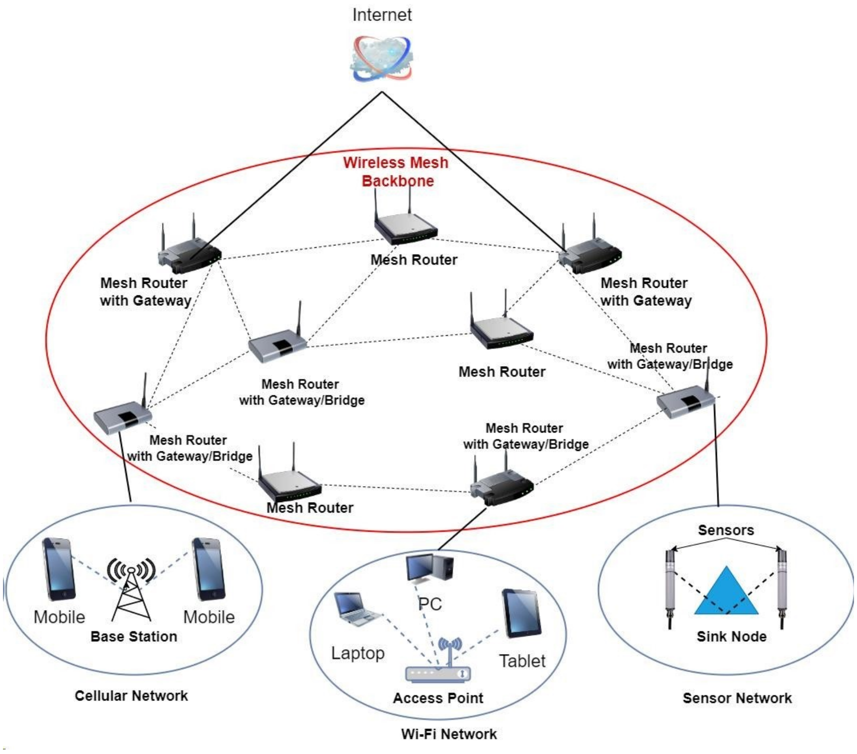


Figure 1.4: Infrastructure/Backbone WMNs

Client WMNs

MCs create a self-organized and self-configurable mesh network with this sort of architecture, for providing point-to-point (peer-to-peer) connections between various users. In these networks, an MR is not necessary; MCs are the only ones. Client WMNs is illustrated in Figure 1.5

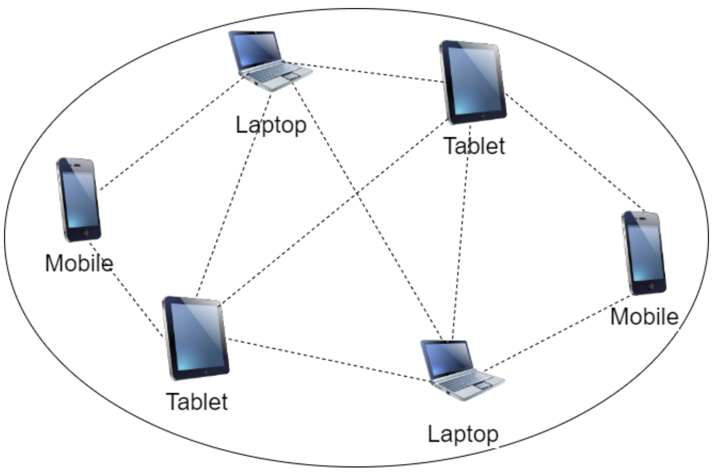


Figure 1.5: Client WMNs

Hybrid WMNs

The hybrid architecture combines the Backbone and Client concepts as illustrated in Figure 1.6. The MCs can communicate directly with other MCs or use the MRs to access the wireless mesh network. The functionality of the routing assigned to the MCs makes it possible to reinforce the connectivity and to ensure a broad network coverage inside the WMN.

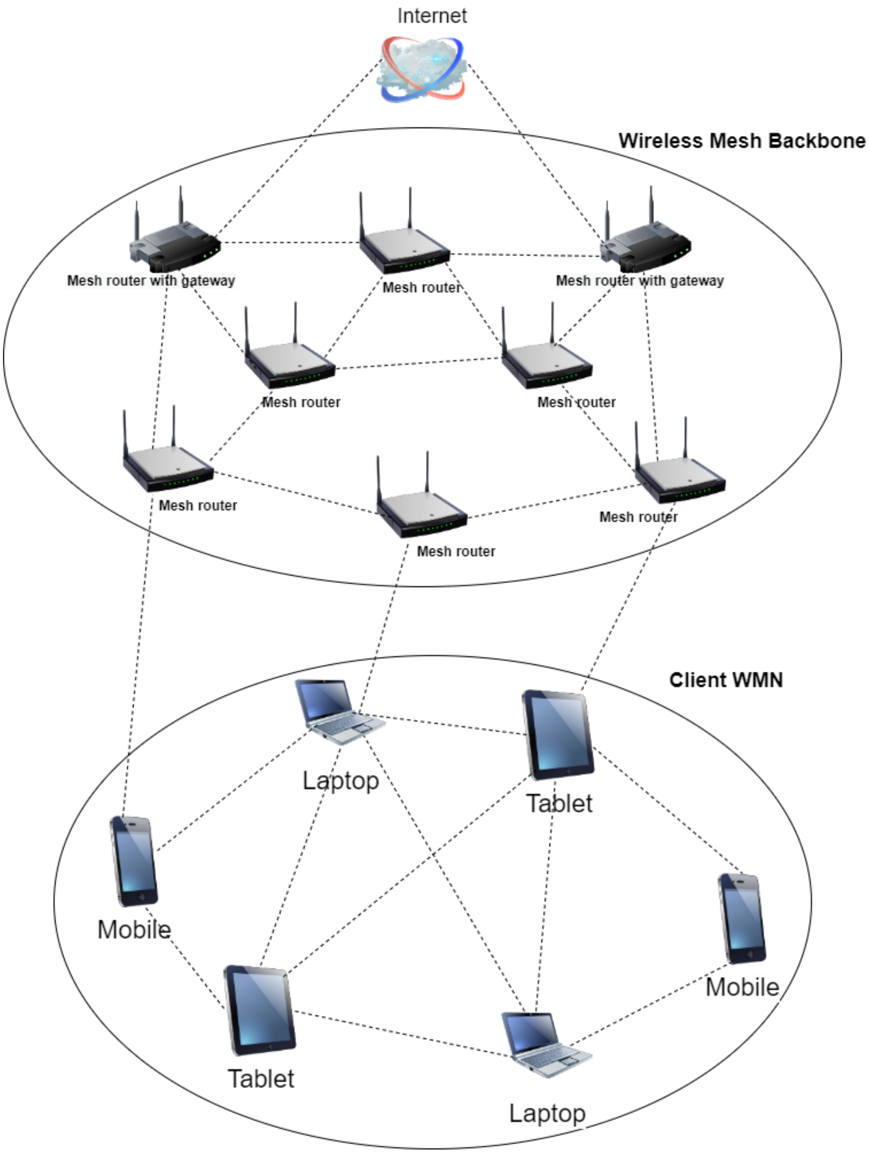


Figure 1.6: Hybrid architecture

1.5 Wireless Mesh Networks Characteristics

Mesh wireless networks have a number of features, including [2, 3, 7, 8, 9]:

Multi-hop Networks

Multi-hops are frequently used by WMNs to get around obstructions, consume less energy, or connect to a node that is outside the communication range of the transmitter.

Multiple radio transmission interfaces

Multiple radio transmission interfaces are built into MRs to enhance the network capacity and the routing functionality.

Mobility

The mobility of nodes in wireless mesh networks varies depending on their kind; while MCs might be immobile or have significant movement, MRs have less mobility.

Energy constraint

The energy constraint depends on the node type. Unlike MCs, MRs have no energy limit because they are powered directly by energy resources.

Interoperability and compatibility WMNs based on IEEE 802.11 standards can support conventional wifi clients and mesh clients. They are also interoperable with other wireless technologies such as Zigbee and WiMax.

Several functionalities of MRs

Some MRs can serve as a gateway or a bridge in addition to their primary duty as a routing device.

1.6 WMNs advantages

WMNs present several advantages such as [3, 4, 10, 7, 8, 9]

Total elimination of wiring

A WMN deployment does not require any wire, which makes it easier to set up, use, and maintain.

Reduced deployment cost

WMNs are exempt from the cost of cabling, which is an additional expense for wired networks. Additionally, the need for intervention is reduced by their independence.

Fault tolerance

If the route between two remote nodes fails during the communication, a connection is created between these two nodes thanks to the multi-path aspect.

Self-configuration and self-organization

WMNs are characterized by self-configuration and self-reorganization, thus they are capable to adapt to the frequent changes in topology and dynamically maintain the connectivity in the event of a failure or outage.

Load balancing

The multi-path aspect, which results in many paths connecting the source nodes with the destination nodes, allows the load balancing to be accomplished.

Scalability

In traditional wireless networks, the network performance degrades as the number of nodes increases. However, with WMNs, adding mores nodes will boost transmission capacity, allowing for better load balancing and alternative routes. In most cases, local packets generated from MCs move more quickly than neighboring packets. Some WMN configurations and communication protocol management techniques are used to achieve this.

Interoperability

WMN are compatible with existing standards including WiMAX, Cellular, Wi-Fi, Zigbee, Bluetooth, Sensor, MANET, Vehicular, etc. As a result, it makes incremental deployment and the reuse of current infrastructures appealing. All of the aforementioned technologies can currently be configured to connect with one another through a WMN or will be able to do so soon. Most of the modifications required for any kind of networks to communicate with one another can be added to the existing standards to keep the interoperability.

1.7 Academic Supporting and Industry Standards

In the context of home networking, business networking, or larger, community- or metro-scale networking, wireless mesh networking is garnering a lot of interest as a low-cost networking platform to offer ubiquitous internet access. Numerous universities are engaged in research on numerous WMN-related topics, such as planning, protocols, applications, and services. In addition, various industry standards groups have developed specifications and protocols for WMNs, such as IEEE 802.11s Mesh WLAN, IEEE 802.15.1 Bluetooth, IEEE 802.15.4 Zigbee, and IEEE 802.16j WiMAX standards [11].

1.7.1 IEEE 802.11s Mesh WLAN

The IEEE 802.11 standards is the preferred solution for low cost data service, it specifies the Medium Access Control (MAC) and physical layers (PHY) specification for wireless devices in WLANs. The 2.4 and 5 GHz unlicensed bands are the main keys to its success. The range (coverage) that WLANs can attain in these bands is constrained by the transmit power restrictions imposed by regulatory constraints. In other hand, however, there is an increasing need for "bigger" wireless infrastructure, which may be deployed anywhere from office/university campuses to entire cities. Thus, data packets must travel over several wireless hops in order to overcome the drawbacks of single-hop communication, necessitating the use of wireless mesh networks. In order to solve the aforementioned need for multi-hop communication, an update to the 802.11 standard called 802.11s has being developed since 2004. In reality, wireless frame forwarding and routing capabilities were introduced at the MAC layer, enabling interworking and security. The default routing protocol for 802.11s is a Hybrid Wireless Mesh Protocol (HWMP), which combines Ad hoc On Demand Distance Vector (AODV) with tree-based routing. Additionally, 802.11s specifies a framework for congestion control, a password-based authentication mechanism, and a key establishment protocol.

1.7.2 IEEE 802.15.1 Bluetooth

The IEEE 802.15.1 standard known commercially as Bluetooth, is focused on Wireless Personal Area Networks (PANs). This standard identifies the necessary mechanisms that must be supplied in the physical and MAC layers of Wireless PANs (WPANs) to enable mesh networking. In WPAN mesh networks, there are two conceivable mesh topologies: full mesh topology and partial mesh topology. Direct connection arrangements are used in full mesh topology. Each wireless node is therefore immediately connected to every other node. In other hand, in a partial mesh topology, certain wireless nodes are connected to every other wireless node while others are only connected to wireless nodes that relay data.

1.7.3 IEEE 802.15.4 Zigbee

This standard is introduced by Motorola, a coordinator can be specified in Zigbee to support mesh topology. In fact, the coordinator is charged of setting up the network topology in a multi-hop manner. Wireless Sensor Mesh Networks (WSMNs) can benefit greatly from this mechanism.

1.7.4 IEEE 802.16j WiMAX

The 802.16j standard was developed by IEEE 802.16 Relay Task Group with the intention of enabling Mobile Multihop Relay (MMR) capabilities. In fact, as a result, a multihop mesh topology is constructed using some WiMAX base stations as relay stations. By using this mechanism, the network coverage can be increased without incurring additional costs for installing fixed line connections.

1.8 Transmission techniques in WMNs

1.8.1 Orthogonal Frequency Division Multiplexing (OFDM)

The principle of OFDM is based on dividing the bandwidth into slices called sub-channels, such that each sub-channel is used as a communication medium between two nodes. This technique is implemented at mesh points of the mesh network. So a Mesh point is equipped with several transmission sub-channels and has several logical interfaces, which decreases the interference and increases the capacity of a mesh point, and consequently it increases the capacity of the network. In addition, the use of the OFDM technique makes it possible to create virtual sub-networks whose nodes use the same sub-channel (see figure 1.7 [6]), which introduces flexibility at the level of the network architecture .

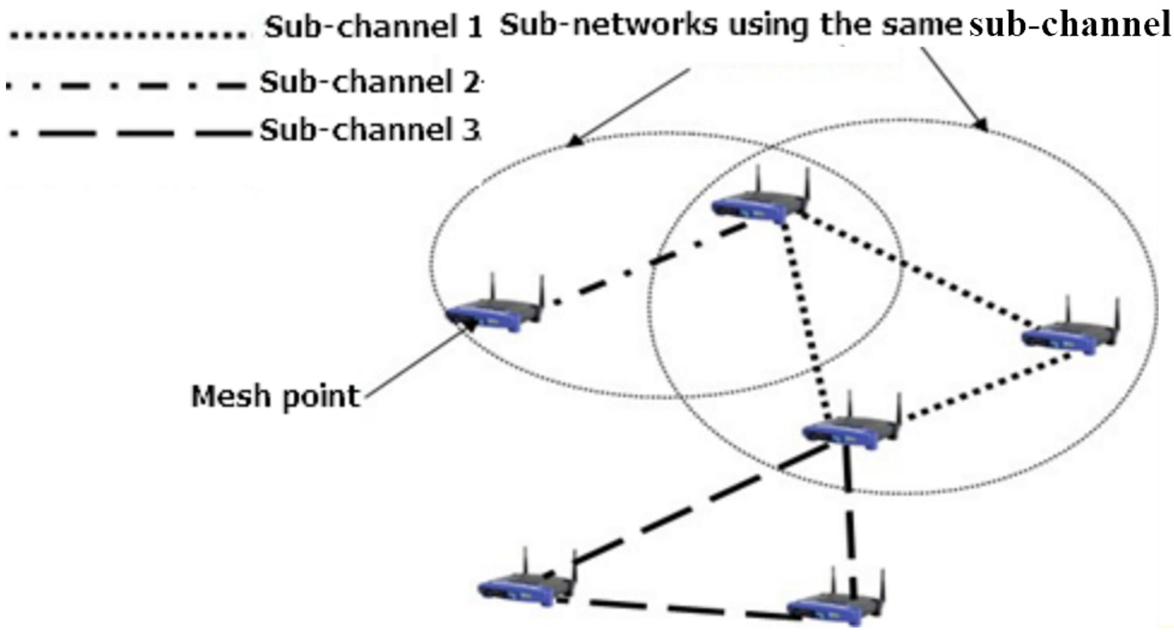


Figure 1.7: Sub-networks created using OFDM technique

1.8.2 Multiple Input-Multiple Output (MIMO)

In general, wireless networks have been successfully deployed in various applications, consequently, the consumption of its resources is increased. MIMO systems have appeared to be able to provide capacities in the transmission side, they make it possible to offer high data rates compared to the classic transmission mode, and proportional to the number of used antennas. The principle of MIMO systems is to equip the nodes of the network with several antennas, these antennas are used during the transmission and reception of data.

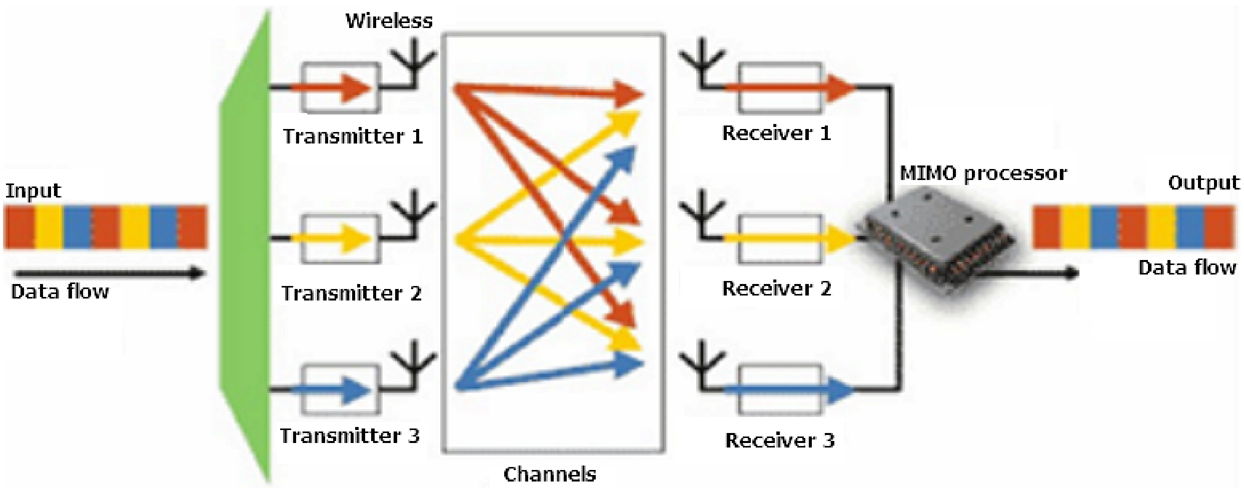


Figure 1.8: MIMO communication technique

As shown in Figure 1.8 [6], the transmitter simultaneously transmits multiple data streams through its antennas (one stream per antenna). The receiver picks up through its antennas transformed and independent versions of the same transmitted signal, then it combines these signals so that the signal results from a lower amplitude variation than the signal picked up by an antenna. WMN used this new technology to meet the needs of its customers under the best conditions (throughput and transmission time). Since the WMN Wireless Backbone is used for providing communication between clients, and also connecting clients to the Internet, MIMO systems are implemented at the physical layer of these Nodes [12].

1.9 Medium access protocols

It is obvious that the performance of the mesh network decreases as the size of the network becomes larger, as well as the traffic around the gateways becomes bulky, which limits the capacity of the network. Therefore the use of medium access protocols is necessary.

1.9.1 Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)

Each node of the network has a transmission zone and an interference zone, when a node establishes a connection with one of its neighbors, the other nodes located in its interference zone must cease all transmission activity under penalty to interfere with communication. If node A wants to send packets to node B, B must be in the coverage area of A. All the nodes located in the interference area of A must be silent, this principle can be achieved by the following steps:

- If a node A wants to send packets, it first listens to the network.
- If a transmission is in progress in its coverage area, the transmission is delayed.
- Otherwise (the medium is free) node A draws a random DIFS (Distributed Inter Frame Space) waiting time and waits for it to elapse. The nodes located in the coverage area of A and which also wish to send packets will do the same task. The first node having finished waiting has the right to transmit, the others suspend the flow of their respective times.
- The selected node then sends an RTS (Ready To Send) message to its destination, this message contains information on the data volume of the packet and its transmission speed.
- If the destination is free (not jammed by other transmissions), it responds with a CTS (Clear To Send) message meaning that it is ready to receive, so the sender begins the data transfer.

- At the end of the communication, the receiving station sends an acknowledgment of receipt in the form of an 'ACK' message (Acknowledgment) signifying that the data has all been successfully transmitted. And the operation can start at the beginning. During all this time (from RTS to ACK) the other nodes which successively hear these messages must cease all communications.

1.9.2 Time Division Multiple Access (TDMA)

TDMA is a medium access technique which facilitates many users to share the same frequency without interference. This technique consists of dividing the time available between the different users into small intervals, called slots. Thus, each user transmits on different time slots. It should be noted that in this technique we do not find the notion of listening to the channel, a node transmits directly if its slot has arrived. To avoid collisions, strong synchronization between users is mandatory.

1.9.3 Hybrid CSMA/TDMA Media Access Control (MAC) protocol

This protocol consists of combining the two techniques CSMA and TDMA in order to take advantage of their advantages: the simplicity of CSMA and its efficiency in the case of low traffic and the optimal use of the bandwidth of TDMA in the case of a heavy traffic.

In mesh networks the nodes are divided into two categories:

- Nodes located in the neighborhood of determined degree (k) of the gateway (where the traffic is important): use the TDMA protocol.
- The rest of the nodes (where the traffic is low): use the CSMA/CA protocol [13]

1.10 Applications scenarios

Some applications cannot be fully supported and guided by other wireless technologies other than WMNs. It was the motivation to create WMNs.

1.10.1 Broadband home networking

Most home networks use IEEE 802.11 technology, which has some limitations. Some examples of these restrictions are communications that must go through Access Points (APs) and the existence of uncovered (dead) regions throughout the home. A proposed approach that can address these issues is the deployment of WMNs using MRs as the communication backbone rather than APs as illustrated in Figure 1.9 [14].

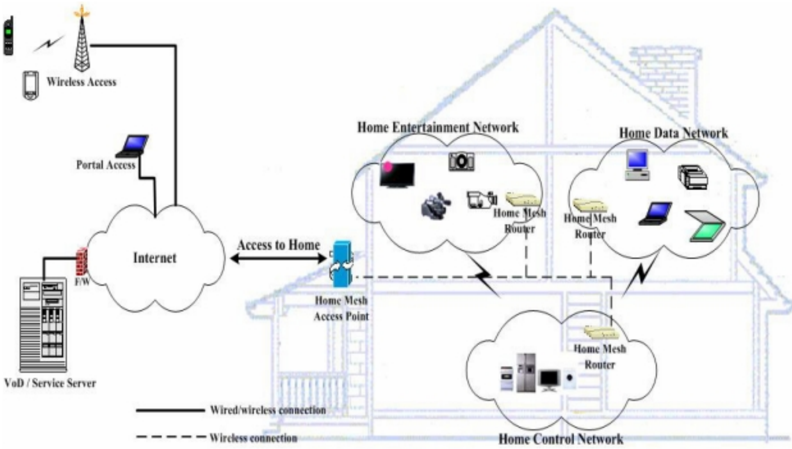


Figure 1.9: WMNs for broadband home networking

1.10.2 Enterprise networking

IEEE 802.11 standard is extensively used in many offices, enterprise networks remain expensive since wired Ethernet connections are required to connect these networks. Ethernet lines can be removed if MRs are used in place of access points, as shown in Figure 1.10 [15]. The size of the business grows, WMNs can readily expand, this form of WMNs is called Wireless Mesh Enterprise Network (WMEN). Many more public and commercial service networking applications, such as airports, hotels, retail centers, convention centers, and sports arenas, can be implemented using the service model of enterprise networking.

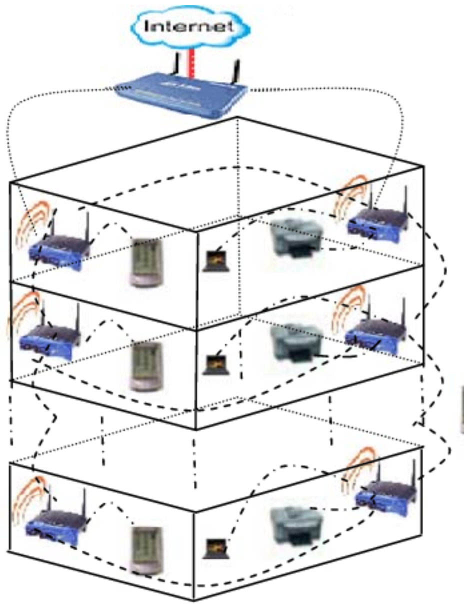


Figure 1.10: WMNs for Enterprise networking

1.10.3 Metropolitan area networking

Communications between users in WMNs are not based on wired links. The use of wireless links in metropolitan networks represents a economic alternative, particularly in remote areas. As shown in Figure 1.11 [16], through the use of the multi-hop principle between users, a service area larger than a house, building or business is offered. Scaling becomes an important factor to take into account in these applications.

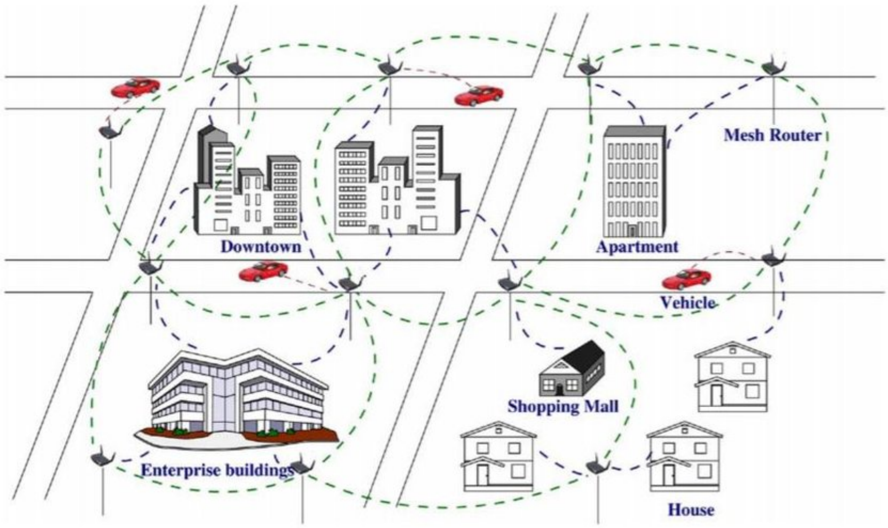


Figure 1.11: WMNs for metropolitan area network

1.10.4 Transportation systems

Despite being constructed on Ethernet cables, IEEE 802.11 and 802.16 have only been used in public areas like stations and bus stops. WMNs may prove to be a much better choice. Additionally, WMNs can assist with connectivity for in-vehicle security, remote control, and passenger information systems. The basic idea underlying these systems is high-speed mobile backhaul from a car to the internet and mobile mesh networks inside the vehicle.

1.10.5 Building automation

As shown in figure 1.12 [17], there are many different types of electrical equipment to be found in an apartment or office building. As a result, this latter equipment needs to be constantly monitored and supervised. In the past, wired networks were used to monitor these devices. Due to the complexity and high cost of the wired network's maintenance, this method is expensive. To solve this issue, Wi-Fi-based networks have been established. However, due to the fact that Wi-Fi-based networks also contain wired distribution systems, they are frequently still expensive and have not produced adequate outcomes. We ought to switch to network routers instead of Building Access Control (BAC) access points to solve these problems, which would greatly decrease the overall cost. Due to the wireless connectivity between the network routers, the deployment process is frequently significantly simpler

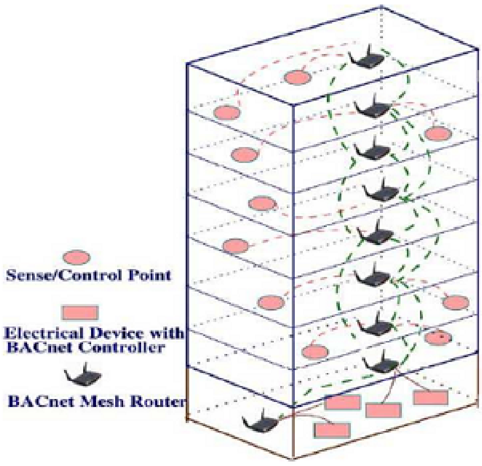


Figure 1.12: WMNs for building automation

1.10.6 Health and medical sciences

Today, in a medical facility, regular data monitoring and diagnostics are conducted using standard wired network technology, which is worthless due to the constant device position changes and massive amounts of data produced by periodic monitoring. As a result, wired networks cannot be used to their full potential. Wi-Fi networks are reliant on Ethernet connections, which might result in a high system cost and complexity. All of the aforementioned problems can be solved by WMNs. An ideal application of WMN in hospital environment is illustrated in Figure 1.13 [18].

1.10.7 Safety and surveillance systems

In today's business environment, security and surveillance are crucial in the enterprise, malls, stores, etc. WMNs perform better than wired networks at enhancing the security of these systems. The capability and freedom that WMN offers are necessary because surveillance systems, image streaming, and video streaming are still the main solutions. An example of the application of WMN for surveillance systems is illustrated in Figure 1.14

Peer-to-peer communication and disaster systems both make use of WMNs. For instance, in the case of emergency networks, firemen fighting fires frequently do not have access to the necessary information. If WMNs are available where they are needed in such situations, locating the areas that require care becomes clear. Similarly, peer-to-peer connection established by wireless networking devices like laptops and PDAs creates a successful solution for information sharing. Again, WMNs are intended to enable these.

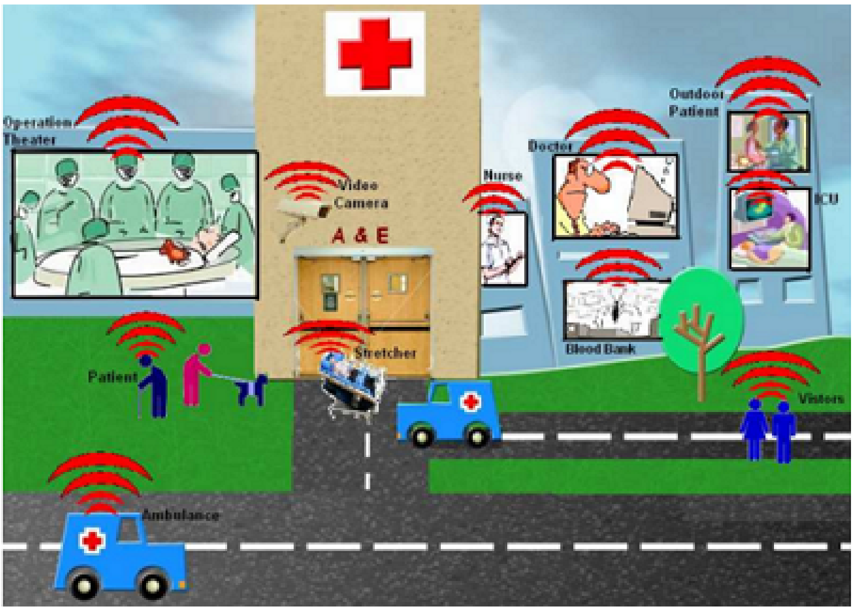


Figure 1.13: Ideal application of WMNs in hospital environment

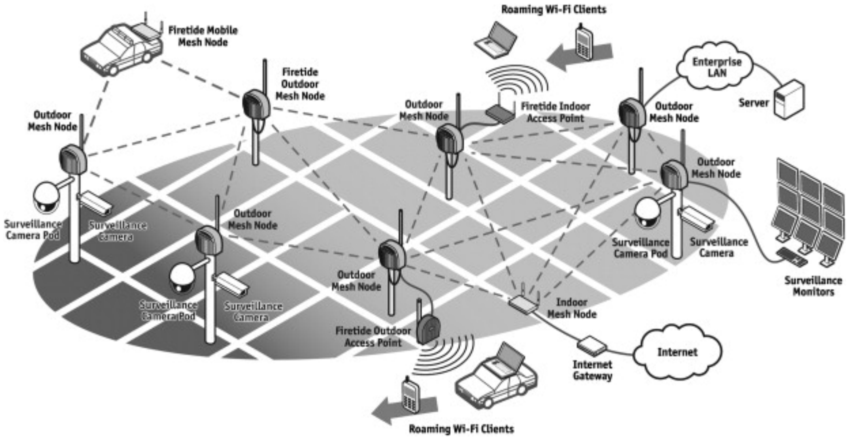


Figure 1.14: WMNs for surveillance systems

1.11 Problems and challenges in WMNs

1.11.1 Physical layer

Single radio single channel, single radio multiple channels, multiple radio multiple channels, and directional antennas are the most current used radio models.

In a single radio single channel, nodes operate in a half-duplex mode . It implies that nodes cannot send and receive a signal at the same time. Thus, there is a huge decrease in bandwidth usage. In a single radio multiple channel system, each node can adjust its single radio to a number of non-overlapping channels to reduce interference and boost capacity. In multiple radio multiple channel model, a node can utilize several non-overlapping channels simultaneously. Furthermore, in directional antennas, multiplexing is employed to decrease interference.

1.11.2 Medium access layer

In comparison with traditional networks, WMNs are characterized by multi-hops. Therefore, to enable communication, the MAC layer must handle multiple hops. In this sense, Multiple Input and Multiple Output (MIMO) radios been proposed to boost WMNs capacity to reduce unauthorized access and under utilization.

