

An Improved Nature Inspired Algorithm to Design Energy Efficient and Stability Aware Routing Protocol for Wireless Sensor Networks

A Thesis

submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Electronics and Communication Engineering Department

by

Supreet Singh

(Reg no: 951806003)

under the supervision of

Late Dr. Urvinder Singh (Professor)

Dr. Kulbir Singh (Professor)



THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Thapar Institute of Engineering and Technology, Patiala, India

July 2024

Declaration

I, Supreet Singh, hereby certify that the thesis entitled "An Improved Nature Inspired Algorithm to Design Energy Efficient and Stability Aware Routing Protocol for Wireless Sensor Networks", submitted to Thapar Institute of Engineering and Technology, Patiala, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electronics and Communication Engineering is an authentic record of my independent and original research work carried out during the period January 2019 to February 2024 under the supervision of Late Dr. Urvinder Singh and Dr. Kulbir Singh. I have cited the due references about any text/figure/table which have been added from their original source.

The work presented in this thesis has not been used for the award of any degree/diploma or some other title from any institution.

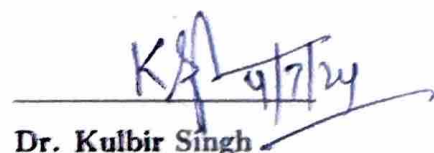


Supreet Singh

Candidate

Date: 04/07/2024,

I hereby declare that the above statement made by the candidate is true to the best of my knowledge.



Dr. Kulbir Singh

Professor (Administrative Supervisor)

ECED

TIET, Patiala.

*In loving memory of
my revered mentor
(The fountain of inspiration)*

Abstract

Nature-inspired algorithms represent a category of highly potent optimization techniques that draw inspiration from natural processes. They have found applications across a broad spectrum of research disciplines. These algorithms are characterized by their simplicity, adaptability, and flexibility, making them suitable for various problem-solving scenarios. These algorithms have existed for generations but have garnered significant interest from researchers over the past two decades. In this thesis, several nature-inspired algorithms have been studied, including CS, GWO, NMRA, SMA, WOA, COOT optimization, and others. These algorithms are recognized as state-of-the-art and have demonstrated their competitiveness and applicability across various research domains. In addition to their advantages, these algorithms exhibit certain drawbacks, including algorithm trapping in local optimal solutions, limited exploration capabilities, slow convergence rates, and complex parameter tuning. To overcome these limitations, it is imperative to design hybrid or enhanced versions of algorithms capable of effectively addressing high-end optimization problems. It has become evident that no single algorithm is universally suited for all optimization challenges. Therefore, the central objective of this work is to introduce an algorithm capable of successfully solving standard IEEE CEC benchmark problems and subjected to various dimensional and statistical analysis to validate results. In this work, a number of hybrid algorithms have been developed inspired from the exploitation capability of NMRA with added features of self adaptability and stagnation avoidance phase.

The widespread use of wireless sensor devices and their advancements in terms of size, deployment cost, measurement of environmental events and user friendly interface have given rise to many applications of WSNs. Such networks need to utilize routing protocols to forward data samples from event regions to sink. The focus of such routing protocols is to forward collected data samples via minimum cost links (in terms of energy consumption and time). Clustering is an efficient data aggregation method that effectively reduces the energy consumption by organizing nodes into groups (clusters). The proposed algorithms have been applied to create cluster-based routing protocols for WSNs. Most previous work on WSNs has focused on minimizing energy consumption during data extraction processes. This emphasis is because SNs operate on batteries that deplete rapidly with each operation. Therefore, proper CH selection and their load balancing using efficient routing protocol is a critical aspect for the long run operation of WSN. In light of this, the research developed nature-inspired algorithm-based clustered routing protocols and minimized the network's energy resource consumption, extending its overall lifetime with consideration for static and mobile SNs.

Acknowledgement

First and foremost, I am highly obliged and thankful to Almighty GOD for blessing me with courage, confidence and grit to carry out the research work. HE gave me strength in hard times and made the way towards pursuing this degree.

I am grateful to our Hon'ble Director, **Prof. Padmakumar Nair** and **Dr. N. Tejo Prakash**, Professor and Dean (Research and Development Cell), for providing me with the opportunity to undertake the Ph.D.

I gratefully acknowledge my Ph.D. supervisor, **Late Dr. Urvinder Singh** and **Dr. Kulbir Singh**, for his invaluable support and guidance throughout my Ph.D. program. I will be forever indebted to his innumerable insights and encouragement in helping me with my research. It is my privilege to have learned from his research enthusiasm.

I am grateful to my Doctoral committee members **Dr. Kulbir Singh**, **Dr. Sanjay Sharma**, **Dr. Neeru Jindal**, and **Dr. Rajiv Kumar** for contributing insightful comments in examining my work.

My heartfelt gratitude to all of my family members who always support me through thick and thin. Their love, understanding and persistent encouragement helped me complete my research successfully.

Supreet Singh

Table of Contents

Title	Page No.
Declaration	ii
Dedication	iii
Abstract	iv
Acknowledgement	v
Table of Contents	vi
List of Figures	xi
List of Tables	xiii
List of Abbreviations	xv
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Nature-inspired optimization algorithms	2
1.2.1 Evolutionary algorithms	3
1.2.2 Swarm intelligent algorithms	3
1.2.3 Physics or chemistry-based algorithms	3
1.3 Application of optimization algorithms	4
1.4 Wireless sensor networks	5
1.4.1 WSN Architecture	6
1.4.2 Routing techniques in WSNs	6
1.4.3 Mobile WSNs	8
1.5 Research question	9
1.5.1 Research objectives	10
1.5.2 Significance of the proposed research	10
1.6 Primary thesis contribution	12
1.7 Organisation of Thesis	13

Chapter 2	Literature Review	14
2.1	Numerical optimization	14
2.1.1	Fundamental concepts	14
2.1.2	Test Suite	15
2.2	Optimization algorithms	16
2.2.1	Genetic algorithm	17
2.2.2	Differential evolution	17
2.2.3	Particle swarm optimization	17
2.2.4	Cuckoo search	18
2.2.5	Marine predator algorithm	19
2.3	Algorithms used for proposed research	19
2.3.1	Naked mole-rat algorithm	19
2.3.2	Slime mould algorithm	22
2.3.3	Whale optimization algorithm	25
2.3.4	COOT optimization algorithm	28
2.4	Routing protocols for WSNs	31
2.5	Nature inspired optimization algorithms based routing protocols for WSNs	33
2.6	Research gaps	34
2.7	Inferences and summary	35
Chapter 3	The Effect of Chaotic Mapping on Naked Mole-rat Algorithm for Energy-efficient WSN	37
3.1	Introduction	37
3.2	Chaotic maps	39
3.3	Chaotic naked mole-rat algorithm	41
3.3.1	Computational efficiency	42
3.4	Result analysis for test suites	43
3.4.1	Test suite	43
3.4.2	Parameter settings	44
3.4.3	Effect of breeding probability	45
3.4.4	Effect of population size	46
3.4.5	Effect of dimension Size	47
3.4.6	Comparison of best proposed NMRA variant concerning other algorithms	50
3.4.7	Convergence profiles	51

3.4.8	Comparison on CEC2014 test suite	52
3.4.9	Comparison on 100-digit challenge benchmark functions	53
3.5	Real-world engineering design optimization problems	54
3.5.1	Pressure vessel design optimization problem	57
3.5.2	Welded beam design optimization problem	58
3.5.3	Tension/compression spring design optimization problem	59
3.6	Energy efficient wireless sensor network optimization problem	61
3.7	Summary	63

Chapter 4 A Novel Self-adaptive Hybrid Slime Mould Naked Mole-rat Algorithm for Numerical Optimization and Energy-efficient WSN 65

4.1	Introduction	65
4.2	Self-adaptive hybrid algorithm: SM-NMRA	68
4.2.1	Initialization of search agents	68
4.2.2	Worker phase	68
4.2.3	Breeder phase	69
4.2.4	Population adaptation	69
4.2.5	Selection phase	70
4.2.6	Parameters adaptation	70
4.2.7	Stagnation phase	71
4.3	Statistical results for numerical test suites	72
4.3.1	Performance evaluation for CEC 2005 test problems	72
4.3.2	Performance evaluation for 30 CEC 2014 test functions	78
4.4	Real-world optimization problem of cluster head selection in WSN	79
4.4.1	Radio energy model	81
4.4.2	Clustering protocol using SM-NMRA	82
4.4.3	Inferences from simulated results	86
4.5	Summary	91

Chapter 5 A Novel Whale Optimization based Naked Mole-rat Algorithm for Energy Efficient WSN 93

5.1	Introduction	93
5.2	Whale optimization based NMRA	95
5.2.1	Intialization	95
5.2.2	Exploration (worker phase)	95

5.2.3	Exploitation (breeder phase)	96
5.2.4	Parameter adaptation:	96
5.2.5	Greedy Selection	97
5.3	Results and discussion	97
5.3.1	Statistical results for CEC 2019 test suite	97
5.3.2	Real world optimization problem: Selection of cluster head in WSN	99
5.4	Summary	105

Chapter 6 An Energy-efficient Clustering Protocol using Novel Self-adaptive Hybrid COOT Optimization based Naked Mole-rat Algorithm for WSN . 107

6.1	Introduction	107
6.2	Hybrid COOT optimization based NMRA	109
6.2.1	Initialization of search agents	110
6.2.2	Exploration phase	111
6.2.3	Exploitation phase	112
6.2.4	Greedy selection	112
6.2.5	Computational complexity	113
6.3	Statistical testing of CONMRA for CEC 2019 benchmark numerical test problems	114
6.4	Proposed clustering protocol using CONMRA for WSN	115
6.4.1	Cluster head election	115
6.4.2	Cluster setup	118
6.4.3	Development of TDMA schedule	119
6.4.4	Evaluation of fitness	119
6.5	Simulation results of clustering protocols for WSN	122
6.6	Summary	125

Chapter 7 A Self-adaptive Attraction and Repulsion-based Naked Mole-Rat Algorithm for Energy-efficient Mobile WSN 128

7.1	Introduction	128
7.2	Attraction and repulsion based NMRA	130
7.2.1	Adoption of attraction and repulsion strategy	130
7.2.2	Parameter adaptation	131
7.3	Results and discussion	131
7.3.1	Statistical results for CEC 2005 numerical benchmark problems	132
7.3.2	Statistical results for CEC 2019 numerical benchmark test suite	133

7.4	Real Time Application: Energy-efficient ARNMRA based routing protocol (EARNRP) for MWSNs	135
7.4.1	Radio model	135
7.4.2	Random waypoint mobility model	137
7.4.3	System assumptions	138
7.4.4	Protocol description	139
7.4.5	CH election in proposed EARNRP protocol	139
7.4.6	Operation of proposed protocol: EARNRP	142
7.4.7	Analysis of results	144
7.5	Summary	148
Chapter 8	Conclusion and Future Work	149
8.1	Conclusion	149
8.2	Future Work	151
	List of Publications	152
	Appendix A	177

List of Figures

Figure No.	Title	Page No.
1.1	Architecture of WSN node	5
1.2	Clustered WSN	7
1.3	Categorization of routing protocols in WSNs	8
3.1	Convergence profiles for unimodal (F1 to F7) and multimodal (F8 to F12) benchmark functions of CEC 2005 test suite	52
3.2	Convergence profiles and box plots of NMRA and CNMRA for 100-digit benchmark functions (CEC 2019)	55
3.2	Convergence profiles and box plots of NMRA and CNMRA for 100-digit benchmark functions (CEC 2019) (continued)	56
3.2	Convergence profiles and box plots of NMRA and CNMRA for 100-digit benchmark functions (CEC 2019) (continued)	59
3.3	Pressure vessel design problem	59
3.4	Welded beam design problem	60
3.5	Design of spring problem	60
4.1	Convergence profiles of NMRA, SMA and SM-NMRA for 12 CEC 2005 numerical test functions	78
4.2	Radio energy dissipation model	82
4.3	Selection of CH using SM-NMRA and exchanging information between sensor nodes and BS in setup phase	83
4.4	Transmission of packets from sensor nodes to CHs in intra-cluster communication . .	85
4.5	Transmission of packets from CHs to BS in inter-cluster communication	85
4.6	Count of alive nodes for homogeneous setup using $E_0 = 1J$ with progress of rounds . .	88
4.7	Average energy over rounds for homogeneous setup using $E_0 = 1J$	89
4.8	Count of alive nodes for heterogeneous setup using $E_0 = 1J$ with progress of rounds .	91
4.9	Average energy over rounds for heterogeneous setup using $E_0 = 1J$	92
5.1	Convergence graphs of WOA, NMRA and WONMRA for 100-digit benchmark functions (CEC 2019)	102

5.2	Box-plots of WOA, NMRA and WONMRA for 100-digit benchmark functions (CEC 2019)	104
6.1	Flow chart of proposed CONMRA	113
6.2	Convergence profiles of DE, COOT, NMRA and CONMRA for 100-digit (CEC 2019) numerical test problems	118
6.3	CH using CONMRA and exchange information between SNs and BS in setup phase .	118
6.4	Radio energy dissipation model	121
6.5	(a) 100 nodes network model (b) set of nodes elected as CHs at time t (c) a new set of sensor nodes elect themselves as CHs at time $t + d$	123
6.6	Count of alive nodes for homogeneous setup using $E_0 = 1J$ with progress of rounds . .	124
6.7	Average energy over rounds for homogeneous setup with $E_0 = 1J$	125
6.8	Count of alive nodes for heterogeneous setup using $E_0 = 1J$ with progress of rounds .	126
6.9	Average energy over rounds for heterogeneous setup using $E_0 = 1J$	127
7.1	Convergence profiles of NMRA and ARNMRA for CEC 2005 numerical benchmark problems	134
7.2	Random waypoint model	138
7.3	For EARNRP, new adaptive TDMA scheduling creation	143
7.4	Number of alive nodes as a function of communication rounds for EARNRP in the presence and absence of mobility.	145
7.5	Average energy as a function of communication rounds for EARNRP in the presence and absence of mobility.	146
7.6	Network lifetime vs Percentage of mobile nodes for varying speed	147
7.7	Network lifetime vs Speed (m/s) for 50% node mobility.	147
7.8	Network lifetime vs Speed (m/s) for 100% node mobility.	148

List of Tables

Table No.	Title	Page No.
3.1	Parameter settings	45
3.2	NMRA simulation results for different breeding probability values for CEC 2005 ($D = 30$)	46
3.3	Simulated outcomes of chaotic variants for population size (30, 60, 90)	48
3.4	Outcomes of chaotic variants for dimension size (10, 30, 50)	49
3.5	Simulated outcomes of CNMRA as compared to competitive algorithms for CEC 2005 test suite	51
3.6	Simulated outcomes of CNMRA as compared to other algorithms for CEC 2014 test suite	54
3.6	Simulated outcomes of CNMRA as compared to other algorithms for CEC 2014 test suite (Continued)	57
3.7	Simulated outcomes for CEC 2019 test functions	58
3.8	Simulation results for three engineering design optimization problems	60
3.9	Comparative analysis of various algorithms for network lifetime estimation in WSN .	63
4.1	Simulated outcomes for dimension size of 10, 50, 100, 200, 500	75
4.2	Comparative analysis of proposed SM-NMRA for 12 CEC 2005 numerical test problems	77
4.3	Simulated outcomes of SM-NMRA for CEC 2014 test problems	80
4.3	Simulated outcomes of SM-NMRA for CEC 2014 test problems (continued)	81
4.4	Parameters selection for setup of WSN	87
4.5	Comparison of network lifespan for homogeneous configuration with $E_0=0.25J$ in terms of dead nodes round history	87
4.6	Comparison of network lifespan for homogeneous configuration with $E_0=0.5J$ in terms of dead nodes round history	88
4.7	Comparison of network lifespan for homogeneous configuration with $E_0=1J$ in terms of dead nodes round history	88
4.8	Comparison of network lifespan for heterogeneous configuration with $E_0=0.25J$ in terms of dead nodes round history	90
4.9	Comparison of network lifespan for heterogeneous configuration with $E_0=0.5J$ in terms of dead nodes round history	90

4.10	Comparison of network lifespan for heterogeneous configuration with $E_0=1J$ in terms of dead nodes round history	90
5.1	Simulated outcomes for CEC 2019 test suite	100
5.2	Comparative analysis of various algorithms for network lifetime estimation in WSN .	105
6.1	Simulated outcomes for CEC 2019 test suite	116
6.2	Characteristics for setup of WSN	122
6.3	Simulation parameters for clustering protocols	122
6.4	Network lifespan analysis for homogeneous configuration with $E_0=1J$ in terms of dead nodes round history	123
6.5	Network lifespan analysis for heterogeneous configuration with $E_0=1J$ in terms of dead nodes round history	124
6.6	Network lifespan of routing protocols for $E_0 = 1J$ with stability and instability periods	126
7.1	Simulated outcomes of ARNMRA for CEC 2005 test suite	133
7.2	Simulated outcomes for CEC 2019 test suite	135
7.3	Network lifespan comparison for homogeneous configuration with $E_0=1J$ in terms of dead nodes round history	147

List of Abbreviations

NIA	Nature Inspired Algorithms
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
DE	Differential Evolution
CS	Cuckoo Search
EP	Evolutionary Programming
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
BBO	Biogeography-Based Optimizer
BSA	Backtracking Search Optimization Algorithm
ACO	Ant Colony Optimization
ABC	Artificial Bee Colony
CS	Cuckoo Search
FA	Firefly Algorithm
KH	Krill Herd
GWO	Grey Wolf Optimizer
SSA	Salp Swarm Algorithm
NMRA	Naked Mole-Rat Algorithm
SMA	Slime Mould Algorithm
WOA	Whale Optimization Algorithm
MFO	Moth Flame Optimization
SA	Simulated Annealing
MOA	Magnetic Optimization Algorithm
GSA	Gravitational Search Algorithm
CRO	Chemical Reaction Optimization
BB-BC	Big-Bang Big-Crunch
HS	Harmony Search
IWD	Intelligent Water Drops
CSS	Charged System Search
VLSI	Very Large Scale Integration
WSNs	Wireless Sensor Networks
SNs	Sensor Nodes

CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
BS	Base Station
CHs	Cluster Heads
MWSNs	Mobile Wireless Sensor Networks
RWP	Random way point mobility model
RPGM	Reference point group mobility model
NP	Non-deterministic polynomial
MPA	Marine Predator Algorithm
NMRs	Naked mole rats
HHA	Harris Hawks Optimization
LEACH	Low Energy Adaptive Clustering Hierarchy
NEAP	Novel Energy Adaptive Protocol
TEEN	Threshold-sensitive Energy Efficient sensor Network
SEP	Stable Election Protocol
TSEP	Threshold Sensitive Stable Election Protocol
DEEC	Distributed Energy-Efficient Clustering
DEC	Deterministic Energy-efficient Clustering
CBR	Cluster-Based Routing
TDMA	Time-Division Multiple Access
MBC	Mobility-Based Clustering
MN	Mobile Node
GCA	Genetic Clustering Algorithm
EAERP	Energy-aware Evolutionary Routing Protocol
SMO	Spider Monkey Optimization
FPA	Flower Pollination Algorithm
ISSA	Improved Squirrel Search Algorithm
NFL	No Free Lunch
OEWOA	Opposition and Exponential Properties based Whale Optimization Algorithm
JADE	Differential Evolution with Adaptive Properties
SaDE	Self-adaptive Differential Evolution
SHADE	Success-history based Differential Evolution
GWO-E	Grey Wolf Optimizer Enhanced
SCCSA	Sine Cosine Crow Search Algorithm

FA-FPO	Fractional Order Flower Pollination Algorithm
GSSA	Gaussian-Salp Swarm Algorithm
RW-GWO	Random Walk Grey Wolf Optimizer
B-BBO	Blended Biogeography-based Optimization
LX-BBO	Laplacian Biogeography-based Optimization
PBIL	Population-based Incremental Learning
MGSCA	Memory Guided Sine Cosine Algorithm
ISOS	Improved Symbiotic Organisms Search
VNBA	Variable Neighbourhood Bat Algorithm
IMEHO	Improved Elephant Herding Optimization
CCS	Chaotic Cuckoo Search
BDE	Beta Differential Evolution
NBA	Novel Bat Algorithm
DA	Dragonfly Algorithm
TLBO	teaching Learning-based Optimization
LebTLBO	Learning Enthusiasm-based TLBO
mGWO	Modified Grey Wolf Optimizer
BA	Bat Algorithm
QoS	Quality of Service
OSI	Open Systems Interconnection
IoT	Internet of Things
CMs	Cluster Members
SNR	Signal-to-Noise Ratio
HRPs	Hierarchical Routing Protocols
SEP-E	Stable Election Protocol Enhanced
ERP	Evolutionary Routing Protocol
SAERP	Stable Aware Evolutionary Routing Protocol
ASSA	Adaptive Salp Swarm Algorithm
HCR	Hierarchical Cluster-based Routing
SEECP	Stable Energy Efficient Clustering Protocol
SOGWO	Selective Operation based Grey Wolf Optimizer
OB-L-EO	Opposition Based Laplacian Equilibrium Optimizer
IWOA	Improved Whale Optimization Algorithm

Chapter 1

Introduction

This chapter presents the essential principles of optimization, NIA, and routing protocols in WSN required for the thesis proposal. First, it draws inspiration from prior research efforts and uses it to formulate new objectives. Subsequently, the chapter defines the primary contributions of the thesis. It delves into the specifics of the various proposed algorithms and how they are applied to real-world optimization problems. The final section provides an overview of the structure and organization of each chapter within the thesis.

1.1 Motivation

In the contemporary world, optimization has found a multitude of practical applications. Now a days, scientific fields and industrial enterprises encounter significant challenges when seeking suitable tools, particularly optimization algorithms, to tackle complex optimization problems. The main objective for the adoption of these algorithms is rooted in the fact that a substantial portion of real-world scenarios can be cast as optimization problems specific to particular domains. This propensity arises from the nature of the problems and the number of objectives requiring optimization.

This trend is underscored by the pervasive utilization of these algorithms across nearly every realm of research, encompassing operations research, signal processing, management, machine learning, and other domains. These optimization quandaries are notably intricate, which poses a formidable challenge for researchers aiming to devise efficient solutions.

Recently, algorithms inspired by natural processes have taken center stage in the landscape of optimization research. These NIA have gained prominence across various domains and are widely employed. These algorithms employ four core procedures to replicate natural behaviors: initialization, local search, global search and selection. Despite their impressive performance, experimental analysis have highlighted their sensitivity to tuning parameters [1]. Initially, it was discovered that employing a trial-and-error approach was necessary to fine-tune these parameters [2] effectively, a method notorious for its laboriousness. Nonetheless, this method has proven to be inefficient. These algorithms are further affected by premature convergence, stagnation at local optima, and sluggish convergence rates, as indicated by [3]. These constraints make the algorithms vulnerable to generate sub-optimal solutions, ultimately resulting in diminished diversity and an inability to achieve the

desired outcomes. Overall, it is evident that although NIA exhibit strengths, they are not exempt from challenges. As such, there is a pressing need for continued research in devising novel algorithms to address these concerns.

This thesis is dedicated to mitigating the aforementioned challenges by introducing improved versions of NIA through innovative strategies. The foundation for these new algorithms is built upon established optimization methodologies like NMRA [4], SMA [5], COOT [6], and others. The rationale behind selecting these algorithms lies in their uncomplicated yet potent search strategies, well-suited for addressing intricate optimization problems at the upper echelons of complexity. Motivated by these insights, the central objective is to elevate the performance of these algorithms, enabling them to tackle a broad spectrum of intricate and demanding optimization issues encompassing real-world scenarios. The IEEE CEC benchmarks, a widely accepted evaluation tool, have been employed to assess their effectiveness. Researchers have consistently turned to these benchmarks to appraise their algorithmic solutions, underlining their credibility [7] [8] [9].

Numerous algorithmic variations stemming from the concepts presented in this thesis have already demonstrated their prowess in addressing these challenges, further underscoring the practicality and relevance of NIA in tackling a diverse range of complex optimization problems. Various sets of CEC benchmark suites have been employed within the realm of single-objective real-world numerical optimization and energy-efficient clustering protocol for WSNs to validate the proposed algorithms empirically. The overarching incentive is to devise novel algorithms that possess the potential to outperform previously reported outcomes in existing literature.

1.2 Nature-inspired optimization algorithms

Traditional optimization techniques, encompassing both single-variable and multivariate approaches, were once the go-to methods for optimization. However, as problems grew more complex, these techniques proved inadequate in providing viable solutions. This spurred the development of novel meta-heuristic techniques designed to replace conventional methods and yield approximate solutions. Many of these meta-heuristics draw inspiration from natural phenomena or the social behaviors observed in various species. The major examples of such techniques include GA [10], PSO [11], DE [12], CS [13], and others. These meta-heuristic algorithms, inspired by nature, have proven effective in addressing the challenges posed by optimization problems.

Categorically, NIA can be grouped based on their sources of inspiration: swarm intelligence, biological systems, and physical or chemical systems. Accordingly, these algorithms can be referred to as bio-inspired, swarm-intelligence-based, or physics and chemistry-based, depending on the underlying

sources of their design.

1.2.1 Evolutionary algorithms

Evolutionary meta-heuristics draw inspiration from the processes of biological evolution in nature. These algorithms are rooted in the principle that species with greater fitness are more likely to survive and thrive, while weaker ones face a higher likelihood of perishing [14]. The classic evolutionary algorithm is GA [10], which emulates Darwin’s concept of ”survival of the fittest.” The foundation of evolutionary algorithms lies in initiating the optimization process with a randomly generated population of individual candidates, each representing a potential solution to the optimization problem. These individuals then undergo a process of evolution guided by mechanisms inspired by natural processes, including selection, reproduction, mutation, and recombination. As these algorithms operate, individuals with higher fitness are more inclined to reproduce and persist across generations. Consequently, the subsequent population is prone to exhibit improvement compared to its predecessors, increasing the likelihood of attaining an optimal or nearly-optimal solution. Some prominent evolutionary algorithms are DE [12], EP [15], CMA-ES [16], BBO [17] and BSA [18]. These algorithms collectively represent diverse approaches within the evolutionary meta-heuristic framework.

1.2.2 Swarm intelligent algorithms

Swarm intelligent algorithms emulate the collective and social intelligence of various species, such as birds, fishes and ants. Among these techniques, PSO is a widely known algorithm [11] and draws inspiration from the flocking behavior of birds. Another prevalent swarm intelligence-based optimization algorithm is ACO [19], which replicates the foraging conduct of certain ant species. Ants leave pheromone trails on the ground to discover optimal short paths, encouraging other colony members to follow these paths to locate food sources. These algorithms mimic the cooperation among numerous similar entities within an environment to accomplish desired objectives. Notable instances of other popular swarm intelligence-based algorithms include ABC [20], CS [13], FA [21], KH [22], GWO [23], SSA [24], NMRA [4], SMA [5], WOA [25] and COOT optimization [6].

1.2.3 Physics or chemistry-based algorithms

This meta-heuristic category comprises methods rooted in physics or chemistry, drawing inspiration from various chemical and physical processes. For example, SA [26] takes cues from the annealing of metals. This involves controlled heating and cooling to enhance crystal size and minimize defects. Another example is MOA [27], which derives its principles from the attractive and repulsive forces among magnetic particles. The GSA [28] is developed based on the laws of gravity, while the CRO

[29] is inspired by the interactions between molecules during a chemical reaction, seeking to achieve a state of low-energy stability. The BB-BC algorithm [30] finds its inspiration in the evolutionary process of the universe. Within this category, other well-regarded algorithms include HS [31], IWD [32], CSS [33], and others. These algorithms creatively mimic the behavior of physical and chemical phenomena to address optimization challenges.

1.3 Application of optimization algorithms

Optimization algorithms have become indispensable in the contemporary landscape, finding utility across virtually every domain [34]. These algorithms find diverse applications, some of which are outlined below:

- **Mechanical and civil engineering**

Optimization algorithms are pivotal in achieving optimal designs for various mechanical components, including linkages, cams, gears, machine tools, conveyors, trucks, and cranes. Moreover, they extend to solving differential equations for rigid body dynamics. In civil engineering, these algorithms are instrumental in efficiently designing structures like frames, foundations, bridges, towers, and dams, ensuring minimal costs.

- **Electrical and electronics engineering**

Optimization methodologies find application in electrical engineering, facilitating the design of electrical machinery encompassing motors, generators, transformers, and antenna configurations. Furthermore, these techniques are integral to optimizing electrical networks and semiconductor designs, including tasks like floor plan optimization in VLSI design.

- **Computer science**

These methodologies address many challenges, from tackling NP-Hard problems and convex optimization dilemmas to resolving p-center issues and other complexities.

- **Generalized applications**

It can be encompassed within operations research, where scientific methodologies are harnessed to attain optimal solutions. In the context of stochastic programming, it aids in formulating dynamic decisions that adapt to evolving events. In economics, these techniques prove invaluable for addressing utility maximization and expenditure minimization quandaries. Moreover, they are pertinent to various mathematical programming methods, encompassing linear programming, global optimization, convex optimization, quadratic programming, and more.

Apart from this, optimization techniques find extensive use in other diverse applications, including the optimal design of control systems, the configuration of chemical processing equipment, site selection for industrial facilities, efficient production planning, image processing [35], meticulous control and scheduling, signal processing, the development of routing protocols for WSNs, node localization in WSNs [36], machine learning, and revenue management. In essence, it can be inferred that optimization is an essential prerequisite for tackling most real-world problems.

1.4 Wireless sensor networks

WSNs function as a system of closely spaced nodes that can sense their environment, process data, perform computations, and communicate. The applications of WSNs are diverse, ranging from monitoring atmospheric conditions to facilitating video surveillance, essentially encompassing any task that necessitates collecting and transmitting information [37]. SNs employed within WSNs are characterized by limited energy resources, modest memory capacities, and relatively low processing capabilities. These compact and affordable devices have become feasible due to recent advancements in CMOS technology. Each sensor node is equipped with a modest CPU, limited memory storage, a radio module for signal reception and transmission, and a power supply unit [38]. The configuration of a WSN node's architecture is depicted in Figure 1.1. Despite their small size, these devices can autonomously establish a network comprising numerous nodes, all without requiring a dedicated infrastructure.

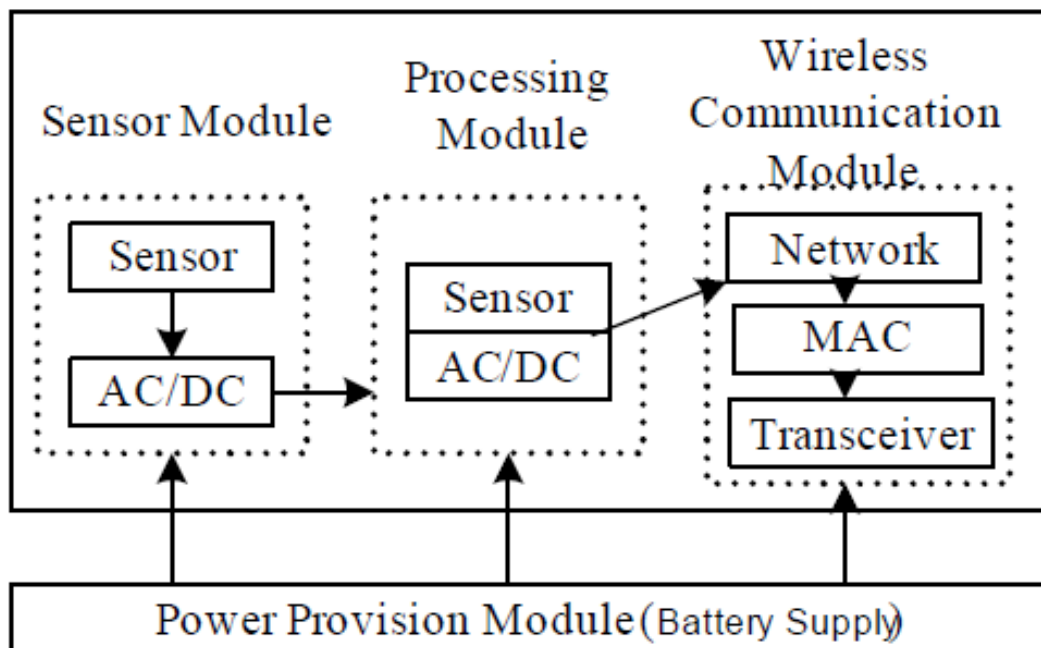


Figure 1.1: Architecture of WSN node

1.4.1 WSN Architecture

WSNs are commonly deployed in challenging environments with limited human access, such as battlefields, forests, and specialized industrial and medical settings. Consequently, WSNs must operate autonomously in a self-configuring manner, forming networks on an ad hoc basis. Given that WSNs have limited energy resources, it's noteworthy that communication consumes more energy compared to sensing and data processing [39]. Therefore, there is a pressing need for an architecture that minimizes transmissions to the BS and delegates decision-making to the individual nodes. As the network size and the quantity of nodes expand, scalability becomes a critical consideration in evaluating the feasibility of deploying the network. In this context, a hierarchical architecture demonstrates its effectiveness in improving network scalability. In a hierarchical architecture, the network is conceptually partitioned into virtual layers, where nodes within the same layer perform analogous functions [40]. One widely adopted technique for implementing this hierarchy is node clustering, a two-tiered strategy for organizing nodes into clusters

To efficiently distribute management responsibilities among the nodes, specific nodes are designated as CHs, entrusted with leading their respective clusters. Clustering leverages data correlations and alleviates network congestion by aggregating data, resulting in more efficient energy utilization. These CHs gather data from ordinary nodes within their clusters, undertake tasks such as data aggregation or fusion, and then relay the consolidated data to the BS using a multi-hop approach. A typical application of a clustered WSNs is depicted in Figure 1.2. As demonstrated in this illustration, nodes in Layer 1 (ordinary nodes) sense the environment, generate data, and transmit it to their assigned CHs. The CHs in the second layer receive this data, process it, and subsequently relay it to the BS via a multi-hop route. Ultimately, users access the data from the BS over the internet.

Numerous clustering strategies have been proposed for WSNs [41], with a primary shared goal of conserving energy. The tasks within clustering protocols are typically categorized into three stages: the election of CHs, cluster formation, and data transmission. Among these stages, the CH selection algorithm is the pivotal component of each approach or protocol, which fundamentally shapes the energy efficiency of the network. Beyond energy efficiency, certain clustering approaches may also focus on additional objectives, such as ensuring load distribution among nodes [41] and achieving complete coverage of the designated area.

1.4.2 Routing techniques in WSNs

Routing within WSNs presents unique challenges compared to other wireless networks such as mobile ad-hoc networks, and this complexity arises from various factors [41]:

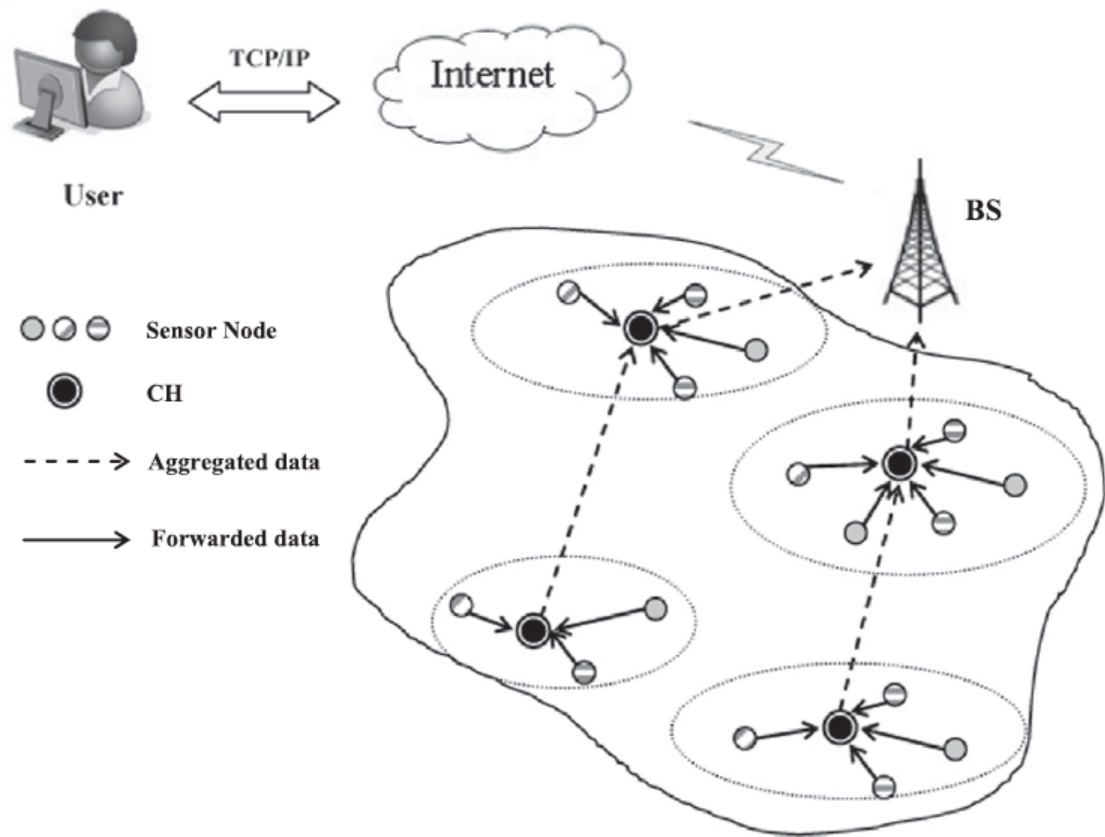


Figure 1.2: Clustered WSN

- **Resource constraints**

SNs in WSNs have strict limitations regarding energy, processing power, and storage capacity, demanding careful resource management.

- **Data convergence**

Most WSN applications require data from multiple sources to flow toward a specific BS.

- **Application specificity**

WSN design requirements vary based on the application, making them highly specialized.

- **Stationary nodes**

WSN nodes are typically stationary after deployment, leading to predictable and infrequent changes in network topology.

- **Location-based data collection**

Data collection in WSNs is often location-dependent, necessitating awareness of sensor node positions.

- **Data redundancy**

There's a high likelihood of redundant data in WSNs, which routing protocols can exploit to enhance energy and bandwidth utilization.

Due to these unique characteristics, various novel routing mechanisms have been developed to address the routing challenges in WSNs. These mechanisms consider the inherent traits of WSNs and the requirements of specific applications and network architectures. An efficient routing scheme can significantly reduce power consumption and prolong network lifespan.

An essential concern in WSNs revolves around the discovery and maintenance of routes. This is due to energy limitations and the unpredictability of node conditions, including inefficiencies or failures, which result in frequent and unexpected changes in network topology. Routing methods for WSNs encompass established approaches customized to the specific characteristics of WSNs with the aim of reducing energy consumption. NIA are employed to create clustering-based routing protocols [42] [43]. These optimization algorithms dynamically select CHs to manage overall energy consumption in WSNs efficiently.

Here, routing protocols in WSNs can be categorized into four main schemes: communication model Scheme, network structure scheme, reliable routing scheme, and topology-Based Scheme, as illustrated in Figure 1.3. These schemes are tailored to address the specific challenges and requirements of WSNs.

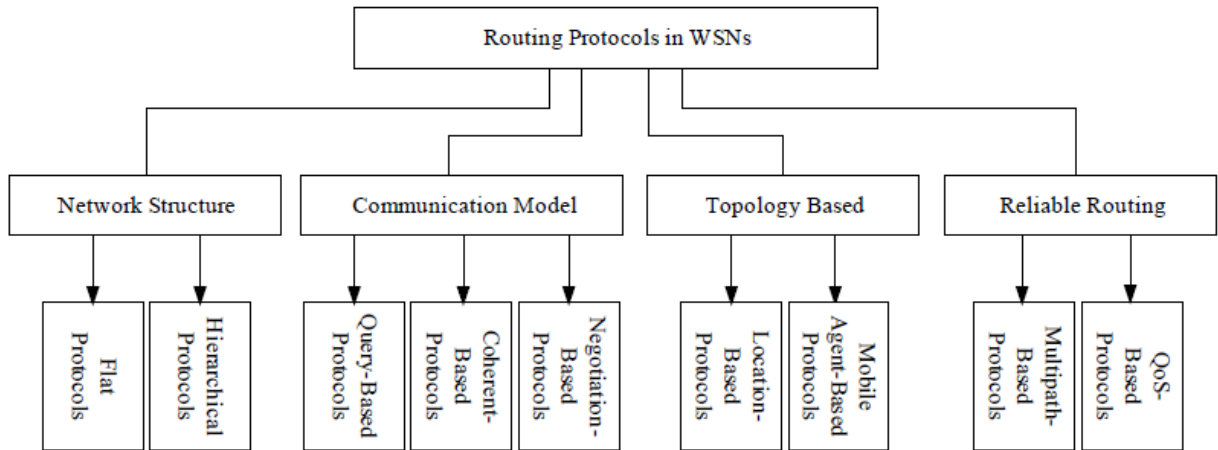


Figure 1.3: Categorization of routing protocols in WSNs

1.4.3 Mobile WSNs

The utilization of MWSNs for environmental monitoring and data collection has seen a notable increase in recent years. MWSNs typically encompass mobile SNs that can move within the network.

Real-world applications exhibit varying demands, with some requiring a mixture of static and mobile SNs within the same WSNs, while others necessitate an entirely mobile sensor environment [44]. Introducing mobility into WSNs primarily aims to conserve individual sensor nodes' energy reserves, extend the network's lifespan, and maintain connectivity among these nodes. To implement MWSNs, various mobility models are employed. These mobility models depict the movements of mobile sensors and how their location, velocity, and acceleration evolve, as mobility patterns can significantly impact protocol performance in MWSNs [36]. Depending on the application and network size, various mobility patterns are employed:

- **Predefined mobility pattern**

In this pattern, mobile elements follow a predefined path within the sensor field, stopping at predetermined positions to carry out specific tasks.

- **Random mobility pattern**

Random mobility patterns are suitable for mobile sensor nodes. In this pattern, mobile elements move randomly within the sensor field. Synthetic mobility models, like the RWP [45] and RPGM [46], simulate this random mobility.

- **Controlled mobility pattern**

Controlled mobility involves guiding mobile elements based on the control of the routing protocol. Factors like energy levels, avoidance of energy holes or hotspot issues, and connectivity are considered to achieve improved results.

Furthermore, hierarchical routing protocols, driven by NIA, are employed to conserve energy, maintain network connectivity, and extend the lifespan of MWSNs. Achieving this entails selecting optimal head nodes and controlling the mobility of mobile elements within MWSNs. This challenge can be categorized as an NP optimization problem and effectively addressed using NIA. These algorithms are instrumental in developing robust hierarchical routing protocols for MWSNs.

1.5 Research question

The primary research challenge involves connecting natural processes and artificial systems. The underlying logic behind harnessing nature to formulate algorithms is grounded in the notion that this approach enables us to establish a more profound resonance with the natural world. This, in turn, allows us to leverage the extensive insights from nature to address the spectrum of optimization challenges that confront us effectively.

The ramifications of this research endeavor are noteworthy. Researchers stand to gain a more expansive and versatile arsenal of algorithms rooted in a fusion of natural principles and computational constructs. These algorithms can be systematically applied across diverse sophisticated computational problems spanning various research domains. In essence, this research effort broadens the landscape of algorithmic frameworks, enriching the toolkit available for tackling intricate challenges within virtually any field of research.

1.5.1 Research objectives

The main objectives of this research are given as follows:

- To develop an improved version of nature-inspired algorithm to solve various numerical benchmark problems (covered in chapter 3,4,5,6 and 7; all the algorithms have fetched improved results)
- To perform parametric, statistical, and dimensional analysis on a newly proposed nature-inspired algorithm. (covered in chapter 3; successfully performed parametric, statistical, and dimensional analysis on all the algorithms)
- Design a cluster-based routing protocol to prolong the network lifetime of WSN using a nature-inspired algorithm. (covered in chapter 6; developed cluster based routing protocol using CONMRA for energy efficient WSN)
- Apply the developed algorithm on mobile WSN to improve various parameters like energy consumption, network lifetime, and stability period. (covered in chapter 7; ARNMRA has been developed for stable energy-efficient mobile WSN)

1.5.2 Significance of the proposed research

The proposed work carries substantial importance by revolutionizing algorithmic development, leveraging nature's wisdom, promoting interdisciplinary collaboration, augmenting problem-solving capabilities, enabling advanced computation, and broadening research opportunities.

- **Limitations of existing algorithms**

Two vital characteristics assist population-based optimization algorithms: exploration and exploitation. The exploration entails the capacity to uncover global optima by venturing into uncharted regions within the solution search space. Conversely, exploitation centers on enhancing solutions through leveraging the knowledge gained from prior successful outcomes. However, a notable weakness pervades most stochastic algorithms, manifested as premature

convergence. Take the ABC algorithm as an example, which demonstrates sub-optimal performance and struggles to explore the solution space effectively. This deficiency can be attributed to a random component in its solution search equation, which aids exploration but compromises the exploitation of the search space [1]. DE exhibits drawbacks in terms of both premature convergence and stagnation. DE sometimes ceases progression towards global optima, even when the population hasn't converged to local optima [47]. PSO showcases a remarkable ability to converge to decent solutions rapidly. Nevertheless, when benchmarked against alternative optimization methods, its capacity to fine-tune optimum solutions is weakened due to reduced diversity in later search stages [48]. Similarly, NMRA excels in exploitation but inefficient in exploration [4].

- **Enhancing exploration and exploitation in NIA**

Many algorithms, each taking inspiration from the natural world, have emerged in literature, with researchers continually proposing new techniques. These algorithms emulate the behaviors of individual species to address optimization challenges. However, a noticeable gap exists in the realm of improving two fundamental attributes—exploration and exploitation within these algorithms. The central focus of this thesis revolves around the conception and refinement of algorithms that exhibit superior exploration and exploitation capabilities. The author's goal is to enhance performance by introducing randomly generated solutions across the search space, rather than directing all random solutions exclusively towards the best-performing particles. In terms of exploitation, it's crucial that these random solutions align with the local group leader or, in other words, the current best solution. Consequently, improving the general equations governing the exploration and exploitation phases can lead to superior outcomes. Furthermore, the equilibrium between exploration and exploitation often defines an algorithm's efficiency. Different algorithms employ various operators to establish this equilibrium.

- **Rapid and highly accurate optimization algorithm**

Existing algorithms commonly encounter the drawback of becoming ensnared within local minima. Consequently, their operational speed is sluggish, and their reliability is compromised. To overcome this limitation, a novel algorithm is proposed, one that boasts a harmonized exploration and exploitation phase. This balanced approach empowers algorithms to refine their solutions dynamically, bypassing the entrapment in local minima. Consequently, the proposed algorithm exhibits swiftness, precision, and heightened dependability, particularly when confronted with higher-dimensional problems.

- **Relevance and contribution to the field**

We have extensively examined established algorithms like CS, GWO, SMA, and others. These cutting-edge algorithms have demonstrated their efficacy in resolving real-world challenges and have found widespread application across diverse research domains. Despite their commendable performance, these algorithms are not without their intrinsic limitations, as elaborated earlier. The domain has witnessed the emergence of refined versions of these foundational algorithms. These enhancements aim to rectify the inherent drawbacks of the original algorithms. Our research has been dedicated to augmenting NIA and evaluating their working ability across numerical simulations and CH selection problem in WSNs.

1.6 Primary thesis contribution

This thesis aims to elevate the performance of a diverse spectrum of NIA in tackling a wide range of optimization challenges encompassing real-world problems. To realize this objective, distinct hybridization concepts and innovative techniques are introduced and seamlessly integrated into the fundamental algorithms under scrutiny. In summary, the thesis offers several significant contributions:

- A chaos theory has been introduced to enhance the exploration characteristics of the NMRA in Chapter 3. Beyond this, parametric adaptation is incorporated to investigate the impact of the mating factor on the performance of NMRA.
- An optimization algorithm based on the hybridization of SMA and NMRA has been developed in Chapter 4. This algorithm integrates SMA's wrap food abilities with NMRA. Besides hybridization, the stagnation phase inspired by the GWO and CS algorithms has been integrated to handle the local optima stagnation problem.
- A hybrid variant of NMRA with WOA was developed in chapter 5. Apart from the added hybridization, simulated annealing-based mutation operation is also incorporated to make the algorithm self-adaptive.
- A novel self-adaptive hybrid COOT optimization NMRA-based clustering protocol is presented to perform the optimal selection of CH and subsequently enhance the network lifespan of WSNs in Chapter 6.
- An attraction and repulsion strategy based NMRA has been developed in Chapter 7. This strategy is utilized to create new breeder rat solutions and address the data gathering aspect in MWSNs to prolong the network lifetime.

This thesis undertakes a comprehensive validation process for the proposed algorithms, utilizing new methodologies and the latest benchmark algorithms. The performance assessment is conducted through various mathematical and statistical tests, which serve as benchmarks to determine the superior effectiveness and performance of the proposed algorithms.

1.7 Organisation of Thesis

The content of this thesis is structured across eight chapters, each outlined briefly as follows:

- **Chapter 1** provides an introductory overview. It touches upon the research motivation and delves into the theory of NIA and their applications, WSNs, routing techniques in WSNs, and Mobile WSNs. Additionally, it underscores the research objectives, the significance of the proposed work, and the thesis contribution. Finally, this chapter outlines the structure and organization of the thesis.
- **Chapter 2** furnishes an exhaustive literature review. It encompasses fundamental optimization concepts, an exploration of numerical benchmark test suites, a comprehensive overview of recently introduced algorithms, and clustering protocols for WSNs.
- **Chapter 3** presents a novel optimization algorithm, chaotic NMRA, based on chaotic maps. The working capability of this algorithm is evaluated for CEC 2005, CEC 2014, CEC 2019 benchmark problems, engineering design problems, and designing of energy-efficient WSNs.
- **Chapter 4** focuses on a new self-adaptive hybrid algorithm slime mould naked mole-rat algorithm (SM-NMRA) for numerical optimization tasks and CH selection problem of WSN to prolong the network lifespan.
- **Chapter 5** deals with the hybridization of NMRA and WOA along with a simulated annealing-based mutation operator for developing the WOA-based NMRA optimization algorithm.
- **Chapter 6** proposes an energy-efficient clustering protocol based on COOT optimization-based NMRA for WSNs.
- **Chapter 7** introduces attraction and repulsion strategy for NMRA to address the data gathering aspect in MWSNs to prolong network lifetime and enhance energy efficiency.
- **Chapter 8** encapsulates the research findings and emphasize the conclusion drawn. Additionally, it highlights potential avenues for future extensions of this work.

Chapter 2

Literature Review

The research background and literature review is introduced in this chapter. This segment will unfold the fundamental principles of numerical optimization and elucidate the rationale behind the development of novel optimization techniques. The central objective is to recapitulate the established optimization algorithms and their diverse applications in benchmark test suites and real-world scenarios. Furthermore, it delves into the fundamental characteristics of prominent benchmark test suites, including CEC 2005, CEC 2014, and CEC 2019, which serve as the basis for evaluating the proposed techniques in subsequent chapters. The chapter highlights well-known optimization algorithms like GA, DE, NMRA, SMA, and various clustering protocols for WSNs. Moreover, it articulates the key insights and driving motivations from this thesis.

2.1 Numerical optimization

2.1.1 Fundamental concepts

Majority of the real-world based challenges are framed as optimization problems, seeking to identify optimal solutions. Consequently, researchers actively collaborate to introduce effective methodologies and tools that cater to such domain-specific optimization issues. In scenarios like these, the key purpose is to ascertain the attainment of global optimal solution for an objective function while adhering to specific decision variables confined within a given range. For generality, these optimization challenges are mathematically expressed as either minimization or maximization problems. The formulated equation for the objective function is stated as:

$$f(\vec{Z}), (f : \Omega) \subseteq R^d \rightarrow R \quad (2.1)$$

where Ω signifies the domain of given problem, consisting of d decision variables or dimensions, each characterized as a vector $\vec{Z} = [z_1, z_2, z_3, \dots, z_d]$. As a consequence, for the given objective function $f(\vec{Z})$, the specified pivotal role of the optimization methodology lies in the quest to locate a prospective solution $\vec{Z}^*$ where $f(\vec{Z}^*) < f(\vec{Z})$ for all Z within the domain Ω , particularly in the reference of a minimization problem.

Given the prevalence of optimization challenges across a multitude of fields, various optimization

algorithms have emerged from the endeavors of researchers. Initially, classical approaches exhibited effectiveness [49] [50]. These classical methods leaned on differential mathematical calculus techniques, necessitating two pivotal criteria. The first requirement entails differentiating the optimization problem’s objective function concerning $\vec{Z}$. Secondly, the derivative must maintain continuity. These prerequisites limit the applicability of classical optimization methodologies, constraining them to specific benchmark based optimization problems and a limited scope of all real-world applications. Furthermore, classical algorithms often succumb to entrapment in local optima, leading to heightened computational demands and costs when contrasted with alternative algorithms [51].

For millions of years, nature has been continually addressing complex challenges, and biological systems like the immune system excel at eradicating foreign entities, while perception aids in pattern recognition. Decision-making processes are employed in developing robots, and learning mechanisms facilitate problem-solving at advanced levels. These biological systems showcase a remarkable level of computational intelligence inherent in nature. Consequently, many optimization algorithms have been developed within this context, commonly called nature-inspired algorithms. They have gained popularity across various domains, including business management, stock market forecasting, optimal route planning, and others [52]. Their widespread adoption is attributed to their adaptability, resilience, and swifter response times compared to conventional optimization methods. An additional advantage lies in population-based optimization algorithms, eliminating the need for an initial guess when addressing problems. Furthermore, these algorithms are favored because they necessitate fewer parameters to fine-tune, rendering them suitable even for highly complex research problems.

2.1.2 Test Suite

This test suite encompasses various functions designed to evaluate the performance of the algorithms under examination. These functions serve as a means to validate and rigorously test newly modified algorithms. Researchers initially formulated them and were first evaluated at the IEEE Congress on Evolutionary Computation (CEC), since then acquiring the name "CEC Benchmark functions (test suite)." The very first set of widely employed functions was introduced by [7] and is referred to as the CEC 2005 test suite. Over the past decade, these benchmarks have become a cornerstone of optimization research and have found applications across numerous research domains. Their popularity stems from their extensive applicability alongwith diversity, separability, high dimensionality, great scalability, and numerous locally and globally located minima. These features have rendered them exceptionally arduous to solve. In essence, these given benchmark functions are designed to emulate practical-world scenarios, presenting extremely demanding challenges for research in various domains. Additionally, this thesis draws upon other CEC benchmarks, including CEC 2014 [8] and

CEC 2019 [9].

Concerning the fundamental characteristics of the mentioned test functions, which span various categories namely uni-modal functions along with multi-modal functions, fixed-dimensional functions also hybrid functions, and lastly, composite functions. These given classifications are based on the number of local and global minimal solutions inherent within them. Uni-modal functions are the simplest, featuring a lone global minimum. It's important to note that the distinction between global and local minima is nonexistent in these functions—they are equivalent. These functions evaluate an algorithm's exploitation capabilities, offering insights into its local search proficiency. The subsequent category encompasses multi-modal functions, characterized by one global minimum and multiple local minima. Due to the proliferation of local minima, these functions present substantial challenges. As the count of local minima escalates, so does the test complexity. These functions are pivotal for assessing an algorithm's exploratory potential. Fixed-dimensional functions, forming the third category, possess a consistent dimension size. They gauge an algorithm's consistency in locating global minima or the ultimate best solution. In essence, these functions shed light on an algorithm's convergence properties. Hybrid and composite functions constitute another classification. Specifically these functions are forged through a fusion of uni-modal and multi-modal functions. They pose formidable challenges and significantly contribute by creating intricate environments for evaluating algorithms. The complexity is further magnified with an increase in dimensionality.

The CEC 2005 benchmark set comprehensively encompasses simple uni-modal, multi-modal and fixed-dimensional functions. Conversely, the subsequent benchmark sets CEC 2014 and CEC 2019, are the most intricate and complex test problems. An algorithm's performance against these test suites could deem it a prospective contender for attaining state-of-the-art status. As we proceed further in the upcoming section, we delve into recently developed algorithms, examining contemporary literature.

2.2 Optimization algorithms

NIA have demonstrated their effectiveness for real optimization problems. These algorithms retain certain past information across successive iterations, necessitate only a minimal set of parameters for tuning, and consequently, offer a straightforward implementation. The subsequent subsections provide a comprehensive review of some well-known algorithms.

2.2.1 Genetic algorithm

GA has been developed by Holland [53] and draws inspiration from the principles of biological evolution. GA's performance is contingent on four primary factors: population size, mutation rate, crossover rate, and the number of generations. Following its inception, numerous researchers have endeavored to refine GA's performance. As a result they introduced various alternative methodologies for crossover and mutation, aiming to elevate solution quality. In the realm of crossover, N-point crossover and segmented crossover emerged as alternatives to the singular crossover point selection, employing multiple points for crossover [54] [55]. Moreover, the introduction of adaptive GA facilitated the utilization of precise parameters for configuring population size, crossover probability, and mutation probability. These parameters exhibit dynamism across iterations. As an example, when the given population's progress is stagnant, the mutation rate increases; conversely, as the population improves, the mutation rate decreases [56].

GA's applications span a wide spectrum, encompassing domains like scheduling [57], machine learning [58], robotics [59], signal processing [60], business [61], mathematics [62], routing [63], and others. It is pertinent to acknowledge that GA's effectiveness diminishes when confronted with exceedingly intricate high-dimensional, multi-modal challenges, particularly where fitness function evaluations entail substantial computational complexity or entail a significantly high number of iterations [64].

2.2.2 Differential evolution

DE stands out noticeably as the most potent algorithms for real-parameter optimization [65]. DE has garnered significant attention since its inception, operating with computational steps akin to a standard evolutionary algorithm. Numerous researchers have proposed many variants of the fundamental algorithm, each aimed at enhancing performance. Notably, a substantial volume of recent research has resulted in a wide array of DE variants. These variants encompass diverse enhancements, encompassing self-adaptation of parameters [66], dynamic population size adaptation [67], strategy adaptation [47], and the refinement of the differential mutation operator [68].

The effectiveness of DE has further been augmented through the integration of novel mechanisms, such as population adaptation [69], genotypic topology [70], and opposition-based initialization [71].

2.2.3 Particle swarm optimization

PSO, introduced by Kennedy and Eberhart in 1995 [11], is a meta-heuristic global optimization technique. PSO takes inspiration from the collective behavior of birds, emulating their flocking

patterns. In a solution space, these virtual birds navigate, and their coordinated movements guide the search for optimal solutions. This simulation captures their journey toward nourishment sources. Over the past two decades, PSO has gained prominence for its application in addressing intricate, unsupervised, multidimensional problems that elude resolution through conventional deterministic algorithms.

Since its inception, several enhancements have turned out in various propositions to refine the performance of the standard PSO. These adaptations encompass a range of variations, including the quantum-behaved PSO incorporating a differential mutation operator for optimizing core fuel management [72], the bare-bones PSO comprising mutation and crossover operators from the DE algorithm [73], adaptive chaotic embedded PSO for wavelet parameter estimation [74], hybrid PSO-GA for channel evaluation in frequency division multiplexing systems [75], and the hybrid PSO-ABC leveraging a recombination procedure [76], among other innovations.

2.2.4 Cuckoo search

CS algorithm, which is a meta-heuristic swarm intelligence approach, that was devised by Xin-She Yang and Suash Deb in 2009 [13]. Since initiation, CS has exhibited considerable promise across various optimization domains. Subsequently, an array of modified versions has emerged to enhance its performance.

Notable modifications of the CS algorithm include a binary version proposed in [77] to tackle binary optimization problems. The discrete version of CS was introduced by [78] to address the traveling salesman problem. A modified CS algorithm surfaced in [79] for solving unconstrained optimization problems. A one-rank CS algorithm targeted the optimization of algorithmic trading systems [80]. Diverse variants of CS include the improved CS algorithm, which modifies step length to address planar graph coloring problems [81], a new CS algorithm boasting enhanced exploration and exploitation attributes for numerical optimization [82], the Chaotic CS algorithm for reconstructing chaotic dynamics [83], and a hybrid CS coupled with GWO for medical image fusion [84].

Beyond these modifications, the CS algorithm has found application in resolving a broad spectrum of real-world optimization problems. These encompass the diagnosis of diabetes [85], clustering of web documents [86], image recognition [87], improvement of low-contrast satellite image quality [88], structural design challenges [89] and embedded system design problems [90].

2.2.5 Marine predator algorithm

MPA represents one of the latest editions within the swarm intelligence-driven nature-inspired algorithms [91]. Anchored in the foraging behaviors and interactions of marine life inhabiting oceans, MPA revolves around three fundamental principles that emulate predator-prey dynamics. These principles encompass:

- *High velocity ratio*: Reflecting instances where the prey surpasses the predator in speed.
- *Uniform velocity ratio*: Signifying situations where the respective prey and the predator move at identically uniform speeds.
- *Low velocity ratio*: Corresponding to scenarios where the predator outpaces the prey in movement.

These tripartite steps dictate the exploration and exploitation operations of MPA. Recent applications of MPA encompass photovoltaic array reconfiguration [92] and multilevel thresholding [93].

2.3 Algorithms used for proposed research

This subsection details the recently introduced optimization algorithms used in this thesis. By combining these algorithms and various modifications, research could benefit from their collective capabilities in exploring and solving complex optimization problems.

2.3.1 Naked mole-rat algorithm

NMRA is a kind of swarm intelligent algorithm that consists of two different types of mole rats, namely worker rats and breeder rats, that are used to define the mating pattern of NMRs [4]. In this algorithm, a single queen heads the entire group of rats. Breeders are responsible for reproduction while workers perform some other tasks. The worker with the best performance can enter into the breeder's group, and the breeder with the best performance among all the breeders will mate with the queen. Thus, the NMRA is basically categorized into three phases: initialization of NMRs, worker phase, and breeders phase.

Initialization of NMRs: In this phase, mole rats population (N) is distributed uniformly in a random manner for a dimension D . The dimension D defines the different parameters which are tested during the problem. Here, the initialization of NMRs is given as:

$$R_{i,j} = R_{min,j} + rand(0, 1) \times (R_{max,j} - R_{min,j}) \quad (2.2)$$

where $i \in [1, 2, 3 \dots N]$, $j \in [1, 2, 3 \dots D]$, $R_{i,j}$ is i^{th} obtained solution in j^{th} dimension, $R_{min,j}$ and $R_{max,j}$ defines the lower and upper limits of the function used in problem. The $rand(0,1)$ is a random number distributed uniformly in $[0, 1]$. After the population of rats initialization, evaluate the objective function and calculate the fitness of the objective function to identify breeders (B) and workers (W). This fitness value is also used to initially calculate the overall best solution NMR_{best} . After this, the naked mole rat (NMR) population is subjected to multiple iterations to complete the search process.

Worker phase (exploration): In this phase, the worker's fitness is evaluated as to whether they are capable of becoming breeders or not. Here, all the workers of NMR try to enhance their fitness and want to enter into the breeders group. A new NMR solution is generated by considering NMR's experience and local information. If the newly developed solution's fitness is superior to the previous one, the previous solution is updated with the newly developed solution; otherwise, retain the old one as it is. After the search process is completed by all the workers' NMR, the final fitness value of all workers is remembered. The following equation is used to obtain the worker's solution:

$$x_i^{t+1} = x_i^t + \lambda(x_a^t - x_b^t) \quad (2.3)$$

where, x_i^t corresponds to i^{th} worker rat solution for t^{th} iteration, factor λ denotes mating, x_i^{t+1} denotes new worker's solution, and x_a^t & x_b^t are randomly selected solutions from the group of workers.

Breeder phase (exploitation): In this phase, breeders also try to enhance their fitness for a breed with the queen and stay in the breeder's pool. Here, breeder rats update their solution based upon the rated value of breeding probability (bp) as per the overall best solution R_{best} . Then the new solution of the breeder is generated as:

$$y_i^{t+1} = (1 - \lambda)y_i^t + \lambda(R_{best} - y_i^t) \quad (2.4)$$

where, y_i^t denotes the i^{th} breeder obtained in t^{th} iteration, λ manage frequency of breeders to breed with the queen and y_i^{t+1} corresponds to new breeder rat. Here, the bp is any value in between $[0,1]$, and initially it has been selected as 0.5.

Here, NMRA is proposed on the basis of the presumption that a single queen is responsible for

performing matting with the best breeder in the breeder's pool. This algorithm identifies workers and breeders from the population of NMRA, and after completing the initial evaluation, the selection of the best worker and breeder is to be done. The fitness value of workers is updated, and this will enhance the possibility of becoming a breeder. In the same way, the breeder's fitness is also updated so that they continue to exist as a breeder and the best male breeder breed with a female for reproduction. The breeder whose fitness is not improved and becomes infertile may be pushed back to a pool of worker rats. The best solution or breeder from the entire population is optimal for the problem under test. The pseudo-code for the NMRA is mentioned in Algorithm 1

Algorithm 1: Pseudo-code of NMRA

Start:

Inputs: Define the random population of NMRA: (N)

Decide count of breeder mole-rats (B) = $N/5$

Decide count of worker rats (W) = $N - B$

Initialize breeding probability: (bp)

Define problem's dimension (D)

Output: Evaluate best search candidate (R_{best})

while $t \leq$ maximum iteration count (t_{max})

Worker phase

for $i = 1:W$

 execution of NMRA worker phase using equation (2.3)

 evaluate x_i^{t+1}

end for

Breeder phase

for $i = 1:B$

 if $\text{rand}(0,1) > bp$

 execution of breeder phase using equation (2.4)

 evaluate y_i^{t+1}

 end if

end for

unite new mole rats (W & B)

update (R_{best})

increment t

end while

 save R_{best}

Stop

NMRA was established to be useful for resolving various optimization problems [94]. However, NMRA shows good exploitation properties but lags behind due to the problem of poor exploration, slow convergence rate, and lack of perfect equilibrium between exploration and exploitation operations. These observations inspired the authors to focus on improving the working ability of NMRA. In [95], three versions of NMRA were developed to improve the exploration and exploitation capabilities of classical NMRA. The first version uses the notion of opposition-based learning; the second version uses the advantage of local neighborhood search, and the third version uses the combined

advantage of these two. After that, the third version is implemented to solve the localization problem of wireless sensor networks. In [96], basic parameters of NMRA have been analyzed for 6 inertia weights (random, exponential decreasing, chaotic, linear decreasing, simulated annealing & logarithmic decreasing) and 5 mutation operators (Cauchy, diversity mutated, trigonometric, neighborhood search and Lévy flight). The Lévy flight-based NMRA is treated as the best algorithm and used for designing the band-notched ultra-wideband antenna. NMRA is hybridized with harmony search to identify language features from audio signal [97]. A modified version of NMRA was developed for handling image segmentation problems [98].

2.3.2 Slime mould algorithm

SMA attains its basis from the foraging behavior of slime molds found in natural environment [5]. Slime mould is generally treated as *Physarum polycephalum* and its life cycle was given by [99]. These slime moulds exhibit the property of using many food sources simultaneously to create a unique network pattern. The numerical model of SMA is based on various characteristics of slime mould and divided into three phases: approach food, wrap food, and oscillation.

Approaching towards food: The approaching behavior of slime mould towards food is based on odour present in the air. To implement this behavior, the mathematical model follows the contraction mode:

$$\vec{x}^{t+1} = \begin{cases} \vec{x}_l^t + \vec{s}_g \cdot (\vec{W} \cdot \vec{x}_p^t - \vec{x}_q^t), & r < c \\ \vec{s}_h \cdot \vec{x}^t, & r \geq c \end{cases} \quad (2.5)$$

where $\vec{x}_l^t$ is the current location of the individual with maximum odour concentration, $\vec{x}_p^t$ and $\vec{x}_q^t$ are anonymously selected solutions among the population of slime mould. The parameter $\vec{s}_g$ is defined in the range of $[-b, b]$, parameter $\vec{s}_h$ is linearly decreasing from one to zero, t is the value of the current iteration, and $\vec{W}$ is used to assign the weight to slime mould.

The formulae to calculate the value of c and b is given as:

$$c = \tanh|f(i) - f_{best}| \quad (2.6)$$

$$b = \arctanh \left[- \left(\frac{t}{t_{max}} \right) + 1 \right] \quad (2.7)$$

where $f(i)$ is the fitness of $\vec{x}$, $i \in [1, 2, \dots, n]$, f_{best} represents best fitness value generated so far after the completion of all the iterations and t_{max} represents the majority number of iterations.

The weight factor $\vec{W}$ of slime mould is formulated as:

$$\vec{W}(f_{sorted}(i)) = \begin{cases} 1 + r \cdot \log\left(\frac{f_b - f(i)}{f_b - f_w}\right), & \text{condition} \\ 1 - r \cdot \log\left(\frac{f_b - f(i)}{f_b - f_w}\right), & \text{others} \end{cases} \quad (2.8)$$

where r is randomly distributed in the range of $[0, 1]$. f_b and f_w shows the best and worst fitness values obtained in the current iteration respectively. *condition* denotes rank of $f(i)$ for first half of the population and f_{sorted} defines the sequence of sorted fitness values.

Here, it can be seen that by using equation (2.5), the position of individual search agent $\vec{x}$ is updated following currently best position $\vec{x}_l$, proper tuning of parameters $\vec{s}_g$, $\vec{s}_h$ and individual position change parameter $\vec{W}$. Thus, in approaching the food phase, individual search agents possibly search all the directions and enhance the algorithm possibility to obtain an optimal solution.

Wrap food: In this phase, slime mould venous network structure is mathematically modeled during the contraction mode of the network. The high thickness of slime mould vein is obtained by a higher volume of food the vein came in contact with, faster movement of cytoplasm and strong generation of the wave by bio-oscillator. Here, the positive and the negative feedback between the concentration of food explored and slime mould vein thickness is mathematically presented by equation (2.8). In this equation, the parameter r defines contraction mode uncertainty of the venous network, $\log$ is responsible for reducing the changing rate value of contraction frequency, and *condition* imitates adjustment of the slime mould searching model in accordance with food quality. When the concentration of food is high in the region, the greater weight is assigned to the region; when the region's food concentration is lower, the weight reduction is observed for that region, and other regions will be explored. Thus, the formula used to update slime mould position is based on above principle:

$$\vec{x}^* = \begin{cases} rand \cdot (ub - lb) + lb, & rand < z \\ \vec{x}_l^t + \vec{s}_g \cdot (\vec{W} \cdot \vec{x}_p^t - \vec{x}_q^t), & r < c \\ \vec{s}_h \cdot \vec{x}^t, & r \geq c \end{cases} \quad (2.9)$$

where lb and ub represent the lower and upper boundaries of the search region. *rand* and r is the random number in the range of $[0,1]$. The value of parameter z is taken as 0.03 in the original SMA.

Oscillation: To move the slime mould in a higher concentration of food, the cytoplasm flow in veins

of the venous network is to be controlled by a propagation wave generated through a bio-oscillator. Here, the width of slime mould venous is varied by using $\vec{W}$, $\vec{s}_g$ and $\vec{s}_h$ parameters.

The weight factor $\vec{W}$ is important in enhancing slime mould efficiency in selecting the optimal food source. Here, $\vec{W}$ imitates the slime mould frequency of oscillation mathematically at various food concentrations. This enables the slime mould to approach food quickly when food quality is high and approach slowly towards food when food quality is low.

The parameter $\vec{s}_g$ is defined for the interval $[-b, b]$ and approaches towards zero over the course of iterations. The oscillation process of this parameter helps to decide slime mould state whether the approach has already found a food source or discovered other food sources. Further, the value of parameter $\vec{s}_h$ is linearly decreased from one to zero at the end of iterations. The pseudo-code of SMA is provided in Algorithm 2.

