Improved Roach-based Algorithms for Global Optimization Problems

Obagbuwa Ibidun Christiana

211559859

A thesis submitted in the fulfilment of the requirements
for the Degree of Doctor of Philosophy in Computer Science

In the

School of Mathematics, Statistics, and Computer Science

University of KwaZulu- Natal
Durban, South Africa

November 2014

UNIVERSITY OF KWAZULU-NATAL
COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE

DECLARATION

The research described in this thesis was performed at the University of KwaZulu-Natal under the supervision of Dr. Adewumi A.O. I hereby declare that all the materials incorporated in this thesis are my own original work except where acknowledgement is made by name or in the form of reference. The work contained herein has not been submitted in part or whole for a degree at any university.

Signed:

Date:

As candidate's supervisor, I have approved the dissertation for submission

Signed:

Date:

UNIVERSITY OF KWAZULU-NATAL
COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE

DECLARATION PLAGIARISM

I, Obagbuwa Ibidun Christiana declare that

1. The research reported in this thesis, except where otherwise indicated, is my original research.
2. This thesis has not been submitted for any degree or examination at any other University.
3. This thesis does not contain other persons data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
4. This thesis does not contain other persons writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
    a. Their words have been re-written and the general information attributed to them has been referenced
    b. Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.
5. This thesis does not contain text, graphics or Tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References section.

Signed:

Student name: Obagbuwa Ibidun Christiana

Dedication

This work is dedicated to the ALMIGHTY GOD, the 'I AM THAT I AM' who is my all in all. May his name be praised forever and ever.

Abstract

Optimization of systems plays an important role in various fields including mathematics, economics, engineering and life sciences. A lot of real world optimization problems exist across field of endeavours such as engineering design, space planning, networking, data analysis, logistic management, financial planning, risk management, and a host of others. These problems are constantly increasing in size and complexity, necessitating the need for improved techniques.

Many conventional approaches have failed to solve complex problems effectively due to increasingly large solution space. This has led to the development of evolutionary algorithms that draw inspiration from the process of natural evolution. It is believed that nature provides inspirations that can lead to innovative models or techniques for solving complex optimization problems. Among the class of paradigm based on this inspiration is Swarm Intelligence (SI).

SI is one of the recent developments of evolutionary computation. A SI paradigm is comprised of algorithms inspired by the social behaviour of animals and insects. SI-based algorithms have attracted interest, gained popularity and attention because of their flexibility and versatility. SI-based algorithms have been found to be efficient in solving real world optimization problems. Examples of SI algorithms include Ant Colony Optimization (ACO) inspired by the pheromone trail-following behaviour of ant species; Particle Swarm Optimization (PSO) inspired by flocking and swarming behaviour of insects and animals; and Bee Colony Optimization (BCO) inspired by bees' food foraging.

Recent emerging techniques in SI includes Roach-based Algorithms (RBA) motivated by cockroaches social behaviour. Two recently introduced RBA algorithms are Roach Infestation Optimization (RIO) and Cockroach Swarm Optimization (CSO) which have been applied to some optimization problems to achieve competitive results when compared to PSO. This study is motivated by the promising results of RBA, which have shown that the algorithms have potentials to be efficient tools for solving optimization problems. Extensive studies of existing RBA were carried out in this work revealing the shortcomings such as slow convergence and entrapment in local minima. The aim of this study is to overcome the identified drawbacks. We investigate RBA variants that are introduced in this work by introducing parameters such as constriction factor and sigmoid function that have proved effective for similar evolutionary algorithms in the literature.

In addition components such as vigilance, cannibalism and hunger are incorporated into existing RBAs. These components are constructed by the use of some known techniques such as simple Euler, partial differential equation, crossover and mutation methods to speed up convergence and enhance the stability, exploitation and exploration of RBA.

Specifically, a stochastic constriction factor was introduced to the existing CSO algorithm to improve its performance and enhance its ability to solve optimization problems involving thousands of variables. A CSO algorithm that was originally designed with three components namely chase-swarming, dispersion and ruthlessness is extended in this work with hunger component to improve its searching ability and diversity. Also, predator-prey evolution using crossover and mutation techniques were introduced into the CSO algorithm to create an adaptive search in each iteration thereby making the algorithm more efficient. In creating a discrete version of a CSO algorithm that can be used to evaluate optimization problems with any discrete range value, we introduced the sigmoid function.

Furthermore, a dynamic step-size adaptation with simple Euler method was introduced to the existing RIO algorithm enhancing swarm stability and improving local and global searching abilities. The existing RIO model was also re-designed with the inclusion of vigilance and cannibalism components.

The improved RBA were tested on established global optimization benchmark problems and results obtained compared with those from the literature. The improved RBA introduced in this work show better improvements over existing ones.

Acknowledgements

Firstly, my acknowledgement goes to the ALMIGHTY GOD, whose presence is with me always. All through the period of this work, I saw HIS finger, HIS hand and HIS being. This work is a proof of the strange works and acts of GOD. GOD the FATHER, GOD the SON and GOD the HOLY SPIRIT are very real; the WORD of GOD is true and is the TRUTH (John 17:17). All inspirations and ideas came from HIM via the Spirit of Truth (1 Corri 2: 9-10). Every challenge encountered during this work gave way for the TRUTH, the TRUTH prevailed in all situation and circumstance. All GLORY about and around this work is ascribed unto the ALMIGHTY GOD.

Also, I acknowledge all the effort of my supervisor, Dr. A.O Adewumi. He has been there for me from inception, beginning from my journey from my country to South Africa. He assisted me in securing an accommodation and ensure that I properly settled down. From the beginning of this work till the end, I enjoyed his guidance. Thank you very much sir. May the good LORD rewards you.

My acknowledgement also goes to my colleagues (Post Doctoral Fellows at the time of study): Dr. Sawyerr Babatunde of University of Lagos; Dr. Nachamada Blamah of University of Jos and Dr. Adebiyi Ayodele of Covenant University Nigeria; I appreciate your encouragement. I equally appreciate my other colleagues in my research group: Ajibola Sunday, Luke Joel, Olusanya Micheal, Matins Arasomwan, Stephen Akandwanaho and Ameline. Also to every staff of school of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, appreciation is given for their kindness.

I am grateful to my employer, Lagos State University, Lagos, Nigeria, for granting me a training leave for this doctoral degree. Also acknowledgement is given to every staff and students of the Department of Computer Sciences, Lagos State University.

I acknowledge my dear father, Dr. Olaniyi David Oyedepo, the presiding Bishop of Living Faith Church Worldwide (aka Winners Chapel International), God has used him greatly to transform my life through the liberation mandate and the preaching of the word of faith. I am able to go this far and still flying higher by the impacts of the liberation messages. Also to Winners family worldwide, acknowledgement is given for sweet fellowship.

Acknowledgement is rendered to Joseph kirk and Jonas Lundgren for making their codes ac-

cessible in Mathworks, which became an inspiration for our study in travelling salesman problems.

Furthermore, my appreciation goes to my sweet family and precious home: To my husband, Mr. Obagbuwa Oloyede (ACA), whom GOD has used to make this programme a reality. I thank him for the rare privilege given to me to be away from home for this long period of time, he gave me the permission to go and promised to take good care of our children which he did beyond expectation. You are the best husband and father in the whole world, you are so kind and committed to the course and advancement of your home and that of every member of the family. You told me that you can do anything to take me to the peak of my carrier and you stand by it. May God Almighty continue to bless you and give unto you long life to eat the fruit of your labor (Isa 65:20-24). And to the wonderful heritage of GOD in my home that I am privileged by God to mother: Oluwadamilola, Oluwatimilehin and Oluwagbemiga, who are outstanding blessings from GOD. I appreciate your true love, patience, understanding and prayer for me. In my lifetime you shall sit on great thrones in the name of Jesus Christ our Savior (AMEN). Also to my late father, HRH Falope Francis Fagite, my mother Mrs. Elizabeth Falope and to all my Siblings, due acknowledgements are rendered.

Contents

List of Figures

List of Tables

List of Acronyms

Abbreviations	Meaning/Interpretation
ACO	Ant Colony Optimization
ACSO	Adaptive Cockroach Swarm Optimization
AE	Artificial Evolution
AIS	Artificial immune system
ANOVA	Analysis of Variance
BA	Bat algorithm
BCO	Bees Colony Optimization
BFOA	Bacteria Foraging Optimization Algorithm
BSO	Bio-luminescent Swarm Optimization
CE	Cultural Evolutionary
CI	Computational Intelligence
CSO	Cockroach Swarm Optimization
CS	Cuckoo search
CSS	Charge system search
COP	Combinatorial Optimization problem
DE	Differential Evolution
DCSO	Discrete Cockroach Swarm Optimization
DCSO-TSP	Discrete Cockroach Swarm Optimization for Travelling Salesman Problem
DPSO	Discrete Particle Swarm Optimization
DSARIO	Dynamic Step-Size Adaptation Roach Infestation Optimization
EA	Evolutionary Algorithm
EC	Evolutionary Computation
EP	Evolutionary Programming
ES	Evolutionary Strategies
FA	Firefly Algorithm
FLS	Fuzzy Logic System
GA	Genetic Algorithm
GP	Genetic programming
GSO	Glow-worm Swarm Optimization
GWO	Grey wolf optimization
GSA	Gravitational search algorithm
HRIO	Hungry Roach Infestation Optimization
IWD	Intelligent water drop
ICSO	Improved Cockroach Swarm Optimization
LSRS	Line Search Restart
MeanOpt	Mean Optimum
MCSO	Modified Cockroach Swarm Optimization
MinOpt	Minimum Optimum
MFE	Mean Function Evaluation
MPP	Modified Predator-Prey
MRIO	Modified Roach Infestation Optimization
NN	Nearest neighbour
NNs	Neural networks

PDE	Partial Differential Equation
PPEM	Predator-Prey Evolution Method
PSO	Particle Swarm Optimization
RBA	Roach-based Algorithm
RIO	Roach Infestation Optimization
SCCSO	Stochastic Constriction Cockroach Swarm Optimization
SFA	Sheep flocks algorithm
SI	Swarm Intelligence
SP	Success Performance
SO	Self Organization
SR	Number of Successful Runs
SRT	Success Rate
STD	Standard Deviation
SCF	Stochastic Constriction Factor
TSP	Traveling Sales Problem
TR	Total Number of Runs

List of Outputs

1 Obagbuwa I.C, Adewumi A.O, Adebiyi A.A. (2014), "Stochastic Constriction Cockroach Swarm Optimization for Multidimensional Space Function Problems, " Journal of Mathematical Problems in Engineering, 2014, Article ID 430949.

2 Obagbuwa I.C. and Adewumi A.O. (2014), "An Improved Cockroach Swarm Optimization," The Scientific World Journal, 2014, Article ID 375358.

3 Obagbuwa I.C, Adewumi A.A, Adebiyi A.A (2014), "A Dynamic StepSize Adaptation Roach Infestation Optimization". In proceedings of International conference on advance computing (IACC 2014), Gurgaon, India, February 2014, pp 1201-1205.

4 Obagbuwa I.C. and Adewumi A.O. (2014), "Modified Roach Infestation Optimization," In proceedings of IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB 2014), Hilton Hawaiian village, Honolulu, Hawaii, USA, May 21st to 24th, 2014.

Chapter 1

Introduction

1.1 Background

Optimization is the search for the best possible solution among diverse solutions to a given problem [1, 2, 3, 4]. The task of finding the solutions that are as good as possible and widely different from each other is referred to as global optimization [2]. Optimization is useful across disciplines for example: airline companies schedule crews and aircraft to minimize cost; business investors make selections that avoid excessive risks, while desiring high return rates; manufacturers strive to maximize efficiency in the design and operation of their production processes; the economist and operational researchers desire optimal allocation of resources in industrial and social settings, and so on.

A lot of real-world and theoretical problems can be modelled as a general framework to maximize or minimize a mathematical function of variables, subject to some constraints. Usually, the terms maximization or minimization is used to represent the term optimization; maximization of the function f is equivalent to minimising $-f$. The mathematical function that is to be optimized is known as the objective function, which normally contain a number of variables [1]. Optimization processes involve the construction of relevant models, by expressing optimization problem in mathematical terms and identifying the objective, the variables and the constraints of the problem [1, 4].

a. An objective is the performance quantitative measure of the system that needs to be minimized or maximized. The objective must be identified first, for example to maximize profits or minimize the cost of production. The objective function represents the goal/objective of the problem in terms of decision variables.

b. The variables or the unknowns are the system components for which we want to find values. For example, the variables can be the amount of each resource consumed or the time spent

on each activity.

c. The constraints are the functions that describe the relationships among the variables and also define the allowable values for the variables. For example, the amount of a resource consumed cannot exceed the available amount [4].

Optimization problems can be generally classified, as depicted in Figure 1.1. An objective function can be a function of a single variable for some practical problems. Optimization problems may involve more than one objective function and are known as multi-objective optimization problems. This refers to a problem where two or more objective functions need to be minimized concurrently. In this case, the optimality of solutions is redefined, since global minimal of different objective functions are rarely achieved at the same minimizer. For instance to develop a new component might involve minimizing weight while maximizing strength or choosing a portfolio might involve maximizing the expected return while minimizing the risk [4].

The nature of a problem determines the type of variable to be used in its model. Variables may be real or integer or a combination of the two [1]. Optimization problems may be constrained or unconstrained. In a constrained optimization problem the search space is usually restricted by its constraints. An unconstrained problem is a problem where constraints are absent or omitted. In unconstrained problems, the whole domain of the objective function is the feasible set. Unconstrained problems arise directly in many practical applications and in the reformulation of constrained optimization problems by replacing the constraints by a penalty term in the objective function [4]. Constrained problems arise from applications where there are explicit constraints on the variables which may vary widely from simple bounds to systems of equalities and inequalities that model complex relationships among the variables. The nature of the constraints such as linear, non-linear, convex and the smoothness of the functions such as differentiable or non-differentiable can be used to further classify constrained optimization problems [4].

This thesis deals mostly with unconstrained minimization tasks, formally described mathematically as:

$$
\begin{gathered}
min_x\ f(x) \\
\text{Given } x \in \mathbb{R}^n \text{ is a real vector with} \\
n \geq 1 \text{ components and} \\
f : \mathbb{R}^n \to \mathbb{R} \text{ being a smooth function.} \\
\text{find } x^* \in \mathbb{R}^n \text{ for which}
\end{gathered}
$$

$$
f(x^*) \leq f(x), \forall x \in \mathbb{R}^n \tag{1.1}
$$

Constrained optimization problems can be described as:

$$\min f(x)$$
$$\text{subject to } c_i(x) = 0 \; \forall i \in E$$

$$c_i(x) \leq 0, \forall i \in I \tag{1.2}$$

where f and the functions $c_i(x)$ are all smooth, real-valued functions on a subset of $\mathbb{R}^n$ and E and I are index sets for equality and inequality constraints, respectively. The feasible set is the set of points x that satisfy the constraints [4].

Function classification deals with properties of mathematical function; the objective or constraint functions could be linear or nonlinear [1, 4]. In a given model, if all the functions are linear, it is being referred to as a linear programming model or a linear model. Otherwise it is referred to as nonlinear. The development of a many optimization techniques are based on convex function [1, 4]. A solution approach in optimization consists of two categories: derivatives and derivatives-free approaches [1]. Derivative-based techniques are capable of determining search directions according to an objective function's derivative information. Derivative-based techniques are used in optimizing non-linear neuro-fuzzy models. Examples include: Descent methods; the method of Steepest Descent; classical Newton's method; stepsize determination and conjugate gradient methods. On the other hand, derivative-free optimization concepts are based on nature's intelligence, such as evolution and thermodynamics. Examples include: genetic algorithm; simulated annealing; random search and downhill simplex search.

Differentiability of a function is similar to functions continuity. Differentiability of function is important if the considered technique is derivative-based. For instance, as shown in Figure 1.1, the properties of a single-objective function, unconstrained problem with continuous variables could be nonlinear, convex, and differential.

In addition to the above general classification, function properties such as unimodal and multimodal, are considered in optimization problem domains. A unimodal function is a function with only one peak-optimum solution. Whereas a unimodal function is a function with more than one peak- local or global optima.

The minimization problems described above can be solved using the techniques called local and global optimization techniques. A local search method iteratively improves its estimate of the minimum by searching for better solutions within a local neighbourhood of the current solution. Since local minima are not guaranteed to be global minima, global optimization methods have been developed to search complicated landscapes of multiple local minima. Global optimization is an inherently complex problem since no general criterion exists for determining when the global optimum is reached. The two competing goals of global optimization techniques are called global reliability (exploration) and local refinement (exploitation) [5]. Exploration is an act of searching

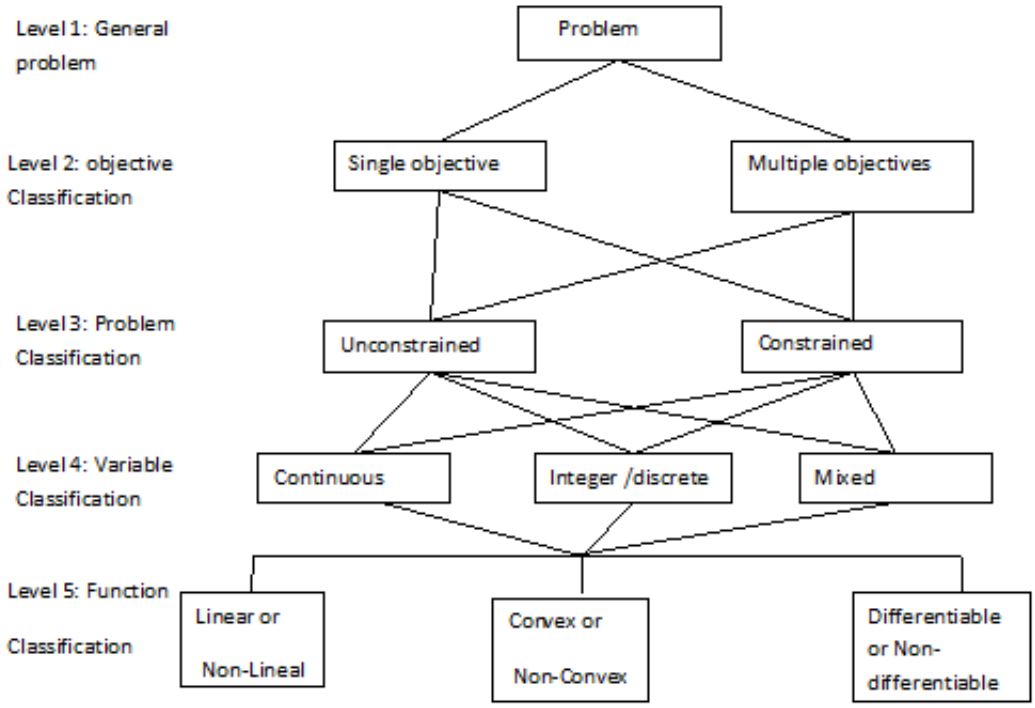

Figure 1.1: Classification of Optimization Problems (Adapted from [1, 3])

the entire search space to find a reasonable estimate of the global optimum, whereas, exploitation is searching the immediate neighbourhood of the current solution for better solutions. Most global optimization techniques achieve these two goals by using a combination of a global and a local strategies [6]. This thesis focuses on global optimization techniques.

Once a model has been prepared for a given optimization problem, suitable optimization techniques can be used to find its solution [1, 2, 3, 4]. Optimization techniques can be classified into deterministic and probabilistic as shown in Figure 1.2. Deterministic technique can be applied to problems whose solution search space can be efficiently explored. Probabilistic technique handles situations where the relations between a solution candidate and its fitness are not so obvious, too complicated, or the dimensionality of the search space is very high [2]. There is a general belief that nature provides inherent solutions to the quest for optimality. This belief led to the development of evolutionary techniques (shown in Figure 1.2) that draw inspiration from the processes of natural evolution, especially the principle of survival of the fittest.

A recent development is the class of swarm intelligence techniques which include bees' algorithm that is based on bees' social behaviour, Ant algorithm that is based on ants' behaviour, Particle Swarm algorithm that is based on birds flocking and RBA that is based on cockroach social behaviour. These techniques were developed from the study of various social behaviours of certain natural animals. They have been successful applied to various classes and complexity of

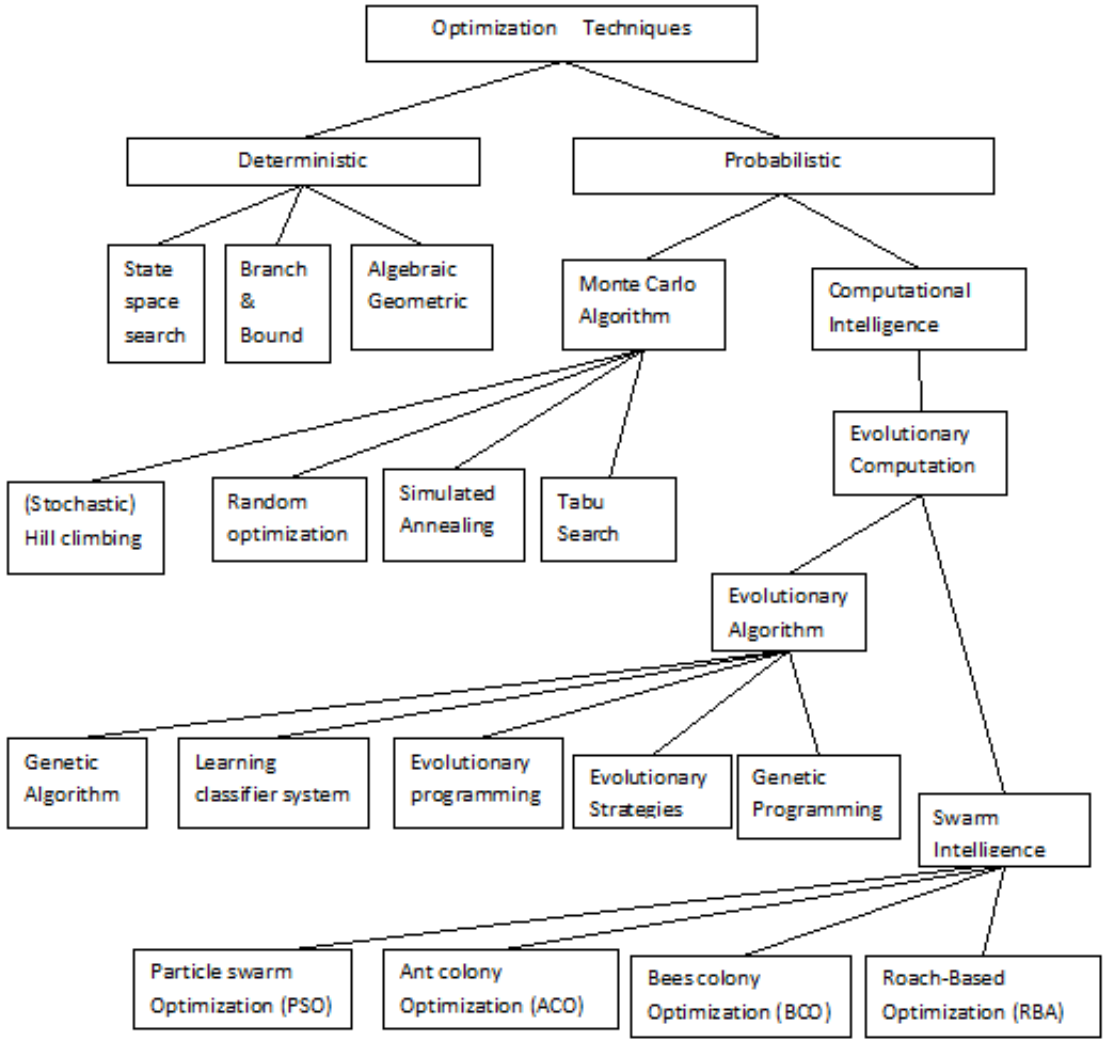

Figure 1.2: Classification of Global Optimization Techniques (Adapted from [2])

optimization problems. However, there is still rooms left for improvement on these techniques. This thesis examines one of the new techniques under swarm intelligence (SI) technique namely RBA. An extensive study of the social behaviour of cockroach swarm and a review of the existing RBA (RIO and CSO algorithms) is carried out in this work. This work investigates the strengths and weaknesses of existing RBA and is aimed at tackling their obvious drawbacks which are slow convergence and trapping into local minima. The aim is achieved by introducing methods, parameters and incorporating new components into the existing RBAs which resulted in the development of variants of RBA with improved speed, swarm stability, balanced exploitation and exploration. The proposed RBA variants were deployed for varying global optimization problems and the obtained results compared empirically and statistically with the existing algorithms. Generally, RBA

variants show an improved performance over the existing one in most cases.

In this work, a multi-valued discrete space cockroach algorithm is also designed to solve optimization problems within discrete value range through the introduction of sigmoid function and an operator into a CSO algorithm for the logical transformation of cockroach positions from a continuous state to discrete.

1.2 Statement of the Problem

The global optimization problem is formulated in terms of finding the point x in a solution space set X, referred to as a feasible region. A function $f : X \to T$, called objective function, locates a minimum or maximum. T is a any ordered set, usually a subset of $\Re$. x could be a continuous variable vector and f a continuous real-valued function. We represent the global optimal solution as x^* and the corresponding global optimization function value as $f(x^*)$ or f^*.

The set X is normally a subset of $\Re^n$ defined by constraints $c_i(x) \geq = \leq 0$, where c_i is a set of m possibly nonlinear functions of x and $\leq$, $=$ and $\geq$ are relations in $\{\leq, =, \geq\}$. The extremal point x can then be written as $(x_1, ..., x_n)$, and the x_i's are called decision variables. The domain X is defined within the upper and lower limits (x^L, x^U) of each dimension; variable bounds can be expressed as $x^L \leq x \leq x^U$ where $(x^L, x^U \in \Re^n)$. Variables may be constrained to only take integer values $x_i \in \mathbb{Z}$ for all i in an index set $Z \subseteq \{1, ..., n\}$.

We consider minimization problems, described in equations (1.1) and (1.2) in this thesis; using RBAs as global optimization techniques. The shortcomings of the existing RBAs are addressed, these are slow convergence and entrapment in local minima. The shortcomings are addressed by improving the searching capabilities of the algorithms via the incorporation of parameters, components, and features that enhance both exploration and exploitation.

1.3 Aim and Objectives of the Study

1.3.1 Aim

The research work aimed to improve existing RBAs with new components, operators and parameters in order to evolve improved variants of RBA.

1.3.2 Objectives

The specific objectives of the study are as follows:

1. To propose improved RBAs from the existing RBA.
2. To make RBA a global optimization technique with improved convergence on global optima, through the extension of its models.
3. To design and test adaptive RBAs based on methods and models that have been proven to be successful when applied to other evolutionary techniques.

4. To design and test multi-valued discrete space RBA for solving global optimization problems with any discrete values, based on the methods that have been successfully applied to other SI techniques.

5. To evaluate the robustness of the proposed RBA variants with standard global optimization test problems and comparing the obtained results with those in the literature.

1.4 Methodology

The proposed improved RBA variants were tested on global optimization problems. The empirical results were analysed statistically and compared with existing algorithms to show the superiority of the improved RBAs over the existing algorithms in most cases. Performance evaluation criteria measures used to evaluate algorithms include: mean function evaluations, run-time and success performance.

1.5 Contributions to Knowledge

In this thesis, the development of methods for extending existing RBAs were carried out, and their abilities are enhanced to locate an objective function minimizer. New components, parameters and methods were incorporated into the RBAs to improve swarm stability, speed, searching capabilities and population diversity which resulted in new RBAs. Also the design and implementation of a multi-valued discrete space RBA that can be used to evaluate optimization problems with any discrete range value was achieved. The major contributions of this thesis are as follows:

1. A stochastic constriction factor was introduced to CSO to develop a stochastic constriction cockroach swarm optimization algorithm.

2. A partial differential equation for migration was used to mimic migration behaviour when a cockroach is hungry and needs to move to an available food source. An improved cockroach swarm optimization was proposed.

3. The Euler method in numerical analysis was used to improve the find dark and friend components of RIO and a dynamic step-size roach infestation optimization algorithm was proposed.

4. A partial differential equation for dispersion was introduced to RIO to encourage the dispersion of cockroaches whenever there is overcrowding.

5. Predator-prey evolution with blend crossover and mutation method was introduced into CSO to mimic the cannibalism and reproductive behaviour of cockroaches. The technique enhances a balance of exploration and exploitation, in such a way that the balance is self-adaptive. Self-adaptation causes populations to diverge or converge to a better region within the given search space.

6. A multi-valued discrete space CSO was introduced to offer solutions to optimization problems in any discrete range.

1.6 Scope of Study

The study was focused on using some techniques for performance enhancement of the existing RBAs in offering solutions to global optimization problems. Performances of the improved RBAs were evaluated with standard global optimization benchmark problems; the results were statistically analysed and compared with the results of existing algorithms.

1.7 Thesis Outline

In the subsequent chapters, we will show the theoretical framework of the research, a review of existing studies, and the methods adopted in improving the existing work to evolve new variants of RBAs. The improved RBAs and application to global optimization problems with empirical results are also presented. The summary of the organization of the remaining part of the thesis is given below:

Chapter 2 gives an introduction to evolutionary computation and swarm intelligence, followed by a review of the social behaviour of cockroaches. Existing works related to improvements and extensions are discussed.

Chapter 3 presents the methodologies engaged in the improvement and extension of the existing RBAs to evolve new and more efficient algorithms.

Chapter 4 shows experimental settings, parameter selection, the statistical analysis tool engaged, the performance evaluation criteria and the global optimization problems solved are also presented.

Chapter 5 presents the improved algorithms with their respective experimental settings, simulation results and comparison results with existing algorithms.

Chapter 6 gives a discussion on the findings of the thesis and the conclusions with a summary of findings of the thesis. Suggestion for future research is also presented.

Finally, the unconstrained test problems used in the study are presented in Appendix A.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews some of the related terms to swarm intelligence. A brief discussion of computational intelligence and evolutionary computation is presented. Emergent behaviour of social animals is discussed followed by a discussion of swarm intelligence techniques. A review of cockroach swarm foundation is given, followed by the presentation of existing RBAs and related works.

2.2 Computational Intelligence

Computational Intelligence (CI) can be described as a set of techniques inspired by natural metaphor to provide solutions to difficult real world problems that conventional approaches have failed to tackle [7]. CI essentially includes Fuzzy logic systems (FLS), Neural Networks (NNs) and Evolutionary Computation (EC) [7]. CI also embrace techniques that stem from the above three or gravitate around one or more of them. Examples are: SI and Artificial immune system which can be seen as a part of EC. The Dempster-Shafer theory, chaos theory and multi-valued logic can be seen as off-springs of FLS [7, 8]. Probabilistic methods such as belief networks are frequently used with CI paradigms. Each of the CI paradigms has its origin in biological systems [7, 8].

EC defines a number of population-based methods and uses a combination of random variation and selection to offer solutions to problems [9]. It models natural evolution including genetic and behavioural evolution, and populations of individuals. Different classes of EC algorithms have been developed, including SI algorithms that are inspired by social behaviour of animals and insects; Genetic Algorithm (GA) which model genetic evolution; genetic programming (GP)

which is based on GA but with individual programs represented as trees; evolutionary programming (EP) which is derived from the simulation of adaptive behaviour in evolution; evolutionary strategies (ES) which are geared towards modelling the strategic parameters that control variation in evolution and differential evolution (DE) which is similar to GA but differing in reproduction mechanisms used.

EC has many application areas including data mining, combinatorial optimization, fault diagnosis, classification, clustering, scheduling and time series approximation [7]. Barricelli began the simulation of evolution in the 1960s [10, 11]. He started by using evolutionary algorithms (EA) and artificial life, with later advances to his work being made by Fraser who made a lot of contributions on simulation of artificial selection. Through Rechenberg who applied ES to solve complex engineering problems in the 1970s, artificial evolution (AE) became an extensively used and popular optimization technique [12]. GA was made popular by John Holland [13]. EA is now very popular and has been applied to solve multi-dimensional optimization problems efficiently [14].

2.3 Swarm Intelligence

SI provides solutions to optimization problems that are motivated by social animals' collective behaviour [15]. Animals swarming has been an area of interest attracting many researchers. The mechanisms that control the social insects and animals' behaviour remained amazing; the animals move in a group, turn together, flow around obstacles and are very clever in their movements [16, 17]. By cooperating, the group is able to achieve complex tasks, despite the fact that the individuals in these colonies are non-sophisticated. Examples of swarms of insects and animals are depicted in Figure 2.1.

2.3.1 Emerging Behaviour in Social Animals

Social animals' collective behaviour is self-organised with no central control [15]. For instance, the trail-following behaviour of insects has resulted in efficient solutions to difficult problems such as finding the shortest route to a food source among innumerable possible paths. The act of searching for food is referred to as foraging, which is an emergent behaviour where the swarm finds and exploits good food sources, and adaptively moves to good new sources as current ones become depleted [16]. Flocking represents group movement as seen in bird flocks and schools of fish in nature. A flocking model was first proposed by Craig Reynolds [18] as a bio-inspired computational model for simulating the animation of a flock of entities called boid. In a flocking model, each boid reacts equally to its neighbouring mates in the flock, and to its environment while

Figure 2.1: Swarm of Insects and Animals [19]

making its own decisions. The decision is based on its movement according to a small number of simple steering rules. Perhaps the most visible phenomenon of swarm intelligence is the travelling behaviour of groups (flocks, schools, swarms, herds, etc) of individuals.

Basic swarm models are controlled by three simple rules namely separation, alignment and

cohesion [18]. The separation rule allows short range repulsion to avoid neighbours crowding together during the flocking process. The alignment rule enables individuals in a swarm to steer towards an average heading of one's neighbours, while the cohesion rule considers the long range attraction whereby individuals in a swarm steer towards the average position of one another [18]. Swarming provides some advantages such as flexibility, robustness, and self-organization. SI is flexible, a group can adapt promptly to a changing environment. SI is robust, a group can still perform necessary tasks when one or some individuals fail. SI is self-organized because a group needs little or no supervision or top-down control [18, 20]. Self-organization is a crucial aspect emanating from the collective behaviour of social animals that results into emergent manifestations. The concept of self-organization is presented in section 2.3.2.

2.3.2 Self Organization in Social Animals

Self-organization (SO) finds application in different fields of study, such as chemistry, physics, biology, linguistics, economics, robotics and computer science. The idea of SO springs from the world of physical chemistry and signifies a process whereby simple local interactions among simple entities produce a high level structure. According to Dario and Claudio [20] a good examples of SO include the appearance of geometric patterns in reaction-diffusion chemical systems, the formation of snow crystals, and the appearance of moving objects within Conway's Game of Life. The resulting global patterns are emergent and more than the sum of constituent parts. The biological world proliferate in collective trends of significant adaptive roles, such as the construction of nests, coordinated collective movement, communication etc. The entities involved rely on simple rules and local information without a global plan or central coordinator, and are dynamic to malfunction or the divergence of some individual. The principle of SO can be used to describe the biological collective phenomenon that ended at structures and functionality that is far beyond the abilities of the entities involved [20].

SO is built on two opposing forces of attraction and repulsion; these forces are termed as positive and negative feedback [21]. Feedback occurs when a measure in a system is fed back into the system to enhance or reduce the extent of the same measure. The interplay of positive and negative feedback results in the equilibrium of self-organization. May shows that positive and negative feedback can display unexpected adjustments or divergence that influence the pattern or the functionality at global level [22]. The interaction between animals happen by means of cues and signals. A cue is an unplanned indicator that can be picked up by animals such as trail-following. The perceiving entity will choose to follow the cue: positive or negative feedback. A signal is an intentional indicator done consciously to affect the behaviour of other receiving animals. Examples are: waggle dance in bees to give food information to others and the alarm cry of birds when a

predator is in close proximity [20].

SO in social animals often requires direct and indirect interactions among them. The indirect interactions that occur between two individuals when one of them modifies the environment and the other responds to the new environment at a later time are called stigmergy. This is a social communication via modification of the environment; the result of work by an entity affects the action of another entity. Pierre-Paul Grassse introduced the concept of stimergy in the 1950s when he explained that indirect communication takes place among social insect groups. A famous example of stigmergy is the pheromonal communication among ants, whereby ants engaging in certain activities, leave a chemical trail which is then followed by their colleagues. Ants, termites and bees, among many other insect species, are known to have a complex social structure (see Figure 2.2). Stigmergy is a form of cue-based communication which is better described with terms such as aggregation, foraging, clustering, nest construction and division of labour [20, 23].

1. Aggregation: Aggregation is an example of self-organization driven by positive and negative feedback. Aggregation refers to a collection of social agents where individual agents behave without central control and each of these agents show emergent behaviour due to interactions with one another and in response to their environment resulting in a global pattern.

2. Foraging: Stigmergy enhances the effectiveness of group foraging. Ants lay a pheromone trail that can be used to choose a path, find the shortest one and established a link between the food source and the nest [20]. Locating food sources depend on the deposition of pheromones by individual ants. In the natural environment, the initial behaviours of a colony of ants in looking for a new food source is for individual ants to roam at random. When an ant find an appropriate food supply it will return to its colony after having laid pheromone trails. Subsequent ants setting out to search for food will sense the pheromone laid down by their precursors, and this will persuade them as to which path they will take up. Pheromone trails evaporate quickly. If the food source is nearby, the first ant to find this source will return quickly, also other ants that take this route will also return quickly. The best routes will enjoy a greater regularity of pheromone laying in a short time, becoming strongly fancied by other ants [16, 24].

3. Clustering and sorting of objects are part of emergent behaviours exhibited by insect species. Examples include the clustering of corpses of dead ants into cemeteries and the arrangement of larvae into similar-age groups (brood-sorting), for instance, in Leptothorax unifasciatus ant colonies, the young ones are organized into concentric rings called annuli [16, 24]. Each ring comprises young of a different type. The youngest (eggs, and micro-larvae) are clustered together in the central cluster. Outwards from the center, progressively older groups of

larvae are encountered. Also, the spaces between these rings increase as one move outwards [16, 24]. In cemetery arrangement, certain ant species are identified to gather their dead into piles, each piles keep a minimal distance from each other. In this manner, corpses are separated from the living environment, and no longer an impediment to the colony [24].

4. Construction: The extraordinary united behaviour that leads to the construction of wasp nests, termite mounds and bee hives are shown in Figure 2.2. Such structures amazes observers when trying to picture how such simple minds can lead to such creations. Stigmergy is the key to such construction [16, 20].

Figure 2.2: Termite Mound and Bees Hive [25, 26]

5. Division of labour: In the insect community, several insect species display division of labour and specialization where particular entities perform various activities simultaneously [20]. Example are seen in the bees family where different categories of bees carry out different assignments simultaneously.

The emergent and collective social behaviours of different animals and insects are modelled into algorithms termed SI techniques which are now very prominent in solving optimization problems.

2.3.3 Swarm Intelligence Techniques

Optimization techniques inspired by SI have attracted interest and gained popularity and attention because of their flexibility, versatility and efficiency in solving real world applications. The techniques are described by the working mechanism that imitates the collective behaviour of social animals in a group where there is no central coordination. Since optimization is everywhere, SI algorithms are applicable across all discipline. People are always optimizing something, either to

minimize the cost, time and resources or to maximize profit, output, performance and efficiency. Examples of SI techniques inspired by social entities, animals and insects' behaviours are shown in Table 2.1. PSO is one of the most popular SI technique in the literature; it is briefly described in section 2.3.4.