1.11.3 Transport layer

Data transfer from one location to another is the transport layer's main function. But current wireless mesh network architectures use conventional ad hoc network transport protocols in the absence of a defined transport protocol. The Transmission Control Protocol (TCP) is ineffective when there is a significant packet loss ratio. Therefore, the use of TCP protocol over a wireless network causes more packet losses, congestion, connection failures, an asymmetric network, and significant Round Trip Time (RTT) changes. Thus, the trend of developing a transport protocol that offers effective data transfer based on present TCP variations is welcomed. UDP is a viable and safer alternative approach as well, it can be utilized with Real-Time Protocol (RTP) for supporting real-time applications. The Rate Control Protocol (RCP), which regulates the quantity of packets sent over a certain path, is capable of handling session control.

1.11.4 Network layer

In WMNs, designing and developing routing protocols is a challenging issue. The following characteristics must be captured by an ideal routing protocol for WMNs [19]:

- **Multiple performance metrics.** Minimum hop-count, expected transmission count (ETX), per-hop RTT, and per-hop packet pair are the most interesting metrics used for selecting the rooting path.
- **Scalability.** In an extremely big wireless network, setting up or maintaining a routing path could take a while. As a result, WMNs must have a scalable routing protocol.
- **Robustness.** WMNs must be robust to link failures or congestion in order to prevent service interruption. Load balancing must be carried out via routing protocols as well.
- **Efficient Routing with Mesh Infrastructure** Assuming that mesh routers have minimal mobility and there are no constraints on power consumption, the routing protocol in mesh infrastructure should be much simpler than ad hoc network routing protocols.

1.11.5 Application layer

WMMs have been applied for various applications due to theirs simple implementation at low cost, easy maintenance, faster capacity, internet connectivity, and so forth. In other hand, Understanding the network's application infrastructure is crucial for managing network heterogeneity via the application layer protocol. In order to accomplish all of the applications, novel techniques at the application layer must be coded.

1.11.6 Topological and Deployment

The design of WMN should be carefully considered in order to provide the high-speed internet connectivity for the end mesh clients. This is a basic problem, and a WMN's ability to assess network performance and provide Quality of Service (QoS) for end users is essential. Planning a WMN consists of determining the optimal placement of mesh routers, taking into account various criterion such as:

- **Cost**: Cost optimization is a fundamental objective when deploying WMNs. This criterion constitutes an objective to be minimized in almost all network optimization problems. It can be represented by various computation costs including the deployment cost, the number of deployed nodes, and the number of installed antennas in the network.

- **Coverage**: Coverage is an important criterion guaranteeing access to users in the desired area. A test point is said covered if it receives a signal power higher than the minimum power threshold or it is within the coverage radius of a mesh router. As an objective, it can be treated by maximizing the number of clients covered by the network and minimizing the number of uncovered points. Also, it can be expressed by maximizing the maximum powers received at service points.

- **Connectivity**: Connectivity is a vital metric ensuring communication within a network. It often refers to the probability that nodes in a network can communicate with each other at any given time. This property is strongly related to the coverage. In fact, it depends on MRs, MGs, and MCs locations and channel conditions and can be determined by using proper propagation prediction tools [20]. As an objective, connectivity is expressed as the biggest sub-network among formed sub-networks regarding the number of mesh nodes.

- **Load-balancing**: Load-balancing is an important parameter allowing the distribution of traffic among different paths. It ensures that each node has equal traffic to forward avoiding over-utilization of channels which is a source of congestion. This metric contributes in network performance enhancement in terms of throughput and delay optimization.

- **Throughput**: Throughput is one of the most important metrics in WMNs. It is defined as the rate of the total number of received packets during a time unit. Throughput can be enhanced by minimizing traffic bottlenecks, minimizing the level of interference, and using multiple channels instead of a single channel.
- **Delay**: Delay is one of the main requirements in WMNs design. It often refers to the accumulated number of communication hops between the MRs and their MGs [21]. For optimizing the network delay, the number of hops between any MR and MG should be lower than a MR-MG hops threshold.
- **Capacity**: Capacity is a key parameter in WMNs Planning. It is measured by the number of MCs per MRs, MR relay load, number of MRs per MGs, link capacity, and radio access interface capacity. Capacity can be increased by minimizing the number of hops and reducing the level of interference.
- **Interference**: Interference is a crucial issue in WMNs that degrades the network performance substantially (considerable packet losses and higher delays). Due to the nature of the transmission medium, nodes located in the same geographic area can interfere with each other when transmitting on the same channel. To deal with this issue, different models are used such as the Protocol Interference Model (PrIM), Physical Interferences Model (PhIM), and Fixed Protocol Interferences Model (FPrIM).

Structured deployment and organic deployment are the two main types of deployment. At a structured deployment, services will be offered in a new region, giving the flexibility of choosing the topology, leading to an improved network performance. On the other hand, if a mesh network is deployed organically, it will be built on top of already-existing infrastructure. Thus, the network architecture has a small number of topology alternatives. Thus, the network architect has a restricted number of topology possibilities.

1.12 Conclusion

In this chapter, we have introduced wireless mesh networks, their characteristics, limits, and applications.

Despite the advantages of WMNs, such networks present many challenges for network operator such as nodes placement problem. In fact, the poor positioning of mesh nodes (MRs/MGs) causes many interference and congestion leading to significant packet loss, high delay, and low throughput. This problem is shown to be NP-hard problem, It is impossible to solve it using traditional exact methods. Approached methods prove to be the most appropriate to solve it.

In the next chapter we will present the optimization problems and their classification, as well as the exact and approached optimization methods. We'll also go over some of the fundamental ideas behind some of the algorithms that are utilized to tackle challenging issues.

CHAPTER 2

OPTIMIZATION METHODS

2.1 Introduction

Optimization occupies a very important place in many fields, such as in operational research, artificial intelligence, biology, mathematics and computer science. A large number of problems can be defined and described as optimization problems. Generally these problems belong to the category of NP-hard problems which do not have an optimal solution. The mesh nodes placement problem is an optimization problem classified as NP-hard, it is impossible to solve it by traditional exact methods. To solve this problem, studies are converging on the use of approximate methods generally inspired by nature. Approximate optimization methods can be classified into two categories, namely heuristics and meta-heuristics.

In this chapter, we will present optimization problem, the notions and concepts related to optimization, the classification of optimization problems regarding several criteria: complexity of the problems, nature of the problems, the number of optima, the type of the objective function and the number of constraints. We then present the optimization methods where we describe the basic principle of some algorithms used for solving complex problems.

2.2 Optimization problem

An optimization problem denoted by $P(X, f)$, is defined as the search of the solution among a set of feasible solutions X (also called decision space or search space), which minimizes or maximizes the objective function f.

In the case of a minimization problem, solving the problem consists to find a solution x^* such that $f(x) \geq f(x^*)$, for any element x in X.

In the case of a maximization problem, solving the problem consists to find a solution x^* such that $f(x) \leq f(x^*)$, for any element x in X.

Let $x = (x_1, \ldots, x_d)$ represents a vector of decision variables, $g(x)$ and $h(x)$ are respectively the inequality and equality constraints, x_I and x_S are respectively the lower and upper boundaries of the search space and $f : D \subset R^d \to R$ is the objective function and D is the feasible domain. The mathematical representation of an optimization problem is given as follows:

$$\text{PO} \begin{cases} \text{ Minimize/Maximize } f(x) \text{ (function to be optimized)} \\ \text{subject to:} \\ g(x) \leq 0 \text{ (n inequality constraints)} \\ h(x) = 0 \text{ (m quality constraints)} \\ x_I \leq x \leq x_S \end{cases} \tag{2.1}$$

It is possible to go from a minimization problem to a maximization problem and vice versa thanks to the following properties:

$$\min_{x \in D} f(x) = \max_{x \in D}(-f(x)) \tag{2.2}$$

$$\max_{x \in D} f(x) = \min_{x \in D}(-f(x)) \tag{2.3}$$

Every point $x \in \mathrm{R}^d$ belonging to D is called feasible solution.

2.3 Notions and concepts relating to optimization

neighborhood

The neighborhood of x, denoted by $V(x)$, is a subset of feasible solutions of X that can be reached from a given transformation of x.

$x^* \in V(x)$ is said to be the neighborhood of x.

Optimum

The optimum is the point where the objective function reaches its minimum or its maximum.

Local optimum

Let $x^* \neq x$ et $V(x^*)$ represents the set of neighboring solutions of x^*†. We say that a solution x^* is a local optimum of the objective function f on $D \subset R^d$ if we have:

$$\forall x \in V(x^*), \begin{cases} f(x^*) \leq f(x) \text{ minimization case} \\ f(x^*) \geq f(x) \text{ maximisation case} \end{cases} \tag{2.4}$$

Global Optimum

We say that a solution x^* is a global optimum of the objective function f on $D \subset R^d$ if we have:

$$\forall x \in D, \begin{cases} f(x^*) \leq f(x) \text{ minimisation case} \\ f(x^*) \geq f(x) \text{ maximisation case} \end{cases} \tag{2.5}$$

The notion of local optimum and global optimum is illustrated in Figure 2.1

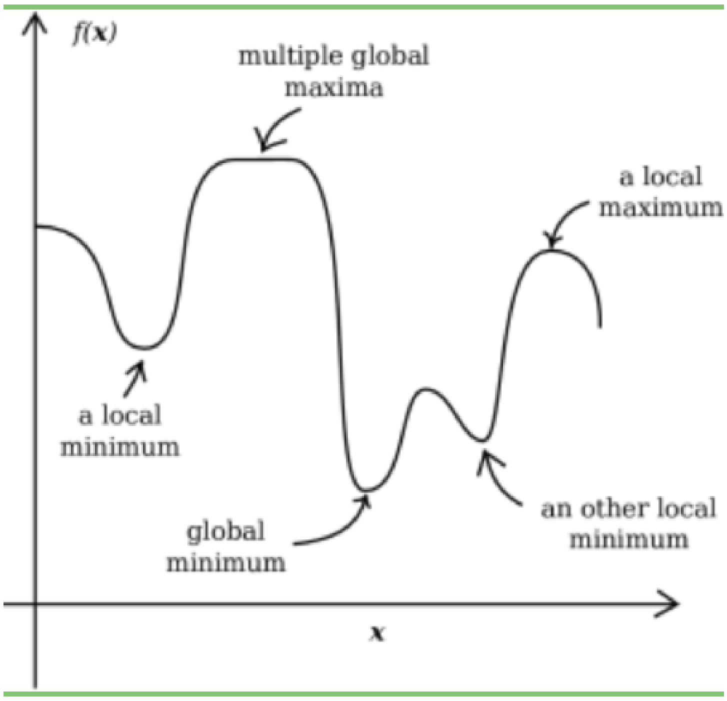


Figure 2.1: A visualisation of global and local optimum

Intensification and diversification

The search for the optimum of an objective function is generally carried out using two fundamental search operators, namely: intensification and diversification.

Intensification

Intensification or exploitation makes it possible to refine and improve the value of a solution found in a certain neighborhood by improving the precision of the optimum.

diversification

Diversification or exploration allows imprecise localization of the global optimum in a larger search space such that the search is not concentrated on a particular area of the search space.

Objective function

The objective function represents the goal to be achieved or reached (minimization or maximization of the function). It defines a space of potential solutions to the problem [22].

Search domain

Search domain is the set of definition domains of the different variables of the optimization problem [22].

Constraints

Represent conditions on the search space that variables must satisfy. These constraints are often inequality or equality constraints, generally used to limit the search space.

2.4 Classification of optimization problems

The classification of optimization problems is a crucial element for their resolution, these problems can be classified according to different criteria (as shown in Figure 2.5): complexity of the problems, nature of the problems, the number of optimums, the type of the objective function and the number of constraints.

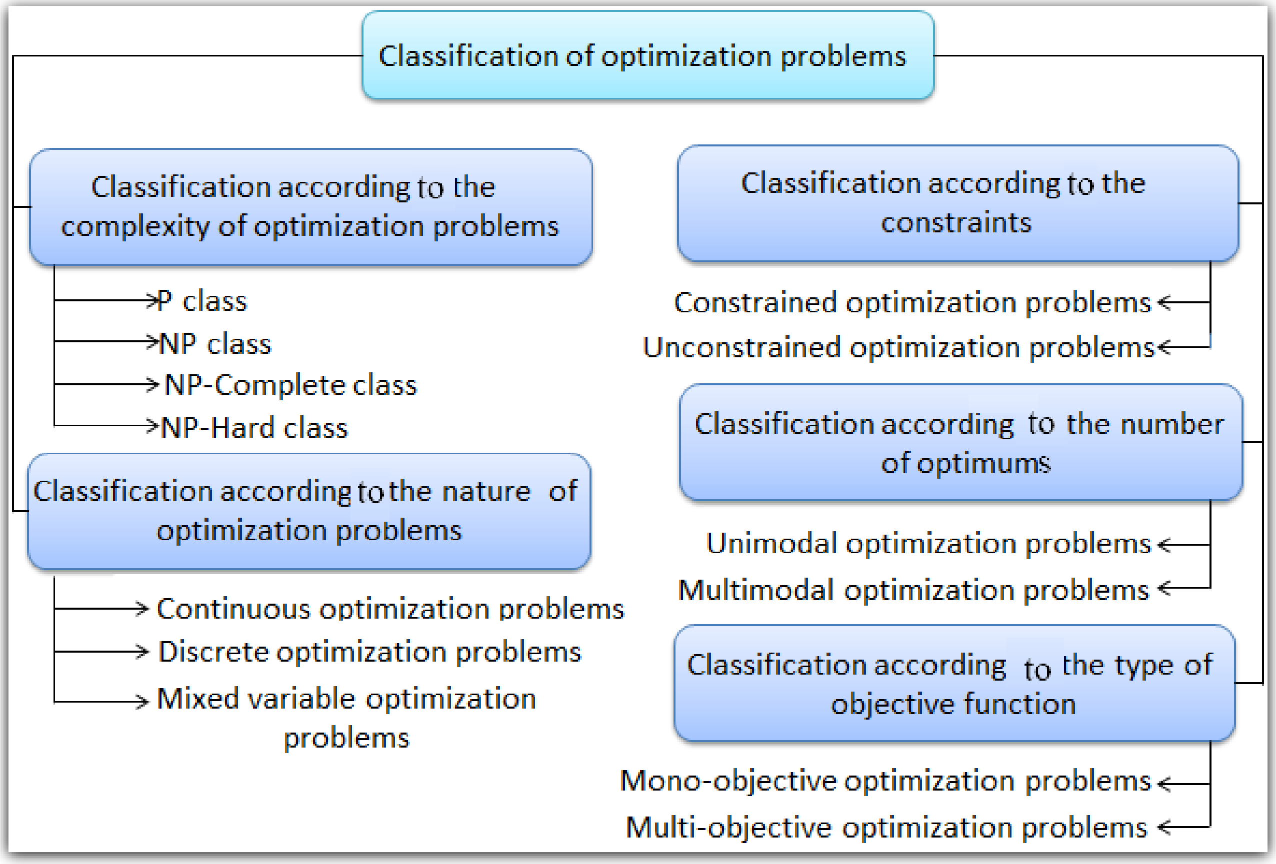


Figure 2.2: Classification of optimization problems

2.4.1 Classification according to the complexity of optimization problems

According to the complexity of optimization problems, generally, four important classes can be identified such as P, NP, NP-Complete, and NP-Hard classes. The relationship between these classes is visualised in Figure 2.3

2.4.1.1 P class

P class includes all relatively simple problems, those for which effective methods are known. Formally, these are the issues for which we can create a deterministic machine with polynomial complexity execution time (the acronym P means "Polynomial time").

2.4.1.2 NP class

NP class contains non-deterministic problems, solved using Turing machine whose execution time is of polynomial complexity (NP acronym means "Non-deterministic Polynomial time"). These problems can also be solved by a polynomial algorithm enumerating the set of possible solutions.

2.4.1.3 NP-Complete

NP-Complete class is a subset within NP class, that contains the most difficult problems. Any NP-Complete issue can be converted (reduced) to it in polynomial time. A problem is NP-Complete when all the problems in NP are reducible to it. If we find a polynomial algorithm for an NP-Complete problem, then we subsequently find a polynomial solution for all the problems of the NP class.

2.4.1.4 NP-Hard class

A problem is said to be NP-Hard if it is more difficult than an NP-Complete problem. More precisely, if there is an NP-complete problem that is reduced to this problem by the Turing reduction. Among the hardest computer science problems we can find: Set Covering Problem, Facility Location Problem, Traveling Salesman Problem, Clustering, and Graph Coloring Problem.

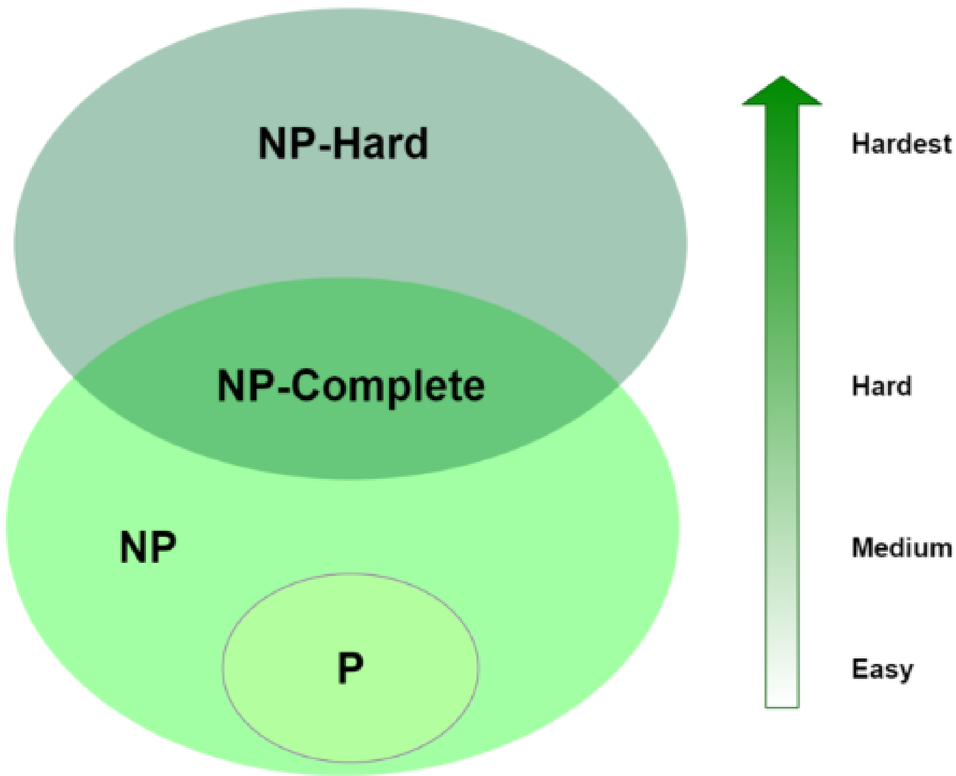


Figure 2.3: Complexity of problems

2.4.2 Classification according to the nature of the search domain

According to the nature of the spaces in which the decision variables take their values, optimization problems can be divided into three classes such as : continuous optimization problems, combinatorial (discrete) optimization problems, and mixed variable optimization problems.

2.4.2.1 Continuous optimization problems

In continuous optimization, the variables in the model belong to continuous set, usually real numbers. There are numerous polynomial algorithms for many of issues in this type of optimization.

2.4.2.2 Combinatorial optimization problems

In contrast to continuous optimization, the variables in a discrete optimization model can be either binary (limited to the values 0 and 1), integer (only integer values are permitted), or more abstract objects selected from sets with finitely many members. In this type of optimization, the problems are often easy to define but generally difficult to solve. Indeed, most combinatorial problems belong to the class of NP-complete problems.

2.4.2.3 Mixed variable optimization problems

Mixed-variable optimization problems involve both continuous and discrete decision variables. The mixed variables definitely increase the complexity of the search space and improve the difficulty of solving these optimization problems. The Mixed-variable optimization problem is said to Mixed Linear Programming (MLP) if the constraints and objective of the problem can be expressed more linearly as a function of its decision variables.

2.4.3 Classification according to the objective function type

2.4.3.1 Mono-objective optimization problems

Mono-optimization problem is defined by a set of variables, a set of constraints, and one objective function. The solution to this problem is a single point.

2.4.3.2 Multi-objective optimization problems

A multi-objective optimization problem is defined by a set of variables, a set of constraints, and a set of objective functions.

In contrast to mono-objective optimization problems, which have a single solution, mono-objective optimization problems have a set of non-dominated solutions, which is known as the Pareto front because of the Pareto dominance concept. A solution is said to be the

Pareto-optimal (or non-dominated) if there is no other feasible solution that can improve one objective without degrading at least one other. Figure 2.4 depicts the set of non-dominated solutions (Pareto front) in the case of two objective functions [23].

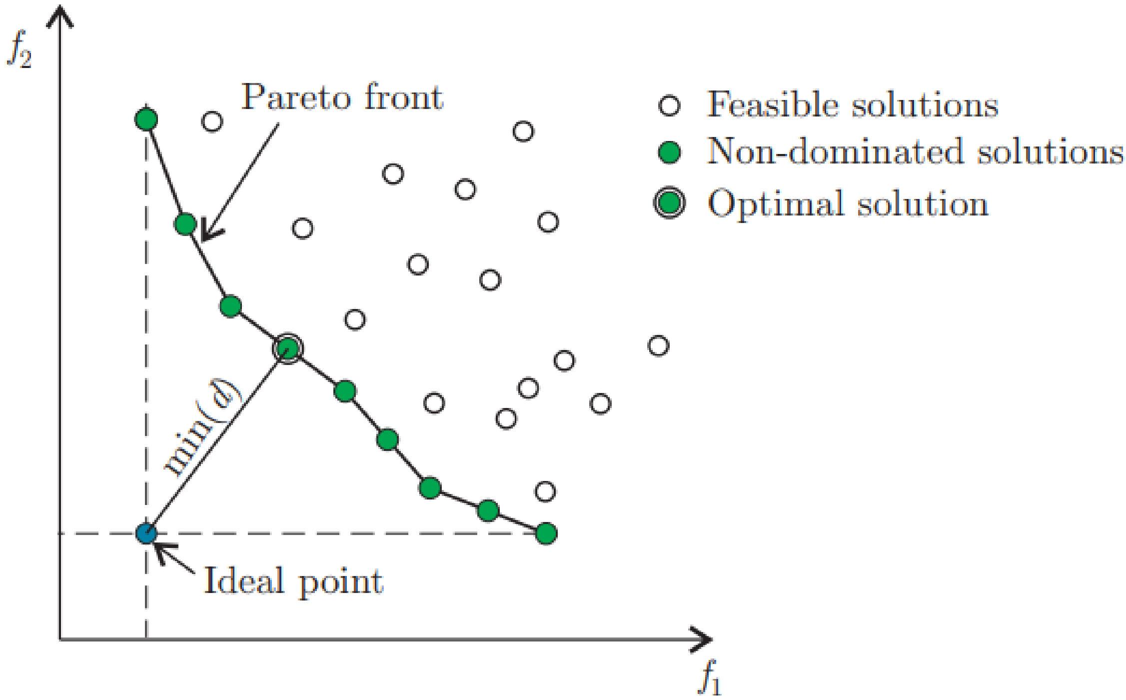


Figure 2.4: Pareto-optimal solutions

Multi-objective optimization problem can be converted to mono-objective optimization problem using sum weighted function [24, 25]. It consists in assigning a weight for each objective for controlling its importance. If several solutions are required, the issue must be resolved numerous times using various weight combinations [26].

2.4.4 Classification according to the constraints

According to the use or not of constraints on the domain space that the variables must satisfy, optimization problems can be classified into two categories, namely constrained optimization problems and unconstrained optimization problems.

2.4.4.1 Constrained optimization problems

Constrained optimization problem is a problem where the objective function is optimized under constraints on the variables. In general, optimization problems are constrained optimization problems, these constraints can have mathematical or symbolic forms.

2.4.4.2 Unconstrained optimization problems

Unconstrained optimization problem is a problem where the objective function is optimized depending on real variables with no restrictions on their values.

2.4.5 Classification regarding the number of optimums

2.4.5.1 Unimodal optimization problems

A unimodal optimization problem is a problem in which the search space contains only a global optimum (a global minimum in the case of a minimization or a global maximum in the case of a maximization).

2.4.5.2 Multimodal optimization problems

A multimodal optimization problem is a problem in which the search space contains several optima (local and global). This problem avoids local optima and allows the localization of several global optima at the same time.

2.5 Optimization methods

To solve optimization problems, several resolution methods have been developed in the literature. In general, to find solutions, optimization methods follow four different approaches such as: the construction approach, the relaxation approach, the neighborhood approach, and the evolution approach. In the field of optimization, we have identified two main categories of optimization methods (Figure 2.5): exact methods that consider the completeness of the solution and approached methods that lose completeness in order to reduce execution time.

2.5.1 Exact methods

Exact optimization methods seek to find the optimal solution by enumerating all the solutions in the search space. Generally, exact methods are used for solving problems of reasonable size. However, the computation time required to find a solution can become very excessive and increases exponentially with the size of the problem and the number of objective functions to be optimized. In this class, we can find the following classical methods: Dynamic Programming and Branch & Bound method.

2.5.1.1 Branch & Bound

Branch and Bound Method is an exact resolution method used for solving combinatorial optimization problems [27]. This method is based on the tree search of an optimal solution by separation and evaluation. First, the set of solutions is separated into smaller subsets. Then, an optimistic evaluation is applied to bound the subsets and choose a solution that is potentially good and better than the current solution. The Branch and Bound method is generally computationally expensive and therefore can only be applied to specific problems. Thus, several improved versions of Branch and Bound algorithm have been proposed. For example, in addition to the method of separation and evaluation, Branch & Cut which uses the secant planes method [28]. The objective of the secant planes method is to add constraints to the linear program to refine it and bring it closer to integral solutions. When solving the relaxed problem gives a fractional solution, the secant planes method is applied at each node of the search tree.

2.5.1.2 Dynamic Programming

Dynamic programming is a recursive method, used to solve a large number of optimization problems. This method is based on the recursive division of a problem into several simpler problems. It is based on Bellman's principle of optimality which says that a sub-problem belonging to an optimal problem is itself optimal. Dynamic programming avoids total enumeration of the search space by eliminating partial decisions that do not lead to the optimal solution [29].

2.5.2 Approached methods

Approached optimization methods have been presented as an interesting alternative for solving large optimization problems if optimality is not paramount. These methods are often inspired by the optimization mechanisms encountered in nature which make it possible to find approached solutions to the problem. They are used to solve problems where we do not know polynomial time resolution algorithms and for which we seek to find an approximate solution to the global optimum. Approximate optimization methods can be divided into two broad categories, namely heuristics and meta-heuristics.

2.5.2.1 Heuristics

Heuristics introduced by Polya in 1945 [30] are designed for NP-hard problems. A heuristic is approximate in the sense that it provides a good solution for relatively little effort, but it does not guarantee the optimality. Heuristics have been proposed, to determine not perfect precision but a satisfactory quality of approximations of exact solutions. These methods were initially based on the knowledge and experience of experts and aimed to explore the search space in a particularly practical way. The most well-known heuristics are greedy algorithms.

a. Greedy algorithms

   Greedy algorithms are a much simpler alternative to program, but the result is not always optimal (except in certain so-called canonical situations). Greedy algorithms build a solution starting from an empty solution and at each iteration a part of solution is constructed. The quality of the constructed solution depends on the heuristic.

2.5.2.2 Meta-Heuristics

Meta-heuristics represent a new generation of powerful approached optimization methods, adaptable and applicable to a large class of problems. They are iterative stochastic methods, which escape local minima and progress towards a global optimum of a function. They allow us to provide good quality of feasible solutions in a reasonable time. We distinguish two classes of meta-heuristics: those based on a single solution and those based on a population of solutions.

a. Single-based meta-heuristics

Single-based meta-heuristics (or trajectory methods) start with a single initial solution and gradually move away from it, building a trajectory in the search space. Essentially, they include the Descent Method (DM), Simulated Annealing (SA), Taboo Search (TS), the Variable Neighborhood Search (VNS), Iterated Local Search (ILS), and their variants.

a.1. Descent Method (DM)

The Descent method is very old enhancement algorithm, it starts from an initial solution x, then chooses a neighbor solution $x^{'}$ that improves the objective function (generally such as $f(x') < f(x)$) at each iteration. DM is formalized as illustrated in Algorithm 1:

Algorithm 1 The pseudo-code of Descent Method

Generate initial solution x
Initialize t
while Stop criterion is not satisfied **do**
    Select $x^{'}$
    if The current solution is better than the previous one **then**
        The previous solution is replaced by the current one
    end if
    t=t+1
end while

a.2. Simulated Annealing (SA)

Simulated Annealing (SA) was developed by Kirkpatrick et al. [31] in 1983. The main concept of SA is based on the annealing theory which simulates the cooling process of metal atoms. Numerous optimization problems, such as the issue of node placement [32, 33, 34], have been addressed using SA. In the SA algorithm, intensification and diversification mechanisms are controlled by the temperature. SA procedure is described in Algorithm 2.

a.3. Tabu Search (TS)

Algorithm 2 The pseudo-code of Simulated Annealing algorithm

Initialize the temperature
Generate initial solution
t=1
while Stop criterion is not satisfied **do** Calculate the temperature
    while The temperature is greater than 0 **do**
        Update the current solution
        if The current solution is better than the previous one **then**
            The previous solution is replaced by the current one
        else
            Decrease the temperature
        end if
    end while
    t=t+1
end while
Return The best solution

Tabu Search (TS) algorithm proposed by Fred Glover in 1986 [35], is advanced local search, based on two tricks such as: the use of the notion of neighborhood and the use of human memory concepts. A memory called a tabu list is used to keep track of the path taken and avoid backtracking. The pseudo-code of TS is illustrated in Algorithm 3.

Algorithm 3 The pseudo-code of Tabu Search

Building initial solution x_0
Calculate the fitness of x_0
Initialize an empty tabu list
Initialize the best solution x_{best}=x_0
initialize t
while Stop criterion is not satisfied **do**
    Choose x_{t+1} in the neighboring of x_t configuration taking into account the taboo list
    Calculate fitness of x_{t+1}
    if fitness of x_{t+1} is better than fitness of x_{best} **then**
        x_{best}=x_{t+1}
    end if
    Update tabu list
    t=t+1
end while

a.4. Variable Neighborhood Search (VNS)

Variable neighborhood search (VNS) is a local search, introduced in 1997 by Mladenovic and Hansen [36]. It is characterized by a simple search principle based on the systematic change of the neighborhood. In contrast to other local search methods that use a single neighborhood to exploit the current solution, VNS uses several neighborhoods in a predefined order. VNS starts the search from an initial solution x. Then, it generates a solution x' of the first neighborhood of the solution x. If x' is better than x based on the objective function, x is replaced by x' and it continues to improve the current solution by generating other solutions of the same neighborhood. Otherwise, the algorithm continues the search with the second neighborhood until the stopping criterion is satisfied. The pseudo-code of VNS is illustrated in Algorithm 4.

Algorithm 4 The pseudo-code of Variable Neighborhood Search

```
Select the set of neighborhood structures Nk, where k ∈ {1, 2, . . . , k_max}
Building initial solution x
Calculate the fitness of x
t=1
while Stop criterion is not satisfied do
    k=1
    while k ≤ k_max do
        Generate x' in the neighboring of x
        Calculate fitness of x'
        if fitness of x' is better than fitness of x then
            x=x'
            K=1
        else
            k=k+1
        end if
    end while
    t=t+1
end while
```

b. Population-based meta-heuristics

Population-based meta-heuristics are iterative methods which improve a population of solutions over the iterations. More precisely, they start from an initial population of solutions, then a new one is generated. This new population is integrated into the current population using a few selection procedures. The iterative process stops when a stopping criterion is achieved. Meta-heuristics with a population of solutions include evolutionary algorithms and swarm intelligence algorithms. The pseudo-code of population-based meta-heuristics is described in Algorithm 5.

b.1. Evolutionary Algorithms (EA)

Evolutionary algorithms (EA) class includes optimization algorithms inspired by biological evolution [37]. They are mainly based of three fundamental elements:

- A generation, or a population includes several individuals, where each individual represents potential solution to concerned problem.
- The best individuals are selected using natural selection and evaluation mechanisms.

Algorithm 5 The pseudo-code of population-based meta-heuristics

```
Generate the population of solutions P
Initialize t
Calculate the fitness of x_0
while Stop criterion is not satisfied do
    Generate the new population P'_t
    Select P_{t+1} population (Pt ∪ P'_t)
    t=t+1
end while
Return the best solution
```

- A population evolution mechanism uses genetic variation operators (crossover and/or mutation) to generate a new descendant individuals called children.

Several evolutionary algorithms have been proposed including Genetic Algorithms (GAs), Evolutionary Strategies (ES), Genetic Programming (GP), Evolutionary Programming (EP), and Differential Evolution (DF). The pseudo-code of evolutionary algorithms is given in Algorithm 6.