Algorithm 2: Pseudo-code of SMA

Start:

Define the population size, maximum iteration count (t_{max})

Initialize slime mould's position: $x_i, (i = 1, 2, \dots, n)$

while ($t \leq t_{max}$)

evaluate fitness of all slime mould

update best fitness value, x_l

evaluate W using equation (2.8)

for whole search space

update the value of c, s_g, s_h

update the positions of slime mould using equation (2.9)

end for

increment t

end while

save x_l

Stop

SMA is one of the most popular stochastic optimization algorithms, proving its effectiveness in solving various optimization problems. Apart from solving optimization problems by original SMA, various modifications have been added to the basic structure of SMA and improve its performance. A novel hybrid SMA-WOA is proposed for optimizing the segmentation problem of chest X-rays [100]. The artificial neural network model is combined with SMA in [101] to estimate the demand for water in urban areas. In [102], SMA is hybridized with adaptive DE to solve global optimization tasks and overcome SMA's local optima stagnation problem. A binary variant of SMA is proposed with the help of eight S-shaped and V-shaped transfer functions and used for feature selection problem [103]. The exploration operation of SMA was improved by using the concept of firefly algorithm [104] for a minimal selection of features from data sets. In [105], chaotic and crossover opposition-based SMA was proposed to optimize the wind energy cost of turbines. The performance of SMA is

enhanced with the help of using Lévy flight instead of random distributions [106]. To improve the exploitation properties of SMA, a new variant leader SMA has been proposed and its performance is tested for multi-spectral images [107]. The concept of opposition-based learning was added to basic SMA to identify defects in taper roller machine [108]. The enhanced version of SMA based on chaos is proposed for minimizing the energy cost of wind turbines [109]. In [110], HHO is hybrid with SMA to improve the convergence speed of classical SMA. An improved version of SMA is proposed in [111] by using the cosine function for controlling parameters. Hybridize GWO with SMA for numerical optimization in [112] and PSO with SMA in [113]. Further, SMA is applied to solve real optimization problems such as parameter estimation of photo-voltaic cell [114], optimal parameter tuning of fuzzy controllers [115], monitoring structural health of beam and bar planar [116], minimization of coupling effects between high voltage transmission line and magnetic pipeline [117], optimized over current relay coordination in power networks [118], designing of optimized path with avoiding obstacles in automatic mobile robots [119], maximizing hosting capacity of distributed networks [120], selection of optimal parameters for designing of power stabilizer [121], designing of passive filter [122] and others.

2.3.3 Whale optimization algorithm

WOA is one of the most popular optimization algorithms proposed by [25]. This algorithm is based on the unique foraging behavior of whales, especially humpback whales, to perform the optimization process. The humpback whales use a special type of hunting method called the bubble-net feeding method used for the development of WOA. The mathematical model of WOA is formulated by considering three phases of the algorithm: initialization (encircling of prey), exploitation phase (spiral bubble net-attacking), and exploration phase (searching for prey).

Encircling of prey: In this phase, humpback whales have identified the position of prey and perform the encircling. During the initial stages, the optimal position of the search agent is not recognized so the best individual is to be assumed as targeted prey by WOA. After defining the best candidate, all other candidates in the search space continuously approach the best candidate by updating their positions. The following mathematical equations are used to present this mechanism:

$$\vec{H} = |\vec{G} \cdot X_{best}^t - \vec{X}^t| \quad (2.10)$$

$$\vec{X}^{t+1} = X_{best}^t - \vec{F} \cdot \vec{H} \quad (2.11)$$

where $\vec{F}$ and $\vec{G}$ are two coefficients for defining position of individuals, t denotes the value of iteration under consideration currently, X_{best}^t defines the position of best candidate, $\vec{X}$ represents the position

vector, $||$ symbol is used for extract absolute value and $.$ performs the multiplication element by element. Here, it should be taken care that the value of $\vec{X}_{best}$ is updated during the execution of each iteration if a better solution is generated in the current iteration.

The coefficients $\vec{F}$ and $\vec{G}$ are formulated as:

$$\vec{F} = 2\vec{f}.rand(0,1) - \vec{f} \quad (2.12)$$

$$\vec{G} = 2.rand(0,1) \quad (2.13)$$

where, $\vec{f}$ follows linear decreasing function ranging from 2 to 0 for total iterations and $rand$ is randomly distributed number between 0 and 1.

Exploitation phase (bubble-net attacking): The shrinking encircling mechanism and the spiral updating position method are both used to model this behavior. The fluctuation range of $\vec{F}$ is determined by $\vec{f}$, which may be found using equation (2.12). Also, the value of $\vec{f}$ drops from 2 to 0 during the period of iterations, affecting the value of $\vec{F}$. So, if $\vec{F}$ is in the range $[-1, 1]$, the agent's next location will be somewhere between the present position and the current best (prey) position. The spiral-based position update approach, on the other hand, calculates the distance between the current location of the whale and that of the prey and then constructs a helix-shaped movement of humpback whales based on this position, which is modeled as:

$$\vec{X}^{t+1} = \vec{H}'.e^{bl}.cos(2\pi l) + X_{best}^t \quad (2.14)$$

where $|\vec{H}' = X_{best}^t - \vec{X}^t|$ denotes the distance that exists between the search candidate (whale) and best search candidate (prey), a constant value of b represents logarithmic spiral form, l is randomly distributed between -1 and 1 . Here, it is assumed that 50% has a chance of selecting either a shrinking mechanism or a spiral model for performing exploitation (prey's attacking). The final equation can be expressed mathematically as follows:

$$\vec{X}^{t+1} = \begin{cases} X_{best}^t - \vec{F}.\vec{H} & \text{if } u < 0.5 \\ \vec{H}'.e^{bl}.cos(2\pi l) + X_{best}^t & \text{if } u \geq 0.5 \end{cases} \quad (2.15)$$

where random number u is distributed uniformly in the range $[0,1]$.

Exploration phase (prey's search): To find prey, modification is applied on $\vec{F}$ and implemented with the help of equation (2.12). Here, whales randomly conduct their foraging based on their relative

positions to one another. Thus, $\vec{F}$ is utilized with random values greater than or equal to 1 to diverge the search agent from the reference whale. During this phase, search agent updates their position based on the randomly chosen solution rather than the best search agent. This technique, along with $|F| > 1$, encourages exploration and enables WOA to do a global search and modeled as:

$$\vec{H} = |\vec{G} \cdot \vec{X}_{rd} - \vec{X}^t| \quad (2.16)$$

$$\vec{X}^{t+1} = \vec{X}_{rd} - \vec{F} \cdot \vec{H} \quad (2.17)$$

where $\vec{X}_{rd}$ denotes a randomly selected search candidate (whale) from an entire population. The pseudo-code of WOA is given in Algorithm 3.

Algorithm 3: Pseudo-code of WOA

Start:

Initialize humpback whale's population: $X_i, (i = 1, 2, \dots, n)$

Parameters initialization: F, G, l & u

Calculate whale's fitness

Best candidate: X_{best}

while ($t \leq t_{max}$)

for each whale

if $u < 0.5$

if $|F| < 1$

Update whale's position using (2.11)

else if $|F| \geq 1$

Update whale's position using (2.17)

end if

else if $u \geq 0.5$

Update whale's position using (2.14)

end if

end for

Fitness evaluation

Updating of X_{best}

$t \rightarrow t + 1$

end while

Save X_{best}

stop

Since its inception, improved versions of WOA have been developed to enhance its working efficiency. One modified version of WOA, namely Enhanced WOA, was introduced by Kaveh and Ghazaan [123], in which three new parameters are incorporated to enhance WOA's exploration and exploitation capabilities. A hybridization approach was also explored, combining pattern search with WOA [124]. This hybridization aimed to fine-tune the best solution obtained by WOA. However, using two algorithms increased the total number of function evaluations, making the approach more complex and time-consuming. To refine WOA's performance, the concept of inertia weight was introduced

[125]. This parameter was added to adjust the influence of the current best solution on WOA's performance, thereby striving to improve its exploration and exploitation characteristics. WOA was extended to multi-objective optimization in studies such as [126] and [127], where Pareto optimality was applied to optimize vehicle fuel consumption and mobile robot path planning, respectively. A substantial body of literature explored WOA's potential in designing photonic crystal filters [128]. Another significant modification incorporated into WOA was integrating the adaptive neuro-fuzzy inference system [129]. These authors subsequently applied the algorithm to feature selection and land pattern classification problems. Further improvements in WOA were introduced in [130], where re-initialization and adaptive strategies were employed to balance exploration and exploitation. It was observed that the algorithm needed additional modifications to enhance its efficiency in exploring the search space.

WOA has been deployed for various optimization problems. In [131], WOA was utilized to determine optimal connection weights in neural networks, facilitating the estimation of mean square errors across different data sets, including blood, breast cancer, diabetes, hepatitis, vertebral, and diagnosis data sets. Another application involved WOA in distributed generator systems to optimize the size of renewable resources [132]. Additionally, it was employed for liver segmentation in MRI images [133]. To enhance WOA's performance, a Levy flight-based variant was introduced [134]. This modification aimed to make WOA more robust and faster while preventing premature convergence by improving its exploratory capabilities.

Further applications of WOA span a wide spectrum, including solving combined economic emission dispatch problems [135], historic manuscript binarization [136], wrapper feature selection [137], optimizing photovoltaic installations [138], multilevel thresholding segmentation [139], developing aging degree evaluation models [140], introducing wavelet mutation [141], and employing support vector machines for spam profile detection [142].

2.3.4 COOT optimization algorithm

The COOT optimization algorithm is based on the behavior of coot birds on the water's surface [6]. Three movements make up the coot swarm's behavior on water: disorganized movement of an activity, synchronized movement, and moving in a chain on the water's surface, with each coot following the one in front of it. The COOT optimization technique imitates collective movements (regular and irregular motions on the water's surface). A few coots in the front of the group are considered as group leaders and direct the entire group toward the desired food. The mathematical model to implement the COOT optimization method is divided into five phases, with initialization

of the population in the first phase, random swaying to one side or the other in the second phase, chain action in the third phase, position updating following the group leaders in fourth phase and leaders directing the group to the best location in the final phase.

Initialize population of coots: The COOT optimization technique starts with random initialization of coot bird population as $CO = [CO_1, CO_2, \dots, CO_n]$. There is no assurance that the optimization technique provides a solution in one run because population-based optimization approaches search for multiple optimization problems. However, the likelihood of discovering the global solution grows with enough random solutions and iterations. The population of coots is defined randomly as:

$$COP_i = rand[1, dim] \times (ub - lb) + lb \quad (2.18)$$

where COP_i is the coot position, dim is the number of variables or the problem's dimension under test, lb is the lower bound and ub is the upper bound of the optimization problem. After the population's initialization, each solution's fitness must be determined using $O_i = f(CO)$, and each search agent's position is established. The GL number of coots are chosen to serve as group leaders.

Random swaying to one side or the other: To carry out this movement of the coot bird, the coot starts moving in the direction of a random location determined by equation (2.19) in the search space.

$$r_{pos} = rand[1, dim] \times (ub - lb) + lb \quad (2.19)$$

This coot movement explores the search space and will help the algorithm escape from local optima if it becomes stuck in the local optimal solution. The following equation is used to determine Coot's new position:

$$COP_i = COP_i + \beta \times r_2 \times (r_{pos} - COP_i) \quad (2.20)$$

$$\beta = 1 - \left(\frac{t}{t_{max}} \right) \quad (2.21)$$

where r_2 is randomly distributed in $[0,1]$, t denotes current iteration value and t_{max} is maximum iterations count for the algorithm.

Movement of chain: The chain movement can be implemented using the average location of two

coots inspired by the SSA [24]. The mathematical equation to update the coot's position in this movement is given as:

$$COP_i = 0.5 \times (COP_{i-1} - COP_i) \quad (2.22)$$

where first position of coot is COP_i and second position of coot is COP_{i-1} .

Update position in accordance with GL: The group is often led by a few coots in the front, and the remainder of the coots must move closer and alter their location following the group's leaders. The coots can adjust their position based on the average position of the leaders, which can be considered. The occurrence of premature convergence may happen due to the average position. Here, leader selection is made by a mechanism based on the formula (2.23).

$$j = 1 + (i \text{MOD}[GL]) \quad (2.23)$$

where j is the index number of leaders, GL is the total number of leaders, and i is the index number of the current coot. The coot must update its location based on the j^{th} leader. The equation (2.24) determines the coot's next location using the chosen leader.

$$COP_i = LOP_j + 2 \times r_2 \times \cos(2\pi r_d) \times (LOP_j - COP_i) \quad (2.24)$$

where COP_i is the coot's current location, LOP_j is the chosen leader position, r_2 is a random number in the range $[0, 1]$, and r_d is a random number in the range $[-1, 1]$.

Leaders directing the group to the best location: The leaders must adjust their location concerning the global optimal solution to guide the group toward it. Leaders' positions should be updated using equation (2.25). Around the current ideal location, this formula searches for better positions.

$$LOP_i = \begin{cases} \gamma \times r_3 \times \cos(2\pi r_d) \times (CO_{best} - LOP_i) + CO_{best} & r_4 < 0.5 \\ \gamma \times r_3 \times \cos(2\pi r_d) \times (CO_{best} - LOP_i) - CO_{best} & r_4 \geq 0.5 \end{cases} \quad (2.25)$$

$$\gamma = 2 - \left(\frac{t}{t_{max}} \right) \quad (2.26)$$

where r_3 and r_4 are randomly distributed in the range $[0, 1]$, and CO_{best} is the best position ever

discovered from the entire population of coots.

A good number of COOT optimization algorithm variants have been developed in which an enhanced COOT optimization algorithm for dimensionality reduction tasks [143]. These improved variants likely build upon the basic principles of Coot Optimization while introducing modifications to address dimensionality reduction challenges better. A modified COOT Optimization Algorithm is an approach aimed at optimizing and identifying key battery parameters [144]. An improved COOT bird algorithm enhances coverage within WSNs [145] and optimizes the operation of renewable energy sources within the power system [146]. To enhance information retrieval by automatically expanding user queries, a hybrid cat and mouse-COOT algorithm has been used [147].

2.4 Routing protocols for WSNs

In recent years, significant research has focused on exploring the potential benefits of collaboration among sensors in data collection and processing. However, it's important to note that sensor nodes face significant constraints regarding energy supply and available bandwidth. Consequently, there is a pressing need for innovative techniques that can eliminate energy inefficiencies, which could otherwise lead to a shortened network lifetime. These constraints become even more challenging when dealing with the typical deployment of many sensor nodes. This situation underscores the importance of incorporating energy-awareness into all layers of the networking protocol stack. For instance, at the network layer, it is crucial to develop methods that efficiently handle route discovery and data relay from sensor nodes to the BS and ensure the reliability of these processes. This reliability is essential for maximizing the overall network lifetime.

Heinzelman *et al.* initially introduced the LEACH protocol for WSNs [148]. This protocol operates in rounds, and CHs selection is a crucial aspect. It is based on a predetermined percentage of CHs for the network and the historical CH selection of each node. Nodes make this decision by generating a random number between 0 and 1. If this number falls below a certain threshold, the node becomes a CH for the current round. This threshold mechanism is thoughtfully designed to ensure that a node becomes a CH only once within a specific time interval. This approach aims to achieve an equitable distribution of energy consumption among all sensors. Subsequently, several algorithms were proposed to improve the performance of LEACH. LEACH-C [149], for instance, shifts the responsibility of CH selection from the node itself to the BS, which possesses global knowledge of the network, enabling more informed decisions. Another protocol called NEAP [150] refines CH election by considering factors such as the node's current energy level, the number of consecutive rounds without being a CH, and the observed performance in CH election. NEAP adapts its threshold value

per round, resulting in a more power-efficient and adaptive clustering hierarchy when compared to LEACH.

TEEN is a reactive protocol designed for event-driven applications [151]. In TEEN, sensor nodes continuously monitor their environment, but data transmission occurs less frequently. CHs in TEEN provide their member nodes with two threshold parameters: hard and soft. The hard threshold represents a specific value of a sensed attribute that, when reached, activates a node to transmit data. On the other hand, the soft threshold indicates a slight change in the attribute's value that can trigger a node to transmit data again. Once the hard threshold value is attained, sensor nodes initiate data transmission. Subsequent data transmission occurs when there is a minimum environmental change, as indicated by the soft threshold value.

The SEP utilizes a similar method to LEACH for selecting CHs [152]. However, SEP introduces an additional parameter to characterize different types of nodes based on their initial energy levels. Specifically, advanced nodes have higher energy reserves than normal nodes. SEP increases the likelihood of advanced nodes being elected as CHs to ensure that all nodes in the network deplete their energy resources around the same time, thereby enhancing the network's stability period. An enhancement of SEP [153] [154] further extends this concept by considering three types of nodes: normal, intermediate, and advanced. This multi-level heterogeneity allows for more precise management of energy resources within the network. On the other hand, the TSEP [155] is a reactive protocol that responds promptly to changes in relevant parameters of interest and incorporates three levels of heterogeneity. TSEP employs threshold-based CH selection. The combination of three levels of heterogeneity and its reactive routing approach contributes to an extended stability period and increased network lifetime when compared to SEP.

The DEEC Protocol is a distributed energy-efficient clustering algorithm designed for heterogeneous WSNs [156]. It selects CHs based on a probability function derived from each node's residual energy ratio to the network's average energy level. DEEC utilizes the network's average energy as a reference point for this calculation, eliminating the need for global knowledge of energy levels during each election round. DEEC demonstrates superior performance, particularly in multi-level heterogeneous WSNs, when compared to LEACH and SEP. In contrast, the DEC protocol [157] employs a deterministic approach to elect CHs, relying on residual energy levels. DEC outperforms probabilistic-based protocols like LEACH, NEAP, and SEP for minimizing energy consumption.

After developing routing protocols for WSNs with static sensor nodes, it became evident that these clustering protocols were unsuitable for networks with mobile sensor nodes as they did not account for node mobility after clustering. To address this, the LEACH-Mobile (LEACH-M) protocol [158] was

introduced. LEACH-M is similar to LEACH regarding CH formation and selection but incorporates membership declaration to handle node mobility. This addition ensures that sensor nodes are included during the steady-state phase, thus improving packet delivery rates, albeit with increased control overhead. Another protocol, the enhanced LEACH mobile (E-LEACH), proposed by Santosh *et al.* [159], utilizes a mobility metric known as remoteness. A CBR protocol for mobile sensor nodes was introduced by Jiguo *et al.* [160] and employs adaptive TDMA scheduling and round-free CHs to accommodate the network's dynamic traffic and mobility conditions. S. Deng *et al.* presented the MBC protocol [161] to enhance the performance of MWSNs. MBC operates in two phases per round: the setup and steady-state phases. In the setup phase, sensor nodes elect themselves as CHs based on energy levels and mobility. Furthermore, the protocol considers connection time when constructing clusters, creating more reliable paths depending on link stability or availability between MN and CH nodes. During the steady-state phase, sensor node data is transferred to the sink through both intra-cluster and inter-cluster communication.

2.5 Nature inspired optimization algorithms based routing protocols for WSNs

In the past decade, researchers have increasingly turned to heuristic-based clustering approaches, which have proven highly effective. These methods facilitate the even distribution of CHs throughout the network, leading to an energy-efficient network with optimal solutions for selecting the best CH nodes and cluster sizes. Various optimization techniques have been employed, such as GA, ACO, PSO, and others. Each method employs different parameters within its fitness functions to achieve specific objectives. Clustering is classified as an NP-hard optimization problem, and different algorithms exhibit varying performances when solving this problem. In addition to energy efficiency, the convergence time is a critical metric in evaluating performance.

A GCA was introduced to dynamically form clusters in WSNs to prolong network lifetime by minimizing energy dissipation [162]. Khalil *et al.* proposed an EAERP for clustered heterogeneous WSNs [163], redefining key features of evolutionary algorithms to provide more robust results than existing heuristics. Tillet *et al.* developed a method using PSO [164] to balance the number of nodes and candidate CHs in each cluster to minimize energy consumption while maximizing data transmission. Salehpour *et al.* introduced an efficient routing algorithm for large-scale cluster-based WSNs using ACO [165], employing two routing levels: intra-cluster and inter-cluster.

Recently, nature-inspired optimization algorithms like DE [166], SMO [167], HS [168], FPA [169], ISSA [170] and others have been employed to develop clustering protocols for energy-efficient WSNs.

The careful selection of CH nodes remains a pivotal element in WSNs, and current research is dedicated to reducing energy consumption during data retrieval, given the constraints of SNs limited energy supply. The judicious choice of the optimization algorithm, in combination with a well-suited fitness function, can effectively harmonize energy consumption and extend the overall network lifetime.

2.6 Research gaps

On study the literature review, the following gaps are found:

- From the literature, it has been observed that evolutionary algorithms suffer from the problem of premature convergence. For example, DE has a deficiency of premature convergence and stagnation [171].
- The SI algorithm PSO can get a good solution at a significantly faster rate but, compared to other optimization techniques, it is weak to refine the optimum solution, mainly due to less diversity in later search.
- The SSA also suffers from unbalanced exploration and exploitation operations and has a slow convergence [172]. On the other hand, in NMRA, the search space is governed by a simple scaling factor, and the algorithm may stuck in some optima or, overall, a poor exploration operation [4].
- In CS, it has been found that the step size generated by the Levy distribution is an organized random moment and corresponds to better exploration properties but not exploitation properties. This is due to the significant steps generated by Levy distribution, which may skip the global minimum and start moving toward some sub-optimal local minimal solution.
- Due to the local optima stagnation problem in the algorithms, they become slow and less reliable in their functioning. So, a proper balance between the exploration and exploitation phases is required to make the algorithm fast, accurate and more reliable for higher dimensional problems.
- In the literature, some algorithms are found to be suitable for one problem while not fit for other problems. So, a general algorithm must be developed to solve most optimization problems with accurate and better results.
- In most of the routing protocols, heuristic techniques are applied to stationary WSNs because many applications of WSNs, like atmospheric monitoring and battlefield surveillance, are found

to be static. On the other hand, some WSN applications are event-based, like habitat monitoring and tracking-based applications in which the event is mobile. So, mobility has become a critical issue, and much research is required to improve the performance of mobile WSNs.

2.7 Inferences and summary

This chapter delves into the realm of numerical optimization techniques, offering a comprehensive mathematical framework applicable to various optimization problems. It underscores the wide-reaching relevance of optimization in nearly every research domain, which has consequently spurred a surge of interest among researchers in developing highly effective optimization methodologies. Over the past few years, there has been a growing trend in utilizing nature-inspired algorithms to tackle complex optimization problems. These algorithms encompass a range of meta-heuristic approaches, often employing swarm intelligent methods that combine global and local search strategies. Algorithms like DE, PSO, CS, MPA, NMRA, SMA, WOA, and COOT optimization have emerged as highly effective optimization techniques for addressing competitive optimization problems.

Although various algorithms reported in the literature are known for their simplicity, flexibility, and adaptability to a wide range of optimization problems. It is important to recognize that these algorithms excel in some scenarios while underperforming in others. The NFL theorem has validated this variation in performance across different problems [173]. This theorem serves as a motivating factor for researchers to explore and develop new algorithms geared toward solving optimization problems. Another compelling reason for developing new algorithms lies in the inherent limitations of existing ones. These limitations include issues like premature convergence and local optima stagnation observed in population-based algorithms such as DE and PSO [174] [175], poor exploration properties in NMRA [4] and WOA [25], poor exploitation properties of SMA [5] and slow convergence rate of COOT optimization [6]. The NFL theorem essentially implies that any algorithm excelling in one class of problems will have corresponding weaknesses in another class. Therefore, researchers are prompted to either combine the strengths of existing algorithms to propose hybrid optimization algorithms or innovate entirely new algorithms to quantitatively address real-world problems. The challenge lies in designing algorithms that deliver high performance, identifying problem classes where these algorithms are most effective, and developing techniques that enhance their functionality [173]. Achieving high performance often hinges on selecting appropriate initial parameters that balance exploration and exploitation during problem-solving. The quality of parameter selection directly influences algorithm convergence. To address these limitations effectively, assessing, hybridizing, and designing new, robust algorithms for solving optimization problems is imperative.

The above-mentioned considerations set a clear path for further investigation in our proposed research. Our overarching objective is to pioneer new and promising algorithms by elevating the performance of existing ones through innovative approaches. These enhanced algorithms are intended to address both recently developed benchmark test suites and real-world applications effectively. In our research endeavor, our primary focus lies in developing highly efficient techniques by augmenting the performance of well-established algorithms such as NMRA, SMA, WOA, and COOT optimization. After this, these algorithms are applied to the real optimization problem of designing clustered routing protocols for WSNs in both static and mobile environments. All these newly proposed algorithms share a common objective: to outperform existing algorithms documented in the literature by producing superior results in terms of benchmark test suites and real-world applications.

Chapter 3

The Effect of Chaotic Mapping on Naked Mole-rat Algorithm for Energy-efficient WSN

In this chapter, chaos theory has been introduced to improve the exploration capabilities of NMRA. Additionally, we incorporated parametric adaptation to assess the impact of the mating factor (λ) of NMRA by employing various chaotic maps like the Chebyshev map, iterative map, tent map, and others. Based on these chaotic maps, we proposed ten different variants of NMRA and evaluated their performance using the CEC 2005 test suite under different population and dimension sizes. Furthermore, we selected the best-performing variant among these proposed versions and evaluated it using the CEC 2014 benchmark functions, numerical problems from CEC 2019, and three engineering design problems. To assess the effectiveness of the selected best variant, we applied it to the real-time problem of designing energy-efficient and long-lasting WSNs. We compared the results of this best-proposed variant with those of other state-of-the-art algorithms. The experimental results confirm the superior performance of our proposed NMRA with the Chebyshev map (CNMRA), as evidenced by the Wilcoxon rank-sum test and the Friedman rank (f-rank) test when compared to the other algorithms used for comparison.

3.1 Introduction

NMRA follows the matting pattern and eusocial behavior of NMRs[4]. Due to the eusocial behavior [176] of NMRs, they are found to live in a group of approximately 295 members, and the average size of a single colony is about 70 [177]. Here, it should be noted that a single female leads the entire colony and acts as a breeding female to breed with fewer male members called breeders. The remaining male members, called workers, must perform other tasks such as construction, maintenance, and defense from predators. The NMRA primarily utilizes two key parameters: one to regulate the breeding frequency of individuals and the other to update the fitness evaluation of these individuals. This streamlined configuration with minimal tuning parameters results in NMRA having a straightforward architecture and requires less computational time for addressing optimization challenges.

NMRA is considered a strong candidate in NIA but suffers from a rapid convergence rate and maintains a well-balanced approach between exploration and exploitation phases. Within NMRA, exploration tasks are undertaken by workers, facilitating an efficient exploration of the search space,

while breeders are responsible for exploitation, steering the algorithm toward a globally optimal solution. To address these issues, various research endeavors have been dedicated to enhancing NIA's exploration and exploitation aspects, integrating chaotic sequences, as discussed by [178]. Over the years, chaotic sequences have drawn more attention and have been applied to different fields such as synchronization [179] and optimization algorithms [180]

Previously, Chaotic maps were used to tune different parameters of optimization algorithms such as GA [181], PSO [182], ACO [183], ABC [184], FA [185], HS [184], MFO [186] and SSA [187]. Combining chaos with different algorithms provides better results when the right set of chaotic maps is integrated with the algorithm. Here, better results are generated due to chaos's high-level mixing capability, and the algorithm's fixed parameter can be replaced with a chaotic map to find better results. The newer solutions generated by the algorithm may have more diversity and mobility. Due to this reason, it may be useful to introduce chaos to newly proposed meta-heuristic nature-inspired optimization algorithms.

Here, the performance of standard NMRA has been investigated for various values of breeding probability for CEC 2005 benchmark problems [7]. Further, ten different chaotic maps are integrated with standard NMRA to analyze their effects on NMRA. Ten NMRA variants, namely chaotic NMRA (CNMRA1 to CNMRA10), have been proposed with the help of these maps. The working capability of all the versions is tested on CEC 2005 numerical problems [7] for different population and dimension sizes. Moreover, the best version among the proposed versions is also compared with other well-known algorithms for CEC 2005 [7], CEC 2014 [8], CEC 2019 benchmark problems [9] and three design problems [188]. Convergence profiles and statistical tests also validate the simulated results of the proposed version.

The chapter is divided into seven different sections: Section 3.1 deals with the introduction, and a detailed description of ten chaotic maps required to generate chaotic sequences for NMRA is given in Section 3.2 and in Section 3.3 proposed variants of NMRA have been discussed. Section 3.4 presents results and discussion for benchmark functions, results for different breeding probabilities and comparison of results among proposed variants and other algorithms have been discussed. Section 3.5 provides the detail of three engineering design optimization problems and evaluates the performance of the proposed CNMRA for these problems. In Section 3.6, performance of proposed CNMRA is analyzed for smart city wireless sensor network. The chapter summary is presented in Section 3.7.

3.2 Chaotic maps

The meta-heuristic optimization algorithms require uniform random sequences, which are generated using some probability distribution functions, such as uniform or Gaussian distributions. The quality of sequence is used to determine the reduction in computational time for getting the desired accuracy [189]. In principle, chaos is deterministic and exhibits the properties of randomness, which is non-linear in nature [184]. Generating a chaotic sequence is easy and fast [190], but it requires some chaotic maps and initial conditions for a long sequence. Here, simple random sequences are replaced with chaotic maps due to better dynamic and statistical properties of chaos [191]. The chaotic sequence has been proven to provide good results in applications such as the transmission of data securely [192], [193], nonlinear complex systems [194], computing of DNA [195], image processing [196] and others. Chaos exhibits mainly three properties: quasi-stochastic, ergodicity, and sensitivity to initial conditions. By using these properties, it can be a guarantee that generated solutions are diverse in nature [197]. The quasi-stochastic property states that simple random variables are replaced with chaotic maps. Ergodicity property defines chaotic maps' ability to search all states non-repeatedly within some certain range. Finally, the property of sensitivity to initial conditions defined as little change in initial points may lead to a non-linear change in future behavior. So, the combination of all these properties enhances the performance of optimization algorithms significantly [198].

This chapter uses ten non-invertible, one-dimensional chaotic maps to create potential solutions for NMRA variants. The ten chaotic maps in which c_n is the n^{th} number of chaotic sequence and n denotes the chaotic sequence c index. These maps with different characteristics are given as:

(i) *Chebyshev map*: The Chebyshev map family is described as [180]:

$$c_{n+1} = \cos(k \cos^{-1}(c_n)) \quad (3.1)$$

(ii) *Circle map*: The definition of circle map [199] is given by following equation:

$$c_{n+1} = c_n + b - \left(\frac{a}{2\pi}\right) \sin(2\pi c_n) \bmod(1) \quad (3.2)$$

where $a = 0.5$ and $b = 0.2$ to generate a chaotic sequence in range of $[0,1]$.

(iii) *Gauss/Mouse map*: The generation of Gaussian map [200] is defined by equation as:

$$c_{n+1} = \begin{cases} 0 & c_n = 0 \\ \frac{1}{c_n \bmod(1)} & otherwise \end{cases} \quad (3.3)$$

where,

$$\frac{1}{c_n \bmod(1)} = \frac{1}{c_n} - \left\lfloor \frac{1}{c_n} \right\rfloor \quad (3.4)$$

which is also used to generate a chaotic sequence within range of $[0,1]$.

(iv) *Iterative map*: The following equation is used to describe the iterative map with infinite collapses [201]:

$$c_{n+1} = \sin\left(\frac{a\pi}{c_n}\right) \quad (3.5)$$

where a is a suitable parameter in range of $[0,1]$.

(v) *Logistic map*: The logistic map is the simplest map that appears in non-linear dynamics of biological population evidencing behavior of chaos [202], and described by the equation as:

$$c_{n+1} = ac_n(1 - c_n) \quad (3.6)$$

where, c_n is n th chaotic number, while n denotes the iteration number. Here, $c \in [0,1]$ with initial condition $c_0 \in [0,1]$ and that $c_0 \notin [0.0,0.25,0.5,0.75,1.0]$. The value of $a=4$ is used in later simulations.

(vi) *Sawtooth map*: The formulation of the sawtooth map can be done by the mod function [203] such as:

$$c_{n+1} = 2c_n \bmod(1) \quad (3.7)$$

(vii) *Sine map*: The following equation is used to define uni-modal sine map [204]:

$$c_{n+1} = \frac{a}{4} \sin(\pi c_n) \quad (3.8)$$

where, a is the parameter in the range of $0 < a \leq 4$.

(viii) *Singer map*: The singer map is defined for one dimensional system and is given by [205]:

$$c_{n+1} = \mu(7.86c_n - 23.31c_n^2 + 28.75c_n^3 - 13.3c_n^4) \quad (3.9)$$

where, μ is suitable parameter defined between 0.9 and 1.08.

(ix) *Sinusoidal map*: The chaotic sequence generated by sinusoidal map act as a sinusoidal iterator [202] and it is represented as:

$$c_{n+1} = ac_n^2 \sin(\pi c_n) \quad (3.10)$$

As a special case when $a = 2.3$ and $c_0 = 0.7$ then it can be represented in simplified form, which also generates chaotic sequence in the range of $[0,1]$ as:

$$c_{n+1} = \sin(\pi c_n) \quad (3.11)$$

(x) *Tent map*: This map resembles a logistic map with some specific effects of chaos [206]. It also generates chaotic sequence in $[0,1]$ by using following equation:

$$c_{n+1} = \begin{cases} \frac{c_n}{0.7} & c_n < 0.7 \\ \frac{10}{3}(1 - c_n) & c_n \geq 0.7 \end{cases} \quad (3.12)$$

3.3 Chaotic naked mole-rat algorithm

In this section, chaotic maps are utilized to improve the working ability of basic NMRA [4] by replacing random sequences with chaotic sequences and linearly decreasing population size with iterations. The classical NMRA encounters issues related to its slow convergence speed and an unequal distribution of effort between exploration and exploitation operations. This algorithm performance is mainly affected by one parameter λ . As seen from equations (2.3) and (2.4), λ is a main parameter which influences new solutions generated for worker and breeder rats. It significantly impacts the balance between exploration and exploitation. A good algorithm should balance these two phenomena properly to approach the near-global optimum. Here, chaotic maps modify λ parameter of equation (2.3) and (2.4) during each iteration and modified equation for workers and breeders are defined as:

$$x_i^{t+1} = x_i^t + CM_i^t(x_j^t - x_k^t) \quad (3.13)$$

$$y_i^{t+1} = (1 - CM_i^t)y_i^t + CM_i^t(NMR_{best} - y_i^t) \quad (3.14)$$

where, CM_i^t is chaotic map value for i^{th} rat (worker or breeder) at t^{th} iteration. The new solution for workers and breeders is generated according to equations (3.13) and (3.14) in which random number λ in $[0,1]$ is replaced with chaotic number in the range of $[0,1]$. All the chaotic maps discussed in Section 3.2 are applied to basic NMRA according to equations (3.13) and (3.14) such as Chebyshev map (CNMRA1), circle map (CNMRA2), gauss map (CNMRA3), iterative map (CNMRA4) and logistic map (CNMRA5). In the same way rest of the chaotic maps are also applied to standard NMRA such as sawtooth map (CNMRA6), sine map (CNMRA7), singer map (CNMRA8), sinusoidal map (CNMRA9) and tent map (CNMRA10). The embedding of these maps for solution generation

can enhance the performance and convergence speed of basic NMRA in some cases and discuss in the results section in detail. Here, the same structure has been followed in all the proposed variants with different types of chaotic maps; hence, different names are assigned. The pseudo-code for all the algorithms is given in Algorithm 4.

Algorithm 4: Pseudo-code of chaotic NMRA

Begin:

Define: *naked mole-rats population size:* (N)

number of breeders: $B=N/5$

number of workers: $W=N-B$

breeding probability: bp

D —*Objective function's dimension size:* $f(x)$

evaluate and find overall best: R_{best}

while *iteration* < *maximum iterations* (t_{max})

generate chaotic sequence using chaotic maps: CM

for $i=1:W$

execute worker phase using $x_i^{t+1} = x_i^t + CM_i^t(x_j^t - x_k^t)$

evaluate: x_i^{t+1}

end for

for $i=1:B$

if *random number* $[0,1] > bp$

execute breeder phase using $y_i^{t+1} = (1 - CM_i^t)y_i^t + CM_i^t(R_{best} - y_i^t)$

evaluate: y_i^{t+1}

end for

combine new population of worker and breeder

evaluate population

update: R_{best}

update t_{max}

end while

save the final best: R_{best}

End

3.3.1 Computational efficiency

Computational efficiency, in the context of testing algorithms, pertains to the run-time complexity of the algorithm under test. In the case of NMRA, its run-time complexity can be represented as $O(N \cdot D \cdot t_{max})$, where t_{max} signifies the maximum number of iterations, N corresponds to the population size of mole-rats, and D represents the dimensionality of the problem being analyzed. The primary objective in determining computational complexity is identifying the most challenging scenarios for NMRA and assessing its performance under specific conditions. The equation is derived from the generalization that, for any problem with a dimensionality of D and a population size of N , the complexity for each member of the population is expressed as $O(D)$ and for the entire population size of N , it is denoted as $O(N \cdot D)$. The algorithm's overall complexity changes to $O(N \cdot D \cdot t_{max})$ since a total of t_{max} iterations must be performed. Based on the complexity study, the population

size, dimension size, and overall number of function evaluations significantly impact how well NMRA performs. Except for population size variations, the proposed CNMRA's computational complexity is comparable to the basic NMRA. Here in CNMRA the population size is linearly reducing and is given by $N_{reduced}$ whereas all other properties remain the same. Therefore, $O(N_{reduced} \cdot D \cdot t_{max})$ is the expression for the computational complexity of CNMRA. As a result, it can be said that the CNMRA is more computationally efficient than the standard NMRA with a smaller population.

3.4 Result analysis for test suites

The performance analysis of proposed NMRA variants is reviewed in this section to see if the proposed variants improve the performance of the original NMRA or not. The simulation results of proposed variants are compared, and the best among them is then compared to various optimization techniques for CEC 2005 numerical benchmark problems [7]. The alternative sets of breeding probability are employed to find the best set on which the algorithm produces good results. A comparison is made between the proposed variants for three different population and dimension size values, and its effect has been analyzed. After the selection of optimal population and dimension size, the best variant is selected, and a comparison with other algorithms has been performed for CEC 2014 [8], 100-digit challenge (CEC 2019) [9] benchmark functions and engineering design optimization problems.

Apart from this, all the proposed variants are run on Dell PC with AMD Ryzen 5 processor, 8.00GB RAM, Windows 10 (x64-bit) OS and MATLAB 2016a version.

3.4.1 Test suite

The efficiency of proposed NMRA variants using chaotic maps has been tested using 12 CEC 2005 [7] numerical functions. A detailed explanation of these functions is provided in Table A.1. These functions are divided into uni-modal functions ($F1$ to $F7$) and multi-modal functions ($F8$ to $F12$). The uni-modal functions have only one global optimum and no local optima (i.e. they are simple to solve) and are used to demonstrate the algorithm's exploitation tendencies. On the other hand, multi-modal functions contain many local minima and are useful for assessing the exploitation and exploration abilities of suggested chaotic map versions. The optimal fitness value for uni-modal and multi-modal functions is $f_{min} = 0$. Note that all these function values are taken from the basic paper, and no alteration has been done in the basic structure of these test functions [7].

3.4.2 Parameter settings

This section delves into the specifics of many algorithms used to compare with the suggested approach. The algorithms under comparison for CEC 2005 [7] test suite are OEWOA [207], JADE [69], SaDE [208], SHADE [209], GWO-E [210], SCCSA [211], FA-FPO [212], EO [213], GSSA [214], CMA-ES [215], NMRA, LSHADE-SPACMA [216], and proposed CNMRA. In the same way, other algorithms under comparison for CEC 2014 [8] benchmark problems are RW-GWO [217], B-BBO [218], LX-BBO [219], ISOS [220], PBIL [221], VNBA [222], MGSCA [223], IMEHO [224], CCS [225], BDE [226], NMRA and proposed CNMRA. The different parameters of these algorithms are taken from their corresponding papers and presented in Table 3.1. The JADE algorithm uses two new parameters, namely parameter adaption control rate (c) and mutation strategy greediness (p), and these parameters are combined with the original DE parameters crossover control parameter (CR) and amplification factor (F). The performance of JADE is found to be best for values $1/c \in [5, 20]$, $p \in [0.05, 0.20]$, $F=0.5$ and $CR=0.9$ as given in [69]. In the case of SaDE, both parameters F and CR is made to be self-adaptive [208]. For OEWOA, parameter $\vec{\alpha}$ follows exponential decreasing function [207]. In the case of GWO-E and RW-GWO algorithms, $\vec{\alpha}$ is the main parameter to decide the ability of the algorithm to perform exploration and exploitation operations. This parameter provides good results for linearly decreasing values ranging from 2 to 0, while other parameters are random numbers [23]. For SCCSA, three parameters (r_1 , r_2 and r_3) are used which are random numbers in $[0, 1]$ and same as [211]. In case of MGSCA, three parameters r_1 , r_2 and r_3 are responsible for the performance of the algorithm. The parameter r_1 plays an important role in balancing exploration and exploitation phases due to its adaptive nature. The parameter r_2 is defined in the range of $[0, 2\pi]$ and responsible for the movement of a solution towards global optima. The last parameter r_3 is uniformly distributed in the range of $[0, 2]$ and contributes to the algorithm's exploration and exploitation operations. For BDE, amplification factor (F) and crossover control rate (CR) parameters follow the beta distribution instead of constant values [226]. In the case of EO, the ability of the exploration phase is controlled by parameter α_1 , the ability of the exploitation phase is controlled by the parameter α_2 and the direction of these two phases are controlled by parameter $sign(r - 0.5)$. The values of all these parameters are kept the same as the original article [213]. The value of all important parameters such as P_{best} , ARC rate, learning rate (c) and probability variable (FCP) are taken from [213] for SHADE and LSHADE-SPACMA. In the case of FA-FPO, switching probability factor (S) and derivative order α are mainly responsible parameters for evaluating the algorithm's efficiency. So, in the present case, the value of α is taken as 0.1, and S is considered as self-adaptive. Three parameters primarily determine the performance of MPA: the step size control parameter (a), the constant number (p), and the random vector ($\vec{r}$). All of these parameters are set to the same

value as the standard MPA in [91]. For LX-BBO and B-BBO, two important parameters, such as emigration, impact the algorithms' performance on the performance of algorithms. The maximum values of these parameters are set to 1 as used by [219]. In the case of ISOS, parasitism parameter (q) is responsible for deciding the percentage of function evaluations used for exploration and exploitation phases, and r is a random number in $[0,1]$. For CCS, PBIL, VNBA, and IMEHO, the values of all the parameters involved in these algorithms are kept the same as [224]. As far as NMRA is concerned, there are only two important parameters: breeding probability (bp), and the second one is mating factor (λ) and used for evaluating new solutions of mole rats. In original NMRA [4], the value of bp is 0.5, and λ is a random number in the range of $[0,1]$, while in proposed CNMRA, the value of bp is varying from 0.05 to 0.90 and λ is a chaotic sequence generated through chaotic maps.

Table 3.1: Parameter settings

Algorithm	Parameters
SaDE [208]	F, CR = self adaptive
OEWOA [207]	$\bar{\alpha}$ = Exponentially decreasing function; $b = 1$
SCCSA [211]	$r_1, r_2, r_3 = [0, 1]$
GWO-E [210]	$\bar{\alpha}$ = Linearly decreasing [2,0]
FA-FPO [227]	$\alpha = [0.1, 1]$, $S = adaptive$
SHADE [213]	$P_{best} = 0.1$, $ARC\ rate = 2$
LSHADE-SPACMA [213]	$c = 0.8$, $P_{best} = 0.11$, $ARC\ rate = 1.4$, $FCP = 0.5$
EO [213]	$a_1 = 2$, $a_2 = 1$, $sign(r - 0.5)$
GSSA [214]	$c_1, c_2, c_3 = constant$
JADE [69]	$F = 0.5$; $CR = 0.9$; $1/c = [5, 20]$; $p = [0.05, 0.20]$
LX-BBO [219]	$H = 1$; $I = 1$
VNBA [224]	Loudness (A)=0.5; Pulse rate (r)=0.5
ISOS [220]	$q \in [1, 100]\%$, $r \in [0, 1]$
B-BBO [219]	$H = 1$; $I = 1$
RW-GWO [217]	$\bar{\alpha}$ = Linearly decreased from 2 to 0
PBIL [224]	Learning rate (LR)=0.1; Mutation probability (p_m)=0.02
CCS [224]	Switch probability (pa)=0.25
BDE [226]	F, CR = Beta distribution
IMEHO [224]	Inertia weight (w)= Linearly decreasing $[0.9, 0.2]$; $\alpha \in [0, 1]$; Probability (p_c)=0.05
MGSCA [223]	r_1 = adaptive; $r_2 = [0, 2\pi]$; r_3 = uniformly distributed $[0, 2]$
CS [82]	Switch Probability (pa)=0.5
SSA [24]	c_1, c_2, c_3
ASSA [228]	c_1, c_2, c_3 ; adaptive population
GWO [23]	α = linearly decreased from 2 to 0
DA [229]	$w, s, a, c, f, e = constant$
FPA [230]	$P = 0.8$
NBA [188]	$A = 0.5$; $r = 0.5$; $\alpha = \gamma = 0.9$
TLBO [188]	Teaching factor (T_F) = rand $[1, 2]$
LebTLBO [188]	$CR = 0.9$; $min_LE = 0.3$; $F = 0.9$
NMRA	$bp = 0.5$; $\lambda = [0, 1]$
CNMRA	$bp = 0.05$; λ = Chaotic Sequence

3.4.3 Effect of breeding probability

To analyze the effect of bp of breeders on standard NMRA, five different values of bp are used as given in Table 3.2. All the values of bp are tested on uni-modal and multi-modal benchmark functions of CEC 2005. All of the results are shown as mean and standard deviation values for 51 runs with a population size of 50, a dimension size of 30, and an algorithm iteration size of 500. For the procedure under test to calculate the mean and standard deviation of best values, the number of runs is set at 51. The performance of basic NMRA degrades as the bp value increases for problems ($F1$ to $F4$). The findings for $F5$, $F6$, $F7$, $F11$, and $F12$ at different sets of bp , on the other hand, are nearly identical. For all values of bp except 0.90, the algorithm can approach the global optimal solution for problems ($F8$ to $F10$). Overall, it has been determined that the algorithm's performance is optimal

at $bp = 0.05$ for all benchmark functions except $F6$ and $F11$.

Table 3.2: NMRA simulation results for different breeding probability values for CEC 2005 ($D = 30$)

Function	bp=0.05		bp=0.25		bp=0.50		bp=0.75		bp=0.90	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	1.241E-177	0	2.820E-138	1.192E-137	6.681E-83	4.722E-82	2.961E-40	1.6711E-39	4.211E-14	1.431E-13
F2	2.430E-89	1.160E-88	3.930E-70	1.670E-69	3.450E-45	1.601E-44	3.871E-22	8.142E-22	9.402E-08	2.420E-07
F3	3.120E-179	0	8.162E-137	5.372E-136	1.122E-86	5.471E-86	1.471E-40	4.831E-40	1.681E-12	8.252E-12
F4	5.152E-89	3.282E-88	4.440E-70	1.410E-69	4.402E-45	1.982E-44	5.652E-22	1.591E-21	1.461E-07	3.381E-07
F5	2.892E+01	1.511E-02	2.891E+01	1.641E-02	2.891E+01	1.950E-02	2.891E+01	2.160E-02	2.890E+01	2.650E-02
F6	6.790E+00	4.940E-01	6.630E+00	5.820E-01	6.732E+00	5.441E-01	6.682E+00	5.111E-01	6.712E+00	6.841E-01
F7	2.882E-04	1.862E-04	4.112E-04	3.471E-04	7.161E-04	5.461E-04	1.500E-03	1.200E-03	4.100E-03	3.101E-03
F8	0	0	0	0	0	0	0	0	2.771E+01	6.011E+01
F9	8.881E-16	0	8.881E-16	0	8.881E-16	0	8.881E-16	0	5.011E-08	1.692E-07
F10	0	0	0	0	0	0	0	0	1.732E-07	1.222E-06
F11	1.131E+00	2.960E-01	1.091E+00	2.720E-01	1.152E+00	2.610E-01	1.092E+00	2.481E-01	1.092E+00	2.371E-01
F12	2.991E+00	4.401E-02	2.991E+00	2.602E-03	2.991E+00	2.002E-03	2.991E+00	2.991E+00	2.991E+00	4.5022E-02

3.4.4 Effect of population size

This section examines the performance of ten proposed NMRA variants across 12 benchmark functions ($F1 - F12$) with varying population sizes. We utilize three population size sets (P) – 30, 60, and 90 alongside a fixed dimension size of 30 and 500 iterations for all NMRA variants. The results, based on the mean and *Std* values of 51 runs, are presented in Table 3.3, and discuss the outcomes as follows:

Population size 30: CNMRA1 stands out as the best performer for $F1$, $F2$, $F3$, and $F4$ among all the variants. However, CNMRA2 and CNMRA7 also produce competitive results. For $F5$, CNMRA1 slightly outperforms other options in terms of *Std* values. In the cases of $F6$ and $F11$, all versions yield nearly identical results, making it challenging to identify a preferred choice. For $F7$, CNMRA2 excels in both mean and *Std* values, although CNMRA1, CNMRA2, and CNMRA7 exhibit comparable performance. CNMRA1, CNMRA2, and CNMRA7 variants achieve global optima for $F8$ and $F10$. Similarly, for $F9$, these same variants approach global minima. For $F12$, CNMRA2 performs better than other variants. In summary, CNMRA1 is the top performer for eight functions, CNMRA2 for five functions, CNMRA7 for three functions, while other variants do not excel in any particular function. Overall, CNMRA1 delivers the best performance for a population size of 30.

Population size 60: CNMRA1 outperforms other variants for $F1$, $F2$, $F3$, and $F4$, although CNMRA2 and CNMRA7 are competitive. For $F5$, CNMRA7 slightly surpasses the others in terms of *Std* values. CNMRA3 excels in *Std* values for $F6$. CNMRA7 is the leader for $F7$ in both mean and *Std* values. CNMRA1, CNMRA2, and CNMRA7 variants achieve global optima for $F8$ and $F10$. For $F9$, CNMRA1, CNMRA2, and CNMRA3 variants attain global minima, while CNMRA6 performs slightly better regarding mean values for $F11$. For $F12$, CNMRA1 outperforms other variants in terms of standard deviation. To summarize, CNMRA1 is the best performer for eight functions,

CNMRA2 for three functions, CNMRA3 for one function, CNMRA6 for one function, and CNMRA7 for four functions. CNMRA1 maintains its overall top performance for a population size of 60.

Population size 90: In this context, CNMRA1, CNMRA2, and CNMRA3 come close to achieving global optima for $F1$, $F2$, $F3$, and $F4$, with CNMRA1 emerging as the best choice. For $F5$, CNMRA1, CNMRA2, and CNMRA7 outperform other variants. Results for $F6$ and $F11$ do not exhibit significant differences. CNMRA2 leads the way for $F7$ in mean and Std values. CNMRA1 and CNMRA7 attain global optima for $F8$, while CNMRA2 and CNMRA7 achieve the same for $F10$. $F9$ results show that CNMRA1, CNMRA2, and CNMRA7 variants reach global minima, while CNMRA6 slightly outperforms the mean values for $F11$. For $F12$, CNMRA1 excels in standard deviation values. To summarize, CNMRA1 is the top performer for nine functions, CNMRA2 for three functions, and CNMRA7 for four functions. Overall, CNMRA1 demonstrates the best performance for a population size of 90.

Inferences: It is evident from the results across different population sizes that performance improves as population size increases. The proposed variants exhibit enhanced performance as the population size approaches 60. It is important to note that as population size increases, the total number of function evaluations also rises. Therefore, for a population size of 60, the CNMRA1 variant is the best choice due to its well-tuned parameter λ , which aligns perfectly with the Chebyshev chaotic map compared to other chaotic maps.

3.4.5 Effect of dimension Size

This section assesses the performance of ten proposed NMRA variants based on chaotic maps across various dimension sizes (10, 30 and 50) using 12 benchmark functions ($F1 - F12$). These evaluations are conducted with a population size of 50 and 500 iterations, and each variant is tested across 51 runs. The results are summarized as follows:

Dimension size 10: Table 3.4 presents the results for dimension size 10. Among the variants, CNMRA1, CNMRA2, and CNMRA7 achieve values close to the global optimal solution for $F1$, $F2$, $F3$, and $F4$, with CNMRA1 exhibiting the best performance. For $F5$, CNMRA1 also performs better in terms of Std values. Benchmark functions $F6$ and $F11$ yield nearly identical results across all variants. For $F7$, CNMRA1, CNMRA2, and CNMRA7 show competitive mean and Std values. These three variants provide exact optimal solutions for $F8$ and $F10$. CNMRA1 and CNMRA2 also outperform other variants for $F9$. CNMRA3 performs somewhat better in terms of Std values for $F12$. In summary, CNMRA1 is the top performer for dimension size 10, achieving good results for nine out of twelve functions.

Table 3.3: Simulated outcomes of chaotic variants for population size (30, 60, 90)

Function	Variant	$P = 30$		$P = 60$		$P = 90$		Function	Variant	$P = 30$		$P = 60$		$P = 90$	
		Mean	Std	Mean	Std	Mean	Std			Mean	Std	Mean	Std	Mean	Std
F1	CNMRA1	8.482E-179	0	1.992E-240	0	1.891E-217	0	F7	CNMRA1	1.001E-03	1.101E-03	4.301E-04	4.451E-04	2.420E-04	2.160E-04
	CNMRA2	4.670E-1353	270E-1342	160E-188	0	4.270E-184	0		CNMRA2	8.460E-04	7.740E-04	2.661E-04	2.711E-04	2.161E-04	1.541E-04
	CNMRA3	9.291E-02	8.232E-02	4.982E-02	3.002E-02	3.992E-02	2.112E-02		CNMRA3	4.072E-00	7.803E-01	3.522E+00	6.533E-01	3.161E+00	6.043E-01
	CNMRA4	7.003E-02	3.191E-01	7.241E-01	2.031E+00	6.281E-01	1.851E+00		CNMRA4	2.400E+002	1.10E+002	730E+002	0.30E+002	860E+001	612E+00
	CNMRA5	9.691E-04	2.201E-03	2.291E-02	1.261E-01	2.001E-01	8.911E-01		CNMRA5	1.082E+001	952E+001	112E+001	892E+001	732E+001	882E+00
	CNMRA6	1.540E-01	2.510E-01	5.900E-01	1.702E+001	272E+003	472E+00		CNMRA6	2.552E+002	1.11E+002	301E+001	841E+002	741E+001	581E+00
	CNMRA7	2.300E-1581	600E-1571	050E-225	0	5.500E-208	0		CNMRA7	9.810E-04	7.590E-04	2.860E-04	1.860E-04	2.420E-04	2.740E-04
	CNMRA8	3.300E-02	5.820E-02	6.540E-02	1.290E-01	2.840E-01	7.460E-01		CNMRA8	1.451E+001	721E+002	001E+001	851E+001	751E+001	501E+00
	CNMRA9	1.511E+002	451E+001	960E+003	610E+002	160E+004	310E+00		CNMRA9	4.230E+001	100E+003	692E+00	6.980E-01	3.552E+00	8.820E-01
	CNMRA10	6.090E-01	1.442E+005	562E-01	1.592E+001	022E+002	532E+00		CNMRA10	2.652E+002	062E+002	782E+001	702E+002	632E+001	622E+00
F2	CNMRA1	1.150E-89	8.170E-89	5.560E-1272	690E-1266	310E-1102	710E-109	F8	CNMRA1	0	0	0	0	0	0
	CNMRA2	1.070E-68	7.490E-68	5.570E-99	3.900E-98	5.290E-92	3.720E-91		CNMRA2	0	0	0	0	3.151E+002	231E+01
	CNMRA3	9.891E+002	971E+008	191E+001	351E+008	131E+001	031E+00		CNMRA3	1.821E+021	901E+011	731E+021	571E+011	631E+021	433E+01
	CNMRA4	4.840E-01	6.38-01	3.153E+005	823E+005	913E+007	113E+00		CNMRA4	3.063E+016	453E+012	760E+015	300E+015	560E+016	200E+01
	CNMRA5	7.380E-02	9.180E-02	1.560E+004	650E+002	550E+003	990E+00		CNMRA5	9.450E-01	4.590E+001	080E+012	591E+012	861E+014	901E+01
	CNMRA6	1.521E+001	081E+002	571E+003	211E+006	343E+005	863E+00		CNMRA6	3.393E+014	013E+014	223E+014	773E+017	003E+015	673E+01
	CNMRA7	6.790E-77	4.800E-76	7.390E-1174	470E-1166	180E-1024	360E-101		CNMRA7	0	0	0	0	0	0
	CNMRA8	1.262E+00	8.070E-01	2.862E+004	432E+004	592E+005	222E+00		CNMRA8	2.102E+011	732E+012	022E+012	282E+014	752E+015	772E+01
	CNMRA9	8.612E+003	792E+008	432E+003	432E+001	042E+014	882E+00		CNMRA9	1.172E+024	812E+011	152E+024	442E+011	242E+024	302E+01
	CNMRA10	2.002E+001	1392E+005	562E-01	5.412E+006	492E+005	532E+00		CNMRA10	5.722E+015	182E+014	762E+015	112E+017	072E+016	672E+01
F3	CNMRA1	4.290E-178	0	4.420E-252	0	5.010E-216	0	F9	CNMRA1	8.880E-16	0	8.880E-16	0	8.881E-16	0
	CNMRA2	6.511E-1363	381E-1351	351E-200	0	1.931E-181	0		CNMRA2	8.882E-16	0	8.882E-16	0	8.882E-16	0
	CNMRA3	6.112E-01	7.412E-01	2.341E-01	2.191E-01	2.281E-01	1.212E-01		CNMRA3	1.111E+003	3.151E-01	9.261E-01	2.030E-01	8.530E-01	1.720E-01
	CNMRA4	1.820E-01	7.400E-01	2.740E-01	8.880E-01	1.731E+005	011E+00		CNMRA4	1.580E-01	2/851E-01	5.861E-01	9.491E-01	7.921E-01	1.261E+00
	CNMRA5	6.201E-03	1.911E-02	3.931E-02	1.601E-01	4.891E-01	1.611E+00		CNMRA5	1.652E-02	2.652E-02	3.052E-01	6.912E-01	6.243E-01	1.041E+00
	CNMRA6	1.021E+001	881E+001	171E+001	961E+001	391E+002	421E+00		CNMRA6	5.133E-01	4.453E-01	5.743E-01	6.603E-01	1.291E+001	161E+00
	CNMRA7	1.661E-1607	281E-1602	631E-225	0	1.841E-200	0		CNMRA7	8.881E-16	0	8.883E-16	0	8.883E-16	0
	CNMRA8	5.543E-01	9.303E-01	4.353E-01	7.133E-01	3.673E-01	4.643E-01		CNMRA8	3.283E-01	2.943E-01	5.401E-01	7.001E-01	7.821E-01	9.641E-01
	CNMRA9	1.011E+011	361E+016	631E+007	801E+005	971E+008	321E+00		CNMRA9	1.981E+001	100E+001	670E+001	050E+002	080E+001	100E+00
	CNMRA10	4.660E+001	130E+012	100E+005	640E+002	990E+005	200E+00		CNMRA10	9.901E-01	7.781E-01	6.761E-01	7.122E-01	1.280E+001	190E+00
F4	CNMRA1	6.522E-86	4.462E-85	4.602E-1223	082E-1213	572E-1071	612E-106	F10	CNMRA1	0	0	0	0	7.152E-07	2.652E-06
	CNMRA2	6.072E-69	3.572E-68	5.812E-1013	202E-100	1.392E-91	7.262E-91		CNMRA2	0	0	0	0	0	0
	CNMRA3	1.552E-01	6.392E-02	1.282E-01	5.172E-02	1.112E-01	4.162E-02		CNMRA3	3.562E-06	2.832E-06	2.472E-06	1.181E-06	2.051E-06	1.321E-06
	CNMRA4	5.021E-02	7.311E-02	1.241E-01	1.881E-01	1.251E-01	2.231E-01		CNMRA4	1.091E-05	2.401E-05	1.841E-06	2.561E-06	3.331E-06	8.641E-06
	CNMRA5	1.431E-02	3.161E-02	4.661E-02	1.131E-02	8.081E-02	1.241E-01		CNMRA5	1.991E-06	3.741E-06	1.211E-06	3.011E-06	6.461E-07	1.701E-06
	CNMRA6	1.471E-01	1.321E-01	2.011E-01	1.820E-01	2.830E-01	2.720E-01		CNMRA6	4.800E-05	1.430E-04	5.870E-06	1.580E-05	2.590E-06	5.530E-06
	CNMRA7	1.970E-79	1.350E-78	1.350E-1119	490E-1117	250E-1015	070E-100		CNMRA7	0	0	0	0	0	0
	CNMRA8	9.740E-02	8.870E-02	1.050E-01	9.640E-02	1.350E-01	1.210E-01		CNMRA8	7.370E-06	1.880E-05	5.010E-06	2.010E-05	2.510E-06	6.630E-06
	CNMRA9	3.940E-01	3.870E-01	4.571E-01	3.621E-01	4.071E-01	3.981E-01		CNMRA9	3.261E-04	1.801E-03	3.871E-06	9.251E-06	2.101E-06	7.051E-06
	CNMRA10	2.972E-01	3.132E-01	2.502E-01	2.552E-01	3.232E-01	2.322E-01		CNMRA10	1.602E-05	3.062E-05	1.522E-05	7.642E-05	3.552E-06	7.152E-06
F5	CNMRA1	2.890E+01	1.942E-02	2.890E+01	1.992E-02	2.890E+01	2.152E-02	F11	CNMRA1	1.300E+00	2.442E-01	1.170E+00	2.232E-01	1.150E+00	2.542E-01
	CNMRA2	2.890E+01	2.233E-02	2.890E+01	1.873E-02	2.890E+01	2.363E-02		CNMRA2	1.230E+00	2.783E-01	1.151E+00	2.313E-01	1.121E+00	2.393E-01
	CNMRA3	2.801E+028	061E+028	311E+013	241E+017	731E+014	451E+01		CNMRA3	1.541E+00	1.233E-01	1.451E+00	1.383E-01	1.421E+00	1.333E-01
	CNMRA4	3.091E+014	021E+004	661E+018	811E+012	561E+024	191E+02		CNMRA4	1.141E+00	3.643E-01	9.243E-01	3.113E-01	7.503E-01	2.903E-01
	CNMRA5	2.911E+01	5.333E-01	4.861E+019	491E+011	1.311E+021	1.881E+02		CNMRA5	1.021E+00	2.943E-01	7.693E-01	2.923E-01	6.943E-01	3.393E-01
	CNMRA6	4.731E+013	091E+017	961E+019	911E+012	692E+002	002E+02		CNMRA6	1.042E+00	3.633E-01	7.341E-01	4.001E-01	6.821E-01	3.281E-01
	CNMRA7	2.892E+01	1.651E-02	2.892E+01	1.731E-02	2.892E+01	2.041E-02		CNMRA7	1.062E+00	2.811E-01	1.152E+00	2.981E-01	1.132E+00	2.361E-01
	CNMRA8	3.982E+011	442E+016	182E+011	292E+025	352E+015	122E+01		CNMRA8	1.062E+00	3.301E-01	8.972E-01	3.452E-01	7.562E-01	3.602E-01
	CNMRA9	4.622E+028	802E+023	552E+024	052E+028	522E+021	292E+03		CNMRA9	1.402E+00	3.082E-01	1.192E+00	4.092E-01	1.042E+00	3.662E-01
	CNMRA10	1.142E+021	542E+022	032E+024	721E+022	041E+023	341E+02		CNMRA10	1.161E+00	3.132E-01	8.082E-01	3.282E-01	8.862E-01	3.932E-01
F6	CNMRA1	6.961E+00	4.482E-01	7.011E+00	3.732E-01	6.951E+00	3.402E-01	F12	CNMRA1	2.991E+00	3.412E-02	2.991E+00	2.202E-03	3.001E+00	4.432E-02
	CNMRA2	6.921E+00	5.212E-01	6.811E+00	5.551E-01	6.781E+00	4.631E-01		CNMRA2	2.991E+00	1.601E-03	2.991E+00	2.501E-03	2.991E+00	7.981E-02
	CNMRA3	7.301E+00	2.791E-01	7.121E+00	2.251E-01	7.031E+00	2.221E-01		CNMRA3	3.601E+00	1.311E-01	3.491E+00	1.311E-01	3.481E+00	1.191E-01
	CNMRA4	7.021E+00	4.071E-01	6.741E+001	270E+007	140E+002	000E+00		CNMRA4	3.090E+00	2.853E-01	3.130E+00	3.803E-01	3.190E+00	4.253E-01
	CNMRA5	6.540E+00	5.673E-01	6.310E+00	9.523E-01	6.360E+001	140E+00		CNMRA5	3.000E+00	2.073E-02	3.040E+00	3.483E-01	3.260E+00	5.433E-01
	CNMRA6	7.180E+00	5.513E-01	7.030E+001	390E+007	650E+001	710E+00		CNMRA6	3.280E+00	3.823E-01	3.280E+00	3.123E-01	3.360E+00	4.843E-01
	CNMRA7	6.690E+00	5.023E-01	6.951E+00	4.720E-01	6.801E+00	5.890E-01		CNMRA7	2.981E+00	7.760E-02	3.011E+00	1.310E-01	2.991E+00	2.280E-02
	CNMRA8	7.051E+00	6.260E-01	6.831E+00	6.400E-01	6.841E+001	681E+00		CNMRA8	3.151E+00	1.850E-01	3.181E+00	2.280E-01	3.171E+00	4.600E-01
	CNMRA9	9.021E+002	901E+009	341E+005	011E+008	601E+002	551E+00		CNMRA9	4.152E+00	9.260E-01	3.952E+00	7.010E-01	3.962E+00	8.050E-01
	CNMRA10	7.502E+00	5.600E-01	7.552E+001	832E+008	242E+002	882E+00		CNMRA10	3.442E+00	5.170E-01	3.502E+00	5.480E-01	3.692E+00	7.860E-01

Functions $F6$ shows similar performance across all variants. CNMRA1 delivers strong results for $F7$ in mean and Std values. CNMRA1, CNMRA2, and CNMRA7 attain global minimal solutions for $F8$ and $F10$. These three variants also perform the best for $F10$ compared to others. CNMRA2 exhibits slightly better Std values for $F11$, while CNMRA7 outperforms others in terms of Std values for $F12$. In summary, CNMRA1 is the top performer for dimension size 50, excelling in nine out of twelve functions, while CNMRA2 and CNMRA7 excel in four functions.

Inferences: The findings suggest that as dimension size increases, the computational complexity of all algorithms also grows. While the proposed variants perform significantly better for dimensions 10 and 30, their performance diminishes as the dimension size approaches 50. Several variants still perform well for more oversized dimensions, but dimension size 30 is identified as the optimal choice for future simulation results.