Table 2.1: Swarm Intelligence Techniques

NO	Algorithm	Author(s)	Year	Inspiration
1	Particle swarm optimization (PSO)	Kennedy and Eberhart	1995	Bird flocking and fish schooling [27]
2	Ant colony optimization (ACO)	Margo Dorigo	1992	Ants food foraging [28]
3	Bee colony optimization (BCO)	Pham et al	2005	Food foraging of a swarm of honey bees [29]
4	Bacteria foraging optimization algorithm (BFOA)	Passino K.	2002	Chemotaxis behaviour of bacteria [30]
5	Glowworm swarm optimization (GSO)	Krishnanand and Ghose	2005	Behavioural patterns of glow worms [31]
6	Bioluminescent swarm optimization (BSO)	Oliveira et al.	2011	Luciferin-based attraction [32]
7	Firefly Algorithm (FA)	Yang X.	2009	Flashing behaviour of fireflies [33]
8	Roach infestation optimization (RIO)	Havens et al.	2008	Cockroach infestation [34]
9	Cockroach Swarm Optimization (CSO)	C. ZhaoHui and T. HaiYan	2010	cockroach swarming[35]
9	Grey wolf optimiser(GWO)	S. Mirjalili, , S. M. Mirjalili and A. Lewis	2014	mimics the leadership hierarchy and hunting mechanism of gray wolves [36]
10	Cuckoo search (CS)	Yang Xin-She and Deb Suash	2009	Brooding behaviour of some cuckoo species [37]
11	Intelligent water drop (IWD)	Shah-Hosseini Hammed	2007	Inspired by natural rivers and how they find almost optimal paths to their destination [38]
12	Gravitational search algorithm (GSA)	Rashedi, Nezamabadipour and Sayazdi	2009	Based on the law of gravity and notion of mass interaction [39]
13	Altruism	Foaster, Wenselees and Ratnieks	2006	Based on Hamilton's rule of kin selection [40]
14	Artificial immune system (AIS)	De Castro, Von Zubens and Nicosia and Cutellos	2002	Based on abstract structure and function of immune system [41]
15	Charged system search (CSS)	Kaveh A. and Talatahari S.	2010	Based on some principle from physics and mechanics [42]
16	Bat Algorithm (BA)	X. S. Yang	2010	inspired by the echolocation [43] behaviour of microbats
17	Sheep flocks Algorithm (SFA)	Nara, Takeyama and Kim	1999	sheep flocks heredity [44]

2.3.4 Particle Swarm Optimization

The inspiration of PSO is from the coordinated interaction of birds flocking and fish schooling [27, 45, 46, 47, 48]. It was originally introduced in 1995 by Eberhart and Kennedy [27]. PSO is a swarm of particles where a particle is a prospective solution to a problem that requires a solution. Individual particles emulate the success of one another based on socio-cognitive factors as they flow through the hyper-dimensional search space of the problem and change particles position in the search region [9, 27, 45, 46, 47, 48]. Individual particles make decisions based on their experience with the environment through individual learning called the cognitive component and experience with their neighbours through cultural transmission which is referred to as the social component. The decision making based on the local interaction with the immediate environment and neighbours result to emergent behaviour of particles in a swarm [21, 50].

The initial values of the PSO control parameters (cognitive factor, social factor and inertial weight), the swarm size, and the maximum iteration number determine the success of the algorithm in finding a global optimum. The velocity of particle is v_i, its present position is $present_i$, the local and global best solutions are $pbest_i$ and $gbest_i$ respectively, rand () is a number that is picked randomly between (0,1) and the learning factors are $c1$ and $c2$.

The computational steps given below show the PSO algorithm in the simplest form.

1. Initialize particle
2. Compute fitness value
3. Calculate particle velocity by equation
   $v_i = v_i + \text{c1} * \text{rand}() * (pbest_i - present_i) + \text{c2} * \text{rand}() * (gbest_i - present_i)$
4. Update particle position by equation
   $present_i = present_i + v_i$
5. If termination condition is reached STOP otherwise return to step 2

PSO has offered solutions to several optimization problems in the literature which include: Power and management processes [51, 52], and combinatorial optimization problems [53].

2.3.5 Discrete Particle Swarm Optimization

The PSO algorithm for combinatorial optimization problems (COP) was first presented by Kennedy and Eberhart [54] where binary sequences were used to represent particles. Velocity of the particle is expressed by the number of bits changed per iteration, or the distance between the

particles at different times [54]. Particle trajectories are changes in the probability that a coordinate assumes value zero or one [54]. PSO continuous space was transformed into states zero or one using sigmoid function.

$$s(x_{ij}) = \frac{1}{1+exp^{-}x_{ij}}$$

$$x_{ij} = \begin{cases} 1 \;\; if \;\; rand() \leq \frac{1}{1+exp^{-}x_{ij}} \\ 0 \;\; otherwise \end{cases} \tag{2.1}$$

$i = 1...N,\ j = i..D$, where N is the swarm size, D is the dimension size and $rand()$ is a randomly generated number between zero and one [54, 55, 56]

The original binary PSO was modified by many researchers in the literature, to suit more than two valued problems including [55, 56, 57]. Osadciw and Veeramachaneni presented a multi valued discrete PSO [55] where transition of particle position is probabilistic like binary PSO [54] described above. Osadciw and Veeramachaneni used sigmoid with additional operators for logistic transformation of particle states. The range of variable is from 0 to $M-1$, where M denotes number base system which can be binary, ternary, denary etc.

$$x_i = round(s(x_{ij}) + (M-1) \times \sigma \times randn(1))$$

The number is rounded to the closest discrete variable with the end point fixed.
where, a sigmoid

$$s(x_{ij}) = \frac{M}{1+e^{-}x_{ij}} \tag{2.2}$$

$$x_{ij} = \begin{cases} M-1, & if \;\; x_{ij} > M-1 \\ 0, & if \;\; x_{ij} < 0 \end{cases} \tag{2.3}$$

Some other methods, different from the sigmoid function described above were used in different SI techniques to discretize a continuous state by transformation of values into a number of possible states. This includes Random key [58, 59], smallest position value [60, 61], modified position equation [62, 63], great value priority [64], and nearest integer [65].

2.3.6 Swarm Intelligence Techniques for Discrete Optimization Problems

Real world optimization problems include diversity of issues in scheduling, COP, transportation, inventory planning, production planning, communication operations, computer operations etc. Section 2.3.7 gives a brief discussion on the SI techniques for the travelling sales-man problem (TSP).

2.3.7 Swarm Intelligence Techniques for TSP

TSP is one of the most popular COP which can be described as given a set of cities, and distances between the cities; the goal of TSP is to find the shortest tour by visiting every city only once and returning to the starting city [66, 67, 68, 69]. A situation where the distance between two cities does not depend on the direction is referred to as a symmetric TSP, and if otherwise asymmetric. TSP finds application in various real life situations such as vehicle routing, robot control, chronological sequencing, x-ray crystallography, logistics and automatic drilling boards [66]. To improve several optimization methods in recent years, TSP has been considered for comparison [66]. Many studies on SI for TSP have been presented, including ACO [70], dynamic Gaussian process regression [68], and PSO [66, 67].

2.3.8 TSP Formulation

Supposing the number of cities to visit is n, the distances between cities are stored in a distance matrix (costs) $Dmat = d_{ij}$ where d_{ij} is the distance between city i and city j, $(i, j = i_1, i_2, i_3 ... i_n)$. Finding the permutation of $(i_1, i_2, i_3 .... i_n)$ integers from 1 to n which minimize the expression: $d_{i1i2} + d_{i2i3} + d_{i3i4} ... + d_{inil}$ is the problem to be optimized. If $d_{ij} = d_{ji}$, the problem is symmetric otherwise asymmetric [66].

Basically in TSP, a tour can be described as the traverse of a salesman across cities, visiting each city once and returning to starting city. The distance between the cities is computed by minimizing the tour length. A tour can be expressed using a recurring permutation π where π_i denotes city i on the tour. TSP is the optimization problem to find a permutation π that minimizes the tour length as shown in equation 2.4 [77].

$$\sum_{i=1}^{n} d_i \pi_i \tag{2.4}$$

Other formulations of TSP, different from permutation problems are graph theoretical problems, integer programming, and linear programming. TSP can be formulated as a graph theoretical problem where it is represented as a complete graph $G = (V, E)$ [71]. The goal is to find a Hamiltonian cycle which visits each graph node exactly once, with the least weight in the graph [68, 71]. Cities are denoted by the node set $V = \{1, 2, ...n\}$ and each edge $e_i \in E$ is the distance between nodes. This is an associated weight w_i to each edge $e_i \in E$.

TSP formulation as integer programming is based on assignment problems with sub-tour elimination constraint as shown below:

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{2.5}$$

$$Subject\ \ to\ \ \sum_{\substack{i=1 \\ i \neq j}}^{n} x_{ij} = 1,\ j = 1, ..., n,$$
$$\sum_{j=1}^{n} x_{ij} = 1,\ j = 1, ..., n,$$
$$x_{ij} = 0\ \ or\ \ 1$$

sub-tour eliminated.

Solution matrix $X = (x_{ij})$ of the assignment problem denotes the tour or a collection of sub-tours. Only the edges that correspond to elements $x_{ij} = 1$ are on tour. The sub-tour elimination constrained added restricts the solution to only proper tours [71].

TSP formulation as linear programming is expressed as:

$$min\ \ \sum_{i=1}^{m} w_i x_i = W^T X \tag{2.6}$$

$$Subject\ \ to\ \ x \in S$$

where m denotes number of edges $e_i \in G$ (Hamiltonian cycle G), $w_i \in W$ denotes the weight of edge e_i and X denotes the incidence vector showing presence and absence of edges in the tour [71].

Among the existing SI techniques shown in Table 2.1, this thesis focuses on RBA (RIO and CSO). A review of cockroach social behaviour and existing RBA is presented in sections 2.4 and 2.5.

2.4 Foundation of Cockroach Swarm

Cockroaches are social insects that exhibit extraordinary variations in their social organization; they are versatile in reproductive mode and feeding habits [72]. Cockroaches naturally seek dark shelters that are rich in nutritional resources. They rest in groups in dark shelters and forage at night in search of food and water, and they remain faithful to their shelter as long as there are enough rich resources in their surroundings to exploit [73]. They are insects that are large in size, and can reach nine centimetres in length and weigh about 30 grams. They have relatively small heads with broad and flattened bodies, large compound eyes, a pair of antennae and two pairs of wings [74]. Cockroaches vary in over four thousand different colours and species [72]. Cockroaches are among the most ancient animals on earth [75]. They are the fastest insect on earth, and can run up to twelve feet in a second [75]. The social world of cockroaches is controlled by the chemical called pheromones, and by responding to pheromone stimuli, cockroaches are able to perform some social functions [72].

2.4.1 Cockroach Social Behaviours

Cockroaches interact with peers and respond to their immediate environment, and make decisions based on their interactions such as finding a good food source, locating other friends, shelter selection, rapid dispersion when danger is noticed, and eating one another when food is scarce. The emergent behaviour in cockroaches is the result of group-based decision making. Individual cockroaches act mainly on local information received from interaction with their peers and their immediate environment [21]. Different studies, including [73, 76, 77, 78, 79, 80, 81, 82], have revealed the emergent social behaviour of cockroaches. Experiments were carried out by Jeanson and his group [77] where a number of cockroaches were tracked in an enclosed circular arena. It was discovered that cockroaches first exhibit a wall-following behaviour along the edge of the arena and some individual cockroaches choose to explore the central zone of the enclosed arena.

Cockroaches feel the presence of one another via their antennae which are 3mm long; they perceive each other within a distance of 6mm, that is, head to head inter-individual distance (see Figure 2.3). They stop and interact with one another and share information on the best food source [77]. In the process more individuals will also stop and join the group, which will eventually lead to aggregation. Jeanson et al., obtained the probability of stopping, for a moving cockroach when N individuals is perceived. It is defined within a detection radius $(1 \leq N \leq 3)$ [77]. The stopped cockroach will aggregate with N individuals $(2 \leq N \leq 3)$ detect a moving cockroach within the detection radius. Cockroaches prefer to aggregate, it was discovered that the probability of moving away from a group is very low. The probability per unit time of stopping when encountering N friends is $1/T_{stop},N$: 0.49/s for (N=1), 0.63/s for (N=2) and 0.65/s for (N=3) [77].

Halloy et.al., [76] and Ame et.al., [78] carried out experiments to show collective decision making of groups of cockroaches on shelter selection. Halloy et. al., performed their experiment with both natural cockroaches and artificial cockroaches (robots scented with a concentration of real cockroach odour). The natural and artificial cockroaches were equally attracted to one another [76]. In the series of experiments performed on shelter selection, both natural and robot cockroaches interacted, and showed a preference for aggregating in dark shelters, which supports the hypothesis that cockroaches make use of just two pieces of information (how dark the place is and how many other cockroaches are there) when choosing where to go. Likewise, it supports the hypothesis that collective behaviour is aggregated from simple decentralized behaviour. Halloy et al., [76], Jeanson et al., [77] and Ame et al., [78] concluded that cockroach aggregations are self-organized systems, resulting from interactions between individuals following simple rules based on local information.

Cockroaches prefer to swarm on food rather than to dine alone. The experiment carried out by Mathieu Lihoreau and his group at Queen Mary, University of London [82] showed that cock-

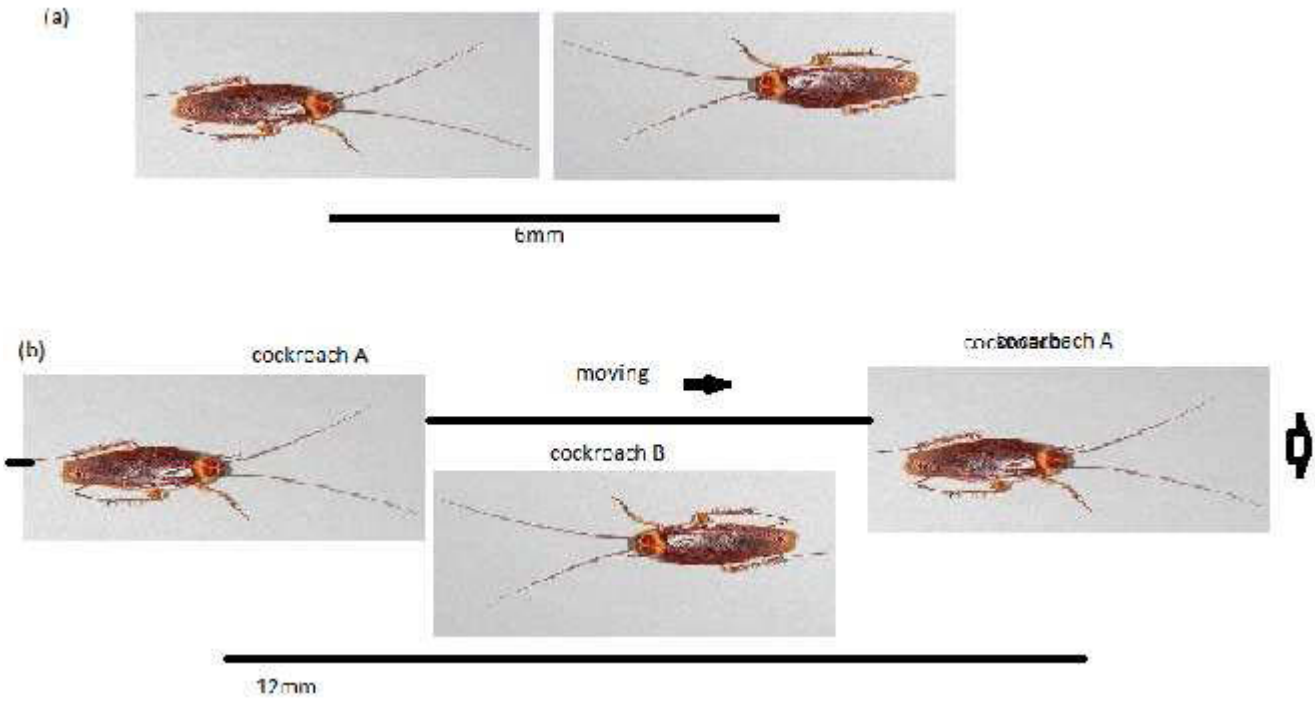

Figure 2.3: (a) Cockroach detect each other by head to head inter-individual distance of less or equal to 6mm. (b) A moving cockroach A encounters a stopped cockroach B at a maximum of 12mm (Adapted from [77]).

roaches follow a crowd when it comes to dining. In the experiments, they presented two piles of food in a closed arena with a number of cockroaches. They discovered after a while that the cockroaches packed themselves on only one pile, instead of dividing into two groups. The more crowded and cockroach infested the food, the more cockroaches were joining the excessive fling. The more the cockroaches gathered on a pile of food the more attractive the food becomes and the longer they stayed at the source [73, 82]. Figure 2.4 shows a number of cockroaches dinning together on a piece of bread.

Miller and Koehler [81] investigated trail following behaviour of cockroaches. Cockroaches leave chemical trails in their faeces which transmit bacteria onto surfaces. They also emit airborne pheromones for swarming and mating. The use of chemical trails for directional orientation is widespread among social insects. Cockroaches exhibit trail-following behaviour, by using physical landmarks and visual clues for directional orientation during the daylight hours, while directional response to odours and chemical stimuli allow for free movement in the dark [81]. The German cockroach is a gregarious, nocturnal insect whose directional locomotory behaviour is influenced by chemical compounds found in their faecal material. It was discovered that German cockroaches leave and return to their harbourage using chemical substances in their faeces for navigation [81].

Figure 2.4: Swarm of cockroaches [83]

Miller and Koehler [81] carried out an experiment, where strips of chromatography paper that had been treated with faecal extract were used to evaluate cockroach trail-following behaviour. Chromatographic paper was cut into uniform strips. For each test, two strips were individually treated; one with a solution of methanol and faecal extract, the other with an equal amount of methanol. These strips of paper were arranged in a shaped configuration and taped to the floor of the test arena. For each replicate of the test, a harbourage was placed at the vertex of the V configuration and opened to release one cockroach. The cockroach was allowed to exit and follow its own course. Each test used 10 cockroaches, and after exiting the harbourage, cockroach movements were captured on a low-light video surveillance camera and recorded. The results of their experiment supported the hypothesis that chemicals in the cockroach faecal material did influence directional orientation [81]. Other cockroaches follow these trails to discover sources of food and water, and also discover where other cockroaches are hiding. Thus, cockroaches can exhibit emergent behaviour in which group or swarm behaviour emerges from a simple set of individual interactions. Some of the emergent behaviour of cockroach are presented below.

2.4.2 Foraging

Cockroaches use recruitment mechanisms to forage. The recruitment mechanisms can be either direct or indirect [50], mass recruitment via a faecal chemical trail is a good example of indirect recruitment. The recruiter and the recruits are not in physical contact with each other, the recruiter deposits faecal pheromone on their way back from a profitable food source and recruits simply follow the trail; communication is via modulation of the environment. Using a direct recruitment mechanism the recruiter transfers information to the recruits [50]. A good example of a direct recruitment mechanism is when a cockroach comes within a detection radius of another cockroach agent via an encounter with their antennae, then perceive each other within a distance of 6mm, (a cockroach antennae is 3mm long) that is head to head inter-individual distance. There is a probability that the cockroaches will stop, socialize and communicate their knowledge of the search space to one another [77].

The foraging behaviour of recruited individuals is influenced by positive feedback enhanced by a faecal trail [82] which leads to the selection of a food location and exploitation/exploration of most the rich food sources. Food closest to harbourage is exploited first, and when depleted, cockroaches forage further and explore other sites [72]. Socially foraging cockroaches, locate food sources and eat the food at the food source [73]. Australian soil-burrowing cockroaches forage, locate food source and transport or relocate a quantity of food to their harbourage. The gathered food is eaten by both the forager and any young offspring in the nest [72]. Food relocation is a proximate mechanism for obtaining and securing food, which helps the cockroaches feed at their leisure in a location relatively safe from predators [72].

2.4.3 Exploitation and Exploration

Exploitation is the use of existing information while exploration is the collection of new information about food sources [50]. When a cockroach exploits a profitable food source, there may still be other, undiscovered, profitable sources that are yet to be exploited [50]. One of the mechanisms for exploration is food foraging. Cockroach forage for food locations near their shelter first, and when that food is depleted, they forage further and explore new food locations. Another mechanism which is a kind of direct mechanism is called vigilance behaviour, which occurs at intervals of time when cockroaches sense danger or a predator and disperse rapidly [72]. Cockroaches rapidly disperse to different locations, this helps them to explore and exploit other areas in the search space for good solutions. Vigilance behaviour enhances high levels of diversity as cockroaches disperse to several locations in the search space and explore the new sites; this helps to prevent local convergence.

2.4.4 Shelter Selection

Cockroaches make use of different criteria for harbourage site selection. Mainly, cockroaches prefer a shelter close to nutritional sources; a site with adequate food and water [72]. The shelter's physical properties such as darkness, height, temperature, texture, orientation of surfaces and the size of the harbourage are influential [72, 73]. The body size of the cockroach also determines shelter selection, as cockroaches may segregate if the harbourage is not spacious enough in terms of cockroach population and shelter height [72]. Cockroaches move to new shelter when the previous shelter is overcrowded or food is depleted [73]. Exploring cockroaches perceive an occupied shelter and switch from search mode to join and settle. The larger the cockroach population the more new cockroaches are joining the group and by this retention effect an aggregation is formed, which eventually leads to the selection of a unique shelter by the entire group [73, 82].

2.4.5 Aggregation

Cockroaches aggregate as a result of attraction of the chemical material deposited on their faeces [84]. Faecal material is identified as the source of a cue [85, 86]. Mechanisms by which a group of cockroaches may gather, was suggested by Kavanaugh [87] as independent individual responses to environmental gradients. This leads to aggregation in an abiotically optimum location; and the individual response to stimuli provided by other individuals, leads to group formation at a common location. Jeanson et al., and Halloy et al., later re-investigated and experimented with this approach [76, 77] and concluded that cockroach aggregations are self-organized systems, resulting from interactions between individuals following simple rules based on local information.

Cockroach larvae exhibits a collective complex behaviour that results in aggregation; the individual decisions of cockroaches transform the collective behaviour of the entire group, which support the hypothesis that collective behaviour is aggregated from simple decentralized behaviour [77].

Female cockroaches exhibit oviposition behaviour, they deposit their oothecae (eggs) with a drop of genital fluid (oviposition pheromone) close to a food supply [88]. The pheromone attracts other females to lay their eggs in the site which eventually results in a cluster of eggs which hatch to form an aggregation of cockroaches [88].

2.4.6 Nourishment and Food Location Selection

Cockroach decisions on food source depends on the nutritional values, distance from the shelter and presence of conspecifics [73]. Aggregation at food sources is enhanced by a social facilitation process for feeding, as cockroaches in large populations feed longer than those in small population

group sizes [73, 82]. Cockroaches exhibit the simplest form of food related grouping behavior; they, cue on conspecifics for food information [76, 89].

Willeyto et al. [90] described the nourishment behaviour as a means of signalling to the unassociated individuals the location of food supply. Cockroaches forage from their shelter to the location of food supply and return back to their shelter. The food patches closer to the shelter are exhausted first before foraging further [91, 92]. Trail pheromones facilitate the movement of cockroaches from their shelter to the food location. The size of a cockroach aggregation is ultimately determined by the availability of adequate food and water [84].

Favourable habitats can result in enormous populations, with the constant renewal of food resources in the surroundings of the shelter amounting to large aggregation [72]. Cockroaches leave behind at the feeding sites a variety of residue in the form of saliva, glandular deposits and faecal pellets, which put a mark on the site and make it more attractive than an unmarked and unvisited site [72]. The traffic of conspecifics makes the place visible for other foraging cockroaches, whether cockroaches are present at the marked site or not.

Williams et al., Cockroaches ecology, behaviour and natural history, states that, cockroaches can feed on anything such as soap, paper, clothes, faeces, human food, their cast-off skin and egg capsules, and will eat one another, with the strong ones eating the weak [72]. When a cockroach is hungry and there is no food, in the surrounding area, the strong ones will eat the weak ones.

Cockroach social behaviours were modelled and presented in the literature as SI optimization algorithms that establish the relationship between social behaviour and optimization paradigms and explored the processes therein to solve optimization problems [34, 35]. The RBAs found in the literature are presented in section 2.5.

2.5 Roach-based Algorithms

2.5.1 Roach Infestation Optimization

RIO was originally introduced by Haven et al., [34] as a cockroach inspired algorithm. RIO was adapted from the traditional PSO algorithm, and therefore it has some elements similar to PSO. A hungry version of PSO and RIO were also described [34]. It is assumed that cockroaches begin as individuals and are ruled by only the find dark behaviour [34]. Whenever a cockroach agent encounters another cockroach agent, it stops and socialises and shares information about the darkest known location. “An encounter is defined as two cockroaches coming within 6mm of one another and a collision is defined as when a moving cockroach encounters a stopped cockroach that is a member of an aggregate” [77]. When a cockroach is hungry it leaves friends and comfortable shelter and searches for food [72, 34]. A hunger counter is defined for each cockroach agent, once

it is reached, the cockroach is transported to a food location within the search region. The find food behaviour causes population diversity, and prevents local convergence [34]. RIO is described with three behaviours [34]:

1. Find dark: "Cockroaches search for the darkest location in the search space and the fitness value is directly proportional to the level of darkness" [34]. Find dark behaviour:

$$v_i = c_0 v_i + c_{max} R_1 (p_i - x_i) \tag{2.7}$$

   where v_i is the velocity of the ith agent, x_i is the current location found by the ith agent, p_i is the best location found by the ith agent, c_0, c_{max} are cockroach parameter, R_1 is a vector of uniform random numbers, and $(p_i - x_i)$ is a velocity change in the direction of the darkest known location for that agent.

2. Find friend: If a cockroach comes within a detection radius of another cockroach agent, it will stop, socialize and share information of the darkest known location by setting local location l

$$l_i = l_j = arg_k min \{F(p_k)\}, k = \{i, j\} \tag{2.8}$$

   where i, j are the indices of the two socializing cockroaches and p_k is the personal best known darkest location. Find darkest model is extended to include find friend behaviour:

$$v_i = c_0 v_i + c_{max}.R_1 (p_i - x_i) + C_{max}.R_2 (l_i - x_i) \tag{2.9}$$

$$x_i = x_i + v_i \tag{2.10}$$

3. Find food: When a cockroach experiences hunger, it leaves the comfort of friends and shelter to search for food and is taken to a food source b that is positioned randomly in the search region.

$$x_i = b \tag{2.11}$$

Algorithm 1 shows the computational steps of RIO.

Algorithm 1 Roach Infestation Optimization Algorithm

INPUT: Fitness function: $f(x), x \in R^D$

Initialization of cockroach population, food location and parameter setting.

for t=1 to Tmax **do**

$M = [M_{ij}] = [||\vec{x}_j - \vec{x}_k||_2]$

$d_g = median\{M_{jk} \in M : 1 \leq j < k \leq N\}$

for i=1 to N **do**

if $f(\vec{x}_i) < f(\vec{p}_i)$ **then**

$\vec{p}_i = \vec{x}_i$

end if

Compute the neighbours of roach i as

$\{j\} = \{k : 1 \leq k \leq N, k \neq i, M_{ik} < d_g\}$

N_i = number of neighbour of $|\{j\}|$

for $q = 1 to N_i$ **do**

if $rand[0,1] < A_{min}\{N,3\}$ **then**

$\vec{l}_i = \vec{l}_j = arg\ min_k\{f(\vec{p}_k)\}, k = \{i, j_q\}$

end if

end for

if $hunger_i < t_{hunger}$ **then**

$\vec{v}_i = C0\vec{v}_i + CmaxR\vec{1}(\vec{p}_i - \vec{x}_i) + CmaxR\vec{1}(\vec{l}_i - \vec{x}_i)$

$\vec{x}_i = \vec{x}_i + \vec{v}_i$

else

$\vec{x}_i = \vec{b}$

Relocate food location randomly

$hunger_i = 0$

end if

if Hungry **then**

Increment $hunger_i$ counters

end if

Check termination condition

end for

end for

RIO achieved good results when applied to benchmark functions and compared to PSO [34]. Hendrawan and Murase proposed a discrete RIO for a feature selection problem using a neural network as a predictive tool where each roach was represented by a binary encoding vector and each dimension represented a feature value of 0 or 1. It is 0 when the corresponding feature is not selected and 1 if otherwise. A mutation operator is used for updating the position of each roach by updating velocity. Two point crossover is used for updating the cognitive and social parts of the cockroach [93].

2.5.2 Cockroach Swarm Optimization

ZhaoHui and HaiYan presented cockroach swarm optimization (CSO) which was inspired by the social behaviour of cockroaches [35]. They show cockroach behaviour in three components: chase-swarming, dispersion and ruthless. In the CSO model there is a cockroach cluster containing N cockroach individuals in the D-dimensional search space R^D where the ith individual is a representation of a D-dimensional vector $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$, $i = 1, 2, ..., N$. Individual cockroach location is a potential solution.

CSO model is presented as [35]:

1. Chase-Swarming behaviour:
   Chase-Swarming behaviour is described as each individual x_i will chase the local optimum individual p_i within its visual scope. Of course, such chasing behaviour also produced cluster simultaneously. There is an assumption, that when an individual is the best one within its visual scope, it will chase the global optimum individual p_g [35]. The model is:

$$x_i = \begin{cases} x_i + step.rand.(p_i - x_i), x_i \neq p_i \\ x_i + step.rand.(p_g - x_i), x_i = p_i \end{cases} \tag{2.12}$$

   where x_i is the cockroach location, $step$ is a fixed parameter, $rand$ is a random number within [0,1],

$$p_i = Opt_j \left\{ x_j, Suchthat | x_i - x_j | \leq visual, j = 1, 2, ..., N \right\} \tag{2.13}$$

   where p_i is the personal best optimum, $visual$ denotes the visual distance of cockroach, N is the population size.

$$(i = 1, 2, ..., N), p_g = Opt_i \{ x_i, i = 1, 2, ..., N \} \tag{2.14}$$

   where p_g is the global optimum individual.

2. Dispersing Behaviour
   Dispersing behaviour is described as when individual cockroach disperses randomly at intervals of time in order to maintain the diversity of current individuals. The model is:

$$x_i = x_i + rand(1, D), i = 1, 2, ..., N \tag{2.15}$$

   where $rand(1, D)$ is a D-dimensional random vector that can be set within a certain range, D is the space dimension.

3. Ruthless

Ruthless behaviour is described as the current best cockroach replacement of the one that has been randomly picked at intervals of time. The stronger eats the weak one. The model is:

$$x_k = p_g \tag{2.16}$$

where k is a random integer within [1,N], p_g is the global best cockroach position.

There is the possibility of CSO falling into local optimum when imitating chase-swarming behaviour due to the formation of cockroach cluster in a location. Dispersing behaviour may keep individual diversity and ruthlessness can improve searching capabilities [35].

2.5.3 A Modified Cockroach Swarm Optimization

ZhaoHui introduced an inertial weight w into the chase-swarming component of CSO [94].

Chase-Swarming behaviour:

$$x_i = \begin{cases} w.x_i + step.rand.(p_i - x_i), x_i \neq p_i \\ w.x_i + step.rand.(p_g - x_i), x_i = p_i \end{cases} \tag{2.17}$$

The computational steps of MCSO is shown in algorithm 2.

CSO algorithm shows competitive results in the literature, where it was applied to some optimization problems and compared to PSO. Its application as reported in the literature includes: solving benchmark functions [35, 94]; Cheng et. al., applied CSO to Travelling salesman problem [95]; ZhaoHui and HaiYan applied CSO to vehicle routing [96]; Wu and Wu proposed a cockroach genetic algorithm for estimating the parameters of biological system in showing the net interactive effect of S-system where cockroach behaviour was imitated and embedded into advanced genetic algorithm to increase exploration and exploitation abilities [97].

2.6 Related Works

A vast new algorithms can be constructed by simply introducing some techniques, models and parameters to existing algorithm for performance improvement. For example, an algorithm can be extended with certain techniques, to perform a rapid local search to refine a solution when necessary. Chapter 3 investigates several techniques for constructing improved RBAs; some of the techniques have been proved successful in previous studies when applied to SI and EC algorithms. A constriction factor was initially introduced into PSO by Clerc and Kennedy to avoid swarm explosion, maintain swarm stability, and enhance the convergence of the algorithm in a multidimensional complex space [98]. Other researchers also investigated constriction factors in studies

Algorithm 2 COCKROACH SWARM OPTIMIZATION [94]

Objective function: $f(x), x \in R^D$
set parameters and generate an initial population of cockroach
set $p_g = x_1$
for $i = 2 \ \ to \ \ N$ **do**
  if $f(x_i) < f(p_g)$ **then**
    $p_g = x_i$
  end if
end for
for $t = 1 \ \ to \ \ T_{max}$ **do**
  for $i = 1 \ \ to \ \ N$ **do**
    for $j = 1 \ \ to \ \ N$ **do**
      if $abs(x_i - x_j) < visual; f(x_j) < f(x_i)$ **then**
        $p_i = x_i$
      end if
    end for
    if $p_i == x_i$ **then**
      $x_i = w.x_i + step * rand * (p_g - x_i)$
    else
      $x_i = w.x_i + step * rand * (p_i - x_i)$
    end if
    if $f(x_i) < f(p_g)$ **then**
      $p_g = x_i$
    end if
  end for
  for $i = 1 \ \ to \ \ N$ **do**
    $x_i = x_i + rand(1, D)$
    if $f(x_i) < f(p_g)$ **then**
      $p_g = x_i$
    end if
  end for
  $k = randint([1, N])$
  $x_k = p_g$
end for

including [99, 100, 101]. The stochastic constriction RBA shown in Chapter 3 was inspired by the studies on constriction factors.

Cai et al., investigated the effect of simple Euler stochastic dynamic step length in PSO [102]. The step length technique improved the searching ability of PSO and maintained a balance between exploration and exploitation. The dynamic step-size adaptation RBA that is presented in Chapter 3 was inspired by studies [102].

Laumanns et al., introduced the basic concept of the predator-prey evolution method [103]. They described a technique that imitated natural predator-prey habits, when a predator kills the weakest prey in its vicinity and a relatively strong prey is created through evolution processes to replace the weak prey. Other studies on this technique include [104, 105, 106, 107, 108, 109, 110, 111]. Chowdury et al., presented a modified predator-prey (MPP) algorithm using blend crossover [104, 109]. The Predator-prey RBA presented in Chapter 3 was inspired by these studies.

SI techniques for transforming a continuous space into binary or multi-valued discrete space in order to offer solutions to discrete optimization problems were investigated in many studies including [54, 55, 56, 57] (See section 2.3.5). A study on multi-valued space discrete cockroach swarm presented in Chapter 3 was inspired by the study carried out by Osadciw and Veeramachaneni on multi valued discrete PSO [55].

Chapter 3

Improved Roach-based Models

3.1 Introduction

This chapter introduces different methods and models for constructing improved RBAs for global optimization problems. The existing RBA presented in section 2.5 were originally designed to provide solutions to the global optimization problem, that is, finding the best solution among several others. The methods and models introduced in this chapter aims to improve the strength of RBA in achieving the goals of global optimization which is known as exploitation and exploration. The first model called stochastic constriction cockroach swarm optimization model, described in section 3.2.1 is similar to "Clerc and Kennedy's particle swarm-explosion, stability, and convergence in a multidimensional complex space" [98]. It uses a stochastic constriction instead of a static one as shown in section 3.2. The second model, described in section 3.3.1 adds a component, that was constructed using a partial differential equation, shown in section 3.3 to the existing CSO, that was originally designed with three components and is called an improved cockroach swarm optimization. The third model described in section 3.4.1 introduces a simple Euler method expressed in section 3.4 into RIO and is called a dynamic step-size adaptation roach infestation optimization. It is similar to Cai et. al., "stochastic dynamic step length particle swarm optimization" [102]. The fourth model called modified roach infestation optimization, described in section 3.6.1 adds two components to the existing RIO model that was originally made up of three components. The two added components were constructed using a partial differential equation and predator-prey evolution methods respectively as shown in sections 3.5 and 3.6. The fifth Model called an adaptive cockroach swarm algorithm was also constructed by modifying the existing CSO algorithm using a predator-prey evolution with crossover and mutation techniques described in section 3.7. The sixth model, described in section 3.8.1 further extended the existing CSO, using a known technique, shown in section 3.8 to construct a multi-valued discrete space CSO that can be used

for optimization problems with any discrete value or base system (binary, ternary, quaternary etc.). It is similar to Osadciw and Veeramachaneni's multi-valued discrete PSO.

3.2 Stochastic Constriction Factor

A constriction factor enhances convergence of an algorithm, and improves both local and global searching abilities. The idea for a constriction factor came from the well-known idea of constriction factor [98, 99, 100, 101] introduced by Clerc and Kennedy into PSO to allow control over the dynamical characteristics of the particle swarm, including its exploration versus exploitation propensities [98].

$$\chi = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}$$

where $\varphi = c_1 + c_2$, $\varphi > 4.0$, $c1$ is the recognition factor and $c2$ is the social factor. Constant χ was described by Shi as a constant which is approximately 0.729 and φ is commonly set to 4.1 [99]. An algorithm with constriction constant 0.729 is equivalent to algorithm of inertia weight 0.729 with $c1 = c2 = 1.49445$ [99]. The comparison study of the effect of the constriction factor and inertial weight on PSO was carried out [100, 101] where it was shown empirically that constriction PSO performs better than inertia weight PSO. This motivated the introduction of stochastic constriction factor (SCF) into CSO algorithm to control cockroach movement during the swarming process (shown in computational steps of section 3.2.1 below) in order to avoid swarm explosion.

SCF enables the algorithm to avoid explosion in regions outside of the search space. It also enhances local and global searches of the algorithm; a cockroach is able to exploit the local neighbourhood and explore the whole search region. SCF allows the generation of different values as constriction factors in each iteration and was found to be efficient, as revealed through the results of our empirical studies. Our results presented in section 5.2 show improved speed and convergence.

3.2.1 Stochastic Constriction Cockroach Swarm Optimization Model

SCF was incorporated into the CSO algorithm to maintain swarm stability and a balance of exploitation and exploration. To achieve this aim, CSO chase-swarming behaviour [35, 94] presented in sections 2.5.2 and 2.5.3 is modified in this work with the introduction of SCF ξ. In each iteration, ξ randomly takes values between zero and one. ξ controls entire cockroach movement, not only cockroach position as with inertial weight w of CSO [94]. In equation (3.1), stochastic

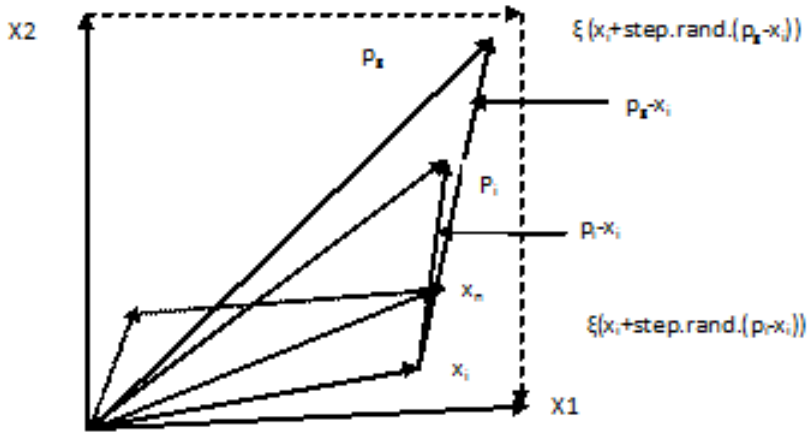

Figure 3.1: Constriction factor constrained cockroach movement within the search region

constriction factor ξ restricts the movement of cockroach swarms within the given search region.

$$x_{i+1} = \begin{cases} \xi(x_i + step.rand.(p_i - x_i)), x_i \neq p_i \\ \xi(x_i + step.rand.(p_g - x_i)), x_i = p_i \end{cases} \tag{3.1}$$

Figure 3.1 depicts the restriction of movement of cockroach within the search space by SCF ξ as x_i always approach to p_i and p_i approach to p_g. The constant $step$ and randomly generated number $rand$ manages the step size of cockroach in iteration towards the optimum individual. Where x_i is the cockroach's present position, x_{i+1} is the new position located by the cockroach, p_i is the personal best optimum and p_g is the global best optimum.