Algorithm 6 The pseudo-code of evolutionary algorithms

```
Generate the initial population of solutions P
Initialize t
while Stop criterion is not satisfied do
    Evaluate the population
    Rank the population
    Generate the new population using selection, crossover, and mutation operators
    t=t+1
end while
Return the best solution
```

b.1.1. Genetic Algorithms (GAs)

Genetic Algorithm (GA) developed by Holland et al. in 1970 [38], is the most popular type of EA. GA is inspired from the natural evolution process, so it uses recombination and mutation operators to seek the solution of a problem. The solution is in the form of strings of numbers (traditionally binary). GA was used for solving several problems such as facility layout problem, scheduling, inventory control, forecasting and network design, and others problems (more details about applications of GA can be found in [39]).

b.1.2. Evolutionary Strategies (ES)

The Evolution Strategy is an evolutionary method proposed by Rechenberg in 1973 to solve practical optimization problems. In ES, the solutions are represented using vectors of real numbers and it uses self-adaptive mutation rates. The first version of ES method manipulates a single individual, a

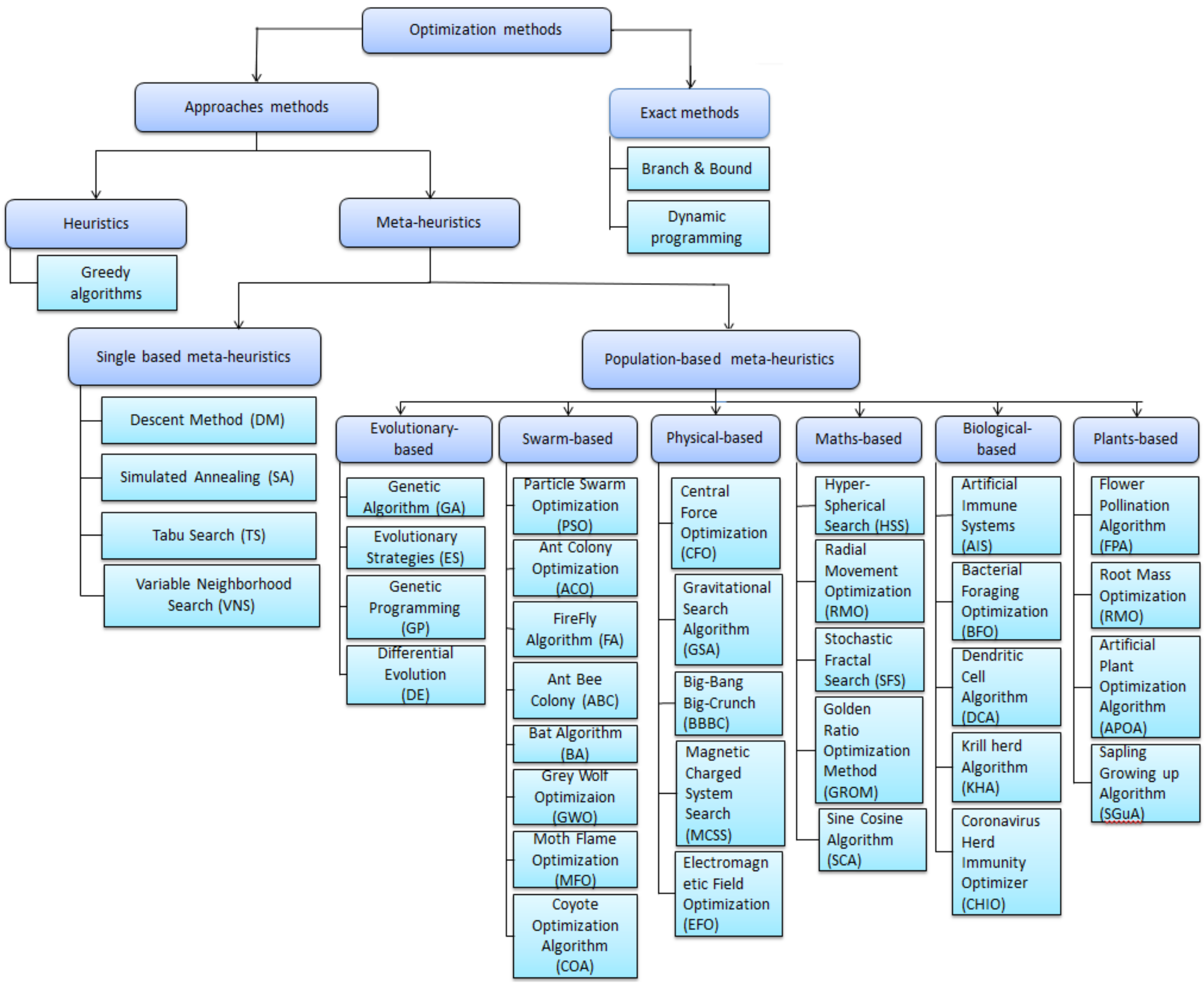


Figure 2.5: Classification of population-based meta-heuristics

child individual is generated by mutation from the parent individual. With the introduction of the recombination variation operator (similar to crossover in GAs), the creation of a new population consists in generating m child individuals from n parent individuals. Routing and networking, biochemistry, optics, and engineering design are some well-known applications of evolution strategies.

b.1.3. Genetic Programming (GP)

Genetic Programming (GP) is an evolutionary method developed by Koza in 1993 [40]. In GP, attributes are represented by programs or instruction sets. Thus, GP generates the solutions in the form of computer programs, and its ability to solve a computational problem is determined by their fitness function. The solution is obtained by using genetic variation operators on the candidate programs with tree encoding.

b.1.4. Evolutionary Programming (EP)

Evolutionary Programming (EP) is an evolutionary method of artificial intelligence, presented and developed by Fogel [41] in 1966 to solve learning problems. EP has no restriction regarding the use of data types of attributes and uses the mutation and replacement operators without taking into account the crossover operator. some suitable areas of application of EP are forecasting, generalization, games, and automatic control.

b.2. Swarm-based Algorithms

Swarm Intelligence Algorithms (SIA) mimics the behavior of swarms of animals that exist in nature such as swarms of fish, insects or birds. Ant Colony Optimization (ACO) [42] and particle swarm optimization (PSO) [43] are the first swarm intelligence algorithms. Other optimization algorithms that come from analogies with natural biological phenomena have been proposed. Among the most significant we cite the Firefly Algorithm (FA) [44], Cuckoo Search [45], Ant Bee Colony (ABC) [46], Bat Algorithm (BA) [47], Grey Wolf Optimization (GWO) [48], Moth Flame Optimization (MFO) [49], Whale Optimization Algorithm (WOA) [50], Dragonfly Algorithm (DA) [51], Coyote Optimization Algorithm (COA) [52], Honey Badger Algorithm (HBA) [53], White Shark Optimization [54], Snake Optimizer (SO) [55], Aquila Optimizer (AO) [56], African Vultures Optimization (AVO) [57], and Salp Swarm Algorithm (SSA). We will present the first developed swarm intelligent algorithms (PSO and ACO).

b.2.1. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) introduced and developed by Russel Eberhart and James Kennedy in 1995 [43], is a simple, effective and global optimization algorithm used to solve various complex problems. It is inspired by the social behavior of animals evolving in swarms such as schools of fish, flocks of migratory birds, or swarms of insects. Each particle is considered as a solution of the problem, and characterized by a velocity V_i and a position X_i. Particle motion is influenced by the following three components:

- A physical component (of inertia): the particle tends to follow its current direction of movement.
- A cognitive component: the particle tends to move towards the best location through which it has already passed.
- A social component: the particle tends to move towards the best site through which its neighbors have already passed.

An example movement of a single particle (index=i), at iteration t, is illustrated in a two-dimensional search space in Figure 2.6. Each particle adjusts its position and moves taking into account its best position P_i and the global best position P_g. Position and velocity are updated according to

Equations 2.6 and 2.7, respectively.

$$V_i(t+1) = wV_i(t) + c_1r_1(P_i(t) - X_i(t)) + c_2r_2(P_g(t) - X_i(t)) \quad (2.6)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2.7)$$

where w is the inertia weight parameter. r_1 and r_2 are random numbers in the range of [0,1]. c_1 and c_2 are acceleration coefficients.

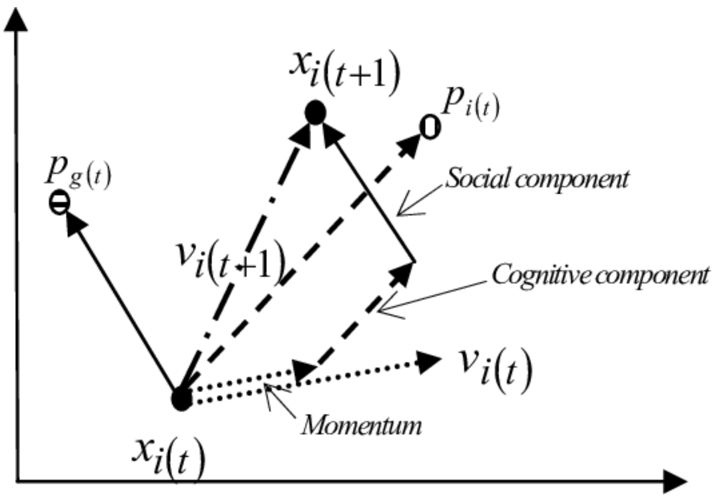


Figure 2.6: Movement strategy of a particle

The pseudo-code of PSO is given in Algorithm 7

Algorithm 7 The pseudo-code of Particle Swarm Optimization

1: Initialize the population of N particles: positions and velocities
2: Evaluate the population and determine P_g t=1
3: **while** St the best soopping criterion is not satisfied **do**
4:     **for** i= 1 to N **do**
5:         Update particle velocity using Equation 2.7
6:         Update particle position using Equation 2.6
7:         Calculate the new fitness value
8:         Do adaptation
9:     **end for**
10:     t=t+1
11: **end while**
12: Return the P_g particle and its fitness value

b.2.2. Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO), introduced by Dorigo in 1990 [42], is a meta-heuristic inspired by nature for solving relatively complex problems. This method was designed to solve the Traveling Salesman Problem (TSP). It is inspired by the behavior of real ants in the search for food . Indeed, the ants manage to find the shortest path between the nest and a source of food by using volatile chemicals called pheromones which they deposit on the ground to mark paths and favorable paths. After some time, the shortest path between the nest and the source presents a greater concentration of pheromone. Consequently, this path will have a greater probability of being chosen and taken by the majority of ants. The general ACO algorithm is decomposed mainly, for each iteration, into three main steps:

- **Solution construction**
  In this step, a colony of artificial ants iteratively generates solutions from the set of possible solutions $S = \left\{s_i^j\right\} (i = 1, 2, \ldots, n; j = 1, 2, \ldots |D_i|)$. First, we start with an empty partial solution $s_p = \emptyset$, then, at each construction step, s_p is extended by adding to it a solution component s_i^j among the set of feasible neighbors $N(s_p) \subseteq S$. The choice of s_i^j in $N(s_p)$ is done in a probabilistic way. Each component $s_i^j \subseteq N(s_p)$ having the probability $P\left(s_i^j \mid s_p\right)$ will be chosen.
- **Daemon actions**
  Daemon actions represent specific actions for a given problem. These actions cannot be performed by each ant separately. Generally, these actions consist of a local search among the built solutions, where only the locally optimized solutions are used to update the pheromone trails.
- **Pheromone update**
  The update of the pheromone traces is carried out in two steps: the step of reducing the pheromone values by a process called evaporation and the step of increasing the pheromone values, associated with a set of best solutions chosen $BSol$, by a process called pheromone deposition. The update of the pheromone traces differs according to the ACO algorithm used. It is usually given by equation 2.8.

$$\tau_{ij} = (1-\rho)\tau_{ij} + \sum_{s \in BSol \mid s_i^j \in S} g(s) \tag{2.8}$$

  Where ρ represents the randomly generated pheromone evaporation rate in the interval $[0, 1]$ and g is the quality function.

b.2.3. Firefly Algorithm (FA)

Firefly algorithm (FA) [44] is population-based meta-heuristic inspired by the blinking behaviour of fireflies. Its concept is based on the bio-luminescence process. FA assumes the following idealized rules:

- All fireflies are unisexual.
- The attractiveness of firefly is proportional to its brightness.

- The brightness decreases as the distance increases.
- If there is no more attractive firefly than a particular one, it will move randomly.

Based on the mentioned rules, the basic steps of FA are described below:

A. A set of initial solutions are randomly generated.

B. Let i and j be two fireflies and their positions be $X_i(x_i, y_i)$ and $X_j(x_j, y_j)$, respectively. The distance r_{ij} between fireflies i and j is calculated as follows:

$$\mathrm{r}_{ij} = \|X_j - X_i\| = \sqrt{(y_j - y_i)^2 - (x_j - x_i)^2} \tag{2.9}$$

C. As mentioned earlier, the attractiveness of a firefly is proportional to its brightness and inversely proportional to the distance between two fireflies. It is calculated as follows:

$$B(r) = \beta_0 e^{-\gamma r^2} \tag{2.10}$$

where $\beta_0(r)$ is the attractiveness of firefly at distance r. β_0 is the attractiveness of firefly at distance 0. γ is the light absorption coefficient.

D. In this step, FA performs the exploration of search space. More precisely, fireflies move according to theirs attractiveness and the attractiveness of other fireflies. When a firefly i is attracted by another firefly j, it moves according to the following equation:

$$X_i = X_i + \beta_0 e^{-\gamma r_{ij}^2}(Xj - Xi) + \alpha(rand - \frac{1}{2}) \tag{2.11}$$

where $rand$ is a random number in [0,1]. α is the randomized parameter.

The pseudo-code of FA is given in Algorithm 8

Algorithm 8 The pseudo-code of Firefly Algorithm (FA)

Set algorithm parameters
Initialize the population of N fireflies
Calculate the fitness of each firefly
t=1
while stopping criterion is not achieved **do**
    for i= 1 to N-1 **do**
        for j= i+1 to N **do**
            if $I_j > I_i$ **then**
                Move firefly i toward firefly j using Equation 2.11
            end if
        end for
    end for
    Evaluate the solutions and determine the best solution
    t=t+1
end while
Return the best firefly and its fitness value

b.3. Physical-based Algorithms

Physics-based Algorithms (PA) mimic the physical rules in the universe. Some of the most popular algorithms are: Central Force Optimization (CFO) [58], Gravitational Search Algorithm (GSA) [59], and Big-Bang Big-Crunch (BBBC) [60]. Other recently developed physics-based algorithms are: Magnetic Charged System Search (MCSS) [61], Electromagnetic Field Optimization (EFO) [62], Water Evaporation Optimization (WEO) [63], Optics Inspired Optimization (OIO) [64], Multi-Verse Optimizer (MVO) [65], Thermal Exchange Optimization (TEO) [66], Sonar Inspired Optimization (SIO) [67], Vibrating Particles System Algorithm (VPSA) [68], and Henry Gas Solubility Optimization (HGSO) [69].

b.4. Maths-based Algorithms (MA)

Maths-based Algorithms (MA) imitate mathematical rules. Some of the most well-known maths-based algorithms are: Hyper-Spherical Search (HSS) algorithm [70], Radial Movement Optimization (RMO) [71], Stochastic Fractal Search (SFS) [72], Golden Ratio Optimization Method (GROM) [73], and Sine Cosine Algorithm (SCA) [74].

b.5. Human-based Algorithms (HA)

Human-based algorithms (HM) are inspired from the human-made events. Some of the most well-regarded human-based algorithms are: Harmony Search (HS) [75], Imperialist Competitive Algorithm (ICA) [76], Fire Work Algorithm (FWA) [77], Teaching Learning-Based Algorithm (TLBA) [78], and Football Game Inspired Algorithm (FGIA) [79].

b.6. Biological-based Algorithms (BA)

Biological-based Algorithms (BA) are developed based on the principles and the inspiration of biological activities. Some of the most well-regarded human-based algorithms are: Artificial Immune Systems (AIS) [80, 81, 82, 83, 84], Bacterial Foraging Optimization (BFO) [85], Dendritic Cell Algorithm (DCA) [86], Krill Herd Algorithm (KHA) [87], and Coronavirus Herd Immunity Optimizer (CHIO) [88].

b.7. Plant-based Algorithms (PA) Some of well-known Plant-based Algorithms (PA) inspired by plant intelligence are: Flower Pollination Algorithm (FPA) [89], Root Mass Optimization (RMO) [90], Artificial Plant Optimization Algorithm (APOA) [91], and Sapling Growing up Algorithm (SGuA)[92].

2.6 Conclusion

In this chapter, we have presented the optimization problem, the notions and concepts related to optimization, the classification of optimization problems according to several criteria. We also presented the optimization methods where we described the basic principle of some algorithms used for solving complex problems.

In the next chapter, we will present in detail an improved version of Moth Flame Optimization for solving the mesh routers placement problem in Wireless Mesh Networks.

CHAPTER 3

THE MESH ROUTERS PLACEMENT PROBLEM IN WMNS BASED ON SWARM INTELLIGENCE METHA-HEURISTICS

3.1 Introduction

This chapter suggests an enhanced version of Moth Flame Optimization (MFO), called ECLO-MFO, based on the integration of three strategies including the chaotic map concept, the Lévy flight strategy (LFD), and the Opposition-Based Learning (OBL) technique to enhance the optimization performance of MFO.

In this chapter, we present a reminder of the methods and approaches proposed in the literature to solve the mesh router nodes placement problem in WMNs, we then describe the formulation of the mesh router nodes placement problem, we also describe ECLO-MFO approach applied to solve the mesh router nodes placement problem in WMNs, and finally we evaluate their performance and compare its characteristics with other optimization methods.

3.2 Related works

WMN nodes placement is known to be an NP-hard problem [93]. So meta-heuristics have been presented as successful optimization algorithms to solve it providing acceptable solutions in a reasonable execution time.

Several works based on meta-heuristics have been proposed in the literature to solve the nodes placement problem in WMNs. Most of the proposed works considered stationary topology [94, 95, 96, 97, 98, 99, 100, 101, 102] while others investigated the dynamic placement of mesh nodes subject to client mobility [103, 104, 105, 106].

To deal with the static variant of the WMNs nodes placement problem, three algorithms have been proposed by Xhafa et al., including Simulated annealing (SA) [94], Hill Climbing (HC) [95], and Tabu Search (TS) [96]. The three algorithms were evaluated in terms of user coverage and network connectivity. The evaluation is done based on 48 benchmark instances using different mesh clients distributions with different grid sizes.

In the work of Nouri et al.[97], an accelerated PSO algorithm (APSO) was also proposed to tackle the mesh routers placement problem in a static environment. APSO was validated in terms of coverage and connectivity and results confirmed its effectiveness when compared with the linearly decreasing weight PSO algorithm.

In the same context, Sayad et al. proposed three new algorithms based on Chemical Reaction Optimization (CRO) algorithm [98], Firefly optimization (FA) algorithm [99], and Electromagnetism-like Mechanism (EM) meta-heuristic [100]. These algorithms were validated using many generated instances with various number of mesh clients and mesh routers, taking into account the coverage and connectivity metrics. Obtained results confirmed the superiority and effectiveness of these algorithms in terms of user coverage and connectivity.

Evolutionary algorithms (e.g. Genetic Algorithm (GA)) have been popular optimization algorithms in this area too [101, 102]. For instance, the mesh router nodes placement problem was solved by Xhafa et al. [101] as a facility location problem using GA. It took into account user coverage and network connectivity metrics. In [102], an improved GA based on coupling GA with Minimum Spanning Tree (MST) was proposed to optimize cost and coverage metrics.

In [103, 104, 105, 106], several methods have been proposed to tackle the dynamic variant of the mesh nodes placement issue. An improved PSO algorithm based on the integration of restriction coefficient into PSO was proposed in [103] to deal with this problem. In another similar work, Lin et al. [104] proposed an improved BA based on the incorporation of the dynamic search scheme into the original BA. The improved BA was validated based on 10 instances, taking into account the coverage and connectivity parameters. In [105], authors focused on the so-called social-aware dynamic router nodes placement in WMNs. They solved this problem using an enhanced PSO including a social-supporting vector, called a social-based-PSO. SA approach was applied in [106] to find the dynamic placement of mesh routers. In addition to user coverage and network connectivity, this approach reduces the average distance traveled by routers. Table 3.1 summarizes some representative works using meta-heuristics for solving the WMNs nodes placement problem.

3.3 Mesh router nodes placement problem formulation

In this section, we describe the system model and the formulation of the mesh router nodes placement problem.

Table 3.1: Summary of some existing WMNs nodes placement representative works

Algorithms	References	Environment	Location	Metrics		
				Cost	Coverage	Connectivity
SA	Xhafa et al.[94]	Static	Discrete		x	x
HC	Xhafa et al.[95]	Static	Discrete		x	x
TS	Xhafa et al.[96]	Static	Discrete		x	x
APSO	Nouri et al. [97]	Static	Continuous		x	x
RCO	Sayad et al. [98]	Static	Discrete		x	x
FA	Sayad et al. [99]	Static	Continuous		x	x
EM	Sayad et al. [100]	Static	Continuous		x	x
GA	Xhafa et al. [101]	Static	Discrete		x	x
Improved GA	Tang et al.[102]	Static	Discrete	x	x	x
MOGAMESH	De marco [107]	Static	Discrete	x	x	x
MOGA and NSGAII	Bello et al. [108]	Static	Discrete	x	x	x
Enhanced PSO	Lin [103]	Dynamic	Continuous		x	x
Improved BA	Lin et al. [104]	Dynamic	Continuous		x	x
Social based-PSO	Lin et al. [105]	Dynamic	Continuous		x	x
SA	Sayad et al. [106]	Dynamic	Continuous		x	x
COA	Proposed method	Static	Continuous		x	x
ECLO-MFO	Proposed method	Static	Continuous		x	x

3.3.1 System model

WMN can be described mathematically as an undirected graph $G = (V, E)$ where V is the set of network vertices (nodes) and E is the set of edges (links) between these vertices. The network G is formed by a set of disjoint sub-networks. In this chapter, we consider the WMN with two types of nodes such as mesh clients and mesh routers. Thus $V = MRS \cup MCS$ where :

- MR is the set of m mesh routers: $MRS = \{mr_1, mr_2, ..., mr_m\}$, Each mesh router is equipped with radio interface with the same coverage radius $CR_1 = CR_2 = ... = CR_m$. Two mesh routers mr_i and mr_j can be connected if and only if the distance between them does not exceed two time the coverage radius CR i.e. $d(mr_i, mr_j) \leq 2CR$.
- MC is the set of n mesh clients $MCS = \{mc_1, mc_2, ..., mc_n\}$, we assume that mesh clients are randomly distributed in $2D$ rectangle area of dimension WxH. A mesh client mc_i is said covered by a mesh router mr_j if it is within the coverage radius of this router: $d(mc_i, mr_j) \leq CR$. It can be associated at most to one router. It can be within coverage radius of various routers but it is associated with the closest router.

3.3.2 Problem Formulation

As per the nature of studied environments (static or dynamic) and the nature of deployment spaces (discrete or continuous), several variants of the WMN router nodes placement problem can be found. In this paper, the static continuous mesh routers nodes placement problem was considered. Therefore, the main goal is to find the optimal placement of m mesh routers in a $2D$ area of dimensions WxH, depending on the location of n mesh clients.

The problem studied in this chapter considers two main objectives that need to be optimized:

- User coverage: It represents the number of covered users by at least one mesh router according to the following equation:

$$\Psi(G) = \sum_{i=1}^{n} (max_{j \in \{1,...m\} \sigma_{ij}}) \tag{3.1}$$

where σ_{ij} defines the coverage variable represented as follows:

$$\sigma_{ij} = \begin{cases} 1 & if\ mesh\ client\ c_i\ is\ covered\ by\ mesh\ router\ r_j, \\ 0 & Otherwise. \end{cases} \tag{3.2}$$

- Network connectivity: It is defined as the geant sub-network among k formed sub-networks with regard to the number of mesh nodes (mesh routers and mesh clients). It is calculated as follows:

$$\Phi(G) = Max_{i \in \{1,...k\}} |G_i| \tag{3.3}$$

where $|G_i|, i \in \{1, k\}$ is the size of i^{th} sub-network and $G = G_1 \cup G_2 \cup ... \cup G_k$.

Figure 5.8 illustrates WMN instance G with 6 MRs and 20 MCs randomly dispersed in rectangle deployment area of $2000m \times 2000m$. MRs and MCs are presented by blue and green points, respectively, while the connection between mesh nodes is presented by a solid line. It can be seen that the network G is composed of 11 sub-nets $G = G1, G2, ...Gt$, $t \in \{1, 11\}$. S is the set of values $= \{10, 1, 1, 4, 1, 1, 1, 3, 2, 1, 1\}$ that refers to the sizes of sub-nets $G1, G2, ...Gt$, respectively. It is clearly seen that the network connectivity of the biggest sub-net is equal to 10 which refers to the size $G1$ sub-net. $G1$ is composed of 3 mesh routers and 7 mesh clients.

In this chapter, we aim to determine the best locations for a given number of MRs in order to maximize simultaneously both client coverage and network connectivity metrics. The multi-objective problem was turned into a mono-objective problem, using the weighted sum function [24, 25]. It consists in assigning a weight for each objective for controlling its importance. In this sense, Client coverage and network connectivity are combined to specify the weighted sum fitness function f, which is used to evaluate the quality of solutions. The objective function is given as follows:

$$f(Pmo_i)) = \lambda . \frac{\Psi(G)}{n} + (1 - \lambda) . \frac{\Phi(G)}{m + n} \tag{3.4}$$

Where G is the graph associated with the solution Pmo_i and λ is a floating parameter with a value in the range $[0, 1]$ that controls the relevance of metrics.

The problem to be solved is viewed as a maximization problem in accordance with this definition of the objective function. However, most of optimization techniques were designed to solve a minimization problem. As a result, we must convert our objective function to a

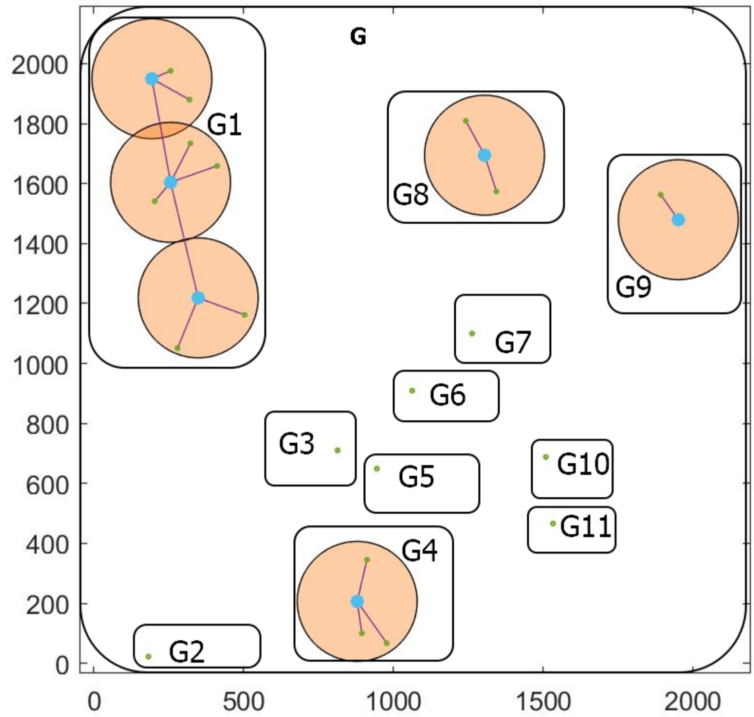


Figure 3.1: An example of WMN instance

minimization function. As result, we have created a new function g, which is as follows:

$$g = 1 - f(Pmo_i) \tag{3.5}$$

In this work, the solution of the mesh router nodes placement problem is represented by an array $Pmo^p = \{x^p_{mo-1}, y^p_{mo-1}, x^p_{mo-2}, y^p_{mo-2}, ...x^p_{mo-m}, y^p_{mo-m}\}$, where (x^p_{mo-k}, y^p_{mo-k}) are the (x, y) coordinates of the mesh router $mo - k$, with $0 \leq x^p_{mo-k} \leq W$ and $0 \leq y^p_{mo-k} \leq L$; $\forall\ m \in \{1, 2, ...m\}$. The solution of setting up eight mesh routers in a $2000m \times 2000m$ deployment area is represented by the array in Table 3.2.

Table 3.2: Representation of the solution of deploying eight mesh routers

mr_1	mr_2	mr_3	mr_4	mr_5	mr_6	mr_7	mr_8
$(200, 20)$	$(15, 450)$	$(400, 700)$	$(140, 900)$	$(1000, 1400$	$(1500, 1100)$	$(300, 1800)$	$(400, 1900)$

3.4 Enhanced Moth Flame Optimization

This section aims to describe MFO, chaotic maps, Lévy distribution, and OBL concepts. Then, we will describe the structure of the enhanced version of MFO.

3.4.1 Moth–Flame Optimization Algorithm

The main inspiration of MFO algorithm [49] is the navigation strategy (movement) of moths at night around flames [49]. This strategy is called transverse orientation. In the mathematical model of MFO algorithm, moths positions are the problem variables, and flames are the best solutions obtained so far [49].

Moths positions are represented by the following matrix:

$$Pmo = \begin{bmatrix} Pmo_{1,1} & Pmo_{1,2} & ... & ... & Pmo_{1,d} \\ Pmo_{2,1} & Pmo_{2,2} & ... & ... & Pmo_{2,d} \\ . & . & . & . & . \\ . & . & . & . & . \\ Pmo_{nm,1} & Pmo_{nm,2} & ... & ... & Pmo_{nm,d} \end{bmatrix}$$

where nm represents the number of moths and d refers to the search space dimension. The fitness values of moths are stored in an array as follows:

$$OPmo = \begin{bmatrix} OPmo_1 \\ OPmo_2 \\ . \\ . \\ OPmo_{nm} \end{bmatrix}$$

Note that the fitness value of each moth is calculated using the following objective function:

$$OPmo(i) = objectivefunction(Pmo(i, 1 : d)) \tag{3.6}$$

On the other hand, flames positions are given by the following matrix:

$$Pfl = \begin{bmatrix} Pfl_{1,1} & Pfl_{1,2} & ... & ... & Pfl_{1,d} \\ Pfl_{2,1} & Pfl_{2,2} & ... & ... & Pfl_{2,d} \\ . & . & . & . & . \\ . & . & . & . & . \\ Pfl_{nm,1} & Pfl_{nm,2} & ... & ... & Pfl_{nm,d} \end{bmatrix}$$

where nm is the number of flames and d represents the search space dimension.

The fitness values of flames are stored in an array as follows:

$$OPfl = \begin{bmatrix} OPfl \\ OPfl \\ . \\ . \\ OPfl \end{bmatrix}$$

The position of the moth is updated around the corresponding flame using the following equation:

$$Pmo_i = S(Pmo_i, Pfl_j) \tag{3.7}$$

where Pmo_i represents the *i-th* moth, Pfl_j refers to the *j-th* flame, and S is the spiral function.

The logarithmic spiral function is chosen for updating the mechanism of MFO algorithm, which is given as follows:

$$S(Pmo_i, Pfl_j) = D_{i,j} * e^{bt} * Cos(2\Pi t) + Pfl_j \tag{3.8}$$

where b is a constant that defines the shape of the logarithm spiral, t is a random number in the range of $[-1, 1]$, and $D_{i,j}$ indicates the distance between the *i-th* moth and the *j-th* flame. $D_{i,j}$ can be represented as:

$$D_{i,j} = |Pfl_j - Pmo_i| \tag{3.9}$$

To obtain a good balance between intensification and diversification, an adaptive scheme is employed to update the number of flames as follows:

$$numbflames = round(R - k * \frac{R-1}{it_{max}}) \tag{3.10}$$

where R denotes the maximum number of flames, k is the current iteration number, and it_{max} represents the total number of iterations. The pseudo-code of the MFO is given in Algorithm 9 [49] and its flowchart is given in Figure 3.5.(a).

Algorithm 9 The pseudo-code of MFO Algorithm

Input:
nm: Number of moths
d: Dimension of the problem
lb: Lower bound
ub: Upper Bound
it_{max}: Maximum number of iterations
Output:
Pmo_{best} the best solution

for i=1 to nm **do**
  Initialize randomly the position of *i-th* moth in the search space
  $OPmo(i)$ = Objective function($Pmo(i, 1 : d)$)
end for
for k=1 to t_{max} **do**
  Update number flames (Eq.3.10)
  if $k == 1$ **then**
    $Pfl = sort(Pmo)$
    $OPfl = sort(OPmo)$
  else
    $Pfl = sort(Pmo_{k-1}, Pmo_t)$
    $OPfl = sort(OPmo_{k-1}, OPmo_t)$
  end if
  for i= 1 to nm **do**
    for j= 1 to d **do**
      Calculate $D_{i,j}$ with respect to its corresponding moth (Eq.3.9)
      Update $Pmo(i, j)$ with respect to its corresponding moth (Eq.3.7)
    end for
  end for
  for i= 1 to nm **do**
    $OPmo(i)$ = Objective function($Pmo(i, 1 : d)$)
  end for
end for
Return the best solution

MFO has a number of features such as simplicity and easy implementation. However, MFO suffers from stagnation in local optima because it focuses on exploitation rather than exploration. Consequently, MFO requires an enhancement to deal with the mesh routers placement problem. The optimization discipline proposes various strategies to enhance the performance of meta-heuristics such as OBL, chaos maps, Lévy distribution, Cauchy mutation, selection schemes, and others. In the rest of this section, we describe the three strategies used to enhance the MFO performance.