Table 3.4: Outcomes of chaotic variants for dimension size (10, 30, 50)

Function	Variant	D = 10		D = 30		D = 50		Function	Variant	D = 10		D = 30		D = 50	
		Mean	Std	Mean	Std	Mean	Std			Mean	Std	Mean	Std	Mean	Std
F1	CNMRA1	1.260E-285	0	6.630E-278	0	6.300E-284	0	F7	CNMRA1	2.400E-04	1.910E-04	3.500E-04	2.730E-04	2.630E-04	2.230E-04
	CNMRA2	5.050E-211	0	2.080E-207	0	5.990E-209	0		CNMRA2	3.870E-04	3.510E-04	4.380E-04	3.740E-04	3.892E-04	3.772E-04
	CNMRA3	5.502E-03	2.802E-03	5.772E-02	3.472E-02	1.512E-01	9.092E-02		CNMRA3	5.212E-01	1.671E-01	3.632E+00	7.581E-01	1.262E+00	1.262E+00
	CNMRA4	7.001E-03	1.681E-02	3.981E-02	1.541E-01	1.711E-02	3.851E-02		CNMRA4	1.951E-01	2.271E-01	2.642E+00	1.982E+00	3.72E+00	5.92E+00
	CNMRA5	1.401E-05	4.041E-04	6.961E-04	1.501E-03	1.801E-03	8.201E-03		CNMRA5	1.621E-02	1.850E-02	7.990E-01	1.602E+00	1.32E+00	3.62E+00
	CNMRA6	2.910E-02	4.720E-02	1.770E-01	3.910E-01	3.350E-01	6.821E-01		CNMRA6	2.561E-01	2.141E-01	3.052E+00	1.562E+00	9.12E+00	6.052E+00
	CNMRA7	5.231E-237	0	3.651E-245	0	1.971E-244	0		CNMRA7	3.321E-04	2.641E-04	3.171E-04	2.991E-04	3.611E-04	3.151E-04
	CNMRA8	1.112E-02	2.652E-02	5.022E-02	9.092E-02	1.252E-01	3.032E-01		CNMRA8	8.782E-02	8.562E-02	1.141E+00	1.611E+00	6.021E+00	6.521E+00
	CNMRA9	1.732E-01	2.732E-01	1.581E+00	2.161E+00	2.001E+00	3.721E+00		CNMRA9	4.792E-01	1.882E-01	3.591E+00	1.081E+00	1.260E+00	1.280E+00
	CNMRA10	4.561E-02	9.521E-02	1.951E-01	3.301E-01	7.031E-01	1.180E+00		CNMRA10	3.151E-01	8.151E-01	2.260E+00	1.770E+00	6.770E+00	6.430E+00
F2	CNMRA1	4.791E-1423	3.61E-1411	5.21E-1431	0.71E-1425	0.61E-1442	8.71E-143	F8	CNMRA1	0	0	0	0	0	0
	CNMRA2	1.781E-1041	2.01E-1034	2.01E-1061	1.901E-1053	9.01E-1032	5.71E-102		CNMRA2	0	0	0	0	0	0
	CNMRA3	1.300E+00	6.611E-01	8.530E+00	1.530E+00	1.680E+01	2.770E+00		CNMRA3	2.830E+016	8.60E+001	7.50E+021	4.50E+013	8.20E+022	2.93E+01
	CNMRA4	7.272E-02	1.472E-01	3.662E-01	5.692E-01	4.622E-01	8.012E-01		CNMRA4	4.263E+009	8.33E+001	1.23E+013	6.13E+017	0.51E+001	6.01E+01
	CNMRA5	9.102E-03	1.402E-02	3.132E-02	5.192E-02	7.342E-02	1.122E-01		CNMRA5	8.192E-02	2.722E-01	2.731E+001	7.11E+01	4.572E-01	1.141E+00
	CNMRA6	3.662E-01	2.532E-01	1.461E+00	1.081E+00	2.551E+00	2.391E+00		CNMRA6	7.241E+009	1.90E+002	1.60E+012	0.90E+013	2.30E+013	2.80E+01
	CNMRA7	1.212E-1198	3.62E-1192	3.82E-1151	6.82E-1142	2.71E-1181	6.01E-117		CNMRA7	0	0	0	0	0	0
	CNMRA8	3.361E-01	1.701E-01	1.030E+00	6.861E-01	2.450E+00	1.490E+00		CNMRA8	5.140E+005	0.00E+002	5.80E+011	6.52E+013	3.42E+012	3.82E+01
	CNMRA9	1.292E+00	7.91E-01	7.792E+00	2.922E+00	1.442E+015	0.012E+00		CNMRA9	2.372E+018	9.22E+001	1.12E+024	1.72E+011	9.72E+028	1.62E+01
	CNMRA10	5.621E-01	4.421E-01	1.661E+00	1.501E+00	3.701E+003	0.061E+00		CNMRA10	1.071E+011	0.091E+013	0.071E+013	0.721E+015	0.501E+015	2.51E+01
F3	CNMRA1	3.771E-280	0	6.791E-283	0	1.031E-280	0	F9	CNMRA1	8.881E-16	0	8.881E-16	0	8.881E-16	0
	CNMRA2	5.530E-208	0	1.540E-207	0	2.360E-205	0		CNMRA2	8.880E-16	0	8.880E-16	0	8.880E-16	0
	CNMRA3	1.590E-02	1.440E-02	3.191E-01	3.421E-01	1.621E+00	1.321E+00		CNMRA3	5.781E-01	2.191E-01	9.981E-01	2.561E-01	1.211E+00	2.591E-01
	CNMRA4	1.641E-02	8.021E-02	6.531E-02	1.481E-01	3.551E-01	8.201E-01		CNMRA4	1.061E-01	2.431E-01	7.842E-02	1.322E-01	1.252E-01	2.052E-01
	CNMRA5	2.012E-04	9.792E-04	3.402E-03	8.102E-03	2.212E-02	6.873E-02		CNMRA5	9.403E-03	2.133E-02	1.333E-02	2.573E-02	1.683E-02	3.333E-02
	CNMRA6	4.003E-02	6.663E-02	3.873E-01	5.713E-01	2.961E+00	4.731E+00		CNMRA6	3.503E-01	3.693E-01	4.733E-01	4.683E-01	4.883E-01	4.843E-01
	CNMRA7	4.603E-248	0	3.283E-233	0	2.573E-243	0		CNMRA7	8.883E-16	0	8.883E-16	0	8.883E-16	0
	CNMRA8	1.453E-02	4.093E-02	3.383E-01	5.291E-01	1.051E+00	2.701E+00		CNMRA8	2.121E-01	2.031E-01	2.241E-01	2.151E-01	4.421E-01	4.251E-01
	CNMRA9	3.561E-01	6.671E-01	6.291E+00	1.021E+01	1.461E+012	0.091E+01		CNMRA9	1.321E+00	9.841E-01	1.821E+001	0.071E+001	9.10E+001	1.40E+00
	CNMRA10	6.561E-02	9.011E-02	1.410E+00	4.480E+00	4.600E+00	7.780E+00		CNMRA10	5.391E-01	5.021E-01	6.601E-01	6.361E-01	6.871E-01	5.191E-01
F4	CNMRA1	5.282E-1412	7.62E-1402	6.12E-1391	5.62E-1381	9.92E-1261	0.32E-125	F10	CNMRA1	0	0	0	0	0	0
	CNMRA2	1.332E-1039	1.02E-1035	6.02E-1013	9.62E-1001	2.72E-1029	0.12E-102		CNMRA2	0	0	0	0	0	0
	CNMRA3	4.342E-02	1.682E-02	1.241E-01	4.751E-02	2.121E-01	1.031E-01		CNMRA3	2.551E-06	1.411E-06	2.911E-06	2.191E-06	2.511E-06	1.311E-06
	CNMRA4	3.561E-02	4.531E-02	4.751E-02	7.550E-02	4.220E-02	6.330E-02		CNMRA4	1.920E-06	5.000E-06	2.200E-06	3.540E-06	2.640E-06	4.460E-06
	CNMRA5	3.500E-03	6.000E-03	9.600E-03	1.650E-02	1.250E-02	2.120E-02		CNMRA5	1.280E-06	2.121E-06	5.121E-07	1.241E-06	8.881E-07	1.901E-06
	CNMRA6	6.041E-02	4.531E-02	1.361E-01	1.081E-01	1.721E-01	1.771E-01		CNMRA6	6.761E-06	2.011E-05	4.911E-06	8.951E-06	5.881E-06	9.981E-06
	CNMRA7	2.051E-1221	3.91E-1212	2.51E-1201	5.91E-1191	7.31E-1231	2.91E-122		CNMRA7	0	0	0	0	0	0
	CNMRA8	4.611E-02	4.521E-02	9.791E-02	9.581E-02	1.181E-01	1.071E-01		CNMRA8	8.791E-06	3.431E-05	4.591E-06	9.111E-06	1.141E-05	3.341E-05
	CNMRA9	2.032E-01	1.952E-01	3.662E-01	3.102E-01	5.522E-01	4.812E-01		CNMRA9	5.792E-06	1.062E-05	5.912E-06	1.042E-05	3.472E-05	1.922E-04
	CNMRA10	1.282E-01	1.220E-01	2.360E-01	8.100E-01	3.270E-01	3.460E-01		CNMRA10	7.100E-06	2.190E-05	1.340E-05	4.460E-05	4.440E-06	7.370E-06
F5	CNMRA1	8.980E+00	1.280E-02	2.890E+01	1.861E-02	4.890E+01	2.081E-02	F11	CNMRA1	1.230E+00	5.951E-01	1.250E+00	2.521E-01	1.240E+00	1.391E-01
	CNMRA2	8.920E+00	1.601E-02	2.890E+01	1.631E-02	4.890E+01	2.321E-02		CNMRA2	8.322E-01	4.722E-01	1.090E+00	2.492E-01	1.190E+00	1.532E-01
	CNMRA3	1.540E+00	1.5170E+00	1.010E+026	3.30E+013	4.50E+024	2.60E+02		CNMRA3	1.970E+00	5.502E-01	1.490E+00	1.192E-01	1.350E+00	9.772E-02
	CNMRA4	9.350E+00	1.170E+00	3.040E+015	1.50E+005	2.00E+014	5.51E+00		CNMRA4	2.542E-01	2.270E-01	8.790E-01	3.410E-01	1.121E+00	1.860E-01
	CNMRA5	8.971E+00	5.450E-02	2.901E+01	1.540E-01	4.911E+01	6.500E-01		CNMRA5	3.620E-01	2.970E-01	8.500E-01	2.600E-01	9.780E-01	2.770E-01
	CNMRA6	1.041E+01	2.641E+00	5.001E+01	3.891E+01	9.841E+01	9.441E+01		CNMRA6	2.410E-01	2.420E-01	7.130E-01	3.660E-01	1.041E+00	3.041E-01
	CNMRA7	8.971E+00	1.851E-02	2.891E+01	1.881E-02	4.891E+01	1.571E-02		CNMRA7	1.091E+00	6.201E-01	1.161E+00	2.661E-01	1.231E+00	1.501E-01
	CNMRA8	1.061E+01	1.671E+00	3.891E+01	1.352E+01	7.002E+01	2.962E+01		CNMRA8	2.951E-01	2.751E-01	7.971E-01	2.961E-01	1.072E+00	2.601E-01
	CNMRA9	5.482E+01	5.412E+01	4.002E+029	4.02E+021	2.82E+031	5.32E+03		CNMRA9	6.901E-01	6.581E-01	1.232E+00	3.971E-01	1.332E+00	1.532E-01
	CNMRA10	1.452E+01	1.002E+01	7.872E+01	8.552E+01	2.472E+024	1.62E+02		CNMRA10	2.302E-01	2.142E-01	8.842E-01	3.572E-01	1.152E-01	2.522E-01
F6	CNMRA1	1.522E+00	7.632E-01	7.051E+00	3.872E-01	1.201E+01	4.042E-01	F12	CNMRA1	9.242E-01	1.532E-01	2.991E+00	9.702E-03	4.991E+00	2.302E-03
	CNMRA2	1.521E+00	6.410E-01	6.861E+00	5.440E-01	1.191E+01	4.760E-01		CNMRA2	9.460E-01	9.690E-02	2.991E+00	1.960E-02	4.991E+00	2.200E-03
	CNMRA3	1.611E+00	8.610E-01	7.131E+00	2.923E-01	1.211E+01	3.283E-01		CNMRA3	1.091E+00	7.273E-02	3.561E+00	1.623E-01	6.071E+00	2.133E-01
	CNMRA4	1.111E+00	6.513E-01	6.591E+00	7.413E-01	1.201E+01	5.273E-01		CNMRA4	6.493E-01	3.683E-01	3.061E+00	2.083E-01	5.101E+00	2.463E-01
	CNMRA5	1.001E+00	6.973E-01	6.151E+00	7.853E-01	1.141E+01	4.483E-01		CNMRA5	6.683E-01	3.453E-01	2.961E+00	1.871E-01	5.000E+00	2.891E-02
	CNMRA6	1.040E+00	6.311E-01	6.940E+00	5.591E-01	1.230E+01	1.040E+00		CNMRA6	5.711E-01	4.031E-01	3.260E+00	2.361E-01	5.500E+00	4.801E-01
	CNMRA7	1.460E+00	7.201E-01	6.980E+00	4.671E-01	1.190E+01	4.161E-01		CNMRA7	9.341E-01	1.501E-01	2.990E+00	4.041E-02	4.990E+00	1.802E-03
	CNMRA8	9.732E-01	7.902E-01	6.960E+00	4.332E-01	1.210E+01	5.592E-01		CNMRA8	7.662E-01	3.472E-01	3.120E+00	1.552E-01	3.540E+00	2.692E-01
	CNMRA9	1.770E+00	4.882E-01	8.560E+00	2.970E+00	1.580E+015	0.630E+00		CNMRA9	9.972E-01	2.842E-01	3.810E+00	2.822E-01	7.190E+001	4.50E+00
	CNMRA10	1.130E+00	5.992E-01	7.360E+00	8.842E-01	1.260E+01	1.000E+00		CNMRA10	6.542E-01	3.432E-01	3.340E+00	8.222E-01	5.620E+00	6.542E-01

3.4.6 Comparison of best proposed NMRA variant concerning other algorithms

From the above discussion, the proposed variant NMRA with Chebyshev map (CNMRA) is best among all the other proposed variants red because mating factor parameter (λ) efficiently mapped with this chaotic map and compared to upgraded versions of other well-known algorithms for linearly population size reduction and dimension size (D) = 30 as given in Table 3.5.

- **Experimental testing**

Table 3.5 presents the simulated outcomes for all the algorithms employed in this comparative study. It is crucial to highlight that all the results are reported in terms of the mean and *std* of fitness values. Looking at Table 3.5, it becomes evident that CNMRA stands out as the algorithm capable of achieving nearly global optimal solutions for $F1$, $F2$, $F3$, and $F4$, while other algorithms often get trapped in local optima. Regarding $F5$, CNMRA boasts the best performance, especially for *std* values. CMA-ES performs well on $F6$, but other methods produce significantly divergent outcomes. For $F7$, CNMRA demonstrates superior performance. GWO-E, FA-FPO, EO, GSSA, NMRA, and CNMRA successfully reach global minima for $F8$ and $F10$. In the case of $F9$, SCCSA, FA-FPO, GSSA, NMRA, and CNMRA all deliver excellent and similar results, making it challenging to determine the best approach. JADE emerges as the top performer for $F11$ and $F12$. In summary, CNMRA excels in eight functions, GWO-E in two, JADE in three, and NMRA in three. After a comprehensive comparison, CNMRA emerges as the overall best-performing algorithm.

- **Statistical testing**

To assess the statistical significance of the proposed CNMRA's performance on the CEC 2005 benchmark problems, we employed two distinct statistical tests: the Wilcoxon rank-sum test [231] and f-rank test [220]. These tests were applied to the data presented in Table 3.5.

Wilcoxon rank-sum test: To assess if the proposed variant is statistically significant, rank-sum test [231] is used at a 5% significance level. As a result, the suggested algorithm CNMRA is compared to others such as CNMRA/CMA-ES, CNMRA/JADE, CNMRA/SaDE, CNMRA/SHADE, CNMRA/OEWOA, CNMRA/GWO-E, CNMRA/LSHADE-SPACMA, CNMRA/SCCSA, CNMRA/FA-FPO, CNMRA/EO, CNMRA/GSSA and CNMRA/ GSSA. Because the CNMRA cannot be compared to itself, NA is substituted. To begin, the symbol '+' is used when CNMRA's performance is better than that of other algorithms in comparison, and this is referred to as a *win(w)* circumstance. Second, a *tie(t)* condition occurs when CNMRA's performance is comparable to other competitor algorithms, as the = sign indicates. Finally,

Table 3.5: Simulated outcomes of CNMRA as compared to competitive algorithms for CEC 2005 test suite

Function		SaDE [69]	OWOA [207]	SCCSA [211]	GWO-E [210]	FA-FPO [212]	CMA-ES [213]	SHADE [213]	LSHADE-SPACMA [213]	EO [213]	GSSA [214]	JADE [69]	NMRA	CNMRA
F_1	mean	4.501E-20	7.751E-176	9.220E-69	3.920E-67	1.510E-184	1.420E-18	1.420E-09	2.230E-01	3.320E-40	3.252E-109	1.802E-60	1.122E-86	5.962E-280
	std	6.902E-20	0	3.812E-68	1.112E-66	0	3.131E-18	3.091E-09	1.481E-01	6.781E-40	1.201E-108	8.401E-60	7.881E-86	0
	f-rank	10	3	6	7	2	11	12	13	9	4	8	5	1
F_2	mean	1.901E-14	1.863E-115	8.253E-41	4.313E-36	5.043E-93	2.983E-07	8.703E-03	2.110E+01	7.123E-23	3.213E-54	1.803E-25	3.463E-45	1.273E-143
	std	1.053E-14	1.323E-114	4.193E-40	6.573E-36	3.473E-93	1.781E+00	2.131E-02	9.571E+00	6.361E-23	2.051E-53	8.801E-25	1.521E-44	9.081E-143
	f-rank	10	2	6	7	3	11	12	13	9	4	8	5	1
F_3	mean	9.000E-37	2.870E+04	4.310E-13	3.750E-37	1.230E-183	1.590E-05	1.540E+01	8.871E+01	8.060E-09	3.880E-106	5.700E-61	2.540E-85	6.040E-277
	std	5.430E-36	1.021E+04	2.830E-30	1.360E-36	0	2.210E-05	9.941E+00	4.721E+01	1.600E-08	2.110E-105	2.700E-60	1.322E-84	0
	f-rank	7	13	8	6	2	10	11	12	9	3	5	4	1
F_4	mean	7.402E-11	1.061E+01	2.152E-17	2.392E-25	9.972E-93	2.012E-06	9.792E-01	2.112E+00	5.391E-10	7.761E-56	8.201E-24	3.501E-45	1.901E-140
	std	1.821E-10	2.222E+01	1.061E-16	6.801E-25	7.311E-93	1.251E-06	7.992E-01	4.922E-01	1.382E-09	2.462E-55	4.002E-23	1.522E-44	7.332E-139
	f-rank	9	13	7	5	2	10	11	12	8	3	6	4	1
F_5	mean	2.102E+01	2.852E+01	5.902E+00	2.651E+01	2.891E+01	3.671E+01	2.441E+01	2.881E+01	2.531E+01	2.901E+01	8.002E-02	2.891E+01	2.891E+01
	std	7.800E+00	2.222E-01	9.132E-01	5.192E-01	1.722E-02	3.340E+01	1.120E+01	8.242E-01	1.692E-01	2.282E-02	5.602E-01	2.542E-02	1.831E-02
	f-rank	11	8	10	7	2	13	12	9	6	5	1	4	3
F_6	mean	9.300E+02	1.620E+00	4.141E-08	2.650E+01	5.880E+00	6.831E-19	5.311E-10	2.481E-01	8.291E-06	9.661E-01	2.900E+00	8.560E+00	6.891E+00
	std	1.801E+02	6.931E-01	5.221E-08	5.191E-01	5.861E-01	6.711E-19	6.351E-10	1.131E-01	5.021E-06	3.771E-01	1.201E+00	5.901E-01	5.761E-01
	f-rank	13	10	3	12	9	1	2	5	4	6	11	8	7
F_7	mean	4.801E-03	1.371E-03	1.331E-03	9.901E-04	1.131E-04	2.751E-02	2.351E-02	4.701E-03	1.171E-03	4.161E-04	6.402E-04	6.942E-04	3.142E-04
	std	1.202E-03	2.852E-03	1.722E-03	8.372E-04	8.942E-04	7.902E-03	8.802E-03	1.902E-03	6.541E-04	4.321E-04	2.501E-04	6.101E-04	3.261E-04
	f-rank	11	9	8	6	2	13	12	10	7	3	4	5	1
F_8	mean	1.201E-03	0	5.461E+00	0	0	2.531E+01	8.531E+00	6.751E+01	0	0	1.001E-04	0	0
	std	6.501E-04	0	5.621E+00	0	0	8.551E+00	2.191E+00	1.001E+01	0	0	6.001E-05	0	0
	f-rank	9	1	10	1	1	12	11	13	1	1	8	1	1
F_9	mean	2.701E-03	3.021E-15	8.881E-16	5.581E-15	8.881E-16	1.551E+01	3.951E-01	3.931E-02	8.341E-14	8.881E-16	8.201E-10	8.881E-16	8.881E-16
	std	5.100E-04	2.270E-15	9.360E-32	1.670E-15	0	7.921E+00	5.860E-01	1.510E-02	2.530E-14	0	6.900E-10	0	0
	f-rank	10	6	5	7	1	13	12	11	8	1	9	1	1
F_{10}	mean	7.800E-04	1.420E-02	3.330E-02	0	0	5.760E-15	4.800E-03	8.940E-01	0	0	9.900E-08	0	0
	std	1.200E-03	1.000E-01	4.560E-02	0	0	6.180E-15	7.700E-03	1.072E-01	0	0	6.002E-07	0	0
	f-rank	9	12	11	1	1	7	10	13	1	1	8	1	1
F_{11}	mean	1.902E-05	1.062E-01	1.342E-02	1.982E-02	8.322E-01	2.872E-16	3.462E-02	8.182E-04	7.972E-07	2.342E-02	4.602E-17	1.092E+00	1.082E+00
	std	9.202E-06	4.972E-02	1.602E-02	1.012E-02	1.782E-01	5.641E-16	8.751E-02	1.001E-03	7.691E-07	1.561E-02	1.901E-16	2.561E-01	2.351E-01
	f-rank	4	10	6	8	11	2	9	5	3	7	1	12	12
F_{12}	mean	6.101E-05	1.032E+00	2.011E-02	2.501E-01	2.942E+00	3.661E-04	7.321E-04	1.021E-02	2.921E-02	2.432E+00	2.001E-16	2.972E+00	2.992E+00
	std	2.000E-05	3.610E-01	7.230E-02	1.630E-01	1.590E-01	2.000E-03	2.800E-03	1.030E-02	3.520E-02	1.162E+00	6.500E-16	1.380E-01	2.200E-03
	f-rank	2	9	6	8	11	3	4	5	7	13	1	12	10
w/l/t		2/10/0	2/9/1	3/9/0	2/8/2	2/7/3	3/9/0	3/9/0	3/9/0	3/7/2	2/7/3	3/9/0	0/9/3	NA
Average f-rank		8.75	8	7.16	6.25	3.92	8.83	9.83	10.08	6	4.25	5.83	5.25	3.33
Overall f-rank		10	9	8	7	2	11	12	13	6	3	5	4	1

a $loss(l)$ situation happens when CNMRA performs worse than the compared algorithm and the $-$ sign is utilized instead. According to the rank-sum test findings in Table 3.5, CNMRA performs significantly better in more than 75% of the cases.

Friedman rank test: The statistical significance of differences between the algorithms was evaluated using a non-parametric f-rank test. This test assigns ranks to the algorithms based on their performance, as illustrated in Table 3.5. The table displays each algorithm's f-rank, and the total f-rank is computed in the final row. Remarkably, the proposed CNMRA consistently achieved a rank of 1st place across the entire test suite, indicating its superior performance compared to all other algorithms included in the comparison.

3.4.7 Convergence profiles

In this subsection, we conducted a convergence analysis of the proposed CNMRA and the basic NMRA on the CEC 2005 benchmark suite. Figure 3.1 displays the convergence curves for NMRA and CNMRA for their respective functions. These convergence plots depict how the best solution

converges over iterations. Examining the convergence curves, it becomes evident that CNMRA quickly identifies the optimal solution in the early iterations for both uni-modal and multi-modal functions. Thus, from convergence analysis, CNMRA is found to be best.

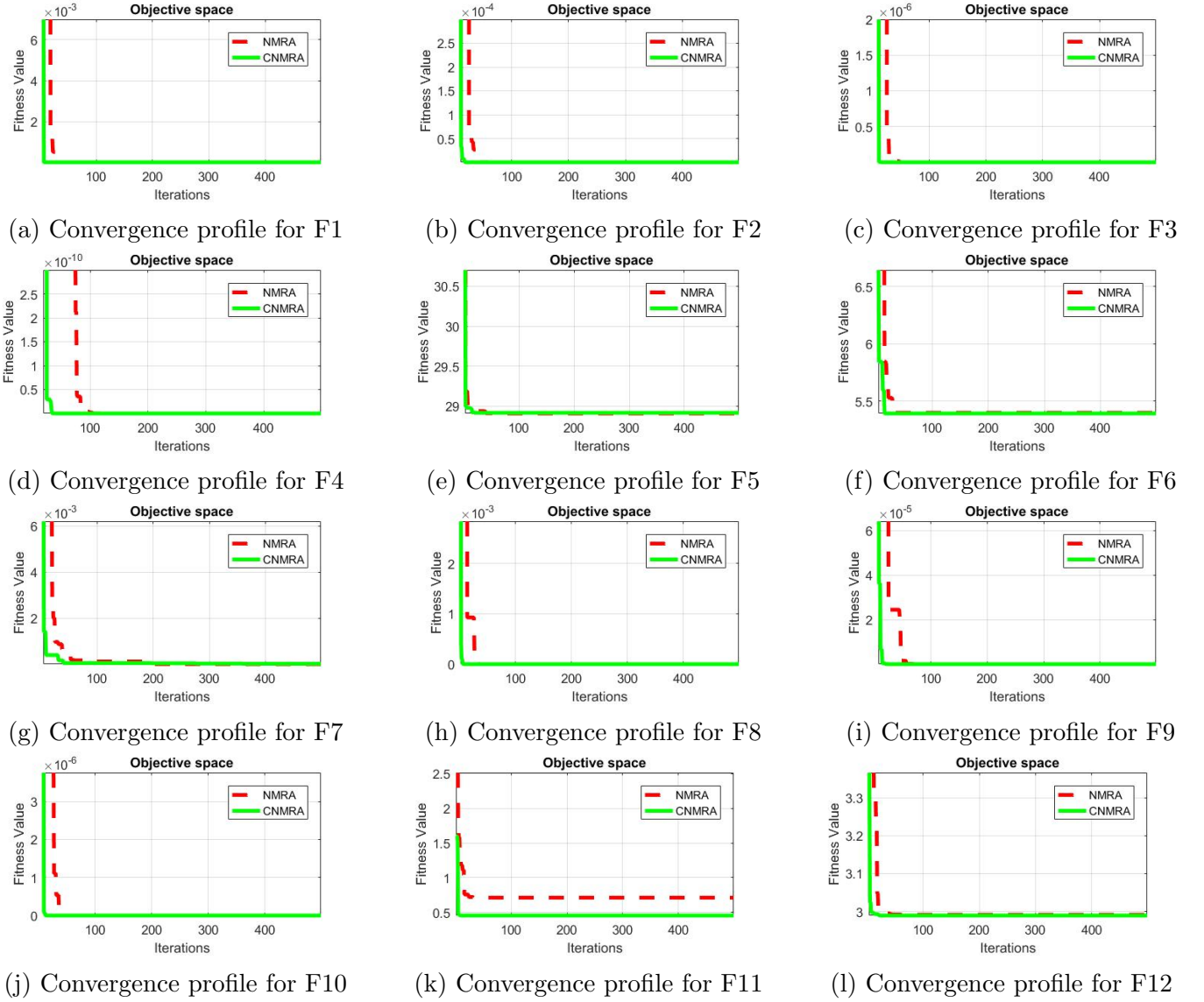


Figure 3.1: Convergence profiles for unimodal (F1 to F7) and multimodal (F8 to F12) benchmark functions of CEC 2005 test suite

3.4.8 Comparison on CEC2014 test suite

In this section, we evaluated the effectiveness of the proposed CNMRA algorithm by testing it on the CEC 2014 test suite and comparing it to other established algorithms. This test suite comprises thirty functions, which are categorized as follows: three uni-modal, thirteen multi-modal, six hybrid, and eight composition functions. Table A.3 provides a detailed overview of these functions. The performance of CNMRA was assessed alongside other well-known algorithms such as LX-BBO, RW-GWO, ISOS, B-BBO, VNBA, IMEHO, PBIL, CCS, BDE, MGSCA, and conventional NMRA.

The simulation results for all algorithms, measured in terms of the mean and *std* errors of fitness values for a dimension size (D) of 30, are presented in Table 3.6. According to the results, CNMRA outperforms other algorithms for F_1 , F_4 , F_{18} , F_{19} , F_{21} , F_{29} , and F_{30} . ISOS delivers the best results in terms of mean and *std* values for F_2 . IMEHO performs exceptionally well for functions F_3 , F_9 , and F_{20} , although CNMRA’s performance is highly competitive. For F_5 , F_{12} , and F_{27} , both LX-BBO and B-BBO produce highly competitive results, with LX-BBO excelling in the mean value. RW-GWO marginally outperforms other algorithms in terms of mean values for F_6 . CNMRA achieves the best performance in terms of mean values for F_7 , F_{10} , F_{16} , F_{23} , and F_{28} , while B-BBO manages to attain a near-optimal solution for F_8 , with CNMRA closely behind. CNMRA demonstrates favorable results compared to other methods for functions F_{11} , F_{22} , and F_{26} in terms of *std* values alone. IMEHO achieves exactly zero mean error values for functions F_{13} , F_{14} , and F_{15} . BDE delivers the best results in terms of both mean and standard deviation values for F_{17} . For functions F_{24} and F_{25} , the performance of NMRA and CNMRA is identical but superior to other algorithms. Overall, CNMRA outperforms all other algorithms for 17 functions, while B-BBO, RW-GWO, BDE, and ISOS excel for 1 function each, LX-BBO for 3 functions, IMEHO for 6 functions, and NMRA for 2 functions. To statistically validate the efficiency of CNMRA, a f-rank test was conducted for all benchmark functions. The last two rows of Table 3.6 summarize the ranks for all benchmark function categories, revealing that CNMRA holds the highest ranking across the entire benchmark suite and within each category of benchmark functions. Consequently, CNMRA’s performance is deemed the best among all tested algorithms.

3.4.9 Comparison on 100-digit challenge benchmark functions

CNMRA’s performance has been evaluated on the 10 CEC 2019 benchmark problems [9]. These functions are characterized by their dimensions, optimum values, search ranges, and descriptions, all provided in Table A.2. CNMRA has been compared to a set of state-of-the-art algorithms, including ASSA [228], CS [82], GWO, DA, SSA, NBA, FPA, and the basic NMRA. The parameter settings for all these methods are outlined in Table 3.1. To ensure robustness, all simulation results are presented in terms of best, worst, mean, median, and *std* values across 50 runs for each method, each run consisting of 500 iterations.

Table 3.7 represents the simulation results for these test functions. CNMRA exhibits superior performance for CEC-01, outperforming other competitive algorithms in terms of best, worst, mean, and median values. For CEC-02, most algorithms produce nearly identical results, with CS being the best according to *std* values. When considering CEC-03 and CEC-05, where direct comparisons are limited, CNMRA demonstrates the lowest *std* values among all compared methods. CNMRA

Table 3.6: Simulated outcomes of CNMRA as compared to other algorithms for CEC 2014 test suite

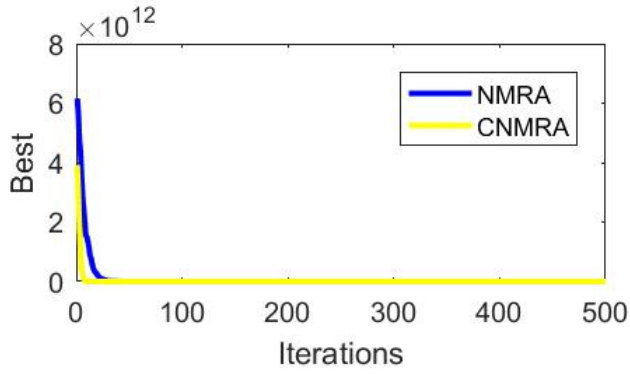
Functions		LX-BBO [219]	B-BBO [219]	RW-GWO [217]	ISOS [220]	PBIL [224]	VNBA [224]	IMEHO [224]	CCS [224]	BDE [223]	MGSCA [223]	NMRA	CNMRA
F_1	mean	1.012E+07	6.502E+06	8.022E+06	9.822E+05	3.422E+08	2.432E+08	2.372E+06	1.462E+08	1.252E+07	2.922E+07	4.512E+06	8.992E+05
	std	1.012E+07	1.302E+06	3.312E+06	7.051E+05	1.091E+08	5.931E+07	4.321E+06	3.270E+07	7.470E+03	2.070E+07	1.240E+06	4.470E+05
	f-rank	7	5	6	2	12	11	3	10	8	9	4	1
F_2	mean	5.340E+04	2.351E+04	2.231E+05	5.271E+00	4.081E+10	1.921E+10	5.491E+03	2.601E+09	9.251E+03	2.262E+09	1.712E+04	2.302E+02
	std	2.142E+04	9.992E+03	5.512E+05	1.722E+01	3.392E+09	4.232E+09	4.872E+03	5.222E+08	6.192E+04	1.692E+09	8.172E+03	1.312E+02
	f-rank	7	6	8	1	12	11	3	10	4	9	5	2
F_3	mean	1.631E+04	6.031E+03	3.161E+02	4.791E+02	9.191E+04	2.931E+04	1.411E+02	2.701E+03	5.271E+02	1.770E+04	2.640E+04	1.440E+03
	std	1.700E+04	3.150E+03	4.340E+02	6.240E+02	1.750E+04	1.390E+04	1.580E+02	7.741E+04	1.111E+03	6.631E+03	4.831E+03	5.801E+02
	f-rank	8	7	2	3	12	11	1	6	4	9	10	5
F_4	mean	9.991E+01	1.021E+02	3.411E+01	5.981E+01	3.431E+03	1.601E+03	1.241E+02	3.221E+02	5.761E+02	2.761E+02	2.091E+01	1.492E+01
	std	2.842E+01	3.132E+01	1.802E+01	3.572E+01	7.562E+02	3.632E+02	4.772E+01	4.092E+01	4.292E+01	6.552E+01	5.942E+00	6.492E+00
	f-rank	5	6	3	4	12	11	7	9	10	8	2	1
F_5	mean	3.062E+00	3.741E+00	2.051E+01	2.031E+01	2.101E+01	2.101E+01	2.101E+01	2.101E+01	2.021E+01	2.041E+01	2.101E+01	2.081E+01
	std	7.861E-01	4.911E-01	7.461E-02	6.671E-02	5.561E-02	5.431E-02	5.991E-02	8.811E-02	1.331E-01	1.440E-01	6.260E-02	7.040E-02
	f-rank	1	2	6	4	9	8	10	11	3	5	12	7
F_6	mean	1.700E+01	1.990E+01	9.840E+00	1.050E+01	3.800E+01	3.300E+01	1.200E+01	2.500E+01	2.380E+01	1.940E+01	2.730E+01	2.030E+01
	std	3.120E+00	2.700E+00	3.491E+00	2.391E+00	1.161E+00	2.581E+00	2.721E+00	2.001E+00	4.391E+00	2.891E+00	1.771E+00	2.251E+00
	f-rank	4	6	1	2	12	11	3	9	8	5	10	7
F_7	mean	1.750E-01	7.810E-02	2.530E-01	1.560E-02	3.401E+02	1.111E+02	0	2.301E+01	1.501E+00	1.991E+01	1.700E-03	0
	std	8.561E-02	4.441E-02	1.431E-01	1.831E-02	2.741E+01	1.811E+01	1.191E-01	3.521E+00	6.081E+00	1.182E+01	5.372E+04	7.971E-06
	f-rank	6	5	7	4	12	11	2	10	8	9	3	1
F_8	mean	5.532E+01	4.711E-01	4.382E+01	1.472E+01	3.002E+02	1.742E+02	3.302E+01	2.902E+02	6.122E+01	1.072E+02	1.082E+02	3.742E+01
	std	3.782E+02	6.791E-01	4.882E+00	3.342E+00	1.032E+01	1.612E+01	9.192E+00	2.232E+01	2.602E+02	1.142E+01	1.282E+01	5.471E+00
	f-rank	6	1	5	2	12	10	3	11	7	8	9	4
F_9	mean	7.661E+01	9.111E+01	6.331E+01	2.561E+02	3.701E+02	2.501E+02	3.201E+01	2.901E+02	1.101E+02	1.391E+02	2.801E+02	2.461E+02
	std	1.611E+01	1.541E+01	1.301E+01	1.341E+01	1.692E+01	2.032E+01	1.152E+01	2.382E+01	4.692E+01	2.602E+01	4.272E+01	3.212E+01
	f-rank	3	4	2	9	12	8	1	11	5	6	10	7
F_{10}	mean	1.252E+04	6.682E+03	9.612E+02	1.782E+03	6.262E+03	3.502E+03	2.262E+03	8.552E+03	1.752E+03	2.822E+03	1.482E+03	6.962E+02
	std	1.162E+02	4.582E+02	2.722E+02	4.092E+01	3.052E+02	3.472E+02	5.722E+02	4.912E+02	7.281E+02	6.831E+02	2.091E+02	1.461E+02
	f-rank	12	10	2	5	9	8	6	11	4	7	3	1
F_{11}	mean	1.231E+04	6.711E+03	2.681E+03	1.481E+03	7.101E+03	6.801E+03	2.861E+03	8.831E+03	3.901E+03	3.301E+03	2.251E+03	3.011E+03
	std	3.412E+02	5.172E+02	3.682E+02	4.542E+02	2.972E+02	3.792E+02	5.380E+02	5.500E+02	7.380E+02	2.260E+02	2.860E+02	2.720E+02
	f-rank	4	8	5	7	3	6	9	10	12	11	2	1
F_{12}	mean	1.112E-02	1.112E-02	5.452E-01	3.552E-01	1.000E+01	1.000E+01	1.000E+01	1.000E+01	3.772E-01	6.332E-01	8.002E-01	6.002E-01
	std	1.752E-18	1.752E-18	1.662E-01	5.732E-02	3.382E-01	3.512E-01	5.262E-01	1.091E+00	1.712E-01	3.362E-01	1.392E-01	1.192E-01
	f-rank	1	1	5	3	9	10	11	12	4	7	8	6
F_{13}	mean	6.552E-01	6.782E-01	2.802E-01	3.772E-01	0	0	0	0	5.132E-01	5.511E-01	3.001E-01	3.001E-01
	std	1.561E-01	7.981E-02	6.301E-02	7.101E-02	2.561E-01	3.641E-01	6.251E-02	1.761E-01	1.411E-01	8.941E-02	3.581E-02	3.121E-02
	f-rank	11	12	5	8	3	4	1	2	9	10	7	6
F_{14}	mean	6.201E-01	3.930E-01	4.230E-01	2.710E-01	1.001E+02	6.001E+01	0	1.001E+01	4.410E-01	2.341E+00	2.000E-01	2.000E-01
	std	2.960E-01	1.550E-01	2.150E-01	5.120E-02	1.161E+01	1.221E+01	9.850E-02	1.881E+00	2.460E-01	3.311E+00	2.640E-02	2.590E-02
	f-rank	8	5	6	4	12	11	1	10	7	9	3	2
F_{15}	mean	1.551E+01	1.882E+01	8.812E+00	1.062E+01	6.842E+05	2.392E+03	0	8.002E+01	3.782E+01	8.722E+01	2.812E+01	2.471E+01
	std	5.491E+00	5.641E+00	1.511E+00	3.711E+00	2.851E+05	1.221E+03	1.351E+00	3.031E+01	9.261E+01	1.011E+02	4.431E+00	3.131E+00
	f-rank	4	5	2	3	12	11	1	9	8	10	7	6

also excels in producing the best, worst, and mean values for CEC-04. SSA performs the best for CEC-06. CNMRA, when compared to other algorithms, approaches the optimum solution closely for CEC-07, CEC-08, CEC-09, and CEC-10. In summary, CNMRA is the best performer for 8 out of the 10 test functions, with CS and SSA being the top choices for the 1 function each.

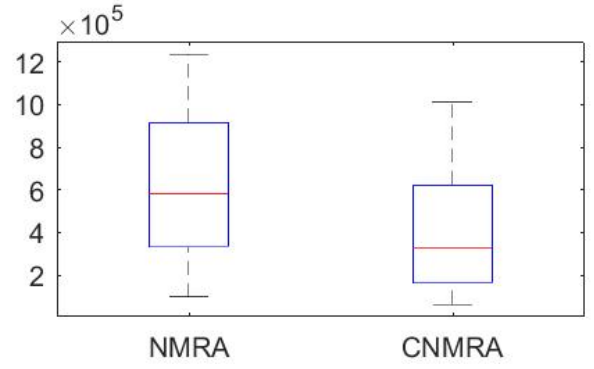
Furthermore, convergence profiles and box plots for NMRA and CNMRA have been constructed for these test functions, as shown in Figure 3.2. The convergence profiles reveal that CNMRA converges to the optimal solution faster than NMRA in most cases, underscoring its superior performance. Box plots depicting the fitness values affirm CNMRA's effectiveness due to its lower median fitness value, consistently delivering strong results in solving these benchmark functions, as depicted in Figure 3.2.

3.5 Real-world engineering design optimization problems

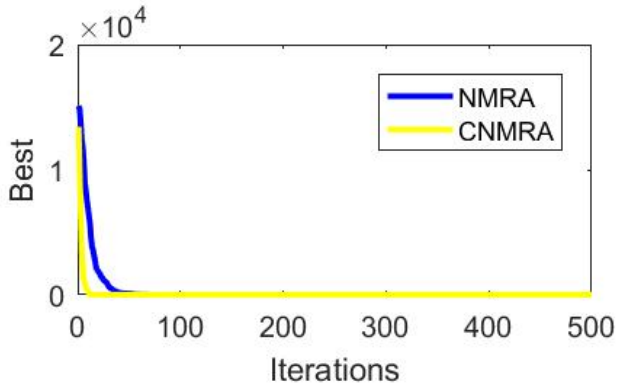
To assess the performance of the proposed CNMRA algorithm, three engineering design-constrained optimization challenges were employed. These challenges involve the design of a pressure vessel, a welded beam, and a tension-compression spring. CNMRA's performance is evaluated compared to other highly competitive algorithms, including DA, NBA, TLBO, and LebTLBO, as presented in



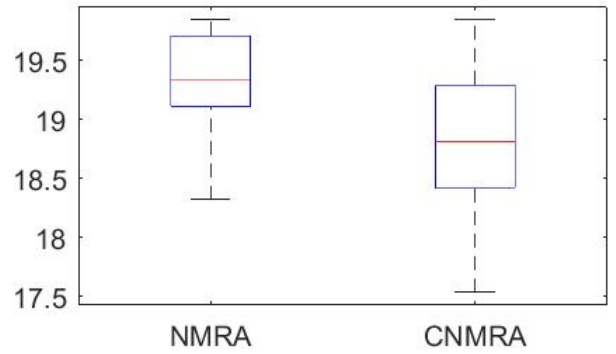
(a) Convergence profile for CEC-01



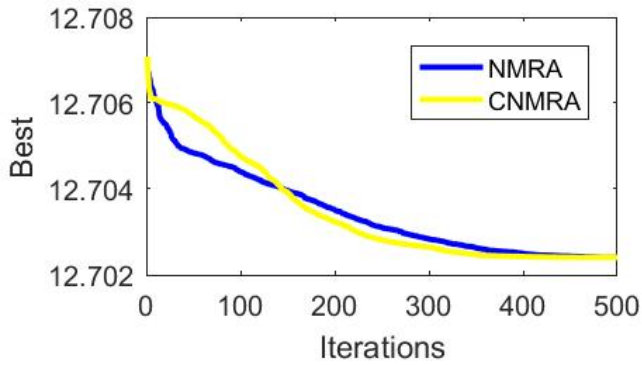
(b) Box plot for CEC-01



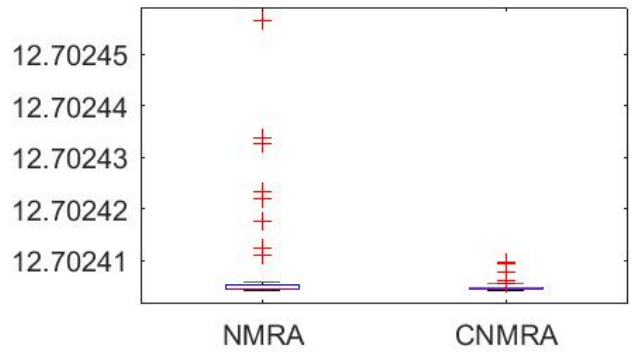
(c) Convergence profile for CEC-02



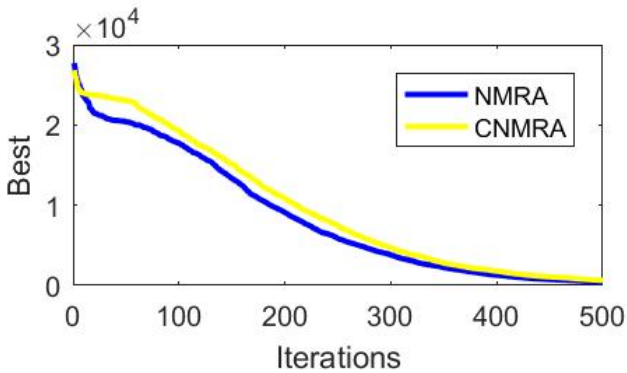
(d) Box plot for CEC-02



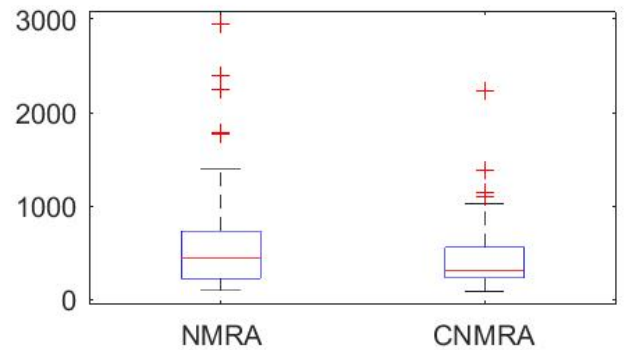
(e) Convergence profile for CEC-03



(f) Box plot for CEC-03

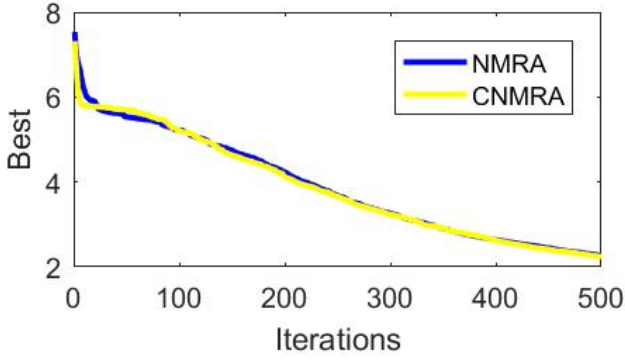


(g) Convergence profile for CEC-04

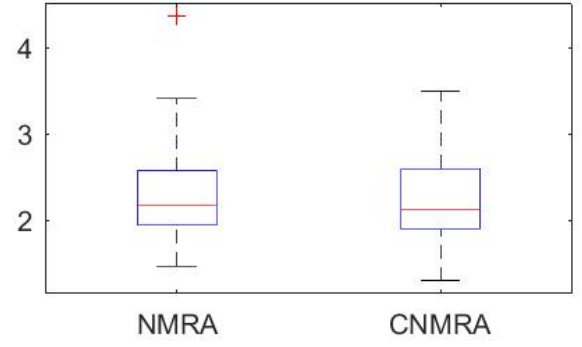


(h) Box plot for CEC-04

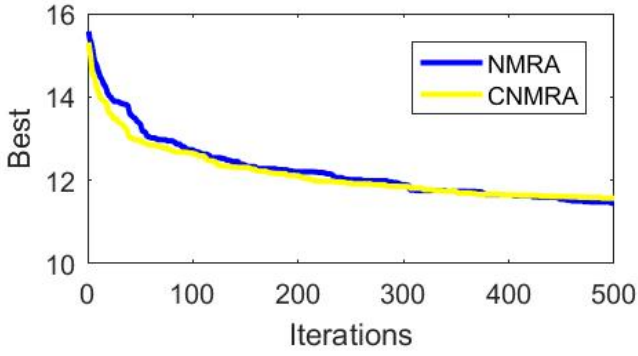
Figure 3.2: Convergence profiles and box plots of NMRA and CNMRA for 100-digit benchmark functions (CEC 2019)



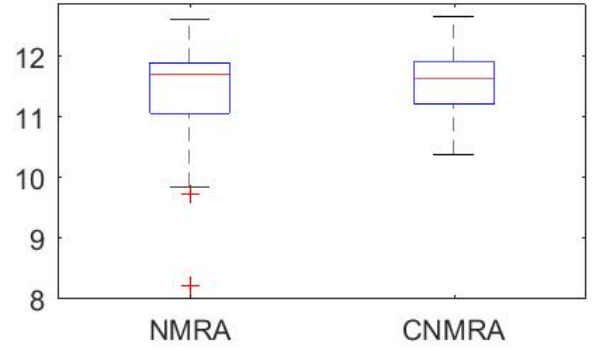
(i) Convergence profile for CEC-05



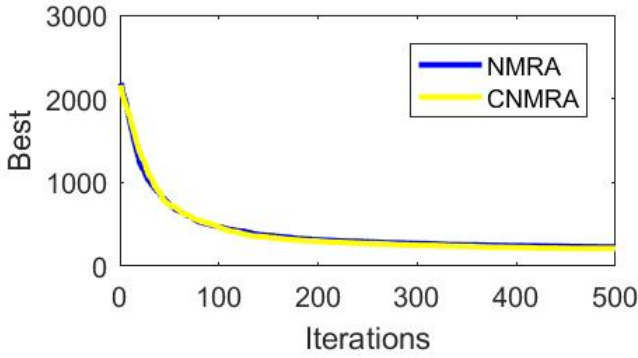
(j) Box plot for CEC-05



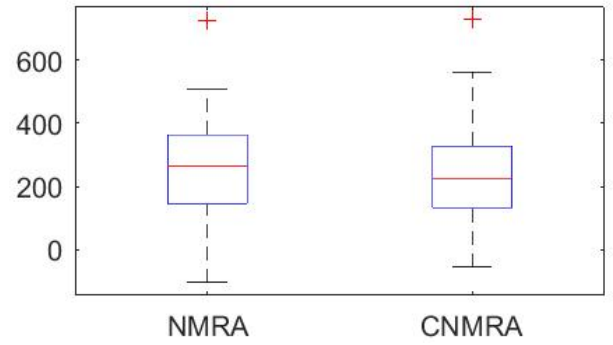
(k) Convergence profile for CEC-06



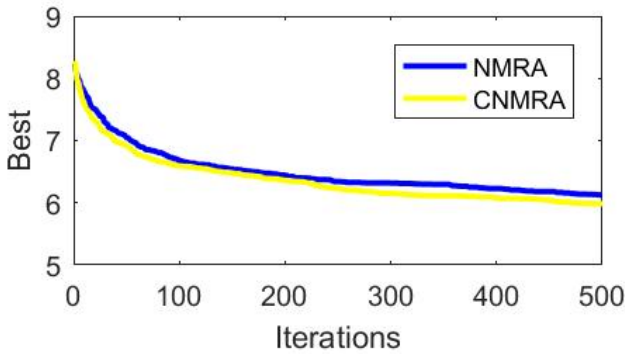
(l) Box plot for CEC-06



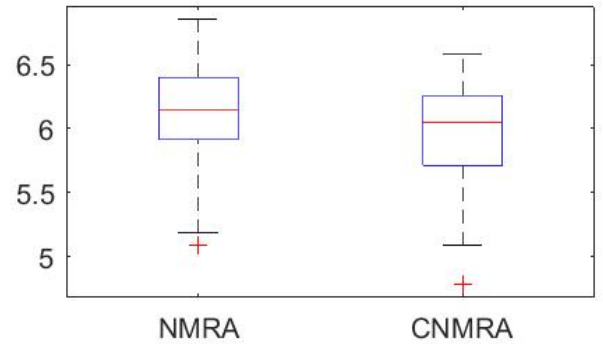
(m) Convergence profile for CEC-07



(n) Box plot for CEC-07



(o) Convergence profile for CEC-08



(p) Box plot for CEC-08

Figure 3.2: Convergence profiles and box plots of NMRA and CNMRA for 100-digit benchmark functions (CEC 2019) (continued)

Table 3.6: Simulated outcomes of CNMRA as compared to other algorithms for CEC 2014 test suite (Continued)

Functions		LX-BBO [219]	B-BBO [219]	RW-GWO [217]	ISOS [220]	PBIL [224]	VNBA [224]	IMEHO [224]	CCS [224]	BDE [223]	MGSCA [223]	NMRA	CNMRA
F_{16}	mean	1.081E+01	1.061E+01	1.081E+01	9.211E+01	2.001E+01	2.001E+01	2.001E+01	2.001E+01	1.261E+01	1.161E+01	1.281E+01	1.041E+01
	std	5.840E-01	6.250E-01	6.110E-01	7.310E-01	2.120E-01	3.660E-01	7.640E-01	1.750E-01	5.921E-01	6.911E-01	3.351E-01	2.571E-01
	f-rank	3	2	4	12	9	10	11	8	6	5	7	1
F_{17}	mean	1.491E+06	1.272E+06	5.712E+05	1.752E+05	9.742E+06	2.532E+06	7.692E+04	1.152E+07	2.802E+04	9.562E+05	1.902E+05	8.872E+04
	std	9.342E+05	5.462E+05	4.102E+05	1.642E+05	2.792E+06	3.342E+06	8.380E+04	4.590E+06	2.240E+04	7.620E+05	7.540E+04	2.420E+04
	f-rank	9	8	6	4	11	10	2	12	1	7	5	3
F_{18}	mean	2.890E+03	8.220E+02	6.520E+03	3.891E+03	6.161E+08	1.661E+08	3.301E+03	1.101E+08	3.541E+04	1.481E+05	1.001E+03	6.461E+02
	std	4.271E+03	1.001E+03	4.631E+02	5.151E+03	1.681E+08	1.031E+08	3.521E+03	4.661E+07	1.631E+05	9.001E+05	3.331E+02	1.971E+02
	f-rank	4	2	7	6	12	11	5	10	8	9	3	1
F_{19}	mean	5.191E+03	7.811E+03	1.141E+01	7.791E+01	1.900E+02	1.200E+02	1.050E+01	4.000E+01	1.110E+01	2.280E+01	1.090E+01	9.301E+00
	std	5.671E+03	4.671E+03	2.031E+00	1.781E+00	3.421E+01	3.821E+01	1.741E+00	5.911E+00	1.081E+01	1.431E+01	1.321E+00	1.111E+00
	f-rank	11	12	5	9	10	6	2	8	4	7	3	1
F_{20}	mean	2.611E+04	1.621E+04	6.270E+02	4.980E+03	3.590E+04	1.690E+04	2.100E+02	1.030E+06	3.020E+03	4.240E+03	1.172E+04	2.762E+03
	std	1.572E+04	4.112E+03	1.122E+03	3.402E+03	1.772E+04	6.572E+03	8.172E+01	9.052E+05	7.272E+03	3.822E+03	3.552E+03	1.442E+03
	f-rank	10	8	2	6	11	9	1	12	4	5	7	3
F_{21}	mean	1.112E+06	1.221E+06	2.581E+05	8.901E+04	2.521E+06	2.301E+06	2.721E+04	5.661E+06	5.081E+04	2.351E+05	5.601E+04	1.491E+03
	std	7.951E+05	7.961E+05	1.761E+05	1.071E+05	1.192E+06	1.352E+06	1.832E+04	2.732E+06	1.242E+05	2.392E+05	2.602E+04	1.322E+04
	f-rank	8	9	7	5	11	10	2	12	3	6	4	1
F_{22}	mean	1.882E+03	1.682E+02	2.082E+02	2.752E+02	1.022E+03	8.400E+02	2.100E+02	1.340E+03	6.810E+02	3.390E+02	4.170E+02	3.510E+02
	std	2.030E+02	2.470E+02	2.081E+02	1.451E+02	1.881E+02	1.281E+02	1.101E+02	1.881E+02	2.111E+02	1.781E+02	1.191E+02	1.052E+02
	f-rank	9	12	10	5	7	4	2	8	11	6	3	1
F_{23}	mean	4.112E+02	3.432E+02	3.152E+02	3.152E+02	6.002E+02	3.902E+02	3.202E+02	3.502E+02	3.152E+02	3.291E+02	2.041E+02	2.031E+02
	std	6.431E+01	2.841E+01	2.771E-01	1.601E+01	6.701E+01	2.471E+01	4.781E-01	7.661E+00	1.201E-01	4.031E+00	2.230E+01	2.230E+01
	f-rank	11	8	4	5	12	10	6	9	3	7	2	1
F_{24}	mean	1.470E+04	3.410E+04	2.000E+02	2.000E+02	4.000E+02	2.300E+02	2.400E+02	2.200E+02	2.460E+02	2.001E+02	2.001E+02	2.001E+02
	std	8.371E+03	2.351E+04	3.041E-03	1.501E-03	1.421E+01	2.531E+01	6.461E+00	2.511E+00	5.691E+00	1.562E-03	0	0
	f-rank	11	12	5	3	10	7	8	6	9	4	1	1
F_{25}	mean	5.291E+02	6.531E+02	2.041E+02	2.001E+02	2.401E+02	2.101E+02	2.101E+02	2.201E+02	2.081E+02	2.111E+02	2.001E+02	2.001E+02
	std	4.361E+01	6.010E+01	1.180E+00	8.072E-01	6.000E+00	1.070E+01	2.080E+00	4.440E+00	4.510E+00	2.820E+00	0	0
	f-rank	11	12	4	3	10	7	6	9	5	8	1	1
F_{26}	mean	2.122E+00	3.642E+01	1.002E+02	1.002E+02	1.002E+02	1.002E+02	1.002E+02	1.002E+02	1.012E+02	1.012E+02	1.001E+02	1.001E+02
	std	3.461E+00	5.621E+01	7.362E-02	9.552E-02	2.092E-01	4.302E-01	6.002E-02	2.062E-01	1.171E+00	1.532E-01	6.252E-02	5.342E-02
	f-rank	11	12	4	5	7	8	2	6	9	10	3	1
F_{27}	mean	1.951E+02	3.041E+02	4.091E+02	5.431E+02	1.081E+03	1.291E+03	5.801E+02	5.301E+02	8.822E+02	8.192E+02	3.582E+02	2.882E+02
	std	1.042E+02	1.602E+02	6.092E+00	1.362E+02	2.802E+02	3.302E+01	1.412E+02	7.922E+01	2.032E+02	9.172E+01	8.882E+01	1.022E+02
	f-rank	1	3	5	7	11	12	8	6	10	9	4	2
F_{28}	mean	1.942E+03	2.122E+03	4.342E+02	9.681E+02	1.391E+03	1.681E+03	9.701E+02	1.441E+03	1.291E+03	9.681E+02	2.901E+02	2.071E+02
	std	1.041E+02	4.441E+02	8.451E+00	4.121E+01	1.331E+02	2.331E+02	2.441E+02	9.461E+02	2.881E+02	1.061E+02	1.041E+02	3.971E+01
	f-rank	11	12	3	4	8	10	6	9	7	5	2	1
F_{29}	mean	1.981E+07	3.090E+07	2.140E+02	5.700E+05	5.700E+06	7.470E+06	1.210E+03	1.200E+06	4.800E+06	1.190E+06	2.090E+02	2.030E+02
	std	3.950E+06	6.910E+06	2.370E+00	2.140E+06	3.330E+06	1.201E+06	2.161E+02	7.031E+05	4.831E+06	3.251E+06	2.821E+00	2.451E+00
	f-rank	11	12	3	5	9	10	4	6	8	7	2	1
F_{30}	mean	6.951E+06	1.381E+07	6.690E+02	2.380E+05	1.490E+05	1.890E+05	4.080E+03	7.670E+04	1.760E+04	1.922E+04	5.152E+02	3.342E+02
	std	1.032E+07	1.082E+07	2.142E+02	1.102E+03	5.552E+04	1.032E+05	1.442E+03	3.372E+04	3.182E+04	8.252E+03	3.341E+02	1.511E+02
	f-rank	11	12	3	10	8	9	4	7	5	6	2	1
Average f-rank		7.30	7.30	4.77	5.00	9.97	9.16	4.37	8.97	6.47	7.43	4.80	2.57
Overall f-rank		7	7	3	5	12	11	2	10	6	9	4	1

[188]. These algorithms adopt the parameter settings outlined in [188], summarized in Table 3.1. To ensure an equitable comparison among these algorithms, simulation results are reported in terms of a fitness function's best, worst, mean, median, and *std* values. These values are computed based on 30 runs of each algorithm. The subsequent subsections comprehensively describe each of the design challenges mentioned above.

3.5.1 Pressure vessel design optimization problem

This problem represents one of the most frequently encountered design optimization challenges and is typically categorized as a mixed-integer problem. The primary objective of this optimization problem is to minimize the total cost associated with vessel material, welding, and pressure vessel formation, as detailed in [188]. Figure 3.3 illustrates the design of the pressure vessel. The design of this problem predominantly revolves around four key variables: the thickness of the cylindrical shell (T_s), the inside radius of the cylindrical shell (R), the head thickness of the cylindrical shell (T_h), and the length of the cylindrical segment (L). These variables are denoted as $\vec{b} = [b_1 b_2 b_3 b_4] = [T_s T_h R L]$.

Inference from results: The pressure vessel design problem simulation results are detailed in Table

Table 3.7: Simulated outcomes for CEC 2019 test functions

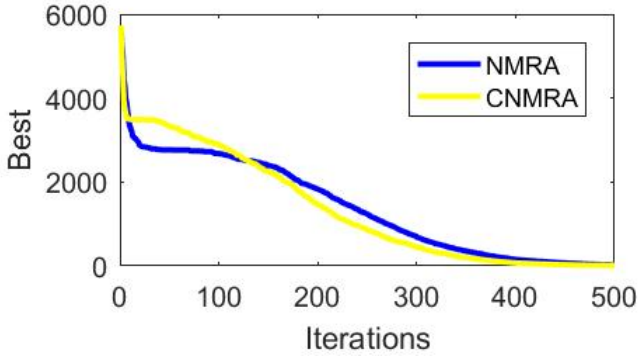
Functions		CS	SSA	ASSA	GWO	DA	NBA	FPA	NMRA	CNMRA
CEC-01	Best	1.000E+10	6.531E+07	6.441E+04	9.911E+04	2.691E+09	6.281E+11	7.331E+07	9.801E+04	6.392E+04
	Worst	1.000E+10	2.590E+10	1.291E+06	8.501E+08	4.291E+11	2.241E+13	9.452E+08	1.232E+06	1.012E+06
	Mean	1.000E+10	7.271E+09	4.941E+05	1.512E+08	8.512E+10	6.372E+12	3.621E+08	6.321E+05	4.090E+05
	Median	1.000E+10	5.921E+09	4.961E+05	7.851E+07	3.842E+10	4.712E+12	3.192E+08	5.832E+05	3.291E+05
	Std	0	6.281E+09	3.631E+05	2.032E+08	1.022E+11	4.962E+12	1.772E+08	3.271E+05	2.841E+05
CEC-02	Best	1.731E+01	1.731E+01	1.731E+01	1.731E+01	1.731E+01	4.511E+03	1.821E+01	1.831E+01	1.751E+01
	Worst	1.731E+01	1.731E+01	1.861E+01	1.731E+01	7.821E+02	2.511E+04	3.061E+01	1.981E+01	1.981E+01
	Mean	1.741E+01	1.731E+02	1.781E+01	1.731E+01	8.601E+01	1.631E+04	2.081E+01	1.931E+01	1.881E+01
	Median	1.731E+01	1.731E+01	1.761E+01	1.731E+01	2.781E+01	1.661E+04	2.011E+01	1.931E+01	1.881E+01
	Std	2.991E-05	3.301E-03	3.511E-01	2.741E-04	1.301E+02	4.581E+03	2.591E+00	6.401E-01	4.681E-01
CEC-03	Best	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01
	Worst	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01
	Mean	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01
	Median	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01	1.271E+01
	Std	1.271E-06	2.931E-04	3.661E-05	1.872E-06	4.272E-04	1.302E-03	1.412E-05	9.722E-06	1.122E-06
CEC-04	Best	1.592E+01	9.942E+00	1.022E+03	1.752E+01	2.022E+01	1.042E+04	7.030E+01	9.560E+01	8.880E+01
	Worst	4.080E+02	8.450E+02	4.760E+03	2.850E+03	4.450E+03	4.860E+04	2.340E+02	2.941E+02	2.221E+02
	Mean	2.812E+02	3.702E+02	2.602E+03	1.792E+03	4.932E+02	2.782E+04	2.222E+02	2.342E+02	2.080E+02
	Median	2.760E+02	3.380E+02	2.450E+03	5.370E+02	3.540E+02	2.710E+04	1.240E+02	4.410E+02	3.141E+02
	Std	5.131E+00	1.771E+01	8.611E+02	4.131E+02	7.181E+02	8.441E+03	2.831E+01	6.252E+02	3.962E+02
CEC-05	Best	1.032E+00	1.042E+00	1.622E+00	1.062E+00	1.082E+00	4.752E+00	1.402E+00	1.462E+00	1.302E+00
	Worst	1.092E+00	1.952E+00	2.872E+00	1.891E+00	2.341E+00	1.101E+01	1.781E+00	4.361E+00	3.491E+00
	Mean	1.061E+00	1.241E+00	2.301E+00	1.411E+00	1.601E+00	8.191E+00	1.582E+00	2.262E+00	2.232E+00
	Median	1.062E+00	1.222E+00	2.292E+00	1.342E+00	1.512E+00	8.272E+00	1.572E+00	2.172E+00	2.122E+00
	Std	1.612E-02	1.582E-01	2.472E-01	2.432E-01	3.352E-01	1.531E+00	8.472E-02	5.012E-02	1.432E-02
CEC-06	Best	7.411E+00	1.851E+00	9.711E+00	7.481E+00	6.491E+00	1.221E+01	8.901E+00	8.211E+00	1.031E+01
	Worst	1.081E+01	9.121E+00	1.251E+01	1.261E+01	1.151E+01	1.841E+01	1.171E+01	1.260E+01	1.260E+01
	Mean	9.360E+00	5.310E+00	1.130E+01	1.100E+01	9.320E+00	1.540E+01	1.060E+01	1.150E+01	1.140E+01
	Median	9.480E+00	5.320E+00	1.130E+01	1.110E+01	9.430E+00	1.561E+01	1.071E+01	1.161E+01	1.161E+01
	Std	8.422E-01	1.711E+00	7.291E-01	9.251E-01	1.311E+00	1.441E+00	6.491E-01	8.601E-01	5.651E-01
CEC-07	Best	8.861E+01	5.291E+02	1.621E+02	7.991E+01	1.451E+02	1.341E+03	8.322E+01	1.782E+02	7.592E+01
	Worst	7.462E+02	1.192E+03	1.382E+03	1.292E+03	1.092E+03	2.642E+03	6.192E+02	6.892E+02	5.651E+02
	Mean	3.081E+02	7.011E+02	7.621E+02	4.551E+02	5.881E+02	2.231E+03	3.110E+02	2.300E+02	2.100E+02
	Median	2.430E+02	2.890E+02	7.590E+02	3.870E+02	6.090E+02	2.270E+03	3.290E+02	2.400E+02	1.970E+02
	Std	8.840E+02	2.540E+02	2.470E+02	2.980E+02	2.290E+02	3.550E+02	2.030E+02	1.751E+02	1.561E+02
CEC-08	Best	4.871E+00	4.791E+00	5.091E+00	5.891E+00	5.281E+00	6.581E+00	4.991E+00	5.091E+00	4.782E+00
	Worst	6.892E+00	6.362E+00	6.982E+00	6.912E+00	6.672E+00	9.482E+00	6.952E+00	6.852E+00	6.582E+00
	Mean	6.012E+00	6.332E+00	6.102E+00	6.002E+00	6.091E+00	8.321E+00	6.851E+00	6.121E+00	5.981E+00
	Median	6.221E+00	6.291E+00	6.121E+00	6.221E+00	6.561E+00	8.441E+00	6.881E+00	6.141E+00	6.041E+00
	Std	3.601E-01	6.611E-01	4.611E-01	1.070E+00	5.871E-01	6.201E-01	2.741E-01	3.781E-01	3.961E-01
CEC-09	Best	3.560E+00	3.360E+00	1.070E+02	3.930E+00	3.870E+00	3.600E+03	3.880E+00	2.740E+00	3.120E+00
	Worst	3.190E+01	3.270E+01	6.561E+02	2.791E+02	6.911E+01	1.051E+04	5.191E+01	2.511E+02	1.921E+02
	Mean	2.831E+01	2.641E+01	3.371E+02	1.221E+01	4.861E+01	6.511E+03	4.051E+01	2.401E+01	1.121E+01
	Median	2.781E+01	2.601E+01	3.381E+02	1.811E+01	4.201E+01	6.211E+03	4.061E+01	4.721E+00	4.181E+00
	Std	1.651E+01	2.021E+01	1.201E+02	3.881E+01	3.661E+01	1.701E+03	5.031E+01	4.901E+01	2.941E+01
CEC-10	Best	2.011E+01	2.001E+01	1.981E+01	2.262E+00	2.002E+01	2.072E+01	1.962E+01	2.032E+01	2.022E+01
	Worst	2.082E+01	2.072E+01	2.082E+01	2.072E+01	2.092E+01	2.142E+01	2.092E+01	2.072E+01	2.061E+01
	Mean	2.061E+01	2.061E+01	2.071E+01	2.051E+01	2.061E+01	2.111E+01	2.071E+01	2.051E+01	2.040E+01
	Median	2.070E+01	2.050E+01	2.060E+01	2.060E+01	2.070E+01	2.120E+01	2.080E+01	2.050E+01	2.050E+01
	Std	6.251E-01	4.181E-01	1.581E-01	2.580E+00	1.751E-01	1.450E-01	1.520E-01	1.020E-01	9.920E-02

3.8. A thorough examination of this table reveals that CNMRA consistently outperforms DA, NBA, TLBO, LebTLBO, and NMRA across all metrics, including best, worst, mean, median, and *std* values. No other algorithm in the comparison achieves the level of performance demonstrated by CNMRA. Therefore, CNMRA is regarded as a highly promising candidate for generating optimal design variables for this specific problem.

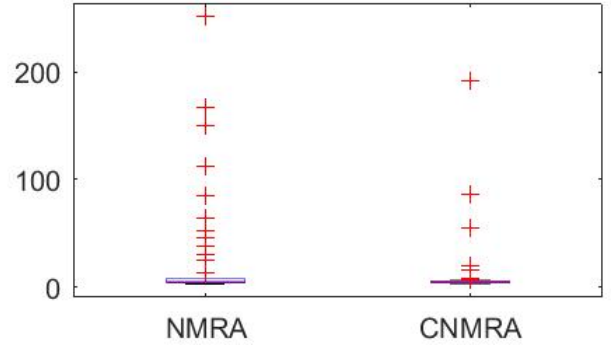
3.5.2 Welded beam design optimization problem

Another prominent engineering design optimization problem is the welded beam design, as discussed in [188]. This problem aims to minimize the production cost of designing a welded beam by determining optimal values for specific design factors. The design of this problem centers around four key factors: the length of the bar (l), the height (t), the thickness (b), and the weld thickness (h). To optimize these design factors, four constraints are considered: buckling load (P_c), shear stress (τ), deflection (δ), and bending stress (σ). These design variables are denoted as $\vec{x} = [x_1 x_2 x_3 x_4] = [h l t b]$. The design of the welded beam problem is illustrated in Figure 3.4.

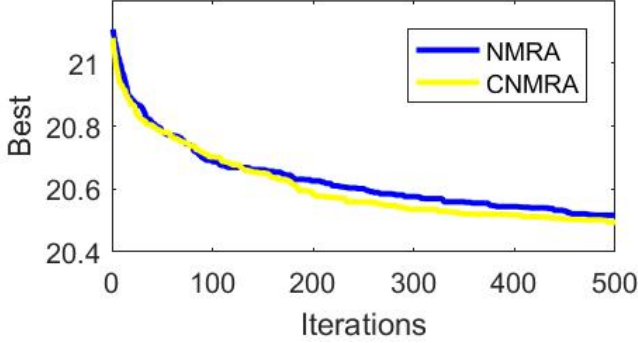
Inference from results: Regarding the welded beam design problem, you can find all the statistical



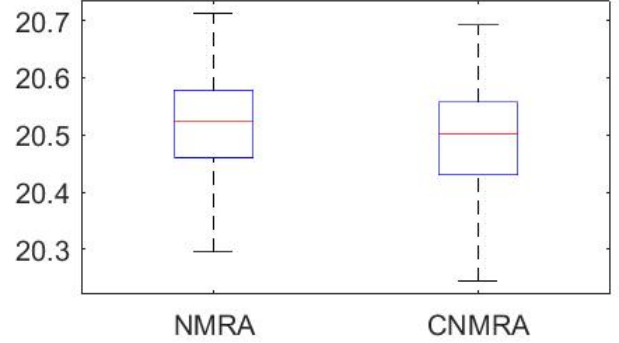
(q) Convergence profile for CEC-09



(r) Box plot for CEC-09



(s) Convergence profile for CEC-10



(t) Box plot for CEC-10

Figure 3.2: Convergence profiles and box plots of NMRA and CNMRA for 100-digit benchmark functions (CEC 2019) (continued)

outcomes in Table 3.8. These results indicate that TLBO outperforms other algorithms by delivering the best solutions in terms of mean and median values. However, it's worth noting that CNMRA excels in providing optimized solutions exclusively for the best values. Therefore, TLBO is a suitable algorithm for optimizing this problem.

3.5.3 Tension/compression spring design optimization problem

Optimizing the engineering design problem related to spring design is a complex multidisciplinary challenge, as discussed in [188]. The objective here is to reduce the weight of the spring design, which

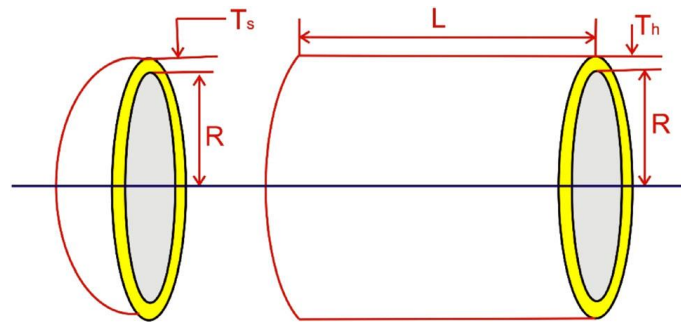


Figure 3.3: Pressure vessel design problem

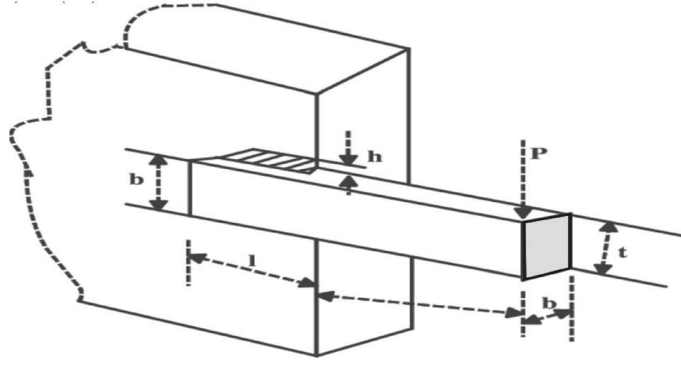


Figure 3.4: Welded beam design problem

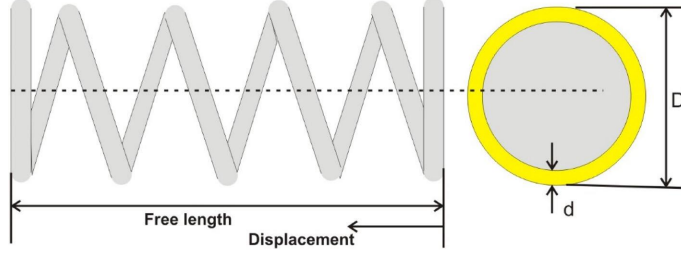


Figure 3.5: Design of spring problem

is depicted in Figure 3.5. The optimization process for this problem is subject to various constraints, including considerations for shear stress, minimum deflection, and limitations on the surge frequency. To tackle this engineering design problem, three design variables are involved: the number of active coils (N), the diameter of the wire (d), and the diameter of the mean coil (D). These design variables are denoted as $\vec{z} = [z_1 z_2 z_3] = [dDN]$.