We proposed a Stochastic Constriction Cockroach Swarm Optimization (SCCSO) for Multi-dimensional Space Function Problems [112]. The algorithmic steps for SCCSO is illustrated in Algorithm 3 and its computational steps given as:

1. Initialise cockroach swarm with uniform distributed random numbers and set all parameters values.
2. Locate the best position p_i and p_g using equations (2.13) and (2.14).
3. Perform chase-swarming using equation (3.1).
4. Carry out dispersion behaviour using equation (2.15).
5. Exhibit ruthless behaviour using equation (2.16).
6. Repeat the loop until stopping criteria is reached.

Algorithm 3 Stochastic Constriction Cockroach Swarm Optimization Algorithm

INPUT: Fitness function:$f(x), x \in R^D$
set parameters and generate an initial population of cockroach
set$p_g = x_1$
for $i = 2$ *to* N **do**
  if $f(x_i) < f(p_g)$ **then**
    $p_g = x_i$
  end if
end for
for $t = 1$ *to* T_{max} **do**
  for $i = 1$ *to* N **do**
    for $j = 1$ *to* N **do**
      if $abs(x_i - x_j) < visual; f(x_j) < f(x_i)$ **then**
        $p_i = x_i$
      end if
    end for
    if $p_i == x_i$ **then**
      $x_{i+1} = \xi(x_i + step.rand.(p_g - x_i))$
    else
      $x_{i+1} = \xi(x_i + step.rand.(p_i - x_i))$
    end if
    if $f(x_i) < f(p_g)$ **then**
      $p_g = x_i$
    end if
  end for
  for $i = 1$ *to* N **do**
    $x_i = x_i + rand(1, D)$
    if $f(x_i) < f(p_g)$ **then**
      $p_g = x_i$
    end if
  end for
  $k = randint([1, N])$
  $x_k = p_g$;
end for
Check termination condition

The performance of the proposed algorithm was evaluated on 9 well known high dimension benchmark problems namely Rastrigin, Levy, Sphere, Rosenbrock, Schwefel, Quadric, Griewangk, Ackley and Sum-Square. The problems are described in Appendix A The results of the simulation studies are presented in section 5.2.

3.3 Partial Differential Equation Method for Migration

A partial differential equation (PDE) contains unknown multi-variable functions and their partial derivatives. Problems that involve functions of several variables can be formulated using PDE method. Such problems can be offered solutions by creating a relevant computer model in the form of an algorithm or mathematical equation to capture the behaviour of the system being modelled. Different physical phenomena can be expressed using PDE method.

A PDE for $u(x_1,\cdots,x_n)$ function is expressed as

$$L\left(x_1,\ldots,x_n,u,\tfrac{\partial u}{\partial x_1},\ldots,\tfrac{\partial u}{\partial x_n},\tfrac{\partial^2 u}{\partial x_1\partial x_1},\ldots,\tfrac{\partial^2 u}{\partial x_1\partial x_n},\ldots\right)=0.$$

When L is a linear function of u and its derivatives, then the PDE is said to be linear. Examples are Laplace, Poisson, Helmholtz, heat, wave, and Klein-Gordon equations.

In this thesis, PDE methods were used to describe behaviours (migration and dispersion). PDE method for migration is discussed in this section while the dispersion method is described in section 3.5. The PDE method for population migration is designed to transport agents in a population from one place to another [113]. The idea comes from the known PDE fundamental solution for migration presented by Kerckhove [113].
PDE migration equation is given as: [114]

$$\frac{\partial u}{\partial t}=-c\frac{\partial u}{\partial x} \tag{3.2}$$

with $u(0,x)=u_0(x)$, parameter c is the controlling speed of the migration. u is the population size, t is time and x is location or position. $u(t,x)$ is the population size at time t in location x while $u(0,x)$=$u_0(x)$ is the initial population distribution

$$\frac{\partial u}{\partial t}=-c\frac{\partial u}{\partial x}$$
$$\frac{\partial u}{\partial t}+c\frac{\partial u}{\partial x}=0$$

The characteristic equations are:

$$\frac{dt}{1}=\frac{dx}{c}=\frac{du}{0}$$
$$dx-cdt=0$$

By integration, we have

$$x-ct=\alpha$$
$$u=u(\alpha)$$
$$u=u(x-ct)\ u[t,x]=u_0[-ct+x]$$

Displacement = speed x time.
In $u_0(x-ct)$; $u_0(x)$ displaces ct.
$u_0(x-ct)$ satisfies migration equation at any initial population distribution $u_0(x)$ [113, 114].

3.3.1 An Improved Cockroach Swarm Optimization Model

CSO was originally designed with three components (Chase-swarming, dispersion and ruthless), it was extended with hunger component and an Improved Cockroach Swarm Optimization (ICSO) is proposed. This work introduces an improvement into the existing CSO algorithm [94]. This is done by introducing a hunger component, which is described by the PDE migration method, as shown above. Hunger behaviour is similar to find food behaviour described in [34]. After an interval of time, when a cockroach is hungry, it migrates from its comfortable shelter and friends to look for food [72, 34]. A cockroach migrates from its shelter to any available food source x_{food} within the search space. In terms of search space, this prevents local optimum and enhances diversity of population. The PDE migration method is appropriate for describing the hunger behaviour of cockroaches where it is transported to food sources when hungry.

In our ICSO, find food behaviour described in [34] is remodelled to use the PDE migration method as derived in equation (3.3). This behaviour is only activated for each cockroach agent whenever the hunger threshold is reached. The hunger parameter is randomly generated in the uniform interval $[0,1]$ and compares with the hunger threshold. If this threshold is reached (i.e.$hunger == t_{hunger}$), then the cockroach agent is made to migrate to the other part of the solution space (in search of food) based on the PDE migration equation. This is illustrated as follows:

$$if(hunger == t_{hunger})$$

$$x_i = x_i + (x_i - ct) + x_{food}; \tag{3.3}$$

where x_i denotes cockroach position, $(x_i - ct)$ denotes cockroach migration from its present position, c is a constant which controls migration speed, t denotes time, x_{food} denotes food locations, t_{hunger} denotes hunger threshold and $hunger$ is a random number [0,1].

The computational steps for ICSO is shown in Algorithm 4.

Algorithm 4 An Improved Cockroach Swarm Optimization Algorithm

INPUT: Fitness function: $f(x), x \in R^D$
set parameters and generate an initial population of cockroach
set $p_g = x_1$
for $i = 2$ *to* N **do**
  if $f(x_i) < f(p_g)$ **then**
    $p_g = x_i$
  end if
end for
for $t = 1$ *to* T_{max} **do**
  for $i = 1$ *to* N **do**
    for $j = 1$ *to* N **do**
      if $abs(x_i - x_j) < visual; f(x_j) < f(x_i)$ **then**
        $p_i = x_i$
      end if
    end for
    if $p_i == x_i$ **then**
      $x_i = w.x_i + step * rand * (p_g - x_i)$
    else
      $x_i = w.x_i + step * rand * (p_i - x_i)$
    end if
    if $f(x_i) < f(p_g)$ **then**
      $p_g = x_i$
    end if
  end for
  if $Hunger == t_{hunger}$ **then**
    $x_i = x_i + (x_i - ct) + x_{fd})$
    $hunger_i = 0$
    Increment $hunger_i$ counters
  end if
  for $i = 1 to N$ **do**
    $x_i = x_i + rand(1, D)$
    if $f(x_i) < f(p_g)$ **then**
      $p_g = x_i$
    end if
  end for
  $k = randint([1, N])$
  $x_k = p_g$;
end for
Check termination condition

3.4 A Dynamic Step-Size Adaptation with Simple Euler Method

A dynamic step-size adaptation roach infestation optimization (DSARIO) is proposed to improve the earlier work on the RIO algorithm [34]. This is achieved by introducing a simple Euler method of equation (3.4) into the original RIO model. The Euler method had been used earlier by Cai et al., to improve PSO algorithm performance [102] with promising results which motivated our usage. The introduction of the dynamic step-size adaptation of the Euler method is to help RIO to avoid local optimum, and maintain a balance between exploration and exploitation. A cockroach agent adjusts its velocity according to its performance. When it explores and locates a better region, it will make a local search around the current point, or use a global search pattern.

The Euler method is the simplest of the RungeKutta and numerical integration methods of ordinary differential equations (ODE) [115]. It can be used as the basis for constructing more complex methods [115].

The Euler method can be described by recalling function derivative definition

$$y'(t) = \frac{y(t+h) - y(t)}{h} \tag{3.4}$$

where y is derivative of a function, t is time and h is the step-size.

According to Haven et al. [34], RIO find dark and friend behaviours, which is presented in section 2.5.1 and given as: If a cockroach comes within a detection radius of another cockroach agent, it will stop, socialize and share information of the darkest known location by setting local location l.

$$l_i = l_j = arg_k min\left\{F(p_k)\right\}, k = \{i, j\} \tag{3.5}$$

where i, j are the indices of the two socializing cockroaches and p_k is the personal best known darkest location. Find darkest model of Equation is extended to include find friend behaviour [34]. Find dark and friend behaviour is:

$$v_i(t+1) = v_i(t) + c_0.R_1(p_i(t) - x_i(t)) + C_{max}.R_2(l_i(t) - x_i(t)) \tag{3.6}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{3.7}$$

where $v_i(t)$ is the velocity of ith cockroach agent and $x_i(t)$ is the current location found by the ith agent at time t, $p_i(t)$ is the best location found by the ith agent and $l_i(t)$ is the best location of the entire swarm., c_0, c_{max} are cockroach parameter, R_1 and R_2 are vector of uniform random numbers, $(p_i(t) - x_i(t))$ is a velocity change in the direction of the darkest known location of a cockroach in a local neighbourhood and $(l_i(t) - x_i(t))$ is a velocity change in the direction of the darkest location

in the global search space.

Introducing Euler equation (3.4) on cockroach velocity $v_i(t)$ and position $x_i(t)$, and setting $h = 1$, we have

$$\frac{dv_i}{dt} = v_i(t+1) - v_i(t) \tag{3.8}$$

$$\frac{dx_i}{dt} = x_i(t+1) - x_i(t) \tag{3.9}$$

$\psi_1 = c_0.R_1$, $\psi_2 = c_{max}.R_2$, and $\psi = \psi_1 + \psi_2$ where ψ_1, ψ_2 and ψ are constants

Substituting equations (3.6) and (3.7) in equations (3.8) and (3.9) we have

$$\frac{dv_i}{dt} = c_0.R_1[p_i(t) - x_i(t)] + c_{max}.R_2[l_i(t) - x_i(t)] - v_i(t) \tag{3.10}$$

$$\frac{dv_i}{dt} = \psi_1[p_i(t) - x_i(t)] + \psi_2[l_i(t) - x_i(t)] - v_i(t) \tag{3.11}$$

By factorization,

$$\frac{dv_i}{dt} = -v_i(t) - x_i(t)[\psi_1 + \psi_2] + (\psi_1 p_i(t) + \psi_2 l_i) \tag{3.12}$$

$$\frac{dv_i}{dt} = -v_i - \psi x_i + (\psi_1 p_i + \psi_2 l_i) \tag{3.13}$$

Likewise for

$$\frac{dx_i}{dt} = x_i(t+1) - x_i(t) \tag{3.14}$$

$$\frac{dx_i}{dt} = x_i(t) + v_i(t+1) - x_i(t) \tag{3.15}$$

$$\frac{dx_i}{dt} = x_i(t) + c_0.R_1[p_i(t) - x_i(t)] + c_{max}.R_2[l_i(t) - x_i(t)] - x_i(t) \tag{3.16}$$

$$\frac{dx_i}{dt} = x_i(t) + \psi_1[p_i(t) - x_i(t)] + \psi_2[l_i(t) - x_i(t)] - x_i(t) \tag{3.17}$$

$$\frac{dx_i}{dt} = -x_i[\psi_1 + \psi_2] + (\psi_1 p_i + \psi_2 l_i) \tag{3.18}$$

$$\frac{dx_i}{dt} = -\psi x_i + (\psi_1 p_i + \psi_2 l_i) \tag{3.19}$$

With Euler step size h,

$$\frac{dv_i}{dt} = \frac{v_i(t+h) - v_i(t)}{h} \tag{3.20}$$

$$\frac{dv_i}{dt} = -v_i - \psi x_i + (\psi_1 p_i + \psi_2 l_i) \tag{3.21}$$

$$\frac{v_i(t+h) - v_i(t)}{h} = -v_i(t) - \psi x_i(t) + (\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.22}$$

Multiplying both sides by h, we have,

$$v_i(t+1) - v_i(t) = -hv_i(t) - h\psi x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.23}$$

$$v_i(t+1) = v_i(t) - hv_i(t) - h\psi x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.24}$$

$$v_i(t+1) = [1-h]v_i(t) - h\psi x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.25}$$

Also, with Euler step size h of equation 3.4,

$$\frac{dx_i}{dt} = \frac{x_i(t+h) - x_i(t)}{h} \tag{3.26}$$

Since,

$$\frac{dx_i}{dt} = -\psi x_i(t) + (\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.27}$$

$$\frac{x_i(t+h) - x_i(t)}{h} = -\psi x_i(t) + (\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.28}$$

Multiplying both sides by h, we have,

$$x_i(t+1) - x_i(t) = -h\psi x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.29}$$

i.e.,

$$x_i(t+1) = x_i(t) - h\psi x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.30}$$

i.e.,

$$x_i(t+1) = [1-h]x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.31}$$

Therefore, find dark and friend components is now:

$$v_i(t+1) = [1-h]v_i(t) - h\psi x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.32}$$

$$x_i(t+1) = [1-h]x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.33}$$

where v_i denotes cockroach velocity, x_i denotes cockroach position at time t, p_i denotes local best position, l_i denotes global best position, $\psi_1 = c_0.R_1$, $\psi_2 = c_{max}.R_2$, $\psi = \psi_1 + \psi_2$ and h is step–size. This new equation is introduced into RIO model as find dark and find friend behaviours, which lead

to DSARIO presented in the next subsection.

3.4.1 Dynamic Step-Size Adaptation Roach Infestation Optimization Model

The Euler method was applied to the original RIO model as shown in section 3.4 and we now have the proposed model DSARIO as:

1. Find Dark and find friend:

$$v_i(t+1) = [1-h]v_i(t) - h\psi x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.34}$$

$$x_i(t+1) = [1-h]x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \tag{3.35}$$

   where v_i denotes cockroach velocity, x_i denotes cockroach position at time t, p_i denotes local best position, l_i denotes global best position, $\psi_1 = c_0.R_1$, $\psi_2 = c_{max}.R_2$, $\psi = \psi_1 + \psi_2$ and h is step-size.

2. Find food: Cockroaches look out for food source when hungry and being transported to a random food location b.

$$x_i = b \tag{3.36}$$

For stability of Euler, Ascher and Petzold [116] recommended small step-size of $(h < 0.2)$ to obtain accuracy per step, hence this is used in our implementation. Computational steps for DSARIO is illustrated in Algorithm 5.

Algorithm 5 Dynamic Step–Size Adaptation Roach Infestation Optimization Algorithm

INPUT: Fitness function: $f(x), x \in R^D$
Initialize cockroach population, food location and set parameters
for $t = 1$ *to* $MaxIter$ **do**
"$M = [M_{ij}] = [||\vec{x}_j - \vec{x}_k||_2]$ $d_g = median\{M_{jk} \in M : 1 \leq j < k \leq N\}$ **for** $i = 1$ *to* N **do**
if $f(\vec{x}_i) < f(\vec{p}_i)$ **then**
$\vec{p}_i = \vec{x}_i$
end if
Compute roach neighbours i as
$\{j\} = \{k : 1 \leq k \leq N, k \neq i, M_{ik} < d_g\}$
N_i = number of neighbour of $|\{j\}|$
for $q = 1$ *to* N_i **do**
if $rand[0,1] < A_{min}\{N,3\}$ **then**
$\vec{l}_i = \vec{l}_j = argmin_k\{f(\vec{p}_k)\}, k = \{i, j_q\}$
end if
end for"
if $hunger_i < t_{hunger}$ **then**
$\vec{v}_i = [1-h]\vec{v}_i - h\psi\vec{x}_i + h(\psi_1\vec{p}_i + \psi_2\vec{l}_i)$
$\vec{x}_i = (1-h\psi)\vec{x}_i + h(\psi_1\vec{p}_i + \psi_2\vec{l}_i)$
else
$\vec{x}_i = \vec{b}$
Randomly positioned food
set hunger counter
end if
if Cockroach hungry **then**
Increment hunger counters
end if
Check termination condition
end for
end for
OUTPUT: Corresponding function fitness value $f(x^*)$ and global solution x^*.

3.5 Partial Differential Equation Method for Population Dispersion

At intervals of time a cockroach exhibits vigilance behaviour whenever light is shown in the dark shelter or the presence of a predator is noticed. The adult cockroaches, at the edge of the group, signal the young ones with body movements. Alarm pheromone will be emitted which results in the rapid dispersion of group members.

As described in section 3.3, PDE are used for formulation of problems, creation of models and expression of a wide variety of physical phenomena. In this section a PDE method for population dispersion which is designed to make overcrowding populations in a location, disperse over a wide area [113] is considered for describing the vigilance behaviour of cockroaches. The idea comes from the known PDE fundamental solution for dispersion presented by Kerckhove [113]. Population size u is a dynamic quantity, its evolution is modelled as a function of time t and location x.

The dispersion equation is given as

$$u_t = g\, u_{xx} \tag{3.37}$$

where u is the population size at time t and location x i.e., $u(t,x)$ and $u(0,x)$=$u_0(x)$ is the initial population distribution. g denotes dispersion coefficient that controls dispersion speed.

The PDE for the population dispersion equation is expressed in [113] as:

$$\frac{\partial u}{\partial t} = g\frac{\partial^2 u}{\partial x^2}; g > 0 \tag{3.38}$$

$u(0,x) = f(x)$ is the Cauchy problem for this equation, where $f(x)$ is an arbitrary function. The general solution for dispersion equation by using the Fourier transform of the form

$$u(t,x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\zeta) e^{-g\zeta^2 t} e^{i\xi x} d\zeta, \tag{3.39}$$

F satisfy initial condition, it is an arbitrary function and given by transform of f, i.e

$$F(\zeta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i\zeta x} dx. \tag{3.40}$$

f denotes intense source of dispersion, delta distribution is used to approximate the proceeding integral times source strength. A source strength normalized to 1, gives:

$$F(\zeta) = \frac{1}{\sqrt{2\pi}}, \tag{3.41}$$

give the resulting dispersion solution as

$$u(t,x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-g\zeta^2 t} e^{i\zeta x} d\zeta. \tag{3.42}$$

This Gaussian integral is solved and we have

$$u(t,x) = \frac{1}{2\sqrt{\pi g t}} \exp\left(-\frac{x^2}{4gt}\right). \tag{3.43}$$

i.e.,

$$u(t,x) = \frac{1}{\sqrt{(4\pi g t)}} e^{\frac{-x^2}{4gt}} \tag{3.44}$$

$u(t,x)$ is the dispersion function that spread out in time t.

From equations (3.37) and (3.38)

$$g u_{xx} = g\frac{1}{\sqrt{(4\pi g t)}} exp[\frac{-x^2}{4gt}] \tag{3.45}$$

We modelled vigilance behaviour as:

$$x_i^n = x_i + (\frac{g}{\sqrt{(4\pi g t)}} exp[\frac{-x_i^2}{4gt}]) \tag{3.46}$$

where x_i^n denotes a cockroach's new position, x_i denotes a cockroach's current position, g denotes dispersion coefficient which is a constant that controls dispersion speed at time t. The larger the dispersion coefficient the faster the dispersion rate [113].

PDE for population dispersion, can make overcrowding populations in a location disperse over a wide area. It is appropriate for modelling the vigilance behaviour of cockroaches where cockroach populations disperse rapidly over a wider area in the search space. The vigilance behaviour is one of the components introduced into the existing RIO algorithm in section 3.6.1 that leads to the construction of modified RIO. It is similar to the dispersion component of the original CSO algorithm.

3.6 Predator-prey with Crossover and Mutation Methods

A crossover method allows for the recombination of two solutions to get a better solution. A different solution is created in successive generations by combining materials from two individual solutions of the previous generation. The two solutions involved in the crossover operation are known as parent solutions and the resulting solutions are known as children solution [117, 118, 119, 120]. The crossover process creates a point in the neighbourhood of the current point and achieves a local search around the current solution, it executes exploitation and leaves exploration for the mutation operator [120]. The mutation operator enhances population diversity by modifying chromosomes randomly. It is an exploration operator that recovers any diversity lost during the selection process and also explores solutions and prevents local convergence [120].

Predator-prey evolution method (PPEM) has been significantly used in optimization problems in several studies including [103, 104, 105, 106, 107, 108, 109, 110, 111]. The idea of crossover and mutation for predator prey habit used in this thesis comes from Chowdhury et al. [104]. The basic concept of PPEM was suggested by Laumanns et al., [103], where an algorithm was described, that mimics the natural phenomena that a predator kills the weakest prey in its neighbourhood and the relatively strong prey that evolves from the next generation, then replaces the weak prey. New operators were introduced by Deb et al., [105, 110]. A dynamic spatial structure version of PPEM was proposed by Li [106]. Other versions were proposed [107, 108]. Chowdury et al., [104, 109] presented a modified predator-prey (MPP) algorithm where they made significant modifications to the pre-predator algorithm regarding selection procedure, predators movements and general dynamics for good convergence and solutions diversity, also reducing function evaluations number. MPP fundamental steps are outlined [109]. One of the significant features of MPP is evolution, which is described as crossover of the two strongest prey to evolve new solutions in a local neighbourhood containing a predator.

The blend crossover which was proposed [110], improved [111] and applied [104]. The blend crossover is described as follow:

$$x_i^{(i,i+1)} = (1-\gamma_i)x_i^{(1,t)} + \gamma_i x_i^{(2,t)} \tag{3.47}$$

$$\gamma_i = (1+2\alpha)U_i - \alpha \tag{3.48}$$

where γ_i is a cross-over operator, $x_i^{(1,t)}$ and $x_i^{(2,t)}$ are parent solutions, $x_i^{(i,i+1)}$ is a child solution of parent $x_i^{(1,t)}$ and $x_i^{(2,t)}$, α is a constants value 0.5 as suggested by Deb [111], $U_i \in U(0,1)$ is a uniform random numbers between zero and one that allows uniformly random distribution of crossover operator values in each iteration.

Crossover offspring prey is subjected to non-uniform mutation that was introduced [121] and modified [109]. This is to allow adjustment in the extent of mutation dynamically.

$$\beta = 10^{-(t/T_{max})} \tag{3.49}$$

$$y_i^{(i,i+1)} = x_i^{(i,i+1)} + \tau\beta(x_i^{UP} - x_i^{LW})\left[1 - r_i^{(1-(t/T_{max}))^b}\right] \tag{3.50}$$

$y_i^{(i,i+1)}$ is a child produced from parent solution $x_i^{(i,i+1)}$, τ is a boolean value -1 or 1, with fifty percent probability each, r_i is randomly generated [0,1], (x_i^{UP} and x_i^{LW}) denotes variable i upper and lower limits. b is a constant of value 1.5 as suggested by Chowdhury et al. [109], b is a scaling parameter, t and T_{max} represent iterations number and maximum iterations respectively.

In this work, we postulate that the predator-prey with crossover and mutation methods approach is suitable for describing cannibalism behaviour of cockroaches. The cannibalism behaviour is described thus: Cockroaches exhibit the predator-prey habit of eating one another, when a cockroach is hungry and there is no food in the neighbourhood, the strong cockroach eats the weak ones [35, 72]. The cannibalism habit of the cockroach is modelled as a situation whereby a strong cockroach called the predator, eats the weakest cockroach called the prey, in its neighbourhood. The generation of new solutions in a local neighbourhood containing a predator, is instigated by crossover of the two local prey and mutation is performed on the crossover child prey.

The evolution method of MPP [104] is adopted for modelling cockroach cannibalism behaviour in this work. This method was investigated first in RIO algorithm to describe a cannibalism habits of cockroaches that was included as one of the new components to the original RIO in this work. We conducted experiments to evaluate the effect of PPEM on RIO and found it efficient, as revealed by the results of the experiment. Motivation from the results led to investigation of PPEM also in CSO algorithm. The ruthless component of the CSO algorithm was modified using the PPEM. Simulation studies were carried out using 70 global optimization test problems and comparisons were done experimentally and statistically with similar algorithms.

Vigilance and cannibalism behaviours of cockroaches were modelled in this work, using the methods described above as shown in section 3.6.1.

3.6.1 Modified Roach Infestation Optimization Model

Havens et al., presented RIO as a PSO-adapted algorithm with a slack cockroach behaviour model [34]. The RIO model as presented in section 2.5.1 has three essential components namely find dark, find friend and find food. In this work, we extended the original RIO model by reconstructing find dark and friend components and also added vigilance and cannibalism components,

to give the modified roach infestation optimization (MRIO). An MRIO algorithm that is tied to cockroach social behaviours is proposed in this work as an improvement over the original RIO algorithm. MRIO is constructed with the following components:

1. Find dark and find friends: Cockroaches always seek dark shelters and also prefer to travel in the dark to locate and socialize with their friends in dark places. Cockroaches usually locate their friends in dark shelters, the darker a location the more cockroaches are found there; darkness corresponds to the fitness value of the fitness function $f(x) \in \mathbb{R}^D$. The level of darkness at a location $p_k \in \mathbb{R}^D$ is directly proportional to the value of the fitness function at that location $f(p_k)$.

   We redesign RIO find dark and friend of section 2.5.1 and present the new model as:

$$x_i^n = \begin{cases} \lambda(x_i + \psi_1(p_i - x_i)) \\ \lambda(x_i + \psi_2(l_i - x_i)) \end{cases} \tag{3.51}$$

   where λ denotes the constrained factor that prevents swarm explosion, maintains swarm stability and controls cockroach movement from its current position towards its new position; x_i denotes the cockroach's current position; x_i^n denotes the cockroach's new position; $\psi_1 = c_0 R_1$ is cockroach's cognitive factor for locating a personal dark position; $\psi_2 = c_{max} R_2$ is the cockroach's social factor for locating global dark position; c_0 and c_{max} are constants while $R_1 \in U(0,1)$ and $R_2 \in U(0,1)$ are random sequences of numbers between 0 and 1 that effect stochastic nature of the algorithm; p_i denotes the cockroach's darkest known position in a local neighbourhood; $(p_i - x_i)$ denotes cockroach movement from current position towards the darkest known position, l_i denotes the global dark position in the entire given search region where other cockroaches can be found. $(l_i - x_i)$ denotes the cockroach's movement from its current position towards the darkest position.

2. Find food: Find food behaviour as presented in section 2.5.1 [34]: At intervals of time when a cockroach agent i becomes hungry, it searches for food and is taken to a random food position b.

$$x_i = b \tag{3.52}$$

3. Vigilance behaviour: At intervals of time, in the aggregation of cockroaches, when the presence of a predator is noticed or light is shown in their shelter, the older cockroaches warn the younger cockroaches with body movement and emit dispersion alarm pheromone [72]

which make cockroaches disperse rapidly over a wider area in the search space. Vigilance behaviour as presented in section 3.5 is given as:

$$x_i^n = x_i + (\frac{g}{\sqrt{(4\pi gt)}} exp[\frac{-x_i^2}{4gt}]) \tag{3.53}$$

where x_i^n denotes the cockroach's new position, x_i denotes a cockroach's current position, g denotes dispersion coefficient which is a constant that controls dispersion speed at time t.

4. Cannibalism behaviour: Very rarely, when hungry and there is no food in the neighbourhood, a cockroach will exhibit cannibalism by eating another [72]. A strong cockroach will eat up the weak ones in its neighbourhood. Chowdhury et al. [104], present a predator–prey method. We adopted the method for cannibalism behaviour. The cannibalism habit represents a situation whereby strong cockroaches (predator), eat the weakest cockroaches (prey), in its neighbourhood and emerge as good solutions or the next generation of prey. The generation of new solutions in a local neighbourhood containing a predator- Strong cockroach, is instigated by crossover of the strongest two local cockroaches-preys. Mutation is performed on the cross-over child prey and the resulting solution is taken as the best solution.
Cannibalism behaviour as presented in section 3.6 is given as:

$$x_i^{(i,i+1)} = (1-\gamma_i)x_i^{(1,t)} + \gamma_i x_i^{(2,t)} \tag{3.54}$$

$$\gamma_i = (1+2\alpha)U_i - \alpha \tag{3.55}$$

$$\beta = 10^{-(t/T_{max})} \tag{3.56}$$

$$y_i^{(i,i+1)} = x_i^{(i,i+1)} + \tau\beta(x_i^{UP} - x_i^{LW}) \left[1 - r_i^{(1-(t/T_{max}))^b}\right] \tag{3.57}$$

where $x_i^{(1,t)}$ and $x_i^{(2,t)}$ are parent solutions from cockroach population, $x_i^{(i,i+1)}$ is a child solution of parent $x_i^{(1,t)}$ and $x_i^{(2,t)}$, α is a constants of value 0.5, $U_i \in U(0,1)$. $y_i^{(i,i+1)}$ is a child produced from parent solution $x_i^{(i,i+1)}$, τ is a boolean of -1 or 1 value with 50% probability, r_i is rand[0,1], (x_i^{UP} and x_i^{LW}) denotes variable i upper and lower limits. b is a scaling parameter, t and T_{max} represent iterations number and maximum iterations respectively. i denotes variable (cockroach) in cockroach population x_i. The child $y_i^{(i,i+1)}$ is considered as

the optimum solution in each iteration.

The MRIO computational steps are given as:

1. Initialise cockroach swarm with uniformly distributed random numbers and set all parameters with values.
2. Locate best position using equations (2.8).
3. Perform find dark and friends behaviour using equation (3.51).
4. Perform find food behaviour using equation (3.52).
5. Perform vigilance behaviour using equation (3.53).
6. Perform cannibalism behaviour using equations (3.54), (3.55) and (3.57)
7. Update global best position.
8. Repeat the loop until stopping criteria is reached.

3.7 Adaptive Cockroach Swarm Model

Establishing a balance of local and global search of a search region, by the crossover and mutation approach is very adequate for complex optimization problems to which classical methods cannot provide solutions [120]. In addition to the establishment of balance of exploration and exploitation, it is good to ensure that such balance is self-adaptive. That is, the distribution of the offspring depends on that of parents [120]. Through self-adaptation, close parents will generate close offspring and vice versa [120]. Recent studies on the blend crossover $(BLX-\alpha)$ [104, 109, 110, 111, 120] have shown good performance of the crossover operator that generates individuals in the exploration zone. The blend crossover operator maintains diversity, prevents the premature convergence to inner points of the search space and generates offspring in the exploration and exploitation zones in the correct proportion. In blend crossover $(BLX-\alpha)$ there is the same probability of generating an offspring between the parents, and in an area close to the parents whose amplitude is modulated by the α parameter[120].

$x_i^{(1,t)}$ and $x_i^{(2,t)}$ are parent solutions [111].
Assuming $x_i^{(1,t)} < x_i^{(2,t)}$
The $BLX-\alpha$ selects a solution at random in this range:

$$[x_i^{(1,t)} - \alpha(x_i^{(2,t)} - x_i^{(1,t)},\ x_i^{(2,t)} + \alpha(x_i^{(2,t)} - x_i^{(1,t)}))]$$

$U_i \in (0,1)$ is the random number between 0 and 1
The child is given as:

$$x_i^{(i,i+1)} = (1-\gamma_i)x_i^{(1,t)} + \gamma_i x_i^{(2,t)}$$

where $\gamma_i = (1+2\alpha)U_i - \alpha$.

Blend crossover $BLX-\alpha$ has an interesting feature for search algorithms to exhibit self-adaptation which causes populations to either diverge or converge for better regions in the search region [111, 120].

In this work, blend crossover $(BLX-\alpha)$ is adopted, being inspired by the results from previous studies which include [104, 109, 110, 111, 120]. We introduce blend crossover PPEM into our CSO algorithm and proposed an adaptive CSO (ACSO). Blend crossover is introduced into CSO to control balance between exploration and exploitation; and improve convergence speed by self-adaptive component. These features are appreciated in any search algorithm for preventing premature convergence and improving local fine-tuning [120]. This technique has been shown effective in many studies as already indicated in this section and has been equally shown in our study on MRIO presented in section 3.6.1 where we constructed one of the components with PPEM.

In the study of ACSO, the ruthless behaviour component of the CSO algorithm which was designed in the original CSO algorithm is modified using PPEM. The ruthless component of the original CSO [35] is described as the current best cockroach replacing the one that is picked randomly at intervals of time. The strong one eats the weak one.

$$x_k = p_g \tag{3.58}$$

where k is a random integer within [1,N], p_g is the global best cockroach.

It is obvious from equation (3.58) that there is the possibility of population collapse as a result of continuous ruthlessness when strong cockroaches keep eating the weak ones. Also if the ruthlessness is high, strong cockroaches may possibly eat another cockroach, which might have been a good solution. To avoid the likelihood of population collapse, we engage the PPEM presented in section 3.6 to modified the ruthless component. When a weak cockroach (prey) is eaten by a strong cockroach (predator), another relatively strong cockroach is created through evolution of the PPEM approach to replace the cockroach prey. By this approach, the population is preserved. The ruthless component is now modelled using PPEM similar to the cannibalism component of MRIO algorithm. ACSO extended our proposed ICSO presented in section 3.3.1 with the PPEM process; the computational steps for ACSO algorithm is illustrated in Algorithm 6.

Algorithm 6 Adaptive Cockroach Swarm Optimization Algorithm

INPUT: Fitness function: $f(x), x \in R^D$
set parameters and generate an initial population of cockroach
set $p_g = x_1$
for $i = 2$ *to* N **do**
  if $f(x_i) < f(p_g)$ **then**
    $p_g = x_i$
  end if
end for
while $Iter < Max_Iter$ **do**
  for $i = 1$ *to* N **do**
    for $j = 1$ *to* N **do**
      if $abs(x_i - x_j) < visual; f(x_j) < f(x_i)$ **then**
        $p_i = x_i$
      end if
    end for
    Chase-swarming behaviour
    if $p_i == x_i$ **then**
      $x_i = w.x_i + step * rand * (p_g - x_i)$
    else
      $x_i = w.x_i + step * rand * (p_i - x_i)$
    end if
    Update global optimum
    if $f(x_i) < f(p_g)$ **then**
      $p_g = x_i$
    end if
  end for
  Dispersion behaviour
  for $i = 1$ *to* N **do**
    $x_i = x_i + rand(1, D)$
    Update global optimum
    if $f(x_i) < f(p_g)$ **then**
      $p_g = x_i$
    end if
  end for
  Hunger behaviour
  if $Hunger == t_{hunger}$ **then**
    $x_i = x_i + (x_i - ct) + x_{fd}$
    $hunger_i = 0$
    Increment $hunger_i$ counters
  end if
  Update global optimum
  if $f(x_i) < f(p_g)$ **then**
    $p_g = x_i$
  end if
  Ruthless behavior
  for $i = 1$ *to* N **do**
    $x_i^{(i,i+1)} = (1 - \gamma_i) x_i^{(1,t)} + \gamma_i x_i^{(2,t)}$
    $y_i^{(i,i+1)} = x_i^{(i,i+1)} + \tau\beta(x_i^{UP} - x_i^{LW}) \left[1 - r_i^{(1-(t/T_{max}))^b}\right]$
  end for
  Update global optimum
  if $f(x_i) < f(p_g)$ **then**
    $p_g = x_i$
  end if
end while
Check termination condition

3.8 Discretization Using Sigmoid Function

Most swarm intelligence algorithms were traditionally designed to provide solutions to continuous optimization problems. Nevertheless they can be adapted to discrete optimization such as binary and combinatorial by discretization of the continuous space [122]. This can be accomplished by transforming continuous values into a limited number of possible states.

The sigmoid function can be used to transform a continuous space into a binary one [54]. The combination of sigmoid function with an additional operator can be used for the transformation of continuous space into multi valued discrete space as shown in these studies [55, 56, 57]. The transformation is applied to each dimension of the solution vector forcing each element to be a binary digit or other digits greater than 2 [122]. The resulting change in position is defined in equations (2.1) and (2.3) of section 2.3.5 for binary and multi-valued discrete spaces respectively.

The idea of using sigmoid function comes from studies on binary and discrete PSO which include [54, 55, 56, 57]. See section 2.3.5 for further details on sigmoid function and discretization.

3.8.1 Discrete Space Cockroach Swarm Models

Our study on a multi-valued discrete space cockroach swarm optimization (DCSO) algorithm was inspired by the previous studies on binary and discrete PSO which include [54, 55, 56, 57], where PSO algorithms that were traditionally designed for continuous optimization problems were translated into binary discrete valued space. DCSO was specifically inspired by the study of Osadciw and Veeramachaneni who presented a multi-valued discrete PSO [55] where they use sigmoid with additional operators for logistic transformation of particle states (See section 2.3.5). We adopted similar techniques for DCSO by extending a CSO algorithm that was originally designed for continuous valued space to construct a multi-valued discrete space algorithm. DCSO can search in discrete space $[0, B-1]$; where B denotes any number system such as binary, ternary and quaternary. DCSO searches from one state to another, for instance in binary space, it searches between states (0 and 1); ternary (0, 1, and 2); quaternary (0, 1, 2 and 3). Cockroach position is translated into number between $[0, B-1]$. An operator μ is introduced in this study for the enhancement of convergence of the DSCO algorithm, optimal results were obtained in each experiment when μ is set to a random number between 0.1 and 0.2.

A sigmoid $S_i = \frac{B}{1+e^{-x_i}}$ with $(B-1)\mu$ is used to generate solution that is rounded to the nearest discrete variable.

$$x_i = round((B-1)\mu + S_i)$$

Cockroach position is updated as:

$$x_i = \begin{cases} B-1, & if \;\; x_i > B-1 \\ 0, & if \;\; x_i < 0 \end{cases} \tag{3.59}$$

where x_i is current position, B is base number considered and μ is an operator which enhances the performance of the algorithm. Cockroach positions are now discrete values between 0 and $B-1$. There is a possibility of selecting a number between $[0, B-1]$.
DCSO algorithm is shown in Algorithm 7 and its computational steps are given as:

1. Initialize cockroach swarm with uniform distributed random numbers in the range $[0, B-1]$ and set all parameters with values.

2. Find best position using equations (2.13) and (2.14).

3. Exhibit chase-swarming behaviour using equation (3.1).

4. Exhibit dispersion behavior using equation (2.15).

4. Exhibit hunger behavior using equation (3.3).

5. Exhibit ruthless behavior

6. Limit cockroach agent i magnitude with constant x_{max} if $x_i > x_{max}$ then
   $x_i = x_{max}$
   elseif $x_i < -(x_{max})$ then
   $x_i = -(x_{max})$

7. Update cockroach position using sigmoid function to transform position probabilistically.
   $S_i = \frac{B}{1+e^{-x_i}}$
   $x_i = round((B-1)\mu + S_i)$
   if $x_i > B-1$ then $x_i = B-1$
   and if $x_i < 0$ then $x_i = 0$.

8. Repeat the loop until stopping criteria is reached.

DCSO was evaluated using benchmark test functions and also on 53 TSP instances. The results of the test are shown in sections 5.7 and 5.8 respectively.