3.4.2 Lévy Flight Distribution

The Lévy-flight [109] originally proposed by the French mathematician Paul Pierre Lévy in 1926 is considered as a non-Gaussian random process with random walks derived from the Lévy stable distribution as shown in Figure 3.2. It has been incorporated favorably into several meta-heuristic search algorithms to provide faster convergence, improve the diversity of the population, exploit the search space in a much better way, and avoid premature convergence. The formula for the Lévy distribution is a simple power-law expression $L(s) \sim s^{-1-a}$ where $0 < a < 2$. Lévy flight distribution can be stated mathematically as follows:

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \exp\left(-\frac{\gamma}{2(s-\mu)}\right) \frac{1}{(s-\mu)^{3/2}}, & 0 < \mu < s < \infty \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

where μ, s, and γ represent the minimum step size, sample size, and control parameters of Lévy distribution, respectively.

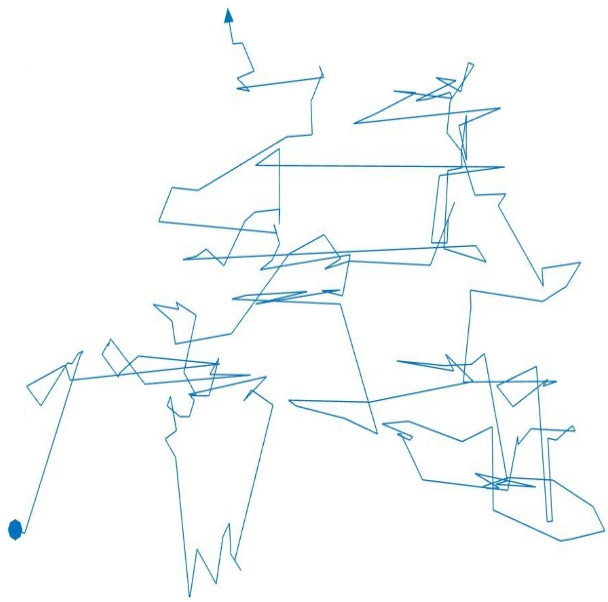


Figure 3.2: Plot of the two dimensional variables from Lévy flights distribution started from the origin (marked with a bold point) and plotted 200 points with a=1.4

3.4.3 Chaotic map

Chaos is a deterministic method used to analyze the behavior of nonlinear and dynamic systems. It has many important characteristics such as ergodicity, regularity, non-converging, non-periodic, bounded, unpredictable, non-repetitive, and stochastic. These characteristics have been transformed into various mathematical equations called chaotic maps used to generate random parameters in meta-heuristics. Integrating chaotic maps with meta-heuristics improves searching capability, increases the convergence rate, and avoids being trapped into local optima. Several chaotic maps have been presented in the literature such as Logistic map, Tent map, Sine map, Gauss map, Sinusoidal map, Chebyshev map, Piecewise map, Iterative map, and Circle map [110, 111]. Sine map is one of the most representative commonly used chaotic maps with simple operations and well dynamic randomness. The equation of Sine map is described as follows:

$$SM_{k+1} = \frac{ac}{4} \sin\left(\pi SM_k\right), 0 \leq SM_O \leq 1 \tag{3.12}$$

Where SM_k is the chaotic map value at the *k-th* iteration, it is in rage of [0, 1]. Here ac is a control parameter and $0 < ac \leq 4$.

According to [112], the sine map seems to be a chaotic logistic map when the factor ac is equal to 4. Again, the sine map exhibits chaotic behavior when ac is near to 4. Specifically, when ac falls within the range of $[3.48, 4]$. Figure 3.3 shows the chaotic dynamics of Sine map SM.

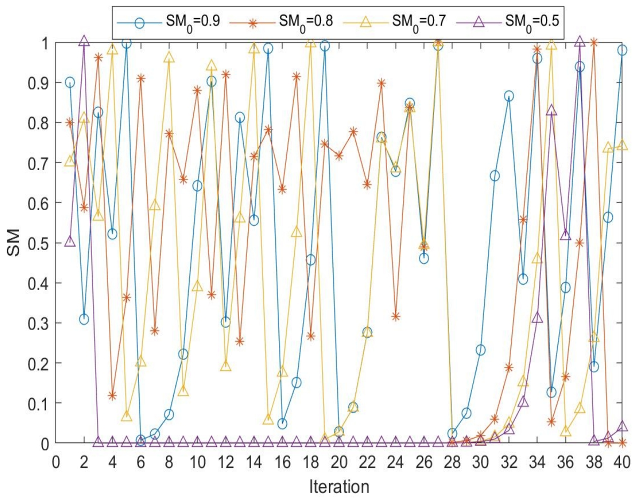


Figure 3.3: The chaotic dynamics of Sine map

3.4.4 Opposition-based learning

The opposition-based learning (OBL) strategy introduced originally by Tizhoosh [113] in 2005 is a well-regarded scheme in the field of machine and computational intelligence. It has been successfully combined with many meta-heuristic optimization algorithms to enhance their convergence speeds and explore the search space effectively. The main concept of OBL strategy is based on opposite numbers to approach the solutions. The opposite of the real number $Pmo \in [lb, ub]$ can be mathematically defined by the following equation:

$$\bar{Pmo} = ub + lb - Pmo \tag{3.13}$$

The opposite point, on the other hand, is computed in the same way as a reflected point when calculated through the center point $((lb + ub)/2)$ for one dimension space

This definition can be extended to multidimensional search space as follows:

$$\bar{Pmo_i} = ub_i + lb_i - Pmo_i, \quad i = 1, 2, \ldots, d \tag{3.14}$$

where $\bar{Pmo} \in R^d$ is the opposite vector from the real vector $P \in R^d$.

3.4.5 The proposed ECLO-MFO for solving the mesh router nodes placement problem

The structure of the suggested ECLO-MFO is described in this section. In order to improve the optimization performance of MFO, three strategies are incorporated including Lévy flight distribution, chaotic map, and opposition-based learning. The integration of Lévy flight distribution with the original MFO enriches searching behavior and avoids stagnation in local optimum. The chaotic sequence is used to increase the chaotic stochastic behavior of the MFO algorithm. Thus, the combination of chaotic sequence and Lévy distribution may yield better results as shown in Figure 3.4. In this sense, the *i-th* moth performs both Lévy distribution and chaotic sequence after the position updating, which is formulated as follows:

$$Pmo_i^k = Pmo_i^k + (SM_k - 0.5) * LFD \tag{3.15}$$

Where LFD is randomly generated by the Lévy flight distribution. SM_k is the chaotic sequence generated by the sine chaotic map.

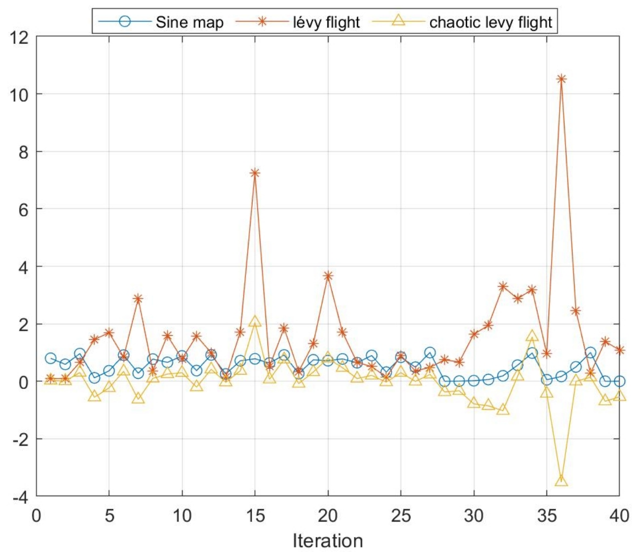


Figure 3.4: The effect of combining sine map with lévy flight

The OBL technique receives the solutions modified using Lévy distribution and chaotic map and selects 50% of them and calculates the opposite solutions. The opposite solution was compared with the current corresponding solution based on the fitness function, and the best of these solutions is selected as the next-generation individual.

Its flowchart is given in Figure 3.5.(b).

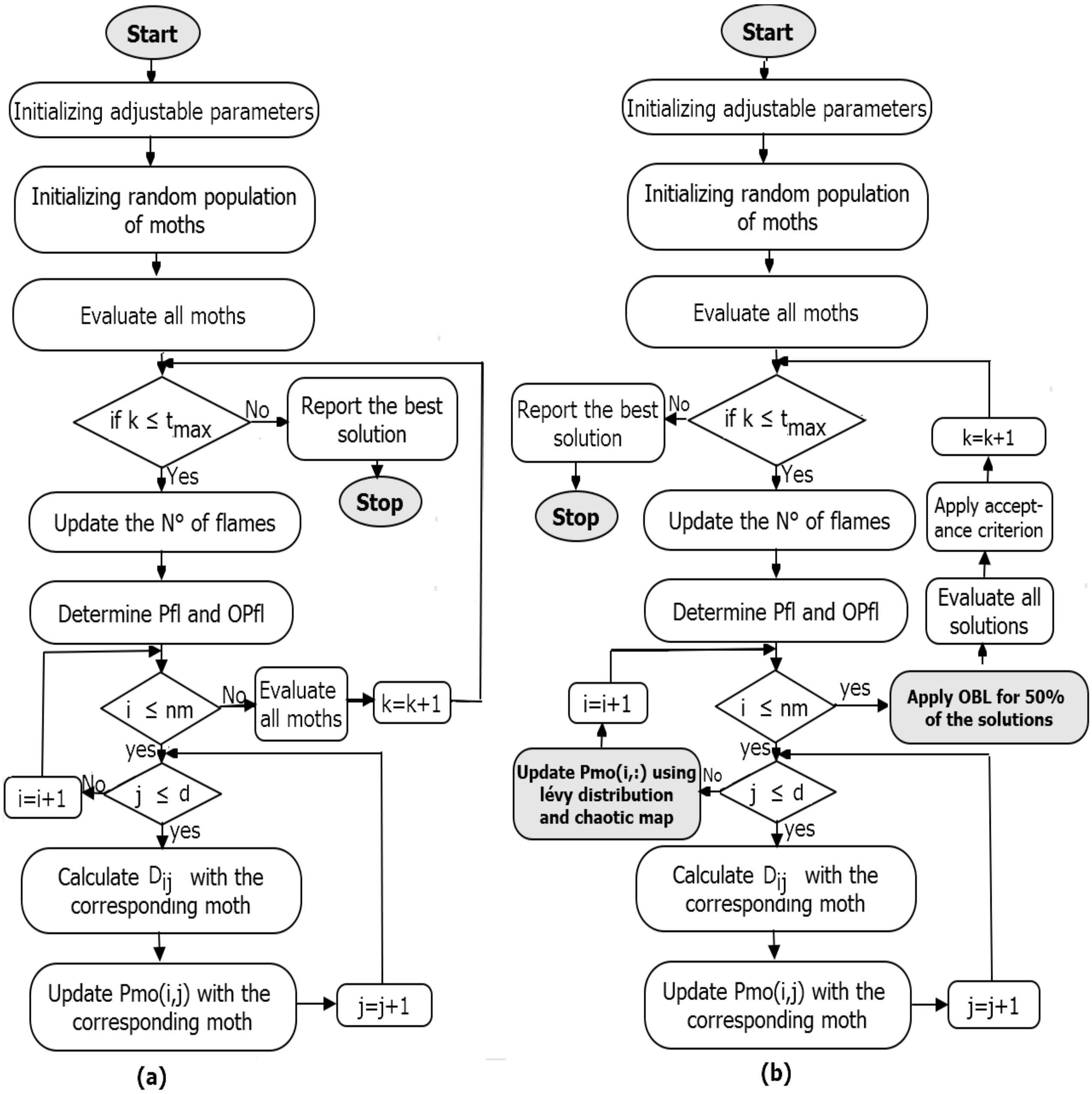


Figure 3.5: The flowchart of: (a) MFO, (b) ECLO-MFO

Algorithm 10 The pseudo-code of the proposed ECLO-MFO for the WMN-MRNP problem

Input:
m: Number of MRs
n: Number of MCs
CR: Coverage radius
nm: Population size (number of moths)
lb: Lower bound
ub: Upper Bound
it_{max}: Maximum number of iterations
Output:
Pmo_{best} the best solution

for j= 1 to m **do**
  Randomly place the *j-th* mesh client in the deployment area
end for
for i= 1 to nm **do**
  Randomly place all mesh routers of the *i-th* solution in the deployment area
  $OPmo(i)$ = g($Pmo(i, 1 : d)$)
end for
for k=1 to it_{max} **do**
  Update number flames (Eq.3.10)
  if $k == 1$ **then**
    $Pfl = sort(Pmo)$
    $OPfl = sort(OPmo)$
  else
    $Pfl = sort(Pmo_{k-1}, Pmo_t)$
    $OPfl = sort(OPmo_{k-1}, OPmo_t)$
  end if
  for i=1 to nm **do**
    for j=1 to n **do**
      Calculate $D_{i,j}$ with respect to its corresponding moth (Eq.3.9)
      Update $Pmo(i, j)$ with respect to its corresponding moth (Eq.3.7)
    end for
    Update $Pmo(i, 1 : d)$ using Lévy distribution and chaotic map (Eq.3.15)
  end for
  for i=1 to nm **do**
    $OPmo(i)$ = g($Pmo(i, 1 : d)$)
    if $mod(i, 2) == 0$ **then**
      Calculate the opposite solution of $OPmo(i)$
      Evaluate the opposite solution of $OPmo(i)$ (Eq.3.5)
      Do the acceptance criterion
    end if
  end for
end for
Return the best solution

The pseudo-code of ECLO-MFO is shown in Algorithm 10. The integration of the three schemes makes MFO able to improve the random initial solutions and converge to the global optimum. In the other hand, the integration of OBL in MFO results in increasing the computation complexity of MFO because the evaluation is done for both the current population and the population produced using the OBL technique. This helps to explain why we applied OBL only for the half of the solutions.

3.5 Simulation results and analysis

In this section, we evaluate and analyze the effectiveness of ECLO-MFO algorithm in solving the mesh router nodes placement problem in WMNs. The evaluation is done considering three metrics such as client coverage Ψ, network connectivity Φ, and fitness value f. Thus, ECLO-MFO is compared with the original MFO and ten well-known meta-heuristics such as GA [114], SA [94], HS [75] , PSO [105], ABC [46], BA [104], CS [45], FA [99], GWO [48], and WOA [50]. All the algorithms are implemented in MATLAB environment version 2020a and executed on a PC with an Intel Core i7-6500U 2.5 GHz-CPU and 8 GB RAM platform running 64-bit windows 10. MRs are deployed in different amounts during simulation to cover 20 to 200 MCs distributed randomly over an area of $4km^2$. The total number of iterations is 2000. The results presented in this section were attained following an average of 30 runs. Table 3.3 describes the common parameters used in simulation.

Table 3.3: Parameters values considered in our simulations

Parameter	**Value**	**Default value**
n	[20 200]	100
m	[5 40]	20
CR	[50 400]	200 m
W	2000	2000 m
H	2000	2000 m
λ	[0 1]	0.5
Population size	50	50
Number of run	50	50
Number of iteration	2000	2000

3.5.1 Effect of varying the λ value

In this scenario, we will evaluate fitness under different λ values (0, 0.25, 0.5, 0.75, 1). Results are summarized in Table 3.4 and Figure 3.6. These findings show that the algorithms under consideration are either not sensitive to changes in the value or only slightly sensitive. Therefore, a value of 0.5 has been used in the remaining simulations to give the same importance to both coverage and connectivity metrics.

Table 3.4: Fitness f for varied λ values

λ	**0**	**0.25**	**0.5**	**0.75**	**1**
ECLO-MFO	0.79	0.76	0.81	0.77	0.85
MFO	0.69	0.72	0.76	0.73	0.86
GA	0.76	0.78	0.73	0.69	0.81
SA	0.71	0.72	0.72	0.72	0.82
HS	0.53	0.61	0.51	0.51	0.60
PSO	0.69	0.64	0.7	0.69	0.76
ABC	0.54	0.54	0.46	0.55	0.61
BA	0.71	0.72	0.72	0.80	0.82
CS	0.74	0.72	0.71	0.73	0.86
FA	0.72	0.68	0.73	0.75	0.83
GWO	0.62	0.57	0.60	0.67	0.75
WOA	0.60	0.59	0.60	0.60	0.68

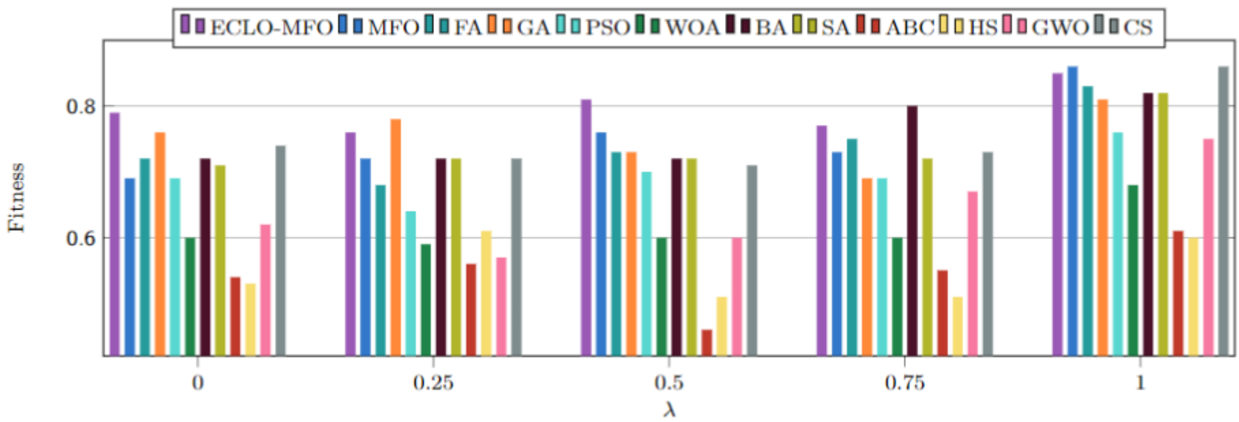


Figure 3.6: Effect of varying λ values on Fitness f.

3.5.2 Effect of varying the number of MCs

In this scenario, we measure the effect of varying the number of MCs on client coverage, network connectivity, and fitness, respectively. The obtained results are shown in Table 5.4 and Figure

Figure 3.7(a) depicts the change in client coverage as the number of MCs changes (from 20 to 200). It is observed that when increasing the number of MCs, the users coverage increases too. It is also shown the effectiveness of our approach in terms of coverage when compared to other algorithms. More precisely, ECLO-MFO covers up to 6.2%, 13%, 31%, 12%, 37%, 41.42%, 9.45%, 8.55%, 17.91%, and 26.32% more users than MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA, respectively.

Figure 3.7(b) shows the effect of rising the number of MCs on network connectivity. It can be noticed that the network connectivity rises as the number of MCs rises. It is also demonstrated that when ECLO-MFO is used, the network connectivity achieves improvements up to 5.8%, 5.9%, 9.32%, 27.8%, 11%, 33.46%, 41.3%, 9.8%, 6.4%, 17.32%, and 26.33% than MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA, respectively.

In terms of fitness, Figure 3.7(c) illustrates that the fitness decreases as the number of MCs rises. In fact, to cover the extra MCs, additional MRs are actually required. However, the number of MRs is fixed. As a result, the newly introduced MCs may not be covered by the provisioned routers, resulting in a reduction in coverage and connectivity metrics that are integrated in the definition of the fitness function. The obtained results revealed the effectiveness and superiority of ECLO-MFO in comparison with MFO, GA, SA, HS, PSO,

Table 3.5: Coverage Ψ, connectivity ϕ, and fitness f for varied number of mesh clients

	n	**20**	**40**	**60**	**80**	**100**	**120**	**140**	**160**	**180**	**200**
Coverage	ECLO-MFO	**19.9**	38.3	55.6	**73.5**	**87.4**	**101.9**	114.8	**135.9**	144.1	162.4
	MFO	19.9	**38.7**	**55.7**	68.5	83.4	96.76	112.8	133	**144.3**	155.1
	GA	17.3	35.2	51.1	64.3	84.4	98.5	**117.9**	126.1	140.5	**163.7**
	SA	19.6	37.5	53.7	67.2	87.3	99.5	114.9	120.7	140.5	161.2
	HS	14.8	27.9	36.9	50.5	65.1	69.5	80.2	91.2	101.5	111.9
	PSO	18.4	33.5	51.3	61.5	77.2	90.9	106.4	121.5	136.7	146.7
	ABC	14.2	24.5	33.1	47.1	56.7	63.9	74.6	86.3	93.1	104.5
	BA	11.8	21.8	31.1	48.2	48.4	58	73.9	80.7	85.8	92.1
	CS	19.2	37.9	51.6	65.9	84.7	94.5	108	120.3	139.4	146.2
	FA	19.6	36	50.5	69	83.7	98.8	109.9	124.7	137.5	153.9
	GWO	17	33.8	50.3	59.1	73.9	86.9	102	117.3	129.1	139.2
	WOA	17.2	28.7	43	52.4	64.7	80.8	84.6	95.4	111.5	121.8
Connectivity	ECLO-MFO	39.5	**57.3**	72.5	**91.7**	**106.9**	**120.5**	134	**155.6**	**163**	180.5
	MFO	**39.9**	56.8	**73.3**	86.1	101.1	116.2	129.8	151.7	161.8	171.1
	GA	37.2	55.2	71.1	84.3	104.4	118.3	**137.5**	146.1	160.5	**183.7**
	SA	35.8	56	70.7	86.4	105.8	117.9	131.4	139.1	159	181.2
	HS	31.6	45.4	51.5	65	80.2	82.7	93.5	105.5	113.8	123.3
	PSO	38.5	53.5	71.3	80.6	97.2	110.9	126.1	141.3	156.3	165.5
	ABC	27.9	37.7	45.7	58.8	67.7	74.2	87.5	95.1	101.6	114.3
	BA	27.6	32.5	39.5	53.7	58.5	67.9	76.6	86.6	90	104.8
	CS	37.5	57.1	70.2	84.8	102.1	110.4	127.7	137.9	157.9	165.4
	FA	38.5	54.1	69	87	103.3	118.1	129	144	157.3	173.7
	GWO	32.6	53.7	69.9	77.3	92.6	106.8	120.9	136.8	147.6	157.7
	WOA	34.8	46.7	57.9	67.7	79.9	78.8	91.2	108.2	131	141.4
Fitness	ECLO-MFO	**0.99**	**0.95**	0.91	**0.91**	**0.88**	**0.85**	0.82	**0.85**	**0.80**	0.81
	MFO	0.99	0.95	**0.92**	0.85	0.83	0.81	0.80	0.83	0.80	0.77
	GA	0.9	0.9	0.87	0.82	0.85	0.83	**0.85**	0.79	0.79	**0.82**
	SA	0.93	0.93	0.88	0.85	0.87	0.83	0.82	0.76	0.78	0.80
	HS	0.76	0.72	0.62	0.64	0.65	0.58	0.57	0.57	0.56	0.55
	PSO	0.94	0.86	0.87	0.78	0.79	0.77	0.77	0.77	0.77	0.74
	ABC	0.70	0.62	0.56	0.58	0.56	0.53	0.53	0.53	0.51	0.52
	BA	0.64	0.54	0.50	0.57	0.48	0.48	0.50	0.49	0.46	0.46
	CS	0.94	0.94	0.86	0.83	0.84	0.78	0.78	0.75	0.78	0.74
	FA	0.96	0.9	0.85	0.86	0.84	0.83	0.79	0.78	0.77	0.77
	GWO	0.83	0.87	0.85	0.75	0.75	0.74	0.74	0.74	0.72	0.7
	WOA	0.86	0.74	0.72	0.66	0.65	0.61	0.58	0.59	0.63	0.62

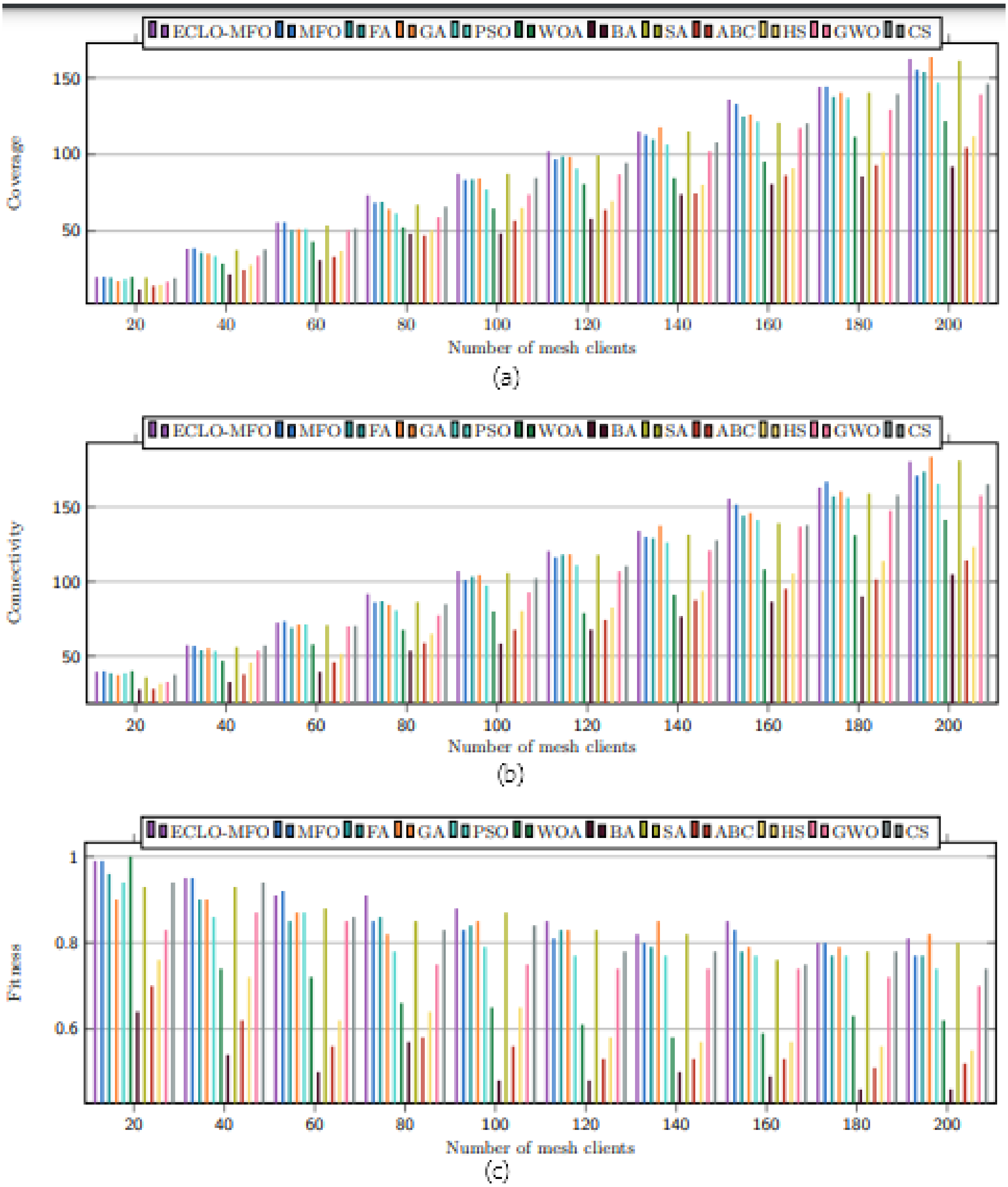


Figure 3.7: Effect of varying the number of MCs on: (a) Coverage Ψ (b) Connectivity Φ (c) Fitness f.

ABC, BA, CS, FA, GWO, and WOA for most of the cases.

3.5.3 Effect of varying the number of MRs

During this scenario, we increased the number of MRs starting from 4 to 40 for covering 100 MCs. Table 3.6 and Figure 3.8 illustrate the effect of varying the number of MRs on coverage, network, and fitness, respectively.

Figure 3.8(a) shows the effect of varying the number of MRs on users coverage. It is demonstrated that as the number of MRs rises, the users coverage rises too. In fact, when adding more MRs in the network, MCs have more chances to be covered by the deployed

Table 3.6: Coverage ψ, connectivity Φ, fitness f under various number of mesh routers

	m	**4**	**8**	**12**	**16**	**20**	**24**	**28**	**32**	**36**	**40**
Coverage	ECLO-MFO	**29**	**51**	**64.1**	**82.4**	**87.4**	**94.6**	**99.5**	**100**	**100**	**100**
	MFO	27.1	46.1	62.4	76.5	78.6	90.4	97	98.2	100	99.9
	GA	23.9	39.5	60.1	71.2	84.4	89.9	95.8	98.4	99.3	99.5
	SA	28.5	43.3	55.7	71.1	84.1	95	98.4	98.73	99.9	100
	HS	25	37.4	45.6	50.8	60.8	66.9	70.5	79.3	80.5	84.1
	PSO	26.4	41.03	54.2	68.2	76.9	85.2	88.2	93.1	98	97.7
	ABC	25	33.4	43.9	46.6	51.5	58.1	68.7	75.2	76.2	81.2
	BA	15.8	25.8	38.4	47.7	48.4	55.4	65.6	73.5	76.9	81.1
	CS	27.6	46.6	56.9	70.3	81.4	89.7	94.6	98	99.8	100
	FA	24.9	41.8	57.3	71.6	80.3	87.4	92.2	96.2	98.2	97.5
	GWO	22.2	39.4	56.9	61	73.9	81.7	89.2	91.2	98.2	97.6
	WOA	23.6	32.3	48.5	58	64.7	70.3	76.4	85	82.5	89.9
Connectivity	ECLO-MFO	**33**	48.8	63.2	**93.1**	**106.9**	118.2	**127.5**	**132**	**136**	**140**
	MFO	22.1	**53.4**	61.5	87.6	94.6	112.8	124.6	130.2	136	139.9
	GA	24.6	47	**70.1**	87.2	104.4	113.9	123.8	130.4	135.3	139.5
	SA	24.9	43.3	60.4	84.9	103.8	**118.9**	126.4	130.5	134.9	140
	HS	26.7	41.5	49.9	60	75.1	87.8	94	107.8	114.9	123.1
	PSO	28	46.8	64.9	83.6	96.6	109.2	115.6	125.1	134	137.7
	ABC	24.5	36	51.4	55	61.9	74.3	89.9	98.5	106.1	121
	BA	15.3	24	35.4	44.4	58.5	69.5	90.8	102.6	101.7	121.1
	CS	31.6	49.1	59.9	85	98.6	113.5	122.4	130	135.8	140
	FA	27.7	46.5	66	86.3	99.7	111	120	128.1	135	139.5
	GWO	24.9	35.8	64.8	76.2	92.6	104.9	117.1	123.2	134.1	137.4
	WOA	25.7	35.1	56.3	72.2	79.9	90.3	102.7	114.6	117.7	128.7
Fitness	ECLO-MFO	**0.30**	**0.48**	0.60	**0.81**	**0.88**	0.94	**0.99**	**1**	**1**	**1**
	MFO	0.24	0.47	0.58	0.76	0.78	0.90	0.97	0.98	1	0.99
	GA	0.23	0.41	**0.61**	0.73	0.85	0.9	0.96	0.98	0.99	0.99
	SA	0.24	0.41	0.54	0.72	0.85	**0.95**	0.98	0.98	0.99	1
	HS	0.25	0.37	0.45	0.51	0.61	0.68	0.71	0.80	0.82	0.86
	PSO	0.26	0.42	0.56	0.7	0.79	0.86	0.89	0.93	0.98	0.98
	ABC	0.24	0.33	0.44	0.46	0.51	0.59	0.69	0.74	0.77	0.83
	BA	0.15	0.23	0.35	0.42	0.48	0.67	0.68	0.75	0.75	0.83
	CS	0.28	0.46	0.54	0.71	0.81	0.9	0.95	0.98	0.99	1
	FA	0.25	0.42	0.58	0.73	0.81	0.88	0.92	0.96	0.99	0.99
	GWO	0.24	0.32	0.49	0.60	0.75	0.83	0.9	0.92	0.98	0.97
	WOA	0.24	0.32	0.49	0.60	0.65	0.71	0.78	0.85	0.84	0.9

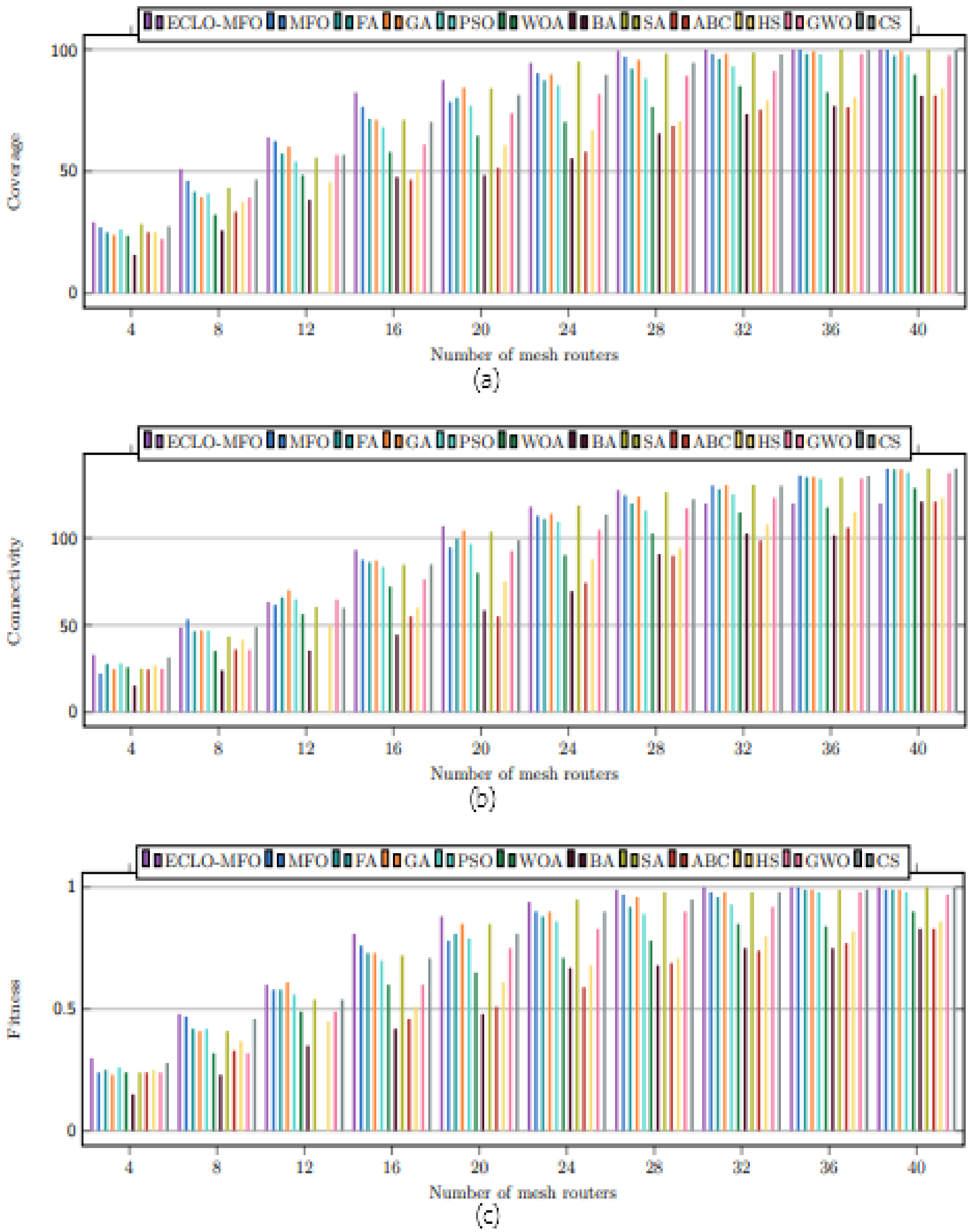


Figure 3.8: Effect of varying the number of MRs on: (a) Coverage ψ (b) Connectivity Φ (c) Fitness f.