Inference from results: To assess the performance of all the algorithms in solving this problem, the simulation results are provided in Table 3.8. A closer examination of these results reveals that, except for NBA, the remaining algorithms yield comparable results in terms of the best values. Therefore, the comparison is conducted based on mean, median, worst, and standard deviation values. In these aspects, CNMRA consistently outperforms all the other algorithms included in the comparison. Consequently, CNMRA emerges as the top-performing candidate for this particular problem.

Table 3.8: Simulation results for three engineering design optimization problems

Design problems		NBA [188]	DA [188]	TLBO [188]	LebTLBO [188]	NMRA	CNMRA
Pressure vessel	Best	5.440E+03	5.350E+03	5.350E+03	5.350E+03	5.370E+03	5.340E+03
	Worst	7.000E+03	1.350E+04	6.340E+03	6.160E+03	7.670E+03	5.730E+03
	Mean	5.890E+03	6.380E+03	5.450E+03	5.480E+03	5.970E+03	5.380E+03
	Median	5.760E+03	6.140E+03	5.350E+03	5.350E+03	5.900E+03	5.350E+03
	Std	4.130E+02	1.490E+02	2.430E+02	2.290E+02	7.140E+02	7.660E+01
Welded beam	Best	1.750E+00	1.740E+00	1.720E+00	1.750E+00	1.780E+00	1.720E+00
	Worst	1.000E+15	3.010E+00	2.700E+00	2.290E+00	4.010E+00	2.550E+00
	Mean	3.330E+13	2.140E+00	1.810E+00	1.820E+00	2.130E+00	1.930E+00
	Median	1.910E+00	2.030E+00	1.730E+00	1.800E+03	2.280E+00	1.870E+00
	Std	1.830E+14	3.240E-01	1.860E-01	9.390E-02	2.050E+00	1.950E-01
Tension spring	Best	9.900E-03	9.870E-03	9.870E-03	9.870E-03	1.000E-02	9.870E-03
	Worst	1.540E-02	1.410E-01	1.480E-02	1.400E-02	1.400E-02	1.040E-02
	Mean	1.170E-02	2.020E-02	1.120E-02	1.120E-02	1.140E-02	1.000E-02
	Median	1.110E-02	1.180E-02	1.100E-02	1.100E-02	1.100E-02	9.870E-03
	Std	1.630E-03	2.450E-02	1.210E-03	1.130E-03	1.200E-03	8.950E-05

3.6 Energy efficient wireless sensor network optimization problem

A WSN comprises many sensing devices that can identify, recognize, and convey the process and physical signals to a base station through the sink. WSNs are now viewed as revolutionary information-gathering technologies compared to wired alternatives since they offer quicker deployment and greater device flexibility [232]. The WSN comprises nodes capable of detecting changes, and all nodes are interconnected and send data to the sink as a group. The combination of sensing, central Processing, and radio to produce many applications is a transition that represents the operation of sensor networks [233]. These applications include creating smart buildings concepts, road network architecture, monitoring sewage to reduce blockages, and assisting persons in finding the closest parking spot in a new city [234] [235] with Kalman-based state estimation [236] and blockchain-enabled framework [237].

WSNs adhere to the OSI architectural model for communication. However, the network's topology experiences frequent changes as nodes join and exit the network concurrently. Energy is vital for sensor operations, and even idle node components consume power [238]. Consequently, energy sources must be recharged or replaced at specific intervals. Nevertheless, there are scenarios where recharging or replacing batteries is challenging due to various factors. Hence, there is a need for reliable alternatives to address such issues [239]. Additionally, SNs are characterized by their small size, which imposes inherent limitations on resources like energy, memory, processing speed, and communication bandwidth. These resource constraints directly impact the QoS offered by the network. QoS, in turn, defines the level of service provided to users [240]. In the context of WSNs, the effectiveness of an application is not solely determined by its broadcasting capabilities but also by its ability to track and monitor events. Therefore, the QoS of a WSN is highly application-specific, encompassing aspects such as event monitoring capability, network coverage area, data transmission latency, power consumption, and more [241].

In the context of WSN architecture, the network can be conceptualized as a smart room, smart building, smart hospital, or any similar environment to reduce the overall energy consumption of the entire network. The primary objective of our work is to introduce a generalized framework that can be applied to any network featuring static nodes. Our aim is to enhance the network's lifespan, ensuring that it remains operational for an extended duration before the first node depletes its energy. This is particularly crucial in healthcare applications, where network longevity is paramount. We organize the sensors into clusters within a cluster-based WSN architecture to achieve broader coverage, as outlined in [239]. These clusters comprise a CH, designated as the leader, and other nodes within the clusters referred to as cluster members. Energy-efficient routing plays a pivotal role

in minimizing energy consumption across the network. This is accomplished by optimizing various functional aspects, including sink communication, cluster design, CH selection criteria, CH rotation mechanisms, as well as intra-cluster and inter-cluster communication strategies, as discussed in [241] and [242].

Reduced energy usage and extended network lifetime are the most difficult parts of establishing and planning WSN operations in smart city WSNs [37] [42]. Clustering protocols similar to LEACH [148] are used to give prolonged lifespan and scalable solutions. The CHs lifetime is crucial in obtaining an extended network lifetime in such cluster-based routing techniques. Some CHs may be overburdened as a result of incorrect cluster formation. An overcharge like this could cause the CH to use too much energy, lowering the WSN's overall performance. As a result, proper CH selection is the most important part of sensor node clustering. It's not easy to develop a clustering algorithm that saves energy. As a result, NIA may be applied to solve the problem of this WSN. Evolutionary algorithms have emerged as effective meta-heuristics for addressing energy-conscious routing challenges. These algorithms involve the development of models aimed at maximizing a suitable energy-aware fitness function, as discussed in [163]. In this study, we leverage a clustering algorithm based on CNMRA to address the load-balancing issue mentioned earlier. The protocol's approach involves dividing the network into clusters, a strategy that contributes to reducing overall energy consumption.

Consider a WSN with n sensor nodes that are distributed randomly and arranged into K clusters: $C_1, C_2, \dots, C_K$. The energy consumption-based fitness function could be written as follows:

$$f = \left(\sum_{i=1}^K \sum_{seC_i} \sum_{TX_{s,CH_i}} + E_{RX} + E_{DA} \right) + \sum_{i=1}^K E_{TX_{CH_i}BS} \quad (3.15)$$

where K represents the total count of CHs, seC_i denotes the quantity of non-CHs affiliated with the i^{th} CH and $E_{TX_{node_1, node_2}}$ signifies the energy expended when transmitting data from $node_1$ to $node_2$.

The following formulas are used to compute radio energy communication costs for a k -bit message as a function of distance d :

$$E_{TX} = \begin{cases} kE_{elec} + k\epsilon_{friss_amp}d^2, & \text{if } d < d_0 \\ kE_{elec} + k\epsilon_{two_ray_amp}d^4, & \text{if } d \geq d_0 \end{cases} \quad (3.16)$$

where E_{elec} refers to the energy dissipation per bit during transmission. The parameters ϵ_{friss_amp} and $\epsilon_{two_ray_amp}$ imply transmitter amplification parameters for free-space and multi-path fading models.

The value of d_0 is calculated as follows:

$$d_0 = \sqrt{\epsilon_{friss_amp}/\epsilon_{two_ray_amp}} \quad (3.17)$$

The following equation can be used to express the reception energy of a k -bit data message:

$$E_{RX} = kE_{elec} \quad (3.18)$$

where E_{elec} is the energy dissipation per bit during receipt.

E_{DA} is the data aggregation energy cost set to $5nj/bit$. The values of the remaining parameters are set to $E_{elec} = 50nj/bit$, $\epsilon_{friss_amp} = 100pj/bit/m^2$ and $\epsilon_{two_ray_amp} = 0.0013pj/bit/m^4$ [243].

For the simulation, a total of 100 nodes are randomly distributed within a sensing field measuring $100m \times 100m$, with a BS located at the center of the field. All homogeneous nodes are initialized with an initial energy level of $E_0 = 1$. The network's lifespan in this analysis is characterized by three key parameters: FND, HND, and LND. These parameters indicate the communication rounds at which the first node, half of the nodes, and the last node in the network exhaust their energy and cease to operate. Here, we evaluate the performance of the proposed clustering protocol, which is based on CNMRA optimization, and compare it with other optimization techniques. These techniques include GA [53], PSO [244], BA [245], CS [246], GWO [23], mGWO [247], and the classical NMRA [4].

Table 3.9 presents the top-performing results for the CH selection problem in WSNs, demonstrating that the CNMRA algorithm consistently outperforms other algorithms. Following CNMRA, the mGWO, NMRA, CS, and GWO algorithms exhibit results that closely follow its performance.

Table 3.9: Comparative analysis of various algorithms for network lifetime estimation in WSN

Algorithm	Optimal cost of FF	FND	HND	LND
GA	39.797 mJ	2047.2	2678.3	3337.4
PSO	39.853 mJ	2034.9	2665.1	3324.7
BA	40.323 mJ	1996.8	2654.9	3298.8
CS	38.411 mJ	2079.3	2702.6	3372.3
GWO	38.559 mJ	2073.8	2696.4	3369.7
mGWO	38.208 mJ	2084.5	2714.5	3382.3
NMRA	38.383 mJ	2079.2	2709.9	3374.6
CNMRA	38.165 mJ	2091.6	2714.8	3391.2

3.7 Summary

This chapter involves enhancements to the performance of NMRA by incorporating chaotic maps. As specified earlier, the newly proposed NMRA variants leverage ten distinct chaotic maps. These variants are rigorously evaluated using CEC 2005 test suite, considering various population and dimension sizes. The empirical findings consistently highlight CNMRA as the top-performing variant

among all. Additionally, CNMRA is subjected to comprehensive comparisons with other state-of-the-art algorithms for CEC 2005, CEC 2014, and CEC 2019 test suites, as well as three practical engineering design problems. A Wilcoxon rank-sum test and f-rank assess the statistical significance of CNMRA's performance. Combining experimental results and statistical analysis affirms CNMRA's superiority over other algorithms in the comparison set. Convergence profiles further authenticate the significance of CNMRA compared to the basic NMRA. Finally, the proposed CNMRA variant effectively implements the smart city concept using a WSN, showcasing its ability to support extended network lifetimes and energy-efficient solutions.

Chapter 4

A Novel Self-adaptive Hybrid Slime Mould Naked Mole-rat Algorithm for Numerical Optimization and Energy-efficient WSN

In this chapter, the focus is on WSN, which consists of nodes spread across a wide sensor field. Clustering is a strategy that helps optimize data collection in WSNs, reducing energy consumption by organizing nodes into clusters. The critical aspect here is the selection of suitable CHs and implementing an effective routing protocol to extend the WSN's lifespan. A novel self-adaptive hybrid algorithm called the Slime Mould Naked Mole-Rat Algorithm (SM-NMRA) has been developed to address these challenges. SM-NMRA combines the foraging abilities of SMA with NMRA to enhance the capabilities of the original NMRA. Additionally, it introduces a novel stagnation phase inspired by GWO and CS algorithms to tackle local optima stagnation issues. Furthermore, self-adaptivity has been incorporated into SM-NMRA's parameters. The performance of SM-NMRA is rigorously evaluated using the CEC 2005 and CEC 2014 numerical test suites. The results, supported by statistical analysis including the f -rank test, p -rank test, and convergence profiles, demonstrate the superior performance of SM-NMRA. Moreover, a clustering protocol for efficient CH selection in WSNs, based on SM-NMRA, outperforms other state-of-the-art techniques by improving network lifespan and reducing energy consumption.

4.1 Introduction

The NMRA [4] and SMA [5] are the two most recent advancements in the category of meta-heuristic optimization approaches. SMA and NMRA beat several other algorithms reported in the literature, including DE [12], FA [21], PSO [244], GWO [23] and others. The SMA is based on the foraging behavior of slime moulds, whereas the NMRA is based on semi-arid mole-rat mating behavior. From the basic structure of SMA and NMRA, it can be concluded that both SMA and NMRA have inherent flaws, such as less exploitation capabilities for SMA and poor exploration for NMRA, as well as early convergence to optimal solution. Therefore, it becomes evident that there is room for enhancing the performance of these algorithms. To address this, a novel hybrid strategy is introduced, leveraging the food-foraging characteristics of SMA and the mating patterns of NMRA. This hybrid approach

considers both algorithms' exploration and exploitation components and incorporates self-adaptive capabilities. The proposed method retains NMRA's fundamental structure but integrates SMA's mathematical equations into the worker phase while keeping the breeder phase unchanged. In this context, the breeder phase is primarily utilized for exploitation, while the worker phase focuses on exploration. Combining the strengths of these two algorithms in their respective phases is anticipated to yield a superior algorithm with a balanced blend of exploration and exploitation capabilities, well-suited for effectively addressing the optimization problems at hand.

In addition to hybridizing two algorithms (SMA and NMRA), this study introduces parameter modifications to enable self-adaptation within the algorithm. Incorporating self-adaptive features empowers the algorithm to function autonomously, eliminating the need for user-initiated adjustments during problem evaluation. The NMRA's two significant parameters mating factor (λ) is employed randomly, and breeding probability (bp) is set to constant value 0.5 in the original approach. After a successful investigation, the NMRA's λ parameter is paired with the linear decreasing distribution from 1 to 0, and bp parameter is paired with Weibull distribution [248] to propose the hybrid slime mould naked mole-rat algorithm (SM-NMRA). This proposed algorithm establishes a set of criteria for generating new solutions, and if these criteria aren't satisfied within a specified number of iterations, the stagnation phase is initiated. The GWO-CS algorithm's equations have implemented this stagnation period. The primary benefit of this proposed optimization approach SM-NMRA is that it retains the essential elements of both algorithms while addressing optimization problems and consistently producing results.

WSNs are gaining popularity in various industrial applications, including automobile traffic monitoring, public safety networks, and IoT [249]. A typical WSN consists of distributed sensors and BS interacting with the environment by sensing specific physical parameters [250]. These WSN nodes rely on non-rechargeable built-in batteries with a limited lifespan, as well as limited sensor memory, storage, and processing capabilities [41]. Furthermore, in addition to network formation, these sensors are often deployed in challenging physical environments where network tampering can incur high costs [41] [251]. Therefore, it is essential to extend the network's lifespan. In WSNs, sensors and the BS communicate through wireless radio signals, and designing an efficient routing algorithm is crucial to reduce energy consumption and extend the overall network lifespan [250]. Transmitting data over a series of shorter distances, with data aggregation and multi-hop routing, is more energy-efficient than direct transmission to the BS [250] [251]. Numerous energy-efficient clustering techniques have been proposed in the literature to address this challenge [250]. Many of these approaches involve clustering sensor nodes to facilitate data transmission to the sink node through multiple hops. The

performance of these algorithms relies on fitness parameters defined in [251]. There is a need for an energy-efficient clustering protocol that aims to improve trust awareness and enhance network lifetime.

The SM-NMRA algorithm is introduced to integrate the enhancements into developing an energy-efficient WSN framework. The proposed algorithm, detailed in the next section, explains how these modifications were incorporated into the NMRA and SMA. The effectiveness of the proposed SM-NMRA is assessed through various evaluations, including CEC 2005 test problems [7], CEC 2014 numerical test problems [8], and addressing the CH selection problem in WSNs [252] to design an energy-efficient network. Selecting the optimal CH is a complex problem with NP-hard characteristics, and one effective approach to tackle this issue is by employing an optimization algorithm. In this context, the SM-NMRA algorithm is utilized to design a trust-aware energy-efficient routing protocol known as TESMNMRP to extend the lifespan of WSNs. This protocol is rigorously compared with various techniques for performance evaluation. Through experimental and statistical analyses, SM-NMRA demonstrates its superiority over existing techniques. The primary contributions of this chapter can be summarized as follows:

- To augment the exploitation and exploration capabilities of the proposed algorithm, incorporated search equations inspired by both SMA and NMRA.
- To make the algorithm free from user-based parametric adjustments, the self-adaptive strategy has been employed.
- To ensure an appropriate equilibrium between exploration and exploitation operations, integrated a stagnation phase inspired by the GWO and CS algorithms.
- To assess the efficiency of SM-NMRA, it has been applied to CEC 2005 and CEC 2014 numerical benchmark problems, as well as the CH selection problems in WSN.

The remaining paper is organized into four sections. Apart from the introduction in Section 4.1, the proposal of SM-NMRA has been discussed in Section 4.2. Statistical analysis and demonstration of the algorithm's superiority over other prominent algorithms in the literature for CEC 2005 and CEC 2014 numerical test suites are covered in Section 4.3. In section 4.4, the proposed algorithm SM-NMRA is used to solve the problem of cluster head selection in WSN. Finally, Section 4.5 summarizes this chapter.

4.2 Self-adaptive hybrid algorithm: SM-NMRA

As mentioned earlier, the SM-NMRA algorithm combines the advantages of both NMRA and SMA. The algorithm comprises seven phases. The first phase involves random solution initialization, while the second phase is the worker phase, which incorporates all the enhancements designed to improve the exploration process. The third phase is the breeder phase, which remains unchanged from the classical NMRA. The fourth and fifth phases involve population adaptation and selection operations, respectively. The sixth phase of the proposed SM-NMRA algorithm is dedicated to parametric adjustments, and the seventh phase introduces a novel stagnation phase based on the GWO-CS algorithm.

4.2.1 Initialization of search agents

The SM-NMRA begins with search agents' initialization randomly within a defined range, with the following general equation:

$$SM_{i,j} = SM_{min,j} + rand(0, 1) \times (SM_{min,j} - SM_{max,j}) \quad (4.1)$$

where i corresponds to $[1, 2, 3, 4, \dots, n]$, n defines count of search candidates (population size), j defines for $[1, 2, 3, \dots, Dim]$, Dim is treated as optimization problem's dimension under test, $SM_{i,j}$ is i^{th} search agent's solution for j^{th} dimension, the test problem's lower and upper boundaries are $SM_{min,j}$ and $SM_{max,j}$, and $rand(0, 1)$ is distributed randomly between 0 & 1.

4.2.2 Worker phase

The exploration phase of the SM-NMRA optimization technique is based on the original SMA and NMRA. It has been observed that the worker phase, responsible for exploration, had room for improvement, prompting us to enhance its functionality. By incorporating essential features from SMA into NMRA's worker phase, we succeeded in enhancing the worker phase of NMRA. To develop SM-NMRA, we integrated the generalized equation from SMA into the worker phase of the original NMRA. It's important to note that using this generalized equation preserves the algorithm's original hierarchical structure. Consequently, NMRA's worker phase equations are applied for the initial half of the iterations, while SMA's equations are employed for the remaining iterations. Additionally, SM-NMRA retains the same parameters as those found in the basic optimization techniques (SMA and NMRA), without introducing new parameters.

4.2.3 Breeder phase

In implementing this phase of the SM-NMRA optimization technique, we use the original structure of NMRA. In this region, the population of breeder rats is limited and is tasked with mating with the queen, representing the global optimum solution. Exploitation is applied during the global search phase as it seeks a solution close to the currently best available one, and it is expected to yield a global solution as the iterations progress. It is important to note that the breeder phase of the new algorithm SM-NMRA remains identical to that of NMRA and has not undergone any modifications.

4.2.4 Population adaptation

The population size plays a crucial role in any optimization process, and it is a parameter of utmost importance. Together with the total number of generations, it determines the number of function evaluations required when solving optimization problems. It's essential to remember that a larger population size results in more function evaluations. Conversely, a smaller population and fewer function evaluations can sometimes lead to stagnation in local optima. This study incorporates a declining population technique in line with the approach outlined in [253]. This population-based strategy initiates the exploration process with the entire population and subsequently transitions to a more restricted population approach, yielding improved elitist solutions in the later stages. Moreover, it enhances genetic diversity, enabling the discovery of new solutions while still retaining the global optimum solution. The population adaptation equation is provided as follows:

$$n_{t+1} = \begin{cases} (1 - \Delta f_t^{best})n_t, & \text{if } \Delta f_t^{best} \leq \Delta f_{max}^{best} \\ (1 - \Delta f_{max}^{best})n_t, & \text{if } \Delta f_t^{best} > \Delta f_{max}^{best} \\ \min_n, & \text{if } n_{t+1} < \min_n \end{cases} \quad (4.2)$$

where n_{t+1} represents the population size at a given iteration t and Δf_t^{best} is defined as $(\frac{f_{t-1}^{best} - f_{t-2}^{best}}{|f_{t-2}^{best}|})$ for best fitness change and Δf_{max}^{best} is threshold value. It's worth noting that a minimum population size has been set to counteract the negative consequences of lower population levels. This type of distribution takes the form of an exponential curve at first, followed by a steady state for a few iterations, then an exponential decrease, and so on. It's worth mentioning that defining a user-based pattern is not required in this case.

4.2.5 Selection phase

The selection step in the proposed SM-NMRA algorithm is deemed highly significant. This study opts for a greedy selection approach, where the newly generated solutions are treated as the current local best solutions. If the newly generated solutions prove to be superior to the solutions from the previous generation, they replace the earlier solutions. The selection method for a generalized minimization process, with fitness, denoted as $f(SM_i^t)$ for the solution SM_i^t , is described by equation (4.3).

$$SM_{new}^{t+1} = \begin{cases} SM_{new} & \text{if } f(SM_{new}) < f(SM_i^t) \\ SM_i^t & \text{otherwise} \end{cases} \quad (4.3)$$

4.2.6 Parameters adaptation

In the proposed optimization technique SM-NMRA, the original NMRA parameters (bp & λ) play the essential role. Simple random values in basic NMRA define these parameters and must be adjusted for better results. As a result, these parameters have been changed so that no user adjustments are necessary.

Firstly, the parameter bp is determined using the Weibull distribution [248], which relies on three primary parameters: the shape parameter (β), the scale parameter (η), and the location parameter (γ). In relation to iteration t , the general distribution for this function is as follows

$$f(t) = \frac{\beta}{\eta} \left(\frac{t - \gamma}{\eta} \right)^{\beta-1} e^{-\left(\frac{t - \gamma}{\eta} \right)^\beta} \quad (4.4)$$

where $f(t) \geq 0$, $t \geq 0$, $-\infty < \gamma < \infty$ or $\gamma, \beta > 0$, $\eta > 0$. The location parameter (γ) is seldom used and is typically set to zero, resulting in a two-parameter Weibull distribution. The shape parameter (β) is the most significant in this context, as it determines the distribution's characteristics. For example, if $\beta = 3.602$, the distribution resembles a normal distribution; if $\beta \leq 1$, it exhibits an L-shaped distribution, while for $\beta > 1$, it takes on a bell-shaped form. A large β value leads to a high peak in the distribution. Additionally, if we simulate the exponential lifetime based on this distribution, the failure rate for a specific application increases over time when $\beta < 1$, decreases when $\beta > 1$, and remains constant when $\beta = 1$. The parameter η is adjusted based on the total number of iterations required, while the other parameters remain unchanged. For this study, a value of $\beta = 2$ was selected following a comprehensive review of the literature on the Weibull distribution

[254].

Secondly, parameter λ is implemented using a linearly decreasing function ranging from 1 to 0, yielding the best randomization outcomes. This technique is implemented using the following generalized equation:

$$\lambda = a \times rand - a \quad (4.5)$$

where a follows a linear decrease from 1 to 0 as iterations progress and $rand$ represents a random vector with values ranging from 0 to 1.

4.2.7 Stagnation phase

Combining the GWO and CS algorithms is a promising approach to address various real-world optimization challenges. This combination aims to strike a balance between exploration and exploitation operations, as demonstrated in previous studies [228]. The generation of new solutions in this hybrid approach is achieved as follows:

$$SM_x = SM_i - U_x(V_x \cdot SM_{new} - SM_i^t); \quad x = 1, 2, 3 \quad (4.6)$$

$$SM_{new}^{t+1} = \frac{SM_1 + SM_2 + SM_3}{3} \quad (4.7)$$

where $U = 2a.r_1 - a$ and $V = 2.r_2$ provides U_1, U_2, U_3 and V_1, V_2, V_3 respectively. The random integers r_1 and r_2 are distributed uniformly; however, parameter a follows a linearly decreasing distribution. Equation (4.6) generates solutions, which are then subjected to a search equation based on CS. To generate the new solution, Cauchy randomization is utilized. The Cauchy-based solution is created using the equation (4.8) as:

$$f_{Cauchy(0,h)}(\delta) = \frac{1}{\pi} \frac{h}{(h^2 + \delta^2)} \quad (4.8)$$

The Cauchy distribution function is computed using the following formula:

$$c = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{\delta}{h}\right) \quad (4.9)$$

The value of δ can be calculated using the equation (4.9), which is written as:

$$\delta = \tan(\pi(w - \frac{1}{2})) \quad (4.10)$$

where h is considered as a scale parameter and its value is set to 1 and $w \in [0, 1]$.

The above equations give you a Cauchy distributed random number between $[0, 1]$ and the final search equation modified by CS is:

$$SM_i^{t+1} = SM_i^t + \alpha \otimes \text{Cauchy}(\delta)(SM_{best} - SM_i^t) \quad (4.11)$$

The primary advantage of using a Cauchy-based distribution lies in its thicker tail, which allows for more substantial steps in exploring the search space. This characteristic helps the algorithm avoid getting trapped in local optima, thereby enhancing its search performance. While algorithms that solely rely on the best available solution may limit their exploration of the search space, the Cauchy-based mutation operator in SM-NMRA encourages the algorithm to thoroughly explore all regions of the search space around the current best solution. This results in more significant step sizes and contributes to SM-NMRA's superior exploration capabilities, achieving a balance between exploration and exploitation. The pseudo-code for the proposed SM-NMRA algorithm is provided in Algorithm 5.

4.3 Statistical results for numerical test suites

This section assesses the effectiveness of the proposed SM-NMRA algorithm across two distinct sets of numerical test problems. The primary objective of this performance evaluation is to check the algorithm's reliability compared to established algorithms. The section is organized into two subsections: the first one examines SM-NMRA's performance on 12 CEC 2005 numerical test problems [7], while the second subsection evaluates its capabilities on 30 CEC 2014 numerical test functions [8]. All simulations were conducted on a laptop equipped with a Ryzen 5 processor, 8GB of RAM, and MATLAB version 2016.

4.3.1 Performance evaluation for CEC 2005 test problems

This section is split into three parts. The first part describes the CEC 2005 test suite, while the second part deals with SM-NMRA dimension analysis with five sets of dimension size. The final part compares SM-NMRA to other hybrid algorithms and convergence profiles of NMRA, SMA and SM-NMRA.

Algorithm 5: Pseudo-code of proposed SM-NMRA

Start:

Inputs: Define the random population of NMRA: (n)

Decide count of breeder mole-rats (B) = $n/5$

Decide count of worker rats (W) = $n - B$

Initialize breeding probability value using Weibull distribution (bp)

Define problem's dimension (Dim)

Define SMA parameters (b & c)

Output: Evaluate best search candidate (SM_{best})

while $t \leq$ maximum iteration count (t_{max})

Worker phase

for $i = 1:W$

if Current iteration $\leq t_{max}/2$

 execution of NMRA worker phase using equation (2.3)

else

 execution of SMA equation by (2.9)

end if

end for

Breeder phase

for $i = 1:B$

 if $\text{rand}(0,1) > bp$

 execution of breeder phase using equation (2.4)

end if

end for

Population adaptation

 perform population adaptation by equation (4.2)

Selection operation

 perform greedy selection by equation (4.3)

Stagnation Phase

 obtain solution using equation (4.11)

Parameter adaptation

 update λ using linearly decreasing distribution

 unite new mole rats (W & B)

 update (SM_{best})

 increment t

end while

 save SM_{best}

Stop

- **Description of CEC 2005 test suite**

This part focuses on the detailed test suite used to assess SM-NMRA's trustworthiness. There are two basic classification versions of test problems: unimodal problems (t_1 to t_7) and multimodal problems (t_8 to t_{12}) [7]. The full summary of test problems that were chosen here is available in Table A.1. Unimodal functions have a single global minimum due to their simplicity and are used to verify the algorithm's credibility in utilizing the search space. Multimodal functions, on the other hand, have numerous local minima and are used to assess the algorithm's exploration capability. All test problems have the same optimum value ($f_{min} = 0$).

- **Dimension size analysis**

This part provides the dimensional analysis of SM-NMRA compared to basic algorithms, notably NMRA and SMA, for twelve CEC 2005 test problems (t_1 to t_{12}). For this analysis, five dimensions (Dim) values of 10, 50, 100, 200 and 500 are used and simulated outcomes for these dimension sizes are presented in Table 4.1. The results are presented in terms of the average (Avg) and Std values obtained from 51 runs, with a consistent total number of iterations set at 500 for all methods. Here are the simulated results for each dimension size:

Dimension size ($Dim = 10$): In this case, it was revealed that SMA and SM-NMRA can give results that are exactly equal to global minima for t_1 and t_3 . SM-NMRA outperformed SMA and NMRA in terms of obtaining global minima for t_2 and t_4 . Avg and Std values were compared in the case of t_5 , t_6 , t_7 , t_{11} , t_{12} and SMA generated better results. For t_8 , t_9 and t_{10} , all the three algorithms under comparison provide similar and optimum results.

Dimension size ($Dim = 50$): In this dimension size, for test problems t_1 , t_2 , t_3 , and t_4 , the proposed approach SM-NMRA is determined to be best and capable to reach optimal value. The results for t_5 , t_6 , t_{11} and t_{12} were compared for Avg and Std values, and SMA was found to be the best approach. In the case of t_7 , it can be demonstrated that SM-NMRA outperforms other approaches. In the last, NMRA, SMA, and SM-NMRA deliver exactly equal results for t_8 , t_9 and t_{10} .

Dimension size ($Dim = 100$): For problems with dimension size of 100, it's evident that SM-NMRA consistently achieves a global minimum solution for problems t_1 , t_2 , t_3 , and t_4 , whereas SMA and NMRA tend to get trapped in local minima. Among the algorithms considered, SMA produces the best results for test problems t_5 , t_6 , t_{11} , and t_{12} . However, when considering both Avg and Std values, the proposed method SM-NMRA surpasses NMRA and SMA for problem t_7 . Notably, the results are uniform for test problems t_8 and t_{10} , and all algorithms yield zero.

Finally, for the last case, t_9 , the solutions generated by all three techniques are near the optimal solution.

Dimension size ($Dim = 200$): The results obtained for this dimension size reveal that SM-NMRA provides the best value for problems t_1 , t_2 , t_3 , t_4 and t_7 . The performance of SMA is optimal for test problems t_5 , t_6 , t_{11} and t_{12} . For problems t_8 , t_9 , and t_{10} , the performance of NMRA, SMA and SM-NMRA is same and produces zero *Std* values.

Dimension size ($Dim = 500$): For test problems t_1 , t_2 , t_3 , t_4 and t_7 , SM-NMRA is proven to be a more efficient optimization algorithm than other optimization methods. The results for test problems t_5 , t_6 , t_{11} and t_{12} were compared for mean values, and SMA was found to be the best. Finally, for the cases t_8 , t_9 and t_{10} , NMRA, SMA, and SM-NMRA can approach an optimal solution.

Inferences: According to the simulation data presented in Table 4.1, it has been analyzed that for $Dim = 10$, all of the optimization methods perform well and provide considerable results. The computing time for the optimization algorithm grows as the dimension size increases and the algorithm's performance diminishes. SM-NMRA's performance has also deteriorated, but only to the point where it can be considered a viable option for higher dimension sizes.

Table 4.1: Simulated outcomes for dimension size of 10, 50, 100, 200, 500

Test problem	Algorithm	$Dim = 10$		$Dim = 50$		$Dim = 100$		$Dim = 200$		$Dim = 500$	
		Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
t_1	NMRA	1.258E-86	6.369E-85	3.266E-88	1.221E-87	2.169E-84	1.521E-83	1.020E-85	7.152E-85	3.468E-86	2.031E-85
	SMA	0	0	6.041E-269	0	2.422E-280	0	1.911E-241	0	1.258E-160	9.04E-160
	SM-NMRA	0	0	0	0	0	0	0	0	0	0
t_2	NMRA	5.811E-46	2.332E-45	1.387E-45	3.488E-45	4.076E-43	2.721E-42	9.987E-44	3.912E-43	4.312E-44	2.776E-43
	SMA	7.651E-190	0	6.722E-167	0	2.366E-144	1.385E-143	1.041E-110	7.088E-110	3.232E-01	1.421E+00
	SM-NMRA	0	0	0	0	0	0	0	0	0	0
t_3	NMRA	1.458E-87	1.031E-86	6.059E-89	4.076E-88	3.157E-83	1.421E-82	8.387E-86	5.931E-85	1.449E-85	7.921E-85
	SMA	0	0	8.448E-267	0	1.456E-199	0	6.078E-128	4.341E-127	2.000E-26	1.432E-25
	SM-NMRA	0	0	0	0	0	0	0	0	0	0
t_4	NMRA	8.758E-45	4.301E-44	5.141E-43	3.600E-42	2.657E-44	1.487E-43	3.651E-45	2.112E-44	3.367E-43	4.131E-44
	SMA	3.668E-196	0	1.149E-143	8.268E-143	1.521E-127	1.087E-126	1.401E-95	1.000E-94	3.366E-43	2.412E-42
	SM-NMRA	0	0	0	0	0	0	0	0	0	0
t_5	NMRA	8.959E+00	2.157E-02	4.887E+01	2.348E-02	9.889E+01	2.686E-02	1.988E+02	1.989E-02	4.979E+02	2.141E-02
	SMA	3.500E-01	1.378E+00	9.011E+00	1.368E+01	2.711E+01	3.451E+01	2.110E+01	3.742E+01	1.511E+02	1.700E+02
	SM-NMRA	8.682E+00	3.044E-01	4.858E+01	2.500E-02	9.818E+01	5.790E-02	1.974E+02	6.950E-02	4.952E+02	3.528E-01
t_6	NMRA	1.349E+00	6.411E-01	1.177E+01	4.500E-01	2.419E+01	4.885E-01	4.920E+01	5.831E-01	1.241E+02	4.551E-01
	SMA	2.191E-05	1.688E-05	6.311E-02	5.669E-02	9.887E-01	1.278E+00	2.900E+00	5.168E+00	1.841E+01	3.000E+01
	SM-NMRA	1.288E-01	8.050E-02	2.439E+00	6.367E-01	5.670E+00	1.505E+00	1.118E+01	2.726E+00	3.291E+01	7.189E+00
t_7	NMRA	6.088E-04	5.068E-04	7.389E-04	5.677E-04	6.985E-04	6.500E-04	6.931E-04	6.669E-04	7.231E-04	6.800E-04
	SMA	6.700E-05	5.351E-05	1.487E-04	1.558E-04	2.311E-04	2.211E-04	3.668E-04	4.621E-04	5.251E-04	4.500E-04
	SM-NMRA	9.317E-05	1.008E-04	9.527E-05	9.768E-05	8.979E-05	7.198E-05	8.538E-05	9.066E-05	9.268E-05	8.405E-05
t_8	NMRA	0	0	0	0	0	0	0	0	0	0
	SMA	0	0	0	0	0	0	0	0	0	0
	SM-NMRA	0	0	0	0	0	0	0	0	0	0
t_9	NMRA	8.881E-16	0	8.881E-16	0	8.881E-16	0	8.881E-16	0	8.881E-16	0
	SMA	8.881E-16	0	8.881E-16	0	8.881E-16	0	8.881E-16	0	8.881E-16	0
	SM-NMRA	8.881E-16	0	8.881E-16	0	8.881E-16	0	8.881E-16	0	8.881E-16	0
t_{10}	NMRA	0	0	0	0	0	0	0	0	0	0
	SMA	0	0	0	0	0	0	0	0	0	0
	SM-NMRA	0	0	0	0	0	0	0	0	0	0
t_{11}	NMRA	7.821E-01	4.212E-01	1.159E+00	1.458E-01	1.177E+00	9.200E-02	1.188E+00	4.200E-02	1.200E+00	3.379E-02
	SMA	2.232E-06	1.265E-06	3.101E-02	4.501E-03	2.300E-03	3.400E-03	2.600E-02	4.200E-02	2.200E-03	3.900E-03
	SM-NMRA	3.270E-02	4.840E-02	6.000E-02	2.450E-02	6.190E-02	1.790E-02	7.370E-02	2.580E-02	7.773E-02	3.600E-02
t_{12}	NMRA	9.031E-01	1.542E-01	4.989E+00	2.600E-03	9.987E+00	2.000E-03	1.989E+01	2.200E-03	1.987E+01	2.700E-03
	SMA	4.021E-04	2.889E-04	1.087E-02	1.641E-02	8.285E-02	1.488E-01	2.176E-01	3.868E-01	2.421E-01	4.878E-01
	SM-NMRA	2.091E-01	1.498E-01	1.912E+00	7.884E-01	3.898E+00	1.627E+00	8.223E+00	4.132E+00	1.851E+01	7.832E+00

- Comparative analysis of SM-NMRA with respect to other hybrid algorithms

This part evaluates the efficiency of SM-NMRA in comparison with other improved optimization

algorithms and is divided into three parts: the first part handles experimental testing (detailing *Avg* and *Std* values), the second part deals with statistical testing (rank-sum test and Friedman test) and last part describes convergence profiles of optimization techniques. All of these testing parameters are discussed as follows:

Experimental Testing: To analyze the working capability of SM-NMRA, it is compared with other improved versions of algorithms such as HHOSSA [255], LSHADE [256], OBL-EO [257], JADE [69], DADE [258], SOGWO [259], IWOA [130], SaDE [66], levy flight NMRA [96], classical NMRA and SMA. All of these methods have the same parameter selection as their original articles. In Table 4.2, the performance of all the algorithms under comparison has been measured in terms of *Avg* and *Std* values. The suggested SM-NMRA has the best performance for problems t_1 , t_2 , t_3 , t_4 and t_7 compared to other optimization methods. The HHOSSA produces the best results in the situation of t_5 . LSHADE can produce a nearly global minimal solution for the numerical problem t_6 ; other techniques may get stuck in a local minimal solution. For test problems t_8 , t_9 and t_{10} , HHOSSA, OB-L-EO, SOGWO, DADE, LNMRA, NMRA, SMA and proposed SM-NMRA all produce identical and exactly optimal results. Finally, in the problems t_{11} and t_{12} , JADE beats other algorithms. Thus, SM-NMRA generates optimal solutions for eight test problems, LSHADE & IWOA for one test problem, HHOSSA for four test problems, JADE and OBL-EO for two test problems, DADE, SOGWO & LNMRA for three test problems and SMA for five test problems from the 12 numerical test problems. As a result, the suggested SM-NMRA method is determined to be more efficient than other algorithms tested.

Statistical Testing: Wilcoxon's rank-sum test (p-rank) [231] and Friedman rank (f-rank) [260] are used to validate the algorithm SM-NMRA's simulated outcomes statistically. These tests aim to see if the suggested optimization algorithm is more efficient than other methods.

The algorithm's statistical significance is determined using the rank-sum test and given a p-rank as a consequence. SM-NMRA's utility is compared to other optimization techniques, and the findings are presented in Table 4.2. Each test problem's p-rank is allocated in the third row of this table, and the results are presented in terms of $win(w)/loss(l)/tie(t)$. Here, if comparing optimization algorithm surpasses SM-NMRA, then it is granted with "+" sign and called a $win(w)$ situation. The test algorithm performs worse than the SM-NMRA in the $loos(l)$ case, hence the "-" sign is used in its place. Finally, the compared algorithms are equally capable of solving the objective function in the $tie(t)$, as indicated by the "=" sign. As shown in the $w/l/t$ row of Table 4.2, the suggested algorithm SM-NMRA outperforms other optimization techniques in most cases.

Table 4.2: Comparative analysis of proposed SM-NMRA for 12 CEC 2005 numerical test problems

Test function		HHOSSA [255]	LSHADE [255]	JADE [69]	IWOA [5]	OBL-EO [257]	DADE [258]	SOGWO [259]	SaDE [69]	LNMR [96]	NMRA	SMA	SM-NMRA
t_1	Avg	7.811E-270	7.752E-19	1.800E-60	8.132E-146	6.758E-212	1.511E-80	6.049E-77	4.502E-20	3.900E-281	8.860E-82	0	0
	Std	0	2.379E-18	8.400E-60	4.366E-145	0	1.548E-80	1.49E-76	6.901E-20	0	6.328E-81	0	0
	p-rank	—	—	—	—	—	—	—	—	—	—	—	—
	f-rank	4	12	10	6	5	8	9	11	3	7	1	1
t_2	Avg	5.111E-141	3.350E-07	1.800E-25	2.378E-102	1.931E-108	4.867E-36	1.176E-44	1.901E-14	8.979E-130	2.385E-45	1.073E-159	0
	Std	2.559E-140	5.611E-07	8.800E-25	6.579E-102	8.000E-108	2.200E-36	1.341E-44	1.051E-14	6.411E-129	1.589E-44	7.666E-159	0
	p-rank	—	—	—	—	—	—	—	—	—	—	—	—
	f-rank	3	12	10	6	5	9	8	11	4	7	2	1
t_3	Avg	2.587E-171	1.679E-02	5.700E-61	1.541E+04	6.931E-187	2.748E-75	5.400E-22	9.000E-37	1.678E-287	5.273E-86	0	0
	Std	0	1.922E-02	2.700E-60	7.421E+03	0	1.948E-75	2.600E-21	5.421E-36	0	2.713E-85	0	0
	p-rank	—	—	—	—	—	—	—	—	—	—	—	—
	f-rank	5	11	8	12	4	7	10	9	3	6	1	1
t_4	Avg	3.221E-137	7.231E-01	8.200E-24	1.311E+01	4.732E-103	2.359E-37	1.178E-19	7.400E-11	3.478E-142	1.546E-44	4.538E-151	0
	Std	1.531E-136	6.258E-01	4.000E-23	1.611E+01	1.600E-102	1.910E-37	1.511E-19	1.800E-10	0	9.838E-44	3.240E-150	0
	p-rank	—	—	—	—	—	—	—	—	—	—	—	—
	f-rank	4	11	8	12	5	7	9	10	3	6	2	1
t_5	Avg	2.551E-02	1.931E+01	8.000E-02	2.641E+01	2.579E+01	2.500E+01	2.652E+01	2.100E+01	8.751E+00	2.896E+01	2.833E+00	2.871E+01
	Std	4.721E-02	1.622E+01	5.600E-01	6.600E-01	1.558E-01	3.800E-02	7.618E-01	7.800E+00	8.831E-02	2.010E-02	7.289E+00	1.410E-02
	p-rank	+	—	+	—	—	—	—	—	—	—	—	—
	f-rank	1	12	2	8	7	5	9	11	6	4	10	3
t_6	Avg	8.316E-04	4.939E-19	2.900E+00	3.628E-02	9.088E-05	4.500E-05	2.829E-01	9.300E+02	7.348E-02	6.839E+00	3.000E-03	1.181E+00
	Std	1.211E-03	1.187E-18	1.200E+00	6.948E-02	5.978E-05	1.331E-06	2.469E-01	1.800E+02	3.088E-02	5.165E-01	2.000E-03	3.888E-01
	p-rank	+	+	—	+	+	+	+	—	+	+	+	+
	f-rank	4	1	11	6	3	2	8	12	7	10	5	9
t_7	Avg	2.548E-04	5.400E-02	6.400E-04	1.849E-03	4.701E-04	2.365E-04	4.929E-04	4.800E-03	3.116E-04	5.579E-04	1.204E-04	9.595E-05
	Std	2.718E-04	3.132E-02	2.500E-04	2.358E-03	3.041E-04	5.987E-01	2.711E-04	1.198E-03	1.119E-04	3.993E-04	1.218E-04	1.057E-04
	p-rank	—	—	—	—	—	—	—	—	—	—	—	—
	f-rank	4	12	9	10	6	3	7	11	5	8	2	1
t_8	Avg	0	7.168E-01	1.000E-04	0	0	0	0	1.200E-03	0	0	0	0
	Std	0	2.119E+00	6.000E-05	0	0	0	0	6.500E-04	0	0	0	0
	p-rank	—	—	—	—	—	—	—	—	—	—	—	—
	f-rank	1	12	10	1	1	1	1	11	1	1	1	1
t_9	Avg	8.881E-16	5.678E+00	8.200E-10	3.728E-15	8.881E-16	8.881E-16	8.881E-16	2.700E-03	8.881E-16	8.881E-16	8.881E-16	8.881E-16
	Std	0	1.448E+00	6.900E-10	2.169E-15	4.011E-31	0	0	5.100E-04	0	0	0	0
	p-rank	—	—	—	—	—	—	—	—	—	—	—	—
	f-rank	1	12	10	9	8	1	1	11	1	1	1	1
t_{10}	Avg	0	1.429E-02	9.900E-08	2.638E-03	0	0	0	7.800E-04	0	0	0	0
	Std	0	1.628E-02	6.000E-07	1.100E-02	0	0	0	1.200E-03	0	0	0	0
	p-rank	—	—	—	—	—	—	—	—	—	—	—	—
	f-rank	1	12	9	11	1	1	1	10	1	1	1	1
t_{11}	Avg	9.588E-05	2.279E+00	4.601E-17	9.300E-03	6.286E-06	2.968E-02	5.608E-02	1.900E-05	2.000E-03	1.112E+00	1.800E-03	4.440E-03
	Std	1.548E-04	3.500E+00	1.900E-16	2.567E-02	4.348E-06	6.557E-03	1.418E-02	9.200E-06	8.487E-04	2.354E-01	2.100E-03	1.650E-03
	p-rank	+	—	+	—	+	+	+	+	+	+	+	+
	f-rank	4	12	1	8	2	9	10	3	5	11	6	7
t_{12}	Avg	2.787E-04	2.018E-01	2.000E-16	1.600E-01	3.828E-02	3.511E-05	3.527E-01	6.100E-05	1.158E-03	2.989E+00	3.200E-03	9.683E-03
	Std	3.817E-04	5.979E-01	6.500E-16	1.367E-01	9.748E-02	1.800E-08	1.279E-01	2.000E-05	6.100E-04	5.120E-02	4.000E-03	4.458E-03
	p-rank	+	+	+	+	+	+	+	+	+	+	+	+
	f-rank	4	10	1	9	8	2	11	3	5	12	6	7
w/l/t		4/5/3	2/10/0	3/9/0	2/9/1	3/7/2	2/7/3	2/7/3	2/10/0	3/6/3	0/9/3	3/4/5	NA
Avg f-rank		3.00	10.73	7.42	8.17	4.58	4.58	7.00	9.42	3.67	6.17	3.17	2.83
f-rank		2	12	9	10	5	5	8	11	4	7	3	1

Friedman rank (f-rank) is a statistical test that assigns f-rank to all optimization techniques under consideration. It is a straightforward test to determine whether the proposed optimizer performs significantly better or worse than other optimization techniques. The rank of each algorithm is listed in the fourth row of Table 4.2 for each numerical test problem. After allocating the f-rank to each algorithm, the overall f-rank and average f-rank value are evaluated in the last two rows of Table 4.2. SM-NMRA has demonstrated its statistical significance as an optimization algorithm, emerging as the top-ranked choice among all other algorithms when evaluated on the CEC 2005 test problems.

Convergence profiles: In this part, the convergence profiles of NMRA, SMA and suggested SM-NMRA for 12 test problems have been explained and displayed in Figure 4.1. To begin, it has been determined that SM-NMRA converges faster than NMRA, and SMA for numerical problems t_1 , t_2 , t_3 , t_4 , t_7 , t_8 , t_9 and t_{10} (see Figure 4.1). Original SMA converges to optimal value during early iterations for problems t_5 and t_6 ; however, SMNMRA and NMRA converge later. SM-NMRA reaches a global minimal value for numerical problems t_{11} and t_{12} in initial

iterations; however, NMRA fails to converge at the last iteration. As a result, when compared to SMA and NMRA, SM-NMRA has a faster convergence rate in most of the cases.

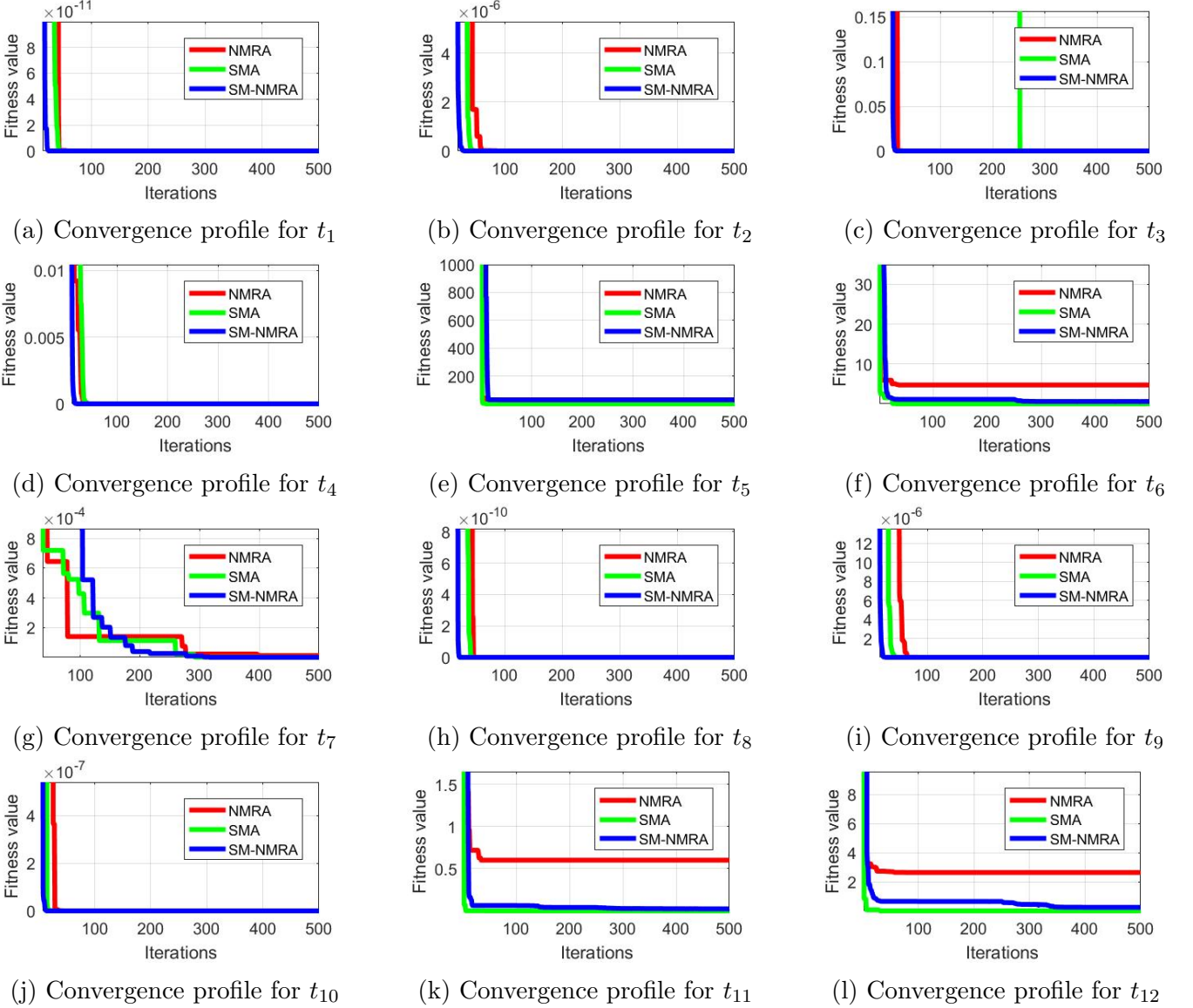


Figure 4.1: Convergence profiles of NMRA, SMA and SM-NMRA for 12 CEC 2005 numerical test functions

4.3.2 Performance evaluation for 30 CEC 2014 test functions

In this section, the SM-NMRA algorithm undergoes a comparative analysis with various well-established optimization algorithms, including HBA [261], MG-SCA [223], m-SSA [262], LX-BBO [219], mGWO [263], ISOS [220], DMFO [264], SMA, and NMRA. This comparison aims to showcase the superior performance of SM-NMRA. The performance evaluation of these methods is conducted using the CEC 2014 test functions [8]. These functions consist of a total of 30 test problems (CEC_{01} to CEC_{30}), comprising three uni-modal functions, thirteen multi-modal functions, six hybrid functions, and eight composition functions. The specifics of these test functions can be found in Table A.3. The maximum allowed number of function evaluations is set at $10^4 \times Dim$ of the optimization

problem. In this context, the dimension size is fixed at 30, resulting in a maximum of 3×10^5 function evaluations. It is important to note that the number of function evaluations is equivalent to the product of the population size and the maximum number of iterations. In this case, the population size is set to 50, and the maximum number of iterations is 6000.

Table 4.3 shows the simulation results for all of the optimization techniques under consideration in terms of *Avg* and *Std* error values for 51 runs. The performance of HBA is best among all the algorithms under comparison for test functions CEC_{01} , CEC_{03} , CEC_{04} , CEC_{06} , CEC_{08} , CEC_{09} , CEC_{10} , CEC_{11} and CEC_{13} . Both *Avg* & *Std* values are compared for the function CEC_{02} and two functions (CEC_{05} & CEC_{17}), where ISOS and m-SSA can yield results close to global minima respectively. In the case of test function CEC_{07} , the proposed SM-NMRA performs well for *Avg* values and is considered as suitable algorithm for this function. The results of LX-BBO optimization technique are better than other techniques for test functions CEC_{12} and CEC_{26} . The suggested algorithm SM-NMRA performed well when compared to other algorithms for functions CEC_{14} , CEC_{15} , CEC_{16} , CEC_{18} , CEC_{19} , CEC_{20} , CEC_{21} , CEC_{22} , CEC_{29} and CEC_{30} . The results of SMA and SM-NMRA for test functions CEC_{23} , CEC_{27} , and CEC_{28} are similar and better than other techniques under evaluation. SMA, m-SSA, NMRA, and SM-NMRA produce similar results and are close to the global optimum solution for test functions CEC_{24} and CEC_{25} . Aside from the simulation findings, a frank statistical test is run, and each optimization method is given a rank based on its performance. This statistical test is intended to determine whether SM-NMRA's performance is considerably superior than that of its competitors. Consequently, HBA provides the best results for nine functions from the total thirty test functions, ISOS for one, m-SSA for four, LX-BBO for two, and suggested SM-NMRA for sixteen functions. As a result, SM-NMRA demonstrates higher performance in most cases and the overall f-rank validates it.

4.4 Real-world optimization problem of cluster head selection in WSN

A WSN is a structure made up of densely distributed nodes that may process data, communicate, and perform computations in addition to supporting sensing. WSNs can be used for various tasks that require sensing and communicating information, including atmospheric monitoring, video surveillance, and a wide range of other tasks [37]. These can be placed anywhere, including open areas, streets, roadside areas, battlefields [41], industrial machinery interiors [37], commercial structures, and even inside or on human bodies.

In wireless networks, it is common practice to transmit sensed data to a BS, which is aggregated in whole or in part, as mentioned in [37]. Within WSNs, clustering is a widely employed technique that

Table 4.3: Simulated outcomes of SM-NMRA for CEC 2014 test problems

Problem		HBA [261]	MG-SCA [223]	m-SSA [262]	LX-BBO [219]	mGWO [263]	ISOS [220]	DMFO [264]	SMA	NMRA	SM-NMRA
CEC ₀₁	Avg	1.900E+05	2.921E+07	3.441E+05	1.011E+07	5.570E+06	9.821E+05	1.741E+06	2.141E+06	3.941E+06	1.462E+06
	Std	2.512E+05	1.242E+06	2.592E+05	1.012E+07	2.722E+06	7.052E+05	9.451E+05	7.801E+05	1.021E+06	1.981E+05
	f-rank	1	9	2	10	8	3	5	6	7	4
CEC ₀₂	Avg	5.180E+03	2.260E+09	1.301E+04	5.341E+04	1.801E+07	5.272E+00	1.412E+04	9.772E+03	7.502E+03	5.341E+06
	Std	7.321E+03	1.691E+09	1.260E+04	2.140E+04	2.630E+07	1.720E+01	1.512E+04	1.122E+04	3.462E+03	3.791E+06
	f-rank	2	10	5	7	9	1	6	4	3	8
CEC ₀₃	Avg	1.061E+01	1.771E+04	3.691E+02	1.632E+04	4.502E+02	4.792E+02	1.762E+04	1.342E+02	1.971E+04	2.841E+03
	Std	2.471E+01	6.631E+03	3.461E+02	1.700E+04	3.190E+02	6.240E+02	8.000E+01	1.651E+02	4.741E+03	2.621E+02
	f-rank	1	8	3	10	4	5	7	2	9	6
CEC ₀₄	Avg	1.081E+01	2.761E+02	2.311E+01	9.991E+01	1.291E+02	5.982E+01	8.002E+01	1.012E+02	1.812E+01	1.350E+02
	Std	2.190E+01	6.550E+01	3.690E+01	2.840E+01	2.290E+01	3.571E+01	3.501E+01	3.311E+01	6.381E+00	2.960E+01
	f-rank	1	4	3	7	10	5	6	8	2	9
CEC ₀₅	Avg	2.000E+01	2.040E+01	2.000E+01	3.060E+00	2.090E+01	2.030E+01	2.000E+01	2.080E+01	2.080E+01	2.050E+01
	Std	5.140E-02	1.441E-01	5.101E-03	7.861E-01	5.861E-02	6.671E-02	6.700E-02	1.930E-01	7.060E-02	1.440E-02
	f-rank	2	5	1	10	9	4	3	8	7	6
CEC ₀₆	Avg	2.520E-01	1.941E+01	7.921E+00	1.701E+01	2.251E+00	1.051E+01	1.881E+01	1.441E+01	2.292E+01	1.092E+01
	Std	7.351E-01	2.892E+00	2.842E+00	3.121E+00	9.787E-01	2.391E+00	3.161E+00	3.400E+00	1.770E+00	3.120E+00
	f-rank	1	9	3	7	2	4	8	6	10	5
CEC ₀₇	Avg	9.850E-01	1.989E+01	1.490E-02	1.750E-01	1.019E+00	1.561E-02	1.181E-02	3.471E-01	7.000E-04	2.320E-04
	Std	9.500E-01	1.182E+01	1.620E-02	8.560E-02	3.240E-01	1.830E-02	1.460E-02	1.340E-01	3.370E-04	7.310E-02
	f-rank	8	10	4	6	9	5	3	7	2	1
CEC ₀₈	Avg	4.370E+00	1.071E+02	7.761E+00	5.531E+01	1.761E+01	1.471E+01	2.951E+01	7.701E+00	9.291E+01	1.521E+01
	Std	1.362E+00	2.142E+01	2.792E+00	3.782E+02	4.952E+00	3.341E+00	8.141E+00	2.491E+00	1.171E+01	2.521E+00
	f-rank	1	10	3	8	6	4	7	2	9	5
CEC ₀₉	Avg	5.661E+00	1.392E+02	1.202E+02	7.662E+01	2.472E+01	2.562E+02	1.382E+02	9.832E+01	2.770E+02	1.880E+02
	Std	1.690E+00	2.600E+01	2.230E+01	1.610E+01	7.401E+00	1.341E+01	3.401E+01	2.011E+01	3.991E+01	1.731E+01
	f-rank	1	7	5	3	2	9	6	4	10	8
CEC ₁₀	Avg	6.431E+01	2.821E+03	3.311E+02	1.250E+04	3.580E+02	1.780E+03	9.920E+02	5.550E+02	1.380E+03	1.390E+03
	Std	1.950E+01	6.830E+02	1.480E+02	1.160E+02	2.090E+02	4.090E+01	3.110E+02	2.000E+02	2.110E+02	1.320E+02
	f-rank	1	9	2	10	3	8	5	4	7	6
CEC ₁₁	Avg	5.611E+01	3.301E+03	2.371E+03	1.231E+04	2.281E+03	1.481E+03	3.542E+03	3.042E+03	3.122E+03	3.272E+03
	Std	6.562E+01	6.261E+02	4.472E+02	3.412E+02	1.412E+03	4.542E+02	5.971E+02	6.801E+02	2.411E+02	4.791E+02
	f-rank	1	8	4	10	3	2	9	5	6	7
CEC ₁₂	Avg	3.458E-01	6.329E-01	1.088E-01	1.109E-02	2.411E+00	3.546E-01	3.011E-01	3.000E-01	8.000E-01	3.000E-01
	Std	7.068E-01	3.358E-01	7.448E-02	1.746E-18	2.619E-01	5.728E-02	1.269E-01	1.668E-01	9.549E-02	1.068E-02
	f-rank	6	8	2	1	10	7	5	4	9	3
CEC ₁₃	Avg	1.651E-02	5.511E-01	5.471E-01	6.551E-01	2.392E-01	3.772E-01	6.442E-01	5.000E-01	3.000E-01	3.000E-01
	Std	2.779E-01	8.938E-02	1.387E-01	1.559E-01	4.200E-02	7.100E-02	1.248E-01	1.011E-01	3.41E-02	3.039E-01
	f-rank	1	7	8	9	2	5	10	6	3	4
CEC ₁₄	Avg	1.358E-01	2.341E+00	5.542E-01	6.200E-01	2.148E-01	2.711E-01	9.000E-01	7.000E-01	2.000E-01	1.000E-01
	Std	4.766E-01	3.312E+00	3.300E-01	2.955E-01	3.341E-02	5.117E-02	3.158E-01	2.879E-01	2.448E-02	1.079E-02
	f-rank	2	10	6	7	4	5	9	8	3	1
CEC ₁₅	Avg	2.821E+01	8.721E+01	6.591E+00	1.551E+01	9.222E+00	1.062E+01	1.462E+01	6.600E+00	2.950E+01	1.750E+00
	Std	7.890E+01	1.011E+02	2.131E+00	5.491E+00	4.521E+00	3.712E+00	6.262E+00	1.972E+00	3.422E+00	3.191E+00
	f-rank	8	10	2	7	4	5	6	3	9	1

significantly reduces the energy consumption of individual sensor nodes. In cluster-based routing protocols within WSNs, CHs play a pivotal role. Their primary responsibility involves collecting periodic data from CMs and then forwarding the aggregated data to the central back-end server (BS). This approach is more energy-efficient for CMs than transmitting data directly to the BS. Consequently, the energy consumption in a cluster-based WSN is notably minimized because only one CH per cluster is engaged in the data aggregation and routing processes.

Employing a meta-heuristic approach is the preferred strategy to improve the performance parameters of routing protocols. In this chapter, the choice of CH relies on the hybrid SM-NMRA method. It is essential to highlight that in WSNs, the selection of CHs presents a binary-coded problem. Consequently, the proposed continuous SM-NMRA cannot be applied to address this specific challenge. Thus, a binary variant of the proposed SM-NMRA is taken into consideration, and the following equation is used to discrete the solution of the search candidate when it is updated:

$$Flag_{i,j} = \begin{cases} 1 & \text{if } SM_{i,j} \geq 0.5, \\ 0 & \text{otherwise} \end{cases} \quad (4.12)$$

where $SM_{i,j}$ denotes i^{th} search candidate's j^{th} solution.

Table 4.3: Simulated outcomes of SM-NMRA for CEC 2014 test problems (continued)

Problem		HBA [261]	MG-SCA [223]	m-SSA [262]	LX-BBO [219]	mGWO [263]	ISOS [220]	DMFO [264]	SMA	NMRA	SM-NMRA
<i>CEC</i> ₁₆	Avg	9.321E+00	1.161E+01	1.091E+01	1.082E+01	9.322E+00	9.212E+01	1.231E+01	1.071E+01	1.271E+01	1.121E+00
	Std	8.189E-01	6.908E-01	6.931E-01	5.839E-01	1.1881E+00	7.311E-01	4.728E-01	5.817E-01	2.035E-01	2.538E-01
	f-rank	2	7	6	5	3	10	8	4	9	1
<i>CEC</i> ₁₇	Avg	7.112E+05	9.562E+05	5.452E+04	1.491E+06	2.531E+05	1.751E+05	2.900E+05	2.970E+05	1.640E+05	1.460E+05
	Std	1.660E+05	7.620E+05	3.240E+04	9.340E+05	1.020E+05	1.640E+05	1.880E+05	1.740E+05	6.300E+04	2.290E+05
	f-rank	8	9	1	10	5	4	6	7	3	2
<i>CEC</i> ₁₈	Avg	4.421E+02	1.481E+05	2.191E+03	2.891E+03	1.921E+03	3.892E+03	5.322E+03	2.192E+04	9.222E+02	3.832E+02
	Std	8.762E+02	9.002E+05	2.571E+03	4.271E+03	1.561E+03	5.151E+03	6.041E+03	6.280E+03	2.910E+02	1.000E+02
	f-rank	2	10	5	6	4	7	8	9	3	1
<i>CEC</i> ₁₉	Avg	1.700E+01	2.280E+01	9.750E+00	5.190E+03	7.481E+00	7.791E+01	1.081E+01	1.822E+01	6.052E+01	4.372E+00
	Std	4.032E+01	1.432E+01	1.582E+00	5.672E+03	1.282E+00	1.782E+00	2.221E+00	2.401E+01	1.431E+00	5.561E+00
	f-rank	5	7	3	10	2	9	4	6	8	1
<i>CEC</i> ₂₀	Avg	2.471E+02	4.242E+03	2.442E+03	2.612E+04	2.372E+02	4.980E+03	3.310E+04	2.300E+02	1.000E+04	1.380E+02
	Std	5.370E+02	3.280E+03	1.240E+03	1.570E+04	5.670E+01	3.400E+03	1.921E+04	1.931E+02	4.371E+03	2.411E+02
	f-rank	4	6	5	9	3	7	10	2	8	1
<i>CEC</i> ₂₁	Avg	2.532E+04	2.352E+05	2.812E+04	1.112E+06	7.591E+04	8.901E+04	1.621E+05	1.461E+05	5.491E+04	1.100E+04
	Std	7.730E+04	2.390E+05	1.470E+04	7.950E+05	6.000E+04	1.071E+05	1.521E+05	6.541E+04	2.541E+04	3.521E+04
	f-rank	2	9	3	10	5	6	8	7	4	1
<i>CEC</i> ₂₂	Avg	4.531E+02	3.391E+02	4.622E+02	1.882E+03	1.482E+02	2.752E+02	5.400E+02	4.990E+02	4.160E+02	1.360E+02
	Std	1.111E+02	1.781E+02	1.321E+02	2.031E+02	6.571E+01	1.451E+02	1.651E+02	2.121E+02	1.172E+02	2.772E+02
	f-rank	6	4	7	10	2	3	9	8	5	1
<i>CEC</i> ₂₃	Avg	3.102E+02	3.290E+02	3.151E+02	4.111E+02	3.151E+02	3.151E+02	3.152E+02	2.002E+02	2.062E+02	2.000E+02
	Std	1.570E+02	4.030E+00	1.921E-06	6.430E+01	3.67E-01	1.600E+01	0	0	2.700E+01	0
	f-rank	7	9	5	10	6	8	3	1	4	1
<i>CEC</i> ₂₄	Avg	2.171E+02	2.000E+02	2.000E+02	1.471E+04	2.000E+02	2.000E+02	2.421E+02	2.000E+02	2.000E+02	2.000E+02
	Std	5.831E+01	1.555E-03	0	8.371E+03	6.448E-03	1.500E-03	5.802E+00	0	0	0
	f-rank	8	6	1	10	7	5	9	1	1	1
<i>CEC</i> ₂₅	Avg	2.011E+02	2.112E+02	2.000E+02	5.287E+02	2.058E+02	2.000E+02	2.000E+02	2.000E+02	2.000E+02	2.000E+02
	Std	9.300E+01	2.819E+00	0	4.358E+01	1.211E+00	8.068E-01	2.565E+00	0	0	0
	f-rank	7	9	1	10	8	5	6	1	1	1
<i>CEC</i> ₂₆	Avg	1.000E+02	1.011E+02	1.739E+02	2.121E+00	1.188E+02	1.000E+02	1.000E+02	1.011E+02	1.000E+02	1.000E+02
	Std	7.029E+01	1.528E-01	4.088E+01	3.457E+00	4.000E+01	9.546E-02	4.768E-01	1.387E-01	5.979E-02	5.011E-01
	f-rank	6	8	10	1	9	3	4	7	2	5
<i>CEC</i> ₂₇	Avg	2.431E+02	8.185E+02	5.011E+02	1.948E+02	3.626E+02	5.429E+02	7.318E+02	2.000E+02	3.411E+02	2.000E+02
	Std	6.438E+01	9.168E+01	9.500E+01	1.041E+02	3.000E+01	1.358E+02	2.269E+02	0	9.658E+01	0
	f-rank	4	10	7	3	6	8	9	1	5	1
<i>CEC</i> ₂₈	Avg	1.738E+02	9.858E+02	9.149E+02	1.936E+03	8.458E+02	9.679E+02	9.632E+02	2.000E+02	2.848E+02	2.000E+02
	Std	3.627E+02	1.058E+02	7.448E+01	1.039E+02	6.041E+01	4.118E+02	6.419E+01	0	1.018E+02	0
	f-rank	3	9	6	10	5	8	7	1	4	1
<i>CEC</i> ₂₉	Avg	2.778E+03	1.186E+06	1.758E+03	1.981E+07	1.1545E+04	5.700E+05	2.656E+03	9.669E+02	2.078E+02	2.000E+02
	Std	4.378E+03	3.2549E+06	4.349E+02	3.946E+06	4.381E+03	2.136E+06	1.839E+03	1.038E+03	3.621E+00	0
	f-rank	6	9	4	10	7	8	5	3	2	1
<i>CEC</i> ₃₀	Avg	1.022E+03	1.921E+04	2.611E+03	6.951E+06	6.391E+03	2.380E+05	4.240E+03	3.130E+03	5.060E+02	2.000E+02
	Std	2.200E+03	8.250E+03	6.151E+02	1.031E+07	1.541E+03	1.101E+03	1.381E+03	4.011E+03	1.101E+02	0
	f-rank	3	8	4	10	7	9	6	5	2	1
Avg f-rank		3.67	8.13	4.03	7.86	5.46	5.73	6.56	4.67	5.23	3.13
f-rank		2	10	3	9	6	7	8	4	5	1

The success of the clustering method in WSNs is significantly influenced by three key factors: the sensor's remaining energy, its distance from the Base Station (BS), and the node's trustworthiness. To create an efficient clustering solution, it's essential to assign appropriate weights to each of these parameters based on their respective impacts on energy consumption and the overall lifespan of the network.