Algorithm 7 Discrete Space Cockroach Swarm Optimization

INPUT: Fitness function:$x_i \in \{0,(B-1)\}^D$, $f : \{0,(B-1)\}^D \to \mathbb{R}^D$
set parameters and generate an initial population of cockroach
set$p_g = x_1$
for $i = 2$ *to* N **do**
  if $f(x_i) < f(p_g)$ **then**
    $p_g = x_i$
  end if
end for
for $t = 1$ *to* T_{max} **do**
  for $i = 1$ *to* N **do**
    for $j = 1$ *to* N **do**
      if $abs(x_i - x_j) < visual; f(x_j) < f(x_i)$ **then**
        $p_i = x_i$
      end if
    end for
    if $p_i == x_i$ **then**
      $x_i = w.x_i + step * rand * (p_g - x_i)$
    else
      $x_i = w.x_i + step * rand * (p_i - x_i)$
    end if
    if $f(x_i) < f(p_g)$ **then**
      $p_g = x_i$
    end if
  end for
  for $i = 1$ *to* N **do**
    $x_i = x_i + rand(1,D)$
    if $f(x_i) < f(p_g)$ **then**
      $p_g = x_i$
    end if
  end for
  if $Hunger == t_{hunger}$ **then**
    $x_i = x_i + (x_i - ct) + x_{fd}$
    $hunger_i = 0$
    Increment $hunger_i$ counters
  end if
  for $i = 1$ *to* N **do**
    $x_i^{(i,i+1)} = (1-\gamma_i)x_i^{(1,t)} + \gamma_i x_i^{(2,t)}$
    $y_i^{(i,i+1)} = x_i^{(i,i+1)} + \tau\beta(x_i^{UP} - x_i^{LW})\left[1 - r_i^{(1-(t/T_{max}))^b}\right]$
  end for
  if $f(x_i) < f(p_g)$ **then**
    $p_g = x_i$
  end if
  Limit cockroach agent i magnitude with constant x_{max}
  Update position;$S_i = \frac{B}{1+e^{-x(i)}}$
  $x_i = round((B-1)\mu + S_i)$
  if $x_i > B-1$ **then**
    $x_i = B-1$
  end if
  if $x_i < 0$ **then**
    $x_i = 0$
  end if
end for

3.8.2 DCSO for Travelling Salesman Problem

DCSO was applied to TSP instances in this work. DCSO is improved for solving TSP by the incorporation of the nearest neighbour (NN) algorithm and iterated 2-opt search method. Improvement is done upon each tour by the employment of 2-opt heuristics and the best result is chosen. The initial tour is constructed with the local search method and 2-opt technique was used for interchanging edges. Nearest neighbour algorithm and iterated 2-opt search method are described in sections 3.8.3 and 3.8.4 respectively.

3.8.3 Tour Construction Method

The nearest neighbour (NN) method is used in this study to construct the initial tour. NN is a simple and well known method for TSP; the method begins with a tour that is made up of randomly chosen city and subsequently added unvisited city to the last city [71]. The algorithm continues until all the cities are on the tour. Computational steps of NN on TSP is given as:

1. Begin with current city (randomly choose a city and set it as the current city)
2. Detect the shortest edge connecting current city and an unvisited city V.
3. Set current city to V and mark it as visited.
4. If all the cities in domain are visited, then terminate, otherwise return to step 2.

3.8.4 Tour Improvement Method

Tour improvement methods are simple local search heuristics for improving initial tour [71]. We use 2-opt method in this paper to improve the initial tour; the tour improved iteratively by moving from one tour to its best neighbour until no further possible improvement is reached. 2-opt searching method is a well-known method for solving TSP; it is mainly used for improving solution. The 2-opt technique check if the exchange of an edge with another edge gives a minimum tour and the process continues until there is no more possible improvements to be done [66, 71]. The 2-opt method consists of three steps:

1 Take two pairs of nodes (P,Q) and (R,S) respectively from a tour and check if the distance PQRS is longer than PSQR.

2 If true, swap P and R i.e reversing the tour between the two nodes.

3 Continue with the improvement process by retuning to step 1, otherwise stop if no improvement is possible.

3.8.5 DCSO-TSP Computational Steps

In a TSP of m-ordered, cockroach's position in cockroach population is denoted as $X_i = (x_{i1}, x_{i1}....x_{im})$ which represents the travelling circle of $x_{i1} \rightarrow x_{i2} \rightarrow ...x_{im} \rightarrow x_{i1}$

The computational steps of the proposed DCSO algorithm for TSP is described as follows:

1 Initialize a cockroach swarm with uniform distributed random numbers in the range $[0, B-1]$, make a tour construction using the nearest neighbour method, evaluate each population member (calculate total distance), initialize each cockroach position: Set p_i as the initial position, search for the best route p_g in the population.

2 Apply the DCSO process to each cockroach, compute new position and update new position.

3 Improve the position of cockroaches using 2-opt method.

4 Check termination condition. If satisfied, stop the algorithm, otherwise return to step 2.

Chapter 4

Experimental Design

4.1 Implementation Platform

All algorithms proposed as presented in Chapter 3 were implemented in MATLAB 7.14 (R2012a) environment on HP ProBook 4530 with a Windows 7 operating system; containing 2.30 GHz processor and 4.00 GB of RAM. The computer system configuration is shown in Table 4.1.

Table 4.1: Computer System Configuration

System	HP ProBook 4530
CPU	2.30 GHz
RAM	4.00 GB
Operating System	Windows 7
Implementation Platform	MATLAB 7.14 (R2012a)

4.2 Parameter Selection

The parameters of RBA include: cockroach population size N, the maximum iteration T_{max}, t is the number of iteration, hunger interval t_{hunger}, Cockroach largest step $Step = 2$. Table 4.2 depicts standard parameter settings that have been used successfully for RIO and CSO in the literature [34, 35, 94] and presented in Chapter 2. Additional parameters used by improved RBAs as presented in Chapter 3 for SCCSO, ICSO, DSARIO, MRIO, ACSO and DCSO are: $\xi = rand[0,1]$ constrained factor, $c = 5$ constant that controls migration speed,$t = 5$ migration time, hunger threshold 0.5, Euler step-size $h = 0.1$, $g = 20$ dispersion coefficient, $\alpha = 0.5$ a constant which is the adaptive coefficient, τ is boolean with value $-1,1$ with the probability of 0.5, $b = 1.5$ is a constant which is a scaling parameter, $U_i = U(0,1)$, B number base (binary, ternary, etc) and $\mu = rand[0.1,0.2]$.

Table 4.2: Experiment Parameters

	Description
N	Swarm size
T_{max}	Maximum iterations
$C_0 = 0.7, C_{max} = 1.43$	Cockroach parameters
$A_1 = 0.49, A_2 = 0.63, A_3 = 0.65$	Probabilities of cockroaches socializing with one another
$t_{hunger} = 100$	hunger interval
$Step = 2$	Cockroach largest step
$w = 0.618$	Inertial weight
$visual = 1$	Cockroach visual scope

4.3 Statistical Analysis Tools

The statistical analysis tools used for analysis of data in this work are Kruskal-Wallis test, a Analysis of variance (ANOVA) test, Jonckheere-Terpstra test and Mann-Whitney U-test as reported in Chapter 5. Tools such as the Kruskal-Wallis test, a Analysis of variance (ANOVA) test and Jonckheere-Terpstra test statistic are used for data that has two or more independent groups. The Mann-Whitney U-test statistic tool is used for data that have only two independent groups.

The statistical evidence derived from any of the statistical tools determines the acceptance or rejection of the Null Hypothesis that proposes that no statistical significance exists in a set of given observations. The calculated p-value is the probability of obtaining either the observed difference or a more extreme value of the difference between the two groups, it is used as a basis for either accepting or rejecting the Null Hypothesis. If the p-value is less than a threshold value of 0.05, we reject the Null Hypothesis and the result is considered significant. On the other hand, if the p-value is greater than 0.05, we accept the Null Hypothesis.

4.4 Global Optimization Test Problems

Global optimization test functions are used for validating and comparing the performance of optimization algorithms. In this thesis, we tested the reliability, efficiency and validation of the improved RBA with test functions of diverse properties in terms of modality, separability and valley landscape.

The validation of the performance of the developed RBA variants over a set of test functions presented in Appendix A were carried out in this work. The test function in Appendix A have been used by many researchers and reported in the literature. The developed RBA variants were deployed for varying global optimization problems and the obtained results compared empirically

and statistically with the exiting algorithms.

Furthermore, discrete cockroach swarm optimization was tested on TSP using 53 instances of TSP from TSP library [123].

4.5 Performance Evaluation Criteria

The performance of evolutionary algorithms can be evaluated using some major criteria such as convergence, speed, robustness and success performance [124, 125]. Convergence measure determines how accurately an algorithm converges to the desired solution through scientific means. We use equation (4.1) to test the convergence of all the algorithms in this work; the equation checks if the difference between the numerical approximation of each algorithm and the global optimum of a problem satisfies the given condition. In all our experiments, a run is terminated when the algorithm satisfies the condition of equation (4.1).

$$|f(x^*) - f_{min}| \leq \varepsilon \tag{4.1}$$

where $f(x^*)$ is best among all the best solutions found, f_{min} is the desired optimal solution and $\varepsilon = 10^{-4}$.

The speed of an algorithm is determined by the number of function evaluations which are machine independent [126, 125, 127, 128]. The mean of function evaluation (MFE) was computed in our experiments. We also measure the speed of algorithms run on the same machine, using run-time which is computed in seconds during experiments.

The robustness of an algorithm is measured by the wide range of problems the algorithm can solve. The improved algorithms presented in this work were tested on large numbers of problems shown in Appendix A. Each experiments runs for a specified number of times.

The following statistical measures were obtained from our experiments using the above mentioned criteria.

1. Minimum Optimum (MinOpt): This is the best of all best solutions found by each algorithm on a test problem at the end of the experiment. It is defined as

$$MinOpt = min\{f_{min1}, f_{min2}, ..., f_{minn}\} \tag{4.2}$$

   where n denotes number of runs.

2. Mean optimal (MeanOpt): This is the average fitness value of all the best solutions found by

each algorithm on a test problem in an experiment. MeanOpt is defined by:

$$MeanOpt = \frac{1}{n}\sum_{i=1}^{n} f_{min\ i} \tag{4.3}$$

where n denotes number of runs.

3. MFE of all the function evaluations of the best solutions found by each algorithm on a test problem per experiment. MFE can be described as:

$$MFE = \frac{1}{n}\sum_{i=1}^{n} FE_i \tag{4.4}$$

4. Standard Deviations: The measure of dispersion of the best solutions of mean optimal was computed for each algorithm per problem.

5. Success Rate: A run is considered successful if an algorithm is able to find the optimal solution for a given problem [124, 125, 129]. The success of an algorithm is determined by the number of successful runs out of the total number of runs. Success rate (SRT) is computed using:

$$SRT = \frac{SR}{TR} \times 100 \tag{4.5}$$

where SR denotes number of successful runs and TR denotes number of total runs.

6. Success Performance: Success performance (SP) can be used to estimate the number of function evaluations needed for an algorithm to successfully solve a problem. SP is computed using equation (4.6) [125, 129, 130].

$$SP = Mean(FEs) \times \frac{TR}{SR} \tag{4.6}$$

where FEs is the function evaluations of successful runs, TR denotes number of total runs and SR denotes number of successful runs. We compute the normalised SP (nSP) by dividing an algorithm SP by the SP of the best performing algorithm (SP_{best}) on a given problem [125].

7. Test statistics of Kruskal-Wallis, ANOVA, Jonckheere-Terpstra, Mann-Whitney U-test were used to determine the significant difference of the improved algorithms over the existing ones. See section 4.3.

8. Error estimate $(Error)$: Numerical error estimate used for measuring the accuracy of the numerical approximation of solutions found, to the true value of global minimum is $Error$.

$$Error = \begin{cases} |f(x^*) - f_{min}|, & if \quad f(x^*) = 0 \\ \frac{|f(x^*) - f_{min}|}{f(x^*)} & otherwise \end{cases} \tag{4.7}$$

Chapter 5

Experimental Settings and Results

5.1 Introduction

The improved RBA variants presented in Chapter 3 were simulated using the experimental designs shown in Chapter 4 and the respective experimental settings shown in this chapter. The empirical results of each RBA variant which were statistically analysed and compared with related existing algorithms are presented.

5.2 Stochastic Constriction Cockroach Swarm Optimization for Multidimensional Space Function Problems

A SCCSO model described in section 3.2.1 was implemented and evaluated on global optimization problems. A series of experiments were conducted in two stages to prove the effect of SCF in the performance of SCCSO on a set of standard benchmark multi-dimension test function problems presented in Table 16 of Appendix A. Each experiment runs 20 times with a maximum iteration of 1000, using cockroach swarm size 50. For each benchmark, global minimum values were computed in each experiment; the best, average and standard deviation of the optimal values were recorded.

In stage I of the experiments, the performance of SCCSO was compared with that of existing CSO and Modified Cockroach Swarm Optimization (MSCO) algorithms of [35, 94] for dimensions 10, 20, 30 and 40. Success rates and computation time in seconds were recorded. Tables 5.1, 5.2, 5.3, 5.4 show the results of stage I experiment while Tables 5.5, 5.6, 5.7 depicts the comparison best, average and standard deviation results respectively. Tables 5.8 and 5.9 show the comparison success rate and execution time respectively.

Stage II experiments show the performance of SCCSO for dimensions 50, 100, 500, 1000,

2000 and 3000 in Tables 5.10, 5.11, 5.12, 5.13,5.14 and 5.15 respectively. The performance of SCCSO was compared with that of a line search restart (LSRS) technique which had been proved in literature for evaluating high dimension test function problems [131]; Table 5.17 shows the comparison average performance of LSRS and SCCSO.

The test statistic of ANOVA was conducted to determine the significant difference between the average performance of SCCSO and LSRS. The results of the test statistic are in table 5.17 and the graphical representation is depicted in Figure 5.1. No. 1 and 2 on x-axis of Figure 5.1 denote LSRS and SCCSO algorithms respectively. The result of the test statistics shows that there is no significant difference between SCCSO and LSRS algorithms.

The effects of the stochastic constriction were revealed in our experiments; SCCSO was shown to have better convergence and speed than the existing CSO and MSCO for dimensions 10,20,30 and 40. SCCSO showed similar performance to LSRS for problems of dimensions 50, 100, 500, 1000 and 2000. SCCSO was tested further for dimensions 3000, and without modifying the algorithm, it can evaluate higher number of variables above 3000 dimensions. The comparisons results of SCCSO with the existing CSO and MCSO proved its superiority, also the comparison of SCCSO with LSRS proved its ability to compete with known global optimization technique.

Table 5.1: Performance of CSO, MCSO and SCCSO for Dimension 10.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	SumSquare
CSO									
Best	4.01E-04	1.66E-07	1.96E-09	6.98E-06	5.40E-04	2.57E-07	6.33E-06	1.70E-05	1.58E-04
Average	9.23	8.07E+01	1.85E-04	1.99E-01	1.11E+06	2.56E-04	2.45E-01	2.44E-01	7.35E02
STD	5.45	2.11E+02	2.75E-04	8.90E-01	2.13E+06	2.89E-04	3.18E-04	7.09E-01	1.64E03
SRT	1/20	13/20	20/20	19/20	1/20	20/20	20/20	17/20	6/20
Time	45.42	32.96	2.26	14.81	44.31	5.99	4.0	20.37	35.30
MCSO									
Best	4.37E-10	1.38	1.20E-21	0.00	9.00	1.59E-21	9.90E-23	0.00	1.97E-17
Average	5.16E-07	1.38	5.21E-14	1.85E-12	9.00	1.41E-12	2.10E-14	8.75E-11	1.59E-12
STD	1.11E-06	2.28E-16	1.58E-13	6.72E-12	0.00	6.26E-12	9.07E-14	3.77E-10	4.09E-12
SRT	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.21	49.73	0.12	0.14	37.01	0.143	0.12	0.15	0.12
SCCSO									
Best	8.88E-16	1.38	2.62E-55	0.00	9.00	1.60E-59	8.10E-52	0.00	1.21E-50
Average	1.07E-15	1.38	1.50E-33	0.00	9.00	5.15E-30	8.10E-26	0.00	6.81E-27
STD	7.94E-16	2.28E-16	6.64E-33	0.00	0.00	2.30E-29	3.62E-25	0.00	3.04E-26
SRT	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.13	48.65	0.12	0.14	36.37	0.13	0.12	0.14	0.12

STD denotes standard deviation. Success denotes success rate. Time denotes execution time in seconds.

Table 5.2: Performance of CSO, MCSO and SCCSO for Dimension 20.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	SumSquare
CSO									
Best	2.00E01	3.38E-05	6.13E-10	1.52E-05	1.61E06	3.61E-06	7.00E-08	1.86E-04	1.58E-04
Average	2.14E01	1.27E04	2.61E-04	8.46E03	1.08E11	3.74E-04	2.43E03	9.22	1.68E06
STD	8.75E-01	1.79E04	2.16E-04	1.52E-05	2.36E11	3.59E-04	1.08E04	1.22E01	4.40E06
SRT	0/20	3/20	20/20	8/20	0/20	20/20	18/20	4/20	1/20
Time	63.95	71.33	18.20	49.34	61.48	16.19	19.06	57.47	57.37
MCSO									
Best	1.12E-08	2.29	1.4881E-20	0.00	1.90E01	9.67E-22	7.29E-19	0.00	1.70E-15
Average	1.02E-06	2.29	1.16E-13	2.80E-11	1.90E01	2.46E-15	3.30E-14	2.27E-10	3.34E-11
STD	1.17E-06	9.11E-16	4.17E-13	9.55E-11	0.00	6.06E-15	1.03E-13	1.01E-09	1.00E-10
SRT	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.21	73.729	0.21	0.20	54.506	0.17	0.18	0.18	0.16
SCCSO									
Best	8.88E-16	2.29	5.25E-49	0.00	1.90E01	7.32E-58	2.13E-55	0.00	6.17E-47
Average	8.88E-16	2.29	1.55E-29	0.00	1.90E01	7.66E-35	6.48E-29	0.00	1.19E-31
STD	0.00	9.11E-16	4.77E-29	0.00	0.00	2.25E-34	2.90E-28	0.00	4.12E-31
SRT	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.18	74.22	0.20	0.18	54.59	0.206	0.17	0.17	0.21

STD denotes standard deviation. SRT denotes success rate. Time denotes execution time in seconds.

Table 5.3: Performance of CSO, MCSO and SCCSO for Dimension 30.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	SumSquare
CSO									
Best	1.06	6.40E-04	9.54E-06	6.01-05	4.77E02	6.75E-08	7.14E-10	1.41E-04	2.11E-05
Average	2.01	3.67E05	2.03E03	1.00E04	9.22E11	3.28	2.61E-04	2.19E01	4.49E06
STD	4.71	1.15E06	8.96E03	3.10E04	3.15E12	1.47E01	3.25E-04	3.10E01	1.65E07
SRT	0/20	1/20	18/20	5/20	0/20	19/20	20/20	2/20	1/20
Time	81.17	112.78	29.03	66.77	79.45	27.03	24.49	75.20	75.62
MCSO									
Best	5.05E-12	3.20	1.05E-20	0.00	2.90E01	8.52E-23	7.29E-21	0.00	1.30E-17
Average	8.31E-06	3.20	7.40E-12	1.16E-12	2.90E01	7.80E-13	2.77E-13	2.14E-12	5.54E-11
STD	3.0E-05	4.56E-16	3.31E-11	3.04E-12	0.00	3.40E-12	8.28E-13	8.05E-12	1.19E-10
SRT	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.23	98.69	0.19	0.21	72.34	0.21	0.19	0.226	0.22
SCCSO									
Best	8.88E-16	3.20	5.77E-57	0.00	2.90E01	1.09E-49	4.55E-53	0.00	3.04E-49
Average	5.64E-13	3.20	1.34E-30	0.00	2.90E01	7.10E-30	6.54E-36	0.00	2.28E-29
STD	2.51E-12	4.55E-16	5.86E-30	0.00	0.00	3.15E-29	2.57E-35	0.00	9.90E-29
SRT	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.23	98.59	0.24	0.24	73.55	0.20	0.24	0.31	0.20

STD denotes standard deviation. SRT denotes success rate. Time denotes execution time in seconds.

Table 5.4: Perfomance of CSO, MCSO and SCCSO for Dimension 40.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	SumSquare
CSO									
Best	1.97E01	2.26E-05	8.12E-07	2.50E-07	6.88E02	3.83E-07	2.27E-07	1.24E-05	1.74E02
Average	2.15E01	4.96E04	6.81E01	2.70E03	4.17E13	1.92E01	3.25E-04	1.87E01	3.17E06
STD	7.77E-01	1.35E05	2.10E-02	5.26E03	1.84E14	8.60E01	2.90E-04	2.26E01	3.66E06
SRT	0/20	1/20	18/20	6/20	0/20	19/20	20/20	5/20	0/20
Time	100.53	127.913	39.61	79.75	98.02	34.90	20.30	88.27	95.44
MCSO									
Best	1.21E-10	4.11	1.64E-21	0.00	3.90E01	6.75E-22	2.29E-23	0.00	5.15E-15
Average	3.05E-06	4.11	1.89E-13	2.85E-10	3.90E01	1.53E-13	2.99E-12	1.57E-12	1.71E-09
STD	1.04E-05	9.11E-16	8.31E-13	9.95E-10	0.00	5.03E-13	1.32E-11	3.85E-12	6.78E-09
SRT	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.26	123.833	0.26	0.27	90.43	0.24	0.25	0.24	0.24
SCCSO									
Best	8.88E-16	4.11	2.8806E-55	0.00	3.90E01	4.56E-54	1.40E-50	0.00	9.19E-48
Average	2.49E-15	4.11	1.56E-32	0.00	3.90E01	1.73E-34	8.76E-31	0.00	1.56E-24
STD	6.36E-15	9.11E-16	6.96E-32	0.00	0.00	6.99E-34	3.91E-30	0.00	6.98E-24
SRT	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.25	123.04	0.24	0.25	90.99	0.25	0.27	0.28	0.31

STD denotes standard deviation. SRT denotes success rate. Time denotes execution time in seconds.

Table 5.5: Comparison of Best Performance of CSO, MCSO and SCCSO.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	SumSq	No. of good optimal
Dim 10										
CSO	4.01E-04	**1.66E-07**	1.93E-09	6.97E-06	**5.40E-04**	2.57E-07	6.33E-06	1.79E-05	1.56E-04	2
MCSO	4.37E-10	1.38	1.20E-21	**0.00**	9.00	1.59E-21	9.90E-23	**0.00**	1.97E-17	2
SCCSO	**8.88E-16**	1.38	**2.62E-55**	**0.00**	9.00	**1.60E-59**	**8.11E-52**	**0.00**	**1.21E-50**	7
Dim 20										
CSO	2.00E01	**3.38E-05**	6.13E-10	1.52E-05	1.61E06	3.61E-06	7.00E-08	1.86E-04	1.58E-04	1
MCSO	1.12E-08	2.29	1.49E-20	**0.00**	**1.90E01**	9.67E-22	7.29E-19	**0.00**	1.709E-15	3
SCCSO	**8.88E-16**	2.29	**5.25E-49**	**0.00**	**1.90E01**	**7.32E-58**	**2.15E-55**	**0.00**	**6.17E-47**	8
Dim 30										
CSO	1.06	**6.41E-04**	9.54E-06	6.01E-05	4.77E02	6.75E-08	7.14E-10	1.41E-04	2.11E-05	1
MCSO	5.05E-12	3.20	1.05E-20	**0.00**	**2.90E01**	8.52E-23	7.29E-21	**0.00**	1.30E-17	3
SCCSO	**8.88E-16**	3.20	**5.77E-57**	**0.00**	**2.90E01**	**1.09E-49**	**4.559E-53**	**0.00**	**3.04E-49**	8
Dim 40										
CSO	1.97E01	**2.26E-05**	8.12E-07	2.50E-07	6.88E02	3.83E-07	2.23E-07	1.24E-05	1.74E02	1
MCSO	1.21E-10	4.11	1.64E-21	**0.00**	**3.90E01**	6.75E-22	2.29E-23	**0.00**	5.15E-15	3
SCCSO	**8.88E-16**	4.11	**2.88E-55**	**0.00**	**3.90E01**	**4.56E-54**	**1.40E-50**	**0.00**	**9.19E-48**	8

Total no. of good optimum (best): CSO (5); MCSO (11); SCCSO (31).

Table 5.6: Comparison of Average Performance of CSO, MCSO and SCCSO.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	SumSq	No. of good optimal
Dim 10										
CSO	9.23	8.07E+01	1.85E-04	1.99E-01	1.11E+06	2.56E-04	2.45E-01	2.44E-01	7.35E02	-
MCSO	5.16E-07	**1.38**	5.21E-14	1.85E-12	**9.00**	1.41E-12	2.10E-14	8.75E-11	1.59E-12	2
SCCSO	**1.07E-15**	**1.38**	**1.50E-33**	**0.00**	**9.00**	**5.15E-30**	**8.10E-26**	**0.00**	**6.81E-27**	9
Dim 20										
CSO	2.14E01	1.27E04	2.61E-04	8.46E03	1.08E11	3.74E-04	2.43E03	9.22	1.68E06	-
MCSO	1.02E-06	**2.29**	1.16E-13	2.80E-11	**1.90E01**	2.46E-15	3.30E-14	2.27E-10	3.34E-11	2
SCCSO	**8.88E-16**	**2.29**	**1.55E-29**	**0.00**	**1.90E01**	**7.66E-35**	**6.48E-29**	**0.00**	**1.19E-31**	9
Dim 30										
CSO	2.01	3.67E05	2.03E03	1.00E04	9.22E11	3.28	2.61E-04	2.19E01	4.49E06	-
MCSO	8.31E-06	**3.20**	7.40E-12	1.16E-12	**2.90E01**	7.80E-13	2.77E-13	2.14E-12	5.54E-11	2
SCCSO	**5.64E-13**	**3.20**	**1.34E-30**	**0.00**	**2.90E01**	**7.10E-30**	**6.54E-36**	**0.00**	**2.28E-29**	9
Dim 40										
CSO	2.15E01	4.96E04	6.81E01	2.70E03	4.17E13	1.92E01	3.25E-04	1.87E01	3.17E06	-
MCSO	3.05E-06	**4.11**	1.89E-13	2.85E-10	**3.90E01**	1.53E-13	2.99E-12	1.57E-12	1.71E-09	2
SCCSO	**2.49E-15**	**4.11**	**1.56E-32**	**0.00**	**3.90E01**	**1.73E-34**	**8.76E-31**	**0.00**	**1.56E-24**	9

Total no. of good optimum (average): CSO (0); MCSO (8); SCCSO (36).

Table 5.7: Comparison of STD of Mean optimal for CSO, MCSO and SCCSO.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	SumSq	No. of good STD
Dim 10										
CSO	5.45	2.11E+02	2.75E-04	8.90E-01	2.13E+06	2.89E-04	3.18E-04	7.09E-01	1.64E03	-
MCSO	1.11E-06	**2.28E-16**	1.58E-13	6.72E-12	**0.00**	6.26E-12	9.07E-14	3.77E-10	4.09E-12	2
SCCSO	**7.94E-16**	**2.28E-16**	**6.64E-33**	**0.00**	**0.00**	**2.30E-29**	**3.62E-25**	**0.00**	**3.04E-26**	9
Dim 20										
CSO	8.75E-01	1.79E04	2.16E-04	1.52E-05	2.36E11	3.59E-04	1.08E04	1.22E01	4.40E06	-
MCSO	1.17E-06	**9.11E-16**	4.17E-13	9.55E-11	**0.00**	6.06E-15	1.03E-13	1.01E-09	1.00E-10	2
SCCSO	**0.00**	**9.11E-16**	**4.77E-29**	**0.00**	**0.00**	**2.25E-34**	**2.90E-28**	**0.00**	**4.12E-31**	9
Dim 30										
CSO	4.71	1.15E06	8.96E03	3.10E04	3.15E12	1.47E01	3.25E-04	3.10E01	1.65E07	-
MCSO	3.0E-05	**4.56E-16**	3.31E-11	3.04E-12	**0.00**	3.40E-12	8.28E-13	8.05E-12	1.19E-10	2
SCCSO	**2.51E-12**	**4.55E-16**	**5.86E-30**	**0.00**	**0.00**	**3.15E-29**	**2.57E-35**	**0.00**	**9.90E-29**	9
Dim 40										
CSO	7.77E-01	1.35E05	2.10E-02	5.26E03	1.84E14	8.60E01	2.90E-04	2.26E01	3.66E06	-
MCSO	1.04E-05	**9.11E-16**	8.31E-13	9.95E-10	**0.00**	5.03E-13	1.32E-11	3.85E-12	6.78E-09	2
SCCSO	**6.36E-15**	**9.11E-16**	**6.96E-32**	**0.00**	**0.00**	**6.99E-34**	**3.91E-30**	**0.00**	**6.98E-24**	9

STD denotes standard deviations. Total no. of good STD: CSO (0); MCSO (8); SCCSO (36).

Table 5.8: Comparison of Success Rate of CSO, MCSO and SCCSO.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	SumSq	No of 100% success rate
Dim 10										
CSO	0.05	0.65	1	0.95	0.05	1	1	0.85	0.3	3
MCSO	1	0	1	1	0	1	1	1	1	7
SCCSO	1	0	1	1	0	1	1	1	1	7
Dim 20										
CSO	0	0.15	1	0.4	0	1	0.9	0.2	0.05	2
MCSO	1	0	1	1	0	1	1	1	1	7
SCCSO	1	0	1	1	0	1	1	1	1	7
Dim 30										
CSO	0	0.05	0.9	0.25	0	0.95	1	0.1	0.1	1
MCSO	1	0	1	1	0	1	1	1	1	7
SCCSO	1	0	1	1	0	1	1	1	1	7
Dim 40										
CSO	0	0.05	0.9	0.3	0	0.95	1	0.25	0	1
MCSO	1	0	1	1	0	1	1	1	1	7
SCCSO	1	0	1	1	0	1	1	1	1	7

Total no. of good success rate: CSO (7); MCSO (28); SCCSO (28).

Table 5.9: Comparison of Execution Time (in seconds) for CSO, MCSO and SCCSO.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	SumSq	No of good time
Dim 10										
CSO	45.42	**32.96**	2.26	14.81	44.31	5.99	4.0	20.37	35.30	1
MCSO	0.21	49.73	**0.12**	**0.14**	37.05	0.14	**0.12**	0.15	**0.12**	4
SCCSO	**0.13**	48.65	**0.12**	**0.14**	**36.37**	**0.13**	**0.12**	**0.14**	**0.12**	8
Dim 20										
CSO	63.95	**71.33**	18.20	49.34	61.48	16.19	19.06	57.47	57.37	1
MCSO	0.21	73.73	0.21	0.20	**54.51**	**0.17**	0.18	0.18	**0.16**	3
SCCSO	**0.18**	74.22	**0.20**	**0.18**	54.59	0.21	**0.17**	**0.17**	0.21	5
Dim 30										
CSO	81.17	112.78	29.03	66.77	79.45	27.03	24.49	75.20	75.62	-
MCSO	**0.23**	98.69	**0.19**	**0.21**	**72.34**	0.21	**0.19**	**0.23**	0.22	6
SCCSO	**0.23**	**98.59**	0.24	0.24	73.55	**0.20**	0.24	0.31	**0.20**	4
Dim 40										
CSO	100.53	127.91	39.61	79.75	98.02	34.90	20.30	88.27	95.44	-
MCSO	0.26	123.83	0.26	0.27	**90.43**	**0.24**	**0.25**	**0.24**	**0.24**	5
SCCSO	**0.25**	**123.04**	**0.24**	**0.25**	90.99	0.25	0.27	0.28	0.31	4

Total no. of good execution time: CSO (2); MCSO (18); SCCSO (21).

Table 5.10: Performance of LSRS and SCCSO for Dimension 50

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	SumSqure
LSRS								
Best	-6.5E-19	2.9E-39	6.27E-19	0.00	2.47E-28	1.86E-11	1.34E-22	9.86E-21
Average	-6.5E-19	2.9E-39	2.33E-18	0.00	1.38E-18	1.91E-11	1.38E-18	1.42E-18
STD	0.00	0.00	8.11E-19	0.00	1.29E-18	4.15E-12	1.29E-18	1.25E-18
SCCSO								
Best	8.88E-16	5.01	5.03E-54	0.00	4.90E01	5.61E-51	1.47E-51	4.43E-45
Average	8.88E-16	5.01	7.25E-31	0.00	4.90E01	1.02E-28	2.62E-28	4.54E-28
STD	0.00	9.15E-16	3.06E-30	0.00	0.00	4.54E-28	1.17E-27	1.98E-27

Table 5.11: Performance of LSRS and SCCSO for Dimension 100

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	SumSqure
LSRS								
Best	-6.5E-19	2.9E-39	9.20E-16	0.00	5.83E-28	7.81E-19	5.34E-19	4.68E-18
Average	-6.5E-19	2.9E-39	1.15E-15	0.00	6.94E-16	3.98E-10	6.94E-16	6.98E-16
STD	0.00	0.00	4.38E-16	0.00	6.63E-16	4.97E-10	6.63E-16	6.58E-16
SCCSO								
Best	8.88E-16	9.56	1.40E-54	0.00	9.90E01	1.43E-51	1.93E-49	4.47E-53
Average	7.63E-14	9.56	3.35E-27	0.00	9.90E01	1.54E-29	1.94E-27	1.95E-27
STD	3.36E-13	3.65E-15	1.50E-26	0.00	0.00	6.46E-29	8.67E-27	5.98E-27

Table 5.12: Performance of LSRS and SCCSO for Dimension 500

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	SumSqure
LSRS								
Best	-4.3E-19	2.9E-39	2.12E-11	0.00	3.4E-27	2.91E-19	4.54E-16	4.05E-35
Average	-4.3E-19	2.9E-39	4.31E-11	0.00	2.61E-11	4.08E-19	9.0E-16	7.96E-35
STD	0.00	0.00	1.14E-11	0.00	2.32E-11	3.62E-20	1.52E-16	1.95E-35
SCCSO								
Best	8.88E-16	4.59E01	2.52E-54	0.00	4.99E02	1.15E-55	1.27E-49	4.56E-46
Average	1.42E-15	4.59E01	2.23E-32	0.00	4.99E02	3.37E-31	6.14E-23	5.18E-22
STD	1.74E-15	0.00	6.86E-32	0.00	0.00	1.50E-30	2.75E-22	2.31E-21

Table 5.13: Performance of LSRS and SCCSO for Dimension 1000

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	SumSqure
LSRS								
Best	1.3E-18	2.9E-39	5.34E-30	0.00	6.84E-27	9.30E-18	7.97E-19	3.78E-33
Average	1.3E-18	2.9E-39	1.38E-29	0.00	7.41E-27	1.12E-17	1.25E-18	7.35E-33
STD	4.8E-33	0.00	3.68E-30	0.00	1.66E-28	7.33E-19	2.05E-19	1.49E-33
SCCSO								
Best	8.88E-16	9.13E01	2.64E-47	0.00	10.0E02	1.94E-47	3.59E-52	3.04E-45
Average	2.84E-15	9.13E01	3.26E-33	0.00	10.0E02	2.68E-30	1.61E-30	8.14E-25
STD	8.74E-15	1.46E-14	1.46E-32	0.00	0.00	7.87E-30	7.17E-30	3.62E-24

Table 5.14: Performance of LSRS and SCCSO for Dimension 2000

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	SumSqure
LSRS								
Best	-4.3E-19	2.9E-39	9.37E-8	0.00	1.40E-26	2.41E-17	9.97E-34	7.58E-31
Average	-4.3E-19	2.9E-39	1.69E-7	0.00	1.48E-26	3.08E-17	2.35E-33	1.27E-30
STD	9.6E-35	0.00	3.42E-8	0.00	2.44E-28	2.31E-18	6.91E-34	2.02E-31
SCCSO								
Best	8.88E-16	1.82E02	6.63E-56	0.00	2.0E03	6.01E-55	5.25E-56	4.24E-41
Average	1.07E-16	1.82E02	1.92E-29	0.00	2.00E03	9.85E-28	1.19E-30	1.29E-22
STD	7.44E-16	5.83E-14	8.32E-29	0.00	0.00	4.23E-27	5.53E-30	5.65E-22

Table 5.15: Performance of SCCSO for Dimension 3000.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	SumSqure
Best	8.88E-16	2.73E01	3.14E-52	0.00	3.0E03	2.35E-53	1.67E-52	4.94E-46
Average	3.02E-15	2.73E01	3.097E-27	0.00	3.0E03	4.66E-30	7.54E-31	2.40E-21
STD	8.73E-15	0.00	1.38E-26	0.00	0.00	2.00E-29	2.51E-31	1.07E-20

Table 5.16: Average Performance of LSRS and SCCSO

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	SumSqure
Dim 50								
LSRS	-6.5E-19	2.9E-39	2.33E-18	0.00	1.38E-18	1.91E-11	1.38E-18	1.42E-18
SCCSO	8.88E-16	5.01	7.25E-31	0.00	4.90E01	1.02E-28	2.62E-28	4.54E-28
Dim 100								
LSRS	-6.5E-19	2.9E-39	1.15E-15	0.00	6.94E-16	3.98E-10	6.94E-16	6.98E-16
SCCSO	7.63E-14	9.56	3.35E-27	0.00	9.90E01	1.54E-29	1.94E-27	1.95E-27
Dim 500								
LSRS	-4.3E-19	2.9E-39	4.31E-11	0.00	2.61E-11	4.08E-19	9.0E-16	7.96E-35
SCCSO	1.42E-15	4.59E01	2.23E-32	0.00	4.99E02	3.37E-31	6.14E-23	5.18E-22
Dim 1000								
LSRS	1.3E-18	2.9E-39	1.38E-29	0.00	7.41E-27	1.12E-17	1.25E-18	7.35E-33
SCCSO	2.84E-15	9.13E01	3.26E-33	0.00	10.0E02	2.68E-30	1.61E-30	8.14E-25
Dim 2000								
LSRS	-4.3E-19	2.9E-39	1.69E-7	0.00	1.48E-26	3.08E-17	2.35E-33	1.27E-30
SCCSO	1.07E-16	1.82E02	1.92E-29	0.00	2.00E03	9.85E-28	1.19E-30	1.29E-22

Table 5.17: ANOVA test for SCCSO and LSRS

		Sum of Square	df	Mean Square	F	Sig.
Ackley	Between Groups	.000	1	.000	1.182	.309
	Within Groups	.000	8	.000		
	Total	.000	9			
Quadric	Between Groups	.000	1	.000	1.001	.346
	Within Groups	.000	8	.000		
	Total	.000	9			
Levy	Between Groups	11140.241	1	11140.241	4.164	.076
	Within Groups	21402.511	8	2675.314		
	Total	32542.752	9			
Rastrigin	Between Groups	.000	1	.000		
	Within Groups	.000	8	.000		
	Total	.000	9			
Rosenbrock	Between Groups	1330061	1	1330050.900	4.091	.076
	Within Groups	2601081	8	325135.150		
	Total	3931142	9			
Schwefel	Between Groups	.000	1	.000	1.123	.320
	Within Groups	.000	8	.000		
	Total	.000	9			
Sphere	Between Groups	.000	1	.000	2.609	.145
	Within Groups	.000	8	.000		
	Total	.000	9			
SumSq	Between Groups	.000	1	.000	1.005	.345
	Within Groups	.000	8	.000		
	Total	.000	9			

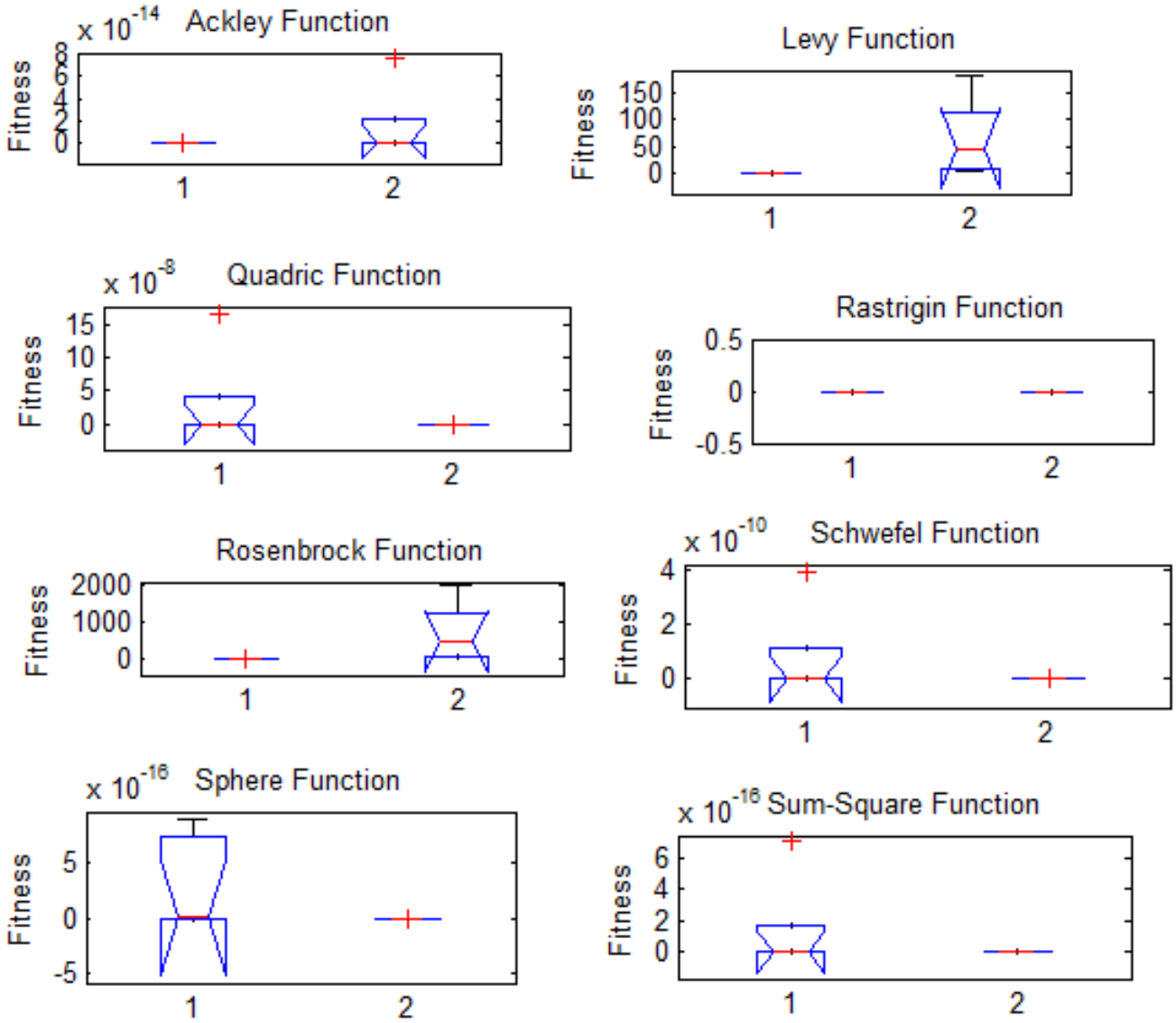

Figure 5.1: ANOVA test for LSRS and SCCSO.