MRs, resulting in increasing the users coverage metric. It is also confirmed the efficiency of our approach in terms of coverage for all cases. More specifically, the coverage is expanded by our approach up to 8.76%, 11.43%, 11.23%, 31.56%, 14.2%, 35.86%, 39.16%, 12.06%, 10.73%, 21.36%, and 24.4% when compared to MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA, respectively.

It can be seen from Figure 3.8(b) that the connectivity of the network increases with the number of MRs. In fact, when more routers are added, some sub-networks will be connected forming largest sub-nets, the largest sub-net will be expanded until all mesh nodes are included. It is also demonstrated the effectiveness of ECLO-MFO in forming the biggest sub-net for most of the cases. More precisely, when ECLO-MFO is used, the network connectivity is increased up to 8.76%, 10.5%, 8.1%, 7.8%, 28.53%, 8.1%, 37.55%, 42%, 7%, 5.9%, 14.59%, and 22.5% when compared to MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA, respectively.

Figure 3.8(c) shows that the fitness increases as more routers are added. Again, ECLO-MFO outperforms when compared to MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA for most of the cases. It requires less routers to achieve full coverage and full connectivity when compared with other algorithms.

3.5.4 Effect of varying coverage radius values

In this scenario, we varied the coverage values from 50 to 500 for covering 100 MCs, using 20 MRs. Table 3.7 and 3.9 illustrate coverage, connectivity, and fitness for a varied coverage radius values.

Figure 3.9(a) shows the impact of expanding the router coverage radius on the users' coverage. It can be seen that the users coverage increases while increasing the router coverage radius. In fact, when the coverage radius is increased, the mesh routers are better equipped to cover vast area until including nearly all MCs (when the coverage radius is greater than 300 m for most algorithms). It is also shown the efficiency of our approach in terms of users coverage for most of the cases. More precisely, when the coverage radius exceeds 100, ECLO-MFO covers up to 7.34%, 12.04%, 9.06%, 27.16%, 27.16%, 24.97%, 35.07%, 34.2%, 13%, 9.63%, and 22.6% more users than MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA, respectively. Results presented in Figure 3.9(b) display the effect of increasing the MR coverage radius on network connectivity. It can be seen that the network connectivity increases proportionally with MR coverage radius. In fact, each mesh router may cover more mesh clients and connect to other MRs when the radius of coverage is increased. As a result, the largest sub-net will grow in size until connecting all of the mesh nodes. When the coverage radius is more than 150 m, ECLO-MFO outperforms other algorithms. More specifically, it increases the network connectivity up to 9.4%, 1.9%, 9.7%, 26.14%, 9.8%, 32.66%, 36.27%, 7%, 8.66%, 10.77%, and 22.77% than MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA, respectively.

The effect of increasing the router coverage radius on fitness is described in Figure 3.9(c).

It is demonstrated that the fitness is inversely correlated with covering radius. When the

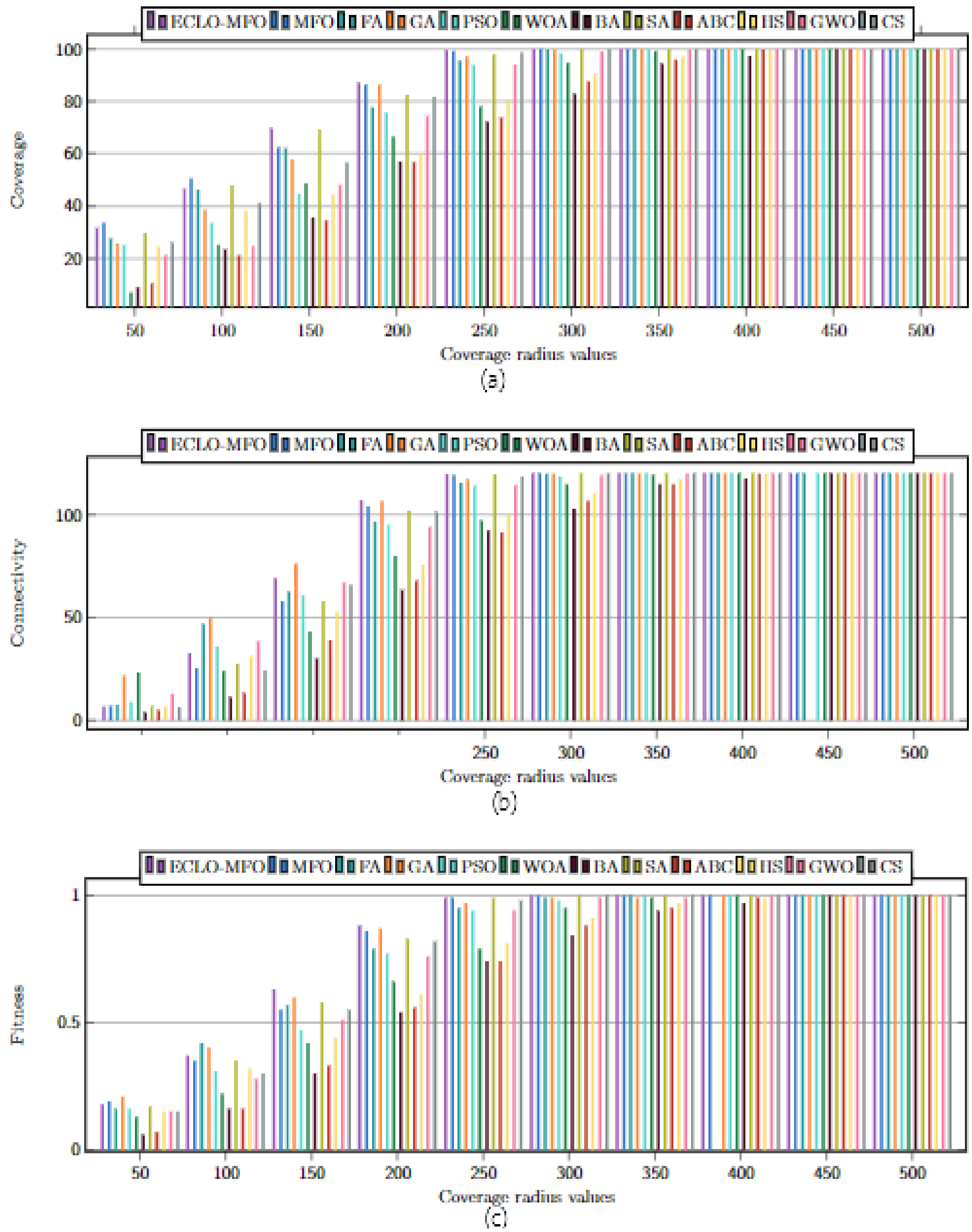


Figure 3.9: Effect of varying coverage radius values on: (a) Coverage Ψ (b) Connectivity Φ (c) Fitness f.

Table 3.7: Coverage Ψ, connectivity Φ, and fitness f for varied coverage radius values

	CR	**50**	**100**	**150**	**200**	**250**	**300**	**350**	**400**	**450**	**500**
Coverage	ECLO-MFO	31.4	47.8	**69.7**	**87.4**	**99.6**	**100**	**100**	**100**	**100**	**100**
	MFO	**33.6**	**50.6**	62.4	86.4	99.2	100	100	100	100	100
	GA	25.5	38.6	57.7	86.5	97.2	99.8	99.9	100	100	100
	SA	29.7	47.6	69.1	82.3	98	100	100	100	100	100
	HS	24.5	38.3	44.3	60.2	80.5	90.8	97	99.9	100	100
	PSO	25.1	33.4	44.7	75.7	93.9	98.5	100	100	100	100
	ABC	10.4	21.2	34.6	56.7	73.9	87.7	95.9	99.8	100	100
	BA	9	23.5	35.5	57.1	72.3	82.9	94.5	97.4	100	100
	CS	26.1	41.2	56.6	81.8	98.6	100	100	100	100	100
	FA	27.7	46.3	62.2	77.7	95.5	99.8	100	100	100	100
	GWO	21.2	24.6	48.1	74.4	94	99	99.9	100	100	100
	WOA	6.9	25.2	48.6	66.4	78	94.6	99.3	100	100	100
Connectivity	ECLO-MFO	6.7	32.83	69.1	**106.9**	**119.5**	**120**	**120**	**120**	**120**	**120**
	MFO	7.1	25.4	57.8	103.9	119.1	120	120	120	120	120
	GA	21.9	**49.9**	**76.1**	106.5	117.2	119.8	119.9	120	120	120
	SA	7.1	27.4	57.5	101.5	118	120	120	120	120	120
	HS	6.7	31.3	52.9	75.6	99.9	110.5	117	119.9	120	120
	PSO	8.9	35.9	60.7	95.1	113.9	118.5	120	120	120	120
	ABC	5	13.4	39	67.7	91.3	106.6	114.5	119.9	120	120
	BA	3.8	11.2	30	63.4	92.2	102.9	114.5	117.4	120	120
	CS	6.6	24.3	65.7	101.3	118.6	120	120	120	120	120
	FA	7.5	46.9	62.5	96.5	115.5	119.8	120	120	120	120
	GWO	12.8	38.6	67	94	114	119	119.9	120	120	120
	WOA	**23.4**	24.1	43.3	79.6	97.1	114.6	119.3	120	120	120
Fitness	ECLO-MFO	0.18	0.37	**0.63**	**0.88**	**0.99**	**1**	**1**	**1**	**1**	**1**
	MFO	0.19	0.35	0.55	0.86	0.99	1	1	1	1	1
	GA	**0.21**	**0.40**	0.60	0.87	0.97	0.99	0.99	1	1	1
	SA	0.17	0.35	0.58	0.83	0.98	1	1	1	1	1
	HS	0.15	0.32	0.44	0.61	0.81	0.91	0.97	0.99	1	1
	PSO	0.16	0.31	0.47	0.77	0.94	0.98	1	1	1	1
	ABC	0.07	0.16	0.33	0.56	0.74	0.88	0.95	0.99	1	1
	BA	0.06	0.16	0.30	0.54	0.74	0.84	0.94	0.97	1	1
	CS	0.15	0.30	0.55	0.82	0.98	1	1	1	1	1
	FA	0.16	0.42	0.57	0.79	0.95	0.99	1	1	1	1
	GWO	0.15	0.28	0.51	0.76	0.94	0.99	0.99	1	1	1
	WOA	0.13	0.22	0.42	0.66	0.79	0.95	0.99	1	1	1

coverage radius value of every MR is increased, mesh routers have more capability of covering more MCs and connecting to other MRs, resulting in increasing the metrics of coverage and connectivity that are involved in fitness. Again, when the router coverage radius is more than 100m, ECLO-MFO outperforms better than MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA.

3.5.5 Convergence Analysis

In this section, we analyze the convergence of ECLO-MFO compared to other algorithms. Table 3.9 shows the convergence analysis of ECLO-MFO, MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA algorithms. Convergence analysis process is done using five network instances of different sizes (i.e. (a) $Instance_1$, (b) $Instance_2$, (c) $Instance_3$, (d) $Instance_4$, and (e) $Instance_5$) as illustrated in Table 3.8. The convergence analysis process is based on convergence efficiency (fitness value) and convergence speed. An average of 30 experiments are performed for each result.

Table 3.8: Network instances taken into account in the convergence analysis

Instance	WxH	n	m	CR
$Instance_1$	$1000m \times 1000m$	5	50	$200m$
$Instance_2$	$1500m \times 1500m$	10	100	$200m$
$Instance_3$	$2000m \times 2000m$	15	150	$200m$
$Instance_4$	$2500m \times 2500m$	20	200	$200m$
$Instance_5$	$3000m \times 3000m$	25	250	$200m$

Figures 3.10a, 3.10b, 3.10c, 3.10d, and 3.10e report an example of the algorithms convergence using $Instance_1$,$Instance_2$, $Instance_3$, $Instance_4$, and $Instance_5$.

Based on the fitness values of ECLO-MFO in comparison with other algorithms as illustrated in Table 3.9, it can be observed that ECLO-MFO gives better results for four instances in both large and small networks. In terms of convergence speed, the results described in Table 3.9 and Figures 3.10a, 3.10b, 3.10c, 3.10d, and 3.10e showed that ECLO-MFO is among algorithms that needs more iterations to achieve the global optimum. However, the fitness reached by ECLO-MFO is the best for most of the cases. The reasons behind this improvement are given below:

Table 3.9: Convergence analysis between ECLO-MFO, MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA algorithms

Instance	$Instance_1$		$Instance_2$		$Instance_3$		$Instance_4$		$Instance_5$	
	Fitness	Iteration	Fitness	Iteration	Fitness	Iteration	Fitness	Iteration	Fitness	Iteration
ECLO-MFO	**0.89**	638	**0.81**	1185	**0.66**	1670	**0.60**	1694	0.53	1727
MFO	0.84	399	0.75	964	0.61	1197	0.57	1616	0.44	1603
GA	0.85	539	0.70	1044	0.65	1285	0.59	1466	**0.54**	1538
SA	0.75	1142	0.81	1484	0.65	1782	0.58	1862	0.49	1915
HS	0.81	1576	0.59	1450	0.44	1192	0.41	1604	0.33	1676
PSO	0.74	142	0.74	528	0.66	737	0.55	758	0.48	1043
ABC	0.79	713	0.58	845	0.44	921	0.37	972	0.27	943
BA	0.62	30	0.48	349	0.44	269	0.32	446	0.26	383
CS	0.85	1235	0.74	1606	0.66	1771	0.53	1826	0.47	1839
FA	0.79	11	0.76	27	0.66	47	0.57	65	0.52	80
GWO	0.88	1144	0.67	1970	0.61	1903	0.47	1834	0.42	1898
WOA	0.82	371	0.62	518	0.54	408	0.42	618	0.36	71

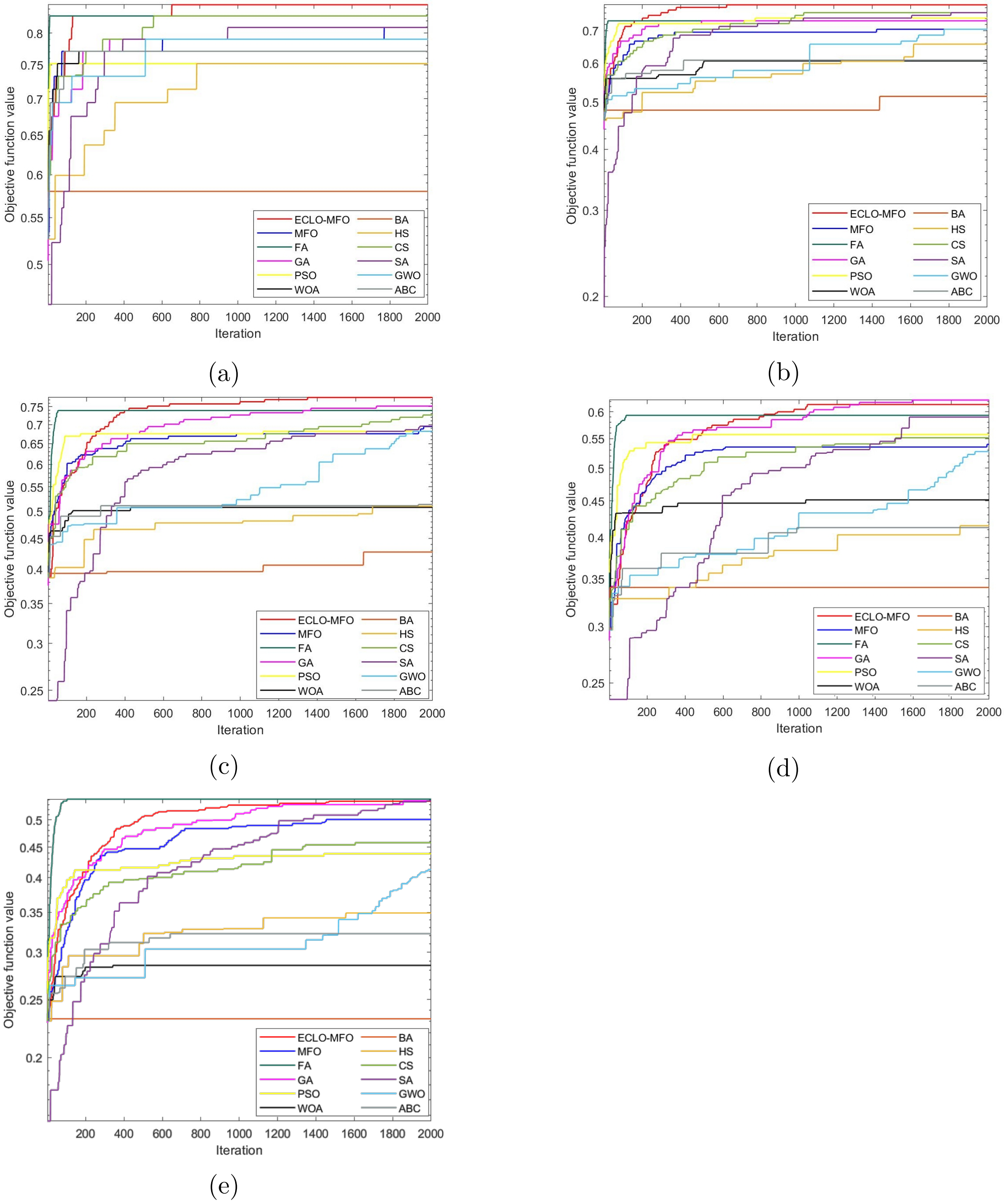


Figure 3.10: Example of algorithms convergence using: (a) $Instance_1$ (b) $Instance_2$ (c) $Instance_3$ (d) $Instance_4$ (e) $Instance_5$

- The adaptive scheme employed in MFO for updating the number of flames leads to a good balance between intensification and diversification.
- The integration of OBL improves the population diversity in the search process.
- The Lévy flight mechanism is helpful for avoiding to trap in a local optimum.
- The integration of sine map with Lévy flight mechanism results in reducing the walk of Lévy distribution. Consequently, better solutions are achieved.

3.6 Conclusion

In this chapter, we have proposed an enhanced version of MFO, called ECLO-MFO to tackle the mesh router nodes placement problem in WMNs. ECLO-MFO is based on the integration of three strategies including: the chaotic map concept, the Lévy flight strategy (LFD), and the Opposition-Based Learning (OBL) technique to enhance the optimization performance of MFO. Its performance was analyzed and evaluated by investigating the impact of varying the number of mesh clients, the number of mesh routers, and coverage radius values. Obtained results revealed the superiority and the effectiveness of ECLO-MFO when compared to other optimization algorithms such as MFO, GA, SA, HS, PSO, ABC, BA, CS, FA, GWO, and WOA in terms of network connectivity and user coverage. In the next chapter, we will propose a hybrid approach for solving the mesh routers placement problem with service priority.

CHAPTER 4

A HYBRID APPROACH FOR THE MESH ROUTERS PLACEMENT PROBLEM WITH SERVICE PRIORITY

4.1 Introduction

This chapter aims to present a hybrid approach, called ASO-SA, based on the combination of an Adaptive Snake Optimizer (ASO) with Simulated Annealing (SA) for solving the mesh routers placement problem with service priority. ASO is based on incorporation of the Generalized Opposition Based-Learning (GOBL) mechanism in the exploration phase of SO for enhancing its performance. ASO-SA combines the global search capability of ASO and the local search capability of SA.

In this chapter, we present a reminder of the hybrid approaches proposed in the literature to solve the mesh nodes placement problem in WMNs, we then describe the formulation of the mesh routers placement problem with service priority, we also present Snake Optimizer algorithm, Simulated Annealing algorithm, Generalized Opposition Based-Learning, and the hybrid approach. Finally, we evaluate the performance of the proposed approach and compare its characteristics with SO-SA, ASO, SO, and SA algorithms.

4.2 Related works

Various hybrid approaches were proposed in the literature to solve the mesh routers placement problem in WMNs. Sakamoto et al. [115, 116] suggested a hybrid approach, called PSO-HC, based on the combination of PSO with Hill-Climbing algorithm, for solving the mesh routers placement problem in WMNs. User coverage, network connectivity, and convergence were considered as metrics to be optimized and results demonstrated the effectiveness of PSOHC in finding the optimal placement of mesh routers. Barolli et al. [117] suggested a hybrid algorithm, called PSO-DGA, based of the hybridization of PSO with distributed

GA (DGA), for addressing the mesh routers placement problem in WMNs. PSO-DGA algorithm was validated under 4 factors including the number of router nodes, number of mesh client nodes, coverage radius, and size of the deployment area. Results of simulated demonstrated the good performance of PSO-DGA. Sakamoto et al [118] suggested a hybrid algorithm, called PSO-HC-DGA, based on hybridizing PSO algorithm with Hill-Climbing and DGA, for solving the mesh routers placement problem in WMNs. PSO-HC-DGA was validated in terms of user coverage and network connectivity metrics. Results of simulation demonstrated the effectiveness of PSO-HC-DGA when compared with PSO-DGA model. A hybrid approach called PSO-SA, based on the combination of PSO and SA, was suggested in the work of Sakamoto et al. [119] for solving the mesh routers placement problem in WMNs. Four replacement methods which are: Constriction Method (CM), Random Inertia Weight Method (RIWM), Linear Decreasing Inertia Weight Method (LDIWM), and Rational of Decrement of Vmax Method (RDVM) were considered to evaluate the performance of PSO-SA. Simulation results showed that PSO-SA converges faster and has better performance using LDIWM and RDVM methods. Taleb et al. [120] applied a hybrid algorithm, called HFPSO, based on the combination of PSO and FA, for solving the mesh routers placement problem in WMNs. HFPSO was validated in terms of users coverage and network connectivity, taking into account the effect of varying the number of mesh routers, the number of mesh clients, and coverage radius values. Simulations results proved the good performance of HFPSO when compared with PSO and FA algorithms. Barolli et. al.[121] suggested a hybrid algorithm, called PSOSA-DGA, based on the combination PSO, SA, and DGA algorithms, for solving the WMN router nodes placement problem. PSO-SA-DGA was validated considering chi-square distribution of mesh clients and two router replacement methods (CM and RIWM). Simulation results showed that PSO-SA-DGA has better performance for CM compared with the case of RIWM. Table 4.1 summarizes some representative works using hybrid approaches for solving the mesh routers placement problem in WMNs.

Table 4.1: Summary of hybrid approaches employed to solve the mesh routers placement problem in WMNs

Algorithms	References	Environment	Location	Metrics		
				Coverage	Connectivity	Load balancing
PSO-HC	Sakamoto et al. [115, 116]	Static	Discrete	x	x	
PSO-HC-DGA	Sakamoto et al [118]	Static	Discrete	x	x	
PSO-SA	Sakamoto et al.[119]	Static	Discrete	x	x	
PSO-SA-DGA	Barolli et al.[121]	Static	Discrete	x	x	
PSO-DGA	Barolli et al.[117]	Static	Discrete	x	x	x
HFPSO	Taleb et al. [120]	Static	Continuous	x	x	

4.3 The mesh routers placement problem formulation

A fully connected graph $G = (V, L)$ can be used to mathematically represent WMN, where V represents the set of MRs and L represents the set of links connecting these MRs. Two sets of mesh nodes where considered as described bellow:

- MRS is the set of n of MRs: $MRS = \{mr_1, mr_2.....mr_m\}$. Each mesh router mr_i is equipped with one radio interface having transmission range CR_i. A connection between two mesh routers mr_i and mr_j is possible only if the distance between them does not exceed the sum of their transmission ranges. Each installed MR must covers at least one MC.
- MCS is the set of m MCs: $MCS = \{mc_1, mc_2.....mc_n\}$. MCs are randomly dispersed in the deployment area. We assume that MCs are not equal, a priority is associated with each mesh client mc_j. The priority of the j^{th} client is denoted by wc_j. A mesh client mc_j is said covered by a mesh router mr_i if it is within the transmission range of this router. It can be linked at most with one router. It can be within the transmission range of several MRs but it is associated with the nearest one.

The coverage variable a_{ij} and given as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } \mathrm{mc_i} \text{ is covered by } \mathrm{mr_j} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

Let j and $l \in MRS$, $j \neq l$, the connectivity variable A_{jl} (adjacency matrix) is specified as follows:

$$A_{jl} = \begin{cases} 1 & \text{if } \mathrm{mr_j} \text{ and } \mathrm{mc_l} \text{ can be connected} \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

Our work's main goal is to determine where n MRs should be placed in a $2D$ space with dimensions $W \times L$, taking into account the positions of m mesh clients and their priorities, maximizing the coverage metric. The problem can be described mathematically as follows:

$$f = \text{Max}\, \frac{\sum_{i=1}^{n}(max_{j\in\{1,...m\}a_{ij}}) * wc_i}{\sum_{i=1}^{n} wc_i} \tag{4.3}$$

Subject to:

$$\sum_{j\in MRS} x_{ij} <= 1 \quad \forall i \in MCS \tag{4.4}$$

$$\sum_{k=1}^{n-1} \mathrm{A}^k \neq 0 \tag{4.5}$$

The objective function in (4.3) maximizes the weighted coverage. Equation 4.4 ensures that each MC can be assigned to at most one mesh router. The equation 4.5 imposes to have at least one path between each pair of MRs. Consequently, the graph G is fully connected.

4.4 Preliminaries

4.4.1 Snake Optimizer

Snake Optimizer (SO) is a swarm intelligence meta-heuristic proposed by Hashim and Hussein [55] in 2022. The main concept of SO is based on the mating behavior of snakes when the temperature is low and food is available. The main phases of SO are explained as following:

4.4.1.1 Initialization

SO process begins by initializing adjusting parameters including:

- ns: The number of snakes in the population.
- d: The search space dimension.
- LB, UB: Lower and upper boundaries of the search space.
- $itmax$: The total number of iterations.
- $Threshold1$, $Threshold2$, c_1, c_2, c_3: Constants equal to 0.25, 0.6, 0.5, 0.05, and 2, respectively.

The population of ns snakes denoted by Pso is initialized, Pso is presented by the following 2D matrix:

$$Pso = \begin{bmatrix} Pso_{1,1} & Pso_{1,2} & ... & ... & Pso_{1,d} \\ Pso_{2,1} & Pso_{2,2} & ... & ... & Pso_{2,d} \\ . & . & . & . & . \\ . & . & . & . & . \\ Pso_{n,1} & Pso_{n,2} & ... & ... & Pso_{ns,d} \end{bmatrix}$$

4.4.1.2 Swarm dividing

Using Equation 4.6, SO splites the swarm into equal male and female groups

$$\begin{gathered} ns_{\mathrm{m}} \approx \frac{ns}{2} \\ ns_f = ns - ns_m \end{gathered} \tag{4.6}$$

where ns_m represents the number of male individuals and ns_f represents the number of female individuals.

4.4.1.3 Definition of the temperature and food quality

For each iteration t, the temperature ($Temp$) and food quantity (FQ) are defined according to Equations 4.7 and 4.8, respectively.

$$Temp = \exp\left(\frac{-t}{itmax}\right) \tag{4.7}$$

$$FQ = c_1 \exp\left(\frac{t - itmax}{itmax}\right) \tag{4.8}$$

4.4.1.4 Exploration (no food)

The exploration phase is simulated when $FQ < Threshold1$. The exploration behavior of males and females can be described mathematically as follows:

- For male snakes

  The position of the new i^{th} male snake at $t+1$ is represented by $new - Psom_i$ and represented as follows:

  $$new - Psom_i(t+1) = Psom_{rand}(t) \pm c_2 \times Am\left((UB - LB) \times \mathrm{r1} + LB\right) \tag{4.9}$$

  where $Psom_{rand}$ is the position of a random male snake, $r1$ is a random number in range [0 1], $fitnessm_{rand}$ is the fitness of the earlier selected random male snake, $fitnessm_i$ is the fitness of $i^t h$ male in the group, and Am is the ability to find the food by the male, calculated as illustrated in Equation 4.10.

  $$Am = \exp\left(\frac{-fitnessm_{rand}}{fitnessm_i}\right) \tag{4.10}$$

- For female snakes

  The position of the new i^{th} female snake at $t+1$ is represented by $new - Psof_i$ and determined as follows:

  $$new - Psof_i(t+1) = Psof_{rand}(t) \pm c_2 \times Af\left((UB - LB) \times \mathrm{r2} + LB\right) \tag{4.11}$$

  where $Psof_{rand}$ is the position of a random female snake, $r2$ is a random number in range [0 1], $fitnessf_{rand}$ is the fitness of the earlier selected random female snake, $fitnessf_i$ is the fitness of $i^t h$ male in the group, and Af is the ability to find the food by the female, determined as illustrated in Equation 4.12.

  $$Af = \exp\left(\frac{-fitnessf_{rand}}{fitnessf_i}\right) \tag{4.12}$$

4.4.1.5 Exploitation

Based on quality of solution FQ and temperature $Temp$ conditions, the exploitation procedure is simulated as follows:

- If ($FQ > Threshold1$) and ($Temp > Threshold2$), the snakes move to find the food. At $t+1$ iteration, the positions of the i^{th} male snake and i^th female snake are determined as illustrated in Equations 4.13 and 4.14.

$$new - Psom_i(t+1) = Psom_{\text{best}} \pm c_3 \times Temp \times \text{ r3 } \times (Pso_{\text{best}} - Psom_i(t)) \quad (4.13)$$

$$new - Psof_i(t+1) = Psof_{\text{best}} \pm c_3 \times Temp \times \text{ r4 } \times (Pso_{\text{best}} - Psof_i(t))$$

(4.14)

  where $r3$ and $r4$ are a random numbers in range [0].