4.4.1 Radio energy model

The energy consumption within a WSN can be categorized into several components, including sensing, data storage, and transmission. To determine the energy spent on packet transmission, the first-order radio model described in [252] is employed, as illustrated in Figure 4.2. In this model, the energy expenditure is calculated based on the separation between the sender and receiver nodes. The friis and two-Ray models estimate path loss depending on whether the separation is less than or equal to the threshold value d_0 . Specifically, the friis model is used when the separation meets this condition, while the two-Ray model is employed otherwise. The following equation determines the energy dissipation for each packet transfer:

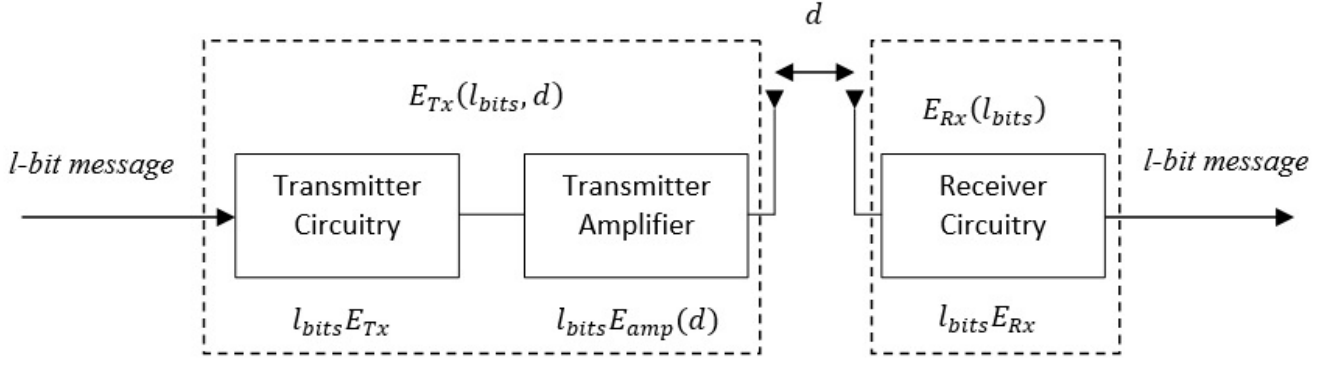


Figure 4.2: Radio energy dissipation model

$$E_{TX}(l_{bits}, d) = \begin{cases} l_{bits}E_{elec} + l_{bits}\varepsilon_{friis_amp}d^2, & \text{if } d < d_0 \\ l_{bits}E_{elec} + l_{bits}\varepsilon_{two_ray_amp}d^4, & \text{if } d \geq d_0 \end{cases} \quad (4.13)$$

where, E_{TX} represents the energy dissipation during transmission, l_{bits} corresponds to the packet length, and d signifies the distance between the transmitter and receiver. Additionally, an energy consumption component is associated with the processing of information by the transceiver circuitry, denoted as E_{elec} . The impact of the transmitter amplification model ($E_{amp}(d)$) is reflected in ε_{friis_amp} and $\varepsilon_{two_ray_amp}$. Determining which model to use relies on the crossover distance parameter, denoted as d_0 . This crossover distance is calculated using the following formula:

$$d_0 = \sqrt{\varepsilon_{friis_amp}/\varepsilon_{two_ray_amp}} \quad (4.14)$$

The energy dissipation for each packet reception is given by:

$$E_{RX}(l_{bits}) = l_{bits}E_{elec} \quad (4.15)$$

4.4.2 Clustering protocol using SM-NMRA

HRPs possess a layered architecture that involves two key stages: cluster formation and CH selection at the lower level, and routing at the upper level. This chapter introduces the use of SM-NMRA to assess the suitability of sensor nodes, addressing uncertainties in the CH selection process. TESM-NMRP is divided into two communication rounds: a setup phase and a steady-state phase. While the steady-state phase resembles that of DRESEP [243], there have been modifications to the CH selection technique during the setup phase. During the setup phase, BS leverages SM-NMRA to de-

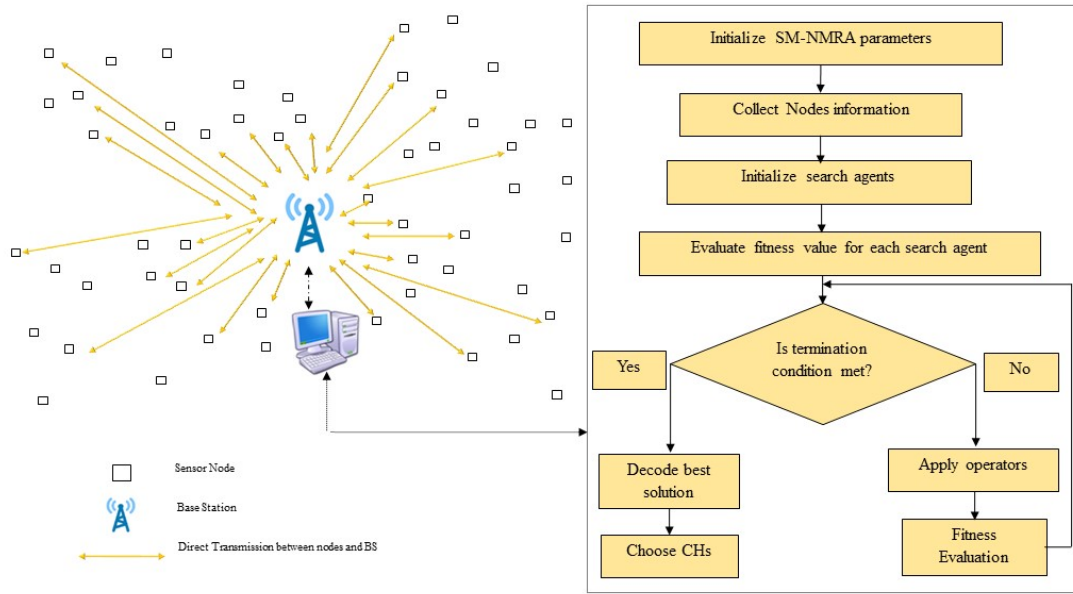


Figure 4.3: Selection of CH using SM-NMRA and exchanging information between sensor nodes and BS in setup phase

velop energy-efficient clusters based on the available number of active sensor nodes and the residual energy in the network. The following aspects are involved in the setup phase:

Election of CH: During the election phase, sensor nodes and the BS exchange critical information to control the network. The selection of CH nodes is carried out using SM-NMRA, utilizing the information gathered from sensor nodes, as depicted in Figure 4.3. The positioning of CHs and their member nodes within WSN is determined through a comprehensive approach, denoted as 1 and 0, respectively. This phase commences by generating an initial population of candidates, after which the fitness of each individual is computed based on a proposed fitness criteria, estimating their energy consumption. Subsequently, this population undergoes a series of iterations to evolve into a refined population. These iterative processes persist until the termination criteria are met. The most fit individual is then employed as a seed for the next step, which involves the establishment of connections between non-CH nodes and their respective CHs to establish clusters.

Setup of cluster: The chosen CHs communicate their status to the network by broadcasting an advertisement message that includes their unique CH ID. Employing the CSMA MAC protocol, nodes within the network respond to join requests from the CH that is geographically closest to them.

Creation of TDMA Schedule: After the CH election, BS takes on the responsibility of allocating

the TDMA schedule to the CHs for communication in each round. When it comes to intra-cluster communication among CHs and their member nodes, the BS assigns a distinct coding scheme to each CH. Subsequently, each CH broadcasts a CH announcement message to the entire network. Individual sensor nodes receive these broadcasted messages from the CHs and select the CH with the highest SNR to establish a connection. These individual sensors then confirm their chosen cluster association for the specific round where the CH is required. The CH subsequently assigns each node within its cluster a dedicated coding scheme and a time slot for transmission during the round. This time slot specifies when the node can transfer its data to the CH. At this stage, all clusters within the WSN have been established, each node has been assigned a specific time slot and coding scheme for communication with its respective CH, and the CHs themselves have been allocated time slots for forwarding all cluster data to the BS. The CHs then collect all the packets from their associated nodes, and this cycle repeats. If a sensor node encounters difficulties in joining a CH due to wireless link failures or the loss of announcement or join messages, it has the option to transmit its data directly to the BS for that particular round, as it is assumed that every node is within the communication range of the BS.

In this context, the steady-state phase can be subdivided into two distinct segments: intra-cluster communication and inter-cluster communication. Due to reactive protocol, sensing nodes engage in periodic data transmission to their respective CHs for intra-cluster communication [243], as illustrated in Figure 4.4. Meanwhile, depending on the distance between the CHs and BS, in the inter-cluster communication phase, CHs convey aggregated data to the BS via either direct transmission or a dual-hop pathway, as depicted in Figure 4.5, following the approach described in [243].

Evaluation of fitness: Consider a randomly distributed WSN with n sensors in the sensing field. For CH election, TESMNMRP employs a population of search agents that evolves to form optimal clusters while consuming the least amount of energy possible. Each search candidate is represented as a fixed-length list with the same length as the WSN's total number of nodes. Here, the population vector of a WSN with n sensors identified as $SM_{i,j} = [SM_1, SM_2, \dots, SM_n]$, where $SM_{i,j} \in [0, 1]$. The values 1 and 0 represent CH and regular nodes, respectively. The initial solutions of search agents are generated at random in terms of 0s and 1s based on the required percentage of CHs, as shown by:

$$SM_{i,j} = \begin{cases} 1, & \text{if } (rand \leq pr \text{ and } E(node_j) \geq E_{avg}(r)) \\ 0, & \text{otherwise} \end{cases} \quad (4.16)$$

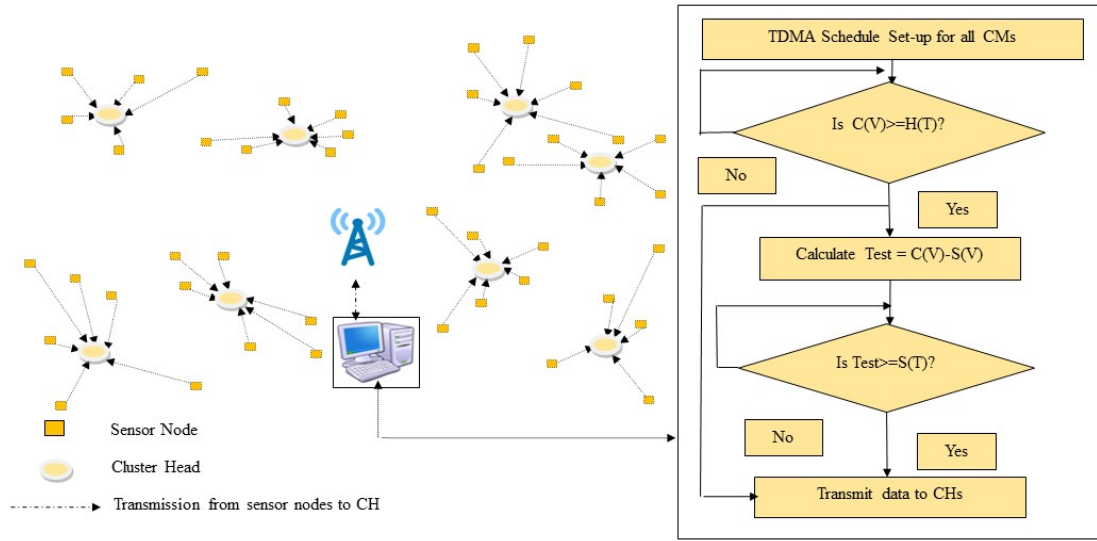


Figure 4.4: Transmission of packets from sensor nodes to CHs in intra-cluster communication

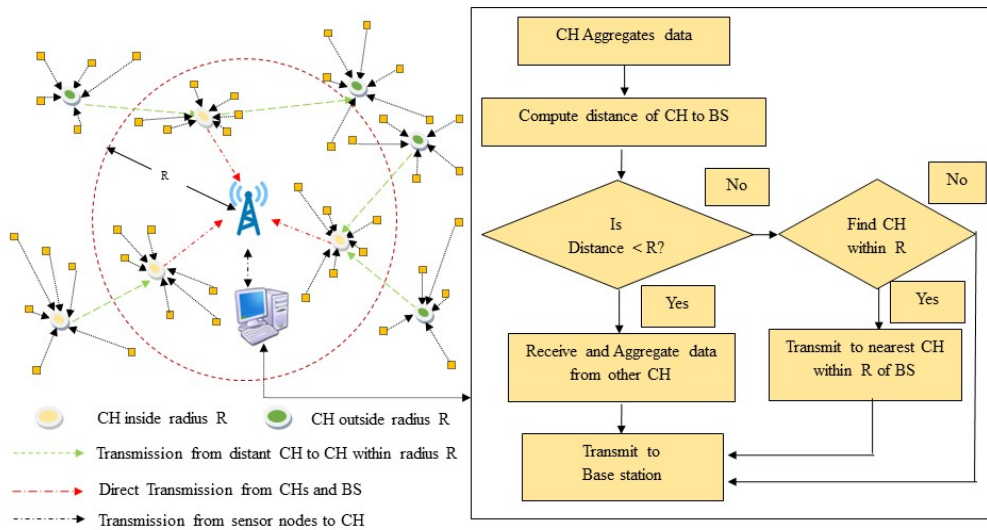


Figure 4.5: Transmission of packets from CHs to BS in inter-cluster communication

where, pr represents the estimated percentage of CHs, while $rand$ signifies a uniformly distributed random value ranging between 0 and 1, $E(node_j)$ signifies the current energy level of sensor node j and $E_{avg}(r)$ corresponds to the average energy of the entire network in the round.

The randomly placed SNs are organized into F clusters denoted as $[CH_1, CH_2, \dots, CH_F]$. The selection of CHs is strategically designed to minimize the cost associated with the fitness function, which is determined by factors such as minimizing energy consumption, minimizing cluster cohesion, and maximizing the separation between clusters. This fitness function can be represented as follows:

$$f_{CH,obj} = \sum_{i=1}^2 w_i \times f_i; \quad \text{subject to } \sum_{i=1}^2 w_i = 1 \quad (4.17)$$

The minimization of energy consumption for each node is of primary interest [163] so that longer stability period can be achieved and defined as:

$$f_1 = \left(\sum_{m=1}^F \sum_{node_j \in CH_m} E_{TX node_j, CH_m} + E_{RX} + E_{DA} \right) + \sum_{m=1}^F E_{TX CH_m, BS} \quad (4.18)$$

where F is the number of CHs, $node_j \in CH_m$ is a cluster member that belongs to the m^{th} CH and $E_{TX_{SN_1, SN_2}}$ is the amount of energy used to send data from SN_1 to SN_2 .

The second objective aims to enhance the quality of clustering, which is evaluated based on cluster separation and cohesion [42]. A good clustering scenario is characterized by a low ratio of cohesion to separation. Consequently, the aim is to reduce the intra-cluster distance (cohesion) between CHs and their associated CMs while simultaneously increasing the minimum inter-cluster distance (separation) between two distinct CHs. The fitness function for this objective is derived by considering the ratio of the overall Euclidean distance between CHs and their CMs to the minimum Euclidean distance between two separate CHs.

$$f_2 = \frac{\sum_{m=1}^F \sum_{node_j \in CH_m} d(node_j, CH_m)}{\min \forall CH_b, CH_m, CH_b \neq CH_m d(CH_b, CH_m)} \quad (4.19)$$

4.4.3 Inferences from simulated results

In this subsection, the performance analysis of TESMNMRP is evaluated in comparison to competitive protocols. To assess its performance under different configurations, MATLAB simulations are conducted. In the homogeneous configuration, a simulation environment is created with 100 sensors

distributed randomly within a $100 \times 100 \text{ m}^2$ area. These sensors start with an initial energy of E_0 , and a sink node is placed at the center. In the heterogeneous configuration, there are super nodes and advanced nodes accounting for 10% and 20% of the total sensors, respectively, and they begin with energy levels of $2E_0$ and $3E_0$, respectively. In this analysis, a communication round is defined as the time interval between successive cluster reformations. The estimation of the number of active sensors in each round is crucial for determining the network's lifetime. The network's lifespan is determined by measuring the number of communication rounds that occur before the last node exhausts its energy, and this also dictates the total amount of data transmitted. The specific parameter settings used for configuring the network are provided in Table 4.4.

Table 4.4: Parameters selection for setup of WSN

Parameter	Value used in simulation
Count of nodes	100
CH selection probability	0.05
Location of BS	(50, 50)
Size of WSN	$100\text{m} \times 100\text{m}$
Radio electronics energy, $E_{TX} = E_{RX}$	50nJ/bit
Energy for data-aggregation, E_{DA}	5nJ/bit
Normal node energy, E_0	0.25J or 0.5J or 1J
Radio amplifier energy, ε_{fries_amp}	100pJ/bit/m^2
Radio amplifier energy, $\varepsilon_{two_ray_amp}$	0.0013pJ/bit/m^4

The network lifespan of TESMNM RP for homogeneous setups with $E_0 = 0.25\text{J}$, 0.5J and 1J improves significantly as shown in Tables 4.5, 4.6 and 4.7 as compared to LEACH [148], ERP [42], DRESEP [243], SAERP [265], HSSTERP [168] and DESTERP [166]. These results highlight node cooperation's value in managing CH activities' burden. As a result, most nodes live together for as long as possible until practically dying together. Individual nodes appear to die less frequently than communities. This is in contrast to the case in which the residual energy between separate sensors has a lot of variation and is not evenly distributed.

As observed in Figure 4.6, the graph representing the number of alive nodes in TESMNM RP exhibits a distinct step-like pattern with a sharp decline. In contrast, the variations of LEACH display a more gradual decrease in the number of alive nodes over time. This implies that in LEACH, a substantial portion of SNs deplete their energy during the initial communication rounds. Conversely, TESMNM RP effectively addresses this issue and continues its efforts to prolong the network's lifespan whenever it proves beneficial to do so.

Table 4.5: Comparison of network lifespan for homogeneous configuration with $E_0 = 0.25\text{J}$ in terms of dead nodes round history

Dead Nodes (%)	LEACH	ERP	DRESEP	SAERP	HSSTERP	DESTERP	TESMNM RP
1 (FND)	461.1	508.5	782.5	609.2	1108.6	1368.3	1372.6
10	487.4	590.2	1130.6	610.8	1137.2	1393.6	1400.0
20	503.4	605.6	1229.3	611.9	1145.8	1406.8	1405.3
30	517.3	629.2	1284.2	612.3	1157.1	1409.1	1416.2
40	519.6	642.1	1340.2	612.8	1159.3	1415.3	1419.8
50 (HND)	565.6	649.2	1393.5	613.4	1173.1	1417.1	1423.6
60	590.3	657.9	1418.3	613.7	1182.3	1420.2	1429.7
70	592.3	673.8	1442.6	614.0	1188.6	1424.4	1431.8
80	609.7	700.2	1490.3	614.6	1218.1	1426.9	1435.9
90	628.2	734.6	1537.6	615.3	1220.3	1430.2	1441.5
100 (LND)	744.3	782.3	1600.2	616.9	1404.3	1432.9	1446.9

Table 4.6: Comparison of network lifespan for homogeneous configuration with $E_0=0.5J$ in terms of dead nodes round history

Dead Nodes (%)	LEACH	ERP	DRESEP	SAERP	HSSTERP	DESTERP	TESMNM RP
1 (FND)	970.8	1038.1	1563.7	1216.8	2400.0	2794.8	2835.4
10	1007.1	1158.9	2260.4	1220.5	2420.6	2822.1	2886.3
20	1039.8	1200.1	2464.3	1224.1	2442.2	2832.8	2896.1
30	1059.6	1236.8	2560.8	1225.9	2455.1	2845.2	2900.0
40	1075.3	1269.2	2681.2	1227.1	2457.6	2849.3	2907.9
50 (HND)	1170.2	1293.6	2782.9	1228.9	2461.3	2849.4	2911.2
60	1205.7	1320.5	2842.3	1231.1	2472.1	2851.1	2917.8
70	1264.8	1359.7	2880.9	1231.7	2484.9	2857.5	2920.4
80	1315.7	1411.2	2983.1	1233.0	2486.4	2860.3	2921.9
90	1370.3	1475.7	3080.6	1234.9	2502.3	2864.3	2926.7
100 (LND)	1671.6	1610.3	3200.5	1241.5	2554.6	2867.5	2930.5

Table 4.7: Comparison of network lifespan for homogeneous configuration with $E_0=1J$ in terms of dead nodes round history

Dead Nodes (%)	LEACH	ERP	DRESEP	SAERP	HSSTERP	DESTERP	TESMNM RP
1 (FND)	1805.1	2113.2	4101.9	2437.8	5635.8	5699.2	5742.9
10	2020.5	2275.8	4503.7	2444.2	5654.6	5711.6	5779.1
20	2067.8	2364.9	4770.2	2445.4	5667.1	5719.1	5795.5
30	2140.5	2437.6	4880.5	2447.6	5670.5	5721.8	5805.3
40	2170.6	2510.3	4984.1	2448.3	5676.3	5722.9	5810.2
50 (HND)	2213.8	2580.5	5127.0	2449.9	5680.2	5726.3	5812.9
60	2281.4	2651.2	5293.8	2451.1	5681.9	5727.1	5815.4
70	2345.8	2745.3	5396.3	2451.8	5688.2	5728.9	5816.3
80	2393.5	2836.9	5620.9	2452.2	5690.4	5733.1	5820.2
90	2484.8	2983.8	5771.5	2453.6	5692.8	5735.9	5823.3
100 (LND)	2763.8	3305.7	6400.8	2454.7	5715.9	5738.6	5834.7

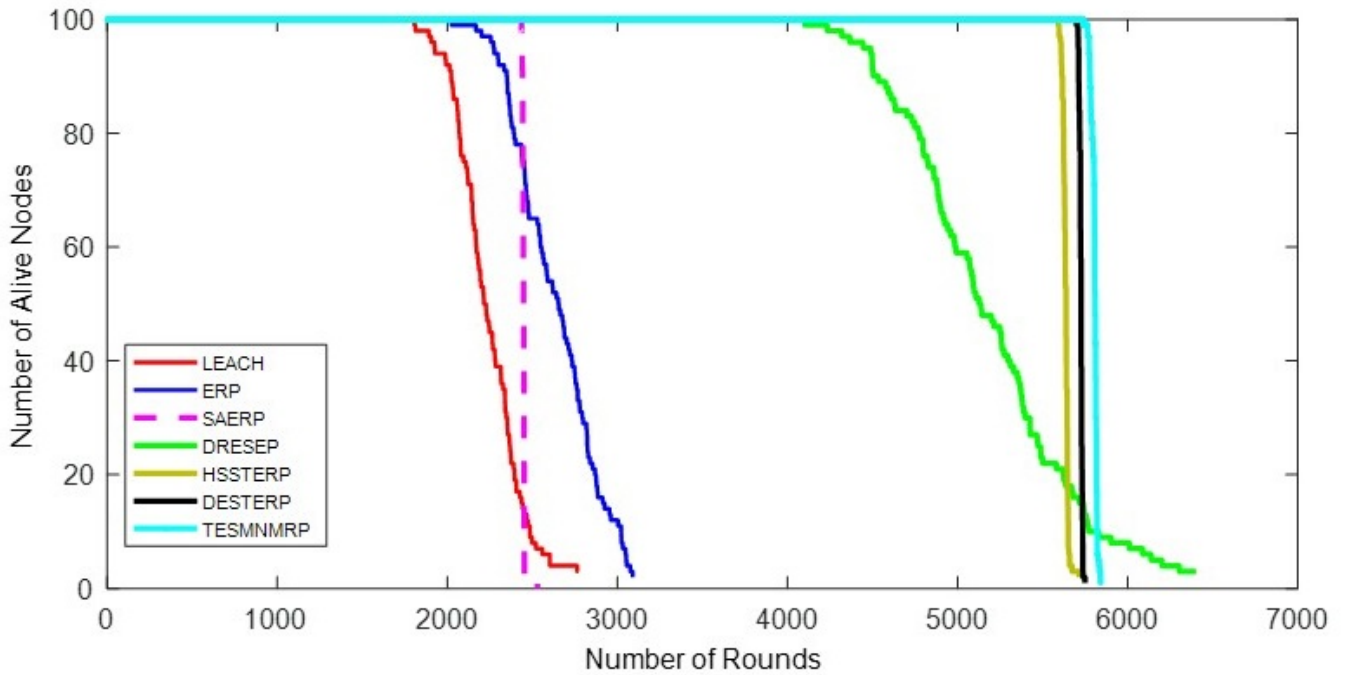


Figure 4.6: Count of alive nodes for homogeneous setup using $E_0=1J$ with progress of rounds

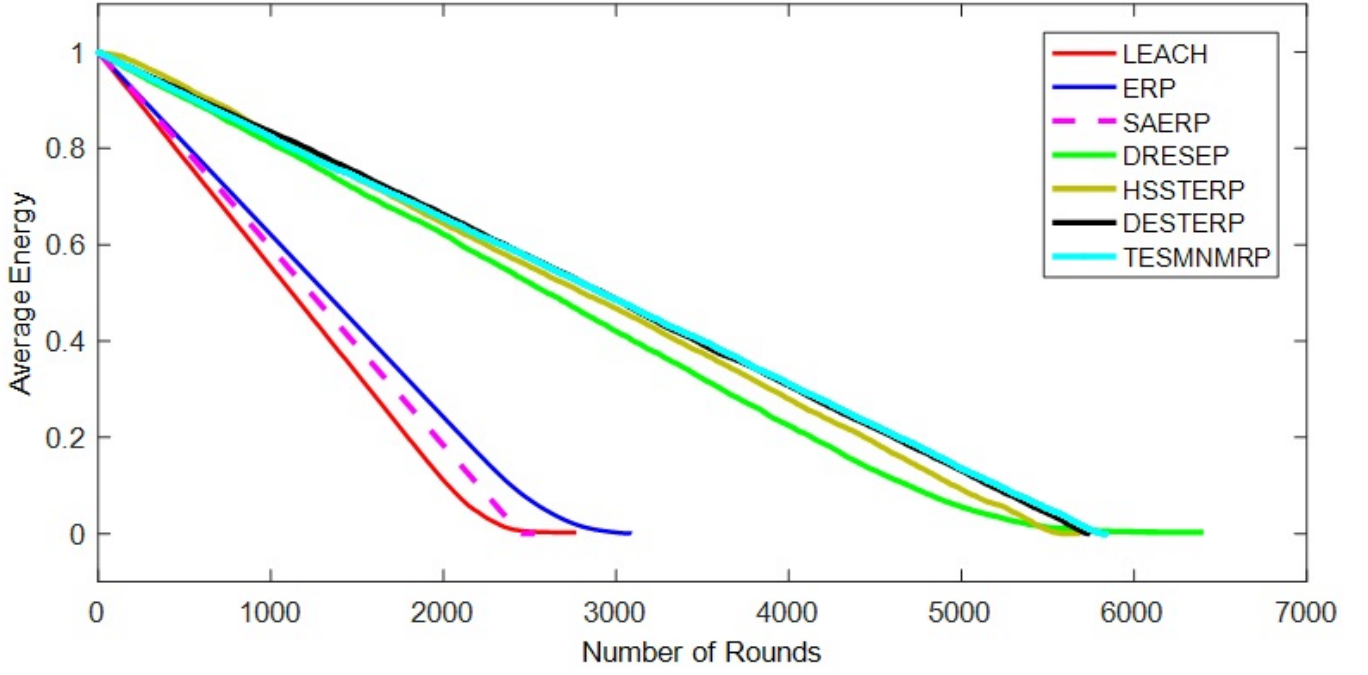


Figure 4.7: Average energy over rounds for homogeneous setup using $E_0 = 1J$

Figure 4.7 depicts the network's average remaining energy during the communication rounds for a homogeneous setup. TESMNMARP consumes less energy overall in comparison with other competitive methods. It demonstrates that the TESMNMARP absorbs energy consistently and steadily during each communication round.

Table 4.8, Table 4.9, and Table 4.10 present the operational efficiency of TESMNMARP in comparison to other state-of-the-art protocols within heterogeneous network configurations. In the case of SEP-E [266], the selection of CHs is entirely random, without distinguishing between regular, intermediate, and advanced nodes. Consequently, SEP-E tends to elect advanced and super nodes as CHs more frequently than regular nodes, leading to an imbalanced load distribution and faster depletion of CHs. DRESEP follows a similar pattern, where normal nodes have an equal chance of becoming CHs in the early rounds, similar to intermediate and advanced nodes. This approach also results in load imbalance, negatively impacting network performance. On the other hand, HSSTERP and DESTERP protocols prioritize selecting intermediate and advanced nodes as CHs during the initial stages, gradually equalizing the energy levels of advanced nodes with standard nodes as the rounds progress. In contrast, TESMNMARP adopts a different strategy for CH selection using SM-NMRA, taking into account residual energy exceeding a node's threshold energy level. Consequently, nodes with higher energy (intermediate and advanced nodes) are more likely to be chosen as CHs in TESMNMARP.

Figure 4.8 provides a visual representation of the number of active nodes throughout the commu-

Table 4.8: Comparison of network lifespan for heterogeneous configuration with $E_0=0.25J$ in terms of dead nodes round history

Dead Nodes (%)	SEP-E	ERP	DRESEP	SAERP	HSSTERP	DESTERP	TESMNMMP
1 (FND)	569.8	511.3	1212.8	746.8	1824.1	1845.4	1972.1
10	582.2	573.9	1338.3	793.5	1830.5	1858.2	1975.3
20	615.7	607.2	1400.6	795.2	1835.6	1861.5	1980.8
30	640.3	646.5	1448.3	795.8	1837.5	1862.1	1981.2
40	656.1	671.8	1504.0	796.3	1838.1	1862.9	1982.9
50 (HND)	677.8	692.8	1538.7	797	1839.8	1863.2	1983.5
60	701.2	776.9	1599.2	797.5	1845.1	1863.8	1986.1
70	815.8	1122.6	1980.5	798.1	1847.2	1864.1	1986.7
80	1070.5	1280.9	2441.3	798.8	1848.3	1864.9	1987.3
90	1155.2	1571.6	2646.8	799.5	1849.8	1865.5	1988.1
100 (LND)	1470.2	2112.3	3634.2	801.5	1851.8	1866.9	1988.7

Table 4.9: Comparison of network lifespan for heterogeneous configuration with $E_0=0.5J$ in terms of dead nodes round history

Dead Nodes (%)	SEP-E	ERP	DRESEP	SAERP	HSSTERP	DESTERP	TESMNMMP
1 (FND)	1151.8	1029.6	2442.7	1600.5	3658.3	3701.5	3655.1
10	1173.1	1154.3	2684.9	1625.1	3675.1	3725.8	3964.5
20	1244.6	1226.5	2818.1	1626.8	3679.5	3730.2	3970.4
30	1294.6	1301.8	2912.5	1627.9	3682.8	3732.1	3971.2
40	1323.2	1341.6	3017.3	1630.1	3686.2	3734.9	3974.6
50 (HND)	1368.7	1395.0	3088.6	1631.3	3690.5	3735.4	3976.0
60	1414.9	1563.9	3205.6	1631.8	3700.0	3736.2	3980.9
70	1643.7	2256.4	3980.7	1632.5	3703.7	3737.4	3981.7
80	2153.8	2569.1	4899.2	1634.2	3707.4	3738.6	3983.2
90	2322.4	3154.8	5302.8	1635.1	3710.1	3739.7	3985.4
100 (LND)	2955.5	4230.6	7283.8	1640.6	3713.9	3742.8	3986.2

Table 4.10: Comparison of network lifespan for heterogeneous configuration with $E_0=1J$ in terms of dead nodes round history

Dead Nodes (%)	SEP-E	ERP	DRESEP	SAERP	HSSTERP	DESTERP	TESMNMMP
1 (FND)	2267.8	2033.5	4848.2	2886.2	7286.1	7380.8	7877.5
10	2309.2	2285.1	5332.8	3134.6	7316.9	7423.1	7888.9
20	2450.6	2423.7	5595.4	3140.4	7325.4	7430.4	7900.7
30	2553.1	2579.9	5791.3	3146.1	7337.2	7434.8	7905.8
40	2605.6	2671.7	5996.1	3148.3	7341.4	7442.3	7914.1
50 (HND)	2700.5	2765.6	6139.8	3150.9	7346.3	7442.9	7926.3
60	2788.3	3101.2	6375.4	3152.6	7366.9	7443.2	7924.2
70	3253.5	4483.7	7919.8	3155.1	7372.6	7447.9	7929.7
80	4270.9	5109.2	9756.5	3158.2	7380.6	7449.3	7930.1
90	4604.4	6282.3	10569.1	3162.3	7386.4	7452.1	7933.8
100 (LND)	5874.1	8431.6	14527.6	3173.8	7396.1	7459.8	7936.9

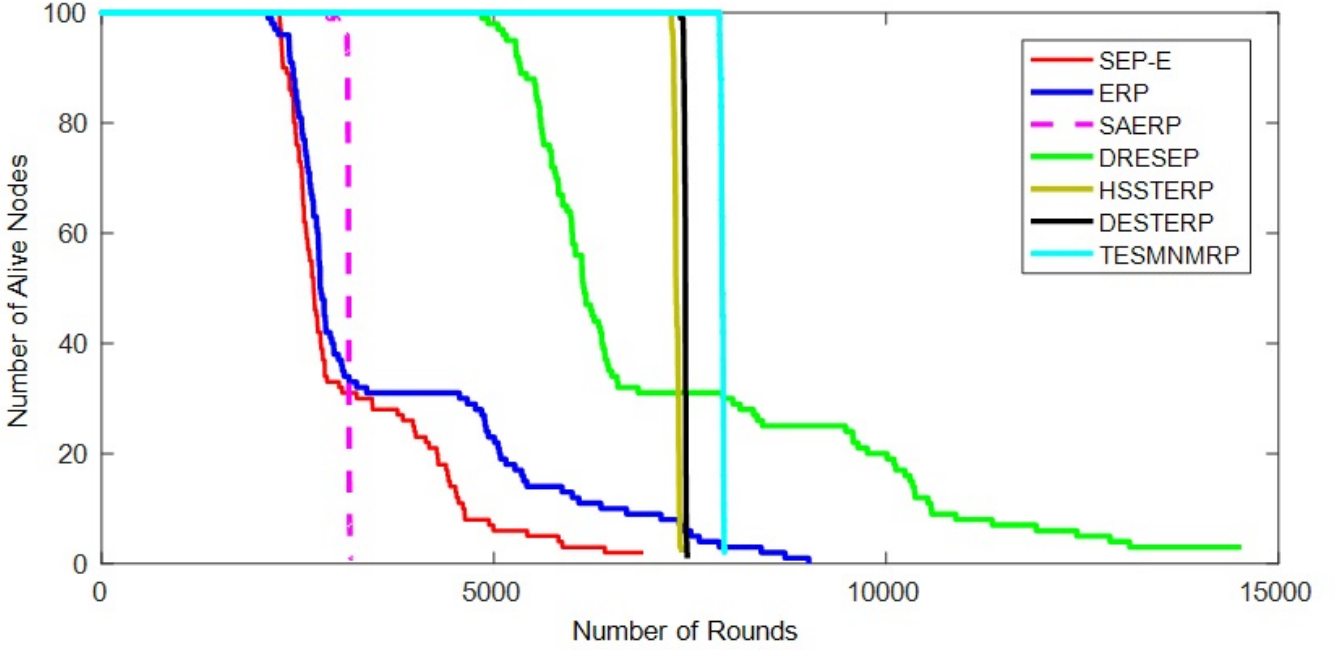


Figure 4.8: Count of alive nodes for heterogeneous setup using $E_0 = 1J$ with progress of rounds

nication rounds in a heterogeneous network setup, with an initial energy of $E_0 = 1J$, comparing the performance of SEP-E, ERP, SAERP, DRESEP, HSSTERP, DESTERP, and TESMNMRP. TESMNMRP outperforms SEP-E, SAERP, HSSTERP, and DESTERP by significantly extending the network's lifespan with a considerably larger number of communication rounds. This enhanced performance can be attributed to TESMNMRP's strategy of factoring in the residual energy of nodes when selecting CHs. As a result, high-energy nodes consistently serve as CHs, leading to a more balanced load distribution across the nodes, particularly during the initial rounds.

In a heterogeneous network configuration, Figure 4.9 depicts the average remaining energy within the network across communication rounds. When compared to other competing protocols, TESMNMRP exhibits more efficient energy utilization, demonstrating a consistent and gradual energy consumption pattern. This indicates that TESMNMRP effectively manages its energy resources, leading to lower overall energy consumption throughout the communication rounds.

4.5 Summary

This chapter presents a novel hybrid optimization algorithm called SM-NMRA, which combines the strengths of SMA and NMRA. In addition to this hybrid approach, SM-NMRA incorporates a unique stagnation phase inspired by a GWO-CS, aimed at preventing the algorithm from getting stuck in local optima. To enhance the adaptability of SM-NMRA, it introduces a self-adaptive mechanism through the parameter λ , which decreases linearly, and bp , is applied with a Weibull distribution. SM-NMRA is also subjected to dimensional analysis and demonstrates superior performance in higher-

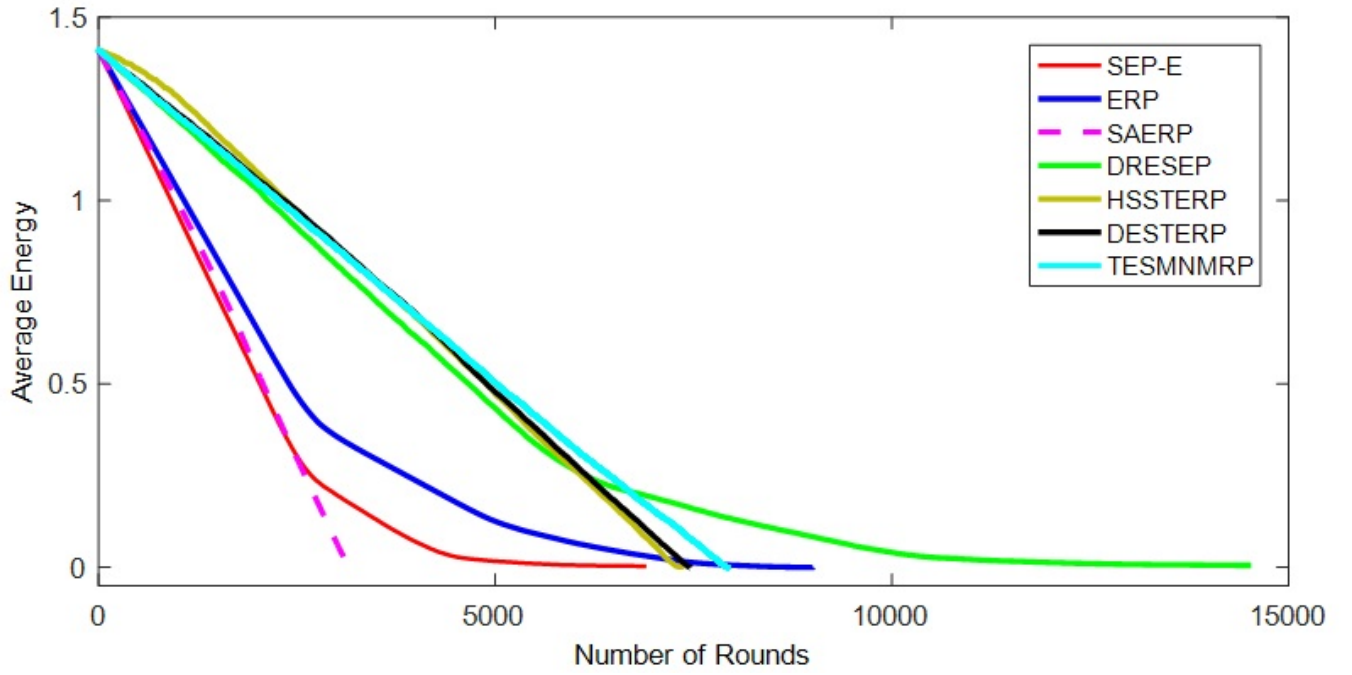


Figure 4.9: Average energy over rounds for heterogeneous setup using $E_0 = 1J$

dimensional problem-solving. The effectiveness of SM-NMRA is evaluated using CEC 2005 and CEC 2014 numerical test problems, outperforming many state-of-the-art algorithms in these benchmarks. Non-parametric statistical tests such as p-rank and f-rank are employed to establish the superiority of this proposed algorithm. Furthermore, SM-NMRA is applied to address the CH selection problem in WSNs and forms the new TESMNM RP clustering protocol. This protocol undergoes simulations across various WSN scenarios to evaluate total energy dissipation and the operational lifespan of the network. The results from these simulations indicate that TESMNM RP surpasses existing clustering techniques in terms of network lifespan and average energy dissipation.

Chapter 5

A Novel Whale Optimization based Naked Mole-rat Algorithm for Energy Efficient WSN

Nature-inspired algorithms have recently gained researchers' attention because of their problem-solving capabilities in real-world applications. WOA and NMRA have recently introduced algorithms and have proved significant advancements in terms of applications as well as modifications. However, current studies suggest that NMRA suffers from the problem of poor exploration, whereas WOA suffers from poor exploitation operation. So in this chapter, a hybrid variant of NMRA with WOA, using the added advantages of better exploitation of NMRA and exploration of WOA, is joined to formulate the whale optimization-based NMRA (WONMRA). In addition to introducing hybridization, the algorithm includes a simulated annealing-based mutation operation to enhance its self-adaptive capabilities. To evaluate the performance of the newly proposed WONMRA algorithm, it has been tested on both the CEC 2019 benchmark problems and the challenging NP-hard clustering problem in WSNs. For statistical validation, p-rank test and f-rank test have been conducted. The experimental outcomes clearly indicate that the proposed algorithm delivers highly significant and superior results when compared to other state-of-the-art algorithms.

5.1 Introduction

The most recent developments in NIA based on swarm intelligence include WOA, as discussed in [207], and NMRA, as introduced in [4]. Despite being the recent versions, WOA and NMRA outperformed a wide range of other algorithms, such as DE [12], FA [21], GWO [23], and others as reported in the literature. The WOA was based on the hunting habits of humpback whales, whereas NMRA drew inspiration from mole-rat mating behavior in semi-arid locations. Here, both algorithms employ a common pattern of foraging and interactions among different entities or species to optimize the given problem.

According to WOA's recent literature, firstly, modified WOA using the concept of population reduction for segmentation of COVID-19 X-ray images [267], improved WOA has been proposed with Gaussian random walks for image segmentation [268] and biogeography-based WOA for optimizing engineering design problems [269]. Secondly, different variants of NMRA have been developed, which include HS and DE-based NMRA for the synthesis of non-uniform circular antenna array [94] and

self-adaptive DE-based NMRA for engineering design problems [270].

Thus, when examining the fundamental structures of WOA and NMRA, it becomes evident that both of these algorithms exhibit inherent limitations. WOA suffers from poor exploitation, while NMRA struggles with insufficient exploration capability and a tendency towards premature convergence to optimal solutions. Therefore, there is a need for modifications to enhance the operational capabilities of these algorithms. A novel hybrid method has been introduced to address these shortcomings and create a more effective optimization approach. This hybrid approach leverages the exploratory strengths of WOA and the exploitative characteristics of NMRA. Furthermore, self-adaptive features have been incorporated into the algorithm and its exploration and exploitation capabilities, making it well-suited for tackling complex real-world optimization problems. In this new algorithm, the foundational framework of NMRA is retained, while all of WOA's mathematical equations are integrated into the worker phase, leaving the breeder phase unaltered. Since the breeder phase primarily focuses on exploitation and the worker phase on exploration, combining the two algorithms within their respective phases is expected to yield a superior algorithm that balances exploration and exploitation.

In this chapter, in addition to combining two algorithms (WOA and NMRA) through hybridization, there is a focus on adapting the algorithm's parameters to enable self-adaptation. The incorporation of self-adaptive features serves the purpose of making the algorithm autonomous, eliminating the need for user-based adjustments when addressing problem scenarios. A thorough investigation determined that NMRA's λ parameter could be integrated with a simulated annealing (*sa*) mutation operator, as described in [271], to create a self-adaptive implementation of the proposed algorithm. Based on these findings, a simplified hybrid version has been developed and named WONMRA. The primary advantage of this approach is its ability to retain the fundamental aspects of both algorithms while effectively addressing the optimization problems at hand, thereby yielding dependable results. In summary, the primary contribution of the proposed algorithm can be summarized as follows:

- To overcome the poor exploration properties of NMRA, the exploration phase inspired by WOA has been added. This is because of the better performance of WOA in terms of exploration operations.
- To make the algorithm self-adaptive, all the parameters are enhanced and made adaptive. The major added advantage is using simulated annealing adaptation instead of a simple random λ parameter.
- Apart from that, a greedy selection mechanism is followed to find the best solutions.

Overall, a hybrid variant, namely WONMRA, with the added advantage of both NMRA and WOA is proposed and applied to CEC 2019 benchmark problems, engineering design problems, and cluster head selection problems in WSNs.

The rest of the chapter is structured into three sections. In addition to the introductory content provided in Section 5.1, Section 5.2 provides the details of the proposed approach, WONMRA. Section 5.3 presents the experimental investigation and highlights the superior performance of the proposed algorithm when compared to other prominent algorithms in the existing literature. This section assesses WONMRA's capabilities by using numerical CEC 2019 benchmark problems and the optimization problem of selecting CHs in WSN. Finally, Section 5.4 provides a summary of the chapter's key points and discussions.

5.2 Whale optimization based NMRA

The newly developed algorithm, WONMRA, represents a hybridization of WOA and NMRA to incorporate the inherent strengths of both algorithms. This novel approach employs self-adaptive parameters, effectively eliminating the requirement for manual parameter tuning by users.

5.2.1 Initialization

The initiation of the WONMRA algorithm involves the random placement of search candidates within a predefined range, and equation (5.1) serves as the general equation for this initial phase.

$$WN_{i,j} = WN_{min,j} + rand(0, 1) \times (WN_{max,j} - WN_{min,j}) \quad (5.1)$$

where $i \in [1, 2, 3, \dots, n]$, n corresponds to number of search candidates (mole-rats), $j \in [1, 2, 3, \dots, d]$, d is treated as the dimension of an optimization problem, $WN_{i,j}$ represents the solution of i^{th} search candidate for j^{th} dimension, $WN_{min,j}$ and $WN_{max,j}$ are the test problem's lower and upper boundaries, and $rand(0, 1)$ is distributed randomly in range $[0, 1]$.

5.2.2 Exploration (worker phase)

The second phase in the WONMRA algorithm is known as the worker phase. In this phase, two randomly selected solutions from the search pool are used to seek a near-optimal solution. This phase is primarily focused on exploration. After a thorough evaluation, it was determined that the worker phase of NMRA is less reliable and requires enhancement to improve its operational characteristics. Consequently, to enhance the worker phase of NMRA, the intrinsic features of WOA

were integrated. In the proposed algorithm, WONMRA, generalized equations such as (2.14), (2.15), (2.16), and (2.17) have been introduced into the worker phase. The introduction of these generalized equations is aimed at preserving the algorithm's original structure. These new equations are executed in the same manner as they would be in a standard WOA. In WONMRA, the worker phase equations of NMRA are executed during the first half of the iterations, while the equations from WOA are executed during the remaining iterations. It is important to note that the parameters utilized in WONMRA remain consistent with the basic algorithms (WOA and NMRA), and no additional parameters have been introduced.

5.2.3 Exploitation (breeder phase)

In the proposed algorithm, WONMRA, the implementation of this phase remains consistent with the original structure of NMRA. In this phase, breeder rats are limited in number and must mate with the queen, representing the global solution. Consequently, the breeder phase aligns with the exploitation operation. The primary rationale for incorporating exploitation in the global search phase is that it seeks a solution close to the current best solution, expecting to produce a global solution as the iterations progress. It is worth noting that the breeder phase in the WONMRA is same as NMRA, and no modifications have been introduced to this phase.

5.2.4 Parameter adaptation:

The mating factor, denoted as (λ) , which is a crucial component of the classical NMRA, holds substantial importance in WONMRA. In basic NMRA, this parameter is determined by simple random values, but in WONMRA, it has been refined to yield improved results. The objective is to adapt this parameter to eliminate the need for user-level adjustments. In WONMRA, the parameter λ is managed through a simulated annealing (*sa*)-based mutation approach, providing superior randomization outcomes. The generalized equation employed to implement this *sa*-based mutation operation is as follows:

$$\alpha_k = \alpha_{min} + (\alpha_{max} - \alpha_{min}) \times b^{(k-1)} \quad (5.2)$$

where, the value of parameter b is fixed at 0.95. The values of α_{max} , α_{min} , and k are randomly generated within the range $[0, 1]$.

5.2.5 Greedy Selection

The selection phase serves as the concluding step in WONMRA. This study adopts a greedy selection approach, where the newly generated solution is treated as the current local best solution. It is then replaced by a new solution if the newly generated solutions are superior to those from the previous generation. The selection strategy for a generalized minimization process, involving the fitness function denoted as $f(WN_i^t)$ for the solution WN_i^t , is outlined in equation (5.3).

$$WN_{new}^{t+1} = \begin{cases} WN_{new} & \text{if } f(WN_{new}) < f(WN_i^t) \\ WN_i^t & \text{otherwise} \end{cases} \quad (5.3)$$

All of the above-mentioned formulations are merged to produce a new WONMRA algorithm. The suggested algorithm tries to keep all major features of both algorithms (WOA and NMRA) and also attempts to give improved end outcomes without compromising the core features of WOA and NMRA. A generalized pseudo-code for the proposed WONMRA is presented in Algorithm 6.

5.3 Results and discussion

This section comprehensively assesses the performance of WONMRA in comparison to other established hybrid optimization algorithms. The evaluation involves conducting various experimental trials and applying statistical tests to both numerical benchmark optimization problems and practical optimization challenges related to CH selection in WSNs. This section is organized into two primary subsections. The first subsection focuses on the performance evaluation of WONMRA concerning the 100-digit challenge presented by the CEC 2019 benchmark problems. In contrast, the second subsection addresses the utilization of WONMRA to address the problem of CH selection within WSNs. Here, all the results are simulated with Ryzen 5 processor with 8GB of RAM and MATLAB version 2016a running on a 64-bit Windows 11 operating system.

5.3.1 Statistical results for CEC 2019 test suite

In this subsection, we evaluate the performance of WONMRA using a set of 10 numerical benchmark problems known as the 100-digit challenge test problems from CEC 2019 [9]. The characteristics of these test problems, including dimension size, search ranges, optimal values, and explanations, can be found in Table A.2. To assess WONMRA's effectiveness, we compare its performance with that of several state-of-the-art optimization algorithms, including ASSA [272], CV 1.0 [82], DA [229], GWO [23], NBA [273], WOA [25], OEWOA [207], and basic NMRA [4]. It's important to note that the

Algorithm 6: Pseudo-code of proposed WONMRA

Start:**Inputs:** Define the random population of NMRs: (n)Decide count of breeder mole-rats(B) = $n/5$ Decide count of worker rats (W) = $n - B$ Initialize breeding probability value (bp)Define problem's dimension (d)**Output:** Evaluate best search candidate (N_{best})**while** $t \leq$ maximum iteration count (t_{max}) **for** $i = 1:W$ **if** Current iteration $\leq t_{max}/2$

execution of NMRA worker phase using equation (2.3)

else

execution of WOA equations by (2.14), (2.15), (2.16) and (2.17)

end if **end for** **for** $i = 1:B$ **if** $\text{rand}(0,1) > bp$

execution of breeder phase using equation (2.4)

end if **end for**

perform greedy selection by equation (5.3)

 update λ using sa mutation operator unite new mole rats (W & B) update (N_{best}) increment t **end while** save N_{best} **Stop**

parameters used for these optimization algorithms remain consistent with those specified in their respective original articles. In our analysis, we conduct 50 runs of each algorithm, consisting of 500 iterations and a population size of 50. The results are presented in terms of various performance metrics, including the best, median, worst, mean, and *Std* values.

The simulated results for these numerical test problems are presented in Table 5.1. The table shows that, except for the standard deviation *Std* values for *CEC* – 01, WONMRA beats other competitor algorithms in various statistical metrics. For *CEC* – 02, where most algorithms produce similar results, WONMRA is the best when considering *Std* values. In the context of *CEC* – 03, where direct comparisons are possible primarily in terms of *Std* values, WONMRA demonstrates the lowest *Std* values compared to other tested algorithms. For *CEC* – 04, WONMRA achieves the best outcomes in terms of mean and *Std* values. In the cases of *CEC* – 05 and *CEC* – 08, OEWOA and GWO outperform other algorithms. For *CEC*₀₆, *CEC*₀₉, and *CEC*₁₀, WONMRA offers solutions that closely approach the optimal values of these test problems. Finally, in the case of *CEC* – 07,

classical WOA exhibits the most efficient performance in terms of mean values. Consequently, out of the total of 10 numerical test problems, WONMRA proves to be efficient for 7 of them, while GWO, OEWOA, and WOA excel in 1 problem each. Further, the working efficiency of WONMRA is also validated by two statistical tests, namely p-rank test [231] and f-rank test [260]. For calculating p-rank, analysis is done in terms of $w(win)/l(loss)/t(tie)$, and the algorithm WONMRA uses NA (Not Applicable). According to the last two rows of Table 5.1, GWO, OEWOA, and WOA win the proposed WONMRA for 2 problems, NMRA, ASSA and DA for 1 function, and CV 1.0 and NBA for none of the functions. After that, the f-rank is determined, and it is discovered that the proposed WONMRA scores first place among all the algorithms under consideration.

Apart from statistical results, convergence graphs and box-plots of WOA, NMRA and WONMRA are plotted for these numerical test problems. In most cases, convergence graphs demonstrate that WONMRA converges to the optimum solution faster than NMRA and WOA, proving its superior performance over both classical algorithms (WOA and NMRA) as shown in Figure 5.1. Further, box-plots were used to evaluate the efficiency of WONMRA in terms of fitness values and due to lower median value of fitness function and consistency in solving these benchmark problems, the WONMRA is proved to be cost-effective, as illustrated in box-plots of Figure 5.2.

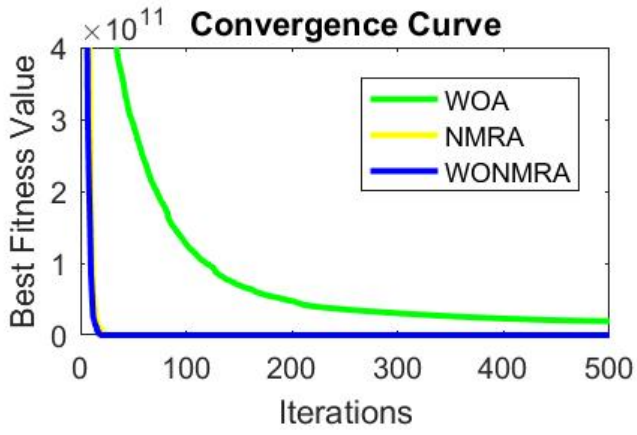
5.3.2 Real world optimization problem: Selection of cluster head in WSN

WSNs comprise numerous sensor nodes that can detect, recognize, and communicate various physical phenomena and processes to a central BS, as discussed in [274]. These networks have found applications in a wide range of fields, including network traffic analysis [275] and multi-path scheduling [276]. In the current context, WSNs are considered groundbreaking tools for data collection due to their ease of deployment and the flexibility of their devices, which surpass traditional wired solutions [232]. WSNs consist of nodes equipped with sensing capabilities, which are interconnected to transmit data to a central sink collectively. This amalgamation of sensing, central processing, and radio communication in sensor networks enables many applications [233] [277]. These applications range from establishing smart road network infrastructure to monitoring sewage systems to prevent blockages and assisting individuals in finding the nearest parking spot in an unfamiliar city.

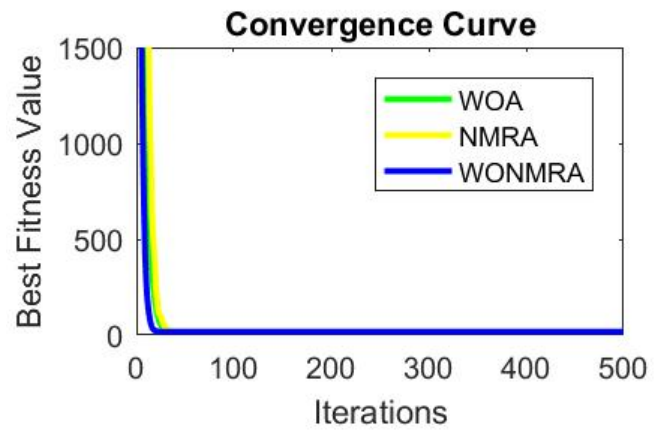
WSNs adhere to the communication architectural model of the OSI framework. However, the network topology in WSNs undergoes frequent changes due to the dynamic nature of nodes simultaneously entering and leaving the network. Energy or power is a critical resource required for sensors to perform various operations, even when they are in an idle state, as discussed in [238]. Consequently, energy sources must be periodically recharged or replaced. However, there are situations where recharging

Table 5.1: Simulated outcomes for CEC 2019 test suite

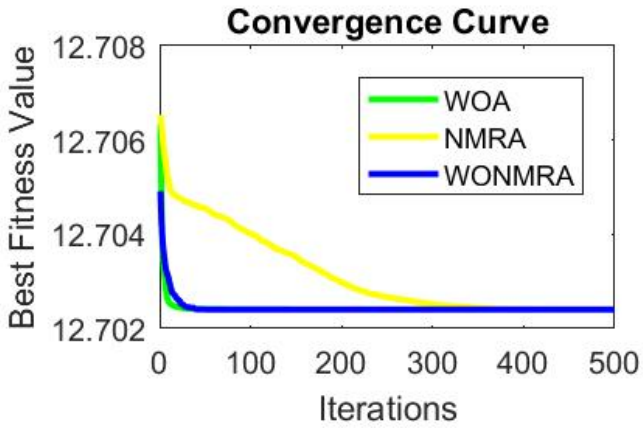
Function		ASSA	CV 1.0	DA	GWO	NBA	WOA	NMRA	OEWOA	WONMRA
CEC_{01}	Best	7.551E+04	1.000E+10	8.080E+08	2.369E+06	4.791E+11	1.021E+06	6.380E+04	4.545E+05	3.971E+04
	Median	2.741E+05	1.000E+10	3.702E+10	2.421E+08	5.102E+12	1.131E+10	2.731E+05	1.680E+07	5.279E+04
	Worst	1.242E+06	1.000E+10	2.722E+11	2.801E+09	1.710E+13	1.400E+11	9.619E+05	1.840E+08	8.858E+05
	Mean	5.568E+05	1.000E+10	6.376E+10	5.771E+08	6.081E+12	1.947E+10	3.101E+05	3.143E+07	1.418E+05
	Std	4.131E+05	0	7.276E+10	7.412E+08	4.456E+12	2.732E+10	2.301E+05	4.198E+07	1.921E+05
	p-rank	—	—	—	—	—	—	—	—	—
CEC_{02}	f-rank	3	6	8	5	9	7	2	4	1
	Best	1.751E+01	7.358E+03	1.731E+01	1.731E+01	8.112E+03	1.731E+01	1.750E+01	1.731E+01	1.731E+01
	Median	1.768E+01	1.611E+04	1.766E+01	1.731E+01	1.502E+04	1.731E+01	1.83E+01	1.731E+01	1.731E+01
	Worst	1.901E+01	2.442E+04	3.120E+02	1.731E+01	2.400E+04	1.742E+01	1.965E+01	1.737E+01	1.731E+01
	Mean	1.788E+01	1.611E+04	7.142E+01	1.731E+01	1.510E+04	1.731E+01	1.842E+01	1.731E+01	1.731E+01
	Std	4.101E-01	4.579E+03	9.132E+01	4.304E-04	3.949E+03	4.400E-03	5.546E-01	3.842E-03	4.512E-06
CEC_{03}	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	5	9	7	2	8	4	6	3	1
	Best	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01
	Median	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01
	Worst	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01
	Mean	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01
CEC_{04}	Std	3.142E-05	9.190E-04	4.876E-04	2.418E-06	1.101E-03	3.079E-07	3.812E-07	3.832E-12	6.691E-15
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	6	8	7	5	9	3	4	2	1
	Best	7.151E+02	9.767E+03	6.210E+01	3.021E+01	1.530E+04	7.487E+01	6.401E+01	4.978E+01	7.959E+01
	Median	2.451E+03	2.820E+04	3.691E+02	6.511E+01	3.268E+04	2.521E+02	2.520E+02	2.636E+02	2.578E+02
	Worst	4.648E+03	5.431E+04	4.072E+03	2.430E+03	6.471E+04	6.381E+02	1.540E+03	8.357E+02	9.000E+02
CEC_{05}	Mean	2.641E+03	2.966E+04	7.269E+02	6.192E+02	3.440E+04	2.821E+02	3.641E+02	3.107E+02	2.730E+02
	Std	1.071E+03	9.601E+03	9.667E+02	6.921E+02	1.100E+04	1.432E+02	3.201E+02	1.861E+02	1.379E+02
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	7	8	6	5	9	2	4	3	1
	Best	2.061E+00	3.532E+00	1.080E+00	1.112E+00	4.287E+00	1.112E+00	1.256E+00	1.051E+00	1.148E+00
	Median	2.471E+00	7.968E+00	1.587E+00	1.521E+00	7.840E+00	1.521E+00	1.882E+00	1.210E+00	2.446E+00
CEC_{06}	Worst	3.091E+00	1.210E+01	2.600E+00	1.800E+00	1.211E+01	2.689E+00	2.841E+00	8.357E+02	9.000E+02
	Mean	2.486E+00	7.988E+00	1.931E+00	1.470E+00	7.900E+00	1.642E+00	1.931E+00	1.252E+00	2.521E+00
	Std	2.911E-01	1.712E+00	3.741E-01	2.969E-01	1.731E+00	3.732E-01	3.841E-01	1.432E-01	7.352E-01
	p-rank	+	—	+	+	—	+	+	+	+
	f-rank	6	9	4	2	8	3	4	1	7
	Best	9.651E+00	1.132E+01	7.021E+00	9.649E+00	1.341E+01	5.852E+00	9.261E+00	5.483E+00	4.151E+00
CEC_{07}	Median	1.162E+01	1.569E+01	9.821E+00	1.122E+01	1.565E+01	8.901E+00	1.132E+01	8.443E+00	7.121E+00
	Worst	1.300E+01	1.751E+01	1.301E+01	1.242E+01	1.741E+01	1.212E+01	1.221E+01	1.199E+01	9.788E+00
	Mean	1.141E+01	1.559E+01	9.868E+00	1.110E+01	1.561E+01	8.832E+00	1.121E+01	8.559E+00	6.921E+00
	Std	9.152E-01	1.221E+00	1.560E+00	7.721E-01	1.050E+00	1.331E+00	6.668E-01	6.037E+00	1.261E+00
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	7	9	4	5	8	3	6	2	1
CEC_{08}	Best	3.848E+02	1.141E+03	-9.587E+01	1.348E+01	1.557E+03	2.441E+01	-2.380E+02	-9.013E+01	-5.131E+01
	Median	9.192E+02	2.121E+03	4.700E+02	4.071E+02	2.311E+03	4.332E+02	1.479E+02	2.780E+02	3.311E+02
	Worst	1.457E+03	2.641E+02	1.100E+03	1.551E+03	2.642E+03	1.181E+03	4.751E+02	1.036E+03	8.212E+02
	Mean	9.071E+02	2.072E+03	4.841E+02	5.466E+02	2.231E+03	1.478E+02	4.651E+02	2.743E+02	3.268E+02
	Std	2.351E+02	4.132E+02	2.900E+02	3.741E+02	3.442E+02	2.191E+02	2.432E+02	2.260E+02	2.121E+02
	p-rank	—	—	—	—	—	+	—	+	+
CEC_{09}	f-rank	7	8	5	6	9	1	4	2	3
	Best	4.668E+00	6.741E+00	4.110E+00	2.451E+00	7.220E+00	4.088E+00	4.948E+00	4.463E+00	3.811E+00
	Median	6.332E+00	8.449E+00	5.868E+00	5.385E+00	8.268E+00	5.679E+00	6.021E+00	6.430E+00	5.675E+00
	Worst	6.900E+00	9.091E+00	6.802E+00	7.011E+00	9.681E+00	7.152E+00	6.551E+00	7.075E+00	6.487E+00
	Mean	6.141E+00	8.322E+00	5.800E+00	5.186E+00	8.400E+00	5.731E+00	5.932E+00	6.249E+00	5.600E+00
	Std	5.841E-01	5.812E-01	6.400E-01	1.041E+00	5.600E-01	5.968E-01	5.077E-01	7.568E-01	3.186E-01
CEC_{10}	p-rank	—	—	+	+	—	—	—	—	—
	f-rank	6	8	4	1	9	3	5	7	2
	Best	1.321E+02	3.269E+03	3.600E+00	3.321E+00	2.812E+03	2.987E+00	3.121E+00	2.848E+00	2.758E+00
	Median	3.286E+02	5.900E+03	5.100E+00	5.577E+00	6.400E+03	4.521E+00	4.342E+00	4.432E+00	4.171E+00
	Worst	7.500E+02	1.069E+04	2.748E+02	7.775E+00	9.168E+03	6.161E+00	3.710E+01	6.684E+00	6.231E+00
	Mean	3.686E+02	6.112E+03	1.521E+01	5.689E+00	6.567E+03	4.421E+00	5.181E+00	4.359E+00	4.170E+00
CEC_{11}	Std	1.411E+02	1.952E+03	4.976E+01	9.789E-01	1.586E+03	7.700E-01	4.712E+00	8.710E-01	9.168E-01
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	7	8	6	5	9	2	4	3	1
	Best	2.031E+01	2.072E+01	2.000E+01	2.031E+01	2.049E+01	2.011E+01	2.011E+01	1.999E+01	1.988E+01
	Median	2.049E+01	2.111E+01	2.011E+01	2.051E+01	2.100E+01	2.011E+01	2.041E+01	2.000E+01	2.000E+01
	Worst	2.071E+01	2.149E+01	2.048E+01	2.056E+01	2.141E+01	2.048E+01	2.069E+01	2.038E+01	2.000E+01
CEC_{12}	Mean	2.049E+01	2.111E+01	2.010E+01	2.049E+01	2.100E+01	2.019E+01	2.041E+01	2.004E+01	2.000E+01
	Std	7.451E-02	2.078E-01	1.410E-01	9.451E-02	1.742E-01	1.031E-01	9.989E-02	9.720E-02	5.000E-03
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	7	9	3	6	8	4	5	2	1
$w/l/t$		1/9/0	0/10/0	1/9/0	2/8/0	0/10/0	2/8/0	1/9/0	2/8/0	NA
Mean f-rank value		6.10	8.20	5.40	4.20	8.60	3.20	4.40	2.90	1.90
Overall f-rank		7	8	6	4	9	3	5	2	1



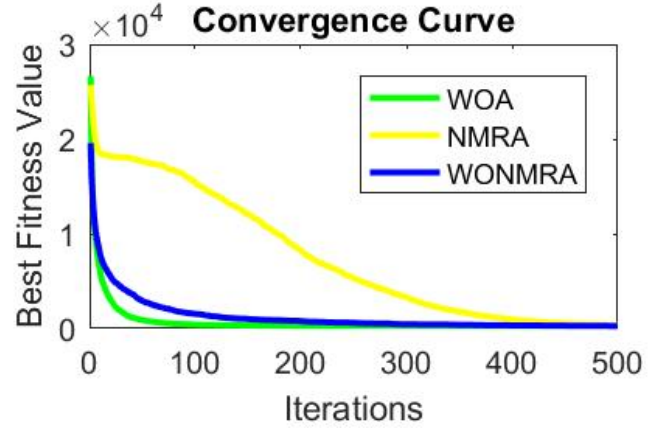
(a) Convergence graph for CEC_{01}



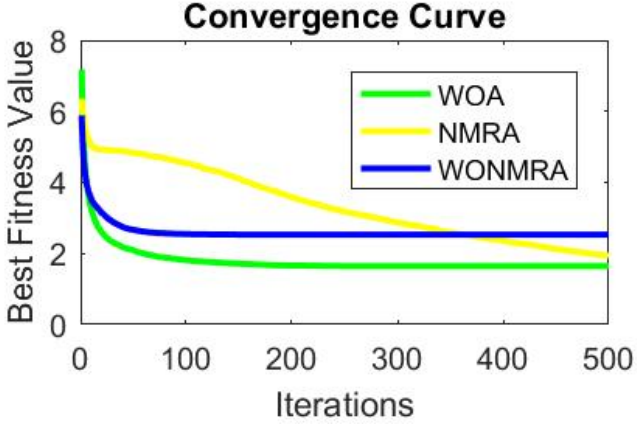
(b) Convergence graph for CEC_{02}



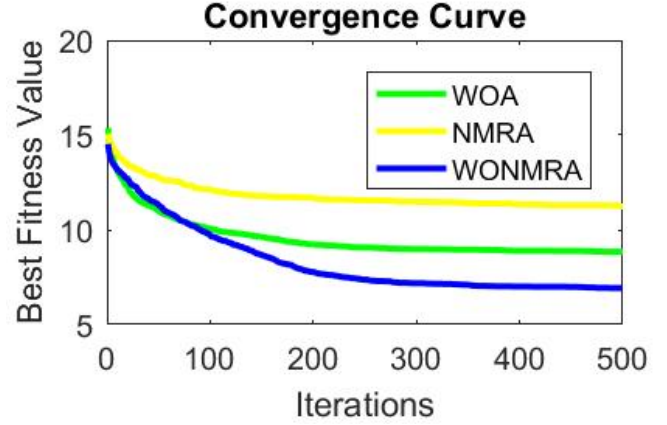
(c) Convergence graph for CEC_{03}



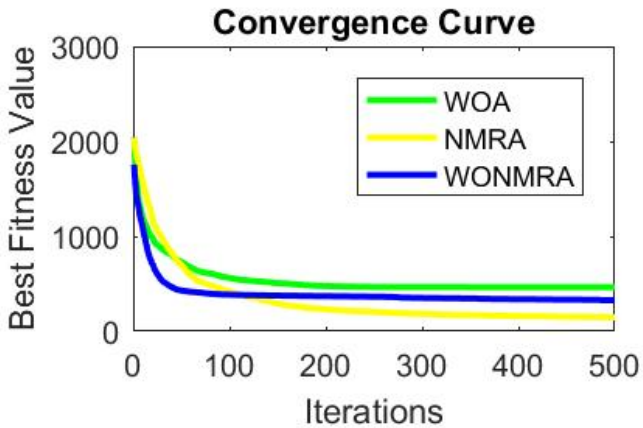
(d) Convergence graph for CEC_{04}



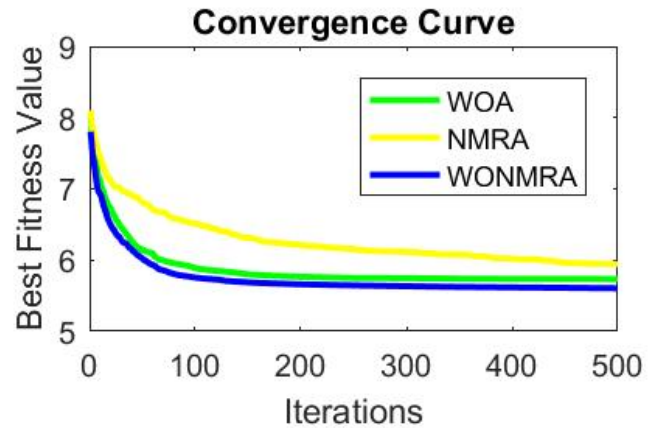
(e) Convergence graph for CEC_{05}



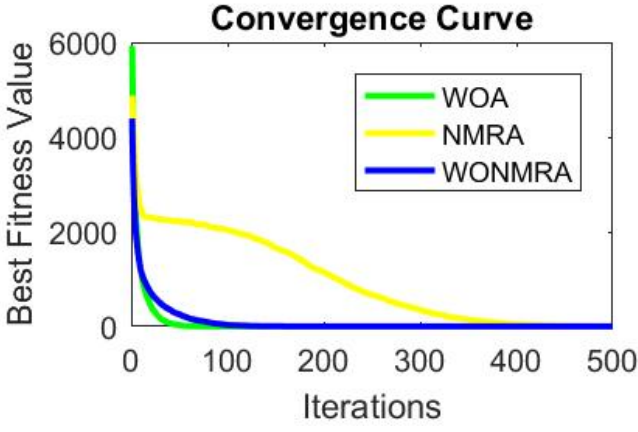
(f) Convergence graph for CEC_{06}



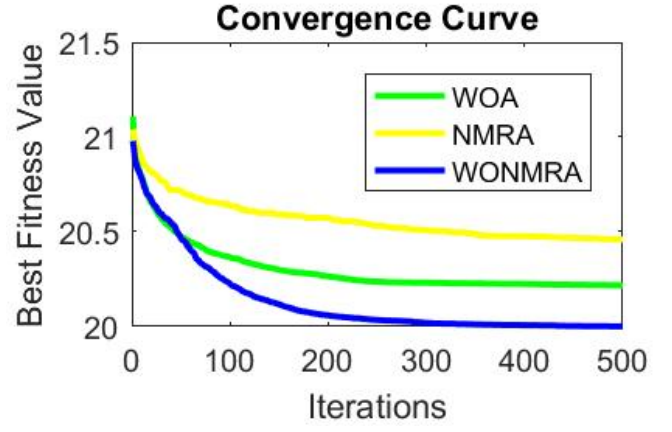
(g) Convergence graph for CEC_{07}



(h) Convergence graph for CEC_{08}



(i) Convergence graph for CEC_{09}



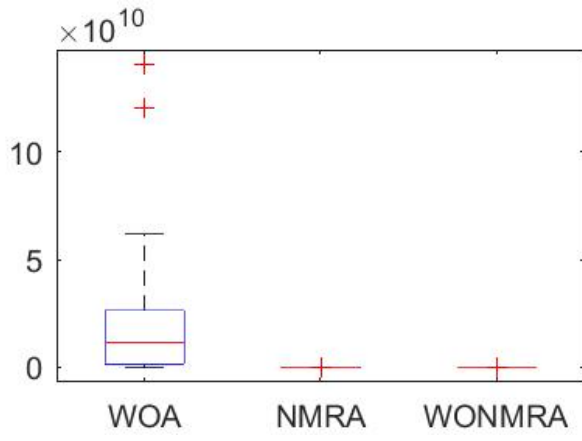
(j) Convergence graph for CEC_{10}

Figure 5.1: Convergence graphs of WOA, NMRA and WONMRA for 100-digit benchmark functions (CEC 2019)

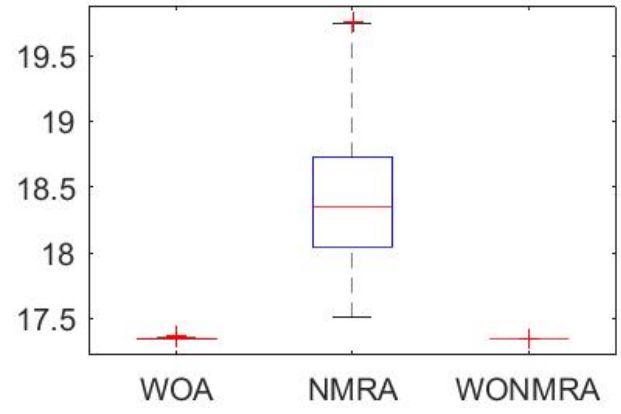
or replacing batteries becomes challenging due to environmental conditions. In such cases, it is essential to have reliable alternatives to address these energy constraints [239]. Furthermore, sensor nodes are characterized by their small size, leading to constraints on various resources, including energy, memory, computational speed, and communication bandwidth. These constraints have a direct impact on the QoS of the network [240]. QoS represents the level of service provided to users, and in the context of WSNs, it encompasses factors such as the network's monitoring and tracking capabilities, the coverage area, data transmission latency, power consumption, and others [241]. The QoS of a WSN is highly application-specific and tailored to the particular requirements of tasks like event monitoring, network coverage, data transmission timeliness, and energy efficiency.