5.3 An Improved Cockroach Swarm Optimization

In section 3.3.1, the proposed ICSO model is shown. Simulation studies were carried out on the proposed algorithm and its performance was evaluated on 23 global optimization benchmark problems presented in Table 17 of Appendix A. A series of experiments were conducted to compare the performance of ICSO with that of the existing cockroach algorithms namely: Roach infestation optimization (RIO), Hungry roach infestation optimization (HRIO), CSO and MCSO. The results of the experiments are shown in Tables 5.18, 5.19 and 5.20.

The comparison results of the best, average and standard deviation performance are shown in Tables 5.22, 5.21, 5.23 while the comparison results of the success performance and execution time in seconds utilized by each algorithm to solve the respective benchmarks, are shown in Tables 5.24 and 5.25.

The test statistic of Jonckheere-Terpstra (J-T) was conducted to determine if the performance of ICSO algorithm is significantly different from the comparison algorithms. The test statistic results in Table 5.26 show that the proposed algorithm performs better than others. The J-T test statistic P-value (Asymp. Sig.) was computed to be 0.001 in Table 5.26. The P-Value is less than the threshold value 0.05; therefore, there is a significant difference in performance of ICSO and that of RIO, HRIO, CSO and MCSO.

The magnitude of the observed effect of the significant difference is measured by computing the effect size. The effect size r, $(1 > r < 0)$ of the significant difference of J-T test was calculated as: [114]

$r = \frac{Z}{\sqrt{N}}$,

where Z is the standard data of J-T statistic as shown in Table 5.26, N is the total number of samples, $N = 114$.

$Z = \frac{x-\mu}{\sigma}$,

x denotes J-T Statistic observed, μ denotes J-T statistic mean and σ denoted the standard deviation of J-T statistic.

$Z = \frac{1952-2599}{199.355} = -3.245,$

$r = \frac{-3.245}{\sqrt{114}} = -0.3.$

Absolute value of $|Z|$ is the distance between the observed data and the mean in units of standard deviation (Z is negative when observed data is below the mean and positive when above). Using Cohen's guideline on effect size, the effect size 0.3 is of medium size [132, 133]. The statistics of 0.3 effect size shows there is significant difference of medium magnitude between the proposed algorithm and the existing algorithms.

The ICSO algorithm is shown [114] to have better performance, minimum STD, minimum

execution time and good success rates than the comparison algorithms.

Table 5.18: Simulation results of RIO, HRIO, CSO, MCSO and ICSO

SN	Fn	Dim	Opt		RIO	HRIO	CSO	MCSO	ICSO
1	Boha1	2	0	Ave	3.4405E-05	3.2877E-04	2.9893E02	3.5153E-09	0.0000
				STD	2.5963E-05	3.0334E-04	5.0332E02	1.4392E-08	0.0000
				Best	1.3520E-07	5.2651E-06	2.0651E-05	0.0000	0.0000
				SRT	20/20	20/20	5/20	20/20	20/20
				Time	1.137525	0.886356	23.913237	0.075212	0.097187
2	Boha2	2	0	Ave	4.2829E-05	4.6703E-04	9.0941E02	8.4459E-12	0.0000
				STD	3.0070E-05	3.4047E-04	1.7794E03	2.9240E-11	0.0000
				Best	2.2910E-06	9.374E-06	1.3775E-05	0.0000	0.0000
				SRT	20/20	20/20	4/20	20/20	20/20
				Time	0.998178	0.946887	26.492095	0.072021	0.074106
3	Boha3	2	0	Ave	5.3479E-05	4.7575E-04	7.4284E02	2.1388E-14	0.0000
				STD	2.9141E-05	2.3273E-04	1.6739E03	4.8670E-14	0.0000
				Best	3.1200E-06	4.6981E-05	2.3093E-07	0.0000	0.0000
				SRT	20/20	20/20	3/20	20/20	20/20
				Time	1.089920	0.885252	25.028054	0.080908	0.068189
4	3camel	2	0	Ave	1.4962E-02	4.3021E-04	5.003E09	7.098E-11	5.9853E-31
				STD	6.6769E-02	2.8371E-04	1.7137E10	3.0201E-10	2.5457E-30
				Best	1.1739E-06	2.2449E-05	1.7642E-05	3.1395E-19	2.2320E-53
				SRT	19/20	20/20	12/20	20/20	20/20
				Time	4.231533	0.794983	18.281683	0.104132	0.078845
5	6camel	2	-1.03163	Ave	-4.3522E-01	-4.7652E-01	1.5763E05	-1.0263E-08	-2.9798E-25
				STD	3.3322E-01	3.1284E-01	7.0503E05	4.4391E-08	1.3325E-24
				Best	-1.0215	-1.0034	-9.4052E-01	-1.9879E-07	-5.9589E-24
				SRT	20/20	20/20	19/20	20/20	20/20
				Time	0.406355	0.330198	5.723039	0.0945856	0.086637
6	Easom	2	-1	Ave	-1	-1	-4.3165E-01	-1	-1
				STD	3.7518E-02	2.1031E-02	3.4470E-01	1.4897E-08	4.4116E-17
				Best	-1	-1	-1	-1	-1
				SRT	20/20	20/20	20/20	20/20	20/20
				Time	0.124022	0.107303	0.106738	0.077179	0.092393
7	Matyax	2	0	Ave	4.9470-05	3.2297E-04	7.5712	2.6876E-13	4.0732E-35
				STD	3.0244E-05	2.6018E-04	1.1247E01	8.9347E-13	1.8125E-34
				Best	6.2897E-06	1.2684E-05	8.8777E-06	6.6695E-21	1.1292E-55
				SRT	20/20	20/20	11/20	20/20	20/20
				Time	0.973322	0.711734	13.559576	0.88536	0.076693
8	Schaffer1	2	-1	Ave	-1.9069	-1.6211	-2.9174E-01	-1	-1
				STD	7.0381E-01	5.9214E-01	7.5142E-01	5.9575E-07	4.1325E-15
				Best	-2.7458	-2.7164	-2.7438	-1	-1
				SRT	20/20	20/20	20/20	20/20	20/20
				Time	0.109048	0.086433	0.119076	0.072400	0.081599
9	Schaffer2	2	0	Ave	2.0179E-03	1.6566E-03	7.1618	3.3168E-04	2.2149E-09
				STD	2.6407E-03	1.4451E-03	5.3095	3.0328E-04	2.9483E-09
				Best	6.2423E-05	4.1422E-04	2.8354E-01	1.5810E-05	1.9383E-14
				SRT	2/20	13/20	0/20	20/20	20/20
				Time	62.567654	31.415836	29.194283	0.084127	0.082320

Dim denotes Dimension. Opt denotes Optimum Value. SRT denotes success rate. Boha1 denotes Bohachevsky1. Boha2 denotes Bohachevsky2. Boha3 denotes Bohachevsky3. 3camel denotes Three hump camel back. 6camel denotes Six hump camel back.

Table 5.19: Simulation results of RIO, HRIO, CSO, MCSO and ICSO

SN	Fn	Dim	Opt		RIO	HRIO	CSO	MCSO	ICSO
10	Sphere	30	0	Ave	2.2168E-05	1.6676E-04	1.8123E02	1.5201E-12	3.3448E-34
				STD	2.4528E-05	2.4018E-04	8.1048E02	6.7224E-12	1.3324E-33
				Best	5.7627E-09	5.5635E-08	4.9195E-07	2.9978E-24	2.8205E-54
				SRT	20/20	20/20	19/20	20/20	20/20
				Time	0.617544	0.557871	25.378161	0.82512	0.199373
11	Rastrigin	30	0	Ave	3.8135E-05	3.2150E-04	3.6022E03	9.1994E-11	0.0000
				STD	3.4436E-05	3.0003E-04	5.5728E03	3.9456E-10	0.0000
				Best	2.7098E-07	2.1450E-07	3.1340E-04	0.0000	0.0000
				SRT	20/20	20/20	5/20	20/20	20/20
				Time	0.956329	0.826770	71.811170	0.175563	0.369987
12	Rosenbrock	30	0	Ave	2.5281E06	3.3571E06	9.5067E11	2.9000E01	2.9000E01
				STD	4.0528E06	7.1150E06	2.2713E12	0.0000	0.0000
				Best	1.6773E04	3.7562E04	4.4068E01	2.9000E01	2.9000E01
				SRT	0/20	0/20	0/20	0/20	0/20
				Time	126.618734	127.469638	81.361663	76.084929	78.572185
13	Ackley	30	0	Ave	2.0001E01	2.0005E01	1.9222E01	5.1593E-06	1.0651E-15
				STD	3.0455E-03	1.5671E-02	5.8258	1.9149E-05	7.9441E-16
				Best	2.0001E01	1.9998E01	2.0133E01	6.4623E-09	8.1818E-16
				SRT	0/20	0/20	0/20	20/20	20/20
				Time	122.216187	117.635854	82.227210	0.235012	0.192339
14	Quadric	30	0	Ave	2.4498E-05	2.2711E-04	3.4991E-04	4.4754E-13	7.2183E-28
				STD	2.7957E-05	2.3635E-04	3.3725E-04	1.9751E-12	3.2218E-27
				Best	1.1360E-08	5.8230E-07	4.1551E-08	5.6309E-23	5.910E-52
				SRT	20/20	20/20	20/20	20/20	20/20
				Time	0.718785	0.512242	31.075809	0.247456	0.227244
15	Schwefel2.22	30	0	Ave	2.3131E02	2.4395E02	2.9013E54	6.3587E-06	6.0407E-16
				STD	1.3193E02	1.2341E02	1.2971E55	1.1936E-05	1.2203E-15
				Best	6.7400E01	1.7354E01	3.6854E01	5.9410E-08	5.1670E-24
				SRT	0/20	0/20	0/20	20/20	20/20
				Time	128.445013	127.084387	79.924516	0.217104	0.219296
16	Griewangk	30	0	Ave	7.9510E-01	7.7746E-01	2.6148E01	3.3151E-11	0.0000
				STD	3.7583E-01	2.5454E-01	3.6626E01	1.4672E-10	0.0000
				Best	2.9324E-01	3.2031E-01	6.3912E-05	0.0000	0.0000
				SRT	0/20	0/20	5/20	20/20	20/20
				Time	126.872461	126.210153	70.852376	0.211351	0.210934
17	Sumsquare	30	0	Ave	1.9818E03	4.6771E03	9.0499E05	4.2446E-11	1.5600E-24
				STD	2.8370E03	6.7104E03	1.0253E06	1.2930E-10	6.9785E-24
				Best	1.6463E01	2.0516E02	1.8730E02	1.49990E-16	1.3765E-47
				SRT	0/20	0/20	0/20	20/20	20/20
				Time	122.748646	125.154349	78.809270	0.273780	0.236129
18	Sinusoidal	30	-3.5	Ave	-4.2587E-01	-3.7898E-01	-2.449	-3.1030	-3.1030
				STD	2.6632E-01	1.9791E-01	1.0203	5.0473E-05	1.9436E-14
				Best	-1.1922	-8.3111E-01	-3.3087	-3.1032	-3.1030
				SRT	20/20	20/20	20/20	20/20	20/20
				Time	0.204559	0.240200	0.234205	0.200361	0.217635

Dim denotes Dimension. SRT denotes success rate. Opt denotes Optimum Value.

Table 5.20: Simulation results of RIO, HRIO, CSO, MCSO and ICSO

SN	Function	Dim	Opt		RIO	HRIO	CSO	MCSO	ICSO
19	Zakharov	30	0	Ave	1.0167E04	1.0216E04	6.3663E18	2.3878E-09	4.1579E-26
				STD	3.8643E03	5.1012E03	2.2732E19	8.8529E-09	1.8549E-25
				Best	2.6634E03	2.3151E03	1.3578E09	2.0954E-15	6.3965E-57
				SRT	0/20	0/20	0/20	20/20	20/20
				Time	115.192226	114.691827	79.926232	0.205280	0.259202
20	Step	30	0	Ave	0.0000	0.0000	2.0004E04	0.0000	0.0000
				STD	0.0000	0.0000	8.4815E04	0.0000	0.0000
				Best	0.0000	0.0000	0.0000	0.0000	0.0000
				SRT	20/20	20/20	16/20	20/20	20/20
				Time	0.686403	0.633264	39.136696	0.239525	0.225102
21	Powell	24	0	Ave	1.8348E-03	3.7434E-03	1.0840E08	2.6031E-12	1.8207E-24
				STD	1.6248E-03	6.1711E-03	4.1180E08	6.9959E-12	5.6824E-24
				Best	9.6693E-05	6.8033E-04	5.2392E01	1.2287E-19	1.2265E-54
				SRT	2/20	12/20	0/20	20/20	20/20
				Time	122.796991	92.876086	74.794730	1.527170	0.853751
22	ST	9	0	Ave	0.0000	0.0000	0.0000	0.0000	0.0000
				STD	0.0000	0.0000	0.0000	0.0000	0.0000
				Best	0.0000	0.0000	0.0000	0.0000	0.0000
				SRT	20/20	20/20	20/20	20/20	20/20
				Time	0.435911	0.426320	0.437944	0.431122	0.436741
23	ST	17	0	Ave	0.0000	0.0000	0.0000	0.0000	0.0000
				STD	0.0000	0.0000	0.0000	0.0000	0.0000
				Best	0.0000	0.0000	0.0000	0.0000	0.0000
				SRT	20/20	20/20	20/20	20/20	20/20
				Time	1.066161	1.052169	1.159830	1.089657	1.147114

Dim denotes Dimension. SRT denotes success rate. Opt denotes Optimum Value.

Table 5.21: Comparison of average performance of RIO, HRIO, CSO, MCSO and ICSO

SN	Function	RIO	HRIO	CSO	MCSO	ICSO	Optimum
1	Bohachevsky1	3.4405E-05	3.2877E-04	2.9893E02	3.5153E-09	**0.0000**	0
2	Bohachevsky2	4.2829E-05	4.6703E-04	9.0941E02	8.4459E-12	**0.0000**	0
3	Bohachevsky3	5.3479E-05	4.7575E-04	7.4284E02	2.1388E-14	**0.0000**	0
4	3 Hump camel back	1.4962E-02	4.3021E-04	5.003E09	7.098E-11	**5.9853E-31**	0
5	6 Hump camel back	-4.3522E-01	-4.7652E-01	1.5763E05	-1.0263E-08	**-2.9798E-25**	-1.03163
6	Easom	**-1**	**-1**	-4.3165E-01	**-1**	**-1**	-1
7	Matyax	4.9470-05	3.2297E-04	7.5712	2.6876E-13	**4.0732E-35**	0
8	Schaffer1	-1.9069	-1.6211	-2.9174E-01	**-1**	**-1**	-1
9	Schaffer2	2.0179E-03	1.6566E-03	7.1618	3.3168E-04	**2.2149E-09**	0
10	Sphere	2.2168E-05	1.6676E-04	1.8123E02	1.5201E-12	**3.3448E-34**	0
11	Rastrigin	3.8135E-05	3.2150E-04	3.6022E03	9.1994E-11	**0.0000**	0
12	Rosenbrock	2.5281E06	3.3571E06	9.5067E11	**2.9000E01**	**2.9000E01**	0
13	Ackley	2.0001E01	2.0005E01	1.9222E01	5.1593E-06	**1.0651E-15**	0
14	Quadric	2.4498E-05	2.2711E-04	3.4991E-04	4.4754E-13	**7.2183E-28**	0
15	Schwefel2.22	2.3131E02	2.4395E02	2.9013E54	6.3587E-06	**6.0407E-16**	0
16	Griewangk	7.9510E-01	7.7746E-01	2.6148E01	3.3151E-11	**0.0000**	0
17	Sumsquare	1.9818E03	4.6771E03	9.0499E05	4.2446E-11	**1.5600E-24**	0
18	Sinusoidal	-4.2587E-01	-3.7898E-01	-2.449	**-3.1030**	**-3.1030**	-3.5
19	Zakharov	1.0167E04	1.0216E04	6.3663E18	2.3878E-09	**4.1579E-26**	0
20	Step	**0.0000**	**0.0000**	2.0004E04	**0.0000**	**0.0000**	0
21	Powell	1.8348E-03	3.7434E-03	1.0840E08	2.6031E-12	**1.8207E-24**	0
22	ST9	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0
23	ST17	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0
No. of good optimum		4	4	2	7	23	

ST9 denotes Storn's Tchebychev 9. ST17 denotes Storn's Tchebychev 17.

Table 5.22: Comparison of best performance of RIO, HRIO, CSO, MCSO, ICSO.

SN	Function	RIO	HRIO	CSO	MCSO	ICSO	Optimum
1	Bohachevsky1	1.3520E-07	5.2651E-06	2.0651E-05	**0.0000**	**0.0000**	0
2	Bohachevsky2	2.2910E-06	9.374E-06	1.3775E-05	**0.0000**	**0.0000**	0
3	Bohachevsky3	3.1200E-06	4.6981E-05	2.3093E-07	**0.0000**	**0.0000**	0
4	3 Hump camel back	1.1739E-06	2.2449E-05	1.7642E-05	3.1395E-19	**2.2320E-53**	0
5	6 Hump camel back	-1.0215	-1.0034	-9.4052E-01	-1.9879E-07	**-5.9589E-24**	-1.03163
6	Easom	**-1**	**-1**	**-1**	**-1**	**-1**	-1
7	Matyax	6.2897E-06	1.2684E-05	8.8777E-06	6.6695E-21	**1.1292E-55**	0
8	Schaffer1	-2.7458	-2.7164	-2.7438	**-1**	**-1**	-1
9	Schaffer2	6.2423E-05	4.1422E-04	2.8354E-01	1.5810E-05	**1.9383E-14**	0
10	Sphere	5.7627E-09	5.5635E-08	4.9195E-07	2.9978E-24	**2.8205E-54**	0
12	Rosenbrock	1.6773E04	3.7562E04	4.4068E01	**2.9000E01**	**2.9000E01**	0
14	Quadric	1.1360E-08	5.8230E-07	4.1551E-08	5.6309E-23	**5.910E-52**	0
15	Schwefel2.22	6.7400E01	1.7354E01	3.6854E01	5.9410E-08	**5.1670E-24**	0
16	Griewangk	2.9324E-01	3.2031E-01	6.3912E-05	**0.0000**	**0.0000**	0
17	Sumsquare	1.6463E01	2.0516E02	1.8730E02	1.49990E-16	**1.3765E-47**	0
18	Sinusoidal	-1.1922	-8.3111E-01	**-3.3087**	-3.1032	-3.1030	-3.5
19	Zakharov	2.6634E03	2.3151E03	1.3578E09	2.0954E-15	**6.3965E-57**	0
20	Step	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0
21	Powell	9.6693E-05	6.8033E-04	5.2392E01	1.2287E-19	**1.2265E-54**	0
22	ST9	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0
23	ST17	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0
No. of good optimum		4	4	5	11	22	

ST9 denotes Storn's Tchebychev 9. ST17 denotes Storn's Tchebychev 17.

Table 5.23: Comparison of RIO, HRIO, CSO, MCSO and ICSO Mean STD.

SN	Function	RIO	HRIO	CSO	MCSO	ICSO
1	Bohachevsky1	2.5963E-05	3.0334E-04	5.0332E02	1.4392E-08	**0.0000**
2	Bohachevsky2	3.0070E-05	3.4047E-04	1.7794E03	2.9240E-11	**0.0000**
3	Bohachevsky3	2.9141E-05	2.3273E-04	1.6739E03	4.8670E-14	**0.0000**
4	3 Hump camel back	6.6769E-02	2.8371E-04	1.7137E10	3.0201E-10	**2.5457E-30**
5	6 Hump camel back	3.3322E-01	3.1284E-01	7.0503E05	4.4391E-08	**1.3325E-24**
6	Easom	3.7518E-02	2.1031E-02	3.4470E-01	1.4897E-08	**4.4116E-17**
7	Matyax	3.0244E-05	2.6018E-04	1.1247E01	8.9347E-13	**1.8125E-34**
8	Schaffer1	7.0381E-01	5.9214E-01	7.5142E-01	5.9575E-07	**4.1325E-15**
9	Schaffer12	2.6407E-03	1.4451E-03	5.3095	3.0328E-04	**2.9483E-09**
10	Sphere	2.4528E-05	2.4018E-04	8.1048E02	6.7224E-12	**1.3324E-33**
11	Rastrigin	3.4436E-05	3.0003E-04	5.5728E03	3.9456E-10	**0.0000**
12	Rosenbrock	4.0528E06	7.1150E06	2.2713E12	**0.0000**	**0.0000**
13	Ackley	3.0455E-03	1.5671E-02	5.8258	1.9149E-05	**7.9441E-16**
14	Quadric	2.7957E-05	2.3635E-04	3.3725E-04	1.9751E-12	**3.2218E-27**
15	Schwefel2.22	1.3193E02	1.2341E02	1.2971E55	1.1936E-05	**1.2203E-15**
16	Griewangk	3.7583E-01	2.5454E-01	3.6626E01	1.4672E-10	**0.0000**
17	SumSquare	2.8370E03	6.7104E03	1.0253E06	1.2930E-10	**6.9785E-24**
18	Sinusoidal	2.6632E-01	1.9791E-01	1.0203	5.0473E-05	**1.9436E-14**
19	Zakharov	3.8643E03	5.1012E03	2.2732E19	8.8529E-09	**1.8549E-25**
20	Step	0.0000	0.0000	8.4815E04	**0.0000**	**0.0000**
21	Powell	1.6248E-03	6.1711E-03	4.1180E08	6.9959E-12	**5.6824E-24**
22	ST9	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
23	ST17	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
No. of good STD		2	2	2	4	23

STD denotes standard deviation. ST9 denotes Storn's Tchebychev 9. ST17 denotes Storn's Tchebychev 17.

Table 5.24: Comparison of success rate of RIO, HRIO, CSO, MCSO and ICSO.

SN	Function	RIO	HRIO	CSO	MCSO	ICSO
1	Bohachevsky1	1	1	2.5	1	1
2	Bohachevsky2	1	1	0.2	1	1
3	Bohachevsky3	1	1	0.15	1	1
4	3 Hump camel back	0.95	1	0.6	1	1
5	6 Hump camel back	1	1	0.95	1	1
6	Easom	1	1	1	1	1
7	Matyax	1	1	0.55	1	1
8	Schaffer1	1	1	1	1	1
9	Schaffer2	0.1	0.65	0	1	1
10	Sphere	1	1	0.95	1	1
11	Rastrigin	1	1	0.25	1	1
12	Rosenbrock	0	0	0	0	0
13	Ackley	0	0	0	1	1
14	Quadric	1	1	1	1	1
15	Schwefel2.22	0	0	0	1	1
16	Griewangk	0	0	0.25	1	1
17	Sumsquare	0	0	0	1	1
18	Sinusoidal	1	1	1	1	1
19	Zakharov	0	0	0	1	1
20	Step	1	1	0.8	1	1
21	Powell	0.1	0.6	0	1	1
22	ST9	1	1	1	1	1
23	ST17	1	1	1	1	1
No. of 100% Success rate		14	15	6	22	22

ST9 denotes Storn's Tchebychev 9. ST17 denotes Storn's Tchebychev 17.

Table 5.25: Comparison of execution time of RIO, HRIO, CSO, MCSO, ICSO.

SN	Function	RIO	HRIO	CSO	MCSO	ICSO
1	Bohachevsky1	1.137525	0.886356	23.913237	**0.075212**	0.097187
2	Bohachevsky2	0.998178	0.946887	26.492095	**0.072021**	0.074106
3	Bohachevsky3	1.089920	0.885252	25.028054	0.080908	**0.068189**
4	3 Hump camel back	4.231533	0.794983	18.281683	0.104132	**0.078845**
5	6 Hump camel back	0.406355	0.330198	5.723039	0.0945856	**0.086637**
6	Easom	0.124022	0.107303	0.106738	**0.077179**	0.092393
7	Matyax	0.973322	0.711734	13.559576	0.88536	**0.076693**
8	Schaffer1	0.109048	0.086433	0.119076	**0.072400**	0.081599
9	Schaffer2	62.567654	31.415836	29.194283	0.084127	**0.082320**
10	Sphere	0.617544	0.557871	25.378161	0.82512	**0.199373**
11	Rastrigin	0.956329	0.826770	71.811170	**0.175563**	0.369987
12	Rosenbrock	126.618734	127.469638	81.361663	**76.084929**	78.572185
13	Ackley	122.216187	117.635854	82.227210	0.235012	**0.192339**
14	Quadric	0.718785	0.512242	31.075809	0.247456	**0.227244**
15	Schwefel2.22	128.445013	127.084387	79.924516	**0.217104**	0.219296
16	Griewangk	126.872461	126.210153	70.852376	0.211351	**0.210934**
17	Sumsquare	122.748646	125.154349	78.809270	0.273780	**0.236129**
18	Sinusoidal	0.204559	0.240200	0.234205	**0.200361**	0.217635
19	Zakharov	115.192226	114.691827	79.926232	**0.205280**	0.259202
20	Step	0.686403	0.633264	39.136696	0.239525	**0.225102**
21	Powell	122.796991	92.876086	74.794730	1.527170	**0.853751**
22	ST9	0.435911	**0.426320**	0.437944	0.431122	0.436741
23	ST17	1.066161	**1.052169**	1.159830	1.089657	1.147114
No. of minimum execution time		-	2	-	9	12

ST9 denotes Storn's Tchebychev 9. ST17 denotes Storn's Tchebychev 17.

Table 5.26: Jonckheere-Terpstra Test statistics.

Jonckheere-Terpstra Test statistics[a]

	Fitness
Number of levels in Algorithm	5
N	114
Observed J-T Statistic	1952.000
Mean J-T Statistic	2599.500
STD of J-T Statistic	199.355
Standard Data of J-T Statistic	-3.245
Asymp. Sig. (2-tailed)	0.001

a Grouping variable: Algorithm

5.4 A Dynamic Step-Size Adaptation Roach Infestation Optimization

Simulation studies were conducted to evaluate the proposed DSARIO algorithm presented in section 3.4. Its performance is evaluated on a set of global optimization test functions shown in Table 18 of Appendix A for dimensions 2, 10 and 30. Its performance was compared with the existing RIO and HRIO statistically in Table 5.28. Figures 5.2 and 5.3 illustrate graphically the comparison performance of DSARIO, RIO and HRIO. The comparison results show that DSARIO performs better than RIO and HRIO.

The test statistic of ANOVA was carried out to determine the significant difference in the performance of DSARIO and RIO with HRIO. The statistic p-value computed from the test is 0.003 as shown in Table 5.29. Since the p-value of the test is less than the literature threshold p-value 0.05, it means that there is significant difference in the performance of DSARIO and comparison algorithms. The effect size of the significant difference was calculated.

η^2=sum of square (between groups)/Total sum of square. From Table 5.29, $\eta^2 = 194.643/438.156 = 0.44$. The effect size 0.44 is very large, according to Cohen's guideline on effect size [132, 133]. The magnitude of the difference between DSARIO and comparison algorithms is very large.

We tested DSARIO on scalable benchmarks up to dimensions 30 as shown in Table 5.27. The algorithm consistently solves the benchmarks to dimension 30 and found optimal minimum values. The graphical illustration of DSARIO performance for dimensions 2, 10 and 30 is shown in Figure 5.3.

The dynamic step-size adaptation method improved the performance of DSARIO, the improved performance was proved experimentally and statistically over the existing RIO and HRIO [134].

Table 5.27: Performance of DSARIO for 2, 10 and 30 dimensions.

Algorithm	Sphere	Rosenbrock	Rastrigin	Griewangk	Ackley	Michalewicz	Easom	Hump
Dim 2								
Average	0.00	7.29E-01	0.00	0.00	8.89E-16	-1.2750	-1	3.45E-01
STD	0.00	2.38E-01	0.00	0.00	0.00	2.95E-01	1.64E-06	3.48E-01
SRT	20/20	0/20	20/20	20/20	20/20	20/20	20/20	0/20
Dim 10								
Average	0.00	8.98	0.00	0.00	8.89E-16	-2.49	-	-
STD	0.00	1.49E-02	0.00	0.00	0.00	5.02	-	-
SRT	20/20	0/20	20/20	20/20	20/20	20/20	-	-
Dim 30								
Average	0.00	2.90E01	0.00	0.00	8.89E-16	-	-	-
STD	0.00	6.25E-03	0.00	0.00	0.00	-	-	-
SRT	20/20	0/20	20/20	20/20	20/20	-	-	-

STD denotes Standard Deviation. SRT denotes Success Rate

Table 5.28: Comparison of Average Optimum Values of HRIO, RIO and DSARIO.

Test Function	Dimensions	RIO	HRIO	DSARIO	Actual Value
Sphere	2	**0.00**	6.50E-23	**0.00**	0
Sphere	10	2.70E-31	4.80E-14	**0.00**	0
Rastrigin	2	**0.00**	**0.00**	**0.00**	0
Rastrigin	10	8.3	11.6	**0.00**	0
Rosenbrock	2	**9.20E-18**	4.80E-12	7.283e-01	0
Rosenbrock	10	7.3	**3.8**	8.9878	0
Ackley	2	**8.89E-16**	1.40E-11	**8.89E-16**	0
Ackley	10	1.6	2.6	**8.89E-16**	0
Griewank	2	1.30E-12	1.40E-05	**0.00**	0
Griewank	10	1.50E-01	1.40E-01	**0.00**	0
Michalewicz	2	**-1.9**	**-1.9**	-1.2750	-1.8013
Michalewicz	10	-5.8	**-6.1**	-2.4864	-9.66015
Easom	2	**-1**	**-1**	**-1**	-1
Hump	2	**4.70E-08**	**4.70E-08**	3.4496e-01	0
No of good optimal values		7	6	9	

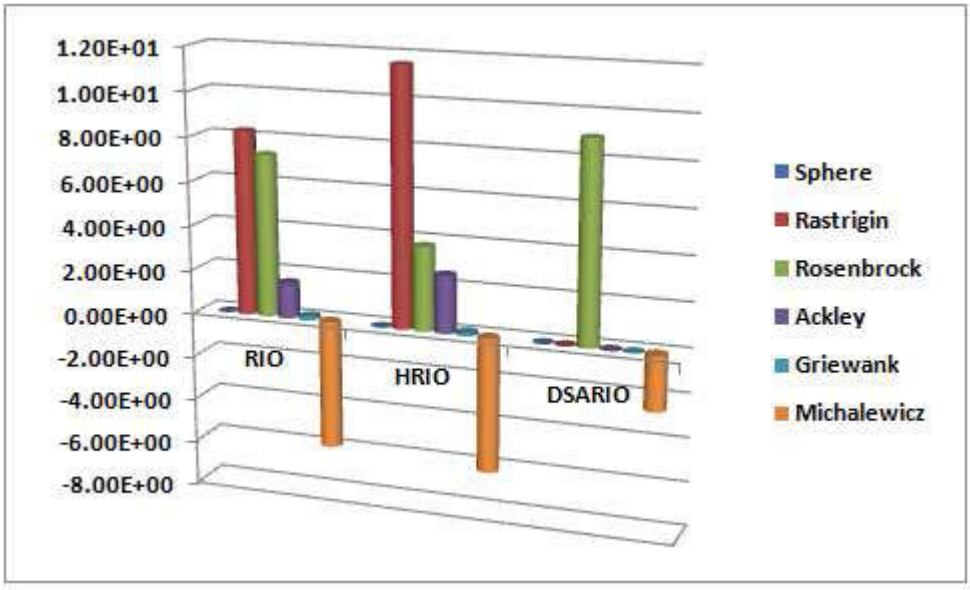

Figure 5.2: Comparison of DSARIO, RIO and HRIO performance for dimensions 10.

Table 5.29: ANOVA test for DSARIO, RIO and HRIO

	Sum of Square	df	Mean Square	F	Sig.
Between Groups	194.643	7	27.806	3.882	.003
Within Groups	243.514	34	7.162		
Total	438.158	41			

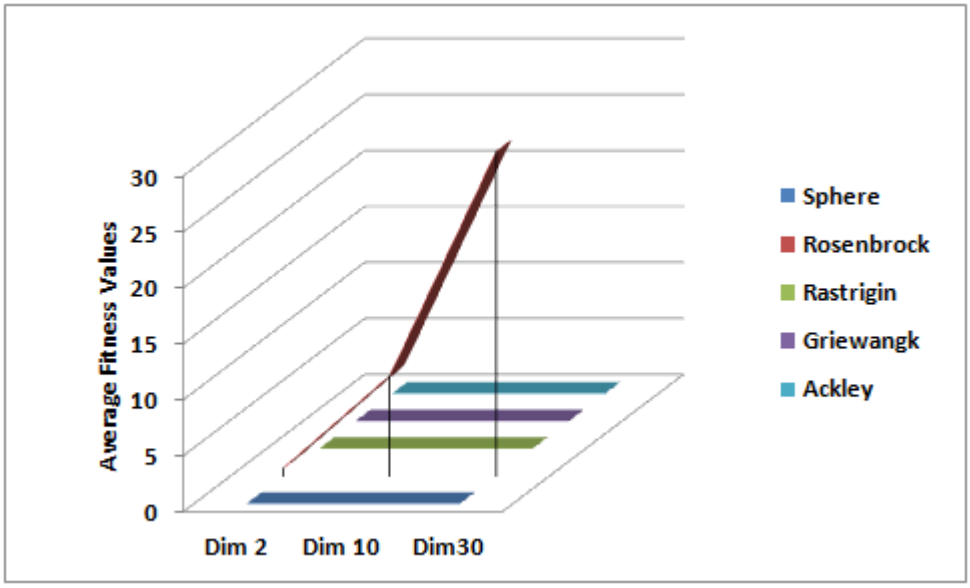

Figure 5.3: Average Performance of DSARIO for Dimensions 2, 10 and 30.

5.5 Empirical Result of Modified Roach Infestation Optimization

We presented the MRIO model in section 3.6.1 and simulation studies were conducted to evaluate and compare the success of the MRIO compared with HRIO and RIO on a set of global optimization problems as shown in Table 18 of Appendix A. Tables 5.30, 5.31 and 5.32 show the result of the experiments. The comparison performance results of MRIO, RIO and HRIO is shown in Table 5.33 and 5.34. Table 5.33 depicts success performance comparison results of MRIO, HRIO and RIO; MRIO are shown to have better success performance. Table 5.34 shows the comparison of average performance of the three algorithm; MRIO is proved to have better average performance [135]. Minimum optimal values are bold in Table 5.34.

The MRIO algorithm show improved performance over RIO and HRIO.

Table 5.30: Simulation Results of MRIO Algorithm. Dimensions 2,10 and 30 for 11 trials at 1000 iterations each.

Function	Dim	Average	STD	Best	Worst	SuccessRate
Sphere	2	1.4612E-199	0.0000	0.0000	1.6073E-198	11/11
	10	0.0000	0.0000	0.0000	0.0000	11/11
	30	0.0000	0.0000	0.0000	0.0000	11/11
Rosenbrock	2	1.9833E-02	3.2967E-02	8.5120E-05	8.6388E-02	5/11
	10	8.7741	2.4048E-01	8.1015	8.9601	0/11
	30	2.8912E01	7.2353E-02	2.8766E01	2.8996E01	0/11
Rastrigin	2	0.0000	0.0000	0.0000	0.0000	11/11
	10	0.0000	0.0000	0.0000	0.0000	11/11
	30	0.0000	0.0000	0.0000	0.0000	11/11
Griewangk	2	0.0000	0.0000	0.0000	0.0000	11/11
	10	0.0000	0.0000	0.0000	0.0000	11/11
	30	0.0000	0.0000	0.0000	0.0000	11/11
Ackley	2	8.8816E-16	0.0000	8.8816E-16	8.8816E-16	11/11
	10	8.8816E-16	0.0000	8.8816E-16	8.8816E-16	11/11
	30	8.8816E-16	0.0000	8.8816E-16	8.8816E-16	11/11
Michalewicz	2	-1.8771	8.0017E-02	-1.9325	-1.6489	11/11
	10	-3.2616	8.1148E-01	-5.1931	-2.2172	11/11
Easom	2	-1	1.9676E-04	-1	-1	11/11
Hump	2	4.4756E-05	4.4756E-05	2.9210E-06	1.9580E-04	11/11

Table 5.31: Simulation Results of HRIO Algorithm. Dimensions 2,10 and 30 for 11 trials at 1000 iterations each.

Function	Dim	Average	STD	Best	Worst	SuccessRate
Sphere	2	0.0000	0.0000	0.0000	0.0000	11/11
	10	0.0000	0.0000	0.0000	0.0000	11/11
	30	0.0000	0.0000	0.0000	0.0000	11/11
Rosenbrock	2	6.9580	1.6948E+01	1.2326E-32	5.5988E+01	7/11
	10	2.3633E+03	4.9607E+03	8.2411	1.5551E+04	0/11
	30	9.9178E+07	7.6510E+07	4.9892E+06	2.8158E+08	0/11
Rastrigin	2	0.0000	0.0000	0.0000	0.0000	11/11
	10	0.0000	0.0000	0.0000	0.0000	11/11
	30	0.0000	0.0000	0.0000	0.0000	11/11
Griewangk	2	4.5286E-03	4.1391E-03	0.0000	9.8647E-03	5/11
	10	1.6952E-01	9.2515E-02	1.9719E-02	3.5871E-01	0/11
	30	1.4431	3.9550E-01	1.0547	2.2690	0/11
Ackley	2	8.8816E-16	0.0000	8.8816E-16	8.8816E-16	11/11
	10	2.0023E+01	5.3492E-02	1.9983E+01	2.0174E+01	0/11
	30	2.0045E+01	6.4953E-02	2.0001E+01	2.0223E+01	0/11
Michalewicz	2	-1.9831	2.2362E-02	-1.9997	-1.9206	11/11
	10	-6.6913	9.5177	-8.1987	-5.6137	11/11
Easom	2	-1	1.3516E-06	-1	-1	11/11
Hump	2	7.4197E-02	2.4608E-01	4.6510E-08	8.1616E-01	10/11

Table 5.32: Simulation Results of RIO Algorithm. Dimensions 2, 10 and 30 for 11 trials at 1000 iterations each.