- If ($FQ > Threshold1$) and ($Temp < Threshold2$), the snakes will then be in one of two modes: fighting or mating. If a random number $r5$ in range [0 1] is less than 0.6, the fighting model is simulated, otherwise, the mating model is simulated. The fighting and mating models are illustrated as follows:

  - Fighting mode
    The position of the new i^{th} male is updated as described in Equation 4.15:

    $$new - Psom_i(t+1) = Psom_i(t) \pm c_3 \times FM \times \text{ r6} \times (Psof_{best} - Psom_i(t)) \quad (4.15)$$

    Where $Psom_{best}$ refers to the best male snake $r6$ is random number in range [0 1] and FM is fighting ability of male snake. Let calculated $fitnessf_{best}$ is the fitness value of the best female snake, FM is calculated as specified in Equation 4.16.

    $$FM = \exp\left(\frac{-fitnessf_{best}}{fitnessm_i}\right) \quad (4.16)$$

    The position of the new i^{th} female snake is updated as described in Equation 4.17:

    $$new - Psof_i(t+1) = Psof_i(t) \pm c_3 \times FF \times \text{ r7} \times (Psom_{best} - Psof_i(t)) \quad (4.17)$$

    where $r7$ is random number in range [0 1] and FF is fighting ability of female snake. Let $fitnessm_{best}$ represents the fitness value of the best male snake, FF is determined as specified in Equation 4.18.

$$FF = \exp\left(\frac{-fitnessm_{\text{best}}}{fitnessf_i}\right) \tag{4.18}$$

- Mating mode

  In this mode, the position of the i^{th} male is determined as follows:

$$Pso_m i(t+1) = Psom_i(t) \pm c_3 \times Mm \times \mathrm{r8} \times (Q \times Psof_i - Psom_i(t)) \tag{4.19}$$

where $r8$ is random number in range [0 1] and Mm refers to the mating ability of male snake, calculated as described in Equation 4.20.

$$Mm = \exp\left(\frac{-fitnessf_i}{fitnessm_i}\right) \tag{4.20}$$

In other hand, the position of the new i^{th} female snake is given as follows:

$$new - Psof_i(t+1) = Psof_t(t) \pm c_3 \times Mf \times \ \mathrm{r9}\ \times (Q \times Psom_i - Psom_i(t+1)) \tag{4.21}$$

where $r9$ is random number in range [0 1] and Mm refers to the mating ability of female snake, calculated as described in Equation 4.22.

$$Mf = \exp\left(\frac{-fitnessm_i}{fitnessf_i}\right) \tag{4.22}$$

4.4.1.6 Evaluate each group and do the new snake acceptance criteria

The *i-th* new male snake (or female snake) is accepted in the male group (or female group) if its fitness value better than the *i-th* male snake (or female snake).

By examining each group for individual best male $Psom_{best}$ and best female $Psof_{best}$, the best individual candidate solution Pso_{best} is identified for each iteration.

The pseudo-code of SO is illustrated in Algorithm 11

Algorithm 11 The pseudo-code of SO algorithm

```
Initialize the adjustable parameters of SO
Generate the initial positions of SO
Divide the population in male and female groups using Equation 4.6
Evaluate male and female groups and identify Psom_best, Psof_best, Pso_food

while t < itmax do
    Define the temperature and food quality according to Equations 4.7 and 4.8
    if FQ < Threshold1 then
        for i ← 1 ns_m do
            Update Psom_{t+1} using Equation 4.9
        end for
        for i ← 1 ns_f do
            Update Psof_{t+1} using Equation 4.11
        end for
        if Temp > Threshold2 then
            for i ← 1 ns_m do
                Update Psom_{t+1} using Equation 4.13
            end for
            for i ← 1 ns_f do
                Update Psof_{t+1} using Equation 4.14
            end for
        else
            if r5 < 0.6 then
                for i ← 1 ns_m do
                    Update Psom_{t+1} using Equation 4.15
                end for
                for i ← 1 ns_f do
                    Update Psof_{t+1} using Equation 4.17
                end for
            else
                for i ← 1 ns_m do
                    Update Psom_{t+1} using Equation 4.19
                end for
                for i ← 1 ns_f do
                    Update Psof_{t+1} using Equation 4.21
                end for
            end if
        end if
    end if
    evaluate male and female snake groups and identify Psom_best, Psof_best, and Pso_best
t = t + 1

end while
Return the best solution Pso_best
```

4.4.2 Generalized opposition based-learning

The opposition-based learning (OBL) technique, proposed by Tizhoosh in 2005, is a well-respected machine and computational intelligence scheme. It has been successfully integrated into many meta-heuristic optimization methods to improve their convergence speed and effectively explore the search space. OBL strategy is based on opposite numbers to approach the solutions. The opposite of real number $S \in [lb, ub]$ can be mathematically defined by the following equation:

$$S_{obl} = ub + lb - S \tag{11}$$

The following definition can be used for a multidimensional search space:

$$S_{i\_obl} = ub_i + lb_i - S_i, \quad i = 1, 2, \ldots, d \tag{12}$$

where $S_{obl} \in R^d$ is the opposite vector from the real vector $S \in R^d$

Wang et al. [122] introduced an extended version of the OBL strategy called Generalized OBL (GOBL). GOBL was integrated into many evolutionary algorithms [123, 124] and revealed its efficiency when compared with the OBL strategy. S_{gobl} is the corresponding generalized opposition-based learning of Pmo, it is calculated as follows:

$$S_{gobl} = k * (ub + lb) - S \tag{13}$$

This definition can be extended to multidimensional search space as follows:

$$S_{i\_gobl} = k * (ub_i + lb_i) - S_i, \quad i = 1, 2, \ldots, d \tag{14}$$

where k is a random number in the range of [0 1].

4.4.3 Simulated Annealing (SA)

In 1983, Kirkpatrick et al. [31] developed the single-based meta-heuristic known as Simulated Annealing (SA). The main concept of SA is based on the annealing theory which simulates the cooling process of metal atoms. Numerous optimization problems, such as the issue of node placement [32, 33, 34], have been addressed using SA.

SA begins with an initial solution X and Temperature Tmp. For each iteration t in $[1tmax]$, SA searches for X' the neighbor of the current solution X. Only two scenarios are acceptable for the solution X': Firstly, if $\delta \leq 0$, where $\delta = f(X') - f(X)$, $f(X')$ and $f(X)$ are fitness values of the neighbor and current solution, respectively. Secondly, if $\delta > 0$ and the Boltzmann probability $P = e^{\Delta/Tmp}$ is greater then a random value r. The temperature Tmp drops with a cooling factor Cf at the end of the iteration. Up until the maximum number of iterations is reached, this process is repeated. Algorithm 12 illustrates the SA algorithm's pseudo-code.

Algorithm 12 The pseudo-code of Simulated Annealing algorithm

Initialize SA parameters: Initial Temperature Tmp_0, cooling factor Cf, and number of the neighborhoods ns in the search space.
Generate initial solution X
while $(t < tmax)$ **do**
  while $(it < n)$ **do**
    Generate X' using equation
    Calculate $\Delta = f(X') - f(X)$
    Generate a random uniform variable r
    if $(\Delta < 0)$ **then**
      Determine X' the neighbor of X
    else
      if $(\exp^{-\Delta/Tmp} > r)$ **then**
        $X = X'$
      end if
    end if
    $it = it + 1$
  end while
  $t = t + 1$
end while
Return The best solution X_{best}

4.5 Hybrid Adaptive Snake Optimizer with Simulated Annealing (ASO-SA) for the mesh routers placement problem

SO is a new swarm intelligence meta-heuristic proposed by Hashim and Hussein [55] for solving various tasks of optimization. The performance of SO was assessed under 29 unconstrained Congress on Evolutionary Computation (CEC) 2017 benchmark functions and four constrained real-world engineering problems. SO is compared with other 9 well-known and newly developed algorithms such as Linear population size reduction-Success-History Adaptation for Differential Evolution (L-SHADE), Ensemble Sinusoidal incorporated with L-SHADE (LSHADE-EpSin), Covariance matrix adaptation evolution strategy (CMAES), Coyote Optimization Algorithm (COA), Moth-flame Optimization, Harris Hawks Optimizer, Thermal Exchange optimization, Grasshopper Optimization Algorithm, and Whale Optimization Algorithm. Simulation results demonstrated the effectiveness of SO with respect exploration-exploitation balance and convergence curve speed. In other hand, like most algorithms, SO may stuck in local optima and is unable to identify the global optimum solution. To overcome these drawbacks, it is advised that researchers adapt it and integrate it with other strategies or meta-heuristics to improve the exploitation and exploration phases. In this chapter, SO is improved as follows:

- The Generalized Opposition Based Learning (GOBL) was integrated in the original SO to improve its performance in terms of exploration (if $FQ < Threshold1$). More precisely, if a male snake or female snake has no improvement in its fitness value in the last iteration, GOBL was used to update the new position of the male snake or the female snake, otherwise, the standard exploration strategy is used. SO based only on the integration of GOBL mechanism is denoted by ASO.
- SA algorithm was used to improve the exploitation phase (if $FQ > Threshold1$). In fact, if a male snake or female snake has an improvement in its fitness value in the last iteration, SA was used to update the new position of the male snake or the female snake, otherwise, the exploitation strategy of SO is used. The hybridization of SO with SA is denoted by SO-SA.

The proposed Hybrid approach based on the integration of GOBl and SA into the original SO is denoted by ASO-SA.

4.6 Simulation results

Table 4.2: Parameters values considered in our simulations

Parameter	Value	Default value
n	[20 160]	140
m	[5 40]	20
CR	[100m 500m]	200m
W	[1000m 2500m]	2000m
H	[1000 2500m]	2000m
Population size	30	30
Number of runs	15	15
Number of iterations	2000	2000

In this section, we will evaluate the performance of ASO-SA hybrid approach in comparison with SO-SA, ASO, SO [55], and SA [94] algorithms. All of the algorithms were coded in Matlab. A Core i7 2.5 GHz computer is used to run the simulations. The total number of iterations is set to 2000. All simulations took into account a population of 30 solutions. The results displayed here are an average of 15 trials. Tables 4.2 and 4.3 summarize the common parameters used in simulation and algorithms parameters, respectively.

The performance of ASO-SA approach was analysed in terms of fitness value f considering the effect of varying three key parameters such as: (1) number of mesh routers (2) number of

Table 4.3: Algorithms parameters

Parameter	**Value**
ASO-SA \ASO \SO	
Constants $Threshold1$	0.25
Constant $Threshold2$	0.6
Constant c_1	0.5
Constant c_2	0.05
Constant c_3	2
SA	
Initial temperature T_i	1
Final temperature T_f	0

mesh clients (3) coverage radius values. A weighted mesh clients were dispersed randomly in the deployment area. Weights are integer numbers in range [1 5]. Initial solutions are chosen from a set of randomly generated solution, taking into account the connectivity requirement. So, to simplify the generation process and get initial solutions in reasonable time execution, mesh routers are positioned in square of $500m \times 500m$.

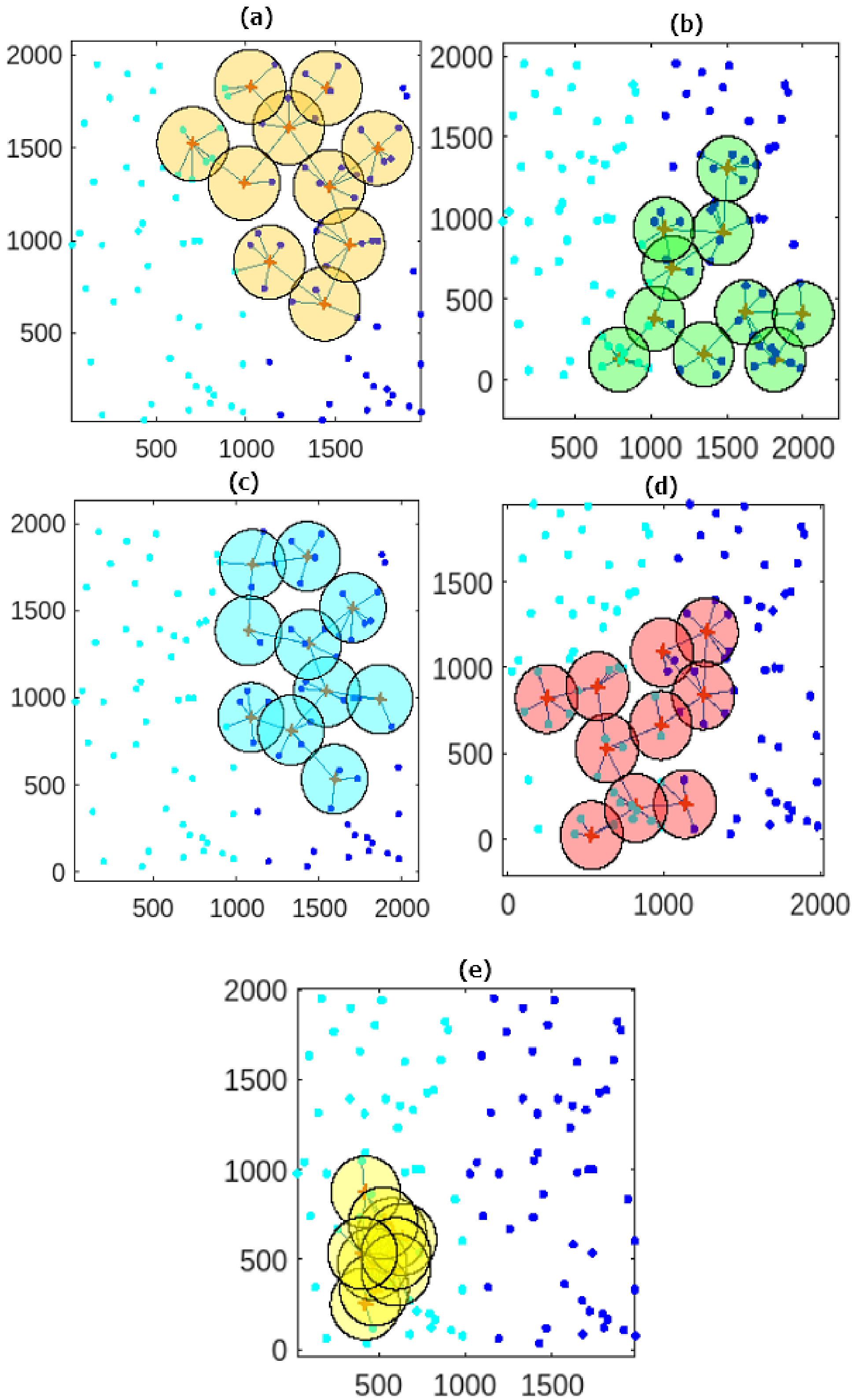
Figures 4.1(a), 4.1(b), 4.1(c), 4.1(d) and 5.3(e) report examples of a planned network using ASO-SA, SO-SA, ASO, SO, and SA, respectively. The planned network is a solution of network instance with 10 MRs and 100 MCs. Light blue points represent MCS having priority equal to 1 (there are 50 MCS distributed in the left half part of the deployment space). Dark blue points represent MCS having priority equal to 5 (there are 50 MCS distributed in the right half part of the deployment space). Each installed MR is represented by orange plus sign. It is clearly seen from this figures that MCs with greater priorities are the first served using ASO-SA, SO-SA, ASO, and SO. Again, ASO-SA, SO-SA, and ASO are more appropriate for solving the mesh routers placement problem with service priority, because the use of this algorithms results in positioning mesh routers in the part where MCs having the higher priority are distributed.

4.6.1 Effect of varying the number of MRs

In the first case of simulation, we will evaluate the performance of the proposed hybrid approach under various number of MRs.

Figure 4.2 and Table 4.4 describe the impact of varying the number of MRs (5 to 40) on fitness value. The presented results showed that the fitness value increases as the number of MRs increases. Again, in comparison with SO-SA, ASO, SO, and SA, the effectiveness of ASO-SA is demonstrated.

Figure 4.1: Placement obtained using: (a) ASO-SA (b) SO-SA (c) ASO (d) SO (e) SA



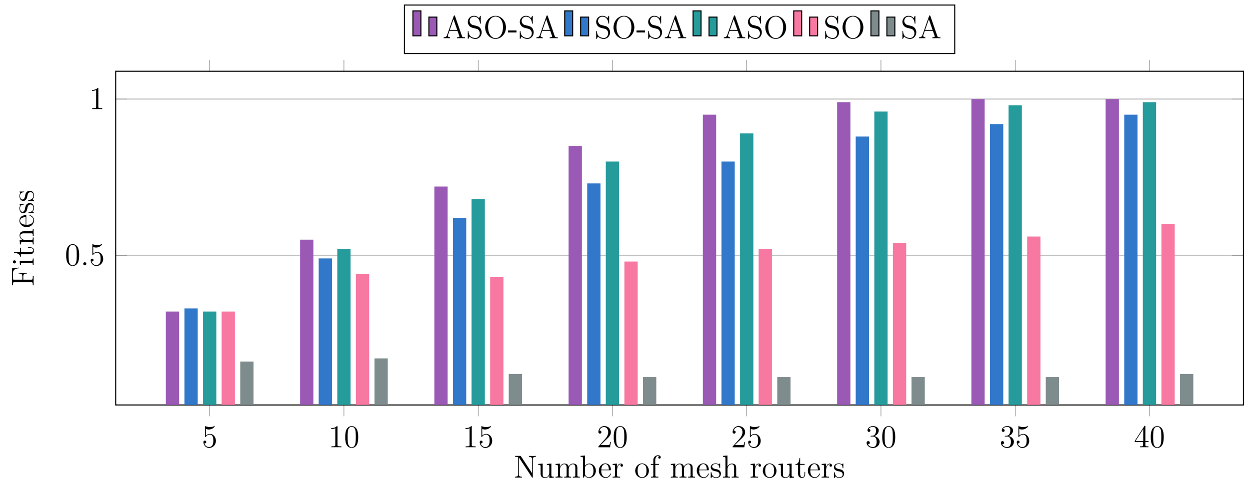


Figure 4.2: Fitness under various number of mesh routers

Table 4.4: Fitness under various number of mesh routers

m	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**
ASO-SA	0.32	**0.55**	**0.72**	**0.85**	**0.95**	**0.99**	**1**	**1**
SO-SA	**0.33**	0.49	0.62	0.73	0.80	0.88	0.92	0.95
ASO	0.32	0.52	0.68	0.80	0.89	0.96	0.98	0.99
SO	0.32	0.44	0.43	0.48	0.52	0.54	0.56	0.6
SA	0.16	0.17	0.12	0.11	0.11	0.11	0.11	0.12

Table 4.5: Fitness under various number of mesh clients

n	**20**	**40**	**60**	**80**	**100**	**120**	**140**	**160**
ASO-SA	**0.88**	**0.90**	**0.92**	**0.90**	**0.88**	**0.87**	**0.85**	0.71
SO-SA	0.78	0.72	0.76	0.77	0.79	0.76	0.73	0.70
ASO	0.83	0.83	0.84	0.84	0.85	0.81	0.80	**0.79**
SO	0.64	0.59	0.59	0.58	0.52	0.48	0.48	0.43
SA	0.13	0.14	0.15	0.14	0.12	0.10	0.11	0.11

4.6.2 Effect of varying the number of MCs

The effect of varying the number of MCs is studied in the second case of simulation. The impact of increasing the number of MCs (20 to 160) on fitness value is depicted in Figure 4.3 and Table 4.5. It is clearly seen that the fitness value slightly decreases when increasing the number of MCs. Once more, ASO-SA performs better than SO-SA, ASO, SO, and SA algorithms for the most of cases.

4.6.3 Effect of varying the coverage radius values

In this scenario, we explored the impact of changing the coverage values from 100 to 500, when trying to cover 140 MCs using 20 MRs. The results of this exploration are shown in Figure 4.4 and Table 4.6. The presented results show that the fitness value increases

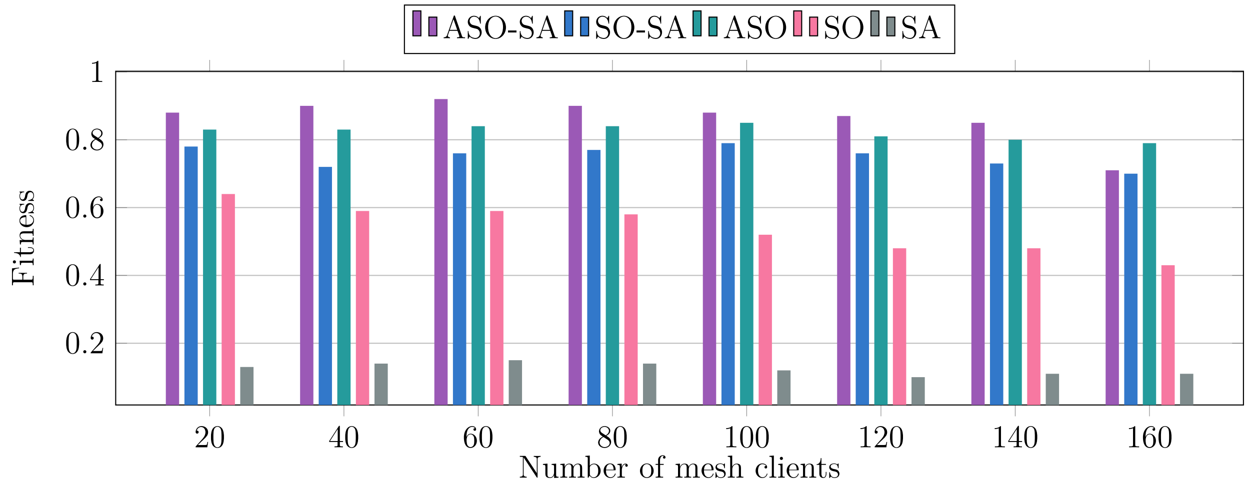


Figure 4.3: Fitness under various number of mesh clients

as the coverage radius of each MR increases. In addition, the effectiveness of ASO-SA is demonstrated in comparison with SO-SA, ASO, SO,and SA algorithms for all cases.

Table 4.6: Fitness under various coverage radius values

CR	**100**	**200**	**300**	**400**	**500**
ASO-SA	**0.28**	**0.85**	**1**	**1**	**1**
SO-SA	0.23	0.73	1	1	1
ASO	0.21	0.80	1	1	1
SO	0.18	0.48	0.80	0.97	1
SA	0.10	0.11	1	1	1

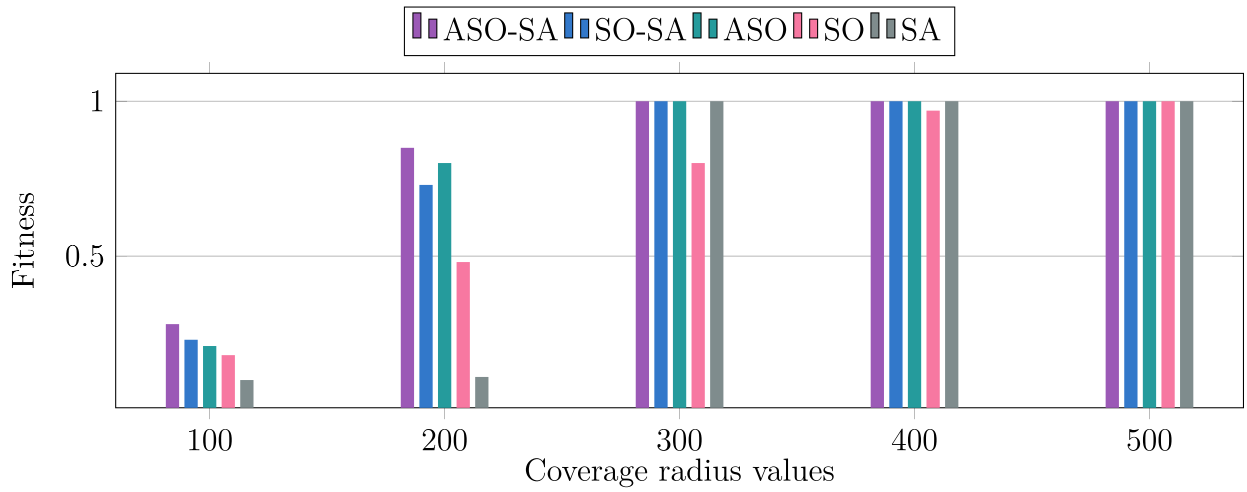


Figure 4.4: Fitness under various coverage radius values

4.6.4 Convergence analysis

In this section, we analyze the convergence of ASO-SA compared to ASO-SA, ASO, SO, and SA algorithms. Table 4.8 shows the convergence analysis of ASO-SA, SO-SA, ASO, SO, SA algorithms. Convergence analysis process is done using four network instances of different sizes (i.e. (a) $Instance_1$, (b) $Instance_2$, (c) $Instance_3$, and (d) $Instance_4$) as illustrated in Table 4.7. The convergence analysis process is based on convergence efficiency (fitness value) and convergence speed. An average of 15 experiments are performed for each result.

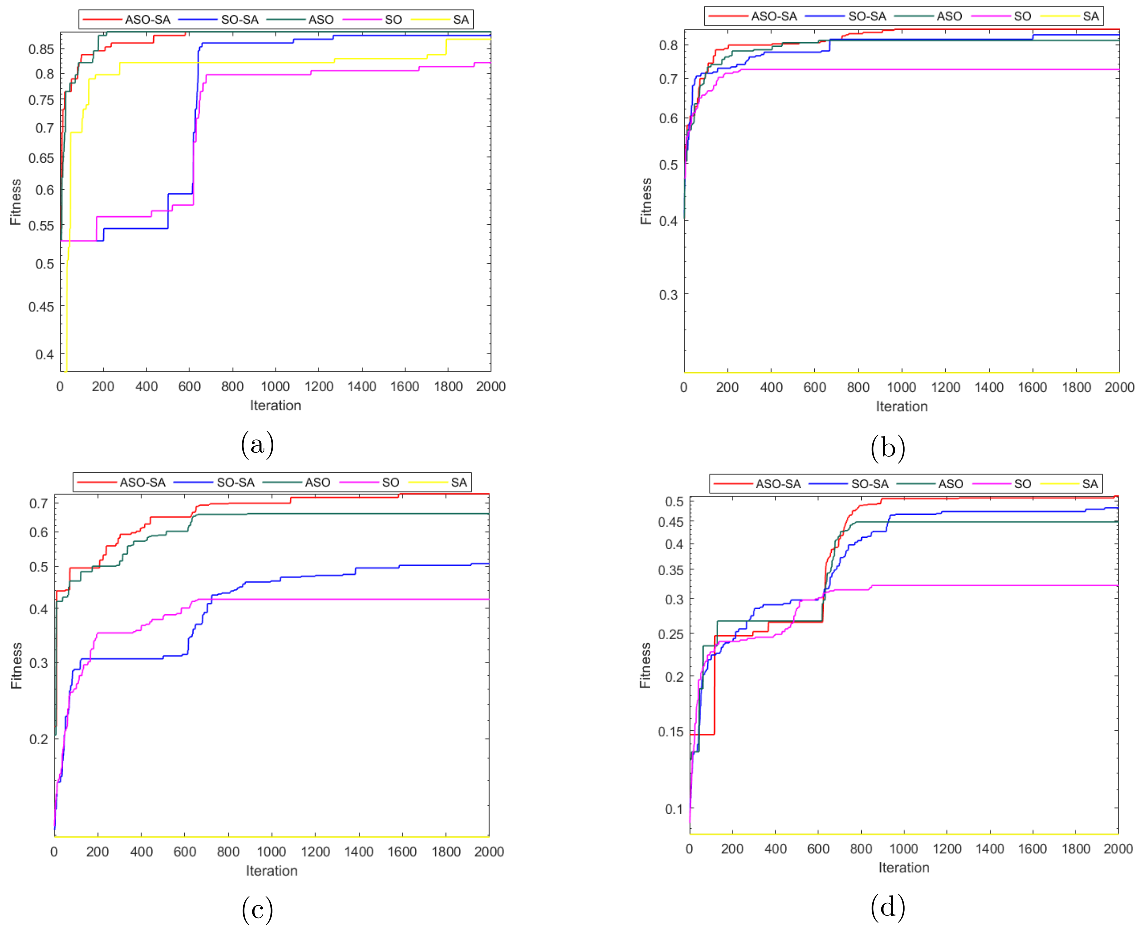


Figure 4.5: Example of algorithms convergence using: (a) $Instance_1$ (b) $Instance_2$ (c) $Instance_3$ (d) $Instance_4$

Table 4.7: Network instances considered in convergence analysis

Instance	WxH	m	n	CR
$Instance_1$	1000x1000	5	50	200
$Instance_1$	1500x1500	10	100	200
$Instance_3$	2000x2000	15	150	200
$Instance_4$	2500x2500	20	200	200

Table 4.8: Convergence analysis between ASO-SA, SO-SA, ASO, SO, and SA algorithms

Instance	$Instance_1$		$Instance_2$		$Instance_3$		$Instance_4$	
	Fitness	Iteration	Fitness	Iteration	Fitness	Iteration	Fitness	Iteration
ASO-SA	**0.89**	914	**0.83**	1357	**0.69**	1326	**0.52**	1478
SO-SA	0.87	1031	0.80	964	0.60	1726	0.50	1477
ASO	0.89	340	0.81	636	0.63	681	0.46	737
SO	0.86	788	0.78	615	0.43	791	0.32	699
SA	0.88	864.6	0.33	386	0.10	1	0.10	1

Figures 4.5a, 4.5b, 4.5c, and 4.5d, report an example of the algorithms convergence using $Instance_1$,$Instance_2$, $Instance_3$, and $Instance_4$.

Based on the fitness values of ASO-SA in comparison with other algorithms as illustrated in Table 4.8, it can be observed that ASO-SA gives better results for four instances in both large and small networks.

In terms of convergence speed, the results described in Table 4.8 and Figures 4.5a, 4.5b, 4.5c, and 4.5d showed that ASO-SA is among algorithms that needs more iterations to achieve the global optimum. However, the fitness reached by ASO-SA is the best for most of the cases. The reasons behind this improvement are given below:

- The movement of the snakes is guided by a combination of attraction forces, repulsion forces, and randomization. This combination of forces allows the Snake optimization algorithm to effectively balance exploration and exploitation, leading to a more efficient search of the search space.
- The integration of GOBL helps to increase the exploration of the search space, while still allowing the algorithm to refine and improve the current best solution.
- The use of SA in the exploitation phase of Snake optimization can help to improve the efficiency of the algorithm and find high-quality solutions.

4.7 Conclusion

In this chapter, we have presented a hybrid approach ASO-SA, based on the combination of ASO and SA for solving the mesh routers placement problem in WMNs with service priority. ASO is based on the integration of GOBL technique for enhancing the exploration phase of SO. Simulation results demonstrated the effectivness of the proposed hybrid approach in comparison with ASO, SO-SA, SO, and SA algorithms.

In the next chapter, we will present a binary approach for solving the topology planning problem in WMNs

CHAPTER 5

BINARY WHALE OPTIMIZATION ALGORITHM FOR TOPOLOGY PLANNING PROBLEM IN WMNS

5.1 Introduction

The topology planning problem in WMN with cost minimization must be solved in discrete space, thus a binary approach is required to solve this issue. In this sense, a Binary Whale Optimization Algorithm (BWOA) is presented in this chapter to solve the concerned problem. Height transfer functions divided into two families such as S-shaped and V-shaped are introduced and analyzed in this study to obtain a binary version of WOA.

In this chapter, we present a reminder of the approaches proposed in the literature to solve the topology planning problem in WMNs with cost minimisation, we then describe the formulation of the topology planning problem in WMNs, we also present Whale Optimization Algorithm (WOA) and the binary version of WOA. Finally, the performance of the BWOA with s-shaped and v-shaped transfer functions is evaluated.

5.2 Related works

The topology planning problem in WMN with cost minimization is combinatorial problem, solved successfully using meta-heuristic approaches SA [125], TS [126], Quantum-Inspired Evolutionary Algorithm (QIEA), GA [127, 128, 129, 130], and PSO [131].

SA was used in the work of Nawaf et al. [125] for solving the gateway placement problem in WMNs. SA algorithm was validated based on 23 instances taking into account 3 metrics such as cost, coverage, and throughput. Simulation results showed that SA algorithm produces a set of effective optimization solutions.

An improved TS was proposed in the work of Wang et al. [126] for optimizing the deployment of mesh-routers in WMNs. The improved TS was validated for the 3D mountain environment in the Wanglang national nature reserve of Sichuan province, taking into account the cost, coverage, and connectivity metrics. Simulation results showed that the mountain is almost completely covered (more than 90%) with only a small part of the not-covered region.