In a cluster-based WSN architecture, sensors are organized into clusters, with one sensor designated CH, while the remaining nodes within each cluster serve as CMs. To facilitate better comprehension, a brief description of each role is provided [240]. Energy-efficient routing strategies within such WSN architectures aim to reduce energy consumption by optimizing various functional aspects, including sink communication, cluster formation, CH selection criteria, CH rotation policies, and strategies for both intra-cluster and inter-cluster communication [241], [242].

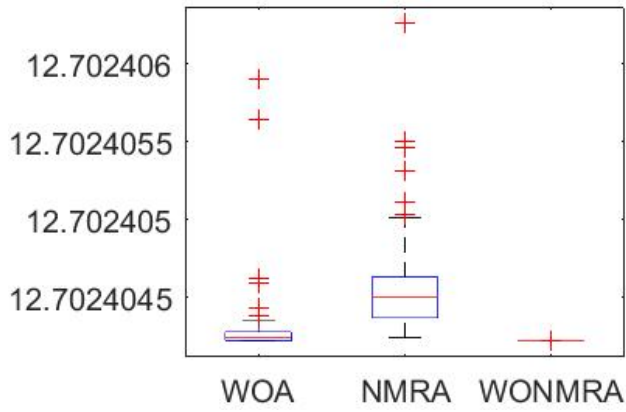
In WSNs, the most challenging aspects of developing and planning WSN operations are reducing energy usage and extending network lifetime [37] [42]. Clustering protocols similar to LEACH [148] are employed to provide extended lifespan and scalable solutions. In such cluster-based routing strategies, the CHs lifetime significantly achieves a longer network lifetime. Some CHs may be overwhelmed due to improper cluster formation. Such an overload could lead the CH to consume much energy and decrease the WSN's overall performance. As a result, the most critical aspect in clustering sensor nodes is effective CH selection. It's not easy to come up with an energy-efficient clustering algorithm.



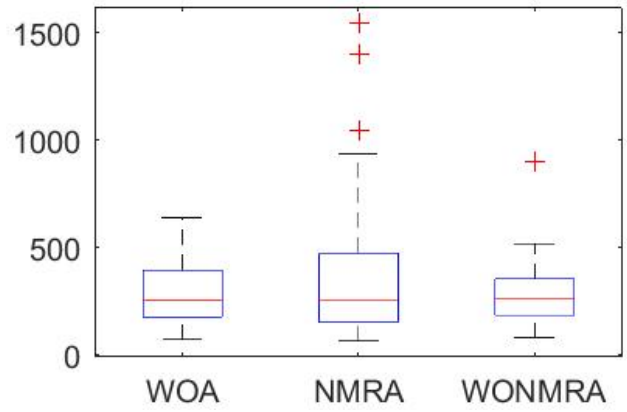
(a) Box-plot for CEC_{01}



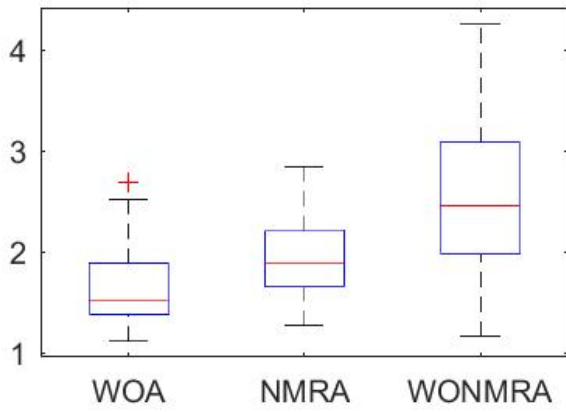
(b) Box-plot for CEC_{02}



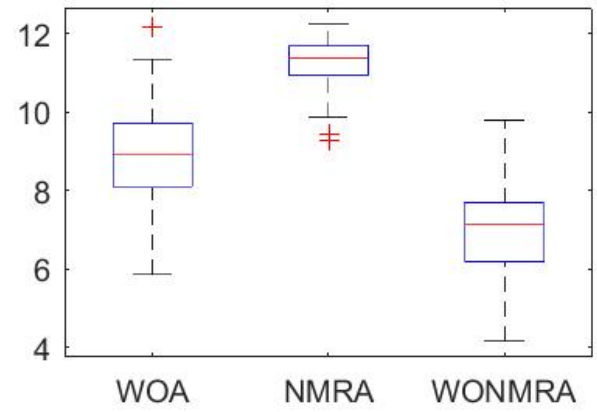
(c) Box-plot for CEC_{03}



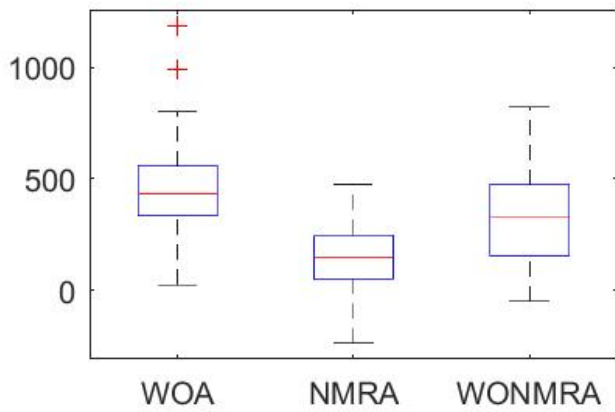
(d) Box-plot for CEC_{04}



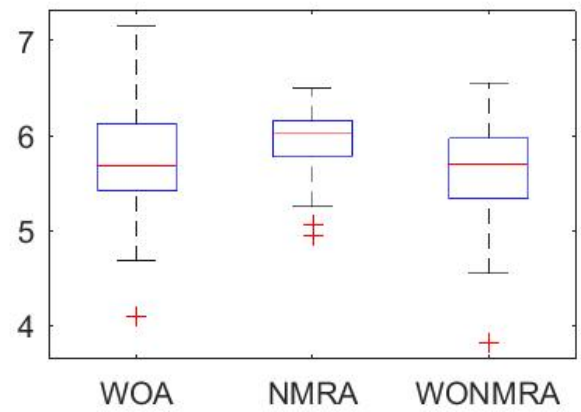
(e) Box-plot for CEC_{05}



(f) Box-plot for CEC_{06}



(g) Box-plot for CEC_{07}



(h) Box-plot for CEC_{08}

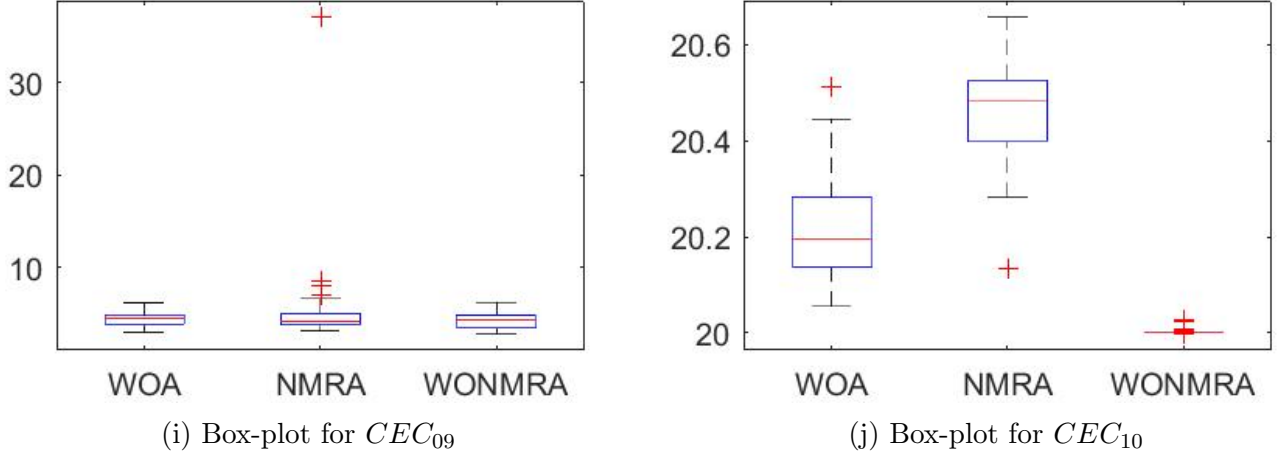


Figure 5.2: Box-plots of WOA, NMRA and WONMRA for 100-digit benchmark functions (CEC 2019)

As a result, nature-inspired optimization solutions could be used to solve this WSN's problem. In recent years, EAs have been employed as meta-heuristics to solve energy-conscious routing problems by creating models that maximize an appropriate energy-aware fitness function [163]. To answer the aforesaid load balancing problem, this research employs a WONMRA-based clustering protocol. The protocol creates clusters, reducing the network's overall energy usage.

Consider a WSN with n sensor nodes that are distributed randomly and arranged into K clusters: $C_1, C_2, \dots, C_K$. The energy consumption-based fitness function could be written as follows:

$$f = \left(\sum_{i=1}^K \sum_{seC_i} \sum_{TX_{s,CH_i}} + E_{RX} + E_{DA} \right) + \sum_{i=1}^K E_{TX_{CH_iBS}} \quad (5.4)$$

where K is the total number of CHs, seC_i is the number of non-CHs associated with the i^{th} CH, and $E_{TX_{node_1, node_2}}$ is the amount of energy used to send data from $node_1$ to $node_2$.

The following formulas are used to compute radio energy communication costs for a k -bit message as a function of distance d :

$$E_{TX} = \begin{cases} kE_{elec} + k\epsilon_{friss_amp}d^2, & \text{if } d < d_0 \\ kE_{elec} + k\epsilon_{two_ray_amp}d^4, & \text{if } d \geq d_0 \end{cases} \quad (5.5)$$

here E_{elec} refers to the energy dissipation per bit during transmission. For free-space and multi-path fading models, the parameters ϵ_{friss_amp} and $\epsilon_{two_ray_amp}$ imply transmitter amplification parameters.

Table 5.2: Comparative analysis of various algorithms for network lifetime estimation in WSN

Algorithm	Optimal cost of FF	FND	HND	LND
GA	39.797 mJ	2047.2	2678.3	3337.4
PSO	39.853 mJ	2034.9	2665.1	3324.7
BA	40.323 mJ	1996.8	2654.9	3298.8
CS	38.411 mJ	2079.3	2702.6	3372.3
GWO	38.559 mJ	2073.8	2696.4	3369.7
mGWO	38.208 mJ	2084.5	2714.5	3382.3
NMRA	38.383 mJ	2079.2	2709.9	3374.6
WONMRA	38.198 mJ	2087.3	2717.6	3387.5

The value of d_0 is calculated as follows:

$$d_0 = \sqrt{\epsilon_{friss_amp} / \epsilon_{two_ray_amp}} \quad (5.6)$$

The following equation can be used to express the reception energy of a k -bit data message:

$$E_{RX} = kE_{elec} \quad (5.7)$$

where E_{elec} is the energy dissipation per bit during receipt.

E_{DA} is the data aggregation energy cost set to $5nj/bit$. The values of the remaining parameters are set to $E_{elec} = 50nj/bit$, $\epsilon_{friss_amp} = 100pj/bit/m^2$ and $\epsilon_{two_ray_amp} = 0.0013pj/bit/m^4$ [243].

In the simulation setup, a total of 100 nodes are randomly distributed across a sensing field measuring $100m \times 100m$. The BS is positioned at the center of this field. All nodes in this homogeneous network initially possess an energy level of $E_0=1J$. To characterize the network's lifetime, three essential parameters are employed: FND (the number of communication rounds before the first node exhausts its energy), HND (the number of communication rounds until half of the nodes have depleted their energy), and LND (the number of communication rounds until the last node in the network runs out of energy). These parameters serve to define the duration of the network's operational lifetime in this analysis [278].

Table 5.2 presents the optimal results achieved by the WSN CH selection problem. The table clearly demonstrates that the WONMRA algorithm consistently outperforms other algorithms in terms of performance. The mGWO, NMRA, CS, and GWO algorithms closely follow the WONMRA results.

5.4 Summary

This chapter presents a new WONMRA algorithm based on the exploitation capabilities of NMRA and the exploration properties of WOA. The algorithm aims to provide better exploration and exploitation operations by using the combined advantages of both algorithms. Also, self-adaptivity

is added by using a simulated annealing-based mutation strategy. The proposed WONMRA, when tested on CEC 2019 benchmark, was found to provide better results than ASSA, CV 1.0, NBA, and other algorithms. The algorithm is also tested on a very challenging problem of clustering in WSNs. The result shows that the proposed algorithm is highly effective for CH selection in the WSN problem because it converges quickly and has fewer chances of getting stuck at local minima. Overall, for all the problems under consideration, the proposed algorithm significantly advances the existing literature and can be considered one of the promising algorithms for future optimization research.

Chapter 6

An Energy-efficient Clustering Protocol using Novel Self-adaptive Hybrid COOT Optimization based Naked Mole-rat Algorithm for WSN

WSNs are widely spaced nodes used for various purposes, including military defence, healthcare, environmental monitoring, and others. The main problem in this field is designing a communication system that enables sensors to reach their destination with the least energy. Extending the network performance parameters through clustering is a successful strategy. An energy-efficient clustering protocol with the best CH selection algorithm is required to create an effective network structure to address the limitations in current protocols. A novel self-adaptive hybrid COOT optimization naked mole-rat algorithm (CONMRA) based clustering protocol is presented to perform the optimal selection of CH and enhance the network lifespan. Here, the effectiveness of CONMRA is assessed using the numerical test problems from CEC 2019 test suite. The statistical findings using p-rank test and f-rank test demonstrate that, for most numerical test problems, the proposed CONMRA outperforms other conventional optimization techniques. Further, an energy-efficient stable clustering protocol using CONMRA for both homogeneous and heterogeneous network setups performs better with respect to other clustering protocols in terms of increased network lifetime and decreased energy consumption.

6.1 Introduction

WSNs are becoming more and more common in a variety of industrial applications, such as public safety networks, automobile traffic monitoring, IoT, etc. [249] [250]. WSNs are useful in a variety of fields, including environmental sensing, healthcare, and industrial monitoring [251]. A WSN is made up of dispersed sensor nodes and BS, which interact with the environment by sensing specific physical properties [250]. The onboard batteries of WSN nodes are non-exchangeable, non-rechargeable, and have a finite lifespan. Additionally, sensors have constrained memory, storage, and processing abilities [41]. Here, sensors are modified to fulfill the rising need for sensing data due to the requirement for complex functional features to achieve a high level of automation. The cost of tampering with the sensor network is quite high when sensors are subjected to a variety of physical environments [251]

[41]. Thus, extending the network's life is an essential requirement.

The design of an efficient routing protocol is paramount in preventing energy depletion and prolonging the lifespan of WSNs, especially when SNs and the BS communicate through wireless radio signals, as emphasized in [250]. In WSNs, direct transmission of data to the sink, or BS, tends to consume more energy than breaking the communication into multiple shorter-distance stages. However, in the context of clustering-based approaches, each sensor node transmits its information directly to the CH, which then collects and forwards the incoming data packets to the back-end server or sink, as discussed in [251] and [250].

Clustering protocols in WSNs can be categorized into two main types: centralized and decentralized methods. Centralized approaches rely on global knowledge of the network, while decentralized methods utilize local information to create clusters. The primary objective of WSNs is to develop energy-efficient routing protocols that facilitate efficient communication among SNs and BS. Numerous energy-efficient clustering techniques have been proposed in the literature, as highlighted in [250], to address this challenge. Many of these techniques involve grouping sensors to enable multi-hop transmission to the sink, which, in turn, simplifies processes such as data aggregation, redundant data reduction, data compression, and multi-hop data transfer. The effectiveness of these routing protocols is contingent on the choice of fitness factors used, as discussed in [251]. In the context of optimizing performance metrics for HRPs, NIA are often preferred. This chapter selects CHs through the proposed CONMRA, as mentioned in [6] and [4].

The COOT optimization method draws inspiration from the movement of COOT birds on the water surface as they seek their food (optimal solution). In contrast, the naked NMRA explores potential solutions for a given problem by emulating the mating patterns of breeders and workers of naked mole rats under the queen's guidance. It's important to note that both of these algorithms exhibit efficiency but share inherent limitations, with COOT optimization being criticized for poor exploitation and NMRA for limited exploration. Additionally, these algorithms tend to suffer from premature convergence, often resulting in stagnation around locally optimal solutions. Consequently, there is a recognized need for enhancements to improve their overall performance. A hybrid concept is proposed with a focus on preserving and combining the exploration capabilities of COOT optimization and the exploitation patterns of NMRA. Furthermore, self-adaptive properties are incorporated into the algorithm to make it well-suited for addressing complex real-world optimization problems. In this proposed algorithm, the foundational structure of NMRA is retained, while all the mathematical equations from COOT optimization are integrated into the exploration phase. This strategic integration, where the breeder phase emphasizes exploitation and the worker phase emphasizes ex-

ploration, is expected to yield a superior algorithm with well-balanced exploration and exploitation capabilities.

The residual energy of the sensor nodes, their distance from the base station, and the node's trust value are a few factors that affect how well the WSN clustering method performs. But if the clustering issue is correctly assessed, several elements might be considered. To find the best clustering solution, each parameter must be weighed according to how it will affect energy consumption and network life. Here, a centralized clustering approach is presented for WSNs, where CHs are selected using the proposed CONMRA optimization technique. The CONMRA algorithm is designed to prioritize the selection of secure CHs for SNs by considering input parameters such as remaining energy, confidence value, and distance from the BS. The primary contributions of this chapter can be summarized as follows:

- A novel self-adaptive hybrid COOT optimization-based NMRA named CONMRA is proposed with improved exploration and exploitation properties.
- The working efficiency of CONMRA is tested for CEC 2019 numerical test suite.
- The statistical results of CONMRA have been compared with other state-of-the-art algorithms and validated by two statistical tests, namely p-rank test and f-rank test.
- A clustering protocol for WSN based on is developed using CONMRA for optimal selection of CH.
- To achieve an extended stability period, a clustering solution inspired by the optimization method CONMRA is considered.

The rest of the paper is structured as follows: Section 6.2 deals with the details of CONMRA, providing insights into its structure and mathematical model. In Section 6.3, you will find a presentation of the statistical results obtained through the hybrid self-adaptive CONMRA algorithm for solving the CEC 2019 numerical test problems. Section 6.4 offers an overview of the network model and introduces the proposed clustering protocol that leverages CONMRA. In Section 6.5, you will discover the simulated results of various clustering protocols designed for WSNs. Finally, Section 6.6 wraps up the chapter with a comprehensive summary of its key points and findings.

6.2 Hybrid COOT optimization based NMRA

The recently proposed two swarm intelligent optimization algorithms (NMRA and COOT optimization) are very promising and reliably solve real-world optimization problems. However, compared

to advanced adaptive and hybrid algorithms, these optimization methods' performance is less reliable. This is mainly due to NMRA has poor exploration capabilities, which keeps it stuck in a local optimal solution and provides less accurate convergence to an optimal solution. To solve this issue, COOT optimization-based equations are added to the worker phase (exploration phase) of NMRA, enhancing the algorithm's capability for global searching. Apart from the modification in the worker phase, the working ability of the breeder phase is also enhanced using attraction and repulsion strategy [279]. The fundamental NMRA's mating factor parameter λ is crucial in deciding how well it performs and must be investigated for a self-sufficient or self-adaptive algorithm. These various modifications are taken into account, and a new optimization technique CONMRA is developed with the following key features:

- The exploration capability of NMRA is enhanced by incorporating COOT optimization equations in the worker phase.
- The worker phase is implemented by employing the classical NMRA equations for the initial half of the iterations, followed by using COOT optimization equations for the second half.
- The breeder phase is modified with the help of attraction and repulsion strategy based on global best and worst solutions [279].
- To develop the algorithm with self-adaptive capabilities and remove the requirement for user-driven parameter adjustments, the parameter λ from the fundamental NMRA algorithm is incorporated, utilizing a mutation operator based on simulated annealing (*sa*), as detailed in [271].

The mathematical model employed to develop CONMRA can be divided into three distinct phases: random initialization of search agents, exploration, and exploitation.

6.2.1 Initialization of search agents

In the CONMRA algorithm, search candidates are initially randomized within predefined bounds, and the general equation governing this phase is as follows:

$$CN_{p,q} = CN_{min,q} + rand[0, 1] \times (CN_{max,q} - CN_{min,q}) \quad (6.1)$$

where $p \in [1, 2, \dots, CN_n]$, CN_n is the count of search agents, $q \in [1, 2, \dots, dim]$. The test problem's lower and upper limits are $CN_{min,q}$ and $CN_{max,q}$, respectively, and its dimension is dim . Additionally, a random number between 0 & 1 is $rand[0, 1]$.

6.2.2 Exploration phase

The exploration phase, also known as the worker phase, in the proposed hybrid algorithm CONMRA combines elements from classical NMRA and COOT optimization techniques. The worker phase of NMRA was initially considered less reliable, prompting efforts to enhance its performance. Significant improvements have been made to the worker phase of NMRA within the CONMRA framework by incorporating the inherent attributes of COOT optimization. To elaborate further, this phase in CONMRA is divided into two segments based on the number of iterations.

a) Phase I: For $t \leq t_{max}/2$: During the initial half of the iterations, there is a need for maximum exploration. The basic NMRA equation is employed to fulfill this requirement, involving two random solutions and a self-adaptive parameter λ . Despite the possibility that the random solutions may occasionally move near each other, the introduction of a larger step size through λ imparts a more exploratory character to the process. This, in turn, facilitates comprehensive exploration of the solution space. The generation of a new solution for the search agent is as follows:

$$CN_p^{t+1} = CN_p^t + \lambda(CN_g^t - CN_h^t) \quad (6.2)$$

where CN_p^{t+1} denoted the search agent's new solution, CN_p^t indicates the solution of the p^{th} search agent for the t^{th} iteration, CN_g^t and CN_h^t are randomly chosen solution from worker rats population. The new adaptive parameter λ is evaluated using equation (6.3) as:

$$\lambda = \lambda_{min} + (\lambda_{max} - \lambda_{min}) \times b^{(c-1)} \quad (6.3)$$

where b is set to 0.95, the values of λ_{min} and λ_{max} are considered as 0.5 and 0.9 respectively, and c is defined as random value for the range between 0 and 1.

b) Phase II: For $t > t_{max}/2$: In this stage, exploitation begins to increase while exploration starts to decrease. For these iterations, the new solution of the search agent is based on COOT optimization's leader equations and implemented as:

$$CN_p^{t+1} = \begin{cases} \gamma \times r_3 \times \cos(2\pi r_d) \times (CN_{best} - CN_p^t) + CN_{best} & r_4 < 0.5 \\ \gamma \times r_3 \times \cos(2\pi r_d) \times (CN_{best} - CN_p^t) - CN_{best} & r_4 \geq 0.5 \end{cases} \quad (6.4)$$

where CN_{best} represents the best solution discovered among the population of search agents. The variables r_d and r_3 are randomly generated within the range of $[-1, 1]$, while r_4 is a random number

within the range of $[0, 1]$. The value of γ is computed using the formula presented in equation (6.5).

$$\gamma = 2 - t \times \left(\frac{1}{t_{max}} \right) \quad (6.5)$$

6.2.3 Exploitation phase

This phase of the proposed algorithm CONMRA is implemented using attraction and repulsion strategy by utilizing both global worst and global best solutions [279]. This approach enables the search agents, represented as breeder rats, to move freely, influenced by both attraction and repulsion forces, ultimately guiding them toward an optimal solution. At this stage, establishing the global worst solution can aid in boosting the population's diversity. The issue of premature convergence is resolved while local search is expanded. The search agents can obtain their optimal solution by using the following search equation, which is based on the attraction-repulsion strategy:

$$CN_p^{t+1} = (1 - \lambda) CN_p^t + \lambda [a_1(CN_{best} - CN_p^t) - a_2(CN_{worst} - CN_p^t)] \quad (6.6)$$

where value of parameter a_1 is 0.5, a_2 is 0.4, CN_{best} represents globally best solution and CN_{worst} represents globally worst solution.

6.2.4 Greedy selection

The selection phase is recognized as the most pivotal aspect of CONMRA. This optimization method utilizes a greedy selection approach, where a newly generated solution is considered the current local best solution and is adopted if it surpasses the performance of the solution from the previous generation. The selection process for a generalized minimization procedure, governed by the fitness function $f(CN_p^t)$ for the solution CN_p^t , is defined by equation (6.7).

$$CN_{new}^{t+1} = \begin{cases} CN_{new} & \text{if } f(CN_{new}) < f(CN_p^t) \\ CN_p^t & \text{otherwise} \end{cases} \quad (6.7)$$

The flow chart of the proposed CONMRA is presented in Figure 6.1.

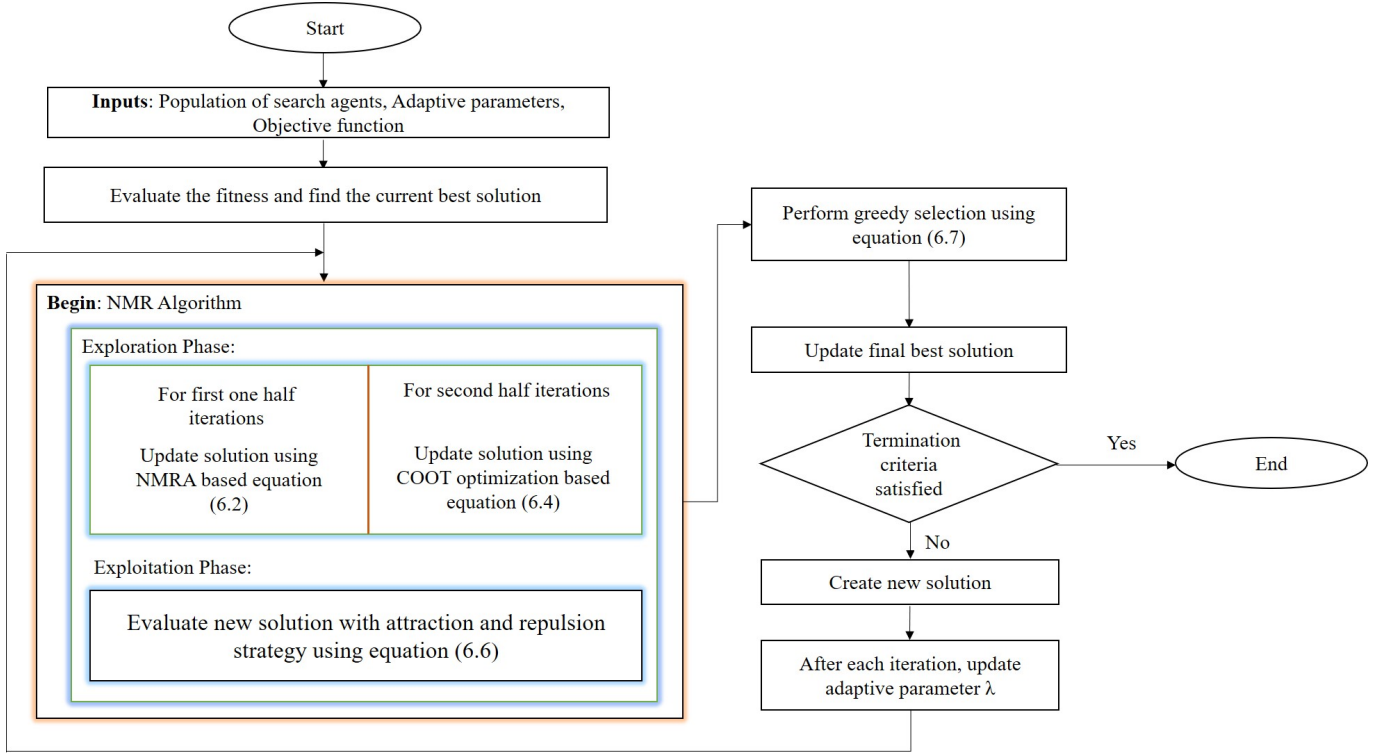


Figure 6.1: Flow chart of proposed CONMRA

6.2.5 Computational complexity

Every algorithm's computational complexity study is carried out to determine its worst-case complexities and to examine its generalized operations. Firstly, the complexity analysis of the fundamental COOT optimization and NMRA has been discussed before investigating the proposed CONMRA's complexity analysis. The computational complexity for COOT optimization depends on the CN_n population members that are present in the dim dimensional search space and runs for t_{max} iterations at most. Hence, it may be claimed that $O(CN_n \times dim \times t_{max})$ is the formula for COOT's optimization overall run-time complexity. Similar circumstances apply to NMRA and its associated run-time complexity. This similarity arises because both algorithms produce comparable overall results, and there are no additional parameters that could influence their performance. The overall estimation for these algorithms is based on the need to perform $O(dim)$ operations for each member of the population, where dim represents the dimension size defining the computational cost for each population member. Consequently, the complexity is given by $O(CN_n \times dim)$ for the entire population size. This complexity is multiplied by the factor of the total number of iterations reaching t_{max} , as observed in this complexity analysis for a single iteration or generation. Consequently, each algorithm exhibits a total complexity of $O(CN_n \times dim \times t_{max})$. Now, when calculating the computational cost of the proposed CONMRA optimization algorithm compared to COOT optimization and NMRA, it becomes evident that the fundamental structure of both algorithms remains unchanged. Therefore, no additional computational burden is introduced due to hybridization. Consequently, CONMRA's

computational complexity remains identical to that of the original COOT optimization and NMRA algorithms.

6.3 Statistical testing of CONMRA for CEC 2019 benchmark numerical test problems

This section evaluates CONMRA's performance using the 10 CEC 2019 numerical test problems [9]. Details regarding the dimension size, search range, optimal values, and descriptions for all these test problems can be found in Table A.2. We compare the proposed CONMRA with other optimization techniques such as GWO [23], ASSA [272], DA [229], NBA [273], CV 1.0 [82], DE [12], COOT algorithm [6] and NMRA [4]. All of these utilize the same parameters as specified in the original articles. We provide the median, best, mean, worst, and *Std* values for each optimization algorithm based on 30 simulation runs, each comprising 500 iterations and a population size of 30.

The simulated outcomes for these numerical test problems are presented in Table 6.1. This table shows that, with the exception of *Std* values for TP_{01} , CONMRA outperforms other competitive algorithms in all statistical terms. In the case of TP_{02} , the majority of the algorithms being tested yield results that are comparable and a comparison of *Std* values shows that CONMRA is the best. For test problem TP_{03} , *Std* values can be used to compare the results and it was found that CONMRA has the lowest standard deviation of all the algorithms that were put to the test. In case of test problems TP_{04} , TP_{06} , TP_{07} and TP_{09} , CONMRA provides results that are close to the test problem's optimal value. COOT produces the best results for TP_{05} and TP_{10} problems. The working effectiveness of COOT algorithm is best for test problem TP_{08} . Thus, out of 10 numerical test problems, CONMRA is effective for seven, GWO for one, and COOT for two problems.

In addition, the rank-sum (p-rank) test from [231] and the Friedman (f-rank) test from [260] statistical support the effectiveness of CONMRA. For computing p-rank, analysis is done in terms of $w(win)/l(loss)/t(tie)$, and the proposed algorithm CONMRA employs NA (Not Applicable). A $win(w)$ case is one in which the contrasted optimization technique outperforms CONMRA and receives a "+" indication. When the test algorithm efficiency is lower than CONMRA, the $loos(l)$ condition is met, and the "-" sign is used in its stead. The "=" indicator indicates that the compared algorithms are equally capable of solving the objective function in the case of $tie(t)$. The $w/l/t$ row of Table 6.1 demonstrates that the proposed optimization method CONMRA performs noticeably better in the majority of cases. The f-rank statistical test also assigns an f-rank to every optimization technique that is being tested.

Determining whether the proposed CONMRA outperforms other optimization strategies using the

f-rank is straightforward. In the last two rows of Table 6.1, the overall f-rank and average f-rank value are calculated after assigning the f-rank to each algorithm. Consequently, it can be deduced from these two rows that CONMRA emerged as the most effective approach for these numerical test problems and is statistically significant as an optimization strategy.

In addition to the statistical data, CONMRA, COOT, and NMRA convergence profiles are plotted for these numerical test problems, as illustrated in Figure 6.2. The convergence profiles typically show that CONMRA converges to the best solution more quickly than NMRA and COOT, demonstrating its superior performance to both traditional techniques (COOT and NMRA) in most cases.

6.4 Proposed clustering protocol using CONMRA for WSN

In the hierarchical architecture of HRP, the bottom level involves cluster formation and CH selection, while routing occurs at the top level. This study uses CONMRA to assess the fitness value for SNs to perform optimal CH selection. The energy-efficient routing protocol based on CONMRA (CNSTERP) consists of two communication phases: setup and steady-state. While the CH selection method in the setup phase has been modified, the steady-state phase remains the same as in DRESEP [243]. During the setup phase, the BS utilizes CONMRA to create energy-efficient clusters for a specified number of operational sensor nodes and network residual energy. It is essential to highlight that CH selection in WSNs is a binary-coded problem, which the continuous CONMRA does not address proposed earlier. To address this, a binary version of the proposed CONMRA is considered, and the following equation is employed:

$$Flag_{p,q} = \begin{cases} 1 & \text{if } CN_{p,q} \geq 0.5, \\ 0 & \text{otherwise} \end{cases} \quad (6.8)$$

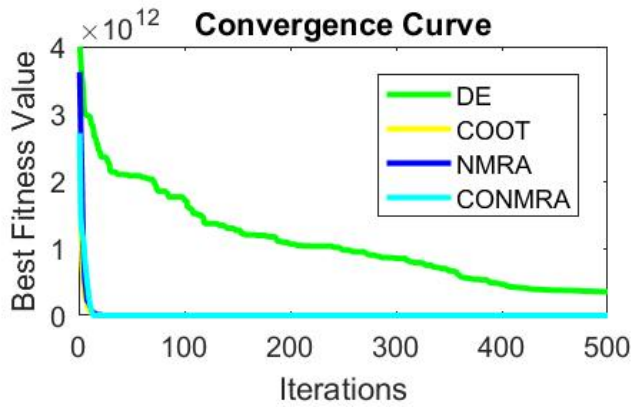
where $CN_{p,q}$ represents the q^{th} solution of the p^{th} search agent of proposed CONMRA.

6.4.1 Cluster head election

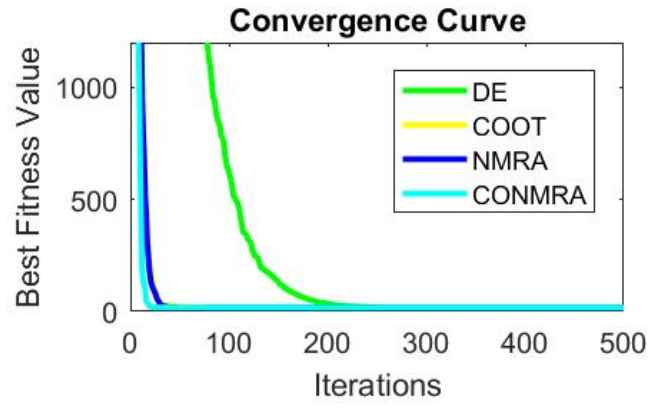
In this reactive protocol, SNs and BS communicate with each other for network control during the election phase. Figure 6.3 illustrates how CH nodes are selected using CONMRA based on data from sensor nodes such as initial energy, remaining energy of nodes, the distance of a node from BS etc. A thorough solution is used to assign one and zero, respectively, to the location of CHs and the members of those CHs in the WSN. This phase begins with forming an initial population and then uses the predicted energy consumption given by the suggested fitness criteria to determine

Table 6.1: Simulated outcomes for CEC 2019 test suite

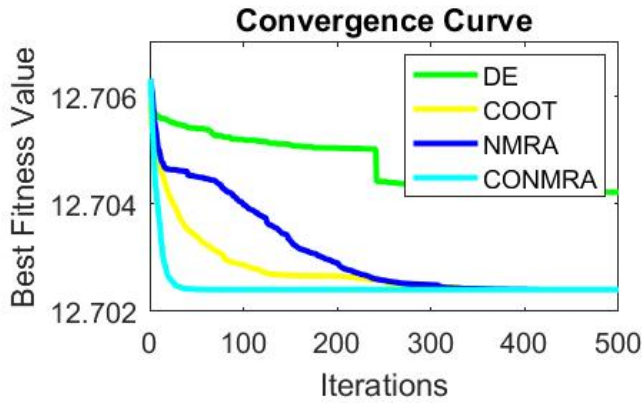
Function		DE	ASSA	CV 1.0	DA	GWO	NBA	COOT	NMRA	CONMRA
TP_{01}	Best	1.805E+10	7.551E+04	1.000E+10	8.080E+08	2.369E+06	4.791E+11	4.211E+04	6.922E+04	3.917E+04
	Median	3.392E+11	2.741E+05	1.000E+10	3.702E+10	2.421E+08	5.102E+12	6.577E+04	4.483E+05	4.583E+04
	Worst	4.117E+12	1.242E+06	1.000E+10	2.722E+11	2.801E+09	1.710E+13	9.762E+05	9.222E+05	6.388E+04
	Mean	5.864E+11	5.568E+05	1.000E+10	6.376E+10	5.771E+08	6.081E+12	2.312E+05	4.356E+05	4.668E+04
	Std	8.359E+11	4.131E+05	0	7.276E+10	7.412E+08	4.456E+12	3.162E+05	2.969E+05	6.777E+03
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	8	4	6	7	5	9	2	3	1
TP_{02}	Best	1.734E+01	1.751E+01	7.358E+03	1.731E+01	1.731E+01	8.112E+03	1.734E+01	1.795E+01	1.734E+01
	Median	1.735E+01	1.768E+01	1.611E+04	1.766E+01	1.731E+01	1.502E+04	1.734E+01	1.878E+01	1.734E+01
	Worst	7.636E+02	1.901E+01	2.442E+04	3.120E+02	1.731E+01	2.400E+04	1.735E+01	1.971E+01	1.734E+01
	Mean	6.844E+01	1.788E+01	1.611E+04	7.142E+01	1.731E+01	1.510E+04	1.734E+01	1.883E+01	1.734E+01
	Std	1.418E+02	4.101E-01	4.579E+03	9.132E+01	4.304E-04	3.949E+03	3.500E-03	6.080E-01	1.274E-04
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	7	4	9	6	2	8	3	5	1
TP_{03}	Best	1.269E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01
	Median	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01
	Worst	1.271E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01
	Mean	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01	1.270E+01
	Std	1.900E-03	3.142E-05	9.190E-04	4.876E-04	2.418E-06	1.101E-03	1.329E-08	4.189E-07	4.942E-14
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	9	5	7	6	4	8	2	3	1
TP_{04}	Best	1.319E+02	7.151E+02	9.767E+03	6.210E+01	3.021E+01	1.530E+04	5.705E+01	7.132E+01	1.758E+01
	Median	8.322E+02	2.451E+03	2.820E+04	3.691E+02	6.511E+01	3.268E+04	4.720E+02	2.995E+02	5.573E+01
	Worst	1.284E+04	4.648E+03	5.431E+04	4.072E+03	2.430E+03	6.471E+04	5.653E+03	1.407E+03	1.364E+02
	Mean	1.539E+03	2.641E+03	2.966E+04	7.269E+02	6.192E+02	3.440E+04	1.008E+03	3.595E+02	5.962E+01
	Std	2.433E+03	1.071E+03	9.601E+03	9.667E+02	6.921E+02	1.100E+04	1.117E+03	2.763E+02	2.623E+01
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	6	7	8	4	3	9	5	2	1
TP_{05}	Best	1.434E+00	2.061E+00	3.532E+00	1.080E+00	1.112E+00	4.287E+00	1.009E+00	1.267E+00	1.251E+00
	Median	2.131E+00	2.471E+00	7.968E+00	1.587E+00	1.521E+00	7.840E+00	1.094E+00	1.946E+00	3.604E+00
	Worst	2.674E+00	3.091E+00	1.210E+01	2.600E+00	1.800E+00	1.211E+01	1.354E+00	2.943E+00	5.726E+00
	Mean	2.116E+00	2.486E+00	7.988E+00	1.931E+00	1.470E+00	7.900E+00	1.106E+00	1.960E+00	3.599E+00
	Std	2.850E-01	2.911E-01	1.712E+00	3.741E-01	2.969E-01	1.731E+00	7.260E-02	3.986E-01	1.089E+00
	p-rank	+	+	—	+	+	—	+	+	—
	f-rank	5	6	9	3	2	8	1	4	7
TP_{06}	Best	6.495E+00	9.651E+00	1.132E+01	7.021E+00	9.649E+00	1.341E+01	6.993E+00	9.112E+00	5.881E+00
	Median	1.035E+01	1.162E+01	1.569E+01	9.821E+00	1.122E+01	1.565E+01	1.171E+01	1.097E+01	9.246E+00
	Worst	1.190E+01	1.300E+01	1.751E+01	1.301E+01	1.242E+01	1.741E+01	1.296E+01	1.169E+01	1.146E+01
	Mean	9.901E+00	1.141E+01	1.559E+01	9.868E+00	1.110E+01	1.561E+01	1.296E+01	1.080E+01	9.011E+00
	Std	1.474E+00	9.152E-01	1.221E+00	1.560E+00	7.721E-01	1.050E+00	1.593E+00	6.591E-01	1.308E+00
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	3	7	8	2	5	9	6	4	1
TP_{07}	Best	5.539E+02	3.848E+02	1.141E+03	-9.587E+01	1.348E+01	1.557E+03	-2.311E+02	-3.526E+01	5.909E+00
	Median	1.237E+03	9.192E+02	2.121E+03	4.700E+02	4.071E+02	2.311E+03	3.072E+02	2.575E+02	1.643E+02
	Worst	1.952E+03	1.457E+03	2.641E+02	1.100E+03	1.551E+03	2.642E+03	6.367E+02	7.411E+02	3.433E+02
	Mean	1.282E+03	9.071E+02	2.072E+03	4.841E+02	5.466E+02	2.231E+03	2.895E+02	3.140E+02	1.726E+02
	Std	4.224E+02	2.351E+02	4.132E+02	2.900E+02	3.741E+02	3.442E+02	2.053E+02	2.104E+02	9.947E+01
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	7	6	8	4	5	9	2	3	1
TP_{08}	Best	6.419E+00	4.668E+00	6.741E+00	4.110E+00	2.451E+00	7.220E+00	3.709E+00	4.839E+00	3.922E+00
	Median	7.014E+00	6.332E+00	8.449E+00	5.868E+00	5.385E+00	8.268E+00	5.376E+00	5.646E+00	5.909E+00
	Worst	7.313E+00	6.900E+00	9.091E+00	6.802E+00	7.011E+00	9.681E+00	6.910E+00	6.354E+00	6.633E+00
	Mean	6.982E+00	6.141E+00	8.322E+00	5.800E+00	5.186E+00	8.400E+00	5.358E+00	5.888E+00	5.601E+00
	Std	2.255E-01	5.841E-01	5.812E-01	6.400E-01	1.041E+00	5.600E-01	8.567E-01	5.754E-01	3.331E-01
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	7	6	8	4	1	9	2	5	3
TP_{09}	Best	3.621E+00	1.321E+02	3.269E+03	3.600E+00	3.321E+00	2.812E+03	3.129E+00	3.124E+00	2.439E+00
	Median	1.644E+02	3.286E+02	5.900E+03	5.100E+00	5.577E+00	6.400E+03	5.191E+00	3.899E+00	2.655E+00
	Worst	1.467E+03	7.500E+02	1.069E+04	2.748E+02	7.775E+00	9.168E+03	7.963E+02	8.782E+01	3.026E+00
	Mean	2.788E+02	3.686E+02	6.112E+03	1.521E+01	5.689E+00	6.567E+03	3.125E+01	6.843E+00	2.681E+00
	Std	3.122E+02	1.411E+02	1.952E+03	4.976E+01	9.789E-01	1.586E+03	1.445E+02	1.531E+01	1.689E-01
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	6	7	8	4	2	9	5	3	1
TP_{10}	Best	2.013E+01	2.031E+01	2.072E+01	2.000E+01	2.031E+01	2.049E+01	1.428E+00	1.949E+01	1.988E+01
	Median	2.040E+01	2.049E+01	2.111E+01	2.011E+01	2.051E+01	2.100E+01	2.035E+01	2.049E+01	2.004E+01
	Worst	2.065E+01	2.071E+01	2.149E+01	2.048E+01	2.056E+01	2.141E+01	2.075E+01	2.065E+01	2.044E+01
	Mean	2.042E+01	2.049E+01	2.111E+01	2.010E+01	2.049E+01	2.100E+01	1.865E+01	2.045E+01	2.008E+01
	Std	1.295E-01	7.451E-02	2.078E-01	1.410E-01	9.451E-02	1.742E-01	5.429E+00	1.771E-01	1.076E-01
	p-rank	—	—	—	—	—	—	—	—	—
	f-rank	4	6	9	3	7	8	1	5	2
$w/l/t$		1/9/0	1/9/0	0/10/0	1/9/0	2/8/0	0/10/0	3/7/0	1/9/0	NA
Average f-rank		6.20	5.80	8.00	4.30	3.60	8.60	2.90	3.70	1.90
Overall f-rank		7	6	8	5	3	9	2	4	1



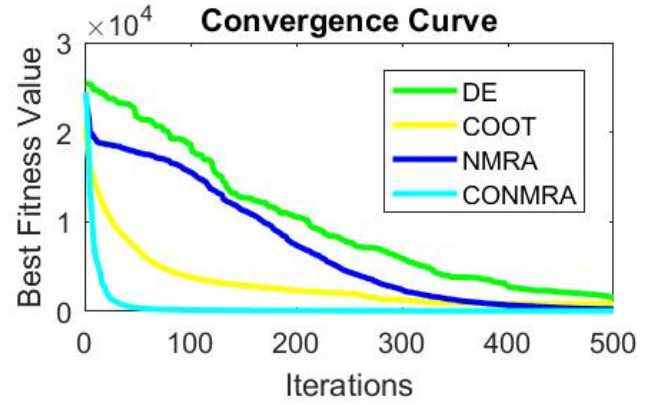
(a) Convergence profile for TP_{01}



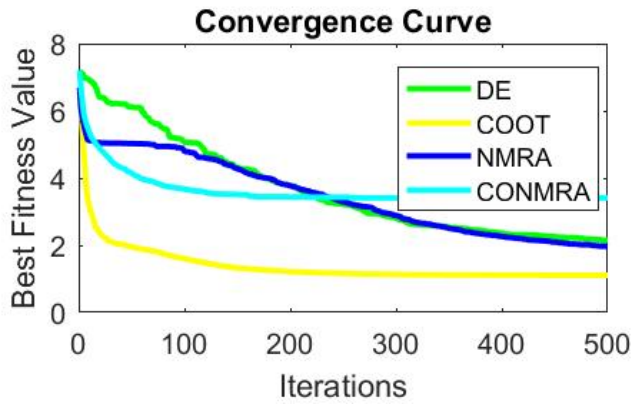
(b) Convergence profile for TP_{02}



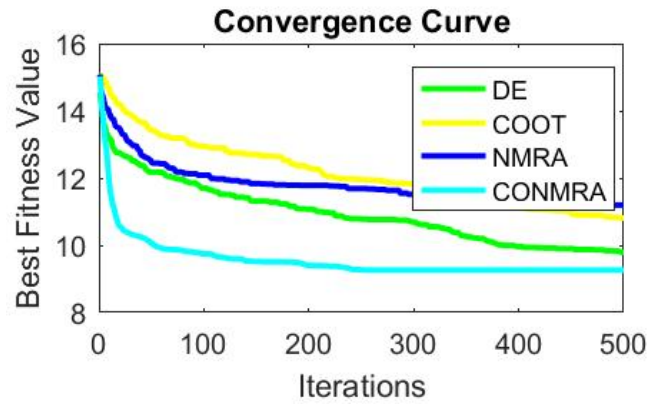
(c) Convergence profile for TP_{03}



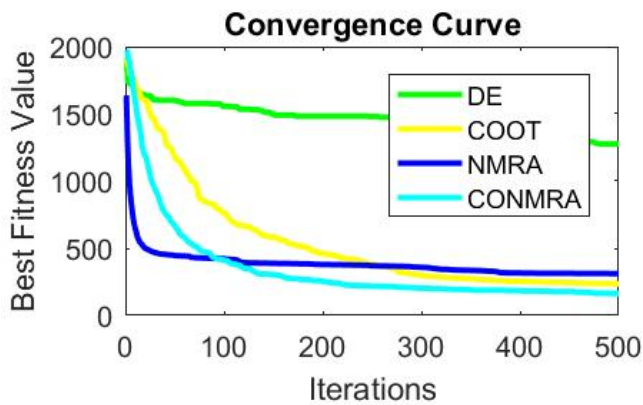
(d) Convergence profile for TP_{04}



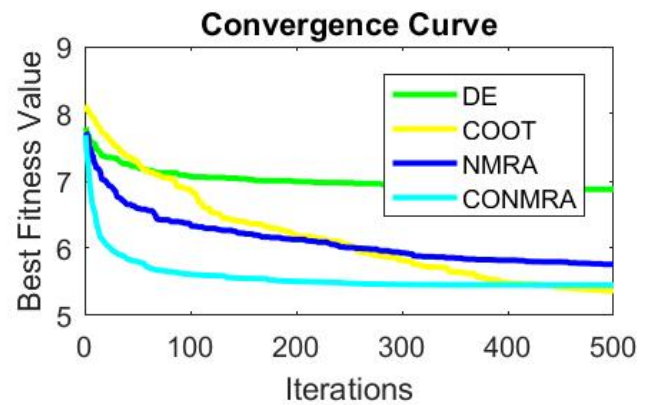
(e) Convergence profile for TP_{05}



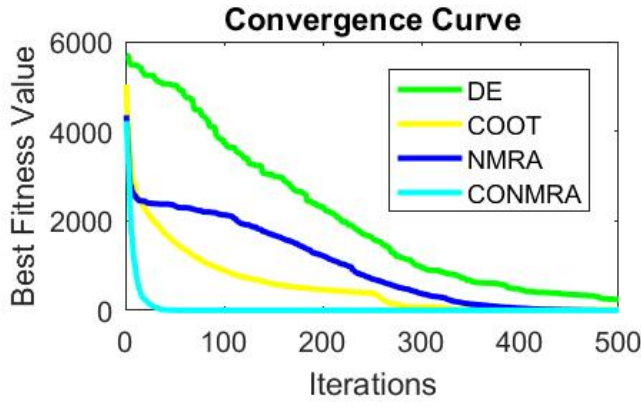
(f) Convergence profile for TP_{06}



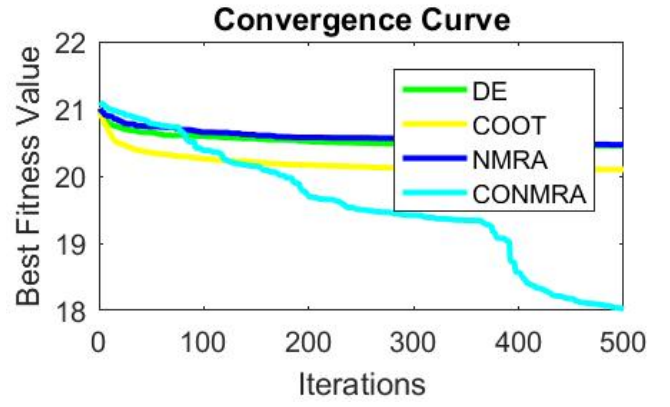
(g) Convergence profile for TP_{07}



(h) Convergence profile for TP_{08}



(i) Convergence profile for TP_{09}



(j) Convergence profile for TP_{10}

Figure 6.2: Convergence profiles of DE, COOT, NMRA and CONMRA for 100-digit (CEC 2019) numerical test problems

each search agent's fitness. The population is then put through several repetitions to create an evolved population. These procedures are iteratively performed until the termination condition is met. Subsequently, the search agent with the highest fitness is utilized to initiate the next step, which involves associating non-CH nodes with their respective CHs to create clusters.

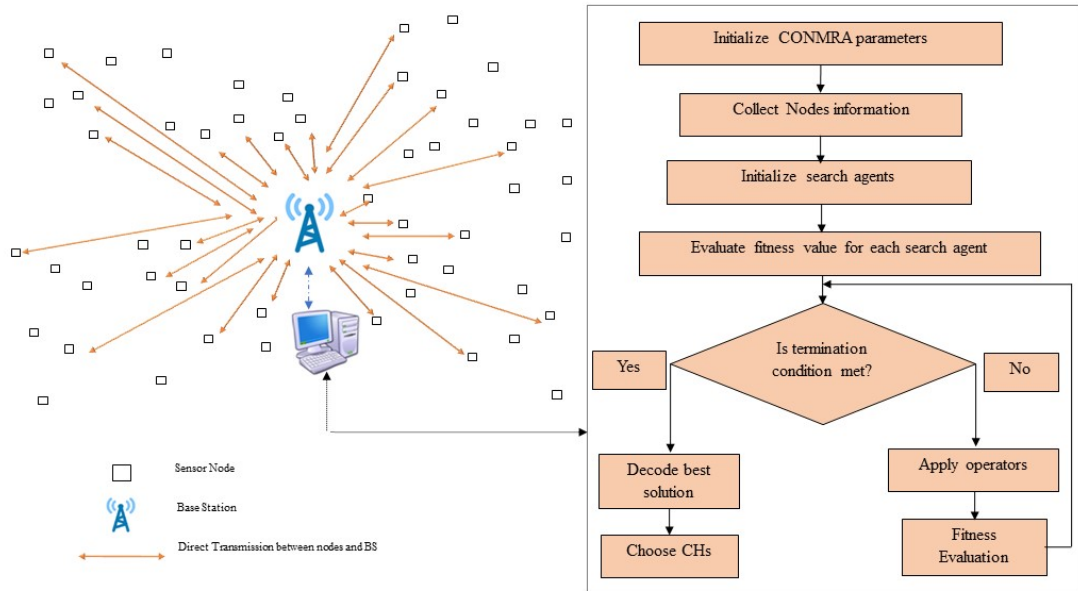


Figure 6.3: CH using CONMRA and exchange information between SNs and BS in setup phase

6.4.2 Cluster setup

The advertisement message containing the CH's ID is crucial in enabling the newly elected CH to communicate network status information. When a node receives a join request from the nearest CH,

it responds using the CSMA MAC protocol.

6.4.3 Development of TDMA schedule

After the CH election process, the BS assigns TDMA schedules to each CH for communication during each round. Additionally, the BS provides each CH with a unique coding scheme for intra-cluster communication with its member nodes. The entire network receives announcement messages from each CH. Upon receiving these broadcast messages, each sensor node selects the connection with the highest SNR. Individual sensor nodes respond to their selected cluster affiliation for the round requiring a CH. The CH then allocates a transmission time slot within the round, along with its coding scheme, for the node to send its data. Each node is assigned a time slot and coding scheme for communication with the CH, while the CH is allocated a time slot for transmitting all of the cluster's data to the BS. At this point, all WSN clusters have been established. This cycle continues until the CH has received all packets from the connected nodes. During a round, an SN can transmit its data directly to the BS, assuming that every node is within the BS's communication range, even if it cannot join a CH due to wireless link failures or issues with announcements or join messages.

The steady-state phase comprises two main elements: intra-cluster communication and inter-cluster communication. In intra-cluster communication, sensing nodes regularly transmit data to their respective CHs as part of the reactive protocol. The CHs, in turn, aggregate this data and communicate it to the BS using either direct or dual-hop channels in the inter-cluster communication phase.

6.4.4 Evaluation of fitness

Consider a WSN containing N sensors distributed randomly across the sensing field. In CNSTERP, the process of electing CHs for efficient cluster formation with minimal energy consumption involves utilizing a population of search agents. Each of these search agents is represented by a fixed-length list, with the length of the list being equal to the total number of nodes present in the WSN. Here, the population vector of a WSN with N sensors is denoted by the symbol $CN_{p,q}$, where $CN_{p,q} \in [0, 1]$. The values 1 and 0 are used for CH and ordinary nodes, respectively. Based on the required percentage of CHs and the use energy aware heuristics in CH selection, search agents randomly produce initial solutions in terms of 0s and 1s, as demonstrated by:

$$CN_{p,q} = \begin{cases} 1, & \text{if } (rand \leq CH_{pr} \text{ and } E(node_j) \geq E_{avg}(r)) \\ 0, & \text{otherwise} \end{cases} \quad (6.9)$$

where variable "rand" denotes a uniformly distributed random value ranging from 0 to 1. CH_{pr}

represents the anticipated percentage of CHs, $E(node_j)$ signifies the current energy level of sensor node j and $E_{avg}(r)$ corresponds to the average energy of the entire network in the round.

The clusters S formed by the randomly placed SNs are designated as $[CH_1, CH_2, \dots, CH_S]$. The least energy dissipation, the least cluster cohesiveness, and the greatest cluster separation are used to pick the CHs in order to minimise the cost of the fitness function. This fitness function can be mathematically expressed as follows:

$$f_{CH,obj} = \sum_{l=1}^2 w_l \times f_l; \quad \text{subject to } \sum_{l=1}^2 w_l = 1 \quad (6.10)$$

The primary goal is to reduce the amount of energy that each node uses overall [163] and this is demonstrated by:

$$f_1 = \left(\sum_{s=1}^S \sum_{node_j \in CH_s} E_{TX_{node_j, CH_s}} + E_{RX} + E_{DA} \right) + \sum_{s=1}^S E_{TX_{CH_s, BS}} \quad (6.11)$$

where S represents the total number of CHs, node j denotes a CM belonging to the s^{th} CH, and $E_{TX_{node_1, node_2}}$ represents the energy consumption associated with transmitting data from $node_1$ to $node_2$.

Energy consumption in WSN is categorized into various components, including sensing, data storage, and transmission. To calculate the energy spent on packet transmission, the first-order radio model, as illustrated in Figure 6.4, is employed. This model considers the separation between the transmitter and receiver, utilizing both the Friis and two-ray models to estimate path loss. The transmission cost for a message with a length of k bits and a distance d between the transmitter and receiver is expressed by the following equation:

$$E_{TX} = \begin{cases} k_{bits} E_{elec} + k_{bits} \varepsilon_{friss_amp} d^2, & \text{if } d < d_0 \\ k_{bits} E_{elec} + k_{bits} \varepsilon_{two_ray_amp} d^4, & \text{if } d \geq d_0 \end{cases} \quad (6.12)$$

The energy consumed per bit during transmission is denoted as E_{elec} . When the distance surpasses the threshold d_0 , the transmission energy follows a proportional relationship with d^4 ; otherwise, it is proportional to d^2 . The transmitter amplification parameters for the free-space and multipath fading models are ε_{friss_amp} and $\varepsilon_{two_ray_amp}$. The formula for computing d_0 is as follows:

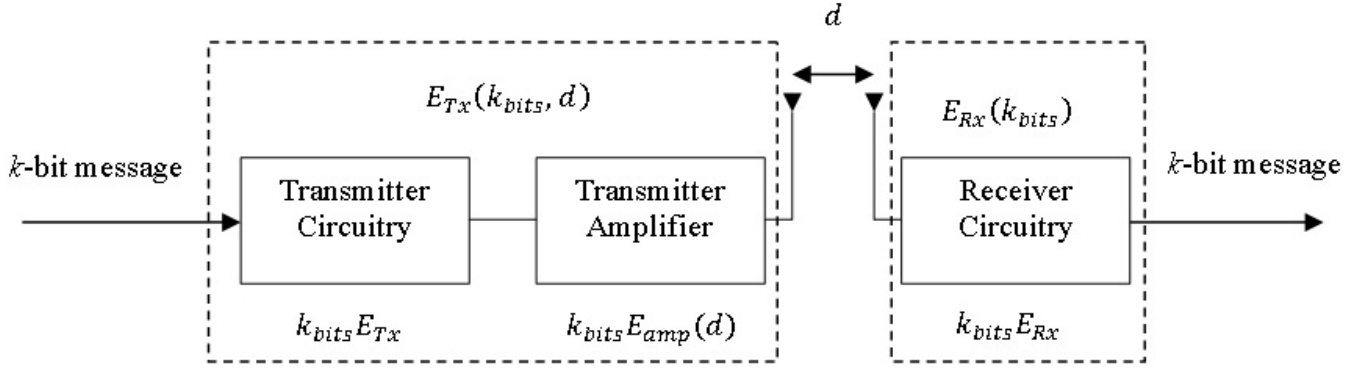


Figure 6.4: Radio energy dissipation model

$$d_0 = \sqrt{\varepsilon_{friss_amp} / \varepsilon_{two_ray_amp}} \quad (6.13)$$

The following provides the reception cost for the k-bit data message:

$$E_{RX} = k_{bits} E_{elec} \quad (6.14)$$

where E_{elec} represents the energy lost during reception per bit.

The second goal is to improve clustering quality, which is determined by cluster separation and cohesion [42]. Clustering is considered advantageous if the cohesiveness-to-separation ratio is low. Hence, the objective is to minimize the intra-cluster distance (enhance cluster cohesiveness) between CHs and their respective CMs, while maximizing the minimum inter-cluster distance (promote cluster separation) between any two different CHs. The fitness function designed to achieve this goal is computed by considering the shortest Euclidean distance between two distinct CHs and the ratio of the total Euclidean distance from CHs to their CMs.

$$f_2 = \frac{\sum_{s=1}^S \sum_{\forall node_j \in CH_s} d(node_j, CH_s)}{\min \forall CH_u, CH_s, CH_u \neq CH_s d(CH_u, CH_s)} \quad (6.15)$$

In clustering protocols, CHs assemble and send periodic data from their CMs to a distant BS. If CH fails, the performance of the entire network can suffer. As a result, the objective is to select the CH node in the cluster with the most energy remaining.

6.5 Simulation results of clustering protocols for WSN

The proposed clustered routing protocol CNSTERP performance has been compared against LEACH [148], HCR [280], ERP [42], SEEC [281], SEP-E [266], DESTERP [166] and HSSTERP [168]. The evaluation criteria include the total remaining energy in the network, the stability period, and the overall network lifetime. These simulations were conducted using MATLAB version 2021a, and a summary of the network properties employed in the clustering protocol simulations can be found in Table 6.2.

Table 6.2: Characteristics for setup of WSN

Parameter	Value used in simulation
Count of nodes	100
CH selection probability	0.05
Position of BS	(50, 50)
Network's size	100m $\times$ 100m
Radio electronics energy, $E_{TX} = E_{RX}$	50nJ/bit
Normal node energy, E_0	0.25J or 0.5J or 1J
Radio amplifier energy, ε_{friss_amp}	100pJ/bit/m ²
Radio amplifier energy, $\varepsilon_{two_ray_amp}$	0.0013pJ/bit/m ⁴

This section examines how the CNSTERP-based clustering protocols affect both homogeneous and heterogeneous situations. Here, to ensure that each fitness parameter is given the same weight in CNSTERP, the fitness function of equation (7.12) is set to the $w_l = 1/2$. In the case of homogeneous design, 100 sensor nodes in a 100m $\times$ 100m area with E_0 starting energy deployed randomly and a middle sink. However, in the case of a heterogeneous situation, super and advanced nodes with initial energies of $2E_0$ and $3E_0$, respectively, make up 10% and 20% of all sensor nodes in the network. The time period between consecutive cluster reconfigurations is called a communication round. The 100 sensor nodes network model is presented in Figure 6.5(a). The set of nodes elected as CHs at time t is shown in Figure 6.5(b), but at time $t + d$, a new set of sensor nodes elect themselves as CHs as shown in Figure 6.5(c). Non-CH nodes are represented by ‘o’, and CHs are represented by ‘★’. The network lifetime is calculated by the projected number of active sensors at each round. The longevity of the network and the total amount of data transferred are determined by counting the communication cycles before the last node dies. The various parameters involved in nature-inspired optimization algorithms-based clustering protocols are listed in Table 6.3.

The network’s stability period refers to the duration from the initial time to the FND occurrence. Another gauge of network lifespan is HND. It is defined as the period between the network’s beginning

Table 6.3: Simulation parameters for clustering protocols

Clustering Protocol	Parameters
LEACH	$CN_n = 30$; $dim = 100$; $t_{max} = 300$
HCR	$CN_n = 30$; $dim = 100$; $t_{max} = 300$; mutation probability ($P_m = 0.03$); crossover probability ($P_c = 0.7$)
ERP	$CN_n = 30$; $dim = 100$; $t_{max} = 300$; mutation probability ($P_m = 0.03$); crossover probability ($P_c = 0.7$)
SEEC	$CN_n = 30$; $dim = 100$; $t_{max} = 300$
DESTERP	$CN_n = 30$; $dim = 100$; $t_{max} = 300$; crossover rate ($CR = 0.5$)
HSSTERP	$CN_n = 30$; $dim = 100$; $t_{max} = 300$; $HMC R = 0.7$; $PAR = 0.1$
CNSTERP	$CN_n = 30$; $dim = 100$; $t_{max} = 300$; λ =adaptive; $a_1=0.5$; $a_2=0.4$

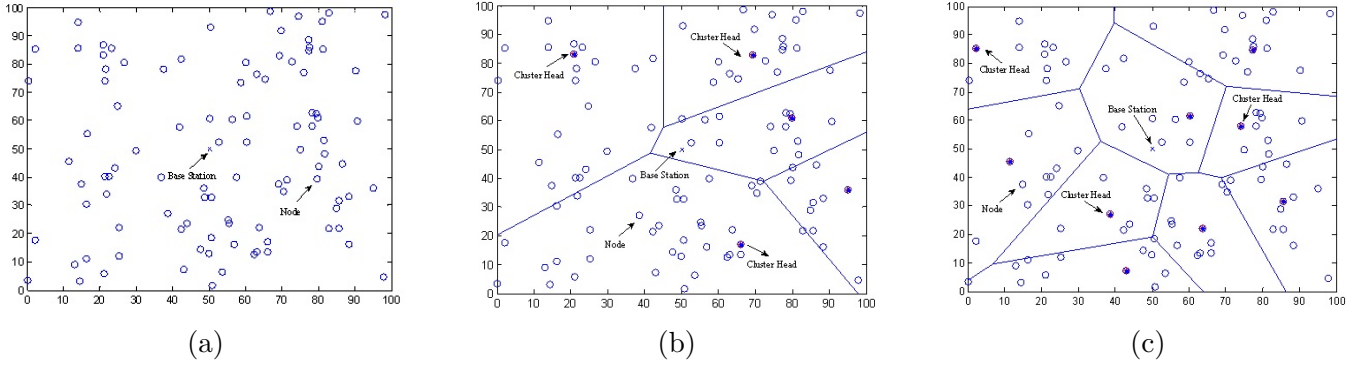


Figure 6.5: (a) 100 nodes network model (b) set of nodes elected as CHs at time t (c) a new set of sensor nodes elect themselves as CHs at time $t + d$

Table 6.4: Network lifespan analysis for homogeneous configuration with $E_0=1J$ in terms of dead nodes round history

Dead Nodes (%)	LEACH	HCR	ERP	SEEC	DESTERP	HSSTERP	CNSTERP
1 (FND)	1799.5	1722.5	2106.3	5101.6	5691.5	5630.3	5819.1
10	2018.4	2044.3	2272.7	5129.1	5708.3	5651.5	5872.3
20	2067.7	2181.6	2360.2	5133.3	5715.6	5659.7	5879.2
30	2138.1	2309.7	2437.6	5134.8	5720.4	5667.3	5883.5
40	2162.3	2417.1	2500.1	5140.5	5721.6	5671.4	5891.3
50 (HND)	2211.2	2520.2	2576.3	5141.7	5723.9	5674.9	5895.7
60	2276.1	2627.3	2650.5	5143.6	5725.8	5678.3	5897.6
70	2341.6	2748.4	2742.1	5147.1	5727.4	5679.1	5904.3
80	2389.7	2911.6	2835.9	5150.0	5731.8	5684.2	5906.1
90	2479.3	3100.5	2981.4	5152.1	5734.6	5687.3	5908.7
100 (LND)	2761.4	3568.3	3300.6	5154.3	5737.7	5711.7	5913.2

and the point at which 50% of its nodes are still operational. When a network reaches the point where no live nodes are left, it has reached its LND state. The instability period is the period between FND and LND. As illustrated in Table 6.4, the network lifespan of CNSTERP improves dramatically for homogeneous setups with $E_0 = 1J$. These findings demonstrate the importance of sensor node collaboration in reducing the workload associated with CH activities. Due to this, most nodes nearly die together after living together for the longest feasible time. These nodes seem to experience failures more frequently within clusters rather than as isolated occurrences. This is in contrast to the situation where there is a lot of fluctuation and uneven distribution of residual energy between different sensor nodes.