Function	Dim	Average	STD	Best	Worst	SuccessRate
Sphere	2	0.0000	0.0000	0.0000	0.0000	11/11
	10	0.0000	0.0000	0.0000	0.0000	11/11
	30	0.0000	0.0000	0.0000	0.0000	11/11
Rosenbrock	2	4.1633	1.0129E+01	3.7383E-28	3.2626E+01	7/11
	10	1.4756E+03	4.0834E+03	8.1718	1.3750E+04	0/11
	30	5.7042E+07	5.7710E+07	1.0729E+07	1.8103E+08	0/11
Rastrigin	2	0.0000	0.0000	0.0000	0.0000	11/11
	10	0.0000	0.0000	0.0000	0.0000	11/11
	30	0.0000	0.0000	0.0000	0.0000	11/11
Griewangk	2	5.3789E-03	3.4547E-03	0.0000	7.3960E-03	3/11
	10	1.3294E-01	1.0519E-01	1.0758E-02	3.3992E-01	0/11
	30	1.3334	2.2574E-01	1.0407	1.8623	0/11
Ackley	2	8.8816E-16	0.0000	8.8816E-16	8.8816E-16	11/11
	10	2.0028E+01	3.7007E-02	2.0000E+01	2.010E+01	0/11
	30	2.0172E+01	3.0123E-01	2.003E+01	2.1054E+01	0/11
Michalewicz	2	-1.9932	6.9080E-03	-1.9999	-1.9817	11/11
	10	-6.2521	1.1008	-7.7463	-4.5536	11/11
Easom	2	-1	8.9143E-13	-1	-1	11/11
Hump	2	4.6510E-08	8.9821E-17	4.6510E-08	4.6510E-08	11/11

Table 5.33: Comparison Success Rate.

Test Function	Dim	RIO	HRIO	MRIO
Sphere	2	1	1	1
	10	1	1	1
	30	1	1	1
Rastrigin	2	1	1	1
	10	1	1	1
	30	1	1	1
Rosenbrock	2	0.64	0.64	0.45
	10	0	0	0
	30	0	0	0
Ackley	2	1	1	1
	10	0	0	1
	30	0	0	1
Griewank	2	0.27	0.45	1
	10	0	0	1
	30	0	0	1
Michalewicz	2	1	1	1
	10	1	1	1
Easom	2	1	1	1
Hump	2	1	0.9	1
No of good success rate		11	11	16

Table 5.34: Comparison Average Performance.

Function	Dim	RIO	HRIO	MRIO	Optimum
Sphere	2	**0.00**	**0.00**	1.46E-199	0
	10	**0.00**	**0.00**	**0.00**	0
	30	**0.00**	**0.00**	**0.00**	0
Rastrigin	2	**0.00**	**0.00**	**0.00**	0
	10	**0.00**	**0.00**	**0.00**	0
	30	**0.00**	**0.00**	**0.00**	0
Rosenbrock	2	4.16	6.96	**1.98E-02**	0
	10	1.48E+03	2.36E+03	**8.77**	0
	30	5.70E+07	9.92E+07	**2.902E+01**	0
Ackley	2	**8.88E-16**	**8.88E-16**	**8.88E-16**	0
	10	2.00E+01	2.00E+01	**8.88E-16**	0
	30	2.02E+01	2.00E+01	**8.88E-16**	0
Griewank	2	5.38E-03	4.5286E-03	**0.00**	0
	10	1.33E-01	1.70E-01	**0.00**	0
	30	1.33	1.44	**0.00**	0
Michalewicz	2	**-1.9**	**-1.9**	**-1.9**	-1.8013
	10	-6.3	**-6.7**	-3.3	-9.66015
Easom	2	**-1**	**-1**	**-1**	-1
Hump	2	**4.65E-08**	7.42E-02	4.48E-05	0
No of good optima values		10	10	16	

5.6 ACSO for Global Optimization Problems Using PPEM

Empirical studies were carried out using 70 global optimization benchmark problems presented in Table 19,20 and 21 of Appendix A to test the successes of the ACSO model shown in section 3.7 and compared with RIO, DSARIO and PSO. The obtained results were statistically analysed and compared.

Several versions of PSO exist in the literature, PSO variant [45] was chosen in this work for high performance. Table 5.35 shows the results of simulation studies carried on ACSO, RIO DSARIO and PSO where the optimum values, mean iterations, mean function evaluations, success rate, normalised success performance and computation time were computed and recorded.

Tables 22, 23, 24 and 25 of Appendix B depict the comparison results of ACSO, RIO, DSARIO and PSO algorithms best, average, standard deviation of Mean, success rate respectively. From the comparison results, generally, ACSO performs better when compared with other algorithms.

Success performance of ACSO, RIO, DSARIO and PSO algorithms were tested. The normalised SP (nSP) was computed by dividing algorithm *SP* by the *SP* of the best performing algorithm (SP_{best}) for a particular benchmark function. The results of *nSP* are shown in the Table 5.35 and Table 26 of Appendix B. The "-" in Table 5.35 and Table 26 of Appendix B indicates that a particular algorithm can not solve a problem successfully.

Comparison of ACSO, RIO, DSARIO and PSO algorithms for the statistic parameters: success rate, mean iteration, mean function evaluation and computation time respectively are given in Tables 5.36, 5.37, 5.38 and 5.39. This is to determine the number of functions that have equal values of success rate, mean iteration, mean function evaluation and computation time. In Tables 5.36, 5.37, 5.38 and 5.39, SAME denotes the number of functions that have equal values and DIFF denotes number of functions that have different values.

The ANOVA test statistic was conducted on the average performance of ACSO, RIO, DSARIO and PSO algorithms to determine if there was a significant difference. The P-Value from the ANOVA test is 0.3933, the null hypothesis suggests that there is no significant difference in the average performance of the algorithms. Figure 5.4 illustrates graphically the result of the ANOVA test; Numbers 1-4 on the x-axis denote ACSO, RIO, DSARIO and PSO respectively.

Table 5.35: Simulation results of ACSO, RIO, DSARIO, PSO

FN	Optimum		ACSO	RIO	DSARIO	PSO
F1	0	Ave	2.1587E-13	1.16661E-01	1.4953E-04	1.2597E-01
		STD	1.0765E-12	9.2344E-02	2.2195E-04	7.0443E-02
		Best	0.0000	3.9382E-02	1.1299E-08	9.8573E-03
		SRT	25/25	0/25	25/25	0/25
		Miter	126.36	2000.00	4.16	2000.00
		MFE	7581.60	120000.00	249.60	120000.00
		nSP	30.38	-	1.00	-
		Time	17.9	399.5	0.8	420.6
F2	0	Ave	0.0000	6.1320E+02	0.0000	4.4884E+02
		STD	0.0000	5.2984E+02	0.0000	2.8334E+02
		Best	0.0000	1.5500E+02	0.0000	2.300E+01
		SRT	25/25	0/25	25/25	0/25
		Miter	58.16	2000.00	3.28	20000
		MFE	3489.60	120000.00	196.80	120000.00
		nSP	17.73	-	1.00	-
		Time	8.3	395.8	0.6	471.1
F3	0	Ave	1.6125E-04	2.5735E-04	1.1382E-04	3.3156E-04
		STD	2.8381E-04	2.3894E-04	1.9129E-04	2.8938E-04
		Best	9.0293E-24	3.9581E-07	1.7095E-10	1.5973E-07
		SRT	25/25	25/25	25/25	25/25
		Miter	52.40	4.04	2.28	4.04
		MFE	3144.00	242.40	136.80	244.80
		nSP	22.98	1.77	1.00	1.79
		Time	7.4	0.7	0.5	0.8
F4	0	Ave	1.3373E-15	8.7191E-04	1.0044E-04	8.7330E-04
		STD	5.211E-15	1.1052E-04	1.2398E-04	1.1570E-04
		Best	2.7637E-23	5.5692E-04	1.4065E-07	5.5964E-04
		SRT	25/25	25/25	25/25	25/25
		Miter	75.97	44.40	3.80	45.00
		MFE	4557.60	2664.00	228.00	2700.00
		nSP	19.98	11.68	1.00	11.84
		Time	10.7	8.6	0.7	9.1
F5	-1.143833	Ave	-1.0745	-0.8511	-2.9387	-0.8638
		STD	4.4159E-01	6.7003E-01	9.0750	6.1386E-01
		Best	-1.0011	-1.0276	-1.0360	-1.0151
		SRT	25/25	23/25	25/25	23/25
		Miter	2.16	161.88	2.28	161.84
		MFE	129.60	9712.80	136.80	9710.40
		nSP	1.00	81.46	1.06	81.44
		Time	0.3	22.4	0.5	22.0
F6	-1	Ave	-1	-1	-1	-1
		STD	1.4544E-01	2.6811E-02	3.9507E-02	2.7422E-02
		Best	-1	-1	-1	-1
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.1	0.2

F7	0	Ave	0.0000	0.0000	0.0000	0.0000
		STD	0.0000	0.0000	0.0000	0.0000
		Best	0.0000	0.0000	0.0000	0.0000
		SRT	25/25	25/25	25/25	25/25
		Miter	2.44	2.00	2.00	2.00
		MFE	46.40	120.00	120.00	120.00
		nSP	1.00	2.59	2.59	2.59
		Time	0.6	0.5	0.5	0.5
F8	0	Ave	0.0000	0.0000	0.0000	0.0000
		STD	0.0000	0.0000	0.0000	0.0000
		Best	0.0000	0.0000	0.0000	0.0000
		SRT	25/25	25/25	25/25	25/25
		Miter	2.60	2.00	2.00	2.00
		MFE	156.00	120.00	120.00	120.00
		nSP	1.30	1.00	1.00	1.00
		Time	1.7	1.3	1.4	1.4
F9	-50	Ave	-29.1370	-12.7710	0.2779	-16.3650
		STD	3.4156	8.9229	5.7431	10.1810
		Best	-33.6890	-31.6910	-10.1710	-38.0600
		SRTs	25/25	25/25	9/25	25/25
		Miter	10.64	6.24	1280.72	6.92
		MFE	638.40	374.40	76843.00	415.20
		nSP	1.71	1	570.12	1.11
		Time	0.7	0.7	157.9	0.8
F10	-210	Ave	-36.7220	31.9200	2.6811	0.8737
		STD	29.848	121.460	11.8800	73.0930
		Best	-137.6200	121.4600	-45.6420	-69.0820
		SRT	25/25	20/25	7/25	23/25
		Miter	39.32	432.04	1440.56	185.20
		MFE	2359.20	25922.40	86433.60	11112.00
		nSP	1.00	13.73	130.85	5.12
		Time	5.6	61.9	200.0	27.2
F11	0	Ave	9.9540E-15	9.9726E-03	1.3465E-04	2.8009E-03
		STD	4.9703E-14	4.5325E-02	1.8891E-04	8.3398E-03
		Best	1.8832E-24	5.9397E-04	1.9627E-10	6.8842E-04
		SRT	25/25	24/25	25/25	22/25
		Miter	114.24	210.28	3.56	375.88
		MFE	6854.40	12616.80	213.60	114.24
		nSP	32.09	61.53	1.00	24400.91
		Time	9.0	29.0	0.5	52.4
F12	0	Ave	4.0871E-18	6.5472E-03	1.0354E-04	1.8167E-03
		STD	2.0280E-17	1.9173E-02	12.7292E-04	1.7136E-03
		Best	9.4229E-25	6.1341E-04	1.2968E-07	9.2828E-04
		SRT	25/25	14/25	25/25	16/25
		Miter	121.60	1022.36	3.08	865.84
		MFE	7296.00	61341.60	184.80	51950.40
		nSP	39.48	592.74	1.00	439.24
		Time	16.1	209.5	0.6	197.2

F13	0	Ave	7.3497E-09	4.7632	3.3592E-04	3.3316
		STD	2.4747E-08	4.9643	2.8583E-04	2.0864
		Best	2.9377E-12	5.5889E-01	8.7281E-06	6.4874E-01
		SRT	25/25	0/25	25/25	0/25
		Miter	97.36	2000.00	5.56	2000.00
		MFE	5841.60	120000.00	333.60	120000.00
		nSP	17.51	-	1.00	-
		Time	13.8	460.5	1.1	536.0
F14	-24777	Ave	-10233	-13541	-49992	-13734
		STD	6.9373E02	8.4103E03	7.0350E03	6.5270E03
		Best	-22184	-24691	-19973	-24565
		SRT	25/25	25/25	25/25	25/25
		Miter	5.52	2.16	2.88	2.24
		MFE	331.20	129.60	172.80	134.40
		nSP	2.56	1.00	1.33	1.04
		Time	0.3	0.1	0.3	0.2
F15	0	Ave	5.3366E-14	0.0000	2.1377E-07	2.9688E-05
		STD	2.6683E-13	0.0000	9.9333E-07	1.4844E-04
		Best	0.0000	0.0000	0.0000	0.0000
		SRT	25/25	25/25	25/25	25/25
		Miter	2.60	2.04	2.00	2.08
		MFE	156.00	122.40	120.00	124.80
		nSP	1.30	1.02	1.00	1.04
		Time	0.6	0.87	0.5	0.8
F16	0	Ave	2.4425E-15	4.1192E-04	1.7616E-04	2.8545E-05
		STD	1.1892E-14	2.8876E-04	2.8737E-04	2.8548E-04
		Best	0.0000	3.595E-06	1.6358E-09	2.1835E-05
		SRT	25/25	25/25	25/25	25/25
		Miter	80.68	12.24	3.32	11.68
		MFE	4840.80	734.40	199.20	700.80
		nSP	24.3	3.69	1.00	3.52
		Time	4.1	1.3	0.4	1.2
F17	0	Ave	8.9534E-05	5.2121E-04	9.2077E-05	5.8413E-04
		STD	2.1866E-04	2.8287E-04	1.0144E-04	2.9537E-04
		Best	1.4699E-27	3.6530E-05	2.4759E-07	4.1298E-05
		SRT	25/25	25/25	25/25	25/25
		Miter	82.56	6.04	2.04	5.80
		MFE	4953.60	362.40	122.40	348.00
		nSP	40.47	2.96	1.00	2.84
		Time	3.9	0.6	0.2	0.6
F18	0	Ave	1.6968E-04	2.5530E-04	1.0955E-04	2.2522E-04
		STD	2.8640E-04	2.9507E-04	1.6544E-04	2.9907E-04
		Best	0.0000	6.7898E-07	3.6719E-09	6.2938E-09
		SRT	25/25	25/25	25/25	25/25
		Miter	37.08	4.16	2.24	4.56
		MFE	2224.80	249.60	134.40	273.60
		nSP	16.55	1.86	1.00	2.04
		Time	5.5	0.9	0.4	0.8

F19	-9.66,152	Ave	-6.6374	-9.5000	-9.0058	-9.6277
		STD	3.0730	3.9689E-01	8.2091E-01	3.4992E-01
		Best	-9.8456	-10.0000	-9.9811	-10.0000
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.3	0.3	0.2
F20	0	Ave	9.1595E-05	2.3375E-04	1.3138E-04	2.5793E-04
		STD	2.1050E-04	2.8747E-04	2.5360E-04	3.0304E-04
		Best	4.3189E-26	4.3762E-08	1.1521E-10	1.8292E-08
		SRT	25/25	25/25	25/25	25/25
		Miter	52.84	4.44	2.44	4.32
		MFE	3170.40	266.40	146.40	259.20
		nSP	21.66	1.82	1.00	1.77
		Time	7.4	0.8	0.4	0.8
F21	0	Ave	1.1460E-04	1.3617E-04	4.2881E-05	1.9027E-04
		STD	2.2402E-04	2.0020E-04	1.4461E-04	2.8286E-04
		Best	1.3520E-41	2.6833E-09	1.9864E-15	6.9725E-13
		SRT	25/25	25/25	25/25	25/25
		Miter	14.56	2.96	2.04	2.96
		MFE	873.60	117.60	122.40	117.60
		nSP	7.43	1.00	1.04	1.00
		Time	2.1	0.5	0.4	0.5
F22	-1	Ave	-1	-1.8842	-2.2495	-2.0333
		STD	8.6883E-01	5.2018E-01	7.4135E-01	5.9152E-01
		Best	-2.6481	-2.7553	22.7553	-2.7499
		SRT	25/25	25/25	25/25	25/25
		Miter	2.04	2.00	2.00	2.00
		MFE	122.00	124.00	120.00	120.00
		nSP	1.01	1.03	1.00	1.00
		Time	0.1	0.1	0.2	0.1
F23	-1.03163	Ave	-4.6410E-01	-7.1969E-01	-3.7566E-01	-6.3740E-01
		STD	3.0142	2.621E-01	2.5677E-01	2.7416E-01
		Best	-9.9277E-01	-1.0316	-9.5735E-01	-1.0090
		SRT	25/25	25/25	25/25	25/25
		Miter	4.84	2.04	2.00	2.00
		MFE	290.40	122.40	120.00	120.00
		nSP	2.42	1.02	1.00	1.00
		Time	0.3	0.2	0.2	0.2
F24	0	Ave	1.4633E-15	9.1175E-03	1.3343E-04	3.6042E-04
		STD	5.0091E-15	4.3583E-02	2.0221E-04	2.1900E-04
		Best	0.0000	1.6630E-05	4.7145E-09	5.8541E-05
		SRT	25/25	24/25	25/25	25/25
		Miter	126.28	92.44	3.48	11.88
		MFE	7576.80	5546.40	208.80	712.80
		nSP	36.29	27.67	1.00	3.41
		Time	6.0	10.1	0.6	1.2

F25	0	Ave	9.6589E-16	4.9737E-04	1.4538E-04	4.4007E-04
		STD	2.9988E-15	2.9271E-04	1.9047E-04	2.2547E-04
		Best	0.0000	2.8642E-08	7.329E-09	3.3781E-05
		SRT	25/25	25/25	25/25	25/25
		Miter	99.32	11.64	3.24	12.20
		MFE	5959.20	698.40	194.40	732.00
		nSP	30.65	3.59	1.00	3.76
		Time	4.7	1.3	0.4	1.2
F26	-186.73	Ave	-5.8022	-7.2260	-6.5406	-7.2458
		STD	2.3471	8.8542E-01	1.0986	7.2980E-01
		Best	-8.2562	-8.2477	-8.1464	-8.2176
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.1	0.1	0.1	0.1
F27	0.0003	Ave	0.0003	0.00014	0.00067	0.000098
		STD	3.1749E-04	1.9548E-04	1.4103E-03	1.8463E-04
		Best	3.7517E-06	2.1093E-08	6.2377E-08	1.3628E-09
		SRT	25/25	25/25	22/25	25/25
		Miter	4.04	2.00	280.88	2.00
		MFE	242.40	120.00	16852.80	120.00
		nSP	2.02	1.00	159.59	1.00
		Time	0.3	0.2	33.8	0.1
F28	-10.15	Ave	-2.1673E-01	-1.0107	-2.4993E-01	-6.0081E-01
		STD	3.3822E-01	9.3313E-01	8.9364E-02	4.4564E-01
		Best	-1.4660	-3.7750	-4.5643E-01	-1.9817
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2
F29	-10.4	Ave	-2.4531E-01	-6.8191E-01	-2.8590E-01	-9.2466E-01
		STD	1.9151E-01	4.9562E-01	1.7102E-01	8.7578E-01
		Best	-7.8120E-01	-2.7559	-12.0036	-3.5417
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2
F30	-10.53	Ave	-2.0936E-01	-7.4732E-01	-4.0651E-01	-8.8047E-01
		STD	1.6936E-01	7.0590E-01	2.834E-01	5.8296E-01
		Best	-5.2573E-01	-3.7600	-1.3069	-2.2810
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2

F31	-3.86	Ave	-2.1273	-3.7400	-3.5107	-3.7105
		STD	1.5484	7.2763E-02	1.8706E-01	9.9101E-02
		Best	-3.8226	-3.8550	-3.8076	-3.8602
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.3	0.2	0.3
F32	-3.32	Ave	-5.8408E-01	-2.2863	-1.7483	-2.4543
		STD	9.4195E-01	4.1581E-01	4.4954E-01	3.2249E-01
		Best	-3.1067	-3.0826	-2.4693	-2.9338
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00.00	1.00	1.00	1.00
		Time	0.3	0.3	0.3	0.2
F33	0	Ave	-3.2120E01	-7.0000	7.6400	-8.6400
		STD	1.9654E01	6.4872	2.1579	6.7631
		Best	-6.1000E01	-2.5000E01	5.0000	-2.2000E01
		SRT	25/25	25/25	0/25	25/25
		Miter	4.96	2.08	2000.00	2.24
		MFE	297.60	124.80	120000.00	134.40
		nSP	2.38	1.00	-	1.08
		Time	0.4	0.2	242.2	0.2
F34	0	Ave	1.2732E-08	7.3086E-01	2.8468E-04	1.3644
		STD	5.8327E-08	8.7925E-01	2.8049E-04	1.0682
		Best	1.9321E-11	5.353E-04	8.4601E-06	5.8690E-04
		SRT	25/25	13/25	25/25	7/25
		Miter	115.96	997.32	5.28	1459.76
		MFE	6957.60	59839.20	316.80	87585.60
		nSP	21.96	363.24	1.00	981.39
		Time	16.7	200.6	1.0	299.0
F35	-1.08	Ave	-6.4271E-17	-1.2332E-16	-8.6860E-17	-1.2918E-10
		STD	6.0609E-17	2.3589E-17	4.6158E-17	-1.5819E-17
		Best	-1.4090E-16	-1.4094E-16	-1.4098E-16	-1.4098E-16
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2
F36	-1.5	Ave	-5.8388E-17	-1.2842E-16	-9.6111E-17	-1.3049E-16
		STD	5.2628E-17	1.9809E-17	4.4372E-17	1.7567E-17
		Best	-1.4060E-16	-1.4098E-16	-1.4094E-16	-1.4099E-16
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2

F37	NA	Ave	-4.6596E-17	-1.3516E-16	-8.8841E-17	-1.2696E-16
		STD	4.9743E-17	9.3102E-18	3.9599E-17	2.5882E-17
		Best	-1.3347E-16	-1.4099E-16	-1.4099E-16	-1.4098E-16
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2
F38	-10.4056	Ave	-5.1432E-01	-1.0019	-8.9258E-01	-1.0250
		STD	3.3458E-01	1.5669E-01	1.4846E-01	1.9542E-01
		Best	-9.8307E-01	-1.3094	-1.4688	-1.5441
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2
F39	-10.2088	Ave	-2.1065E-01	-3.2412E-01	-2.9201E-01	-3.1824E-01
		STD	1.0759E-01	1.5440E-02	1.7900E-02	1.3756E-02
		Best	-3.4013E-01	-3.5228E-01	-3.4584E-01	-3.4372E-01
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2
F40	-0.3523	Ave	-1.2487E-01	-1.4533E-01	-1.1179E-01	-2.1600E-01
		STD	1.0522E-012	1.1236E-01	9.5984E-02	1.0183E-01
		Best	-3.5211E-01	-3.4109E-01	-3.1769E-01	-3.4459E-01
		SRT	25/25	25/25	25/25	25/25
		Miter	5.60	2.16	2.00	2.20
		MFE	336.00	129.60	120.00	132.00
		nSP	2.80	1.08	1.00	1.10
		Time	0.3	0.1	0.2	0.2
F41	0	Ave	7.6633	4.5971E-04	3.9890E-01	4.5304E-04
		STD	3.8209E+01	3.2123E-04	3.7194E-01	2.9985E-04
		Best	1.70027E-06	2.3286E-06	3.1320E-02	3.6012E-05
		SRT	23/25	25/25	0/25	25/25
		Miter	1006.56	6.04	2000.00	6.60
		MFE	60393.60	362.40	120000.00	396.00
		nSP	181.14	1	-	1.09
		Time	47.4	0.6	219.6	0.7
F42	0	Ave	7.9618E-05	5.8447E-04	1.5175E-04	3.9017E-04
		STD	2.3721E-04	2.8022E-04	1.7251E-04	3.1285E-04
		Best	7.5833E-25	6.2857E-05	3.9838E-08	1.8839E-05
		SRT	25/25	25/25	25/25	25/25
		Miter	96.08	5.24	2.16	8.40
		MFE	5764.80	314.40	129.60	504.00
		nSP	44.48	2.43	1.00	3.89
		Time	4.7	0.6	0.2	0.9

F43	0	Ave	4.0446E+09	1.4550E+01	8.9798	3.1514E+01
		STD	7.1440E+09	2.2233E+01	2.1289E-02	5.1801E+01
		Best	1.0769E+07	3.7859E-01	8.9087	8.2952E-02
		SRT	0/25	0/25	0/25	0/25
		Miter	2000.00	2000.00	2000.00	2000.00
		MFE	120000.00	120000.00	120000.00	120000.00
		nSP	-	-	-	-
		Time	283.1	454.1	447.0	507.5
F44	0	Ave	1.7646E+04	5.8276E-04	3.7712E-01	5.3484E-04
		STD	5.8577E+04	2.9033E-04	2.4840E-01	3.2449E-04
		Best	1.4162E-05	7.5074E-05	4.6735E-03	1.8000E-06
		SRT	12/25	25/25	0/25	25/25
		Miter	1485.80	12.75	2000.00	12.92
		MFE	89148.00	765.60	120000.00	775.20
		nSP	132.32	1.00	-	1.01
		Time	207.5	2.4	434.5	2.7
F45	0.4E-04	Ave	8.4621E-05	9.7778E-05	2.8271E-03	2.2392E-04
		STD	1.6341E-04	1.5542E-04	5.7177E-03	2.5941E-04
		Best	7.2454E-10	4.0475E-07	3.8019E-06	1.8895E-07
		SRT	25/25	25/25	14/25	25/25
		Miter	3.28	2.16	881.12	2.08
		MFE	196.80	129.60	52867.20	124.80
		nSP	1.58	1.04	756.45	1.00
		Time	0.4	0.3	107.2	0.2
F46	0	Ave	2.7967	9.7615E-04	1.2896E-01	9.8618E-04
		STD	1.0426E+01	3.9289E-05	1.030E-01	3.0972E-05
		Best	1.0604E-04	8.6098E-04	1.4066E-02	8.4200E-04
		SRT	17/25	25/25	0/25	25/25
		Miter	1115.64	80.88	2000.00	102.84
		MFE	66938.40	4552.80	12000.00	6170.40
		nSP	21.62	1.00	-	1.36
		Time	65.1	9.5	233.3	11.6
F47	1.29695	Ave	2.6646E-09	1.7253E-34	1.9672E-16	1.2699E-34
		STD	8.2204E-09	1.7253E-34	1.9672E-16	1.2699E-34
		Best	7.2141E-243	0.0000	4.5782E-22	0.0000
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.1	0.1
F48	0	Ave	4.9627E06	3.9447	8.1848E01	3.9447
		STD	1.0891E07	6.7093E-15	1.0394E+02	8.1018E-15
		Best	1.1781E01	3.9447	4.5521	3.9447
		SRT	0/25	0/25	0/25	0/25
		Miter	2000.00	2000.00	2000.00	2000.00
		MFE	120000.00	120000.00	120000.00	120000.00
		nSP	-	-	-	-
		Time	124.9	236.6	244.9	232.7

F49	-210	Ave	-2.4102E04	-1.0015E03	-5.0677E02	-1.0086E03
		STD	2.6887E04	2.613E02	2.3337E02	3.0581E02
		Best	-7.9172E04	-7.7673E03	-1.0906E03	01.7252E03
		SRT	25/25	25/25	25/25	25/25
		Miter	2.56	2.00	2.00	2.00
		MFE	153.60	120.00	120.00	120.00
		nSP	1.28	1.00	1.00	1.00
		Time	0.4	0.2	0.3	0.2
F50	-45.778	Ave	-4.5881E22	-2.7607E13	-2.2138E17	-3.4309E13
		STD	2.2197E23	3.1804E13	1.7242E17	5.6333E13
		Best	-1.9141E01	-5.6207E11	-7.6812E17	-1.6121E11
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.4	0.5	0.4	0.3
F51	0.9	Ave	1.1554	0.9	0.9	0.9
		STD	4.2083E-01	4.8190E-02	3.3993E-16	4.5826E-02
		Best	0.9	0.9	0.9	0.9
		SRT	0/25	0/25	0/25	0/25
		Miter	2000.00	2000.00	2000.00	2000.00
		MFE	120000.00	120000.00	120000.00	120000.00
		nSP	-	-	-	-
		Time	95.7	212.6	221.1	204.2
F52	0	Ave	1.4758E-09	1.0387E-01	2.3258E-04	1.0789E-01
		STD	6.614E-09	2.0000E-02	2.1929E-04	2.7891E-02
		Best	1.0991E-14	9.9873E-02	1.1205E-05	9.9873E-02
		SRT	25/25	0/25	25/25	0/25
		Miter	99.96	2000.00	4.68	2000.00
		MFE	5997.60	12000.00	280.80	120000.00
		nSP	21.36	-	1.00	-
		Time	6.0	235.4	0.6	225.2
F53	0	Ave	6.6572E-10	2.5987E-01	2.751E-04	2.5984E-01
		STD	1.0155E-09	9.5143E-02	2.8179E-04	1.1902E-01
		Best	9.8233E-13	9.9873E-02	1.0445E-05	9.9873E-02
		SRT	25/25	0/25	25/25	0/25
		Miter	100.44	2000.00	5.56	2000.00
		MFE	6026.40	120000.00	333.60	120000.00
		nSP	18.06	-	1.00	-
		Time	7.7	268.2	0.8	261.4
F54	0	Ave	5.2004E-05	1.8339E-03	4.8227E-04	1.8059E-03
		STD	8.2843E-05	1.4509E-03	1.2317E-04	1.4764E-03
		Best	7.6923E-07	4.1395E-04	4.1395E-04	4.1625E-04
		SRT	25/25	15/25	25/25	15/25
		Miter	86.84	928.24	5.56	866.44
		MFE	5210.40	55694.40	333.60	51986.40
		nSP	15.62	278.25	1.00	259.72
		Time	4.2	101.4	0.7	89.8

F55	-3.5	Ave	-2.9110E-01	-3.7784E-01	-3.6724E-01	-3.6488E-01
		STD	1.7196E-01	1.2494E-01	2.2953E-01	2.0094E-01
		Best	-9.499E-01	1.2494E-01	-1.1933	-8.0514E-01
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2
F56	-3.5	Ave	-6.3524E-03	-9.6019E-03	-1.4117E-01	-1.1676E-02
		STD	6.4343E-03	9.7767E-03	6.1956E-01	1.1532E-02
		Best	-2.6155E-02	-4.1685E-02	-3.1086	-4.3874E-02
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2
F57	-2.3458	Ave	-1.9594E02	-2.8143E01	-2.0993	-3.7641E03
		STD	6.7986E02	5.8096E01	1.8136E-01	1.0042E04
		Best	-3.1282E03	-2.2252E02	-2.328	-3.8513E04
		SRT	25/25	25/25	25/25	25/25
		Miter	2.24	2.00	2.00	2.00
		MFE	134.40	120.00	120.00	120.00
		nSP	1.12	1.00	1.00	1.00
		Time	0.1	0.1	0.1	0.1
F58	-1.9133	Ave	-3.8574	-1.8162	-1.4894	-1.8169
		STD	2.6835	4.6137E-01	3.2371E-01	1.3196
		Best	-8.1589	-3.3818	-1.9022	-2.0481
		SRT	25/25	25/25	25/25	25/25
		Miter	2.52	2.00	2.00	2.00
		MFE	151.20	120.00	120.00	120.00
		nSP	1.26	1.00	1.00	1.00
		Time	0.2	0.1	0.1	0.1
F59	0	Ave	9.7618E04	4.4141E-03	3.8105E-01	3.4730E-03
		STD	1.8022E05	3.4632E-03	2.4691E-01	3.5715E-03
		Best	2.8211E-04	7.2138E-06	2.1574E-03	3.8011E-05
		SRT	1/25	11/25	0/25	14/25
		Miter	1963.80	1147.68	2000.00	896.80
		MFE	117828.00	68860.80	120000.00	53808.00
		nSP	30.66	1.63	-	1.00
		Time	93.1	112.6	221.5	91.2
F60	0	Ave	2.6000E04	5.0687E-03	6.4212E01	1.3925E-03
		STD	4.3273E04	2.1839E-02	2.6652E01	3.4963E-03
		Best	1.3204E02	9.1670E-05	6.9628	2.3656E-04
		SRT	0/25	24/25	0/25	24/25
		Miter	2000.00	566.44	2000.00	713.64
		MFE	120000.00	33986.40	120000.00	42818.40
		nSP	-	1.00	-	1.26
		Time	187.3	63.5	228.6	75.0

F61	0.2	Ave	-5.8459	-1.8680E01	-1.7583	-1.8769E01
		STD	4.9619	2.1696E01	1.9731E-01	2.1904E01
		Best	-2.6427E01	-1.0574E02	-2.0569	-1.1730E02
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.1	0.1	0.1	0.2
F62	0.4	Ave	-2.1117E01	-2.0972E01	-2.8178	-1.5381E01
		STD	1.8927E01	1.7100E01	3.0635E-01	7.4261
		Best	-5.4858E01	-9.2125E01	-3.4265	-3.9841E01
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.2	0.2	0.2	0.2
F63	0	Ave	2.8826E-04	1.174E-04	6.1723E-01	1.0401E-04
		STD	3.0041E-04	2.6589E-04	8.5085E-03	1.9838E-04
		Best	2.0836E-22	3.8070E-46	6.0731E-01	1.1217E-69
		SRT	25/25	25/25	0/25	25/25
		Miter	6.96	2.62	2000.00	2.76
		MFE	417.60	158.40	120000.00	165.60
		nSP	2.52	0.96	-	1.00
		Time	0.8	0.3	283.1	0.3
F64	0	Ave	2.0765E+05	5.3430E-04	1.3750	4.8829-04
		STD	6.4456+05	2.9125E-04	1.2711	3.3238E-04
		Best	7.9563E-05	3.5228E-05	1.3170E-02	8.1844E-06
		SRT	17/25	25/25	0/25	25/25
		Miter	1245.72	10.00	2000.00	16.64
		MFE	74743.20	600.00	120000.00	998.40
		nSP	183.19	1.00	-	1.67
		Time	74.8	1.1	239.9	1.9
F65	-4.687658	Ave	0.0000	0.0000	0.0000	0.0000
		STD	0.0000	0.0000	0.0000	0.0000
		Best	0.0000	0.0000	0.0000	0.0000
		SRT	25/25	25/25	25/25	25/25
		Miter	2.00	2.00	2.00	2.00
		MFE	120.00	120.00	120.00	120.00
		nSP	1.00	1.00	1.00	1.00
		Time	0.1	0.2	0.2	0.2
F66	0.998	Ave	4.6078E+02	0.998	4.3453E+02	0.998
		STD	1.2878E+02	2.2662E-16	1.5088E+02	2.2662E-16
		Best	1.9575E+01	0.998	1.2550	0.998
		SRT	0/25	0/25	0/25	0/25
		Miter	2000.00	2000.00	2000.00	2000.00
		MFE	120000.00	120000.00	120000.00	120000.00
		SP	-	-	-	-
		Time	142.6	261.9	261.9	256.5

F67	3	Ave	3.3689E+03	3.0000	2.9140E+01	4.0800
		STD	9.3311E03	7.0799E-10	2.7704E01	5.4000
		Best	3.6434	3.0000	3.1334	3.0000
		SRT	0/25	0/25	0/25	0/25
		Miter	2000.00	2000.00	2000.00	2000.00
		MFE	120000.00	120000.00	120000.00	120000.00
		SP	-	-	-	-
		Time	98.9	216.7	223.3	208.1
F68	0	Ave	2.8324E01	5.3490E-04	3.6745E-01	5.2574E-04
		STD	4.4115E+01	2.4954E-04	2.2470E-01	2.4292E-04
		Best	8.5823E-04	3.0519E-05	1.4519E-02	1.1017E-04
		SRT	1/25	25/25	0/25	25/25
		Miter	1942.20	8.72	2000.00	9.12
		MFE	116532.00	523.20	120000.00	547.20
		nSP	4994.83	1.00	-	1.05
		Time	205.0	1.1	248.0	1.2
F69	0	Ave	3.4994E+02	4.8456E-04	7.3302	4.0821E-04
		STD	9.7496E+02	2.9911E-04	6.8336	2.9635E-04
		Best	4.3013E-04	3.2116E-05	3.6923E-03	4.1482E-05
		SRT	3/25	25/25	0/25	25/25
		Miter	1934.48	7.72	2000.00	7.40
		MFE	116068.80	463.20	120000.00	444.00
		nSP	2178.46	1.04	-	1.00
		Time	89.0	0.8	212.8	0.8
F70	0	Ave	-1.4110E+13	-3.3199E+03	-9.6055E+03	-8.6682E+04
		STD	5.1521E+13	1.2363E+04	1.7384E+04	4.3008E+05
		Best	-2.5272E+14	-6.2056E+04	-6.3024E+04	-2.1511E+06
		SRT	25/25	25/25	25/25	25/25
		Miter	77.36	6.60	2.00	11.76
		MFE	4641.60	396.00	120.00	705.60
		nSP	38.68	3.30	1.00	5.88
		Time	4.4	0.7	0.2	1.2

Ave denotes average(mean optima); STD denotes standard deviation of mean optima; SRT denotes success rate; Best denotes best optima; Miter denotes mean iteration. MFE denotes mean function evaluation; Time denotes computation time in seconds; nSP denotes normalized success performance; "-" for nSP indicates that the algorithm can not solve the problem successfully.

Table 5.36: Comparison of ACSO, RIO, DSARIO, PSO for success Rate.

FN	ACSO	RIO	DSARIO	PSO
SAME	57 (for the functions of: F1,F2,F3,F4,F5 F6,F7,F8,F9,F10 F11,F12,F13,F14,F15 F16,F17,F18,F19,F20 F21,F22,F23,F24,F25 F26,F27,F28,F29,F30 F31,F23,F33,F34,F35 F36,F37,F38,F39,F40 F42,F45,F47,F49,F50 F52,F53,F54,F55,F56 F57,F58,F61,F62,F63 F65,F70)	51 (for the functions of: F3,F4,F5,F6,F8 F9,F11,F14,F15,F16 F17,F18,F19,F20,F21 F22,F23,F25,F26,F27 F28,F29,F30,F31,F32 F33,F35,F36,F37,F38 F39,F40,F41,F42,F44 F45,F46,F47,F49,F50 F55,F56,F57,F58,F61 F62,F63,F64, F68,F69 F70)	51 (for the functions of: F1,F2,F3,F4,F5 F6,F7,F8,F11,F12 F13,F14,F15,F16,F17 F18,F19,F20,F21,F22 F23,F24,F25,F26,F28 F29,F30,F31,F32,F34 F35,F36,F37,F38,F39 F40,F42,F47, F49, F50 F52,F53,F54,F55,F56 F57,F58,F61,F62,F65 F70)	52 (for the functions of: F3,F4,F6,F7,F8 F9,F14,F15,F16,F17 F18,F19,F20,F21,F22 F23,F24,F25,F26,F27 F28,F29,F30,F31,F32 F33,F35,F36,F37,F38 F39,F40,F41,F42,F44 F45,F46,F47,F49,F50 F55,F56,F57,F58,F61 F62,F63,F64,F65,F68 F69,F70)
DIFF	13 (for the functions of: F41,F43,F44,F46,F48 F51,F59,F60,F64,F66 F67,F68,F69)	19 (for the functions of: F1,F2,F5,F10,F11 F12,F13,F24,F34,F43 F48,F51,F52,F53,F54 F59,F60,F66,F67)	19 (for the functions of: F9,F10,F27,F33,F41 F43,F44,F45,F46,F48 F51,F59,F60, F63,F64 F66,F67 ,F68,F69	18 (for the functions of: F1,F2,F5,F10,F11 F12,F13,F34,F43,F48 F50,F51,F52,F53,F59 F60,F66,F67

SAME denotes the number of functions that have equal values of success rate for ACSO, RIO, DSARIO, PSO algorithms, DIFF denotes number of functions that have different values of success rate for ACSO, RIO, DSARIO, PSO algorithms.