Evolutionary algorithms (e.g. Genetic Algorithm (GA)) have been popular optimization algorithms in this area too . For instance, Ahmed et al. [127] applied GA approach for solving the gateways placement problem in WMNs. GA approach was validated using many generated instances under different conditions (population size, tournament size, crossover type, and mutation type). Experimental results showed the robustness of the GA approach in terms of deployment cost, convergence rate, and scalability. A modified GA, called MTMG was proposed in [132], for solving the gateways deployment problem in WMNs. MTMG was validated based on 6 instances with different gateway deployment locations taking into account the cost and throughput metrics. Simulation results demonstrated the effectiveness of MTMG when compared with ICLB-GPS algorithm and weighted recursive algorithm. Authors in [128] proposed a multi-objective GA approach, called EGA-GP, for solving the gateways placement problem in WMNs. EGA-GP was assessed taking into account the deployment cost and delay metrics. Simulation results proved the performance of EGA-GP when compared with greedy algorithm and GA. Again, a modified GA was proposed in the work of Tang and Chen [129], called Repairing GA (RGA), for solving the gateways placement problem in WMNs. RGA was validated in terms of number of gateways and computational time using 10 test problems of different sizes. Experimental results demonstrated the satisfactory performance of RGA when compared with the incremental clustering algorithm.

Le et al. [131] applied PSO algorithm for solving the gateway placement problem in WMNs. PSO algorithm was validated based on two experiments taking into account the deployment cost and throughput metrics. Simulation results showed the superiority of PSO algorithm compared to other WMNs planning studies found in the literature.

5.3 The system model and problem formulation

WMN considered in this work is represented by a graph $G = (V, E)$ where V is the set of MRs and E describes the set of links between these MRs. Each MR is equipped with a radio interface with the same coverage radius CR. Let $MCS = \{MC_1, ..., MC_n\}$ be the set of MCs and $CS = \{CS_1, ..., CS_s\}$ be the set of Candidate Sites to host a MR. Each MC is said to be covered by various installed MRs if it is within its transmission range. It is assigned to the closest one. A link can be established between two installed MRs if the distance between them is less than the sum of the transmission ranges of the two MRs.

The coverage variable a_{ij} is given as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } \mathrm{MC_i} \text{ is covered by } \mathrm{CS_j} \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

Let $i \in MCS, j \in CS$, the assignment variable x_{ij} is described as follows:

$$x_{ij} = \begin{cases} 1 & \text{if } \mathrm{MC_i} \text{ is assigned to } \mathrm{CS_j} \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

The installation variables r_j is given as follows:

$$r_j = \begin{cases} 1 & \text{if MR is installed in } \mathrm{CS_j}, \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

Let j and $l \in CS$, $j \neq l$, the connectivity variable A_{jl} (adjacency matrix) is specified as follows:

$$A_{jl} = \begin{cases} 1 & \text{if } \mathrm{CS_j} \text{ and } \mathrm{CS_l} \text{ can be connected} \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

The objective of our planning problem is to select the set of candidate sites where MRs can be installed to meet the full coverage and full connectivity requirements. The problem can be formulated as follows:

$$m = \text{Min} \sum_{j \in CS} r_j \tag{5.5}$$

Subject to:

$$\sum_{j \in CS} x_{ij} = 1 \quad \forall i \in MCS \tag{5.6}$$

$$x_{ij} \leqslant r_j a_{ij} \quad \forall i \in MCS \quad \forall j \in CS, \tag{5.7}$$

$$\sum_{k=1}^{m-1} \mathrm{A}^k \neq 0 \tag{5.8}$$

The objective function in (5.5) reduces the overall number of installed MRs m in the network. Equation 5.6 ensures the full coverage of all MCs. Inequality 5.7 implies that a MC_i is affected and covered by an installed MR in CS_j. The equation 5.8 imposes to have at least one path between each pair of MRs. Consequently, the graph G is fully connected.

5.4 Whale Optimization Algorithm (WOA)

Whale Optimization Algorithm (WOA) is a new swarm intelligence method introduced by Mirjalili and Lewis [50] in 2016. The humpback whales' hunting strategies form the basic concept of WOA. The exploitation and exploration phases of WOA are simulated based on the encircling prey mechanism and spiral updating position approach. The description is given as follows:

- **Encircling prey: if ($\boldsymbol{P} < \mathbf{0.5}$ and $|\boldsymbol{A}| < \mathbf{1}$)** The position of the solution $X(t+1)$ is updated using equations (5.9) and (5.10):

$$\overrightarrow{D'} = |C\overrightarrow{X}_{best}(t) - \overrightarrow{X}(t)| \tag{5.9}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X}_{best}(t) - \overrightarrow{A}.\overrightarrow{D'} \tag{5.10}$$

  where t denotes the current iteration, $\overrightarrow{X_{best}}$ and $\overrightarrow{X}$ represent the best and the current solutions, respectively. The vectors of coefficients $\overrightarrow{A}$ and $\overrightarrow{C}$ are calculated as in equations (5.11) and (5.12).

$$\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r1} - \overrightarrow{a} \tag{5.11}$$

$$\overrightarrow{A} = 2\overrightarrow{r} \tag{5.12}$$

  where $\overrightarrow{a}$ drops linearly from 2 to 0 over iterations (simulating the shrinking encircling behavior as in equation (5.11)) and $\overrightarrow{r1}$ is a random vector in the range $[0, 1]$. The formula for a is given below.

$$a = 2(1 - t/t_{max}) \tag{5.13}$$

  where t_{max} is the total number of iterations.

- **Spiral updating position: if $\boldsymbol{P} < \mathbf{0.5}$**

  The spiral-shaped path followed by the whales is modeled using the spiral rule in equation (5.14)

$$\overrightarrow{X}(t+1) = \overrightarrow{D}.e^{bl}.cos(2\pi l) + \overrightarrow{X}_{best}(t) \tag{5.14}$$

$$\overrightarrow{D} = |\overrightarrow{A}.\ \overrightarrow{X}_{best}(t) - \ \overrightarrow{X}(t)| \tag{5.15}$$

  where b is a constant, l is a random number in the interval [-1,1], $\overrightarrow{D}$ indicates the distance between the current solution and $\overrightarrow{X_{best}}$ at iteration t.

To update the location of the whale as shown in equation (5.16), a search agent is chosen randomly from the population.

$$\overrightarrow{D} = |\overrightarrow{C}\overrightarrow{X}_{rand}(t) - \overrightarrow{X}(t)| \tag{5.16}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X}_{rand}(t) - \overrightarrow{A}.\overrightarrow{D} \tag{5.17}$$

where $\overrightarrow{X}_{rand}$ is a randomly selected search agent from the current population, $\overrightarrow{A}$ is a vector with random values in range $[-1, 1]$.

The main steps of WOA are given in algorithm 13.

Algorithm 13 The WOA Algorithm

Initialize WOA parameters
Randomly initialize the population of solutions
Evaluate the population and determine X_{best}
for t=1 to t_{max} **do**
    for i=1 to N **do**
        Update a, A, C, l and P
        if $P < 0.5$ **then**
            if $|A| < 1$ **then**
                Calculate $X_i(t+1)$ using equation 5.10
            else
                Calculate $X_i(t+1)$ using equation 5.14
            end if
        else
            Calculate $X_i(t+1)$ using equation 5.17
        end if
    end for
    Evaluate the population and determine X_{best}
end for
Return the best solution

5.5 Binary Whale Optimization Algorithm (BWOA)

The topology planning problem belongs to the family of NP-hard problems. For this reason, meta-heuristic algorithms can be a suitable alternative to solve this problem with reasonable time execution. In this chapter, we use a similar notion by applying a meta-heuristic algorithm to solve the considered algorithm. The proposed approach is the new bio-inspired optimization algorithm called Whale Optimization Algorithm (WOA), inspired by the hunting behavior of humpback behavior. WOA was used to solve a variety of problems such as classification [133, 134, 135], path planning [136, 137, 138, 139, 140], clustering [141, 142, 143], placement problems [144, 145, 146, 147]. The good performance of WOA motivates our attempt to apply WOA for solving the topology planning problem. WOA may result in premature convergence leading the search space to be trapped in local optimum [148]. The design problem tackled in this paper must solve in discrete space, thus we have to implement a binary version of WOA to deal with this problem.

The binarization of WOA requires the use of transfer functions in order to obtain '0' or

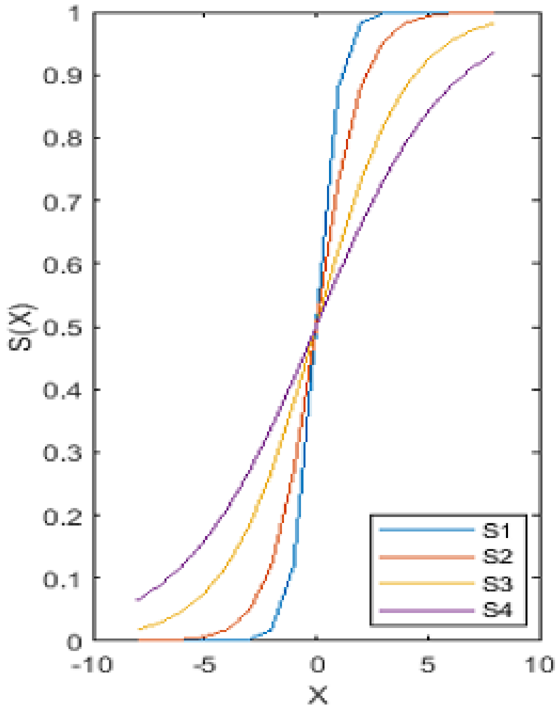


Figure 5.1: S-shaped family of transfer functions.

'1' values. In this chapter, we use eight transfer functions divided into two families including S-shaped and V-shaped. The four transfer functions belonging to the S-shaped family are denoted by S1, S2, S3, and S4, respectively. The binary versions of WOA based on these functions are denoted by BWOAS1, BWOAS2, BWOAS3, and BWOAS4, respectively. S-shaped transfer functions are described in Table 5.1 [149] and Figure 5.1.

Table 5.1: S-shaped family of transfer functions

Experiment name	Name	Transfer function
BWOAS1	S1	$S(x) = \frac{1}{1+e^{-2x}}$
BWOAS2	S2	$S(x) = \frac{1}{1+e^{-2x}}$
BWOAS3	S3	$S(x) = \frac{1}{1+e^{(-x/2)}}$
BWOAS4	S4	$S(x) = \frac{1}{1+e^{(-x/3)}}$

V1, V2, V3, and V4 represent the transfer functions belonging to the V-shaped family. BWOAV1, BWOAV2, BWOAV3, and BWOAV4 are the binary versions of WOA based on V1, V2, V3, and V4, respectively. V-shaped transfer functions are described in Table 5.2 [149] and Figure 5.2. Based on the S-shaped function, the binarization method is specified as follows:

$$X_i^j(t+1) = \begin{cases} 1, & \text{if } \operatorname{rand}() < S\left(X_i^j(t+1)\right) \\ 0, & \text{if } \operatorname{rand}() \geq S\left(X_i^j(t+1)\right) \end{cases} \tag{5.18}$$

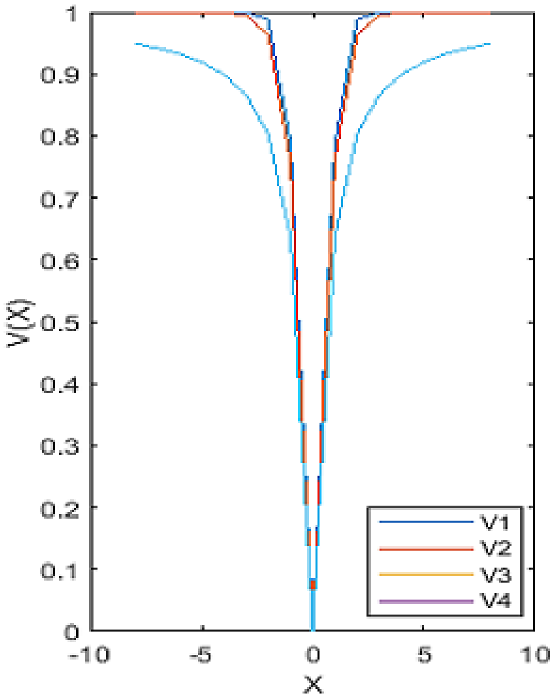


Figure 5.2: V-shaped family of transfer functions.

Table 5.2: V-shaped family of transfer functions

Exprement name	Name	Transfer function
BWOAV1	V1	$V(x) = \left\lvert \mathrm{erf}\left(\frac{\sqrt{\pi}}{2}x\right)\right\rvert = \left\lvert \frac{\sqrt{\pi}}{2}\int_0^{(\sqrt{\pi}/2)x} e^{-t^2}dt\right\rvert$
BWOAV2	V2	$V(x) = \lvert\tanh(x)\rvert$
BWOAV3	V3	$V(x) = \lvert(x)/\sqrt{1+x^2}\rvert$
BWOAV4	V4	$V(x) = \left\lvert\frac{2}{\pi}\arctan\left(\frac{\pi}{2}x\right)\right\rvert$

On the other hand, based on V-shaped functions, the binarization method is specified as follows:

$$X_i^j(t+1) = \begin{cases} \sim X_i^j(t) & \text{if rand } < V\left(X_i^j(t+1)\right) \\ X_i^d(t) & \text{else} \end{cases} \tag{5.19}$$

5.6 Simulation results

In this section, we will evaluate the performance of the eight binary versions of WOA. All binary versions of WOA are coded in Matlab. All simulations are carried out on a Core i7 machine. A population of 20 solutions was considered in all simulations. The total number of iterations is set to 1000. Each result presented in this section is an average of 10 runs.

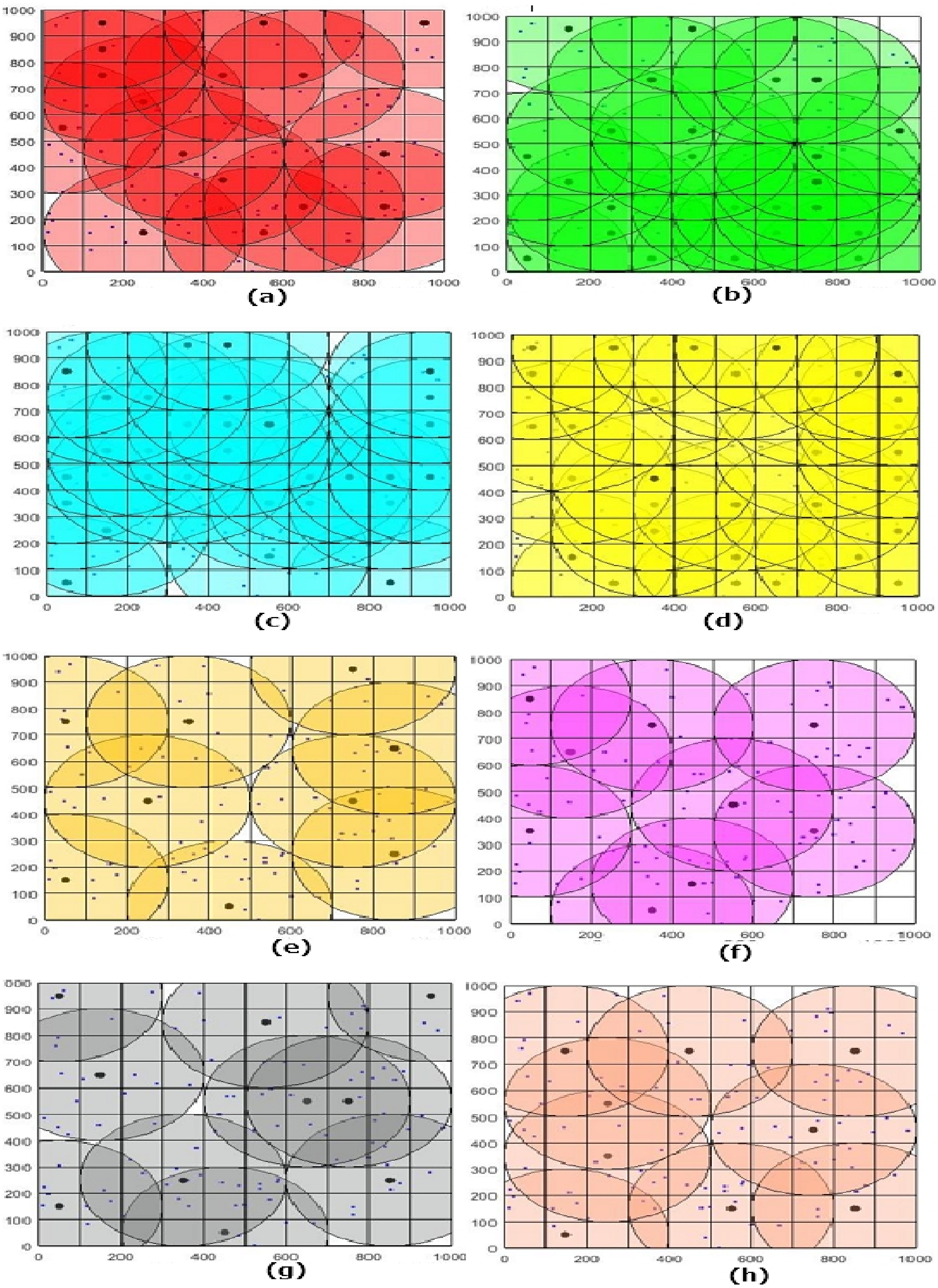


Figure 5.3: Obtained placements using: (a) BWOAS1 (b) BWOAS2 (c) BWOAS3 (d) BWOAS4 (e) BWOAV1 (f) BWOAV2 (g) BWOAV3 (h) BWOAV4

The evaluation process is done in terms of three metrics including minimum, maximum, and average number of MRs, taking into account various numbers of MCs (from 20 to 100) and different transmission ranges (from $250m$ to $500m$). A grid topology of 10×10 was considered for the candidate sites. Figures 5.3.a, 5.3.b, 5.3.c, 5.3.d, 5.3.e, 5.3.f, 5.3.g, 5.3.h

Table 5.3: Parameters values considered in our simulations

Parameter	**Value**	**Default value**
n	[20 100]	100
CR	[250m 500m]	250m
Grid	10x10	10x10
Population size	20	20
Number of runs	10	10
Number of iterations	1000	1000

report an example of planned network using BWOAS1, BWOAS2, BWOAS3, BWOAS4, BWOAV1, BWOAV2, BWOAV3, and BWOAV4, respectively. The planned network is a solution of network instance with 100 MCs randomly distributed in a deployment area of $1000m \times 1000m$ (grid 10×10), represented by blue points. Each installed MR is represented by the Bold black point. It is clearly seen that binary versions of WOA based V-shaped functions are more appropriate for solving the topology planning problem in WMNs.

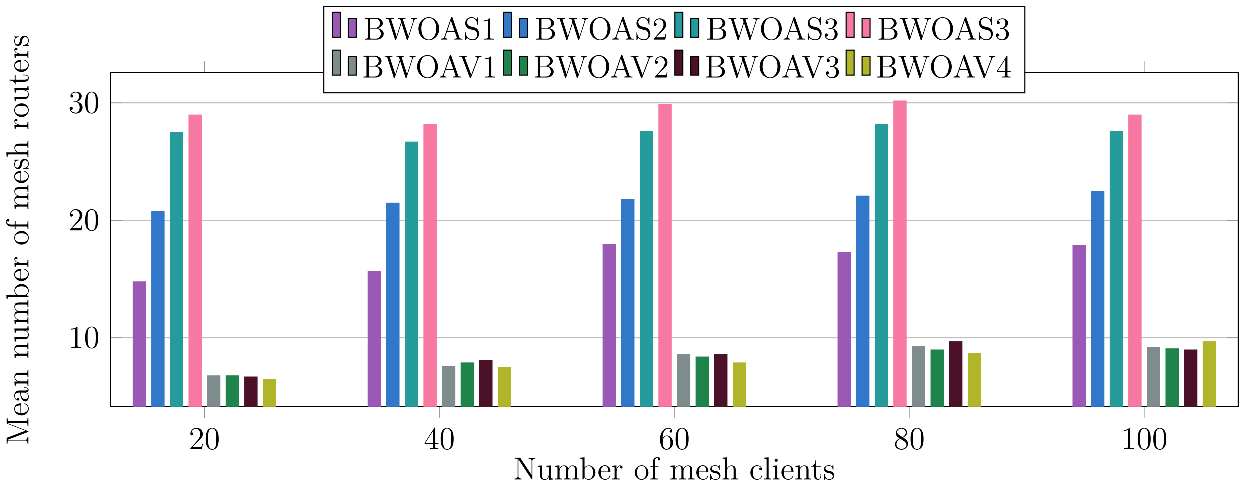


Figure 5.4: Mean number of mesh routers under various number of mesh clients

5.6.1 Effect of varying the number of MCs

In the first case of simulation, we will evaluate the performance of the binary versions of WOA under various numbers of mesh clients. The results are presented in Table 5.4 and Figure 5.4. It's clearly seen that the number of mesh routers increases while increasing the number of mesh clients. In fact, more routers are needed for covering the newly added mesh clients. Again, it can be stated that binary versions of WOA employing the V-shaped transfer functions are capable of finding global solutions better when compared with those utilizing S-shaped functions. The best results of the BWOA algorithms with V-shaped transfer function are those obtained by BWOAV4. On the other hand, among BWOA algorithms with S-shaped transfer functions, BWOAS1, which uses S1, gets better results when compared with BWOAS2, BWOAS3, and BWOAS4 algorithms.

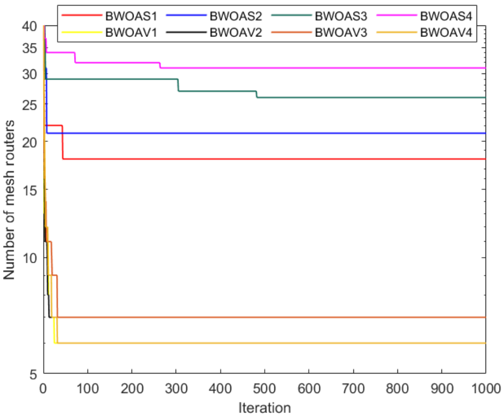
Figures 5.5a, 5.5b, 5.5c, 5.5d, and 5.5e depict the convergence of the proposed algorithms under various numbers of mesh clients (20, 40, 60, 80, and 100, respectively). Its clearly seen that BWOA with V-shaped family of transfer functions are capable of determining the global solution with few needed iterations.

Table 5.4: Maximum, minimum, and mean number of mesh routers under various number of mesh clients

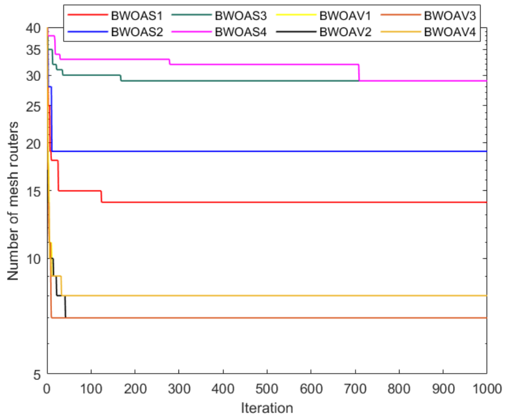
S-shaped						**V-shaped**					
n	**20**	**40**	**60**	**80**	**100**	n	**20**	**40**	**60**	**80**	**100**
BWOAS1						**BWOAV1**					
Maximum	16	18	19	20	21	Maximum	8	9	10	10	10
Mean	14.8	15.7	18	17.3	17.9	Mean	6.8	7.6	8.6	9.3	9.2
Minimum	12	13	17	14	16	Minimum	6	7	8	8	7
BWOAS2						**BWOAV2**					
Maximum	24	24	25	25	26	Maximum	8	9	10	10	10
Mean	20.8	21.5	21.8	22.1	22.5	Mean	6.8	7.9	8.4	9	9.1
Minimum	17	16	19	18	19	Minimum	6	7	7	8	8
BWOAS3						**BWOAV3**					
Maximum	30	28	31	31	30	Maximum	8	9	9	11	10
Mean	27.5	26.7	27.6	28.2	27.6	Mean	6.7	8.1	8.6	9.7	9
Minimum	24	23	24	24	25	Minimum	6	6	8	8	8
BWOAS4						**BWOAV4**					
Maximum	32	32	33	33	31	Maximum	8	8	9	11	11
Mean	29	28.2	29.9	30.2	29	Mean	**6.5**	**7.5**	**7.9**	**8.7**	9.7
Minimum	24	24	24	27	26	Minimum	6	7	7	8	8

5.6.2 Effect of varying the coverage radius values

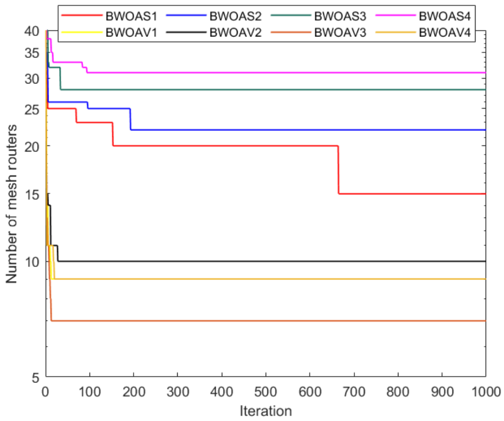
In the second case of simulation, the performance of the binary versions of WOA is evaluated under various coverage radius values. The results are presented in Table 5.5 and Figure 5.6. It's clearly seen that the number of mesh routers decreases as the coverage radius value of each router increases. In fact, increasing the coverage radius of a mesh router effectively increases the area that a single router can cover. This means that with a larger coverage radius, fewer mesh routers are needed to cover the same number of mesh clients as before, since each router can cover a larger area. It can be stated that binary implementations of WOA that utilize V-shaped transfer functions have an improved ability to locate global solutions when compared to those that employ S-shaped functions. The most favorable outcomes of the BWOA algorithms with a V-shaped transfer function were achieved by BWOAV2. Conversely, among the BWOA algorithms that use an S-shaped transfer function,



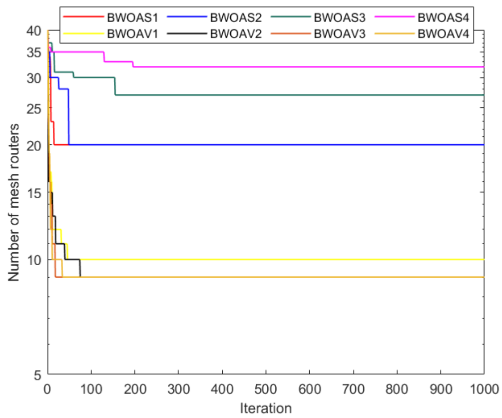
(a)



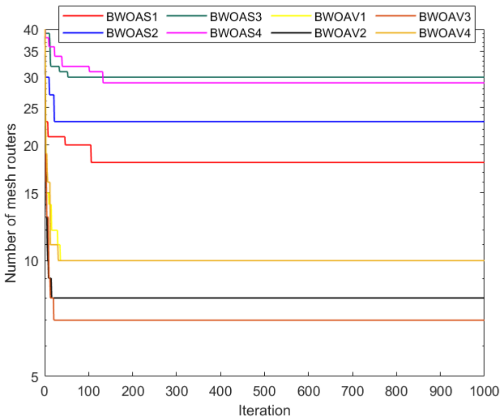
(b)



(c)



(d)



(e)

Figure 5.5: Example of algorithms convergence using: (a) 20 MCs (b) 40MCs (c) 60 MCs (d) 80 MCs (e) 100 MCs

BWOAS1, utilizing S1, produced better results compared to the BWOAS2, BWOAS3, and BWOAS4 algorithms.

Table 5.5: Maximum, minimum, and mean number of mesh routers under various coverage radius values

S-shaped							V-shaped						
CR	**250**	**300**	**350**	**400**	**450**	**500**	CR	**250**	**300**	**350**	**400**	**450**	**500**
BWOAS1							**BWOAV1**						
Maximum	21	16	16	16	16	16	Maximum	8	8	6	5	5	4
Mean	17.9	15.1	13	13.6	12.4	13.1	Mean	9.2	7.3	5.4	4.4	3.9	3.1
Minimum	16	14	10	9	8	10	Minimum	7	4	4	3	3	7
BWOAS2							**BWOAV2**						
Maximum	26	24	23	23	23	21	Maximum	10	8	6	4	5	4
Mean	22.5	21.4	21.1	19.8	20.1	18.6	Mean	**9.1**	**7**	**5.1**	**4**	4	3.3
Minimum	19	19	18	16	16	14	Minimum	8	6	4	4	3	3
BWOAS3							**BWOAV3**						
Maximum	30	29	30	30	30	29	Maximum	10	9	6	5	4	3
Mean	27.6	26.7	27.8	27	27.2	27.5	Mean	9	7.2	5.7	4.5	**3.5**	**3**
Minimum	25	23	23	25	25	26	Minimum	8	6	5	4	3	3
BWOAS4							**BWOAV4**						
Maximum	31	32	32	32	33	31	Maximum	11	9	7	6	5	4
Mean	29	29.2	28.9	29.7	29.4	28.6	Mean	9.7	7.2	5.7	4.4	3.7	3.1
Minimum	26	27	27	28	27	26	Minimum	8	6	4	4	3	2

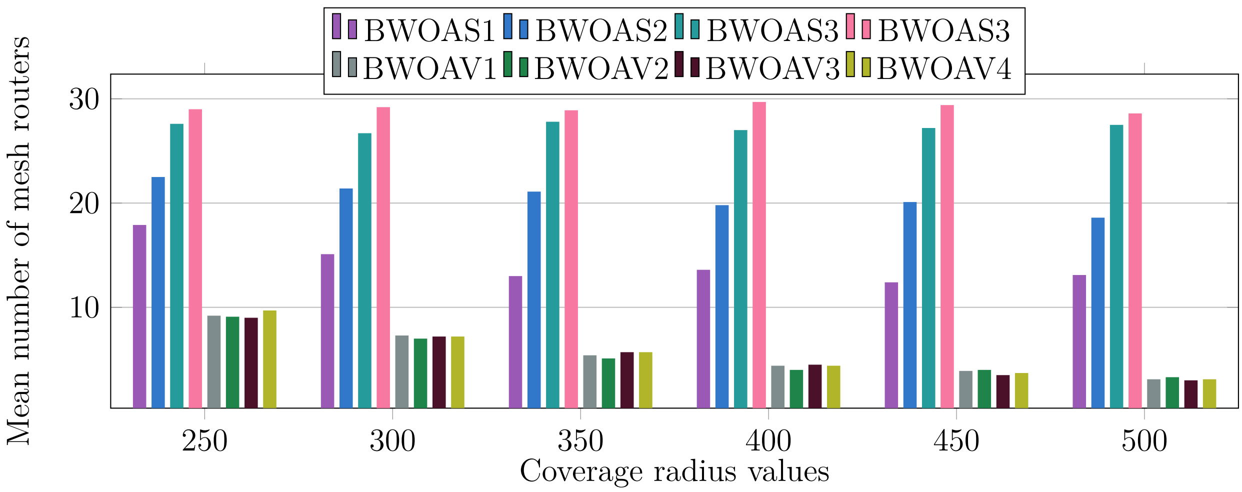


Figure 5.6: Mean number of mesh routers under various coverage radius values

Figures 5.7a, 5.7b, 5.7c, 5.7d, and 5.7e illustrate the convergence of the proposed algorithms with various coverage radius values (300m, 350m, 400m, 450m, and 500m, respectively). It is evident that BWOA using the V-shaped family of transfer functions is able to find the global solution with a minimal number of iterations.