The line showing the alive nodes in the CNSTERP gives the appearance of a step function with a sharp reduction, whereas the alive nodes in the LEACH variants seem to be falling more gradually, as seen in Figure 6.6. Hence, most sensor nodes in LEACH die off in the early communication rounds. Contrarily, the CNSTERP fixes this problem and keeps trying to lengthen the network's lifespan whenever it is worth keeping it alive.

The graph in Figure 6.7 displays the average residual energy of the network throughout the communication rounds in a homogeneous scenario. When compared to other competing technologies, CNSTERP demonstrates more efficient energy utilization. This suggests that CNSTERP effectively manages and distributes energy stably and consistently throughout each communication ses-

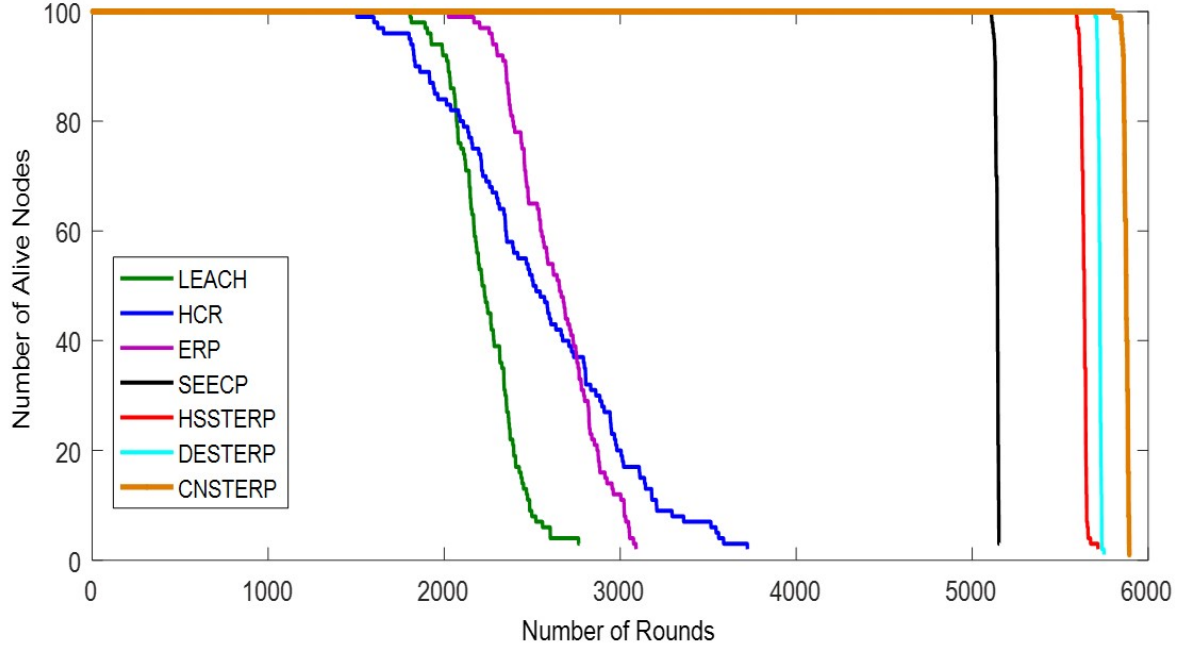


Figure 6.6: Count of alive nodes for homogeneous setup using $E_0 = 1J$ with progress of rounds

Table 6.5: Network lifespan analysis for heterogeneous configuration with $E_0 = 1J$ in terms of dead nodes round history

Dead Nodes (%)	SEP-E	HCR	ERP	SEECP	DESTERP	HSSTERP	CNSTERP
1 (FND)	2266.4	2506.8	2031.7	7166.4	7374.1	7281.6	7430.2
10	2306.3	2448.2	2279.4	7183.7	7419.7	7315.4	8096.6
20	2450.7	2647.1	2421.9	7191.8	7429.6	7322.8	8110.3
30	2549.8	2845.6	2578.6	7193.3	7433.4	7335.1	8121.4
40	2604.1	2991.3	2670.1	7196.2	7439.2	7339.5	8129.7
50 (HND)	2698.6	3020.4	2762.3	7199.1	7441.5	7343.4	8133.9
60	2785.4	3124.7	3098.4	7204.4	7442.2	7363.2	8138.1
70	3248.3	4024.4	4483.2	7206.6	7446.7	7370.7	8141.5
80	4269.5	4990.8	5104.7	7209.4	7448.4	7379.2	8147.2
90	4600.3	7984.9	6275.9	7209.8	7450.2	7386.9	8155.1
100 (LND)	5870.2	9862.4	8431.6	7210.9	7458.3	7393.7	8183.9

sion.

Table 6.5 presents a comparison of CNSTERP's performance with other state-of-the-art protocols in a heterogeneous network scenario. SEP-E doesn't distinguish between regular, intermediate, and advanced nodes when selecting CHs, leading to a higher frequency of selecting advanced and/or super nodes as CHs. This imbalance in the load causes CHs to deplete energy more rapidly (normal nodes). For HSSTERP and DESTERP, where intermediate and/or advanced nodes are initially chosen as CHs, the energy of advanced nodes decreases as the rounds progress until it becomes comparable to the energy of standard nodes. In CNSTERP, CHs are selected using CONMRA if their residual energy exceeds a node's threshold energy level. Consequently, nodes with higher energy levels (intermediate and/or advanced nodes) are more likely to be chosen as CHs.

The count of alive nodes during communication rounds for the HCR, DESTERP, SEECP, SEP-E, ERP, HSSTERP, and CNSTERP in heterogeneous configuration with $E_0 = 1J$ is shown in Figure

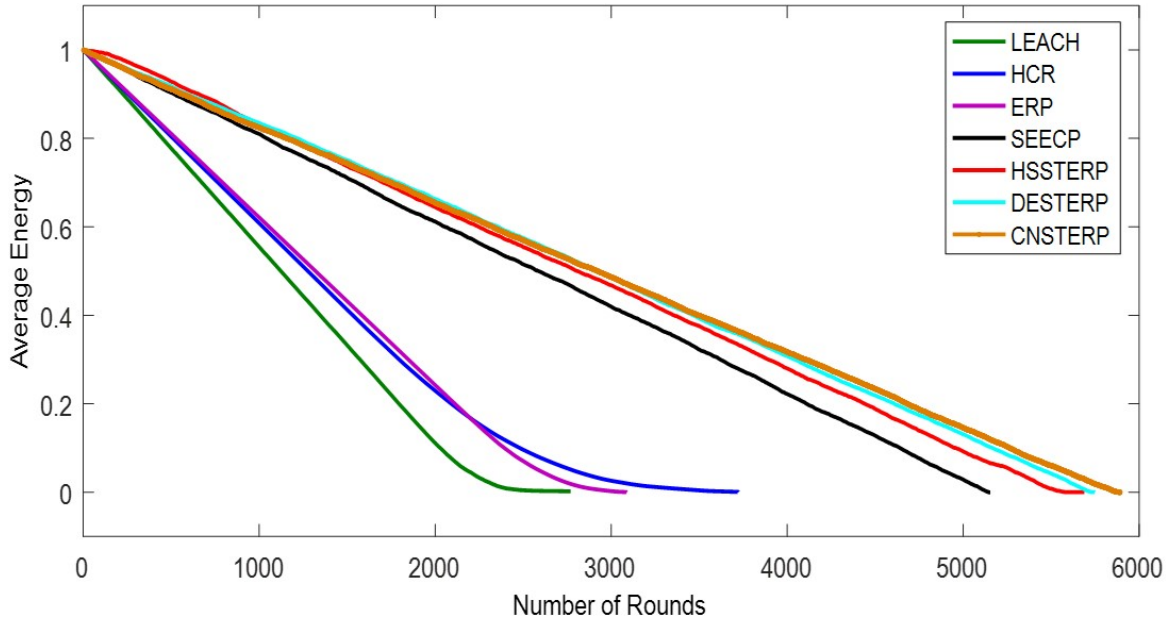


Figure 6.7: Average energy over rounds for homogeneous setup with $E_0 = 1J$

6.8. For extending the network lifetime with a much higher number of communication rounds, CNSTERP outperforms other competitive protocols in this situation. The initial rounds of the CNSTERP consider the residual energy of nodes when choosing CH, and high-energy nodes are always chosen as CHs, improving the load distribution among the nodes.

Figure 6.9 shows the network's typical leftover energy for communication rounds in a heterogeneous network configuration. CNSTERP consumes less energy overall than other rival techniques. It demonstrates that the CNSTERP seems to continuously and steadily absorb energy throughout each communication loop.

Table 6.6 provides a comparison of CNSTERP, along with competing algorithms, in terms of FND, HND, and LND, as well as the stability and instability periods for an initial energy budget of $E_0 = 1J$. CNSTERP exhibits a significantly longer stability period and outperforms the competing protocols due to its efficiency in selecting optimal CHs.

6.6 Summary

This chapter introduces a novel self-adaptive hybrid optimization approach called CONMRA, which combines the strengths of the recently developed NMRA and COOT optimization algorithms. The effectiveness of CONMRA is evaluated using the CEC 2019 test suite. It is observed that CONMRA outperforms most of the compared algorithms across various test problems. The superiority of

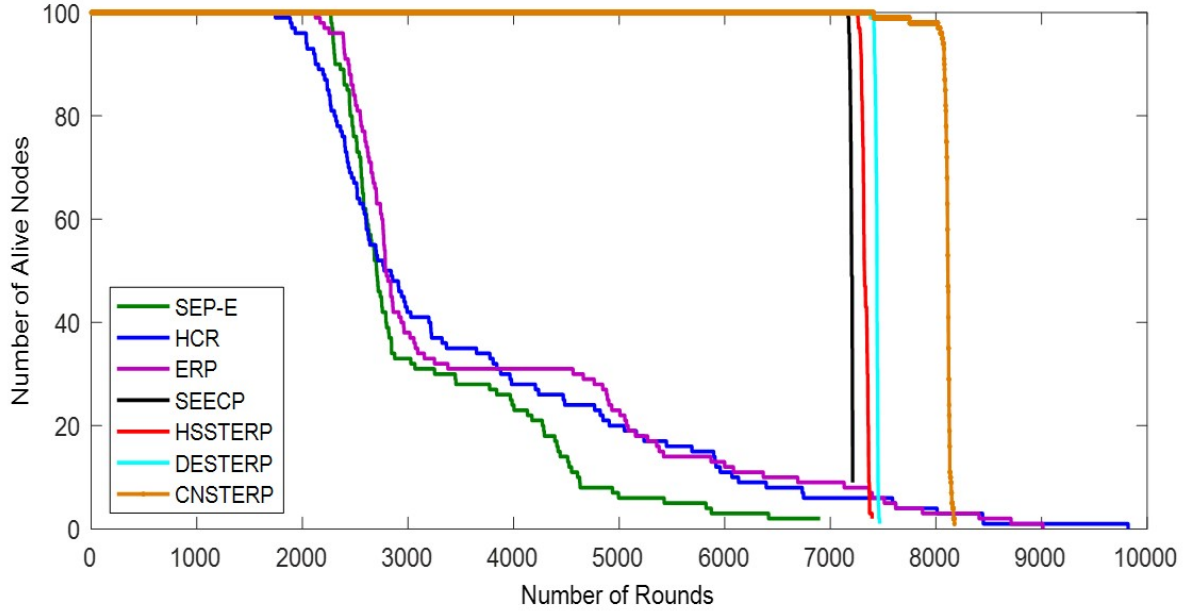


Figure 6.8: Count of alive nodes for heterogeneous setup using $E_0 = 1J$ with progress of rounds

Table 6.6: Network lifespan of routing protocols for $E_0 = 1J$ with stability and instability periods

Setup	Protocol	FND	HND	LND	Stability period	Instability period
Homogeneous Setup	LEACH	1799.5	2211.2	2761.4	1799.5	961.9
	HCR	1722.5	2520.2	3568.3	1722.5	1845.8
	ERP	2103.3	2576.3	3300.6	2103.3	1197.3
	SEECP	5101.6	5141.7	5154.3	5101.6	52.7
	DESTERP	5691.5	5723.9	5737.7	5691.5	46.2
	HSSTERP	5630.3	5674.9	5711.7	5630.3	81.4
	CNSTERP	5819.1	5895.7	5913.2	5819.1	94.1
Heterogeneous Setup	SEP-E	2266.4	2698.6	5870.2	2266.4	3603.8
	HCR	2506.8	3020.4	9862.4	2506.8	7355.6
	ERP	2031.7	2762.3	8431.6	2031.7	6399.9
	SEECP	7166.4	7199.1	7210.9	7166.4	44.5
	DESTERP	7374.1	7441.5	7458.3	7374.1	84.2
	HSSTERP	7281.6	7343.4	7393.7	7281.6	112.1
	CNSTERP	7430.2	8139.9	8183.9	7430.2	753.7

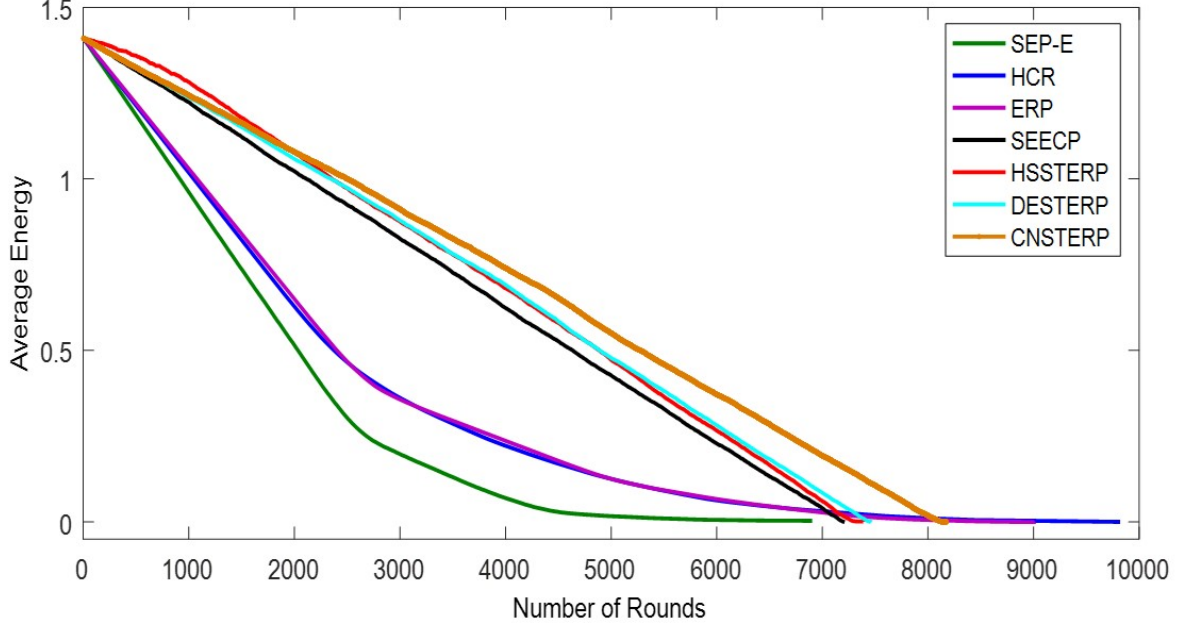


Figure 6.9: Average energy over rounds for heterogeneous setup using $E_0 = 1J$

CONMRA is further confirmed through non-parametric statistical tests (p-rank and f-rank), demonstrating its statistical significance. Furthermore, the proposed clustering protocol CNSTERP is introduced, efficiently utilizing input parameters to make informed decisions regarding CH selection. CNSTERP is evaluated in both homogeneous and heterogeneous network scenarios, considering factors like overall energy consumption and network lifetime in WSNs. The simulation results indicate that CNSTERP outperforms state-of-the-art clustering protocols in terms of network lifetime and average energy consumption.

Chapter 7

A Self-adaptive Attraction and Repulsion-based Naked Mole-Rat Algorithm for Energy-efficient Mobile WSN

NMRA is a swarm intelligence-based algorithm that draws inspiration from the mating behavior of mole rats (workers and breeders). This approach, which is based on the ability of breeders to reproduce with the queen, has been utilized to tackle optimization problems. The algorithm, however, suffers from local optima stagnation problems and a slower rate of convergence in order to provide global optimal solution. This study develops attraction and repulsion strategy based NMRA (ARNMRA) along with self-adaptive properties to avoid solution trapping in local optima. This strategy is utilized to create new breeder rat solutions and mating factor λ is made self-adaptive using (sa) based mutation operator. ARNMRA is evaluated on CEC 2005 numerical benchmark problems and is superior to other algorithms, including well-known ones like SOGWO, OB-L-EO, IWOA, SHADE, and original NMRA. Further, according to experimental results, the performance of ARNMRA is likewise superior to the NMRA for the CEC 2019 numerical problems. Convergence profiles and statistical tests (rank-sum test and Friedman test) are employed further to validate the experimental results. Moreover, this chapter extends the application of ARNMRA to address the data gathering aspect in MWSNs to prolong network lifetime and enhance energy efficiency. In this MWSN-based protocol, a sensor node is elected as a CH based on factors like mobility, residual energy, and connection time. The protocol aims to maximize the system lifetime by efficiently collecting data from all sensors and transmitting it to the BS. The study emphasizes the significance of considering dynamic node densities and speed when designing effective data-gathering protocols for MWSNs.

7.1 Introduction

NMRA is a recently proposed optimization technique that mimics the natural breeding habits of mole rats. This algorithm requires tuning of only two parameters such mating factor λ and breeding probability bp . NMRA is used to address some actual optimization issues, including the location of nodes in wireless sensor networks that are [282] and the design of ultra wideband antennas with DE hybridization [283]. Despite being a competitive algorithm, NMRA suffers from a local optimal

stagnation problem, which spurs the researcher to create an improved version of the classical NMRA. In the present work, utilizing both the global worst and global best solutions [279] to implement the attraction and repulsion strategy in the breeder phase (exploitation phase) of the proposed algorithm named ARNMRA. This method enables the search agents (breeder rats) to wander arbitrarily under the effect of both attraction and repulsion to find the best solution. At this point, implementing the global worst solution can help increase population diversity, and the issue of premature convergence is handled.

WSNs consist of compact, energy-efficient sensor nodes that communicate without the need for wired connections. These nodes are designed to monitor and gather data from the surrounding physical environment using wireless communication technology [149]. These networks are typically used for applications such as environmental monitoring, industrial automation, healthcare, and smart cities. MWSNs introduce additional challenges compared to traditional static WSNs due to the mobility of sensor nodes. The movement of sensor nodes introduces dynamic changes in network topology, which can disrupt network connectivity and routing paths. Nodes may join or leave the network, leading to frequent topology re-configurations. Efficient mechanisms for node tracking, localization, and adaptability to node mobility are required. In MWSNs, sensor nodes are typically powered by batteries, and energy efficiency is crucial for prolonging the network's lifetime. However, node mobility can cause energy imbalances due to varying distances, resulting in some nodes depleting their energy faster than others. Strategies for energy-efficient routing, power management, and dynamic energy replenishment become essential.

Maintaining connectivity in a mobile environment is challenging. Nodes may move out of each other's communication range, leading to link failures and frequent disconnections. Reliable and robust communication protocols must be employed to handle intermittent connections, link quality variations, and node mobility-induced network partitions. Mobile sensor nodes often generate large volumes of data. Efficient data fusion and aggregation techniques are required to reduce redundancy and minimize the amount of data transmitted. However, due to node mobility, data fusion becomes more challenging as nodes move in and out of each other's sensing ranges. Routing in MWSNs becomes complex due to dynamic topology changes caused by node mobility. Traditional routing protocols designed for static WSNs may not be suitable. Adaptive routing algorithms that can dynamically adjust to changing network conditions, consider node mobility, and provide efficient path planning are necessary. Scalability is a concern in MWSNs, especially when many mobile nodes are involved. Efficient protocols and mechanisms are needed to handle the increased network size, dynamic topology changes, and frequent node movements without sacrificing network performance and resource

utilization. Addressing these challenges requires the development of specialized algorithms, protocols, and system designs tailored for MWSNs. Researchers continue to explore innovative solutions to overcome these obstacles and unlock the full potential of mobile sensing applications.

This chapter is divided into various sections, with an introduction to the article discussed in Section 7.1. The proposed approach for enhancing the effectiveness of the fundamental NMRA is presented in Section 7.2 and the statistical findings for the CEC 2005 and CEC 2019 numerical benchmark problems are covered in Section 7.3. Section 7.4 details the clustering algorithm for energy-efficient mobile WSN. Finally, Section 7.5 includes the chapter's summary.

7.2 Attraction and repulsion based NMRA

NMRA has drawn the interest of many researchers due to its success in solving a wide range of practical problems. Since it was developed, the basic, linear approach has been utilized to resolve optimization problems, but it still has the drawback of trapping in local optimal solution. Because of this, we present an enhanced version of NMRA in this study that is based on the attraction and repulsion strategy [279]. This suggested algorithm also contains a *sa*-based mutation operation [271] to confirm the mating factor λ applicability to NMRA. Here, it is important to ensure that the attraction and repulsion strategy has been used to modify the equation (2.4) and *sa* mutation operator is applied to λ parameter of the classical NMRA.

7.2.1 Adoption of attraction and repulsion strategy

The global optimal solution directs the breeder rats movement toward the ideal solution (mating with the queen). However, if the solution is trapped in the local optimal and unable to escape, the entire population will likely stagnate. This proposed strategy, ARNMRA uses the attraction-repulsion method to address this issue. Here, breeder rat solution equation (2.4) has been modified with the help of the global best and worst solution using the attraction-repulsion principle moving at random due to the effects of attraction and repulsiveness to discover the best optimal solution. The population's diversity will dramatically diminish as the algorithm's iterative process progresses, leading to premature occurrence. Introducing the global worst solution may contribute to a rise in population diversity. It solves the issue of early convergence and broadens the population's scope in the local search process. The following equation is used to obtain an attraction-repulsion strategy-based breeder rat solution:

$$MB_p^{t+1} = (1 - \lambda)MB_p^t + \lambda[\beta_1(MR_{best} - MB_p^t) - \beta_2(MR_{worst} - MB_p^t)] \quad (7.1)$$

where MB_p^t is the breeder mole rat solution for the current iteration t , MR_{worst} is the global worst position, MR_{best} is the global optimum solution. The values of β_1 and β_2 are considered as 0.5 and 0.4 respectively. The incorporation of attraction and repulsion during the mating process in the breeding mole rats within the ARNMRA algorithm leads to several advantages. Firstly, this interaction enhances the influence on the breeder mole rats, allowing them to be more strongly guided towards the global best solution, represented by the queen. By being influenced by both attraction and repulsion, the breeders are able to explore the solution space more effectively, increasing the likelihood of finding better solutions. Secondly, the interaction of attraction and repulsion promotes a higher level of diversity within the population. The attraction component encourages convergence towards promising regions of the search space, while the repulsion component discourages the population from clustering around local optima. As a result, the population maintains a greater diversity of solutions, which is beneficial for avoiding premature convergence and increasing the chances of finding the global optimum.

7.2.2 Parameter adaptation

The suggested optimization technique ARNMRA heavily relies on the original NMRA mating factor (λ). This parameter must be altered from its default definition in the basic NMRA to achieve better results. As a result, this option has been altered so that no changes at the user level are necessary and that the parameter λ is implemented using *sa* based mutation, which yields the best randomization outcomes. The *sa* based mutation technique is carried out using the generalised equation:

$$\beta_k = \beta_{min} + (\beta_{max} - \beta_{min}) \times d^{(s-1)} \quad (7.2)$$

where d is fixed at 0.95, β_{max} , β_{min} and s are produced at random from values between [0,1].

Overall, the ARNMRA optimization algorithm is developed using attraction and repulsion strategy and adaption of important parameter λ . The major purpose of these modifications is to increase mole-rats diversity, which will enhance exploitation properties and convergence activities.

7.3 Results and discussion

Two sets of numerical benchmark problems, including the CEC 2019 [9] and CEC 2005 numerical benchmark problems [7], are used to evaluate the effectiveness of the proposed ARNMRA. For these test problems, statistical results are produced and are then contrasted with other competitive meta-heuristic methods. The following subsections discuss the statistical outcomes of the proposed

optimization technique ARNMRA for these numerical problems.

7.3.1 Statistical results for CEC 2005 numerical benchmark problems

The statistical findings for the proposed ARNMRA and other competing algorithms are analyzed in this subsection. The 12 numerical problems from the CEC 2005 test suite are chosen, and descriptions of these problems are provided in Table A.1 to evaluate the efficiency of ARNMRA. The numerical problems utilized here can be broadly categorized into two groups: uni-modal problems (p_1 to p_7) and multi-modal problems (p_8 to p_{12}). Here, all of the statistical findings are attained with a 30 dimension size, 50 mole-rats population, and 500 iterations.

The statistical findings of ARNMRA and various competitive algorithms, including traditional NMRA, SHADE, OB-L-EO, SOGWO and IWOA, are presented in Table 7.1. For 51 runs of the algorithm, the statistical findings are presented as mean and standard deviation (std) values. According to these findings, ARNMRA outperforms all other algorithms for problems p_1 , p_2 , p_3 , p_4 , p_5 and p_7 . SHADE performs the best for problem p_6 and p_{12} . For numerical problems p_8 and p_{10} , the four approaches (OB-L-EO, SOGWO, NMRA and ARNMRA) achieve global minimal values while NMRA and ARNMRA provide identical results for p_9 . In the case of problem p_{10} , the working efficiency of OB-L-EO is superior compared to other optimization techniques.

To assess the operational efficiency of ARNMRA, the study employs two statistical tests: the rank-sum (p-rank) test [231] and the Friedman (f-rank) test [260]. In the p-rank test, the variables $win(w)/loss(l)/tie(t)$ are used to evaluate the performance of ARNMRA compared to other optimization techniques. A " + " symbol indicates that the compared technique outperforms ARNMRA $win(w)$, a " - " symbol denotes that it performs worse $loss(l)$, and " = " represents an equal performance $tie(t)$. The results in Table 7.1 demonstrate that ARNMRA outperforms the majority of the tested problems. Additionally, the f-rank test assigns a rank to each optimization method being evaluated. The fourth row of Table 7.1 displays the rank of each technique for each numerical test problem. After assigning f-ranks to all algorithms, the average f-rank value and overall f-rank are calculated (last two rows of Table 7.1). These results indicate that ARNMRA is a statistically significant optimization technique, consistently ranking at the top among all other algorithms. Here, both the p-rank and f-rank tests confirm the superior performance of ARNMRA. The p-rank test demonstrates that ARNMRA outperforms other techniques in most test problems, while the f-rank test reinforces its statistical significance and consistent top ranking among the evaluated algorithms.

For each numerical problem in the CEC 2005 test suite, convergence profiles of the basic NMRA and purposed ARNMRA are also drawn after the simulated results. The convergence graphs shown

in Figure 7.1 confirm that the proposed ARNMRA converges to the optimal solution more quickly than the conventional NMRA.

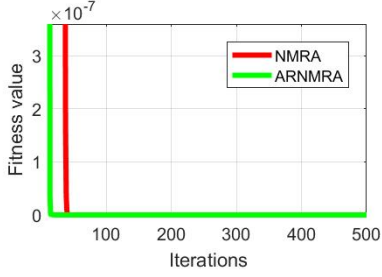
Table 7.1: Simulated outcomes of ARNMRA for CEC 2005 test suite

Problem		SHADE [213]	OB-L-EO [257]	SOGWO [259]	IWOA [5]	NMRA	ARNMRA
p_1	mean	1.421E-09	6.758E-212	6.050E-77	8.131E-146	1.539E-87	3.246E-292
	std	3.090E-09	0	1.487E-76	4.358E-145	1.097E-86	0
	p-rank	—	—	—	—	—	—
	f-rank	6	2	5	3	4	1
p_2	mean	8.700E-03	1.931E-108	1.179E-44	2.379E-102	2.134E-45	2.155E-150
	std	2.132E-02	8.000E-108	1.341E-44	6.577E-102	8.585E-45	1.164E-149
	p-rank	—	—	—	—	—	—
	f-rank	6	2	5	3	4	1
p_3	mean	1.542E+01	6.931E-187	6.821E-87	1.540E+04	4.503E-104	6.686E-250
	std	9.940E+00	0	3.420E-86	7.421E+03	3.029E-103	0
	p-rank	—	—	—	—	—	—
	f-rank	6	2	4	5	3	1
p_4	mean	9.786E-01	4.732E-103	1.131E-45	1.311E+01	5.526E-45	8.827E-145
	std	7.989E-01	1.600E-102	3.976E-45	1.612E+01	2.212E-44	6.294E-144
	p-rank	—	—	—	—	—	—
	f-rank	5	2	3	6	4	1
p_5	mean	2.441E+01	2.579E+01	2.887E+01	2.651E+01	2.897E+01	2.176E+01
	std	1.120E+01	1.556E-01	1.879E-02	6.600E-01	1.840E-02	4.070E-02
	p-rank	—	—	—	—	—	—
	f-rank	2	3	5	4	6	1
p_6	mean	5.311E-10	9.091E-05	6.776E+00	3.630E-02	6.610E+00	1.510E+00
	std	6.352E-10	5.977E-05	5.788E-01	6.947E-02	5.673E-01	5.178E-01
	p-rank	+	+	—	+	—	—
	f-rank	1	2	6	3	5	4
p_7	mean	2.352E-02	4.700E-04	5.932E-04	1.851E-03	6.230E-04	3.053E-04
	std	8.800E-03	3.041E-04	4.957E-04	2.358E-03	5.151E-04	3.296E-04
	p-rank	—	—	—	—	—	—
	f-rank	6	2	3	5	4	1
p_8	mean	8.531E+00	0	0	0	0	0
	std	2.190E+00	0	0	0	0	0
	p-rank	—	=	=	=	=	=
	f-rank	6	1	1	1	1	1
p_9	mean	3.952E-01	8.881E-16	8.881E-16	3.731E-15	8.881E-16	8.881E-16
	std	5.857E-01	0	0	2.168E-01	0	0
	p-rank	—	=	=	—	=	=
	f-rank	6	1	1	5	1	1
p_{10}	mean	4.800E-03	0	0	2.641E-03	0	0
	std	7.700E-03	0	0	1.100E-02	0	0
	p-rank	—	=	=	—	=	=
	f-rank	5	1	1	6	1	1
p_{11}	mean	3.458E-02	6.287E-06	5.611E-02	9.300E-03	1.118E+00	1.295E-01
	std	8.751E-02	4.351E-06	1.421E-02	2.561E-02	2.598E-01	9.180E-02
	p-rank	+	+	+	+	—	—
	f-rank	3	1	4	2	6	5
p_{12}	mean	7.321E-04	3.832E-02	3.531E-01	1.600E-01	2.991E+00	8.406E-01
	std	2.800E-03	9.751E-02	1.276E-01	1.365E-01	4.150E-02	3.230E-01
	p-rank	+	+	+	+	—	—
	f-rank	1	2	4	3	6	5
$w/l/t$		3/9/0	3/6/3	2/7/3	3/8/1	0/9/3	
f-rank value		4.41	2.00	3.50	3.83	3.75	1.91
Overall f-rank		6	2	3	5	4	1

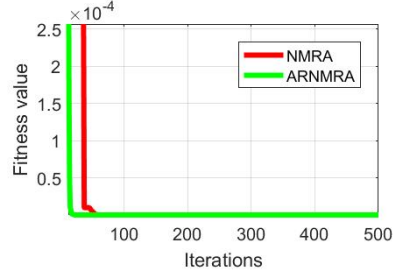
7.3.2 Statistical results for CEC 2019 numerical benchmark test suite

This subsection evaluates the proposed ARNMRA's performance for the 100-digit challenge (CEC 2019) problems. This test suite contains 10 numerical problems, and an explanation of every test problem can be found in Table A.2. Here, statistical findings for 51 runs of the algorithms for the original NMRA and the suggested ARNMRA are presented as average, best, median, worst, and standard deviation (std) values. The number of mole-rats should be set to 50, the problem's dimension should match the description, and maximum iterations of 500 should be used for generating the results.

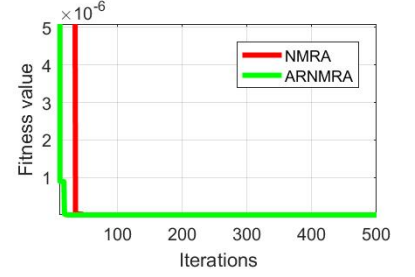
Table 7.2 contains the statistical findings produced for these numerical benchmark functions. For the benchmark problem b_1 , ARNMRA has a better operating capacity than classical NMRA. The suggested technique ARNMRA is the best for all performance measures for the benchmark problems b_2 , b_4 , b_5 , b_6 , b_8 and b_{10} test functions. For problem b_3 , the results are compared for standard values, and



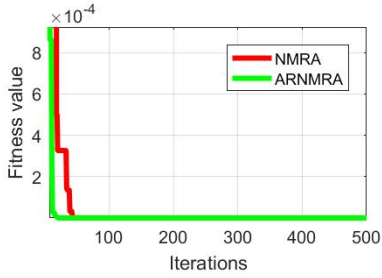
(a) Convergence profile (p_1)



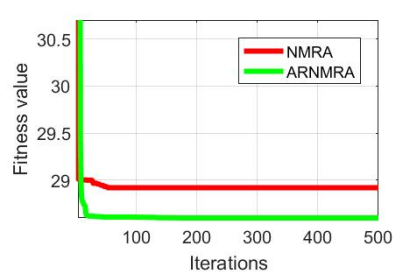
(b) Convergence profile (p_2)



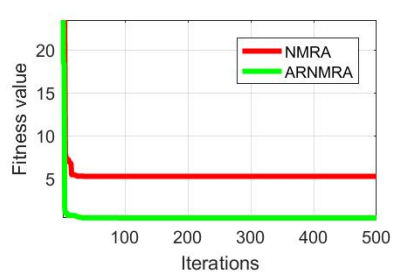
(c) Convergence profile (p_3)



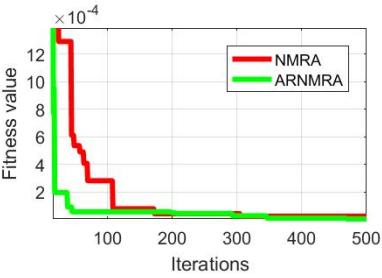
(d) Convergence profile (p_4)



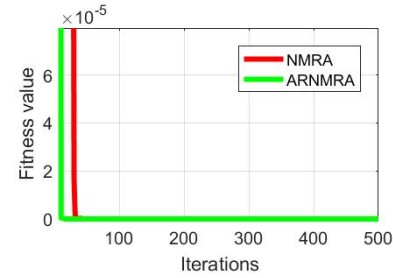
(e) Convergence profile (p_5)



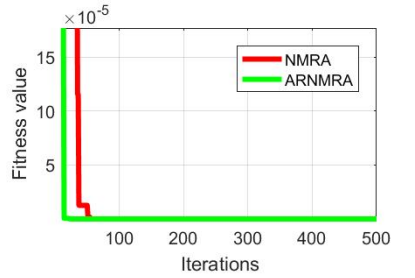
(f) Convergence profile (p_6)



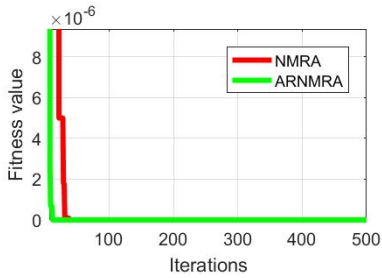
(g) Convergence profile (p_7)



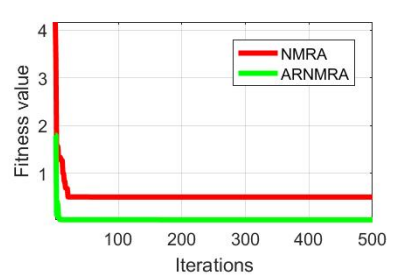
(h) Convergence profile (p_8)



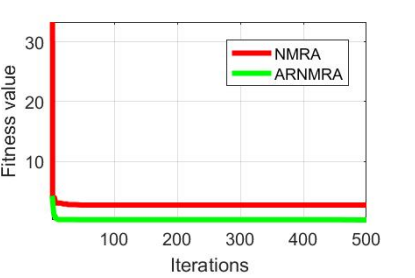
(i) Convergence profile (p_9)



(j) Convergence profile (p_{10})



(k) Convergence profile (p_{11})



(l) Convergence profile (p_{12})

Figure 7.1: Convergence profiles of NMRA and ARNMRA for CEC 2005 numerical benchmark problems

Table 7.2: Simulated outcomes for CEC 2019 test suite

Problem	Algorithm	Best	Median	Mean	Worst	Std
b_1	NMRA	8.320E+04	2.551E+05	3.221E+05	8.397E+05	2.071E+05
	ARNMRA	5.003E+04	1.168E+05	1.350E+05	4.848E+05	7.839E+04
b_2	NMRA	1.759E+01	1.858E+01	1.855E+01	1.971E+01	6.112E-01
	ARNMRA	1.778E+01	1.778E+01	1.784E+01	1.866E+01	3.432E-01
b_3	NMRA	1.270E+01	1.270E+01	1.270E+01	1.270E+01	2.956E-07
	ARNMRA	1.270E+01	1.270E+01	1.270E+01	1.270E+01	3.245E-08
b_4	NMRA	9.163E+01	3.663E+02	4.741E+02	1.636E+03	3.660E+02
	ARNMRA	9.600E+01	3.014E+02	3.528E+02	1.297E+03	2.361E+02
b_5	NMRA	1.331E+00	1.978E+00	1.985E+00	2.624E+00	2.792E-01
	ARNMRA	1.407E+00	1.939E+00	1.939E+00	2.589E+00	3.127E-01
b_6	NMRA	9.074E+00	1.121E+01	1.115E+01	1.255E+01	8.049E-01
	ARNMRA	9.124E+00	1.039E+01	1.036E+01	1.179E+01	6.533E-01
b_7	NMRA	-1.243E+02	1.981E+02	1.533E+02	4.042E+02	1.227E+02
	ARNMRA	-2.958E+02	1.019E+01	-2.860E+00	2.191E+02	1.192E+02
b_8	NMRA	4.974E+00	6.003E+00	5.919E+00	6.534E+00	3.304E-01
	ARNMRA	4.565E+00	5.663E+00	5.594E+00	6.151E+00	3.227E-01
b_9	NMRA	3.079E+00	4.179E+00	4.354E+00	1.016E+01	1.041E+00
	ARNMRA	3.061E+00	5.180E+00	1.126E+01	1.097E+02	1.932E+01
b_{10}	NMRA	1.951E+01	2.049E+01	2.048E+01	2.065E+01	1.628E-01
	ARNMRA	2.017E+01	2.037E+01	2.037E+01	2.057E+01	7.670E-02

ARNMRA provides the best performance. In the case of problems b_7 and b_9 , NMRA performs better than the proposed ARNMRA. Overall, it is determined that the suggested optimization technique ARNMRA is shown superior performance for most of these numerical benchmark problems.

7.4 Real Time Application: Energy-efficient ARNMRA based routing protocol (EARNRP) for MWSNs

Clustering in sensor networks is utilized to reduce energy consumption. However, existing clustering protocols for mobile sensor nodes encounter difficulties in maintaining energy efficiency because they do not adequately consider node movement after clustering. While mobile sensor nodes can offer improved network coverage and connectivity compared to static nodes, effectively managing their operation in line with specific application requirements is complex. In various mobile scenarios, existing clustering protocols for mobile sensor nodes often struggle to tackle the challenges related to energy efficiency adequately. One of the key reasons behind this is their failure to account for node movement after clustering. To address the energy efficiency challenges in mobile scenarios, developing clustering protocols that can adapt to node mobility becomes crucial. These protocols should consider the movement patterns of nodes and incorporate energy-efficient strategies. By incorporating both node mobility and energy constraints into the design of clustering protocols, it becomes possible to develop more effective solutions that significantly enhance the energy efficiency of mobile sensor networks.

7.4.1 Radio model

The first-order radio communication model [148] is a widely used model in Wireless Sensor Networks (WSNs) to estimate energy consumption during communication. It considers the energy the radio electronics and power amplifier dissipates when transmitting a packet from a sender to a receiver. The

energy consumption is influenced by factors such as the packet size, distance between the sender and receiver, path loss, and fading effects. Figure 6.4 illustrates the components involved in the energy dissipation process. To characterize signal propagation in wireless communication, two commonly used models are the free space model and the multipath fading model. These models define how the signal strength and quality vary with the transmitter and receiver separation (d). The free space model assumes that the signal propagates through a clear and unobstructed space, whereas the multipath fading model accounts for the effects of reflections, diffractions, and scattering caused by objects and the environment. By utilizing these models, the first-order radio communication model estimates energy consumption based on the distance between communicating nodes and other relevant factors. It is a valuable tool for assessing and optimizing energy efficiency in WSNs.

In the free space model, the assumption is made that wireless signals propagate through unobstructed space without encountering obstacles or interference. This model is typically applicable when the distance between the transmitter and receiver is relatively small, below a certain threshold distance (d_0). According to the free space model, the received signal power diminishes in proportion to the square of the distance, as the inverse square law describes.

On the other hand, the multipath fading model considers the effects of signal reflections, diffractions, and scattering caused by obstacles in the propagation environment. It is used for longer distances where the signal experiences multiple paths due to reflections and diffractions. The multipath fading model considers these multiple signal paths' constructive and destructive interference, resulting in fluctuations in the received signal power. By using these models based on the transmitter and receiver separation, wireless communication systems can better understand and adapt to the characteristics of the propagation environment, leading to improved performance and reliability.

The energy consumed for transmission (E_{Tx}) using the above said radio model can be calculated using the following equation:

$$E_{Tx}(d) = E_{elec}(k) + E_{amp} * k * d \quad (7.3)$$

where E_{elec} is the energy dissipation of the radio electronics to run the transmitter and receiver circuitry. It is a device-specific parameter and is set to 50 nJ/bit [1]. E_{amp} is the transmit amplification energy, k is the packet size in bits, and d is the sender and receiver separation.

It's important to note that equation (7.4) only accounts for the energy consumption of the receiver circuitry, as the receiver does not require amplification energy for reception.

$$E_{Rx}(k) = E_{elec} * k \quad (7.4)$$

The first-order radio communication model is a simplified representation of the energy consumption in WSNs and does not capture all the complexities of real-world wireless communication. However, it provides a useful approximation for estimating energy consumption in WSNs.

7.4.2 Random waypoint mobility model

This mobility model is an extensively employed model for simulating node movement in WSNs. It is widely recognized and adopted due to its ability to provide a realistic depiction of the mobility patterns of nodes within the network.

In the random waypoint model [284], each node in the network follows a random movement pattern. The nodes have a predefined simulation area or region where they can move. The model assumes that nodes pause for a fixed duration at specific locations and then select a random destination and speed to move toward. Once they reach the destination, they pause again and repeat the process for the duration of the simulation as shown in Figure 7.2.

The key parameters in the random waypoint model include:

- Pause Time (T_p): The duration for which a node remains stationary at a particular location before selecting a new destination.
- Minimum Velocity (V_{min}) and Maximum Velocity (V_{max}): The range of speeds at which nodes can move between different locations in the simulation area.
- Direction Range: The possible range of directions in which nodes can move. Typically represented as an angle or a range of angles (e.g., between 0 and 2π).

These parameters govern the movement behavior of nodes in the random waypoint model, allowing researchers to study various aspects of WSNs, such as network connectivity, routing protocols, and energy consumption.

By using the random waypoint mobility model, researchers can evaluate the performance of WSNs under realistic node movement scenarios and develop strategies to optimize network protocols and algorithms to better adapt to dynamic environments.

In the context of our proposed work, we establish several assumptions concerning the random waypoint mobility model. These assumptions form the basis for our research and guide our approach in simulating node movement:

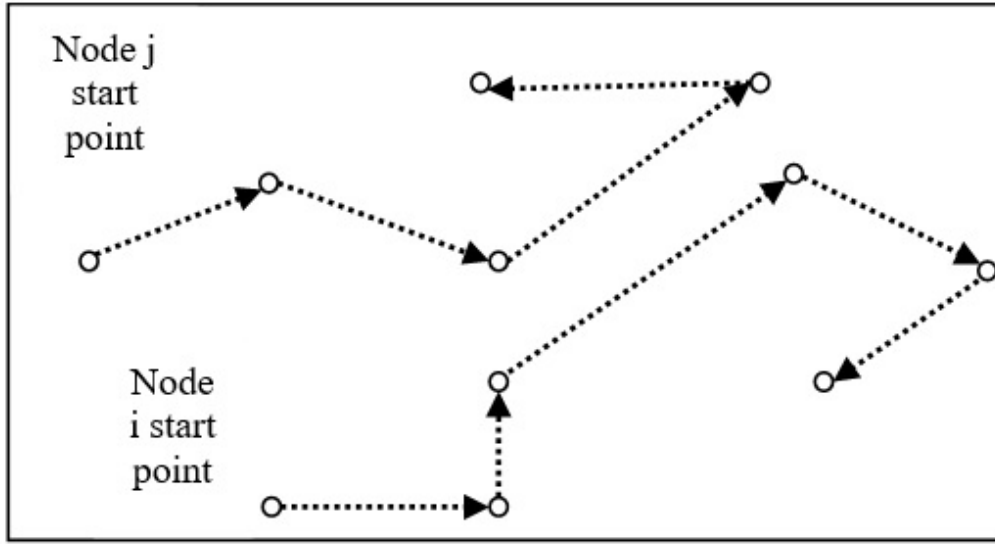


Figure 7.2: Random waypoint model

- The simulation area is two-dimensional.
- P_s represents the percentage of static nodes in the network.
- T_p denotes the pause time for each node.
- V_{min} and V_{max} correspond to the minimum and maximum velocities of each individual node, respectively. These parameters determine the range within which the velocities of the nodes can vary during simulation.
- The direction of each node lies within the range of 0 to 2π , covering a full circle of possible directions.

7.4.3 System assumptions

The proposed clustering protocol assumes the following:

- Homogeneous and Mobile Sensors: The sensors in the network are homogeneous, meaning they have the same physical characteristics and capabilities. Additionally, these sensors are mobile and capable of movement within the network.
- Location and Velocity Awareness: Each sensor in the network has an awareness of its location and velocity. This information is crucial for making informed decisions during the clustering process.
- Time Synchronization: Sensor nodes in the network are assumed to be time synchronized with each other. Time synchronization ensures coordinated operations and facilitates efficient communication among the sensors.

- **Transmission Time Calculation:** Each sensor can calculate the time it takes for data transmission. This information is utilized during the clustering process and other communication activities within the network.
- **Single Stationary Sink:** A single sink node is deployed at the center of the network, and it remains stationary. The sink acts as a central point of data collection and aggregation for the sensor nodes.

These assumptions provide a foundation for the proposed clustering protocol, enabling efficient and coordinated operation of the mobile sensor network.

7.4.4 Protocol description

Existing protocols such as M-LEACH and CBR [285] consider residual energy for CH election to maximize network lifetime. However, these protocols select CHs based on the minimum distance between nodes, which can lead to packet loss when mobile nodes move out of the cluster. Join request messages are used to mitigate this issue, but they consume additional energy and introduce control overhead.

To overcome the challenges mentioned, our proposed protocol, EARNRP, focuses on establishing stable links between CHs and their member nodes. The protocol aims to improve reliable packet delivery rates by actively searching for stable connections between non-CH nodes and CHs. During the steady state phase, sensor nodes send advertisement messages when they become disconnected from a CH, mitigating packet loss. The selection of new clusters is based on a value associated with each CH, which indicates the connection's suitability. This value helps in making informed decisions regarding cluster formation. Additionally, after the transmission process, both the CH and its members assess whether a node should remain in the cluster. If a node is deemed unsuitable, the CH removes it from the TDMA schedule and allocates the slot to an alternative node. This approach aims to improve packet delivery rates while minimizing control overhead, thereby enhancing the overall efficiency of the protocol.

7.4.5 CH election in proposed EARNRP protocol

Unlike the LEACH protocol, which overlooks factors such as remaining energy and node location during CH election, our proposed protocol takes these considerations into account. This is crucial to prevent unbalanced energy consumption and premature depletion of nodes with low residual energy. In our protocol, the CH election process incorporates multiple parameters to make informed decisions. These parameters include the count of previous CHs, the remaining energy of the nodes, the distance

between nodes and the sink, the proximity to existing CHs, node mobility, and the duration of their connection time. By considering these factors, we aim to ensure a fair and efficient selection of CHs in the network. Integrating these additional parameters enhances the CH election process, leading to a more balanced distribution of energy consumption and prolonging the overall network lifetime. By factoring in the remaining energy and node location, our proposed protocol addresses the limitations of the LEACH protocol and contributes to improved energy efficiency and longevity of the network.

The combined threshold parameter $T(\text{EARNRP}(n))$ is calculated using the formula:

$$T_{\text{EARNRP}(n)} = \begin{cases} Z(n) & \text{if } E(n) \geq \frac{1}{N} \sum_{i=1}^N E(i) \\ 0 & \text{if } E(n) < \frac{1}{N} \sum_{i=1}^N E(i) \end{cases} \quad (7.5)$$

where,

$$Z(n) = a_1 T_1(n) + a_2 T_2(n) + a_3 T_3(n) + a_4 T_4(n) + a_5 T_5(n) \quad (7.6)$$

is the weighted sum of five sub-threshold parameters, and the weights (a_1 , a_2 , a_3 , a_4 , and a_5) are assigned to balance their contributions. The five sub-thresholds are calculated as follows:

$$T_1(n) = \frac{V_{\max} - v_{\text{current}}}{V_{\max}} \quad (7.7)$$

$$T_2(n) = \frac{E_{\text{current}}}{E_{\text{avg}}} \quad (7.8)$$

$$T_3(n) = \frac{R_{\text{tran}} - d_{ij}}{R_{\text{tran}}} \quad (7.9)$$

$$T_4(n) = \frac{\Delta t_{ij}}{t_{\text{frame}}} \quad (7.10)$$

$$T_5(n) = \frac{N_{\text{ch}}(n)}{N \sum_{i=1}^N N_{\text{ch}}(i)} \quad (7.11)$$

where current speed of the node presented by (v_{current}) and the maximum speed of the node by ($V_{\max}$). We also take into account the average energy of the nodes (E_{avg}) and the current energy of the specific node (E_{current}). The distance between node i and cluster head j is denoted as d_{ij} ,

while R_{tran} represents the transmission range of the nodes. To prevent the factor $R_{tran} - d_{ij}$ from exceeding 1, we ensure that it remains within a valid range. Furthermore, we estimate the connection time (Δt_{ij}) between node i and CH j . This estimation helps determine the connection's duration, which influences the CH selection process. Additionally, we consider the number of rounds that a node has been a CH so far ($N_{ch(n)}$), as well as the alive nodes count in the current round (N). These parameters are instrumental in evaluating the history and status of the nodes during the CH election. By considering all of these factors, our proposed protocol ensures a comprehensive and informed CH election process. It allows for the effective selection of CHs based on factors such as node speed, energy levels, distance to the CH, connection time estimation, and historical information of CH participation.

To optimize the parameters such as a_1 , a_2 , a_3 , a_4 , and a_5 , which are bounded within the range of $[0, 1]$, we encounter an NP-hard problem. Consequently, evolutionary algorithms are considered the most suitable choice for solving such optimization problems.

In this chapter, we employ the ARNMRA algorithm to tackle this optimization problem and obtain optimal results. The fitness function used in the EARNRP protocol is based on multiple thresholds. To evaluate the performance of the proposed algorithm, a multi-objective fitness function is calculated:

Maximize:

$$Fitness = a_1 T_1(n) + a_2 T_2(n) + a_3 T_3(n) + a_4 T_4(n) + a_5 T_5(n) \quad (7.12)$$

Subject to:

$$\sum_{i=1}^5 a_i = 1 \quad (7.13)$$

In our proposed fitness function, we incorporate five fixed and equal weighting parameters, denoted as a_1 , a_2 , a_3 , a_4 , and a_5 . These parameters are utilized to adjust the relative importance of the five objective sub-threshold parameters within the fitness function.

Equations (7.7) to (7.11) are employed to calculate the specific values of these sub-threshold parameters. These equations capture the relationship between the objectives and define the sub-threshold values based on certain criteria or metrics. By incorporating the five weighting parameters and utilizing the sub-threshold equations, our fitness function comprehensively evaluates the multiple objectives being considered in the optimization problem.

7.4.6 Operation of proposed protocol: EARNRP

The proposed protocol EARNRP operates in two phases: the setup and the steady phase.

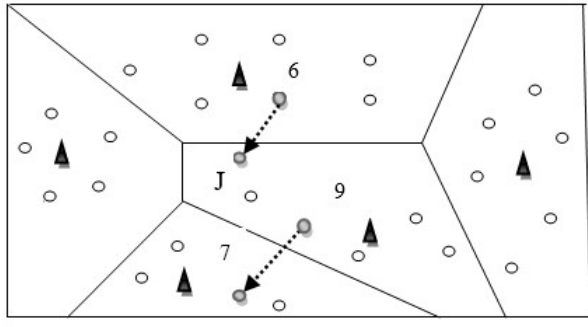
Set up phase: During this phase, each node in the network generates a random number within the range of 0 to 1. If the generated random number is less than the adaptive threshold $T_{EARNRP}(n)$, the node declares itself as a CH. The threshold $T_{EARNRP}(n)$ is calculated using equation (7.5), which determines the likelihood of a node becoming a CH. Once a CH is selected, it initiates the broadcasting of an advertisement message to the member nodes in its vicinity. This message contains relevant information such as the CH's location and velocity. The advertisement is transmitted using the CSMA/CA MAC protocol. Upon receiving the advertisement message, the member nodes assess their options and decide which cluster to join. This decision is based on evaluating the minimum distance between nodes, which is determined by the received signal strength of the advertisement message. To establish a stable link between the CH and its members and to minimize packet loss and energy consumption, a value is assigned to each CH. This value is calculated using equation (7.6), which indicates the stability of the link.

TDMA schedule creation: The process of creating a TDMA schedule involves the following steps:

- **Advertisement Message:** When a CH receives advertisement messages from nodes expressing their interest in joining the cluster, it initiates the process of creating a TDMA schedule.
- **Determining time slots:** The TDMA schedule is prepared based on the number of nodes in the cluster. time slots are allotted for data transmission.
- **Sequential Transmission:** A total of n data frames are assumed to be sent consecutively. Each node, represented by its sequence number n_i , initiates transmission at time $(n_i + k(1/(p-1)))\tau$, where k ranges from 0 to n . Here, τ represents the duration of a time slot.
- **Order of Δt_{ij} :** The TDMA schedule is organized in increasing order of Δt_{ij} , which is the time interval between the start of transmission for node i and the arrival of data at node j . This organization ensures that the constraint $(n_i + k(1/(p-1)))\tau \leq \Delta t_{ij}$ is satisfied, maximizing the number of successfully transmitted data packets within the cluster.

By following these steps, the TDMA schedule is created so that data transmission among the nodes is coordinated, allowing for efficient utilization of timeslots and maximizing the number of successfully transmitted data packets within each cluster.

Steady state phase: In the protocol, the following actions and mechanisms are in place to ensure efficient communication and avoid packet loss:



▲ CH node ● Switch CH node ○ Common node

(a) Movement of nodes for joining a new cluster

Old TDMA Scheduling					
Node 2	Node 3	Node 4	Node 6	Node 9	Node 5
New TDMA Scheduling					
Node 2	Node 3	Node 4	Node 9	Node 7	Node 5

(b) Adaptive TDMA scheduling (old and new)

Figure 7.3: For EARNRP, new adaptive TDMA scheduling creation

- **Data Packet Loss:** If the CH does not receive data packets from a sensor node, it considers them lost. As a result, the CH eliminates the corresponding member node from its TDMA schedule.
- **Join Request:** If member nodes do not receive a data request message from the CH, they send join request messages to CHs in other clusters, expressing their intention to join. This allows member nodes to find alternative clusters to transmit their data.
- **ACK Message:** Once the CH successfully receives a data packet, it sends an acknowledgment (ACK) message to the member nodes, indicating that it was received.
- **Cluster Join Request:** Upon receiving a cluster join request message from a member node, the CH sends an advertisement message to that node, similar to the setup phase. This process eliminates the need for membership declarations and reduces overhead.
- **Estimated Connection Time:** Both the CH and member nodes maintain information based on estimated connection time. They periodically check whether a node intends to stay in the cluster. If a node plans to join a new cluster, it sends a join request message to avoid potential packet loss before disconnecting from the current CH. The CH, in turn, removes the membership declaration of the node.

By implementing these actions and mechanisms, the protocol ensures that data packets are not lost due to communication failures. It allows member nodes to switch clusters if necessary and reduces unnecessary overhead by eliminating membership declarations. The use of estimated connection time helps in managing the cluster membership effectively and avoiding packet loss during transitions.

Figure 7.3 (a) illustrates the process of a node leaving its old cluster and joining a new cluster. In this scenario, Node 9 leaves Cluster J, while Node 6 joins the cluster instead. Consequently, the CH

of Cluster J modifies the TDMA schedule accordingly. Node 6 is included in the schedule, while Node 9 is removed. The adjustment of the TDMA schedule is based on the estimated connection time (Δt_{ij}), which is organized in ascending order among the member nodes and the CH node. This ordering allows for a systematic arrangement of the nodes within the schedule, ensuring efficient utilization of time slots and effective communication within the cluster.

In Figure 7.3 (b), the updated adaptive TDMA schedule is depicted after the adjustments have been made. Notably, the timeslot previously assigned to Node 9 has been replaced with the timeslot assigned to Node 6. This adaptive TDMA scheduling, facilitated by the adjustments, brings several benefits to the system. Firstly, it improves the rate of successful packet delivery, ensuring that communication between nodes within the cluster is reliable and efficient. By reassigning the timeslot to Node 6, which has joined the cluster, the scheduling enhances the utilization of the channel resources. Furthermore, adaptive scheduling ensures that the communication resources are efficiently utilized even when nodes join or leave the cluster. This seamless accommodation of node mobility and changes in cluster membership contributes to the system's overall performance. It allows for effective utilization of available resources while maintaining the desired communication quality and efficiency level.

7.4.7 Analysis of results

MATLAB is used to analyze the performance of the EARNRP. The simulations are conducted on a network comprising 100 nodes within a $100m \times 100m$ area, with the sink positioned at the center. Figure 7.4 illustrates the evolution of the count of alive nodes over multiple rounds in all the protocols, excluding the EARNRP protocol, specifically for stationary nodes. This depiction offers valuable insights into the overall behavior and stability of the network nodes over time.

In Figure 7.4, the alive nodes count is also depicted considering the presence of mobile nodes. The count of alive nodes is measured against the variation in node mobility factor (M) between 20 and 100, and the maximum fixed speed (FS) is set to 2m/s. Here, the EARNRP with all stationary nodes are designated as EARNRP-FS0-M0 (i.e. EARNRP with fixed speed of 0m/s and mobility factor 0). Similarly, EARNRP-FS2-M100 represents EARNRP with FS 2m/s and mobility factor 100. This analysis aims to estimate the network's lifetime by observing the alive nodes count in each round. In Figure 7.5, we can observe the average remaining energy of the network during the communication rounds. This figure indicates the consumption of overall energy as compared to other competitive methods. Moreover, EARNRP exhibits a consistent and steady energy absorption pattern in each communication round.

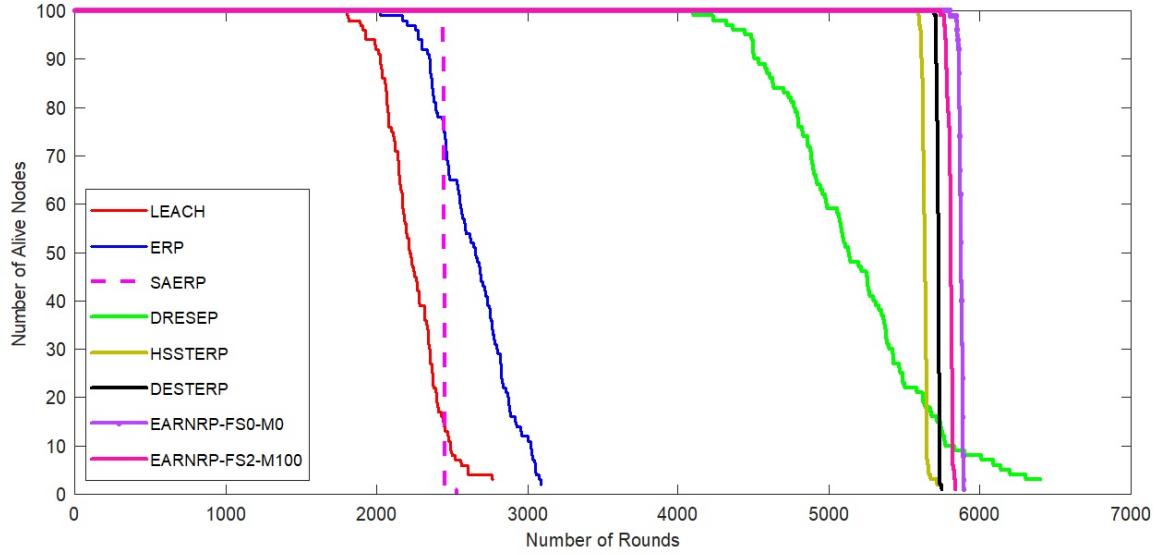


Figure 7.4: Number of alive nodes as a function of communication rounds for EARNRP in the presence and absence of mobility.

To evaluate the performance of the EARNRP protocol against other protocols, several metrics are considered: the first node dies (FND), the half node dies (HND), and the last node dies (LND) as shown in Table 7.3. These metrics provide insights into the resilience and longevity of the network under different scenarios.

The research work encompasses three cases for simulations:

Case 1: A fixed speed is assumed while varying node densities. The speed is set to 2m/s, and simulations are conducted with a dynamic mobility rate ranging from 0 to 100, as shown in Figure 7.6. The node count is varied, and its effect on the network's lifetime is analyzed. The simulations carried out in this scenario considered a range of 0 to 100 mobile nodes. The observation reveals that as the mobile nodes count increases, the network lifetime decreases because of the presence of node mobility.

Case 2: The dynamic speeds are assumed while maintaining fixed mobility, as illustrated in Figure 7.7. The simulations are conducted with fixed mobility ($M=50$) by varying the velocity of nodes from 2m/s to 20m/s. The analysis focuses on the impact of increasing node velocity on the network's lifetime.

Case 3: The third case assumes fixed mobility ($M=100$) with speed variation, as shown in Figure 7.8. The simulations are conducted with fixed mobility while varying the node count. This scenario considers 100% fixed mobility, and the results are plotted by varying the speed from 2m/s to 20m/s.

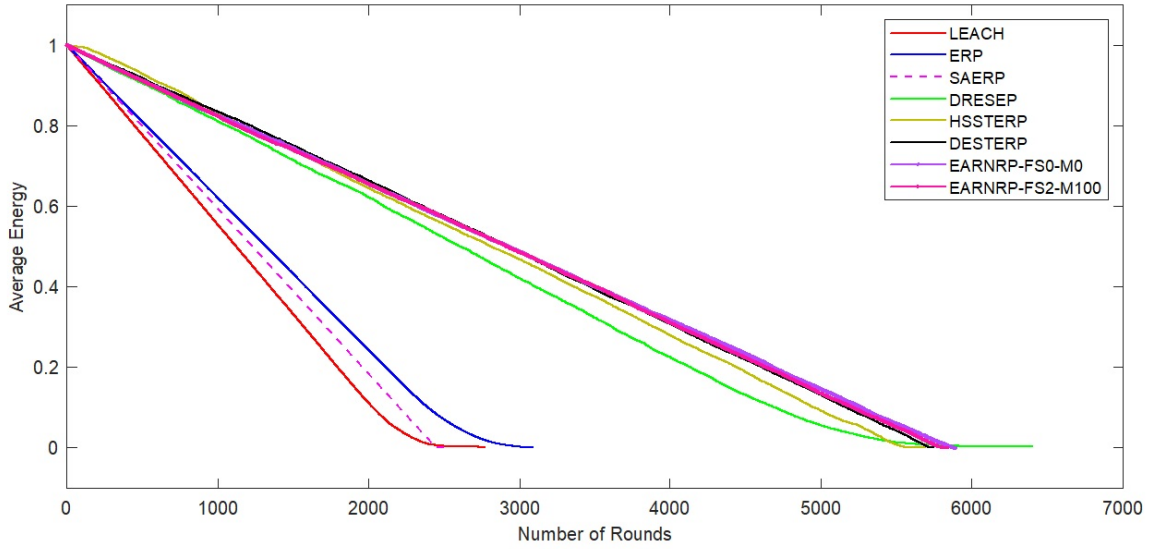


Figure 7.5: Average energy as a function of communication rounds for EARNRP in the presence and absence of mobility.

These three cases comprehensively evaluate the EARNRP protocol under different scenarios, considering node densities, speeds, and mobility factors. The simulations enable us to assess the network's performance, lifetime, and stability in various settings.

In the simulations, the performance of the proposed algorithm is evaluated by considering various factors, including the network lifetime. The network lifetime is an important metric that reflects the longevity of the network and its ability to function effectively over time. One of the factors that impact the network lifetime is the percentage of mobile nodes in the network. As the mobile nodes ,count increases, the network becomes more dynamic, which can introduce challenges such as node mobility, connectivity, and energy consumption. The proposed algorithm aims to optimize the network's performance in the presence of mobile nodes, ensuring efficient resource allocation and minimizing energy consumption to extend the network's lifetime. Another factor that affects the network lifetime is the speed of the nodes. Higher speeds can result in increased collisions among nodes, leading to network congestion. This congestion can have a negative impact on the network's performance, reducing its lifetime. The proposed algorithm takes into account the node speeds and dynamically adjusts the network parameters to mitigate congestion and optimize resource allocation, thus improving the network's lifetime. By conducting simulations and analyzing the network lifetime under different scenarios, the performance of the proposed algorithm has been assessed. The results of these simulations provide insights into the algorithm's effectiveness in addressing challenges related to mobile nodes, node mobility, and network congestion, ultimately contributing to the improvement of overall network performance and longevity.

Table 7.3: Network lifespan comparison for homogeneous configuration with $E_0=1J$ in terms of dead nodes round history

Dead Nodes (%)	LEACH	ERP	DRESEP	SAERP	HSSTERP	DESTERP	EARNRP- FS0-M0	EARNRP- FS2-M100
1 (FND)	1805.1	2113.2	4101.9	2437.8	5635.8	5699.2	5742.9	5720.6
10	2020.5	2275.8	4503.7	2444.2	5654.6	5711.6	5779.1	5726.3
20	2067.8	2364.9	4770.2	2445.4	5667.1	5719.1	5795.5	5727.8
30	2140.5	2437.6	4880.5	2447.6	5670.5	5721.8	5805.3	5729.2
40	2170.6	2510.3	4984.1	2448.3	5676.3	5722.9	5810.2	5734.4
50 (HND)	2213.8	2580.5	5127.0	2449.9	5680.2	5726.3	5812.9	5762.3
60	2281.4	2651.2	5293.8	2451.1	5681.9	5727.1	5815.4	5780.1
70	2345.8	2745.3	5396.3	2451.8	5688.2	5728.9	5816.3	5784.2
80	2393.5	2836.9	5620.9	2452.2	5690.4	5733.1	5820.2	5787.8
90	2484.8	2983.8	5771.5	2453.6	5692.8	5735.9	5823.3	5788.3
100 (LND)	2763.8	3305.7	6400.8	2454.7	5715.9	5738.6	5834.7	5789.6

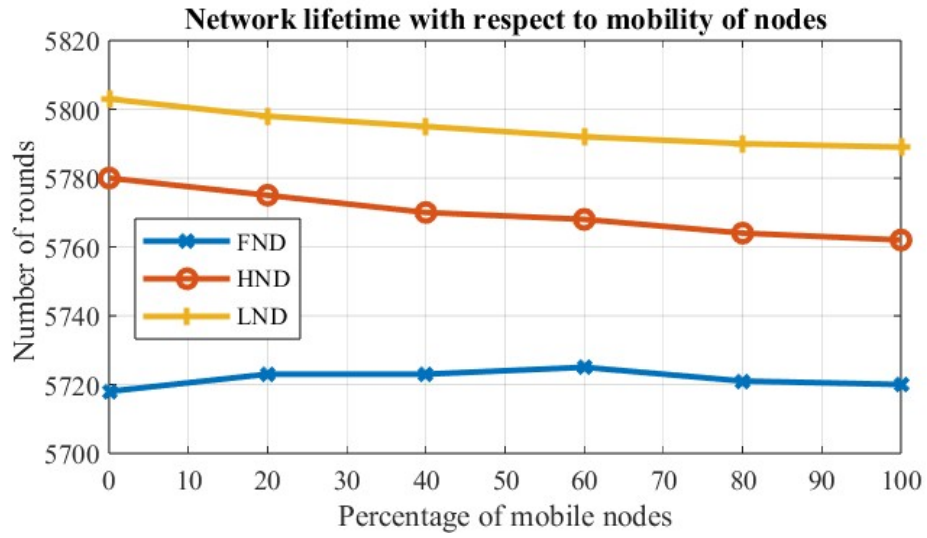


Figure 7.6: Network lifetime vs Percentage of mobile nodes for varying speed

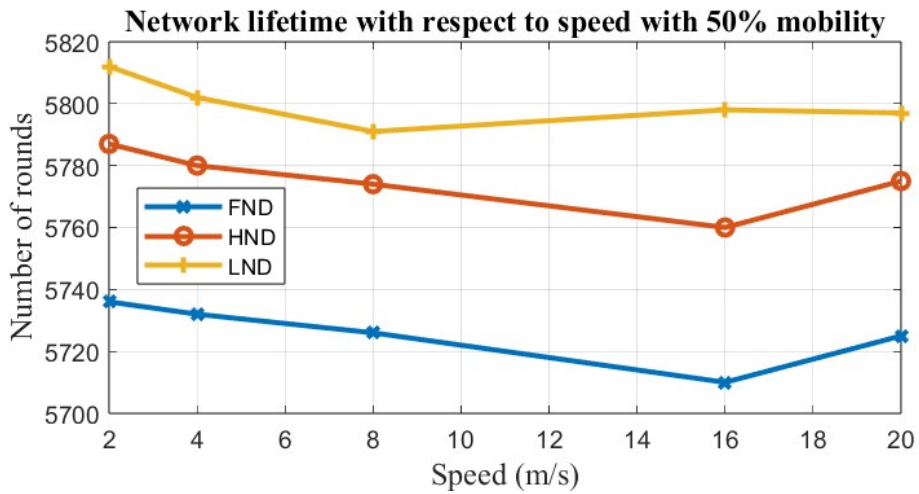


Figure 7.7: Network lifetime vs Speed (m/s) for 50% node mobility.