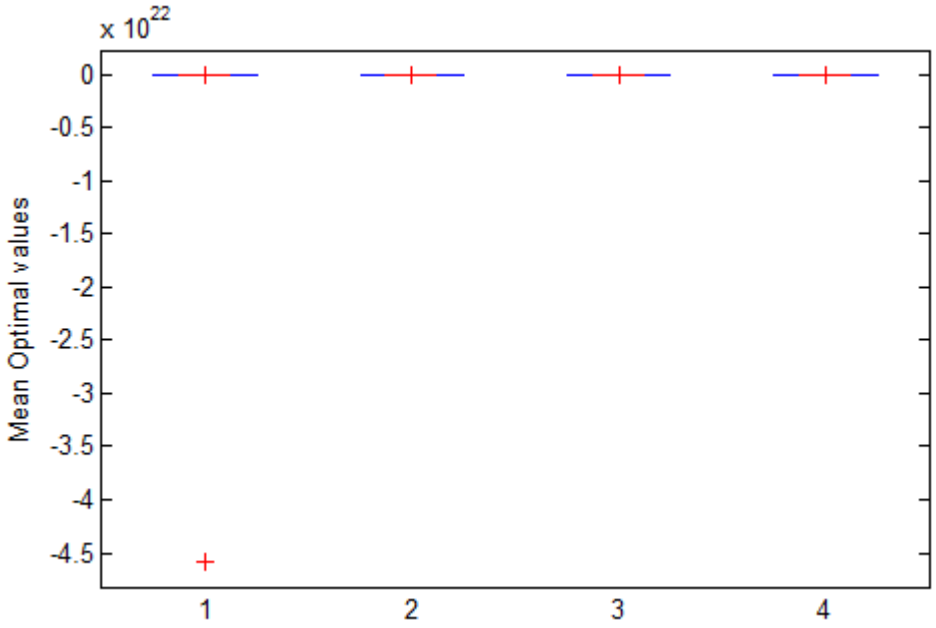

Figure 5.4: ANOVA test for ACSO,RIO,DSARIO and PSO algorithms.

Table 5.37: Comparison of ACSO, RIO, DSARIO, PSO for Mean Iteration.

FN	ACSO	RIO	DSARIO	PSO
SAME	20 (for the functions of: F6,F19,F26,F28,F29 F30,F31,F32,F35,F36 F37,F38,F39,F47,F50 F55,F56,F61,F62,F65)	27 (for the functions of: F6,F7,F8,F19,F22 F26,F27,F28,F29,F30 F31,F32,F35,F36,F37 F38, F39,F47,F49,F50 F55 F56,F57,F58,F61 F62,F65)	30 (for the functions of: F6,F7,F8,F15,F19 F22,F23,F26,F28,F29 F30,F31,F32,F35,F36 F37,F38,F39,F40,F47 F49,F50,F55,F56,F57 F58,F61,F62,F65,F70	28 (for the functions of: F6,F7,F8,F19,F22 F23,F26,F27,F28,F29 F30,F31,F32,F35,F36 F37,F38,F39,F47,F49 F50,F55,F56,F57,F58 F61,F62,F65
	6 (for the functions of: F43,F48,F51,F60,F66 F67)	10 (for the functions of: F1,F2,F13,F43,F48 F51,F52,F53,F66,F67)	15 (for the functions of: F33,F41,F43,F44,F46 F48,F51,F59,F60,F63 F64,F66,F67,F68,F69	10 (for the functions of: F1,F2,F13,F43,F48 F51,F52,F3,F66,F67
	2(for the functions of: F8,F15)	2 (for the functions of: F15,F23) 2 (for the functions of: F17,F41)	2 (for the functions of: F3,F5) 2 (for the functions of: F17,F21)	2 (for the functions of: F14,F34)
DIFF	42 (for the functions of: F1,F2,F3,F4,F5 F7,F9,F10,F11,F12 F13,F14,F16,F17,F18 F20,F21,F22,F23,F24 F25,F27,F33,F34,F40 F41,F42,F44,F45,F46 F49,F52,F53,F54, F57 F58,F59,F63,F64,F68 F69,F70)	29 (for the functions of: F3,F4,F5,F9,F10 F11,F12,F14,F16,F18 F20,F21,F24,F25,F33 F34,F40,F42,F44,F45 F46,F54,F59,F60,F63 F64,F68,F69,F70)	21 (for the functions of: F1,F2,F4,F9,F10 F11,F112,F13,F14,F16 F18,F20,F24,F25,F27 F34,F42,F45,F52,F53 F54)	30 (for the functions of: F3,F4,F5,F9,F10 F11,F12,F15,F16, F17 F18,F20,F21,F24,F25 F34,F40,F41,F42,F44 F45,F46,F54,F59,F60 F63,F64,F68,F69,F70

SAME denotes the number of functions that have equal values of mean iteration for ACSO, RIO, DSARIO, PSO algorithms, DIFF denotes number of functions that have different values of mean iteration for ACSO, RIO, DSARIO, PSO algorithms.

Table 5.38: Comparison of ACSO, RIO, DSARIO, PSO for Mean Function Evaluation.

FN	ACSO	RIO	DSARIO	PSO
SAME	20 (for the functions of: F6,F19,F26,F28,F29 F30,F31,F32,F35,F36 F37,F38,F39,F47,F50 F55,F56,F61,F62,F65)	26 (for the functions of: F6,F7,F8,F19,F26 F27,F28,F29,F30,F31 F32,F35,F36,F37,F38 F39,F47,F49,F50,F55 F56,F57,F58,F61,F62 F65)	30 (for the functions of: F6,F7,F8,F15,F19 F22,F23,F26,F28,F29 F30,F31,F32,F35,F36 F37,F38,F39,F40,F47 F49,F50,F55,F56,F57 F58,F61,F62,F65,F70	28 (for the functions of: F6,F7,F8,F19, F22 F23, F26,F27,F28,F29 F30,F31,F32,F35,F36 F37,F38,F39,F47,F49 F50,F55,F56,F57,F58 F61,F62,F65
	6 (for the functions of: F43,F48,F51,F60,F66 F7)	10 (for the functions of: F1,F2,F13,F43,F48 F51,F52,F53,F66,F67	14 (for the functions of: F33,F41,F43,F44,F46 F48,F51,F60,F63,F64 F66,F67,F68,F69)	10 (for the functions of: F1,F2,F13,F43,F48 F51,F52,F53,F66,F67
	2 (for the functions of: F8,F15)	3 (for the functions of: F14,F40,F45)	2 (for the functions of: F3,F5)	2 (for the functions of: F14,F33)
		2 (for the functions of: F17,F41)	3 (for the functions of: F53,F45, F13)	2 (for the functions of: F15,F45)
		2 (for the functions of: F15,F23)		
DIFF	42 (for the functions of: F1,F2,F3,F4,F5 F7,F9,F10,F11,F12 F13,F14,F16,F17,F18 F20,F21,F22,F23,F24 F25,F27,F33,F34,F40 F41,F42,F44,F45,F46 F49,F52,F53,F54,F57 F58,F59,F63,F64,F68 F69,F70)	27 (for the functions of: F3,F4,F5,F9,F10 F11,F12,F16,F18,F20 F21,F22,F24,F25,F33 F34,F42,F44,F46,F54 F59,F60,F63,F64,F68 F69,F70)	21 (for the functions of: F1,F2,F4,F9,F10 F11,F12,F14,F16,F17 F18,F20,F21,F24,F25 F27,F34,F42,F45,F52 F59)	28 (for the functions of: F3,F4,F56,F9,F10 F11,F12,F16,F17,F18 F20,F21,F24,F25,F34 F40,F41,F42,F44,F46 F54,F59,F60,F63,F64 F68,F69,F70)

SAME denotes the number of functions that have equal values of mean function evaluation for ACSO, RIO, DSARIO, PSO algorithms, DIFF denotes number of functions that have different values of mean function evaluation for ACSO, RIO, DSARIO, PSO algorithms.

Table 5.39: Comparison of ACSO, RIO, DSARIO, PSO for computation time.

FN	ACSO	RIO	DSARIO	PSO
SAME	16 (for the functions of: F6,F19,F28,F29,F30 F31,F35,F36,F37,F38 F39,F47,F55,F56,F58 F62)	18 (for the functions of: F6,F23,F27,F28,F29 F30,F33,F35,F36,F37 F38,F39,F47,F49,F55 F56,F62,F65)	19 (for the functions of: F17,F22,F23,F28,F29 F30,F31,F35,F36,F37 F38,F39,F40,F42,F55 F56,F62,F65,F70)	22 (for the functions of: F6,F14,F19,F23,F28 F28,F30,F32,F33,F35 F36,F37,F38,F39,F40 F45,F49,F55,F56,F61 F62,F65)
	5 (for the functions of: F22,F26,F57,F61,F65)	7 (for the functions of: F14,F22,F26,F40,F57 F58,F61)	6(for the functions of: F6,F26,F47,F57,F58 F61)	6(for the functions of: F22,F26,F27,F47,F57 F58)
	6(for the functions of: F5,F14,F23,F27,F32 F40)	5 (for the functions of: F19,F31,F32,F45,F63	7(for the functions of: F16,F18,F19,F20,F21 F25,F50)	6(for the functions of: F3,F9,F15,F18,F20 F69)
	4(for the functions of: F33,F45,F49,F50)	3 (for the functions of: F7,F21,F50)	4(for the functions of: F14,F19,F32,F49)	5(for the functions of: F16,F24,F25,F68,F70)
	2(for the functions of: F7,F15)	3 (for the functions of: F17,F41,F42)	4(for the functions of: F3,F7,F11,F15)	3(for the functions of: F31,F50,F63)
	2(for the functions of: F24,F52)	3 (for the functions of: F3,F9,F70)	4(for the functions of: F2,F12,F24,F52)	
		2 (for the functions of: F20,F69)	2 (for the functions of: F4,F54)	
		2 (for the functions of: F64,F68)	2 (for the functions of: F1,F53)	
		2 (for the functions of: F16,F25)		
DIFF	35 (for the functions of: F1,F2,F3,F4,F8 F9,F10,F11,F12,F13 F16,F17,F18,F20,F21 F25,F34,F41,F42,F43 F44,F46,F48,F512,F53 F54,F59,F60,F63,F64 F66,F67,F68,F69,F70)	25 (for the functions of: F1,F2,F4,F5,F8 F10,F11,F12,F13,F15 F18,F24,F34,F43,F44 F46,F48,F51,F52,F53 F54,F59,F60,F66,F67)	22 (for the functions of: F8,F9,F10,F13,F27 F33,F34,F41,F43,F44 F45,F46,F48,F51,F59 F60,F63,F64,F66,F67 F68,F69)	28 (for the functions of: F1,F2,F4,F5,F7 F8,F10,F11,F12,F13 F17,F21,F34,F41,F42 F43,F44,F46,F48,F51 F52,F53,F54,F59,F60 F64,F66,F67)

SAME denotes the number of functions that have equal values of computation time for ACSO, RIO, DSARIO, PSO algorithms, DIFF denotes number of functions that have different values of computation time for ACSO, RIO, DSARIO, PSO algorithms.

5.7 Multi-Valued Discrete Space Cockroach Swarm Optimization

DCSO model of section 3.8.1 was simulated and its performance evaluated in this section. The performance of the DCSO algorithm with binary, ternary and quaternary base systems was tested on a set of global optimization benchmarks shown in Table 15 of Appendix A and compared with the results of the existing discrete particle swarm optimization (DPSO)[55]. The benchmarks are adopted from [55, 136, 137] and with their characteristics include uni-modal, multi-modal, shifted, rotated, separable, non-separable, scalable, and noise in fitness. Experimental settings of [55] is considered in this study for good comparison; 10 dimensions are considered for the benchmarks, swarm size is 20, $x_{max} = 4$. The algorithm run 25 times for each benchmark with 5000 iterations each. The numerical results of DCSO for binary, ternary and quaternary number bases were recorded in this work after implementing the algorithm. The results are depicted in Table 5.42. The numerical results of DPSO (binary DPSO, ternary DPSO and quaternary DPSO) is as adopted in [55].

The comparison results of DCSO with DPSO [55] is depicted in Tables 5.43 and 5.44. Table 5.43 shows the comparison average performance while Table 5.44 shows comparison standard deviation of the average optimal values of DCSO and DPSO for binary, ternary and quaternary search spaces respectively. The DCSO algorithm is shown statistically to have improved performance over DPSO algorithm. The DCSO algorithm has consistent performance in each iteration. This is proved by low standard deviation of the optimal mean in Table 5.42.

Mann-Whitney U-test statistic test was carried out in this work for algorithm performance comparison. This is to determine if DCSO algorithm performance is significantly different from that of the DPSO algorithm. Results of the Mann-Whitney U-test is shown in Tables 5.40 and 5.41. P-values (Asymp. Sig.) for the Mann-Whitney U-test shown in Table 5.41 is 0.005, it is less than the threshold value 0.05. Therefore there is significant difference in the performance of DCSO and DPSO on the selected benchmark problems.

Table 5.40: Mann-Whitney U-test statistics on the performance of DCSO and DPSO.

Ranks

	Algorithm	N	Mean Rank	Sum of Ranks
Fitness	1.00	15	11.00	165.00
	2.00	15	20.00	300.00
	Total	30		

The effect size r of the significant difference was computed from Mann-Whitney U-test of Tables 5.40 and 5.41. $r = Z/\sqrt{N}$; $Z = -2.808$ and $N = 30$ (total number of samples). Effect size r

Table 5.41: Mann-Whitney U-test statistics.

Test Statistics[b]

	Fitness
Mann-Whitney U	45.000
Wilcoxon W	165.000
Z	-2.808
Asymp.Sig(2 tailed)	.005
Exact Sig[2*(1 tailed Sig)]	.004[a]

a Not corrected for ties
b Computing variable: Algorithms

is calculated to be -0.4870 (Z is negative when observed data is below the mean and positive when above). The effect size 0.4870 is of large effect size (close to large effect size of 0.5), using the Cohen guideline on effect size [132, 133], r $[Small : 0.1, Medium : 0.3, Large : 0.5]$. This means that there is a difference of a large magnitude in the performance of DCSO and DPSO.

Table 5.42: Performance of DCSO.

	F1	F2	F3	F4	F5
Binary DCSO					
Best	-1.4000E+02	-3.3000E+02	-3.3000E+02	9.0000E+02	-4.6000E+02
Average	-1.3898E+02	-3.3000E+02	-3.2988E+02	9.0000E+01	-4.5736E+02
STD	1.6614	0.0000	3.3166E-01	0.0000	4.7341
Ternary DSCO					
Best	-1.4000E+02	-3.3000E+02	-3.3000E+02	9.0000E+01	-4.60004+02
Average	-1.3811E+02	-3.2940E+02	-3.2976E+02	9.0000E+01	-4.5426E+02
STD	1.8486	5.0000E-01	4.3589E-01	0.0000	1.8461E+01
Quaternary DSCO					
Best	-1.40000E+02	-3.3000E+02	-3.3000E+02	9.0000E+01	-4.6000E+02
Average	-1.3768E+02	-3.2920E+02	-3.2940E+02	9.0000E+01	-4.5438E+02
STD	1.7761	4.0825E-01	5.0000E-01	0.0000	1.3696E+01

STD denotes standard deviation.

Table 5.43: Comparison of Average Performance of DCSO and DPSO.

	F1	F2	F3	F4	F5
Binary					
DCSO	**-138.98**	**-330**	**-329.88**	**90**	**-457.36**
DPSO	-119.63	-287.08	-271.18	99.03	255596.9
Ternary					
DCSO	**-138.11**	**-329.40**	**-329.76**	**90**	**-454..26**
DPSO	-119.66	-300.72	-276.1	97.83	23199.19
Quaternary					
DCSO	**-137.68**	**-329.20**	**-329.40**	**90**	**-454.38**
DPSO	-119.64	-309.12	-283.87	97.50	14986.13
Optimum	-140	-330	-330	90	-460

Table 5.44: Comparison of STD of mean optimal of DCSO and DPSO.

	F1	F2	F3	F4	F5
Binary					
DCSO	1.6614	**0.0000**	**0.33166**	**0.0000**	4.7341
DPSO	**0.06**	5.80	6.15	0.49	6231.9
Ternary					
DCSO	1.8486	**0.5**	**0.43589**	**0.0000**	**18.4610**
DPSO	**0.07**	4.11	8.45	0.87	6847.43
Quaternary					
DCSO	1.7761	**0.40825**	**0.5**	**0.0000**	**13.6960**
DPSO	**0.068**	4.68	6.39	0.89	5215.12

STD denotes standard deviation.

5.8 DCSO-TSP

DCSO application to TSP described in section 3.8.2 was simulated and tested on 53 instances of TSP taken from TSP library [123] to validate its feasibility and effectiveness. Two stages of experiments were conducted. Stage one experiments compared the results of DSCO-TSP with the results of existing CSO and PSO for TSP for the OLIVER 30 problem reported by Cheng et al., [95] using the same experimental settings of 200 cockroaches, 1000 maximum iterations and the experiment runs 8 times. The comparison results depicted in Table 5.45 show that DCSO-TSP also finds the optimum path of 423.74 like the comparison algorithm. Figures 5.5 and 5.6 depict the graphical illustration of results of the proposed algorithm on Oliver 30 instance.

The stage two experiments were conducted to evaluate the algorithm on many instances of TSP. DCSO-TSP was used to solve 52 TSP instances and compared with the results of hybrid genetic algorithm presented by Ahmed [138] for symmetric TSP instances. We used the experimental settings of [138] for the experiments; 20 population size and 20 trials. Maximum iteration 10E4 was used for our experiments.

The results of the experiments of DCSO-TSP were recorded: Best solution, percentage errors $(Error(\%))$ of the best solution, average solution and average complete computational time (CTime) are depicted in Table 5.46. The percentage errors were computed using equation (5.1). The results of HGA is as recorded by Ahmed [138]. Tables 5.47 and 5.48 show the comparison results of DSCO-TSP and HGA. The comparison results show that DCSO-TSP is able to compete with the well-known GA for evaluating 52 TSP instances. Comparison results show that DCSO-TSP is able to obtain better tour for some instances than HGA such as burma 14, ulysses 16, dantzig 42, ulysses 22, and gr 99.

$$Error = \frac{|f(x^*) - f_{min}|}{f(x^*)} \times 100\% \tag{5.1}$$

Table 5.46 shows the results of DCSO-TSP for 52 instances from a small number of cities to a large number of cities. Figures 1, 2,3,4, 5, 6, 7, 8, 9,10, 11 and 12 of Appendix C depicts the graphical illustration of results of the DCSO-TSP for 14, 42, 225, 280, 262 and 299 cities respectively. The Figures show the total distance (minimum distance), the city locations and the solution found so far for each instance.

ANOVA test statistic was conducted on the best and average solutions of HGA and DCSO-TSP to verify significant difference. The P-Value from the ANOVA test is 0.237 and 0.348 for best and average solutions respectively. The null hypothesis suggests that there is no significant difference in the best and average performances of HGA and DCSO in solving TSP problems. Tables 5.49 and 5.50 show the results of the ANOVA test for the best and average solutions.

Table 5.45: Performance of DCSO-TSP, CSO and PSO on OLIVER 30 TSP Instance

Ranks

	Algorithm	N	Run1	Run2	Run3	Run4	Run5	Run6	Run7	Run8	Ave	Errors
DCSO-TSP	200	423.74	423.74	423.74	423.74	423.74	423.74	423.74	423.74	423.74	423.74	0.00%
CSO	200	423.74	423.74	423.74	423.74	423.74	423.74	423.74	425.48	423.74	423.96	0.05%
PSO	800	432.66	423.74	425.27	425.48	423.95	423.74	426.31	423.74	424.49	424.49	0.28%

AVE denotes average length of eight experiments. Error is computed using equation 5.1

Table 5.46: DCSO-TSP for Symmetric TSPLIB Instances

SN	Instance	BestKnownTour	BestSol	Error(%)	AvgSol	CTime
1	eil51	426	428	0.67	436.67	27.83
2	pr76	108159	109123	0.89	112124.66	30.95
3	berlin52	7542	7544	0.03	7930.24	35.01
4	st70	675	682	1.13	698.32	46.65
5	burma14	3323	30	-99.07	30.87	22.92
6	ulysses16	6859	73	-98.92	73.99	25.14
7	bayg29	1610	9074	463.61	9080.29	34.29
8	bays29	2020	9074	349.22	9086.43	36.32
9	dantzig42	699	688	-1.53	688.31	27.48
10	swiss42	1273	1273	0	1292.5	23.06
11	ulysses22	7013	75	-98.93	75.42	46.22
12	eil76	538	547	1.73	565.61	84.04
13	gr96	55209	520	-99.06	529.61	56.61
14	rat99	1211	1248	3.08	1290.97	66.73
15	kroA100	21282	21756	2.23	22599.28	35.64
16	kroB100	22141	22741	2.71	23358.66	41.26
17	kroC100	20749	21144	1.9	21793.43	43.38
18	kroD100	21294	21681	1.82	22490.95	55.57
19	kroE100	22063	22422	1.63	23008.95	62.82
20	rd100	7910	7981	0.9	8340.21	68.6
21	eil101	629	654	4.03	668.74	65.95
22	lin105	14379	14560	1.26	15265.15	64.21
23	pr107	44303	44509	0.47	45487.11	54.28
24	gr120	6942	1655	-76.15	1706.49	112.5
25	pr124	59030	59887	1.45	61245.03	56.39
26	bier127	118282	120863	2.18	124839.96	121.8
27	ch130	6110	6256	2.4	6459.56	149.39
28	pr136	96772	99416	2.73	102675.18	117.95
29	gr137	69853	712	-98.98	732.04	70.77
30	pr144	58537	58761	0.38	61219.27	35.34
31	kroA150	26524	26696	0.65	28177.86	54.7
32	kroB150	26130	26786	2.51	27755.99	65.32
33	ch150	6528	6620	1.42	6999.02	78.58
34	pr152	73682	74655	1.32	76385.98	59.72
35	u159	42080	42391	0.74	42415.31	40.57
36	rat195	2323	2412	3.86	2479.88	63.67
37	d198	15780	16215	2.76	16595.11	73.93
38	kroA200	29368	29981	2.09	31683.45	226.88
39	kroB200	29437	30952	5.15	31822.45	257.58
40	gr202	40160	496	-98.76	507.61	14520
41	ts225	126643	126963	0.25	133386.97	148.04
42	tsp225	3916	3997	2.07	4122.34	180.52
43	pr226	80369	81494	1.4	85001.4	128.65
44	gr229	134602	1682	-98.75	1715.64	91.9
45	gil262	2378	2524	6.17	2587.83	137.11
46	pr264	49135	54161	10.23	56485.01	115.16
47	a280	2579	2722	5.58	2740.49	106.44
48	pr299	48191	51138	6.12	52978.37	135.35
49	lin318	42029	44945	6.94	46483.71	187.75
50	rd400	15281	16747	9.6	17127.75	413.27
51	fl417	11861	12537	5.71	12979.42	336.22
52	gr431	171414	2049	-98.80	2083.21	260.03

Table 5.47: DCSO-TSP and HGA for Symmetric TSPLIB Instances

			HGA [114]				DCSO-TSP			
SN	Instance	BestKnownTour	BestSol	Error(%)	AvgSol	CTime	BestSol	Error(%)	AvgSol	CTime
1	eil51	426	681	-1.16	681.00	1.73	428	0.67	436.67	27.83
2	pr76	108159	108159	0.00	108254.55	7.46	109123	0.89	112124.66	30.95
3	berlin52	7542	7542	0.00	7542.00	3.63	7544	0.03	7930.24	35.01
4	st70	675	675	0.00	677.15	6.15	682	1.13	698.32	46.65
5	burma14	3323	3621	0.00	3621	0.25	30	-99.07	30.87	22.92
6	ulysses16	6859	7303	0.00	7303.00	0.30	73	-98.92	73.99	25.14
7	bayg29	1610	2144	0.00	2144.00	0.93	9074	463.61	9080.29	34.29
8	bays29	2020	2702	0.00	2702.00	0.94	9074	349.22	9086.43	36.32
9	dantzig42	699	699	0.00	699.00	1.15	688	-1.53	688.31	27.48
10	swiss42	1273	1919	0.00	1923	1.59	1273	0.00	1292.50	23.06
11	ulysses22	7013	8190	0.00	8190.00	0.54	75	-98.93	75.42	46.22
12	eil76	538	538	0.00	539.26	7.33	547	1.73	565.61	84.04
13	gr96	55209	55209	0.00	55672.85	12.07	520	-99.06	529.61	56.61
14	rat99	1211	1211	0.00	1218.40	1.16	1248	3.08	1290.97	66.73
15	kroA100	21282	21282	0.00	21321.80	13.02	21756	2.23	22599.28	35.64
16	kroB100	22141	22141	0.00	22193.15	13.48	22741	2.71	23358.66	41.26
17	kroC100	20749	20749	0.00	20789.45	12.62	21144	1.90	21793.43	43.38
18	kroD100	21294	21294	0.00	21389.11	11.16	21681	1.82	22490.95	55.57
19	kroE100	22063	22068	0.00	22116.39	14.52	22422	1.63	23008.956	62.82
20	rd100	7910	7910	0.00	7932.70	13.46	7981	0.90	8340.21	68.60
21	eil101	629	629	0.00	632.75	16.07	654	4.03	668.74	65.95
22	lin105	14379	14329	0.00	14416.65	14.39	14560	1.26	15265.15	64.21
23	pr107	44303	44303	0.00	44405.67	14.56	44509	0.47	45487.11	54.28
24	gr120	6942	6942	0.00	6986.95	20.73	1655	-76.15	1706.49	112.50
25	pr124	59030	59030	0.00	59181.75	20.77	59887	1.45	61245.03	56.39

Table 5.48: DCSO-TSP and HGA for Symmetric TSPLIB Instances continued

			HGA [114]				DCSO-TSP			
SN	Instance	BestKnownTour	BestSol	Error(%)	AvgSol	CTime	BestSol	Error(%)	AvgSol	CTime
26	bier127	118282	118282	0.00	118419.60	28.27	120863	2.18	124839.96	121.80
27	ch130	6110	6110	0.00	6150.50	30.12	6256	2.40	6459.56	149.39
28	pr136	96772	96722	0.00	97240.80	28.36	99416	2.73	102675.18	117.95
29	gr137	69853	69853	0.00	70429.50	28.16	712	-98.98	732.04	70.77
30	pr144	58537	58537	0.00	5867.19	30.83	58761	0.38	61219.27	35.34
31	kroA150	26524	26524	0.00	26629.65	35.85	26696	0.65	28177.86	54.70
32	kroB150	26130	26130	0.00	26264.23	38.07	26786	2.51	27755.99	65.32
33	ch150	6528	6528	0.00	6556.31	38.59	6620	1.42	6999.02	78.58
34	pr152	73682	73682	0.00	74017.45	34.25	74655	1.32	76385.98	59.72
35	u159	42080	42080	0.00	42336.10	38.87	42391	0.74	42415.31	40.57
36	rat195	2323	2323	0.00	2362.20	71.36	2412	3.86	2479.88	63.67
37	d198	15780	15800	0.13	15858.25	77.79	16215	2.76	16595.11	73.93
38	kroA200	29368	29420	0.18	29618.80	70.41	29981	2.09	31683.45	226.88
39	kroB200	29437	29403	0.09	29807.00	75.01	30952	5.15	31822.45	257.58
40	gr202	40160	40160	0.00	40413.85	97.25	496	-98.76	507.61	14520
41	ts225	126643	126643	0.00	127006.83	93.85	126963	0.25	133386.97	148.04
42	tsp225	3916	3923	0.18	3967.35	105.30	3997	2.07	4122.34	180.52
43	pr226	80369	80467	0.12	80953.60	87.69	81494	1.40	85001.40	128.65
44	gr229	134602	134957	0.26	136184.35	119.79	1682	-98.75	1715.64	91.90
45	gil262	2378	2391	0.55	2403.15	151.26	2524	6.17	2587.83	137.11
46	pr264	49135	49219	0.17	49814.45	149.99	54161	10.23	56485.01	115.16
47	a280	2579	2585	0.23	2614.05	187.03	2722	5.58	2740.49	106.44
48	pr299	48191	48375	0.38	48857.06	202.84	51138	6.12	52978.37	135.35
49	lin318	42029	42301	0.65	42679.65	253.92	44945	6.94	46483.71	187.75
50	rd400	15281	15370	0.58	15452.20	546.74	16747	9.60	17127.75	413.27
51	fl417	11861	11930	0.58	12004.63	544.17	12537	5.71	12979.42	336.22
52	gr431	171414	173270	1.08	176047.20	716.64	2049	-98.80	2083.21	260.03

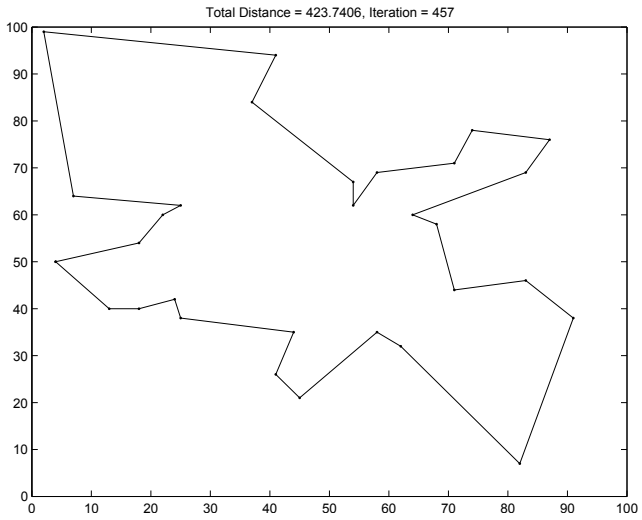

Figure 5.5: Graphs illustrating Oliver 30 instance total distance.

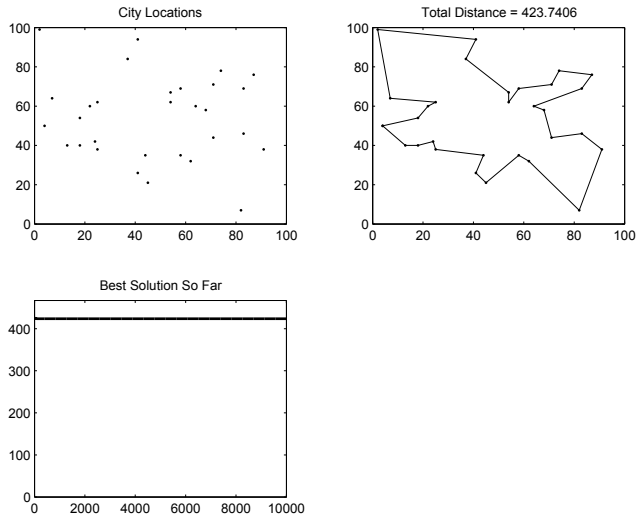

Figure 5.6: Graphs illustrating DCSO-TSP Tour for Oliver 30 Cities.

Table 5.49: ANOVA test for HGA and DCSO on Best Solutions

	Sum of Square	df	Mean Square	F	Sig.
Between Groups	2.0E+009	1	1952614454	1.417	.237
Within Groups	1.4E+011	102	1378199403		
Total	1.4E+011	103			

Table 5.50: ANOVA test for HGA and DCSO on Average Solutions

	Sum of Square	df	Mean Square	F	Sig.
Between Groups	1.3E+009	1	12745438884	.889	.348
Within Groups	1.5E+011	102	1433892446		
Total	1.5E+011	103			

Chapter 6

Discussion and Conclusion

6.1 Discussion

The effects of the new components, operators and methods introduced into RBA to evolve the improved RBA have been shown through empirical and statistical analysis results. The searching abilities, convergence speed, exploitation and exploration have been enhanced as shown in the improved performance of RBA variants over the existing RBA and PSO algorithms shown in Chapter 5.

The effect of stochastic constriction on characteristics of the CSO algorithm is shown in this work in section 5.2. Simulation results revealed that the proposed SCCSO algorithm has good convergence capability. A constriction factor enabled the algorithm to maintain swarm stability and enhances local and global searches which resulted in improved convergence and speed of the algorithm. The SCCSO algorithm runs fast, solving benchmark problems up to 3000 dimensions, without modifying the algorithm, and it can evaluate a higher number of variables above 3000 dimensions. Comparisons of results of SCCSO with the existing CSO and MCSO show its superiority. A comparison of SCCSO with LSRS which has been proved in the literature, for high dimensions up to 2000, shows SCCSO has the ability to compete with existing global optimization technique.

A further improvement was made on the CSO algorithm by extending it with a hunger component whereby enhancing the algorithm's diversity and searching capabilities. ICSO is proposed, and the efficiency of the proposed algorithm has been shown through empirical studies where its performance was compared with that of existing RBA: CSO, MSCO, RIO and HRIO. Results show that ICSO is significantly better than the existing algorithms. The statistics of 0.3 effect size shows there is a significant difference of medium magnitude between the proposed algorithm and the existing algorithms. The Jonckheere-Terpstra test statistic was used for the statistical analysis to

compare the results of ICSO with existing algorithms. Test statistic results revealed that there is a significant difference of medium magnitude in the performance of ICSO with CSO, MCSO, RIO and HRIO algorithms.

Likewise, the effect of a simple Euler step–size adaptation is examined on a RIO algorithm. Simulation studies show improved performance of the proposed DSARIO algorithm over RIO and HRIO algorithms. The numerical results show that DSARIO solves multi–dimensional benchmark problems, where it was tested up to 30 dimensions, and it can still solve a higher number of variables without modification. The evaluation of the results of the algorithms were carried out by test statistics of analysis of variance (ANOVA). The results of ANOVA shows there is a significant difference of very large effect in the performance of DSARIO and that of RIO and HRIO.

Also, the MRIO algorithm that is tied to the social behaviour of cockroaches is presented in this work to improve on the original RIO algorithm. Modification of the existing RIO models was done and two new components namely vigilance and cannibalism were added. MRIO was evaluated on known function benchmarks empirically, and the proposed algorithm found global optimal for the selected benchmarks. The superiority of MRIO to the exiting RIO and HRIO was revealed through performance comparisons.

Moreover, the introduction of PPEM with crossover and mutation techniques to the CSO algorithm helped the algorithm to create adaptive searches in each iteration enhancing population diversity and local and global searches as revealed by improved results through simulation studies. 70 global optimization benchmark problems were used to evaluate the success of ACSO, RIO, DSARIO and PSO and their results were statistically compared. The searching techniques of ACSO are similar to DSARIO in that, they both create adaptive searches in each iteration maintaining a balance of exploitation and exploration. The Euler step-size h created adaptive searches and enhanced swarm stability in DSARIO. PSO and RIO have similar searching methods but RIO has the greater capability of escaping local convergence than has PSO. The find food component of RIO enhances diversity. The known stability problem of PSO limits its success rate despite its ability to solve many benchmark functions. From the comparison results, ACSO performs better that the comparison algorithms.

Finally, the proposed multi-valued discrete cockroach swarm optimization (DCSO) algorithm was tested using standard test function benchmarks and compared with discrete particle swarm optimization (DPSO). Empirical and statistical analysis results show that DCSO outperforms DPSO. The proposed algorithm was also applied to TSP using 53 instances of TSP problems from TSPLIB. The results of DCSO on TSP were compared with the results of existing CSO, PSO and HGA from the literature; the comparison results show similar performance.

6.2 Summary of Findings

This thesis set out to improve the performance of the existing RBA. The research work achieved its stated objectives which are: proposing improved RBAs from the existing RBA; making RBA a global optimization technique with improved convergence on global optima by extending its models; constructing and testing adaptive RBAs based on methods and models that have been proven to be successful when applied to other evolutionary techniques; constructing and testing discrete value space RBA for solving global optimization problems with discrete values, based on the methods that have been successfully applied to other SI techniques and evaluation of the proposed RBA variants with standard global optimization test problems and comparing the obtained results with those in the literature.

The speed and the performance of the CSO algorithm was improved by the incorporation of a stochastic constriction factor. In this thesis a stochastic constriction cockroach swarm optimization that is able to solve problems with thousands of variables was developed.

The thesis set out to improve the searching abilities of the CSO algorithm by the inclusion of an additional component called hunger. The hunger component enhanced diversity, exploitation and exploration of the algorithm.

Furthermore, predator-prey evolution with cross over and mutation approach were employed to create an adaptive search. The CSO algorithm is able to recursively create a pattern matrix in each iteration which improves exploitation, exploration and good convergence. The technique ultimately helps the algorithm to avoid getting trapped at local optima. Diversity is introduced into a population by mutation, whenever the population tends to converge at a local optimal point. Also a point is created in the neighbourhood of the current point by a mutation process, which allows a local search around the point. The technique also helps the algorithm to avoid the possibility of population collapse through the evolution process.

Real world problems are majorly discrete-valued in nature. This thesis extends the existing CSO algorithm that was primarily designed for continuous problems and constructs a multi-valued discrete algorithm. This is achieved with the use of sigmoid function and an operator for the logistic transformation of values and the generation of a solution that is rounded to the nearest discrete variable.

Similarly, the RIO algorithm was improved upon by the engagement of a simple Euler step-size adaptation method. This technique helps RIO to avoid falling into local optimum by maintaining balance between exploitation and exploration. The technique also enhances swarm stability.

Furthermore, the performance and the searching ability of RIO were greatly enhanced by redesigning the existing components and the incorporation of two additional components namely vigilance and cannibalism in the modifications. The vigilance component creates population di-

versity, as cockroaches are able to explore the search region for a solution. The cannibalism component enhances both local and global searches.

Finally, the performance of the improved algorithms have been compared with existing algorithms, both empirically and statistically. The comparison results confirmed the superiority of the improved algorithms over the existing algorithms.

The new components and operators, introduced into cockroach algorithms to evolve into the improved algorithms have been discussed and shown to perform better than the existing algorithms.

6.3 Suggestions for Future Research

Every answered questions makes room for many new ones to come to light. A few thoughts for extensions and future work are highlighted below:

1. Constrained global optimization problems: In this thesis, the improved algorithms were used mainly for solving unconstrained continuous optimization problems. Application of the improved RBA to various constrained problems is recommended for further work.

2. Combinatorial optimization problems: The developed discrete-valued cockroach swarm optimization was tested on TSP. Its application to more combinatorial optimization problems is recommended.

3. Multi-objective RBA Optimization: Most real world problems are situations where decision making involves a trade-offs between two or more conflicting objectives. Scenarios such as product and design, automobile design, finance, aircraft design, and the oil and gas industry. Cockroach algorithms have searching strategies that can be used for multi-objective optimization. Extension of the improved algorithms for multi-objective problems can be followed up.

4. Hybridization of improved RBA with various techniques: Development of hybrid cockroach algorithms is another possible extension of this work. Cockroach algorithm can be hybridized with techniques such as Simulated Annealing, Tabu search etc. for better performance.

Bibliography

[1] R. A. Sarker and C.S. Newton, "Optimization Modelling: A practical Approach", CRC Press, Taylor and Francis Group, (2008).

[2] T. Weise, "Global Optimization Algorithms: Theory and Application", Available Online, http://www.it-weise.de/, (2009).

[3] E. Parsopoulos and N. Vrahatis, "Particle Swarm Optimization and Intelligence: Advances and Applications" United State of America: IGI Global, (2010).

[4] Wisconsin Institute of Discovery, "NEOS Optimization Guide", University of Wisconsin-Madison, (2013).

[5] A. A. Torn and A. Zilinskas, "Global Optimization", Lecture Notes in Computer Science, Springer, **350**, (1989).

[6] W. E. Hart, "Adaptive Global Optimization with Local Search", PhD Thesis, University of California, San Diego, USA, (1994).