5.7 Conclusion

Height binary versions of WOA were suggested in this chapter. S-shaped and V-shaped transfer functions are used to transform the original version of WOA into a binary version. Our primary objective is to reduce the number of MRs required to provide full coverage

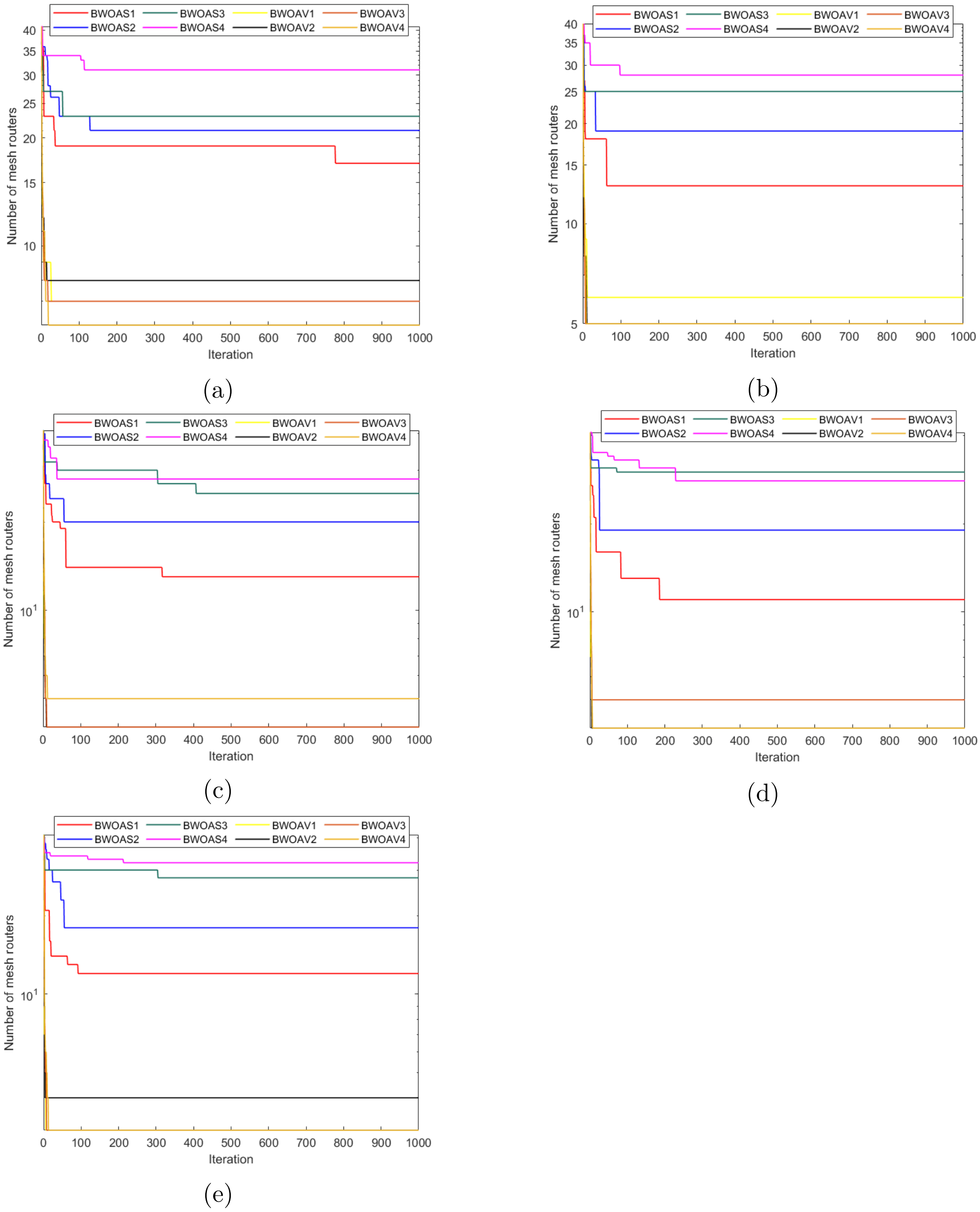


Figure 5.7: Example of algorithms convergence using various coverage radius values: (a) $300m$ (b) $350m$ (c) $400m$ (d) $450m$ (e) $500m$

and full connectivity. The evaluation is conducted in terms of three metrics, including minimum, maximum, and average number of mesh routers, using different scenarios with various number of mesh clients and various coverage radius values. The results of the simulations demonstrated that BWOA based on V-shaped transfer functions gives better results when compared with those based on S-shaped transfer functions.

GENERAL CONCLUSION

WMNs play a key role in new generation wireless networks. This is due both to their easy implementation, their low deployment cost as well as to the many fields of application. However, one of the constraints of these networks is the problem of positioning mesh routers in manner to optimize performance metrics and satisfying quality of service requirements. The objective of this thesis is to solve the mesh routers placement problem considering different formulations and different approaches. We offered three major contributions:

In our first contribution, we presented an improved version of MFO, called ECLO-MFO, based on the incorporation of three strategies including: the chaotic map, LFD strategy, and OBL technique. The experimental results demonstrated the superiority and efficiency of the proposed algorithm in comparison with other well known algorithms in the literature.

In the second contribution, to solve the mesh routers placement problem with service priority, we have proposed a hybrid approach, called ASO-SA, based on the global search of an Adaptive SO (ASO) an the local search of SA. ASO is based on the incorporation of GOBL technique to improve the exploration phase of SO. Simulation results demonstrated the good performance of ASO-SA in comparison with SO, ASO-SA, ASO, and SA algorithms.

In the last contribution, a binary WOA was proposed for solving the topology planning problem in WMNs. Eight transfer functions divided into two families such as S-shaped and V-shaped are introduced and analyzed to obtain a binary version of WOA. The results of the simulations demonstrated that BWOA based on V-shaped transfer functions gives better results when compared with those based on S-shaped transfer functions.

In perspective, we aim to apply multi-objective approaches for optimizing multiple metrics simultaneously while satisfying the quality of service requirements. Furthermore, we propose to investigate joint design, including mesh router placement, gateway selection, antenna placement, frequency assignment, and routing. Additionally, we aim to utilize Machine Learning (ML) techniques for solving the design problem in Wireless Mesh Networks (WMNs).

BIBLIOGRAPHY

[1] Sylia Mekhmoukh Taleb, Yassine Meraihi, Asma Benmessaoud Gabis, Seyedali Mirjalili, and Amar Ramdane-Cherif. Nodes placement in wireless mesh networks using optimization approaches: a survey. *Neural Computing and Applications*, pages 1–37, 2022.

[2] Anis Ouni. *Optimisation de la capacité et de la consommation énergétique dans les réseaux maillés sans fil.* PhD thesis, INSA de Lyon, 2013.

[3] Ian F Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Computer networks*, 47(4):445–487, 2005.

[4] Ian F Akyildiz and Xudong Wang. Wireless mesh networks (vol. 3), 2009.

[5] Yuan Chai and Xiao-Jun Zeng. Load-and interference-balance hybrid routing protocol for hybrid wireless mesh network. In *2019 Wireless Days (WD)*, pages 1–4. IEEE, 2019.

[6] Mohammed Mahseur. *Routage dans les réseaux maillés sans fil.* PhD thesis, Université de Boumerdès-M'hamed Bougara, 2011.

[7] Usman Ashraf. *Qualité de service et routage dans les réseaux maillés sans fil.* PhD thesis, Toulouse, INSA, 2010.

[8] Christelle Molle. *Optimisation de la capacité des réseaux radio maillés.* PhD thesis, Université Nice Sophia Antipolis, 2009.

[9] Mustapha Boushaba. Routage adaptatif et stabilité dans les réseaux maillés sans fil. 2013.

[10] Claude Roy. *Amélioration de la performance des réseaux maillés sans fil cognitifs.* PhD thesis, École de technologie supérieure, 2012.

[11] Bing He. *Architecture design and performance optimization of wireless mesh networks.* University of Cincinnati, 2010.

[12] Jérémie Defaye. Les différents types de réseaux sans fil, 2006.

[13] Ian F Akyildiz and Xudong Wang. A survey on wireless mesh networks. *IEEE Communications magazine*, 43(9):S23–S30, 2005.

[14] Md Hamid, Mohammad Abdullah-Al-Wadud, Choong Seon Hong, Oksam Chae, Sungwon Lee, et al. A robust security scheme for wireless mesh enterprise networks. *annals of telecommunications-annales des télécommunications*, 64(5):401–413, 2009.

[15] Bok-Nyong Park, Wonjun Lee, Sanghyun Ahn, and Sungjoon Ahn. Qos-driven wireless broadband home networking based on multihop wireless mesh networks. *IEEE Transactions on Consumer Electronics*, 52(4):1220–1228, 2006.

[16] Kundan Kumar Kaushik and Neetesh Kumar Gupta. Artificial intelligence based energy efficient dch based wsn using fuzzy inference system. In *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pages 1–5. IEEE, 2020.

[17] GUO Wenqi Wendy, Willam M Healy, and ZHOU Mengchu. Wireless mesh networks in intelligent building automation control: A survey. *International Journal of Intelligent Control and Systems*, 16(1):28–36, 2011.

[18] Kashif Saleem, Khan Zeb, Abdelouhid Derhab, Haider Abbas, Jalal Al-Muhtadi, Mehmet A Orgun, and Amjad Gawanmeh. Survey on cybersecurity issues in wireless mesh networks based ehealthcare. In *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–7. IEEE, 2016.

[19] Ian F Akyildiz and Xudong Wang. A survey on wireless mesh networks. *IEEE Communications magazine*, 43(9):S23–S30, 2005.

[20] Mohamed Abou El Saoud, Hussein Al-Zubaidy, and Samy Mahmoud. Connectivity model for wireless mesh networks. In *2008 IEEE International Conference on Communications*, pages 2889–2894. IEEE, 2008.

[21] Bassam Aoun, Raouf Boutaba, Youssef Iraqi, and Gary Kenward. Gateway placement optimization in wireless mesh networks with qos constraints. *IEEE Journal on Selected Areas in Communications*, 24(11):2127–2136, 2006.

[22] Hanaa Hachimi. *Hybridations d'algorithmes métaheuristiques en optimisation globale et leurs applications.* PhD thesis, INSA de Rouen; École Mohammadia d'ingénieurs (Rabat, Maroc), 2013.

[23] Facundo Bre and Víctor D Fachinotti. A computational multi-objective optimization method to improve energy efficiency and thermal comfort in dwellings. *Energy and Buildings*, 154:283–294, 2017.

[24] K Naidu, Hazlie Mokhlis, and AH Abu Bakar. Multiobjective optimization using weighted sum artificial bee colony algorithm for load frequency control. *International Journal of Electrical Power & Energy Systems*, 55:657–667, 2014.

[25] R Timothy Marler and Jasbir S Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41(6):853–862, 2010.

[26] Abdullah Konak, David W Coit, and Alice E Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability engineering & system safety*, 91(9):992–1007, 2006.

[27] Ailsa H Land and Alison G Doig. An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958-2008*, pages 105–132. Springer, 2010.

[28] John E Mitchell. Branch-and-cut algorithms for integer programming. 1998.

[29] Sukanta Nayak. *Fundamentals of Optimization Techniques with Algorithms*. Academic Press, 2020.

[30] G Polya. How to solve it, princeton univ. *Press, Princeton, NJ*, 1945.

[31] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[32] Nicholas Hao Zheng Lim, Ying Loong Lee, Mau Luen Tham, Yoong Choon Chang, Allyson Gek Hong Sim, and Donghong Qin. Coverage optimization for uav base stations using simulated annealing. In *2021 IEEE 15th Malaysia International Conference on Communication (MICC)*, pages 43–48. IEEE, 2021.

[33] Majid Raissi-Dehkordi, Karthikeyan Chandrashekar, and John S Baras. Uav placement for enhanced connectivity in wireless ad-hoc networks. Technical report, 2004.

[34] Fatos Xhafa, Admir Barolli, Christian Sánchez, and Leonard Barolli. A simulated annealing algorithm for router nodes placement problem in wireless mesh networks. *Simulation Modelling Practice and Theory*, 19(10):2276–2284, 2011.

[35] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549, 1986.

[36] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100, 1997.

[37] Ilhem Boussaid. *Perfectionnement de métaheuristiques pour l'optimisation continue.* PhD thesis, Paris Est, 2013.

[38] Holland Jh. Adaptation in natural and artificial systems. *Ann Arbor*, 1975.

[39] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5):8091–8126, 2021.

[40] John R Koza. Genetic programming: On the programming of computers by means of natural selection (complex adaptive systems). *A Bradford Book*, 1:18, 1993.

[41] LJ Fogel, AJ Owens, and MJ Walsh. Introduction to artificial intelligence through simulated evolution. *Artificial Intelligence: Critical Concepts*, 3:259, 2000.

[42] Marco Dorigo and Gianni Di Caro. Ant colony optimization: a new meta-heuristic. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, volume 2, pages 1470–1477. IEEE, 1999.

[43] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[44] Xin-She Yang. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, pages 169–178. Springer, 2009.

[45] Xin-She Yang and Suash Deb. Cuckoo search via lévy flights. In *2009 World congress on nature & biologically inspired computing (NaBIC)*, pages 210–214. Ieee, 2009.

[46] Dervis Karaboga. Artificial bee colony algorithm. *scholarpedia*, 5(3):6915, 2010.

[47] Xin-She Yang and Amir Hossein Gandomi. Bat algorithm: a novel approach for global engineering optimization. *Engineering computations*, 2012.

[48] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

[49] Seyedali Mirjalili. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 89:228–249, 2015.

[50] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in engineering software*, 95:51–67, 2016.

[51] Seyedali Mirjalili. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural computing and applications*, 27(4):1053–1073, 2016.

[52] Juliano Pierezan and Leandro Dos Santos Coelho. Coyote optimization algorithm: a new metaheuristic for global optimization problems. In *2018 IEEE congress on evolutionary computation (CEC)*, pages 1–8. IEEE, 2018.

[53] Fatma A Hashim, Essam H Houssein, Kashif Hussain, Mai S Mabrouk, and Walid Al-Atabany. Honey badger algorithm: New metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation*, 192:84–110, 2022.

[54] Malik Braik, Abdelaziz Hammouri, Jaffar Atwan, Mohammed Azmi Al-Betar, and Mohammed A Awadallah. White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems*, 243:108457, 2022.

[55] Fatma A Hashim and Abdelazim G Hussien. Snake optimizer: A novel meta-heuristic optimization algorithm. *Knowledge-Based Systems*, 242:108320, 2022.

[56] Laith Abualigah, Dalia Yousri, Mohamed Abd Elaziz, Ahmed A Ewees, Mohammed AA Al-Qaness, and Amir H Gandomi. Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, 157:107250, 2021.

[57] Benyamin Abdollahzadeh, Farhad Soleimanian Gharehchopogh, and Seyedali Mirjalili. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Computers & Industrial Engineering*, 158:107408, 2021.

[58] Richard A Formato. Central force optimization. *Prog Electromagn Res*, 77(1):425–491, 2007.

[59] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009.

[60] Osman K Erol and Ibrahim Eksin. A new optimization method: big bang–big crunch. *Advances in Engineering Software*, 37(2):106–111, 2006.

[61] A Kaveh, Mohammad A Motie Share, and M Moslehi. Magnetic charged system search: a new meta-heuristic algorithm for optimization. *Acta Mechanica*, 224(1):85–107, 2013.

[62] Hosein Abedinpourshotorban, Siti Mariyam Shamsuddin, Zahra Beheshti, and Dayang NA Jawawi. Electromagnetic field optimization: a physics-inspired

metaheuristic optimization algorithm. *Swarm and Evolutionary Computation*, 26:8–22, 2016.

[63] A Kaveh and T Bakhshpoori. Water evaporation optimization: a novel physically inspired optimization algorithm. *Computers & Structures*, 167:69–85, 2016.

[64] Ali Husseinzadeh Kashan. A new metaheuristic for optimization: optics inspired optimization (oio). *Computers & Operations Research*, 55:99–125, 2015.

[65] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Abdolreza Hatamlou. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27(2):495–513, 2016.

[66] A Kaveh and Armin Dadras. A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Advances in Engineering Software*, 110:69–84, 2017.

[67] Alexandros Tzanetos and Georgios Dounias. A new metaheuristic method for optimization: sonar inspired optimization. In *International Conference on Engineering Applications of Neural Networks*, pages 417–428. Springer, 2017.

[68] A Kaveh and M Ilchi Ghazaan. A new meta-heuristic algorithm: vibrating particles system. *Scientia Iranica. Transaction A, Civil Engineering*, 24(2):551, 2017.

[69] Fatma A Hashim, Essam H Houssein, Mai S Mabrouk, Walid Al-Atabany, and Seyedali Mirjalili. Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems*, 101:646–667, 2019.

[70] Hossein Karami, Mohammad Javad Sanjari, and Gevorg B Gharehpetian. Hyper-spherical search (hss) algorithm: a novel meta-heuristic algorithm to optimize nonlinear functions. *Neural Computing and Applications*, 25(6):1455–1465, 2014.

[71] Rasoul Rahmani and Rubiyah Yusof. A new simple, fast and efficient algorithm for global optimization over continuous search-space problems: radial movement optimization. *Applied Mathematics and Computation*, 248:287–300, 2014.

[72] Hamid Salimi. Stochastic fractal search: a powerful metaheuristic algorithm. *Knowledge-Based Systems*, 75:1–18, 2015.

[73] Amin Foroughi Nematollahi, Abolfazl Rahiminejad, and Behrooz Vahidi. A novel meta-heuristic optimization method based on golden ratio in nature. *Soft Computing*, 24(2):1117–1151, 2020.

[74] Seyedali Mirjalili. Sca: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96:120–133, 2016.

[75] Xin-She Yang. Harmony search as a metaheuristic algorithm. In *Music-inspired harmony search algorithm*, pages 1–14. Springer, 2009.

[76] Esmaeil Atashpaz-Gargari and Caro Lucas. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *2007 IEEE congress on evolutionary computation*, pages 4661–4667. Ieee, 2007.

[77] Ying Tan and Yuanchun Zhu. Fireworks algorithm for optimization. In *International conference in swarm intelligence*, pages 355–364. Springer, 2010.

[78] R Venkata Rao, Vimal J Savsani, and DP Vakharia. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-aided design*, 43(3):303–315, 2011.

[79] Elyas Fadakar and Masoud Ebrahimi. A new metaheuristic football game inspired algorithm. In *2016 1st conference on swarm intelligence and evolutionary computation (CSIEC)*, pages 6–11. IEEE, 2016.

[80] Stephanie Forrest, Steven A Hofmeyr, and Anil Somayaji. Computer immunology. *Communications of the ACM*, 40(10):88–96, 1997.

[81] Jon Timmis, Mark Neal, and John Hunt. An artificial immune system for data analysis. *Biosystems*, 55(1-3):143–150, 2000.

[82] Mark Neal. Meta-stable memory in an artificial immune network. In *International Conference on Artificial Immune Systems*, pages 168–180. Springer, 2003.

[83] Raul Galvan-Correa, Mauricio Olguin-Carbajal, Juan Carlos Herrera-Lozada, Jacobo Sandoval-Gutierrez, José Félix Serrano-Talamantes, Rodrigo Cadena-Martinez, and Carlos Aquino-Ruíz. Micro artificial immune system for traffic light control. *Applied Sciences*, 10(21):7933, 2020.

[84] Selcuk Aslan and Sercan Demirci. Immune plasma algorithm: a novel meta-heuristic for optimization problems. *Ieee Access*, 8:220227–220245, 2020.

[85] Kevin M Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE control systems magazine*, 22(3):52–67, 2002.

[86] Julie Greensmith and Uwe Aickelin. Artificial dendritic cells: multi-faceted perspectives. In *Human-Centric Information Processing Through Granular Modelling*, pages 375–395. Springer, 2009.

[87] Amir Hossein Gandomi and Amir Hossein Alavi. Krill herd: a new bio-inspired optimization algorithm. *Communications in nonlinear science and numerical simulation*, 17(12):4831–4845, 2012.

[88] Mohammed Azmi Al-Betar, Zaid Abdi Alkareem Alyasseri, Mohammed A Awadallah, and Iyad Abu Doush. Coronavirus herd immunity optimizer (chio). *Neural Computing and Applications*, 33(10):5011–5042, 2021.

[89] Xin-She Yang. Flower pollination algorithm for global optimization. In *International conference on unconventional computing and natural computation*, pages 240–249. Springer, 2012.

[90] Xiangbo Qi, Yunlong Zhu, Hanning Chen, Dingyi Zhang, and Ben Niu. An idea based on plant root growth for numerical optimization. In *International Conference on Intelligent Computing*, pages 571–578. Springer, 2013.

[91] Ziqiang Zhao, Zhihua Cui, Jianchao Zeng, and Xiaoguang Yue. Artificial plant optimization algorithm for constrained optimization problems. In *2011 second international conference on innovations in bio-inspired computing and applications*, pages 120–123. IEEE, 2011.

[92] Ali Karci. Human being properties of saplings growing up algorithm. In *2007 IEEE International Conference on Computational Cybernetics*, pages 227–232. IEEE, 2007.

[93] Edoardo Amaldi, Antonio Capone, Matteo Cesana, Ilario Filippini, and Federico Malucelli. Optimization models and methods for planning wireless mesh networks. *Computer Networks*, 52(11):2159–2171, 2008.

[94] Fatos Xhafa, Admir Barolli, Christian Sánchez, and Leonard Barolli. A simulated annealing algorithm for router nodes placement problem in wireless mesh networks. *Simulation Modelling Practice and Theory*, 19(10):2276–2284, 2011.

[95] Fatos Xhafa, Christian Sánchez, and Leonard Barolli. Local search methods for efficient router nodes placement in wireless mesh networks. *Journal of Intelligent Manufacturing*, 23(4):1293–1303, 2012.

[96] Fatos Xhafa, Christian Sánchez, Admir Barolli, and Makoto Takizawa. Solving mesh router nodes placement problem in wireless mesh networks by tabu search algorithm. *Journal of Computer and System Sciences*, 81(8):1417–1428, 2015.

[97] Nabil Abdelkader Nouri, Zibouda Aliouat, Abdenacer Naouri, and Soufiene Ali Hassak. Accelerated pso algorithm applied to clients coverage and routers connectivity in wireless mesh networks. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–15, 2021.

[98] Lamri Sayad, Louiza Bouallouche-Medjkoune, and Djamil Aissani. A chemical reaction algorithm to solve the router node placement in wireless mesh networks. *Mobile Networks and Applications*, pages 1–14, 2017.

[99] Lamri Sayad, Djamil Aissani, and Louiza Bouallouche-Medjkoune. Placement optimization of wireless mesh routers using firefly optimization algorithm. In *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, pages 144–148. IEEE, 2018.

[100] Lamri Sayad, Louiza Bouallouche-Medjkoune, and Djamil Aissani. An electromagnetism-like mechanism algorithm for the router node placement in wireless mesh networks. *Soft Computing*, 23(12):4407–4419, 2019.

[101] Fatos Xhafa, Christian Sanchez, Leonard Barolli, and Evjola Spaho. Evaluation of genetic algorithms for mesh router nodes placement in wireless mesh networks. *Journal of Ambient Intelligence and Humanized Computing*, 1(4):271–282, 2010.

[102] Liang Tang, Zhichao Wang, Jun Huang, and Li Bian. A general purpose deployment method for wireless mesh network. *International Journal of Internet Protocol Technology*, 12(1):44–50, 2019.

[103] Chun-Cheng Lin. Dynamic router node placement in wireless mesh networks: A pso approach with constriction coefficient and its convergence analysis. *Information Sciences*, 232:294–308, 2013.

[104] Chun-Cheng Lin, Yan-Sing Li, and Der-Jiunn Deng. A bat-inspired algorithm for router node placement with weighted clients in wireless mesh networks. In *9th International Conference on Communications and Networking in China*, pages 139–143. IEEE, 2014.

[105] Chun-Cheng Lin, Pei-Tsung Tseng, Ting-Yu Wu, and Der-Jiunn Deng. Social-aware dynamic router node placement in wireless mesh networks. *Wireless Networks*, 22(4):1235–1250, 2016.

[106] Lamri Sayad, Louiza Bouallouche-Medjkoune, and Djamil Aissani. A simulated annealing algorithm for the placement of dynamic mesh routers in a wireless mesh network with mobile clients. *Internet Technology Letters*, 1(5):e35, 2018.

[107] Giuseppe De Marco. Mogamesh: A multi-objective algorithm for node placement in wireless mesh networks based on genetic algorithms. In *2009 6th International Symposium on Wireless Communication Systems*, pages 388–392. IEEE, 2009.

[108] Oumarou Mamadou Bello and Kolyang Dina Taiwe. Mesh node placement in wireless mesh network based on multiobjective evolutionary metaheuristic. *International Journal of Autonomic Computing*, 2(3):231–254, 2017.

[109] Alexei V Chechkin, Ralf Metzler, Joseph Klafter, Vsevolod Yu Gonchar, et al. Introduction to the theory of lévy flights. *Anomalous transport*, 129, 2008.

[110] Yassine Meraihi, Dalila Acheli, and Amar Ramdane-Cherif. Qos multicast routing for wireless mesh network based on a modified binary bat algorithm. *Neural Computing and Applications*, 31(7):3057–3073, 2019.

[111] Shahrzad Saremi, Seyedali Mirjalili, and Andrew Lewis. Biogeography-based optimisation with chaos. *Neural Computing and Applications*, 25(5):1077–1097, 2014.

[112] Ali Mansouri and Xingyuan Wang. A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme. *Information Sciences*, 520:46–62, 2020.

[113] Hamid R Tizhoosh. Opposition-based learning: a new scheme for machine intelligence. In *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*, volume 1, pages 695–701. IEEE, 2005.

[114] Tetsuya Oda, Donald Elmazi, Admir Barolli, Shinji Sakamoto, Leonard Barolli, and Fatos Xhafa. A genetic algorithm-based system for wireless mesh networks: analysis of system data considering different routing protocols and architectures. *Soft Computing*, 20(7):2627–2640, 2016.

[115] Shinji Sakamoto, Kosuke Ozera, Tetsuya Oda, Makoto Ikeda, and Leonard Barolli. Performance evaluation of wmn-psohc and wmn-pso simulation systems for node placement in wireless mesh networks: a comparison study. In *International Conference on Emerging Internetworking, Data & Web Technologies*, pages 64–74. Springer, 2017.

[116] Shinji Sakamoto, Yi Liu, Leonard Barolli, and Shusuke Okamoto. Performance evaluation of cm and riwm router replacement methods for wmns by wmn-psohc hybrid intelligent simulation system considering chi-square distribution of mesh clients. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 179–187. Springer, 2021.

[117] Admir Barolli, Kevin Bylykbashi, Ermioni Qafzezi, Shinji Sakamoto, Leonard Barolli, and Makoto Takizawa. A comparison study of chi-square and uniform distributions of mesh clients for different router replacement methods using wmn-psodga hybrid intelligent simulation system. *Journal of High Speed Networks*, (Preprint):1–16, 2021.

[118] Shinji Sakamoto, Admir Barolli, Leonald Barolli, and Makoto Takizawa. Design and implementation of a hybrid intelligent system based on particle swarm optimization, hill climbing and distributed genetic algorithm for node placement problem in wmns: a comparison study. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 678–685. IEEE, 2018.

[119] Shinji Sakamoto, Kosuke Ozera, Admir Barolli, Makoto Ikeda, Leonard Barolli, and Makoto Takizawa. Implementation of an intelligent hybrid simulation systems for wmns based on particle swarm optimization and simulated annealing: performance evaluation for different replacement methods. *Soft Computing*, 23(9):3029–3035, 2019.

[120] Sylia Mekhmoukh Taleb, Yassine Meraihi, Asma Benmessaoud Gabis, and Seyedali Mirjalili. A hybrid firefly algorithm and particle swarm optimization algorithm for mesh routers placement problem in wireless mesh networks. In *Proceedings of 7th International Conference on Harmony Search, Soft Computing and Applications*, pages 301–310. Springer, 2022.

[121] Admir Barolli, Shinji Sakamoto, Phudit Ampririt, Seiji Ohara, Leonard Barolli, and Makoto Takizawa. A comparison study of constriction and random inertia weight router replacement methods for wmns by wmn-psosa-dga hybrid simulation system considering chi-square distribution of mesh clients. In *International Conference on Broadband and Wireless Computing, Communication and Applications*, pages 11–21. Springer, 2020.

[122] Hui Wang, Zhijian Wu, Yong Liu, Jing Wang, Dazhi Jiang, and Lili Chen. Space transformation search: a new evolutionary technique. In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, pages 537–544. 2009.

[123] Jing Wang, Zhijian Wu, and Hui Wang. Hybrid differential evolution algorithm with chaos and generalized opposition-based learning. In *International Symposium on Intelligence Computation and Applications*, pages 103–111. Springer, 2010.

[124] Hong Gang Xia, Qing Zhou Wang, and Li Qun Gao. Opposition-based improved harmony search algorithm solve unconstrained optimization problems. In *Applied Mechanics and Materials*, volume 365, pages 170–173. Trans Tech Publ, 2013.

[125] Liqaa Nawaf, Stuart M Allen, and Omer Rana. Internet transit access point placement and bandwidth allocation in wireless mesh networks. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1–8. IEEE, 2017.

[126] Qian Wang, Hongpeng Wang, Ming He, Haihua Chen, and Xiaoyang Zhang. Optimization deployment of wl-mesh network with base-station in nature environment. In *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 107–112. IEEE, 2019.

[127] Awadallah M Ahmed and Aisha Hassan A Hashim. A genetic approach for gateway

placement in wireless mesh networks. *International Journal of Computer Science and Network Security (IJCSNS)*, 15(7):11, 2015.

[128] Zeineb Lazrag, Monia Hamdi, and Mourad Zaied. Bi-objective ga for cost-effective and delay-aware gateway placement in wireless mesh networks. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 103–108. IEEE, 2017.

[129] Maolin Tang and Chien-An Chen. Wireless network gateway placement by evolutionary graph clustering. In *International Conference on Neural Information Processing*, pages 894–902. Springer, 2017.

[130] H Mostafa Ali, Saeed Ashrafinia, Jiangchuan Liu, and Daniel C Lee. Wireless mesh network planning using quantum inspired evolutionary algorithm. In *2011 IEEE Vehicular Technology Conference (VTC Fall)*, pages 1–5. IEEE, 2011.

[131] Vinh Trong Le, Nghia Huu Dinh, and Nhu Gia Nguyen. A novel pso-based algorithm for gateway placement in wireless mesh networks. In *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 41–45. IEEE, 2011.

[132] Wei Liu, Chang Xu, Zhao Tian, Dong-Kun Li, Guan-Zhong Lu, and Wei She. Research on gateway deployment for throughput optimization in wireless mesh networks. In *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–5. IEEE, 2019.

[133] Marwa Sharawi, Hossam M Zawbaa, and Eid Emary. Feature selection approach based on whale optimization algorithm. In *2017 Ninth international conference on advanced computational intelligence (ICACI)*, pages 163–168. IEEE, 2017.

[134] Majdi M Mafarja and Seyedali Mirjalili. Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*, 260:302–312, 2017.

[135] Mohammad Tubishat, Mohammad AM Abushariah, Norisma Idris, and Ibrahim Aljarah. Improved whale optimization algorithm for feature selection in arabic sentiment analysis. *Applied Intelligence*, 49(5):1688–1707, 2019.

[136] Thi-Kien Dao, Tien-Szu Pan, and Jeng-Shyang Pan. A multi-objective optimal mobile robot path planning based on whale optimization algorithm. In *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pages 337–342. IEEE, 2016.

[137] Jianfa Wu, Honglun Wang, Na Li, Peng Yao, Yu Huang, and Hemeng Yang. Path planning for solar-powered uav in urban environment. *Neurocomputing*, 275:2055–2065, 2018.

[138] S Vinoth Kumar, R Jayaparvathy, and BN Priyanka. Efficient path planning of auvs for container ship oil spill detection in coastal areas. *Ocean Engineering*, 217:107932, 2020.

[139] Zheping Yan, Jinzhong Zhang, Zewen Yang, and Jialing Tang. Two-dimensional optimal path planning for autonomous underwater vehicle using a whale optimization algorithm. *Concurrency and Computation: Practice and Experience*, 33(9):e6140, 2021.

[140] Ankit Chhillar and Ankur Choudhary. Mobile robot path planning based upon updated whale optimization algorithm. In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 684–691. IEEE, 2020.

[141] Jhila Nasiri and Farzin Modarres Khiyabani. A whale optimization algorithm (woa) approach for clustering. *Cogent Mathematics & Statistics*, 5(1):1483565, 2018.

[142] Biswa Mohan Sahoo, Hari Mohan Pandey, and Tarachand Amgoth. A whale optimization (woa): Meta-heuristic based energy improvement clustering in wireless sensor networks. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 649–654. IEEE, 2021.

[143] MM Kumar and Aparna Chaparala. Obc-woa: opposition-based chaotic whale optimization algorithm for energy efficient clustering in wireless sensor network. *intelligence*, 250(1), 2019.

[144] Samineh Nasrollahzadeh, Mohsen Maadani, and Mohammad Ali Pourmina. Optimal motion sensor placement in smart homes and intelligent environments using a hybrid woa-pso algorithm. *Journal of Reliable Intelligent Environments*, pages 1–13, 2021.

[145] Puja Singh and Shashi Prakash. Optical network unit placement in fiber-wireless (fiwi) access network by whale optimization algorithm. *Optical Fiber Technology*, 52:101965, 2019.

[146] Ammar Al-Moalmi, Juan Luo, Ahmad Salah, Kenli Li, and Luxiu Yin. A whale optimization system for energy-efficient container placement in data centers. *Expert Systems with Applications*, 164:113719, 2021.

[147] Khalid Mohammed Saffer Alzaidi, Oguz Bayat, and Osman N Uçan. Multiple dgs for reducing total power losses in radial distribution systems using hybrid woa-ssa algorithm. *International Journal of Photoenergy*, 2019, 2019.

[148] Hardi M Mohammed, Shahla U Umar, and Tarik A Rashid. A systematic and meta-analysis survey of whale optimization algorithm. *Computational intelligence and neuroscience*, 2019, 2019.

[149] Seyedali Mirjalili and Andrew Lewis. S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, 9:1–14, 2013.