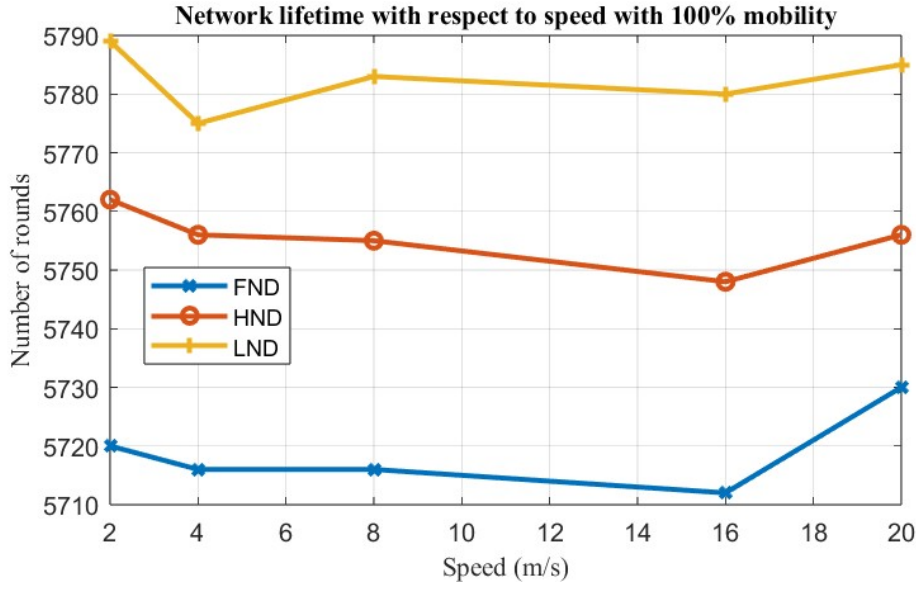


Figure 7.8: Network lifetime vs Speed (m/s) for 100% node mobility.

7.5 Summary

This study develops ARNMRA as an enhancement to the original NMRA. The primary issue with NMRA is its susceptibility to premature convergence and getting trapped in local optima. To overcome this problem, the ARNMRA incorporates an attraction and repulsion strategy and self-adaptation of the mating factor. The algorithm's effectiveness is evaluated on benchmark problems from the CEC 2005 and CEC 2019 test suites, and its performance is compared to other algorithms including SHADE, OB-L-EO, SOGWO, IWOA, and NMRA. Furthermore, a mobility-based EARNRP for WSNs with mobile nodes is proposed in this chapter. Here, simulations are conducted to evaluate the performance of the proposed algorithm and protocol. One key metric analyzed is the network lifetime, which is influenced by factors like the percentage of mobile nodes and varying speeds. Increased mobility and speed can lead to more frequent node collisions, resulting in network congestion. The impact of this congestion on network lifetime and overall performance is investigated. By studying the network lifetime under different mobility and speed conditions, the research provides insights into the effectiveness of the proposed protocol in dynamic network environments. The findings emphasize the importance of considering mobility and speed factors in designing routing protocols for WSNs with mobile nodes, as they directly impact network performance and longevity.

Chapter 8

Conclusion and Future Work

This chapter presents concluding remarks regarding the various algorithms discussed in the literature. It includes discussions on the enhancements made to each algorithm and outlines the research direction to address existing algorithms' shortcomings. This chapter also delves into the significant limitations and the need for hybridization, mutations, adaptivity, and other techniques while developing innovative algorithms. Furthermore, it explores potential method for future research, considering the applicability and modification of these approaches.

8.1 Conclusion

Over the past two decades, nature-inspired algorithms have played a dominant role in the field of optimization research in engineering and science. These algorithms are designed to provide optimal solutions when conventional mathematical models prove inadequate. Many real-world problems fall into the category of optimization problems due to their inherent complexity, multi-objective functions, computational demands, scalability issues, high dimensionality, and numerous local optimal solutions. Addressing such problems has necessitated the development of more efficient algorithms capable of delivering high-quality solutions. To tackle these problems, researchers have developed various nature-inspired algorithms, including the CS, DE, SSA, PSO, MPA, NMRA, SMA, WOA, and COOT optimization.

This thesis covers powerful nature-inspired algorithms, namely NMRA, SMA, WOA, and COOT optimization. The primary objective is to propose novel hybrid and improved variants of these algorithms that efficiently solve various optimization problems, outperforming other algorithms from the existing literature. In Chapter 3, the NMRA is enhanced to improve its exploration capabilities by employing chaos theory using ten chaotic maps. This improved CNMRA's performance has been evaluated using the CEC 2005, CEC 2014, CEC 2019 benchmark functions and three real-world engineering design problems. A Wilcoxon rank-sum test and f-rank have been conducted to assess the statistical significance of CNMRA's performance. Both the experimental and statistical analyses confirm that CNMRA surpasses all other algorithms in the comparison. After the CEC benchmark functions, CNMRA has been successfully applied to WSN, ensuring an extended network lifetime and an energy-efficient solution.

Chapter 4 deals with a novel self-adaptive hybrid SM-NMRA based on the hybridization of SMA and NMRA and also adds self-adaptive properties to NMRA's parameters. This algorithm also includes the stagnation phase based on GWO and CS algorithms to minimize the algorithm's trapping in local optimal solutions. The performance of the proposed SM-NMRA algorithm has been assessed through CEC 2005 and CEC 2014 numerical test problems and found that SM-NMRA outperforms as compared to other state-of-the-art algorithms. Non-parametric statistical tests, p-rank and f-rank tests have been employed to establish the superiority of the SM-NMRA algorithm. After this, the SM-NMRA has been applied to address the CH selection problem in WSNs. This application has led to the development of the novel TESMNM RP clustering protocol. This protocol has been tested for total energy dissipation and network operational lifetime. The results of simulations demonstrate that TESMNM RP surpasses existing clustering techniques in terms of network lifespan and average energy dissipation.

A new hybrid algorithm WONMRA has been developed in Chapter 5, which combines the exploitation capabilities of NMRA with the exploration properties of WOA. Additionally, self-adaptivity is incorporated into the WONMRA through a simulated annealing-based mutation strategy. This algorithm was subjected to CEC 2019 benchmark problems for testing its working efficiency and demonstrated superior performance compared to other algorithms. We also evaluated this algorithm for clustering in WSNs. The results clearly indicate that the WONMRA is highly effective for CH selection in WSNs.

Chapter 6 deals with a self-adaptive hybrid optimization method called CONMRA, achieved by hybridizing the recently developed NMRA with the COOT optimization. The effectiveness of CONMRA has been evaluated using challenging 100-digit (CEC 2019) benchmark problems and found that CONMRA outperforms most of the other algorithms considered for these problems. Here, we propose the CNSTERP clustering protocol using the CONMRA, which efficiently combines input factors to make informed decisions regarding CH selection. CNSTERP is evaluated in both homogeneous and heterogeneous network setups for assessing overall energy consumption and network lifetime in WSNs.

A novel attraction and repulsion-based NMRA has been proposed for energy-efficient mobile WSNs in Chapter 7. The algorithm's effectiveness is evaluated using benchmark problems for CEC 2005 and CEC 2019 test suites. After this, a mobility-based protocol called EARNRP for WSNs has been introduced. Here, the critical metric under consideration is network lifetime, which is influenced by factors such as the percentage of mobile nodes and their varying speeds. Increased mobility and higher speeds can lead to more frequent node collisions, potentially causing network congestion. The

chapter investigates the impact of this congestion on network lifetime and overall performance. By analyzing network lifetime across various mobility and speed scenarios, the research provides valuable insights into the effectiveness of the proposed protocol in dynamic network environments.

Overall, a total of five new algorithms have been introduced in this thesis. Each of these proposed algorithms incorporates adaptive parametric settings, hybridization concepts, and improved exploration and exploitation capabilities. The primary objective of all these proposed algorithms is to address the limitations present in existing algorithms. Experimental, statistical, and real-world scenario-based assessments demonstrate the effectiveness of these proposed algorithms in generating high-quality solutions compared to other algorithms found in the literature.

8.2 Future Work

This thesis has introduced several enhanced and hybrid algorithms to tackle various numerical and real-world optimization problems. As part of future work, there is potential to further improve these algorithms by incorporating advanced mutation and adaptive strategies to address upcoming research challenges. In addition to this, there is room for more work in specific challenging conditions. This includes optimization problems related to various diverse fields, including dynamic and stochastic optimization, where problem dimensions change over time. Addressing multi-objective optimization is another challenging task, aiming to simultaneously optimize two or more conflicting objectives. Multi-modal and large-scale optimization, characterized by many global optima and variable dimension sizes on the order of thousands, offer additional research opportunities. Memetic algorithms can also be explored, combining various techniques to improve algorithm performance. Additionally, parameter tuning and parametric adaptations should be considered for algorithms concerning specific problem characteristics during the search process. It is essential to consider these points when conducting further investigations before proposing new algorithms.

It is recommended to evaluate algorithms using highly challenging datasets, such as CEC 2021, CEC 2022, and CEC 2023, rather than relying solely on classical benchmarks. This comparison **should** encompass high-dimensional, multi-modal, hybrid, and composite functions from the mentioned datasets. Apart from these benchmarks, apply the algorithms to a wide range of optimization problems, including feature selection, image segmentation, economic load dispatch problems, antenna design, and various others.

List of Publications

1. S. Singh, U.Singh, "The effect of chaotic mapping on naked mole-rat algorithm for energy-efficient smart city wireless sensor network," *Computers & Industrial Engineering*, vol.173, 108655, 2022. (Impact Factor: 7.9).
DOI: <https://doi.org/10.1016/j.cie.2022.108655>
2. S. Singh, U.Singh, "A novel self-adaptive hybrid slime mould naked mole-rat algorithm for numerical optimization and energy-efficient wireless sensor network," *Concurrency and Computation: Practice and Experience*, e7809, 2023. (Impact Factor: 2.0).
DOI: <https://doi.org/10.1002/cpe.7809>
3. S. Singh, U.Singh, R. Salgotra, N. Mittal, "A novel whale optimization based naked mole-rat algorithm for energy-efficient wireless sensor network," *IEEE China Communications*, 2023. (Impact Factor: 4.1).
DOI: <https://doi.org/10.23919/JCC.ja.2021-0895>
4. S. Singh, U.Singh, N. Mittal, F. Gared, "A self-adaptive attraction and repulsion-based naked mole-rat algorithm for energy-efficient mobile wireless sensor networks," *Scientific Reports*, 2024. (Impact Factor: 4.6).
DOI: <https://doi.org/10.1038/s41598-024-51218-0>
5. S. Singh, U.Singh, "An energy-efficient clustering protocol using novel self-adaptive hybrid COOT optimization based naked mole-rat algorithm for WSN," *Cybernetics and Systems*, 2023. (Impact Factor: 1.7) (Revision Submitted).
6. S. Singh, N. Mittal, A. Nayyar, U.Singh, S.Singh, "A hybrid transient search naked mole-rat optimizer for image segmentation using multilevel thresholding," *Expert Systems with Applications*, vol.213, 119021, 2023. (Impact Factor: 8.5).
DOI: <https://doi.org/10.1016/j.eswa.2022.119021>
7. N. Mittal, S. Singh, A. Nayyar, U.Singh, "Hybrid sooty tern naked mole-rat algorithm and Fuzzy Type-2 logic-based trust and energy-aware stable clustering protocol," *Expert Systems with Applications*, vol.219, 119706, 2023. (Impact Factor: 8.5).
DOI: <https://doi.org/10.1016/j.eswa.2023.119706>
8. S. Singh, N. Mittal, U.Singh, R.Salgotra, A. Zaguia, D. Singh, "A Novel Hybrid Tunicate Swarm Naked Mole-Rat Algorithm for Image Segmentation and Numerical Optimization," *Computers*,

Materials & Continua, vol.71, no.2, 2022. (Impact Factor: 3.1).

DOI: <https://10.32604/cmc.2022.023004>

References

- [1] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (abc) algorithm,” *Applied soft computing*, vol. 8, no. 1, pp. 687–697, 2008.
- [2] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.
- [3] A. Draa, “On the performances of the flower pollination algorithm—qualitative and quantitative analyses,” *Applied Soft Computing*, vol. 34, pp. 349–371, 2015.
- [4] R. Salgotra and U. Singh, “The naked mole-rat algorithm,” *Neural Computing and Applications*, vol. 31, pp. 8837–8857, 2019.
- [5] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, “Slime mould algorithm: A new method for stochastic optimization,” *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.
- [6] I. Naruei and F. Keynia, “A new optimization method based on coot bird natural life model,” *Expert Systems with Applications*, vol. 183, p. 115 352, 2021.
- [7] P. N. Suganthan, N. Hansen, J. J. Liang, *et al.*, “Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization,” *KanGAL report*, vol. 20050005, no. 2005, p. 2005, 2005.
- [8] J. J. Liang, B. Y. Qu, and P. N. Suganthan, “Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization,” *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, vol. 635, no. 2, 2013.
- [9] K. Price, N. Awad, M. Ali, and P. Suganthan, “Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization,” in *Technical report*, Nanyang Technological University Singapore, 2018.
- [10] S. Mirjalili and S. Mirjalili, “Genetic algorithm,” *Evolutionary Algorithms and Neural Networks: Theory and Applications*, pp. 43–55, 2019.
- [11] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-international conference on neural networks*, IEEE, vol. 4, 1995, pp. 1942–1948.
- [12] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of global optimization*, vol. 11, pp. 341–359, 1997.

- [13] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *2009 World congress on nature & biologically inspired computing (NaBIC)*, Ieee, 2009, pp. 210–214.
- [14] A. P. Engelbrecht, *Computational intelligence: an introduction*. John Wiley & Sons, 2007.
- [15] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [16] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary computation*, vol. 15, no. 1, pp. 1–28, 2007.
- [17] D. Simon, "Biogeography-based optimization," *IEEE transactions on evolutionary computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [18] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," *Applied Mathematics and computation*, vol. 219, no. 15, pp. 8121–8144, 2013.
- [19] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [20] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied mathematics and computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [21] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International journal of bio-inspired computation*, vol. 2, no. 2, pp. 78–84, 2010.
- [22] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in nonlinear science and numerical simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.
- [23] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.
- [24] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Advances in engineering software*, vol. 114, pp. 163–191, 2017.
- [25] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.
- [26] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [27] M.-H. Tayarani-N and M.-R. Akbarzadeh-T, "Magnetic-inspired optimization algorithms: Operators and structures," *Swarm and Evolutionary Computation*, vol. 19, pp. 82–101, 2014.
- [28] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Gsa: A gravitational search algorithm," *Information sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

- [29] A. Y. Lam and V. O. Li, "Chemical reaction optimization for cognitive radio spectrum allocation," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, IEEE, 2010, pp. 1–5.
- [30] O. K. Erol and I. Eksin, "A new optimization method: Big bang–big crunch," *Advances in engineering software*, vol. 37, no. 2, pp. 106–111, 2006.
- [31] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [32] H. Shah-Hosseini, "Optimization with the nature-inspired intelligent water drops algorithm," *Evolutionary computation*, vol. 57, no. 2, pp. 297–320, 2009.
- [33] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: Charged system search," *Acta mechanica*, vol. 213, no. 3-4, pp. 267–289, 2010.
- [34] S. S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2019.
- [35] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Progressive 3-d coding of hyperspectral images based on jpeg 2000," *IEEE Geoscience and remote sensing letters*, vol. 3, no. 1, pp. 125–129, 2006.
- [36] P. Singh and S. Agrawal, "Node localization in wireless sensor networks using the m5p tree and smoreg algorithms," in *2013 5th International Conference and Computational Intelligence and Communication Networks*, IEEE, 2013, pp. 104–104.
- [37] M. M. Afsar and M.-H. Tayarani-N, "Clustering in sensor networks: A literature survey," *Journal of Network and Computer applications*, vol. 46, pp. 198–226, 2014.
- [38] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [39] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [40] L. Kleinrock and F. Kamoun, "Hierarchical routing for large networks performance evaluation and optimization," *Computer Networks (1976)*, vol. 1, no. 3, pp. 155–174, 1977.
- [41] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: A survey," *IEEE Communications surveys & tutorials*, vol. 15, no. 2, pp. 551–591, 2012.
- [42] A. A. Bara'a and E. A. Khalil, "A new evolutionary based routing protocol for clustered heterogeneous wireless sensor networks," *Applied Soft Computing*, vol. 12, no. 7, pp. 1950–1957, 2012.

- [43] P. Kuila, S. K. Gupta, and P. K. Jana, "A novel evolutionary approach for load balanced clustering problem for wireless sensor networks," *Swarm and Evolutionary Computation*, vol. 12, pp. 48–56, 2013.
- [44] X. Lai, Q. Liu, X. Wei, W. Wang, G. Zhou, and G. Han, "A survey of body sensor networks," *Sensors*, vol. 13, no. 5, pp. 5406–5447, 2013.
- [45] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.
- [46] G. Jayakumar, G. Ganapathi, *et al.*, "Reference point group mobility and random waypoint models in performance evaluation of manet routing protocols," *Journal of Computer Networks and Communications*, vol. 2008, 2008.
- [47] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2008.
- [48] J. Xin, G. Chen, and Y. Hai, "A particle swarm optimizer with multi-stage linearly-decreasing inertia weight," in *2009 International joint conference on computational sciences and optimization*, IEEE, vol. 1, 2009, pp. 505–508.
- [49] P. E. Gill and W. Murray, "Quasi-newton methods for unconstrained optimization," *IMA Journal of Applied Mathematics*, vol. 9, no. 1, pp. 91–108, 1972.
- [50] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & operations research*, vol. 13, no. 5, pp. 533–549, 1986.
- [51] H. Schramm and J. Zowe, "A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results," *SIAM journal on optimization*, vol. 2, no. 1, pp. 121–152, 1992.
- [52] W. J. Gutjahr, "Convergence analysis of metaheuristics," in *Matheuristics: hybridizing metaheuristics and mathematical programming*, Springer, 2009, pp. 159–187.
- [53] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.
- [54] G. Üçoluk, "Genetic algorithm solution of the tsp avoiding special crossover and mutation," *Intelligent Automation & Soft Computing*, vol. 8, no. 3, pp. 265–272, 2002.
- [55] K. A. De Jong and W. M. Spears, "A formal analysis of the role of multi-point crossover in genetic algorithms," *Annals of mathematics and Artificial intelligence*, vol. 5, pp. 1–26, 1992.
- [56] F. C. C. Mei, S. Phon-Amnuaisuk, M. Y. Alias, and P. W. Leong, "Adaptive ga: An essential ingredient in high-level synthesis," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, IEEE, 2008, pp. 3837–3844.

- [57] M. Godinho Filho, C. F. Barco, and R. F. Tavares Neto, "Using genetic algorithms to solve scheduling problems on flexible manufacturing systems (fms): A literature survey, classification and analysis," *Flexible Services and Manufacturing Journal*, vol. 26, pp. 408–431, 2014.
- [58] J. Sachdeva, V. Kumar, I. Gupta, N. Khandelwal, and C. K. Ahuja, "Multiclass brain tumor classification using ga-svm," in *2011 Developments in E-systems Engineering*, IEEE, 2011, pp. 182–187.
- [59] A. Khuntia, B. Choudhury, B. Biswal, and K. Dash, "A heuristics based multi-robot task allocation," in *2011 IEEE Recent Advances in Intelligent Computational Systems*, IEEE, 2011, pp. 407–410.
- [60] C. C. Kang, Y. J. Chuang, K. C. Tung, *et al.*, "A genetic algorithm-based boolean delay model of intracellular signal transduction in inflammation," *BMC bioinformatics*, vol. 12, pp. 1–8, 2011.
- [61] L. Foschini and M. Tortonesi, "Adaptive and business-driven service placement in federated cloud computing environments," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, IEEE, 2013, pp. 1245–1251.
- [62] H.-h. Fu, Z.-j. Li, G.-w. Li, X.-t. Jin, and P.-h. Zhu, "Modeling and controlling of engineering ship based on genetic algorithm," in *2012 Proceedings of International Conference on Modelling, Identification and Control*, IEEE, 2012, pp. 394–398.
- [63] X. Jing, Y. Liu, and W. Cao, "A hybrid genetic algorithm for route optimization in multimodal transport," in *2012 Fifth International Symposium on Computational Intelligence and Design*, IEEE, vol. 1, 2012, pp. 261–264.
- [64] C. R. Reeves, "Genetic algorithms," *Handbook of metaheuristics*, pp. 109–139, 2010.
- [65] K. V. Price, "Differential evolution," in *Handbook of optimization: From classical to modern approach*, Springer, 2013, pp. 187–214.
- [66] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE transactions on evolutionary computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [67] J. Brest and M. Sepesy Maučec, "Population size reduction for the differential evolution algorithm," *Applied Intelligence*, vol. 29, pp. 228–247, 2008.
- [68] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 99–119, 2011.
- [69] J. Zhang and A. C. Sanderson, "Jade: Adaptive differential evolution with optional external archive," *IEEE Transactions on evolutionary computation*, vol. 13, no. 5, pp. 945–958, 2009.

- [70] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, “Differential evolution using a neighborhood-based mutation operator,” *IEEE transactions on evolutionary computation*, vol. 13, no. 3, pp. 526–553, 2009.
- [71] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, “Opposition-based differential evolution,” *IEEE Transactions on Evolutionary computation*, vol. 12, no. 1, pp. 64–79, 2008.
- [72] M. Jamalipour, R. Sayareh, M. Gharib, F. Khoshahval, and M. R. Karimi, “Quantum behaved particle swarm optimization with differential mutation operator applied to wwer-1000 in-core fuel management optimization,” *Annals of Nuclear Energy*, vol. 54, pp. 134–140, 2013.
- [73] H. Zhang, D. D. Kennedy, G. P. Rangaiah, and A. Bonilla-Petriciolet, “Novel bare-bones particle swarm optimization and its performance for modeling vapor–liquid equilibrium data,” *Fluid Phase Equilibria*, vol. 301, no. 1, pp. 33–45, 2011.
- [74] Y. Dai, Y. Wei, J. Chen, Y. Zhang, and J. Ding, “Seismic wavelet estimation based on adaptive chaotic embedded particle swarm optimization algorithm,” in *2012 Fifth International Symposium on Computational Intelligence and Design*, IEEE, vol. 2, 2012, pp. 57–60.
- [75] K. Vidhya and K. Shankar Kumar, “Channel estimation of mimo–ofdm system using pso and ga,” *Arabian Journal for Science and Engineering*, vol. 39, pp. 4047–4056, 2014.
- [76] M. S. Kiran and M. Gündüz, “A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems,” *Applied Soft Computing*, vol. 13, no. 4, pp. 2188–2203, 2013.
- [77] A. Gherboudj, A. Layeb, and S. Chikhi, “Solving 0-1 knapsack problems by a discrete binary version of cuckoo search algorithm,” *International Journal of Bio-Inspired Computation*, vol. 4, no. 4, pp. 229–236, 2012.
- [78] X. Ouyang, Y. Zhou, Q. Luo, and H. Chen, “A novel discrete cuckoo search algorithm for spherical traveling salesman problem,” *Applied mathematics & information sciences*, vol. 7, no. 2, pp. 777–784, 2013.
- [79] M. Tuba, M. Subotic, and N. Stanarevic, “Modified cuckoo search algorithm for unconstrained optimization problems,” in *Proceedings of the 5th European conference on European computing conference*, 2011, pp. 263–268.
- [80] A. S. Tawfik, A. A. Badr, and I. F. Abdel-Rahman, “One rank cuckoo search algorithm with application to algorithmic trading systems optimization,” *International journal of computer applications*, vol. 64, no. 6, 2013.
- [81] Y. Zhou, H. Zheng, Q. Luo, and J. Wu, “An improved cuckoo search algorithm for solving planar graph coloring problem,” *Applied Mathematics & Information Sciences*, vol. 7, no. 2, pp. 785–792, 2013.

- [82] R. Salgotra, U. Singh, and S. Saha, "New cuckoo search algorithms with enhanced exploration and exploitation properties," *Expert Systems with Applications*, vol. 95, pp. 384–420, 2018.
- [83] J.-H. Lin, I.-H. Lee, *et al.*, "Emotional chaotic cuckoo search for the reconstruction of chaotic dynamics," *Latest advances in systems science & computational intelligence*, pp. 123–128, 2012.
- [84] E. Daniel, J. Anitha, and J. Gnanaraj, "Optimum laplacian wavelet mask based medical image using hybrid cuckoo search–grey wolf optimization algorithm," *Knowledge-Based Systems*, vol. 131, pp. 58–69, 2017.
- [85] D. Giveki, H. Salimi, G. Bahmanyar, and Y. Khademian, "Automatic detection of diabetes diagnosis using feature weighted support vector machines based on mutual information and modified cuckoo search," *arXiv preprint arXiv:1201.2173*, 2012.
- [86] M. M. Zaw and E. E. Mon, "Web document clustering using cuckoo search clustering algorithm based on levy flight," *International Journal of Innovation and Applied Studies*, vol. 4, no. 1, pp. 182–188, 2013.
- [87] V. Tiwari, "Face recognition based on cuckoo search algorithm," *image*, vol. 7, no. 8, p. 9, 2012.
- [88] A. Bhandari, V. Soni, A. Kumar, and G. K. Singh, "Cuckoo search algorithm based satellite image contrast and brightness enhancement using dwt–svd," *ISA transactions*, vol. 53, no. 4, pp. 1286–1296, 2014.
- [89] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Engineering with computers*, vol. 29, pp. 17–35, 2013.
- [90] A. Kumar and S. Chakarverty, "Design optimization for reliable embedded system using cuckoo search," in *2011 3rd international conference on electronics computer technology*, IEEE, vol. 1, 2011, pp. 264–268.
- [91] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," *Expert systems with applications*, vol. 152, p. 113 377, 2020.
- [92] D. Yousri, T. S. Babu, E. Beshr, M. B. Eteiba, and D. Allam, "A robust strategy based on marine predators algorithm for large scale photovoltaic array reconfiguration to mitigate the partial shading effect on the performance of pv system," *IEEE Access*, vol. 8, pp. 112 407–112 426, 2020.
- [93] M. Abd Elaziz, A. A. Ewees, D. Yousri, *et al.*, "An improved marine predators algorithm with fuzzy entropy for multi-level thresholding: Real world example of covid-19 ct image segmentation," *Ieee Access*, vol. 8, pp. 125 306–125 330, 2020.

- [94] H. Singh, M. Abouhawwash, N. Mittal, R. Salgotra, S. Mahajan, and A. K. Pandit, "Performance evaluation of non-uniform circular antenna array using integrated harmony search with differential evolution based naked mole rat algorithm," *Expert Systems with Applications*, vol. 189, p. 116 146, 2022.
- [95] P. Singh, N. Mittal, U. Singh, and R. Salgotra, "Naked mole-rat algorithm with improved exploration and exploitation capabilities to determine 2d and 3d coordinates of sensor nodes in wsns," *Arabian Journal for Science and Engineering*, vol. 46, no. 2, pp. 1155–1178, 2021.
- [96] G. Singh, U. Singh, and R. Salgotra, "Effect of parametric enhancements on naked mole-rat algorithm for global optimization," *Engineering with Computers*, pp. 1–29, 2021.
- [97] S. Guha, A. Das, P. K. Singh, A. Ahmadian, N. Senu, and R. Sarkar, "Hybrid feature selection method based on harmony search and naked mole-rat algorithms for spoken language identification from audio signals," *IEEE Access*, vol. 8, pp. 182 868–182 887, 2020.
- [98] S. Singh, N. Mittal, and H. Singh, "A feature level image fusion for ir and visible image using mnmra based segmentation," *Neural Computing and Applications*, pp. 1–18, 2022.
- [99] F. L. Howard, "The life history of physarum polycephalum," *American journal of botany*, pp. 116–133, 1931.
- [100] M. Abdel-Basset, V. Chang, and R. Mohamed, "Hsma_woa: A hybrid novel slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest x-ray images," *Applied Soft Computing*, vol. 95, p. 106 642, 2020.
- [101] S. L. Zubaidi, I. H. Abdulkareem, K. S. Hashim, *et al.*, "Hybridised artificial neural network model with slime mould algorithm: A novel methodology for prediction of urban stochastic water demand," *Water*, vol. 12, no. 10, p. 2692, 2020.
- [102] E. H. Houssein, M. A. Mahdy, M. J. Blondin, D. Shebl, and W. M. Mohamed, "Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems," *Expert Systems with Applications*, vol. 174, p. 114 689, 2021.
- [103] M. Abdel-Basset, R. Mohamed, R. K. Chakraborty, M. J. Ryan, and S. Mirjalili, "An efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection," *Computers & Industrial Engineering*, vol. 153, p. 107 078, 2021.
- [104] A. A. Ewees, L. Abualigah, D. Yousri, *et al.*, "Improved slime mould algorithm based on firefly algorithm for feature selection: A case study on qsar model," *Engineering with Computers*, pp. 1–15, 2021.
- [105] R. M. Rizk-Allah, A. E. Hassanien, and D. Song, "Chaos-opposition-enhanced slime mould algorithm for minimizing the cost of energy for the wind turbines on high-altitude sites," *ISA Transactions*, 2021.

- [106] J. Zhao, Z.-M. Gao, and W. Sun, "The improved slime mould algorithm with levy flight," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1617, 2020, p. 012 033.
- [107] M. K. Naik, R. Panda, and A. Abraham, "Normalized square difference based multilevel thresholding technique for multispectral images using leader slime mould algorithm," *Journal of King Saud University-Computer and Information Sciences*, 2020.
- [108] G. Vashishtha, S. Chauhan, M. Singh, and R. Kumar, "Bearing defect identification by swarm decomposition considering permutation entropy measure and opposition-based slime mould algorithm," *Measurement*, p. 109 389, 2021.
- [109] R. M. Rizk-Allah, A. E. Hassanien, and D. Song, "Chaos-opposition-enhanced slime mould algorithm for minimizing the cost of energy for the wind turbines on high-altitude sites," *ISA transactions*, vol. 121, pp. 191–205, 2022.
- [110] J. Zhao and Z.-M. Gao, "The hybridized harris hawk optimization and slime mould algorithm," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1682, 2020, p. 012 029.
- [111] Z.-M. Gao, J. Zhao, and S.-R. Li, "The improved slime mould algorithm with cosine controlling parameters," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1631, 2020, p. 012 083.
- [112] Z.-M. Gao, J. Zhao, Y. Yang, and X.-J. Tian, "The hybrid grey wolf optimization-slime mould algorithm," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1617, 2020, p. 012 034.
- [113] Z. Gao, J. Zhao, and S. Li, "The hybridized slime mould and particle swarm optimization algorithms," in *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, IEEE, 2020, pp. 304–308.
- [114] C. Kumar, T. D. Raj, M. Premkumar, and T. D. Raj, "A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters," *Optik*, vol. 223, p. 165 277, 2020.
- [115] R.-E. Precup, R.-C. David, R.-C. Roman, E. M. Petriu, and A.-I. Szedlak-Stinean, "Slime mould algorithm-based tuning of cost-effective fuzzy controllers for servo systems," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 1042–1052, 2021.
- [116] S. Tiachacht, S. Khatir, C. Le Thanh, R. V. Rao, S. Mirjalili, and M. A. Wahab, "Inverse problem for dynamic structural health monitoring based on slime mould algorithm," *Engineering with Computers*, pp. 1–24, 2021.
- [117] R. Djekidel, B. Bentouati, M. Javaid, H. Bouchekara, A. S. Bayoumi, and R. A. El-Sehiemy, "Mitigating the effects of magnetic coupling between hv transmission line and metallic pipeline

using slime mould algorithm,” *Journal of Magnetism and Magnetic Materials*, vol. 529, p. 167 865, 2021.

- [118] A. Draz, M. M. Elkholy, and A. A. El-Fergany, “Slime mould algorithm constrained by the relay operating time for optimal coordination of directional overcurrent relays using multiple standardized tripping curves,” *Neural Computing and Applications*, pp. 1–13, 2021.
- [119] D. Agarwal and P. S. Bharti, “Implementing modified swarm intelligence algorithm based on slime moulds for path planning and obstacle avoidance problem in mobile robots,” *Applied Soft Computing*, p. 107 372, 2021.
- [120] T. Gush, C.-H. Kim, S. Admasie, J.-S. Kim, and J.-S. Song, “Optimal smart inverter control for pv and bess to improve pv hosting capacity of distribution networks using slime mould algorithm,” *IEEE Access*, 2021.
- [121] S. Ekinici, D. Izci, H. L. Zeynelgil, and S. Orenc, “An application of slime mould algorithm for optimizing parameters of power system stabilizer,” in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, 2020, pp. 1–5.
- [122] A. M. Zobaa, S. H. A. Aleem, and H. K. Youssef, “Comparative analysis of double-tuned harmonic passive filter design methodologies using slime mould optimization algorithm,” in *2021 IEEE Texas Power and Energy Conference (TPEC)*, IEEE, 2021, pp. 1–6.
- [123] A. Kaveh and M. I. Ghazaan, “Enhanced whale optimization algorithm for sizing optimization of skeletal structures,” *Mechanics Based design of structures and Machines*, vol. 45, no. 3, pp. 345–362, 2017.
- [124] B. Bentouati, L. Chaib, and S. Chettih, “A hybrid whale algorithm and pattern search technique for optimal power flow problem,” in *2016 8th international conference on modelling, identification and control (ICMIC)*, IEEE, 2016, pp. 1048–1053.
- [125] H. Hu, Y. Bai, and T. Xu, “A whale optimization algorithm with inertia weight,” *WSEAS Trans. Comput*, vol. 15, no. 8, 2016.
- [126] M.-F. Horng, T.-K. Dao, C.-S. Shieh, and T.-T. Nguyen, “A multi-objective optimal vehicle fuel consumption based on whale optimization algorithm,” in *Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceeding of the Twelfth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Nov., 21-23, 2016, Kaohsiung, Taiwan, Volume 2*, Springer, 2017, pp. 371–380.
- [127] T.-K. Dao, T.-S. Pan, and J.-S. Pan, “A multi-objective optimal mobile robot path planning based on whale optimization algorithm,” in *2016 IEEE 13th international conference on signal processing (ICSP)*, IEEE, 2016, pp. 337–342.

- [128] S. Mirjalili, S. M. Mirjalili, S. Saremi, and S. Mirjalili, "Whale optimization algorithm: Theory, literature review, and application in designing photonic crystal filters," *Nature-inspired optimizers: theories, literature reviews and applications*, pp. 219–238, 2020.
- [129] Q.-T. Bui, M. V. Pham, Q.-H. Nguyen, L. X. Nguyen, and H. M. Pham, "Whale optimization algorithm and adaptive neuro-fuzzy inference system: A hybrid method for feature selection and land pattern classification," *International journal of remote sensing*, vol. 40, no. 13, pp. 5078–5093, 2019.
- [130] S. Mostafa Bozorgi and S. Yazdani, "Iwoa: An improved whale optimization algorithm for optimization problems," *Journal of Computational Design and Engineering*, vol. 6, no. 3, pp. 243–259, 2019.
- [131] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Computing*, vol. 22, pp. 1–15, 2018.
- [132] P. D. P. Reddy, V. V. Reddy, and T. G. Manohar, "Whale optimization algorithm for optimal sizing of renewable resources for loss reduction in distribution systems," *Renewables: wind, water, and solar*, vol. 4, pp. 1–13, 2017.
- [133] A. Mostafa, A. E. Hassanien, M. Houseni, and H. Hefny, "Liver segmentation in mri images based on whale optimization algorithm," *Multimedia Tools and Applications*, vol. 76, pp. 24 931–24 954, 2017.
- [134] Y. Ling, Y. Zhou, and Q. Luo, "Lévy flight trajectory-based whale optimization algorithm for global optimization," *IEEE access*, vol. 5, pp. 6168–6186, 2017.
- [135] I. N. Trivedi, M. Bhoje, R. Bhesdadiya, P. Jangir, N. Jangir, and A. Kumar, "An emission constraint environment dispatch problem solution with microgrid using whale optimization algorithm," in *2016 National Power Systems Conference (NPSC)*, IEEE, 2016, pp. 1–6.
- [136] A. E. Hassanien, M. Abd Elfattah, S. Aboulenin, G. Schaefer, S. Y. Zhu, and I. Korovin, "Historic handwritten manuscript binarisation using whale optimisation," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2016, pp. 003 842–003 846.
- [137] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Applied Soft Computing*, vol. 62, pp. 441–453, 2018.
- [138] H. M. Hasanien, "Performance improvement of photovoltaic power systems using an optimal control strategy based on whale optimization algorithm," *Electric Power Systems Research*, vol. 157, pp. 168–176, 2018.

- [139] M. A. El Aziz, A. A. Ewees, A. E. Hassanien, M. Mudhsh, and S. Xiong, "Multi-objective whale optimization algorithm for multilevel thresholding segmentation," *Advances in soft computing and machine learning in image processing*, pp. 23–39, 2018.
- [140] L.-L. Li, J. Sun, M.-L. Tseng, and Z.-G. Li, "Extreme learning machine optimized by whale optimization algorithm using insulated gate bipolar transistor module aging degree evaluation," *Expert Systems with Applications*, vol. 127, pp. 58–67, 2019.
- [141] V. Mukherjee, A. Mukherjee, and D. Prasad, "Whale optimization algorithm with wavelet mutation for the solution of optimal power flow problem," in *Handbook of research on predictive modeling and optimization methods in science and engineering*, IGI Global, 2018, pp. 500–553.
- [142] A.-Z. Ala'M, H. Faris, J. Alqatawna, and M. A. Hassonah, "Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts," *Knowledge-Based Systems*, vol. 153, pp. 91–104, 2018.
- [143] R. R. Mostafa, A. G. Hussien, M. A. Khan, S. Kadry, and F. A. Hashim, "Enhanced coot optimization algorithm for dimensionality reduction," in *2022 Fifth international conference of women in data science at prince sultan university (WiDS PSU)*, IEEE, 2022, pp. 43–48.
- [144] E. H. Houssein, F. A. Hashim, S. Ferahtia, and H. Rezk, "Battery parameter identification strategy based on modified coot optimization algorithm," *Journal of Energy Storage*, vol. 46, p. 103848, 2022.
- [145] Y. Huang, J. Zhang, W. Wei, *et al.*, "Research on coverage optimization in a wsn based on an improved coot bird algorithm," *Sensors*, vol. 22, no. 9, p. 3383, 2022.
- [146] W. Sheng, R. Li, T. Yan, M.-L. Tseng, J. Lou, and L. Li, "A hybrid dynamic economics emissions dispatch model: Distributed renewable power systems based on improved coot optimization algorithm," *Renewable Energy*, vol. 204, pp. 493–506, 2023.
- [147] A. S. Alqahtani, P. Saravanan, M. Maheswari, and S. Alshmrany, "An automatic query expansion based on hybrid cmo-coot algorithm for optimized information retrieval," *The Journal of Supercomputing*, pp. 1–19, 2022.
- [148] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd annual Hawaii international conference on system sciences*, IEEE, 2000, 10–pp.
- [149] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on wireless communications*, vol. 1, no. 4, pp. 660–670, 2002.

- [150] M. Golsorkhtabar, F. K. Nia, M. Hosseinzadeh, and Y. Vejdandparast, "The novel energy adaptive protocol for heterogeneous wireless sensor networks," in *2010 3rd International Conference on Computer Science and Information Technology*, IEEE, vol. 2, 2010, pp. 178–182.
- [151] A. Manjeshwar and D. P. Agrawal, "Teen: A routing protocol for enhanced efficiency in wireless sensor networks," in *ipdps*, vol. 1, 2001, p. 189.
- [152] G. Smaragdakis, I. Matta, A. Bestavros, *et al.*, "Sep: A stable election protocol for clustered heterogeneous wireless sensor networks," in *Second international workshop on sensor and actor network protocols and applications (SANPA 2004)*, Boston, MA; vol. 3, 2004.
- [153] F. A. Aderohunmu, J. D. Deng, and M. Purvis, "Enhancing clustering in wireless sensor networks with energy heterogeneity," *International Journal of Business Data Communications and Networking (IJBDCN)*, vol. 7, no. 4, pp. 18–31, 2011.
- [154] D. Kumar, T. C. Aseri, and R. Patel, "Eehc: Energy efficient heterogeneous clustered scheme for wireless sensor networks," *computer communications*, vol. 32, no. 4, pp. 662–667, 2009.
- [155] A. Kashaf, N. Javaid, Z. A. Khan, and I. A. Khan, "Tsep: Threshold-sensitive stable election protocol for wsns," in *2012 10th international conference on frontiers of information technology*, IEEE, 2012, pp. 164–168.
- [156] L. Qing, Q. Zhu, and M. Wang, "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks," *Computer communications*, vol. 29, no. 12, pp. 2230–2237, 2006.
- [157] F. A. Aderohunmu, J. D. Deng, and M. K. Purvis, "A deterministic energy-efficient clustering protocol for wireless sensor networks," in *2011 Seventh international Conference on intelligent sensors, sensor networks and information processing*, IEEE, 2011, pp. 341–346.
- [158] D.-S. Kim and Y.-J. Chung, "Self-organization routing protocol supporting mobile nodes for wireless sensor network," in *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*, IEEE, vol. 2, 2006, pp. 622–626.
- [159] G. S. Kumar, P. M. Vinu, and K. P. Jacob, "Mobility metric based leach-mobile protocol," in *2008 16th International conference on advanced computing and communications*, IEEE, 2008, pp. 248–253.
- [160] J. Yu, Y. Qi, G. Wang, and X. Gu, "A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution," *AEU-International Journal of Electronics and Communications*, vol. 66, no. 1, pp. 54–61, 2012.
- [161] S. Deng, J. Li, and L. Shen, "Mobility-based clustering protocol for wireless sensor networks with mobile nodes," *IET wireless sensor systems*, vol. 1, no. 1, pp. 39–47, 2011.

- [162] S. Mudundi and H. H. Ali, "A new robust genetic algorithm for dynamic cluster formation in wireless sensor networks," *Proceedings of Wireless and Optical Communications, Montreal, Quebec, Canada*, 2007.
- [163] E. A. Khalil and A. A. Bara'a, "Energy-aware evolutionary routing protocol for dynamic clustering of wireless sensor networks," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 195–203, 2011.
- [164] J. Tillett, R. Rao, and F. Sahin, "Cluster-head identification in ad hoc sensor networks using particle swarm optimization," in *2002 IEEE International Conference on Personal Wireless Communications*, IEEE, 2002, pp. 201–205.
- [165] A.-A. Salehpour, B. Mirmobin, A. Afzali-Kusha, and S. Mohammadi, "An energy efficient routing protocol for cluster-based wireless sensor networks using ant colony optimization," in *2008 International Conference on Innovations in Information Technology*, IEEE, 2008, pp. 455–459.
- [166] N. Mittal, U. Singh, and B. S. Sohi, "A novel energy efficient stable clustering approach for wireless sensor networks," *Wireless Personal Communications*, vol. 95, pp. 2947–2971, 2017.
- [167] N. Mittal, U. Singh, R. Salgotra, and B. S. Sohi, "A boolean spider monkey optimization based energy efficient clustering approach for wsns," *Wireless Networks*, vol. 24, pp. 2093–2109, 2018.
- [168] N. Mittal, U. Singh, and B. S. Sohi, "Harmony search algorithm based threshold-sensitive energy-efficient clustering protocols for wsns.," *Adhoc & Sensor Wireless Networks*, vol. 36, 2017.
- [169] R. Sharma, N. Mittal, and B. S. Sohi, "Flower pollination algorithm-based energy-efficient stable clustering approach for wsns," *International Journal of Communication Systems*, vol. 33, no. 7, e4337, 2020.
- [170] A. A. Qaffas, "Applying an improved squirrel search algorithm (issa) for clustering and low-energy routing in wireless sensor networks (wsns)," *Mobile Networks and Applications*, pp. 1–16, 2023.
- [171] I. Zelinka and J. Lampinen, "On stagnation of the differential evolution algorithm," in *Proceedings of Mendel, 6th International Mendel Conference on Soft Computing*, 2000.
- [172] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Enhanced salp swarm algorithm: Application to variable speed wind generators," *Engineering Applications of Artificial Intelligence*, vol. 80, pp. 82–96, 2019.
- [173] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

- [174] J. Lampinen, I. Zelinka, *et al.*, “On stagnation of the differential evolution algorithm,” in *Proceedings of MENDEL*, Citeseer, vol. 6, 2000, pp. 76–83.
- [175] Y. Shi and R. C. Eberhart, “Parameter selection in particle swarm optimization,” in *Evolutionary Programming VII: 7th International Conference, EP98 San Diego, California, USA, March 25–27, 1998 Proceedings* 7, Springer, 1998, pp. 591–600.
- [176] J. Jarvis, “Eusociality in a mammal: Cooperative breeding in naked mole-rat colonies,” *Science*, vol. 212, no. 4494, pp. 571–573, 1981.
- [177] R. A. Brett, “The population structure of naked mole-rat colonies,” *The biology of the naked mole-rat*, pp. 97–136, 1991.
- [178] D. Yang, G. Li, and G. Cheng, “On the efficiency of chaos optimization algorithms for global optimization,” *Chaos, Solitons & Fractals*, vol. 34, no. 4, pp. 1366–1375, 2007.
- [179] Z. Zhu, S. Li, and H. Yu, “A new approach to generalized chaos synchronization based on the stability of the error system,” *Kybernetika*, vol. 44, no. 4, pp. 492–500, 2008.
- [180] M. S. Tavazoei and M. Haeri, “Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms,” *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 1076–1085, 2007.
- [181] G. Gharooni-fard, F. Moein-darbari, H. Deldari, and A. Morvaridi, “Scheduling of scientific workflows using a chaos-genetic algorithm,” *Procedia Computer Science*, vol. 1, no. 1, pp. 1445–1454, 2010.
- [182] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, “Chaos-enhanced accelerated particle swarm optimization,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 2, pp. 327–340, 2013.
- [183] W. Gong and S. Wang, “Chaos ant colony optimization and application,” in *2009 Fourth International Conference on Internet Computing for Science and Engineering*, IEEE, 2009, pp. 301–303.
- [184] B. Alatas, “Chaotic bee colony algorithms for global numerical optimization,” *Expert systems with applications*, vol. 37, no. 8, pp. 5682–5687, 2010.
- [185] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, “Firefly algorithm with chaos,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 1, pp. 89–98, 2013.
- [186] S. Kadry, R. G. Crespo, M. El Bitar, R. Mohan, and V. Rajinikanth, “Chaotic-moth-flame-algorithm based scheme to design pid controller for benchmark avr,” in *Metaheuristics and Optimization in Computer and Electrical Engineering: Volume 2: Hybrid and Improved Algorithms*, Springer, 2023, pp. 473–491.

- [187] G. I. Sayed, G. Khoriba, and M. H. Haggag, "A novel chaotic salp swarm algorithm for global optimization and feature selection," *Applied Intelligence*, vol. 48, no. 10, pp. 3462–3481, 2018.
- [188] N. Mittal, A. Garg, P. Singh, S. Singh, and H. Singh, "Improvement in learning enthusiasm-based tlbo algorithm with enhanced exploration and exploitation properties," *Natural Computing*, pp. 1–33, 2020.
- [189] J. J. Żebrowski, "Hénon: 'deterministic chaos-an introduction'," *Mathematica Applicanda*, vol. 25, no. 39, pp. 83–86, 2016.
- [190] G. Heidari-Bateni and C. D. McGillem, "A chaotic direct-sequence spread-spectrum communication system," *IEEE Transactions on communications*, vol. 42, no. 234, pp. 1524–1527, 1994.
- [191] M. S. Tavazoei and M. Haeri, "An optimization algorithm based on chaotic behavior and fractal nature," *Journal of Computational and Applied Mathematics*, vol. 206, no. 2, pp. 1070–1081, 2007.
- [192] K.-w. Wong, K.-P. Man, S. Li, and X. Liao, "A more secure chaotic cryptographic scheme based on the dynamic look-up table," *Circuits, Systems and Signal Processing*, vol. 24, no. 5, pp. 571–584, 2005.
- [193] M. Suneel, "Chaotic sequences for secure cdma," *arXiv preprint nlin/0602018*, 2006.
- [194] P. Arena, R. Caponetto, L. Fortuna, A. Rizzo, and M. La Rosa, "Self-organization in non-recurrent complex systems," *International Journal of Bifurcation and Chaos*, vol. 10, no. 05, pp. 1115–1125, 2000.
- [195] G. Manganaro and J. P. De Gyvez, "Dna computing based on chaos," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, IEEE, 1997, pp. 255–260.
- [196] H. Gao, Y. Zhang, S. Liang, and D. Li, "A new chaotic algorithm for image encryption," *Chaos, Solitons & Fractals*, vol. 29, no. 2, pp. 393–399, 2006.
- [197] G.-G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang, "Chaotic krill herd algorithm," *Information Sciences*, vol. 274, pp. 17–34, 2014.
- [198] Q. Zhang, Z. Li, C. Zhou, and X. Wei, "Bayesian network structure learning based on the chaotic particle swarm optimization algorithm," *Genet Mol Res*, vol. 12, no. 4, pp. 4468–4479, 2013.
- [199] R. C. Hilborn *et al.*, *Chaos and nonlinear dynamics: an introduction for scientists and engineers*. Oxford University Press on Demand, 2000.

- [200] D. He, C. He, L.-G. Jiang, H.-w. Zhu, and G.-r. Hu, "Chaotic characteristics of a one-dimensional iterative map with infinite collapses," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 7, pp. 900–906, 2001.
- [201] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.
- [202] Y. Li, S. Deng, and D. Xiao, "A novel hash algorithm construction based on chaotic neural network," *Neural Computing and Applications*, vol. 20, no. 1, pp. 133–141, 2011.
- [203] A. Wolf *et al.*, "Quantifying chaos with lyapunov exponents," *Chaos*, vol. 16, pp. 285–317, 1986.
- [204] R. L. Devaney and S. Wiggins, *A review of: "an introduction to chaotic dynamical systems" (benjamin/cummings, menlo park, 1986)*, 1987.
- [205] H.-O. Peitgen, H. Jürgens, and D. Saupe, *Introduction to fractals and chaos*. Springer-Verlag, 1992, vol. 1.
- [206] E. Ott, *Chaos in dynamical systems*. Cambridge university press, 2002.
- [207] R. Salgotra, U. Singh, and S. Saha, "On some improved versions of whale optimization algorithm," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9653–9691, 2019.
- [208] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *2005 IEEE congress on evolutionary computation*, IEEE, vol. 2, 2005, pp. 1785–1791.
- [209] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *2013 IEEE congress on evolutionary computation*, IEEE, 2013, pp. 71–78.
- [210] R. Salgotra, U. Singh, and S. Sharma, "On the improvement in grey wolf optimization," *Neural Computing and Applications*, pp. 1–40, 2019.
- [211] S. Khalilpourazari and S. H. R. Pasandideh, "Sine-cosine crow search algorithm: Theory and applications," *Neural Computing and Applications*, pp. 1–18, 2019.
- [212] D. Yousri, M. Abd Elaziz, and S. Mirjalili, "Fractional-order calculus-based flower pollination algorithm with local search for global optimization and image segmentation," *Knowledge-Based Systems*, p. 105889, 2020.
- [213] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, p. 105190, 2020.
- [214] B. Nautiyal, R. Prakash, V. Vimal, G. Liang, and H. Chen, "Improved salp swarm algorithm with mutation schemes for solving global optimization and engineering problems," *Engineering with Computers*, pp. 1–23, 2021.

- [215] N. Hansen, S. D. Müller, and P. Koumoutsakos, “Reducing the time complexity of the de-randomized evolution strategy with covariance matrix adaptation (cma-es),” *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.
- [216] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, “Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems,” in *2017 IEEE Congress on evolutionary computation (CEC)*, IEEE, 2017, pp. 145–152.
- [217] S. Gupta and K. Deep, “A novel random walk grey wolf optimizer,” *Swarm and evolutionary computation*, vol. 44, pp. 101–112, 2019.
- [218] H. Ma and D. Simon, “Blended biogeography-based optimization for constrained optimization,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 3, pp. 517–525, 2011.
- [219] V. Garg and K. Deep, “Performance of laplacian biogeography-based optimization algorithm on cec 2014 continuous optimization benchmarks and camera calibration problem,” *Swarm and Evolutionary Computation*, vol. 27, pp. 132–144, 2016.
- [220] G. G. Tejani, V. J. Savsani, V. K. Patel, and S. Mirjalili, “Truss optimization with natural frequency bounds using improved symbiotic organisms search,” *Knowledge-Based Systems*, vol. 143, pp. 162–178, 2018.
- [221] S. Baluja, “Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning,” Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science, Tech. Rep., 1994.
- [222] G.-G. Wang, M. Lu, and X.-J. Zhao, “An improved bat algorithm with variable neighborhood search for global optimization,” in *2016 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2016, pp. 1773–1778.
- [223] S. Gupta, K. Deep, and A. P. Engelbrecht, “A memory guided sine cosine algorithm for global optimization,” *Engineering Applications of Artificial Intelligence*, vol. 93, p. 103718, 2020.
- [224] W. Li, G.-G. Wang, and A. H. Alavi, “Learning-based elephant herding optimization algorithm for solving numerical optimization problems,” *Knowledge-Based Systems*, p. 105675, 2020.
- [225] G.-G. Wang, S. Deb, A. H. Gandomi, Z. Zhang, and A. H. Alavi, “Chaotic cuckoo search,” *Soft Computing*, vol. 20, no. 9, pp. 3349–3362, 2016.
- [226] A. K. Bhandari, “A novel beta differential evolution algorithm-based fast multilevel thresholding for color image segmentation,” *Neural Computing and Applications*, vol. 32, no. 9, pp. 4583–4613, 2020.
- [227] D. Yousri, M. Abd Elaziz, and S. Mirjalili, “Fractional-order calculus-based flower pollination algorithm with local search for global optimization and image segmentation,” *Knowledge-Based Systems*, p. 105889, 2020.

- [228] R. Salgotra, U. Singh, S. Singh, G. Singh, and N. Mittal, "Self-adaptive salp swarm algorithm for engineering optimization problems," *Applied Mathematical Modelling*, vol. 89, pp. 188–207, 2020.
- [229] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.
- [230] X.-S. Yang, "Flower pollination algorithm for global optimization," in *International conference on unconventional computing and natural computation*, Springer, 2012, pp. 240–249.
- [231] F. Wilcoxon, S. Katti, and R. A. Wilcox, "Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test," *Selected tables in mathematical statistics*, vol. 1, pp. 171–259, 1970.
- [232] S. Trab, A. Zouinkhi, B. Boussaid, and M. N. Abdelkrim, "Application of distributed fault detection in wsn to dangerous chemical products based on bayesian approach," in *2013 International Conference on Advanced Logistics and Transport*, IEEE, 2013, pp. 129–134.
- [233] R. Anitha and P. Kamalakkannan, "Enhanced cluster based routing protocol for mobile nodes in wireless sensor network," in *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, IEEE, 2013, pp. 187–193.
- [234] W. Lu, Y. Gong, X. Liu, J. Wu, and H. Peng, "Collaborative energy and information transfer in green wireless sensor networks for smart cities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1585–1593, 2017.
- [235] A. Khalifeh, K. A. Darabkh, A. M. Khasawneh, *et al.*, "Wireless sensor networks for smart cities: Network design, implementation and performance evaluation," *Electronics*, vol. 10, no. 2, p. 218, 2021.
- [236] J. Vázquez-Castillo, A. Castillo-Atoche, J. Estrada-López, *et al.*, "Energy-saving techniques for urban noise wsn with kalman-based state estimation and green facade energy harvester," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022.
- [237] S. K. Singh, Y. Pan, and J. H. Park, "Blockchain-enabled secure framework for energy-efficient smart parking in sustainable city environment," *Sustainable Cities and Society*, vol. 76, p. 103364, 2022.
- [238] R. M. B. Hani and A. A. Ijjeh, "A survey on leach-based energy aware protocols for wireless sensor networks," *Journal of Communications*, vol. 8, no. 3, pp. 192–206, 2013.
- [239] D. V. Le, H. Oh, and S. Yoon, "Rocomar: Robots' controllable mobility aided routing and relay architecture for mobile sensor networks," *Sensors*, vol. 13, no. 7, pp. 8695–8721, 2013.

- [240] Z.-x. Liu, L.-l. Dai, M. Kai, and X.-p. Guan, "Balance energy-efficient and real-time with reliable communication protocol for wireless sensor network," *The Journal of China Universities of Posts and Telecommunications*, vol. 20, no. 1, pp. 37–46, 2013.
- [241] T. Beluch, D. Dragomirescu, and R. Plana, "A sub-nanosecond synchronized mac-phy cross-layer design for wireless sensor networks," *Ad hoc networks*, vol. 11, no. 3, pp. 833–845, 2013.
- [242] J. Sen, "An adaptive and multi-service routing protocol for wireless sensor networks," in *2010 16th Asia-Pacific Conference on Communications (APCC)*, IEEE, 2010, pp. 273–278.
- [243] N. Mittal and U. Singh, "Distance-based residual energy-efficient stable election protocol for wsns," *Arabian Journal for Science and Engineering*, vol. 40, no. 6, pp. 1637–1646, 2015.
- [244] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Ieee, 1995, pp. 39–43.
- [245] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, 2010, pp. 65–74.
- [246] X.-S. Yang and S. Deb, "Cuckoo search: Recent advances and applications," *Neural Computing and applications*, vol. 24, no. 1, pp. 169–174, 2014.
- [247] N. Mittal, U. Singh, and B. S. Sohi, "Modified grey wolf optimizer for global engineering optimization," *Applied Computational Intelligence and Soft Computing*, vol. 2016, 2016.
- [248] W. Weibull, "A statistical distribution function of wide applicability," *J Appl Mech*, vol. 18, pp. 290–293, 1951.
- [249] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [250] M. H. Anisi, G. Abdul-Salaam, M. Idris, Y. Idna, A. W. A. Wahab, and I. Ahmedy, "Energy harvesting and battery power based routing in wireless sensor networks," *Wireless Networks*, vol. 23, no. 1, pp. 249–266, 2017.
- [251] S. Halawani and A. W. Khan, "Sensors lifetime enhancement techniques in wireless sensor networks-a survey," *arXiv preprint arXiv:1005.4013*, 2010.
- [252] M. Adnan, M. A. Razzaque, I. Ahmed, I. F. Isnin, *et al.*, "Bio-mimic optimization strategies in wireless sensor networks: A survey," *Sensors*, vol. 14, no. 1, pp. 299–345, 2014.
- [253] J. W. Hallam, O. Akman, and F. Akman, "Genetic algorithms with shrinking population size," *Computational Statistics*, vol. 25, no. 4, pp. 691–705, 2010.
- [254] F. Scholz, "Inference for the weibull distribution," *Stat 498B Industrial Statistics*, vol. 632, p. 59, 2008.

- [255] M. Abd Elaziz, A. A. Heidari, H. Fujita, and H. Moayedi, "A competitive chain-based harris hawks optimizer for global optimization and multi-level image thresholding problems," *Applied Soft Computing*, vol. 95, p. 106347, 2020.
- [256] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *2014 IEEE congress on evolutionary computation (CEC)*, IEEE, 2014, pp. 1658–1665.
- [257] S. K. Dinkar, K. Deep, S. Mirjalili, and S. Thapliyal, "Opposition-based laplacian equilibrium optimizer with application in image segmentation using multilevel thresholding," *Expert Systems with Applications*, p. 114766, 2021.
- [258] S. Debnath, S. Baishya, D. Sen, and W. Arif, "A hybrid memory-based dragonfly algorithm with differential evolution for engineering application," *Engineering with Computers*, pp. 1–28, 2020.
- [259] S. Dhargupta, M. Ghosh, S. Mirjalili, and R. Sarkar, "Selective opposition based grey wolf optimization," *Expert Systems with Applications*, vol. 151, p. 113389, 2020.
- [260] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.
- [261] Q. Liu, L. Wu, W. Xiao, F. Wang, and L. Zhang, "A novel hybrid bat algorithm for solving continuous optimization problems," *Applied Soft Computing*, vol. 73, pp. 67–82, 2018.
- [262] Y. Yin, Q. Tu, and X. Chen, "Enhanced salp swarm algorithm based on random walk and its application to training feedforward neural networks," *Soft Computing*, pp. 1–17, 2020.
- [263] S. Gupta and K. Deep, "A memory-based grey wolf optimizer for global optimization tasks," *Applied Soft Computing*, vol. 93, p. 106367, 2020.
- [264] L. Ma, C. Wang, N.-g. Xie, M. Shi, Y. Ye, and L. Wang, "Moth-flame optimization algorithm based on diversity and mutation strategy," *Applied Intelligence*, pp. 1–37, 2021.
- [265] E. A. Khalil and B. A. Attea, "Stable-aware evolutionary routing protocol for wireless sensor networks," *Wireless Personal Communications*, vol. 69, no. 4, pp. 1799–1817, 2013.
- [266] F. A. Aderohunmu, J. D. Deng, *et al.*, "An enhanced stable election protocol (sep) for clustered heterogeneous wsn," *Department of Information Science, University of Otago, New Zealand*, 2009.
- [267] S. Chakraborty, A. K. Saha, S. Nama, and S. Debnath, "Covid-19 x-ray image segmentation by modified whale optimization algorithm with population reduction," *Computers in Biology and Medicine*, vol. 139, p. 104984, 2021.

- [268] A. G. Hussien, A. A. Heidari, X. Ye, G. Liang, H. Chen, and Z. Pan, “Boosting whale optimization with evolution strategy and gaussian random walks: An image segmentation method,” *Engineering with Computers*, pp. 1–45, 2022.
- [269] J. Tu, H. Chen, J. Liu, *et al.*, “Evolutionary biogeography-based whale optimization methods with communication structure: Towards measuring the balance,” *Knowledge-Based Systems*, vol. 212, p. 106 642, 2021.
- [270] R. Salgotra, U. Singh, G. Singh, N. Mittal, and A. H. Gandomi, “A self-adaptive hybridized differential evolution naked mole-rat algorithm for engineering optimization problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 383, p. 113 916, 2021.
- [271] W. Al-Hassan, M. Fayek, and S. Shaheen, “Psosa: An optimized particle swarm technique for solving the urban planning problem,” in *2006 International Conference on Computer Engineering and Systems*, IEEE, 2006, pp. 401–405.
- [272] R. Salgotra, U. Singh, S. Singh, G. Singh, and N. Mittal, “Self-adaptive salp swarm algorithm for engineering optimization problems,” *Applied Mathematical Modelling*, vol. 89, pp. 188–207, 2021.
- [273] X.-B. Meng, X. Z. Gao, Y. Liu, and H. Zhang, “A novel bat algorithm with habitat selection and doppler effect in echoes for optimization,” *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6350–6364, 2015.
- [274] S. Tabatabaei, “A novel fault tolerance energy-aware clustering method via social spider optimization (sso) and fuzzy logic and mobile sink in wireless sensor networks (wsns).,” *Comput. Syst. Sci. Eng.*, vol. 35, no. 6, pp. 477–494, 2020.
- [275] J. Wang, H. Han, H. Li, S. He, P. K. Sharma, and L. Chen, “Multiple strategies differential privacy on sparse tensor factorization for network traffic analysis in 5g,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1939–1948, 2021.
- [276] J. Wang, Y. Gao, C. Zhou, S. Sherratt, and L. Wang, “Optimal coverage multi-path scheduling scheme with multiple mobile sinks for wsns,” *Computers, Materials & Continua*, vol. 62, no. 2, pp. 695–711, 2020.
- [277] J. Wang, Y. Gao, X. Yin, F. Li, and H.-J. Kim, “An enhanced pegasis algorithm with mobile sink support for wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 2018, 2018.
- [278] N. Mittal, U. Singh, R. Salgotra, and B. S. Sohi, “An energy efficient stable clustering approach using fuzzy extended grey wolf optimization algorithm for wsns,” *Wireless Networks*, vol. 25, no. 8, pp. 5151–5172, 2019.

- [279] H. Yu, S. Qiao, A. A. Heidari, C. Bi, and H. Chen, "Individual disturbance and attraction repulsion strategy enhanced seagull optimization for engineering design," *Mathematics*, vol. 10, no. 2, p. 276, 2022.
- [280] S. Hussain and A. W. Matin, "Hierarchical cluster-based routing in wireless sensor networks," in *IEEE/ACM international conference on information processing in sensor networks, IPSN*, 2006.
- [281] N. Mittal, U. Singh, and B. S. Sohi, "A stable energy efficient clustering protocol for wireless sensor networks," *Wireless Networks*, vol. 23, no. 6, pp. 1809–1821, 2017.
- [282] P. Singh, P. Singh, and N. Mittal, "Optimized localization using naked mole-rat algorithm in dynamic wireless sensor networks," *International Journal of Communication Systems*, vol. 34, no. 10, e4832, 2021.
- [283] G. Singh and U. Singh, "Triple band-notched uwb antenna design using a novel hybrid optimization technique based on de and nmr algorithms," *Expert Systems with Applications*, vol. 184, p. 115 299, 2021.
- [284] A. A. Taleb, T. A. Alhmiedat, R. A. Taleb, and O. A.-H. Hassan, "Sink mobility model for wireless sensor networks," *Arabian Journal for Science and Engineering*, vol. 39, pp. 1775–1784, 2014.
- [285] Y. Yuan, M. Chen, and T. Kwon, "A novel cluster-based cooperative mimo scheme for multi-hop wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, pp. 1–9, 2006.

Appendix A

CEC Numerical Benchmark Functions/Problems/Test suite

Table A.1: CEC 2005 benchmark functions/problems/test suite

Function	Dim	Range	Shift position	f_{min}
Unimodal functions				
$F1(x)/t_1(x)/p_1(x) = \sum_{i=1}^n x_i^2$	30	[-100, 100]	[-30, -30, ..., -30]	0
$F2(x)/t_2(x)/p_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10, 10]	[-3, -3, ..., -3]	0
$F3(x)/t_3(x)/p_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	[-100, 100]	[-3, -3, ..., -3]	0
$F4(x)/t_4(x)/p_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	[-100, 100]	[-3, -3, ..., -3]	0
$F5(x)/t_5(x)/p_5(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_1 - 1)^2$	30	[-30, 30]	[-3, -3, ..., -3]	0
$F6(x)/t_6(x)/p_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	[-10, 10]	[-3, -3, ..., -3]	0
$F7(x)/t_7(x)/p_7(x) = \sum_{i=1}^n ix_i^4 = random[0, 1]$	30	[-1.28, 1.28]	[-3, -3, ..., -3]	0
Multimodal functions				
$F8(x)/t_8(x)/p_8(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	[-30, -30, ..., -30]	0
$F9(x)/t_9(x)/p_9(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	[-100, 100]	[-30, -30, ..., -30]	0
$F10(x)/t_{10}(x)/p_{10}(x) = \frac{1}{4000}\sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-600, 600]	[-30, -30, ..., -30]	0
$F11(x)/t_{11}(x)/p_{11}(x) = \frac{\pi}{n}10\sin(\pi y_1) + \sum_{i=1}^n -1(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})]$ $(y_n - 1)^2 + \sum_{i=1}^n (x_i, 10, 100, 4)y_i = 1 = \frac{x_i + 1}{4}$	30	[-50, 50]	[-30, -30, ..., -30]	0
$F12(x)/t_{12}(x)/p_{12}(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2(1 + \sin^2(3\pi x_i + 1)))$ $+ 0.1((x_n - 1)^2[1 + \sin^2(2\pi x_n)] + \sum_{i=1}^n u(x_1, 5, 100, 4))$	30	[-50, 50]	[-30, -30, ..., -30]	0

Table A.2: 100-digit challenge (CEC 2019) benchmark functions/problems/test suite

Function No.	Functions	$F_i^* = F_i(x^*)$	Dim	Search range
CEC - 01/TP ₀₁ /b ₁	Storn's Chebyshev Polynomial Fitting Problem	1	9	[-8192, 8192]
CEC - 02/TP ₀₂ /b ₂	Inverse Hilbert Matrix Problem	1	16	[-16,384, 16384]
CEC - 03/TP ₀₃ /b ₃	Lennard-Jones Minimum Energy Cluster	1	18	[-4, 4]
CEC - 04/TP ₀₄ /b ₄	Rastrigin's Function	1	10	[-100, 100]
CEC - 05/TP ₀₅ /b ₅	Griewangk's Function	1	10	[-100, 100]
CEC - 06/TP ₀₆ /b ₆	Weierstrass Function	1	10	[-100, 100]
CEC - 07/TP ₀₇ /b ₇	Modified Schwefel's Function	1	10	[-100, 100]
CEC - 08/TP ₀₈ /b ₈	Expanded Schaffer's F6 Function	1	10	[-100, 100]
CEC - 09/TP ₀₉ /b ₉	Happy Cat Function	1	10	[-100, 100]
CEC - 10/TP ₁₀ /b ₁₀	Ackley Function	1	10	[-100, 100]

Table A.3: CEC 2014 benchmark functions/problems/test suite

	Function No.	Functions	$F_i^* = F_i(x^*)$
Unimodal Functions	F_1/CEC_{01}	Rotated High Conditioned Elliptic	100
	F_2/CEC_{02}	Rotated Cigar	200
	F_3/CEC_{03}	Rotated Discus	300
	F_4/CEC_{04}	Shifted and Rotated Rosenbrock	400
Simple Multimodal Functions	F_5/CEC_{05}	Shifted and Rotated Ackley	500
	F_6/CEC_{06}	Shifted and Rotated Weierstrass	600
	F_7/CEC_{07}	Shifted and Rotated Griewank	700
	F_8/CEC_{08}	Shifted Rastrigin	800
	F_9/CEC_{09}	Shifted and Rotated Rastrigin	900
	F_{10}/CEC_{10}	Shifted Schwefel	1000
	F_{11}/CEC_{11}	Shifted and Rotated Schwefel	1100
	F_{12}/CEC_{12}	Shifted and Rotated katsuura	1200
	F_{13}/CEC_{13}	Shifted and Rotated HappyCat	1300
	F_{14}/CEC_{14}	Shifted and Rotated HGBat	1400
	F_{15}/CEC_{15}	Shifted and Rotated Expanded Griewank plus Rosenbrock	1500
	F_{16}/CEC_{16}	Shifted and Rotated Expanded Scaffer	1600
Hybrid Functions	F_{17}/CEC_{17}	Hybrid 1($N = 3$)	1700
	F_{18}/CEC_{18}	Hybrid 2($N = 3$)	1800
	F_{19}/CEC_{19}	Hybrid 3($N = 4$)	1900
	F_{20}/CEC_{20}	Hybrid 4($N = 4$)	2000
	F_{21}/CEC_{21}	Hybrid 5($N = 5$)	2100
	F_{22}/CEC_{22}	Hybrid 6($N = 5$)	2200
	F_{23}/CEC_{23}	Composition 1($N = 5$)	2300
Composition Functions	F_{24}/CEC_{24}	Composition 2($N = 3$)	2400
	F_{25}/CEC_{25}	Composition 3($N = 3$)	2500
	F_{26}/CEC_{26}	Composition 4($N = 5$)	2600
	F_{27}/CEC_{27}	Composition 5($N = 5$)	2700
	F_{28}/CEC_{28}	Composition 6($N = 5$)	2800
	F_{29}/CEC_{29}	Composition 7($N = 3$)	2900
	F_{30}/CEC_{30}	Composition 8($N = 3$)	3000
Search Range: $[-100, 100]^D$			