[7] A. Engelbrecht, "Computational Intelligence", John Willey and Sons Limited, (2002).

[8] K. Amit, " Computational Intelligence Principles, Techniques and Applications", New York: Springer Berlin, Heidelberg, (2005).

[9] F. D. Bergh, "An Analysis of Particle Swarm Optimizer", A PhD Thesis Submitted to Faculty of Natural and Agricultural Science, University of Pretoria, Pretoria ,South Africa, (2001).

[10] A. Barricelli, "Numerical Testing of Evolution Theories, Part I-Theoretical Introduction and Basic Test", ACTA, Biotheoretical Journal of Humanities Social Science and Law, **16 (1-2)**, 69-98, (1961).

[11] A. Barricelli, "Numerical Testing of Evolution Theories, Part II- Premiminary Performance of Symbiogenesis and Terrestrial Life", ACTA, Biotheoretical Journal of Humanities Social Sciences and Law, **16 (3-4)**, 99-126, (1963).

[12] I. Rechenberg, “Evolutions Strategie Optimierung ”, Technischer Systeme der Biologischen Evolution, (1971).

[13] J. Holland, “ Adaptation in Natural and Artificial Systems”, Ann Arbor: University of Michigan Press, (1975).

[14] M. Jamshidi, “Tools for Intelligent Control: Fuzzy Controllers, Neural Networks and Genetic Algorithms”, Philosophical Transaction Series, Mathematical, Physical and Engineering Sciences, **361(1809)**, 781-808, (2003).

[15] C. Blum and D. Merkle, “Swarm Intelligence Introduction and Applications”, Natural Computing Series, Springer-verlag Berlin Heidelberg, (2008).

[16] D. Karaboga and A. Bahriye, “A survey: Algorithm Simulating Bee Swarm Intelligence”, Springer Science and Business Media, (2009).

[17] E. Bonabeau, and M. Christopher, “Swarm Intelligence: A Whole Way of Thinking About Business”, Harvard Business School Publishing Corporation, 106-114, (2001).

[18] C. Reynolds, “Flocks, Herds and Schools: A Distributed Behavioural Model”, Computer Graphics, **21(4)**, (1987).

[19] Swarm behaviour (2015 March 2). Retrieved from http://en.wikipedia.org/wiki/Swarm-behaviour.

[20] F. Dario and M. Claudio, “Bio-Inspired Artificial Intelligence: Theories, Methods and Technology”, Massachusetts, London, England: MIT Press Cambridge, (2008).

[21] S. Camazine, J. Deneubourg, N. Frank, J. Sneyd, G. Thranlaz and E. Bonabeau, “Self Organization in Biological Systems”, Princeton NJ: Princeton University Press, (2001).

[22] R. May, “Biological Population with Non-overlapping Generations: Stable Points, Stable Cycles, and Chaos”, Science, **186**, 645-647, (1974).

[23] E. Bonabeau and M Dorigo, “Swarm Intelligence: From Natural to Artificial Systems”, New York: Oxford University Press, (1999).

[24] C. David, A. Reynold, and E. Bonabeau, ”Swarm Intelligence, “ In Handbook of Natural Computing, Springer Berlin Heidelberg, 1599-1622, (2012).

[25] Termite Mound (July 2013). Retrieved from http://www.metropolismag.com/Point-of-View/July-2013/New-York-Paves-the-Way-for-Nature-Inspired-Innovation/

[26] Bees Hive. Retrieved from http://www.gettyimages.com/detail/photo/honey-bees-full-frame-at-dusk-royalty-free-image/73092580.

[27] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization", IEEE Neural Networks Proceedings, **4**, 1942-1948, (1995).

[28] M. Dorigo, "Optimization, Learning and Natural Algorithms," PhD Thesis, Politecnico di Milano, Italy, (1992)

[29] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi,"The Bees Algorithm :Technical Note", UK: Manufacturing Engineering Center, Cardiff University, (2005).

[30] K. Passino, "Biomimicry of Bacteria Foraging for Distributed Optimization and Control", IEEE Control Systems Magazine, (2002).

[31] K. Krishnanand and D. Ghose, " Detection of Multiple Sources Location Using a Glow-worm Metaphor with Applications to Collective Robotics", IEEE Swarm Intelligence Symposium, 84-91, (2005).

[32] D. R. Oliveira,R.S. Parpinelli and H.S Lopes, "Bioluminescent Swarm Optimization Algorithm", In-Tech, (2011).

[33] X. Yang, "Firefly Algorithms for Multimodal Optimization", Lecture Notes in Computer Sciences, 5792, 169-178, 2009.

[34] T. Havens, C. Spain, N. Salmon and J. Keller, "Roach Infestation Optimization", IEEE Swarm Intelligence Symposium, (2008).

[35] C. ZhaoHui and T. HaiYan, "Cockroach Swarm Optimization", Computer Engineering and Technology (ICCET) Second International Conference, IEEE, **6**, (2010)

[36] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", Advances in Engineering Software, **69**, 46-61, 2014.

[37] X.S Yang and S. Deb, "Cuckoo Search via Levy Flights", World Congress on Nature and Biologically Inspired Computing (NaBIC 2009), IEEE Publications, 210-214, (2009).

[38] H. Shah-Hosseini, "Problem Solving by Intelligent Water Drops", Proceedings of the IEEE Congress on Evolutionary Computation, 3226-3231, (2007).

[39] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm", Information Science, **179 (13)**, 2232-2248, (2009).

[40] K. R. Foster, T. Wenseleers and L. W. Ratnieks, "Kin Selection is the Key to Altruism", Trends Ecol Evol, **21**, 57-60, (2006).

[41] L. N. De-Castro and F. J. Von Zuben, "Learning and Optimization Using the Clonal Selection Principle", IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems (IEEE), **6 (3)**, 239-251, (2002).

[42] A. Kaveh and S. Talatahari, "A Novel Heuristic Optimization Method: Charged System Search", Acta Mech, **213**, 267-289, (2010).

[43] X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", In: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin, **284**, Springer, 65-74, (2010).

[44] K. Nara, T. Takeyama, and H. Kim, "A New Evolutionary Algorithm Based on Sheep Flocks Heredity Model and its Application to Scheduling Problem", In Systems, Man, and Cybernetics, 1999 IEEE SMC'99 Conference Proceedings, **6**, 503-508, (1999).

[45] M. Clerc, "Particle Swarm Optimization", Newport Beach, CA: ISTE USA, (2006).

[46] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer", IEEE International Conference on Evolutionary Computation, 69-73, 1998.

[47] J. Kennedy and R. Eberhart, "Swarm Intelligence", Morgan Kaufmann, San Francisco, USA, (2001).

[48] D. Bratton and J. Kennedy, "Defining A Standard for Particle Swarm Optimization", Proceedings of 2007 IEEE Swarm Intelligence Symposium, (2007).

[49] X. Cui, "Swarm Intelligence", Applied Software Engineering Research Group Oak Ridge National Laboratory, (2009).

[50] M. Beekman, G. Sword and S. Simpson, "Biological Foundation of Swarm Intelligence", Swarm Intelligence, Natural Computing Series, Springer-verlag Berline Heidelberg, ISBN 978-3-540-74088-9, 3-41, (2008).

[51] F. Yoshikazu, N. Hideyuki and T. Yuji, "Particle Swarm Optimization for the Optimal Operational Planning of Energy Plants" Innovations in Swarm Intelligence Studies in Computational Intelligence, Edited by Chee Pen Lim, Lakhmi C. Jain, and Satchidananda Dehuri, Springer-Verlag Berlin Heidelberg, 159-174, (2009).

[52] P. Chen, "Particle Swarm Optimization for Power Dispatch with Pumped Hydro", Particle Swarm Optimization, Edited by Aleksandar Lazinica, Published by In-Tech, Printed in Croatia, 131-144, (2009).

[53] I. Omar, and B. Cees, "A particle optimization Approach to Graph Permutation", Particle Swarm Optimization, Edited by Aleksandar Lazinica, Published by In-Tech, Printed in Croatia, 291-312, (2009).

[54] J. Kennedy and R. Eberhart, "A Discrete Binary Version of the Particle Swarm Algorithm", IEEE International Conference on System, Man, and Cybernetics, (1997).

[55] K. Veeramacheneni, L. Osadciw and G. Kamath, "Probabilistically Driven Particle Swarms for Optimization of Multi-valued Discrete Problems: Design and Analysis", In IEEE swarm intelligence symposium, 141-149 (2007).

[56] J. pugh and A. Martinoli, "Discrete Multi-valued Particle Swarm Optimization", IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA, (2006)

[57] B. Al-Kazemi and C.K. Mohan, "Multi-Phase Discrete Particle Swarm Optimization", Proceedings of fourth International Workshop on Frontiers in Evolutionary Algorithms, (2002).

[58] H. Chen, S. Li, and Z. Tang, "Hybrid Gravitational Search Algorithm with Random-key Encoding Scheme Combined with Simulated Annealing", International Journal of Computer Science and Network Security, **11 (6)**, 208-217, (2011).

[59] X. Li, J. Wang, J. Zhou and M. Yin, "An Effective GSA Based Memetic Algorithm for Permutation Flow-shop Scheduling", In: IEEE Congress on Evolutionary Computation, 1-6, (2010).

[60] R. Verma and S. Kumar, "DSAPSO: DNA Sequence Assembly using Continuous Particle Swarm Optimization with Smallest Position Value Rule", In: 1st International Conference on Recent Advances in Information Technology, 410-415, (2012).

[61] A. Yousif, A. H. Abdullah, S. M Nor, and A. A. Abdelaziz, "Scheduling Jobs on Grid Computing using Firefly Algorithm", Journal of Theoretical and Applied Information Technology, **33 (2)**, 155-164, (2011).

[62] Q.-Q. Pan, M.F. Tasgetiren, and Y. -C. Liang, "A Discrete Particle Swarm Optimization Algorithm for the No-wait Flow-shop Scheduling Problem", Computers and Operations Research, **35 (9)**, 2807-2839, (2008).

[63] M. F. Tasgetiren, P. N. Suganthan and Q.-Q. Pan, "A discrete Particle Swarm Optimization Algorithm for the Generalized Travelling Salesman Problem", In: 9th Annual Conference on Genetic and Evolutionary Computation. ACM Press, New York, USA, 158-165, (2007).

[64] L. Congying, Z. Huanping, and Y. Xinfeng, "Particle Swarm Optimization Algorithm for Quadratic Assignment Problem", In: International Conference on Computer Science and Network Technology, **3**, 1728-1731, (2011).

[65] S. Burnwal and S. Deb, "Scheduling Optimization of Flexible Manufacturing System using Cuckoo Search-based Approach", The International Journal of Advanced Manufacturing Technology, 1-9, (2012).

[66] S. Labed, A. Gherboudj and S. Chikhi, "A modified hybrid particle swarm optimization algorithm for solving the travelling salesman problem", Journal of Theoretical and Applied Information Technology, **39(2)**, (2012).

[67] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu and Q. X. Wang, "Particle Swarm Optimization-based Algorithms for TSP and Generalised TSP", Information Processing Letters, **103(2007)**, 169-176, (2007).

[68] S. M. Akandwanaho, A. O. Adewumi and A. A. Adebiyi, "Solving Dynamic Travelling Salesman Problem using Dynamic Gaussian Process RegressioN", Journal of Applied Mathematics, **2014**, article ID 818529, (2014).

[69] X. Wang, A. Mu and S. Zhu, "A New Way to Solve Travelling Salesman Problem", Intelligent Control and Automation, **4**, 122-125, (2013).

[70] M. Dorigo, V. Manniezzo and A. Colorni, "Ant System Optimization by a Colony of Cooperating Agents", IEEE Trans on Systems Man, and Cybernetics- Part B: Cybernetics, **26** 29-41, (1996).

[71] M. Hahsler and K. Hornik, "TSP Infrastructure for the Travelling Salesperson Problem", Journal of Statistical Software, **23(2)**, 1-21, (2007). URL http://www.jstatsoft.org/v23/i02/.

[72] J. B. Williams, M. Louis, R. Christine and A. Nalepal, "Cockroaches Ecology, Behaviour and Natural History", Johns Hopkins University Press Baltimore, 2007.

[73] M. Lihoreau, J. Costa and C. Rivault, "The Social Biology of Domiciliary Cockroaches: Colony Structure, Kin Recognition and Collective Decision" International Union for the Study of Social Insect, Published Online, (2012).

[74] H. Hoell, J. Doyenand and A. Purcell, "Introduction to Insect Biology and Diversity", Oxford University Press, second edition, (1998).

[75] C. Marion, "Cockroaches", London Reaktion Books Limited, (2003).

[76] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour and F. Tache, "Social Integration of Robots into Groups of Cockroaches to Control Self-organized choices", Science, **318**,(2007).

[77] R. Jeanson, et al. "Self-organized Aggregation in Cockroaches", Animal Behaviour,**69**, 169-180, (2005).

[78] J. Ame, J. Halloy, C. Rivault, C. Detrain, and J. Deneubourg, "Collegial Decision Making Based on Social Amplification Leads to Optimal Group Formation", Proceedings Natl. Acad. Sci., **103(15)**, 5835-5840, (2006).

[79] H. Watanabe and M. Mizunami, "Pavolv's Ccockroach: Classical Conditioning of Salivation in an Insect", PLoS ONE, **2(6)**, e529, (2007).

[80] S. Garnier, et al, "Collective Decision-making by a Group of Cockroach-like Robots", Proceedings of 2005 Swarm Intelligence Symposium, 233-240, (2005).

[81] D. Miller and P. Koehler, "Trail-Following Behaviour in the German Cockroach", Entomological Society of America, (2000).

[82] M. Lihoreau, J. Deneubourg and C. Rivault, "Collective Foraging Decision in a Gregarious Insect", Springer-Verlag, Behaviour Ecology Sociobiology, **64**, 1577-1587, (2010).

[83] Cockroach infestation (2014, January 10th). Retrieved from http://www.cindysbeentrippin.com/2014/01/10/foster-farms-cockroach-infestation-closes-poultry-plant/

[84] D. Tallamy and T. Wood, "Convergence patterns in social insects", An Annual Review of Entomology, **31**, 369-390, (1986).

[85] S. Ishii and Y. Kuwahara, "An Aggregation of the German Cockroach Blattella Germanica L (Orthoptera: Blattellidae)", Applied Entomology and Zoology, **2**, 203–217, (1967).

[86] S. Ishii and Y. Kuwahara, "Aggregation of the German cockroach Blattella Germanica" Experientia, **24**, 88-89, (1968)

[87] D. Kavanaugh, "An Example of Aggregation in Scaphinotus Subgenus Brennus Mintschulsky (Coleoptera: Crabidae:Cychirini)", The PAN-Pacific Entomologist, **53**, 27-31, (1977).

[88] L. Edmund, "Observations on the Biology and Life History of the Brown Cockroach Periplaneta Brunnea Burnmeister", Proceedings of the Entomological Society of Washington, **59**, 283-286, (1957).

[89] D. Mock, T. Lamey and D. Thompson, "Falsifiability and the Information Center Hypothesis", Ornis Scandinavia, **19**, 231-248, (1988).

[90] E. Willeyto, G. Boush and L. Gawin, " Function of Cockroach Aggregation Behaviour", Enviromental Enmotology, **13**, 1557-1560, (1984).

[91] C. Rivault and A. Cloarec, "Exploitation of Food Resources by the Cockroach Blattella Germanica in an Urban Habitat", Entomologia Experimentalis et Applicata, **61**, 149-158, (1991).

[92] D. Rierson, "Baits for German Cockroach Control. In understanding and Controlling the German Cockroach", New York: Oxford University Press, (1995).

[93] Y. Hendrawan and H. Murase, "Neural-discrete Hungry Roach Infestation Optimization to Select Informative Textural Features for Determining Water Content of Cultured Sunagoke Moss", Environment Control in Biology, **49 (1)**, 1-21, (2011).

[94] C. ZhaoHui, " A modified Cockroach Swarm Optimization", SciVerse ScienceDirect, Elsevier Ltd, (2011).

[95] L. Cheng, Z. Wang, S. Yanhong and A. Guo, "Cockroach Swarm Optimization Algorithm for TSP", Advanced Engineering Forum, **1**, 226-229, (2011).

[96] C. ZhaoHui and T. HaiYan, "Cockroach swarm optimization for vehicle Routing Problems", SciVerse ScienceDirect, Energy Procedia, **13**, 30-35, (2011).

[97] S. -J. Wu and C. -T. Wu, "Computational Optimization for S-type Biology Systems: Cockroach Genetic Algorithm", Mathematical Biosciences, **245(2013)**, 299-313, (2013).

[98] M. Clerc and J. Kennedy, "The particle swarm–explosion, stability, and convergence in a multidimensional complex space." *IEEE trans. evolution computation*, **6(1)**, 58–73, (2002).

[99] Y. Shi, "Particle Swarm Optimization", IEEE neural networks society, 8-13, (2004).

[100] R. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization", IEEE In proceedings of the Congress on Evolutionary Computing, 84-89, (2000).

[101] C. Yan, B. Guo and X. Wu, "Empirical Study of the Inertia Weight Particle Swarm Optimization with Constrained Factor", International Journal of Soft Computing and Software Engineering, **2(2)**, (2012).

[102] X. Cai and Z. Cui and Y. Tan, "Stochastic Dynamic Step Length Particle Swarm Optimization", Proceedings of the fourth IEEE International Conference on Innovative Computing, Information and Control, (2009).

[103] M. Laumanns, G. Rudolph and H. P. Schwefel, "A Spatial Predator-prey Approach to Multi-objective Optimization: A Preliminary Study", In Proceedings of the Parallel Problem Solving from Nature, **5**, 241-249, (1998).

[104] S. Chowdhury, G. Dulikravich, and R. Moral, "Modified Predator-prey (MPP) Algorithm for Constrained Multi-objective Optimization", Proceedings of EUROGEN, (2009).

[105] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and Elitist Multi-objective Genetic Algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, **6(2)**, 182-197, (2002).

[106] X. Li, "A Real-coded Predator-prey Genetic Algorithm for Multi-objective Optimization", Lecture Notes in Computer Science, **2632**, 69, (2003).

[107] C. Grimme and K. Schmitt, "Inside a Predator-prey Model for Multi Objective Optimization A Second Study", In Proc. Genetic and Evolutionary Computation Conf.(GECCO 2006), Seattle WA, ACM Press, New York, 707-714, (2006).

[108] A. Silva, E. Neves and E. Costa, "An Empirical Comparison of Particle Swarm and Predator-prey Optimization", Lecture Notes in Computer Science, **2464**, 103-110.

[109] S. Chowdhury, R.J. Moral and G.S. Dulikravich, "Predator-prey Evolutionary Multi-objective Optimization Algorithm: Performance and Improvements", 7th ASMO-UK/ISSMO International Conference on Engineering Design Optimization, Bath, UK, (2008).

[110] L. J. Eshelman and J. D. Schaffer, "Real Coded Genetic Algorithms and Interval Schemata", In Foundations of Genetic Algorithms 2 (FOGA 2), 187-202, (1993).

[111] K. Deb, " Multi-objective Optimization Using Evolutionary Algorithms, " Chichester,UK, Wiley, (2002).

[112] I. C. Obagbuwa, A. O. Adewumi and A. A. Adebiyi, "Stochastic Constriction Cockroach Swarm Optimization for Multidimensional Space Function Problems", Mathematical Problems in Engineering, **2014**, Article ID 430949, (2014).

[113] M. Kerckhove, “From Population Dynamics to Partial Differential Equations”, The Mathematical Journal, **14**, (2012).

[114] I. C Obagbuwa and Adewumi A.O. “An Improved Cockroach Swarm Optimization”, The scientific world Journal, **2014**, Article ID 375358, (2014).

[115] A. Kendall Atkinson, “ An Introduction to Numerical Analysis (2nd ed.)”, New York: John Wiley and Sons, ISBN 978-0-471-50023-0, (1989).

[116] U. M. Ascher and L. Petzold, “Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations”, SIAM, Philadelphia, (1998).

[117] L. Chambers, “The Practical Handbook of Genetic Algorithms, Application”, Chapman and Hall/CRC, second edition, Chapter one, (2001).

[118] Z. Ye, Z. Li and M. Xie, “Some Improvements on Adaptive Genetic Algorithm for Reliability Related Applications”, Reliability Engineering and System Safety, **95(2010)**, 120-126, (2010).

[119] P. S. Oliveto and C. Witt, “On the Run-time Analysis of the Simple Genetic Algorithm”, Theoretical Computer Science, (2013).

[120] D. Ortiz-Boyer, C. Hervs-Martnez, and N. Garca-Pedrajas, “CIXL2: A Crossover Operator for Evolutionary Algorithms Based on Population Features”, Journal of Artificial Intelligence Res.(JAIR), **24**, 1-48, (2005).

[121] J. Michalewicz, “Genetic Algorithms + Data Structures = Evolutionary Programs”, Berlin, Springer-Verlag, (1992).

[122] J. Krause, J. Cordeiro, R.S. Parpinelli and H. S. Lopes, “A survey of Swarm Algorithms Applied to Discrete Optimization Problems”, Swarm Intelligence and Bio-inspired Computation: Theory and Applications. Elsevier Science and Technology Books, 169-191, (2013).

[123] G. Reinelt, “Tsplib Discrete and Combinatorial Optimization, https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/, (1995).

[124] J. Andre, P. Siarry and T. Dognon, “An Improvement of the Standard Genetic Algorithm Fighting Premature Convergence in Continuous Optimization”, Advances in Engineering Software, **32**, 49-60, (2001).

[125] B. A. Sawyer, "Hybrid Real Coded Genetic Algorithms with Pattern Search and Projection", PhD Thesis Submitted to Department of Computer Science, Faculty of Science, University of Lagos, (2010).

[126] J. S. Arora, "Guide to Structural Optimization", ASCE Manuals and Reports on Engineering Practice No. 90, American Society of Civil Engineers, New York, (1997)

[127] J. S. Arora, "Computational Design Optimization: A Review and Future Directions", Structural Safety 7, 131-148, (1990).

[128] C. Khompatraporn, Z. B. Zabinsky Z. B. and J. Pinter, "Comparative Assessment of Algorithms and Software for Global Optimization", Journal of Global Optimization, **31**, 613-633, (2005).

[129] A. Auger and N. Hansen, "Performance Evaluation of an Advanced Local Search Evolutionary Algorithm", D. Corne, Z. Michalewicz, B. McKay, G. Eiben, D. Fogel, C. Fonseca, G. Greenwood, G. Raidl, K. C. Tan and A. Zalzala (Eds), Proceedings of the 2005 IEEE Congress on Evolutionary Computation, IEEE Press, Edinburgh, Scotland, UK, **2**, 1777-1784, (2005).

[130] A. K. Qin , V. L. Huang and P. N. Suganthan, "Differential Evolution Algorithm with Strategy Adaptation for Global Optimization Numerical Optimization", IEEE, Transactions on Evolutionary Computation, **13(2)**, (2009).

[131] C. Grosan and A. Abraham, "A Novel Global Optimization Technique for High Dimensional Functions", International Journal of Intelligent systems, **24**, 421-440, (2009).

[132] J. Cohen, "Statistical Power Analysis for the Behavioral Science", Hills-dale, New Jersey, Hove and London, second edition, (1988).

[133] J. Cohen, "Statistical Power Analysis for the Behavioral sSience", Current Direction in Psychological Science, **1(3)**, 98-101, (1992).

[134] I. C. Obagbuwa, A. O. Adewumi and A. A. Adebiyi, "A Dynamic Step-Size Adaptation Roach Infestation Optimization", In Proceedings of IEEE International conference on advance computing (IACC 2014), Gurgaon, Indian, 21-22, (2014).

[135] I. C. Obagbuwa and A. O. Adewumi, "Modified Roach Infestation Optimization", IEEE conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2014), Hilton Hawaiian village, Honolulu, Hawaii, USA, (2014).

[136] P. N. Suganthan et al., "Problem Definitions and Evaluation Criteria for the CEC 2005, Special Session on Real-parameter Optimization", Technical Report, Nanyang Technology University, Singapore and KanGAL Report Number 2005005(Kanpur Genetic Algorithms Laboratory, IIT Kanpur), (2005).

[137] J. J. Liang, P. N. Suganthan and K. Deb, "Novel Comparison Test Functions for Numerical Global Optimization", IEEE Swarm Intelligence Symposium, 68-75, (2005).

[138] Z. H. Ahmed, "The Ordered Clustered Travelling Salesman Problem: A Hybrid Genetic Algorithm", The Scientific World Journal, **2014**, Article ID 258207, (2014).

[139] M. Ali, C. Khompatraporn and Z. Zabinsky, "A Numerical Evaluation of Several Stochastic Algorithm on Selected Continuous Global Optimization Test Problems", Journal of global optimization, **31**, 635-672 (2005).

[140] D. Karaboga and B. Akay, "A Comparative Study of Artificial Bee Colony Algorithm", Applied mathematics and computation, Elsevier Inc, **214**, 108-132, (2009).

[141] M. Jamil and X-S. Yang, "A Literature Survey of Benchmark Functions for Global Optimization Problems", International Journal of Mathematical Modeling and Numerical Optimization, **4(2)**, 150-194 (2013).

Appendices

Appendix A: Standard Global Optimization Benchmark Test Functions

A large test suite that include a wide variety of problems such as unimodal, multimodal, regular, irregular, separable, non-separable, and multidimensional are presented in this Appendix I. Some of these problems can be found in optimization books, research articles or different optimization websites. A collection of these problems can be found [139, 140, 141]. These are minimization problems for unconstrained optimization. These optimization problems were used to validate the performance of the improved cockroach optimization algorithms. The dimension, problem domain size and the optimal solution are denoted by D, $Lb \leq xi \leq Ub$ and $f(x^*) = f(x_1, ..., x_n)$ respectively. Lb and Ub denote lower and upper bounds of the variables respectively. It should be noted that in several cases the optimal point vector and corresponding global point are known only as a numerical approximation.

1. Aluffi-Pentini's Problem

$$\min_x f(x) = \quad 0.25{x_1}^4 - 0.5{x_1}^2 + 0.1x_1 + 0.5{x_2}^2 \tag{1}$$

$$\text{subject to} \qquad -10 \leq x_1, x_2 \leq 10. \tag{2}$$

The function has two local minimal, one of them is global with $f(x^*) \approx -0.3523$ located at $(-1.0465, 0)$.

2. Becker and Lago Problem

$$\min_x f(x) = \quad (|x_1| - 5)^2 + (|x_2| - 5)^2 \tag{3}$$

$$\text{subject to} \qquad -10 \leq x_1, x_2 \leq 10. \tag{4}$$

The function has four minima located at $(\pm 5, \pm 5)$, all with $f(x^*) = 0$.

3. Bohachevsky 1 Problem

$$\min_x f(x) = \quad x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \tag{5}$$

$$\text{subject to} \qquad -50 \leq x_1, x_2 \leq 50. \tag{6}$$

The number of local minimal is unknown but the global minimizer is located at $x^* = (0, 0)$ with $f(x^*) = 0$.

4. Bohachevsky 2 Problem

$$\min_x f(x) = \quad x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3 \tag{7}$$
$$\text{subject to} \qquad -50 \le x_1, x_2 \le 50. \tag{8}$$

The number of local minimal is unknown but the global minimizer is located at $x^* = (0,0)$ with $f(x^*) = 0$.

5. Branin Problem

$$\min_x f(x) = \quad a(x_2 - b{x_1}^2 + cx_1 - d)^2 + g(1-h)\cos(x_1) + g\,, \tag{9}$$
$$\text{subject to} \qquad -5 \le x_1 \le 10, 0 \le x_2 \le 15, \tag{10}$$

where $a = 1$, $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $d = 6$, $g = 10$, $h = 1/(8\pi)$. There are three minima, all global, in this region. The minimizers are
$x^* \approx (-\pi, 12.275), (\pi, 2.275), (3\pi, 2.475)$ with $f(x^*) = 5/(4\pi)$.

6. Camel Back–3 Three Hump Problem

$$\min_x f(x) = \quad 2x_1^2 - 1.05x_1^4 + \tfrac{1}{6}x_1^6 + x_1x_2 + x_2^2 \tag{11}$$
$$\text{subject to} \qquad -5 \le x_1, x_2 \le 5. \tag{12}$$

The function has three local minima, one of them is global located at $x^* = (0,0)$ with $f(x^*) = 0$.

7. Camel Back–6 Six Hump Problem

$$\min_x f(x) = \quad 4x_1^2 - 2.1x_1^4 + \tfrac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \tag{13}$$
$$\text{subject to} \qquad -5 \le x_1, x_2 \le 5. \tag{14}$$

This function is symmetric about the origin and has three conjugate pairs of local minima with values $f \approx -1.0316, -0.2154, 2.1042$. The function has two global minima at $x^* \approx (0.089842, -0.712656)$ and $(-0.089842, 0.712656)$ with $f(x^*) \approx -1.0316$.

8. Cosine Mixture Problem

$$\max_x f(x) = \quad 0.1\sum_{i=1}^n \cos(5\pi x_i) - \sum_{i=1}^n x_i^2 \tag{15}$$
$$\text{subject to} \quad -1 \le x_i \le 1, i \in \{1, 2, \ldots, n\}. \tag{16}$$

The global maxima are located at the origin with the function values 0.20 and 0.40 for $n = 2$ and $n = 4$, respectively.

9. Dekkers and Aarts Problem

$$\min_x f(x) = \quad 10^5 x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5}(x_1^2 + x_2^2)^4 \tag{17}$$

$$\text{subject to} \qquad -20 \le x_1, x_2 \le 20. \tag{18}$$

The origin is a local minimizer, but there are two global minimizer located at $x^* = (0, 15)$ and $(0, -15)$ with $f(x^*) = -24776.518$.

10. Easom Problem (EP)

$$\min_x f(x) = \quad -\cos(x_1)\cos(x_2)\exp\left(-(x_1-\pi)^2 - (x_2-\pi)^2\right) \tag{19}$$

$$\text{subject to} \qquad -10 \le x_1, x_2 \le 10. \tag{20}$$

The minimum value is located at (π, π) with $f(x^*) = -1$. The function value rapidly approaches zero, when away from (π, π).

11. Goldstein and Price

$$\min_x f(x) = \quad \left[1 + (x_1 + x_2 + 1)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \tag{21}$$

$$\times \left[30 + (2x_1 - 3x_2)^2 \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$$

$$\text{subject to} \qquad -2 \le x_1, x_2 \le 2. \tag{22}$$

There are four local minima and the global minima is located at $x^* = (0, -1)$, with $f(x^*) = 3$.

12. Hosaki Problem

$$\min_x f(x_1, x_2) = \quad \left(1 - 8x_1 + 7x_1^2 - \tfrac{7}{3}x_1^3 + \tfrac{1}{4}x_1^4\right) x_2^2 \exp(-x_2) \tag{23}$$

$$\text{subject to} \qquad 0 \le x_1 \le 5\,, 0 \le x_2 \le 6. \tag{24}$$

There are two minima of which the global minimum is $f(x^*) \approx -2.3458$ with $x^* = (4, 2)$.

13. McCormick Problem

$$\min_x f(x) = \quad \sin(x_1 + x_2) + (x_1 - x_2)^2 - (3/2)x_1 + (5/2)x_2 + 1 \tag{25}$$

$$\text{subject to} \qquad -1.5 \le x_1 \le 4, -3 \le x_2 \le 3. \tag{26}$$

This problem has a local minimum at $(2.59, 1.59)$ and a global minimum at $(-0.547, -1.547)$ with $f(x^*) \approx -1.9133$.

14. Modified Rosenbrock Problem

$$\min_x f(x) = \quad 100(x_2 - {x_1}^2)^2 + \left[6.4(x_2 - 0.5)^2 - x_1 - 0.6\right]^2 \tag{27}$$

$$\text{subject to} \qquad -5 \le x_1, x_2 \le 5. \tag{28}$$

This function has two global minima each with $f(x^*) = 0$ (corresponding to the intersection of two parabolas) and a local minimum (where the parabolas approach without intersection). The global minima are located at $x^* \approx (0.3412, 0.1164), (1,1)$.

15. Multi-Gaussian Problem

$$\max_x f(x) = \quad \sum_{i=1}^{5} a_i \exp\left(-((x_1 - b_i)^2 + (x_2 - c_i)^2)/{d_i}^2\right) \tag{29}$$

$$\text{subject to} \qquad -2 \le x_1, x_2 \le 2. \tag{30}$$

The function has one global maximum at $x^* \approx (-0.01356, -0.01356)$ with $f(x^*) \approx 1.29695$. There are also 4 other local maxima and a saddle point. Values for the parameters a_i, b_i, c_i, and d_i are given in Table 1.

Table 1: Appendix: Data for Multi-Gaussian problem

i	a_i	b_i	c_i	d_i
1	0.5	0.0	0.0	0.1
2	1.2	1.0	0.0	0.5
3	1.0	0.0	-0.5	0.5
4	1.0	-0.5	0.0	0.5
5	1.2	0.0	1.0	0.5

16. Periodic Problem

$$\min_x f(x) = \quad 1 + \sin^2 x_1 + \sin^2 x_2 - 0.1 \exp(-{x_1}^2 - {x_2}^2) \tag{31}$$

$$\text{subject to} \qquad -10 \le x_1, x_2 \le 10. \tag{32}$$

There are 49 local minima all with minimum values 1 and global minimum located at $x^* = (0,0)$ with $f(x^*) = 0.9$.

17. Schaffer 1 Problem

$$\min_x f(x) = \quad 0.5 + \frac{\left(\sin\sqrt{x_1^2 + x_2^2}\right)^2 - 0.5}{\left(1 + 0.001\left(x_1^2 + x_2^2\right)\right)^2} \tag{33}$$

$$\text{subject to} \qquad -100 \le x_1, x_2 \le 100. \tag{34}$$

The number of local minima is not known, but the global minimum is located at $x^* = (0,0)$ with $f(x^*) = 0$.

18. Schaffer 2 Problem

$$\min_x f(x) = \quad (x_1^2 + x_2^2)^{0.25} \left(\sin^2\left(50(x_1^2 + x_2^2)^{0.1}\right) + 1\right) \tag{35}$$

$$\text{subject to} \qquad -100 \leq x_1, x_2 \leq 100. \tag{36}$$

The number of local minima is not known, but the global minimum is located at $x^* = (0,0)$ with $f(x^*) = 0$.

19. Schubert Problem

$$\min_x f(x) = \quad \prod_{i=1}^{n} \left(\sum_{j=1}^{5} j\cos\left((j+1)x_i + j\right)\right) \tag{37}$$

$$\text{subject to} \quad -10 \leq x_i \leq 10, i \in \{1,2,\ldots,n\}. \tag{38}$$

Our tests were performed with $n = 2$. The number of local minima for this problem (given n) is not known but for $n = 2$, the function has 760 local minima, 18 of which are global with $f(x^*) \approx -186.7309$. All two dimensional global minimizers are listed in Table 2:

Table 2: Global optimizers for Schubert problem

x^*				
(-7.0835,4.8580),	(-7.0835,-7.7083),	(-1.4251,-7.0835),	(5.4828,4.8580),	(-1.4251,-0.8003),
(4.8580,5.4828),	(-7.7083,-7.0835),	(-7.0835,-1.4251),	(-7.7083,-0.8003),	(-7.7083,5.4828),
(-0.8003,-7.7083),	(-0.8003,-1.4251),	(-0.8003,4.8580),	(-1.4251,5.4828),	(5.4828,-7.7083),
(4.8580,-7.0835),	(5.4828,-1.4251),	(4.8580,-0.8003)		

20. Gulf Research Problem

$$\min_x f(x) = \qquad \sum_{i=1}^{99} \left[\exp\left(-\frac{(u_i - x_2)^{x_3}}{x_1}\right) - 0.01 \times i\right]^2, \tag{39}$$

$$\text{subject to} \quad 0.1 \leq x_1 \leq 100, 0 \leq x_2 \leq 25.6, \text{ and } 0 \leq x_3 \leq 5, \tag{40}$$

where $u_i = 25 + [-50\ln(0.01 \times i)]^{1/1.5}$ This problem has a global minimizer at $(50, 25, 1.5)$ with $f(x^*) = 0$.

21. Hartman 3 Problem

$$\min_x f(x) = \quad -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right] \tag{41}$$

$$\text{subject to} \qquad 0 \le x_j \le 1, j \in \{1,2,3\} \tag{42}$$

with constants a_{ij}, p_{ij} and c_i given in Table 3. There are four local minima, $x_{local} \approx (p_{i1}, p_{i2}, p_{i3})$ with $f(x_{local}) \approx -c_i$. The global minimum is located at $x^* \approx (0.114614, 0.555649, 0.852547)$ with $f(x^*) \approx -3.862782$.

Table 3: Data for Hartman 3 problem

i	c_i	a_{ij}			p_{ij}		
		$j=1$	2	3	$j=1$	2	3
1	1.0	3.0	10	30	0.36890	0.1170	0.2673
2	1.2	0.1	10	35	0.46990	0.4387	0.7470
3	3.0	3.0	10	30	0.10910	0.8732	0.5547
4	3.2	0.1	10	35	0.03815	0.5743	0.8828

22. Helical Valley Problem

$$\min_x f(x) = \quad 100\left[(x_2 - 10\theta)^2 + (\sqrt{(x_1^2 + x_2^2)} - 1)^2\right] + x_3^2 \tag{43}$$

$$\text{subject to} \qquad -10 \le x_1, x_2, x_3 \le 10 \tag{44}$$

where

$$\theta = \begin{cases} \frac{1}{2\pi}\tan^{-1}\frac{x_2}{x_1}, & \text{if} x_1 \ge 0 \\ \frac{1}{2\pi}\tan^{-1}\frac{x_2}{x_1} + \frac{1}{2}, & \text{if} x_1 < 0 \end{cases} \tag{45}$$

This is a steep-sided valley which follows a helical path. The minimum is located at $x^* = (1,0,0)$ with $f(x^*) = 0$.

23. Levy and Montalvo 1 Problem (LM1)

$$\min_x f(x) = \quad \frac{\pi}{n}\left(10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right]\right) \tag{46}$$

$$+\frac{\pi}{n}(y_n - 1)^2$$

$$\text{subject to} \qquad -10 \le x_i \le 10, i \in \{1,2,\ldots,n\} \tag{47}$$

where $y_i = 1 + \frac{1}{4}(x_i + 1)$. There are approximately 5^n local minima and the global minimum is known to be $f(x^*) = 0$ with $x^* = (-1,-1,\ldots,-1)$. Our tests were performed with $n = 3$.

24. Meyer and Roth Problem (MR)

$$\min_x f(x) = \quad \sum_{i=1}^{5} \left(\frac{x_1 x_3 t_i}{(1+x_1 t_i + x_2 v_i)} - y_i \right)^2 \tag{48}$$

$$\text{subject to} \quad -20 \leq x_i \leq 20, i \in \{1,2,3\}. \tag{49}$$

This is a least squares problem with minimum value $f(x^*) \approx 0.4 \times 10^{-4}$ located at $x^* \approx (3.13, 15.16, 0.78)$. Table 4 lists the parameter values of this problem.

Table 4: Data for Meyer and Roth problem

i	t_i	v_i	y_i
1	1.0	1.0	0.126
2	2.0	1.0	0.219
3	1.0	2.0	0.076
4	2.0	2.0	0.126
5	0.1	0.0	0.186

25. Cosine Mixture Problem

Same as problem number 8, but tests were performed with $n = 4$.

26. Kowalik Problem (KL)

$$\min_x f(x) = \quad \sum_{i=1}^{11} \left(a_i - \frac{x_1(1+x_2 b_i)}{(1+x_3 b_i + x_4 b_i^2)} \right)^2 \tag{50}$$

$$\text{subject to} \quad 0 \leq x_i \leq 0.42, i \in \{1,2,3,4\}. \tag{51}$$

The values for a_i and b_i are given in Table 5:

Table 5: Data for Kowalik problem

i	1	2	3	4	5	6	7	8	9	10	11
a_i	0.1957	0.1947	0.1735	0.16	0.0844	0.0627	0.0456	0.0342	0.0323	0.0235	0.0246
b_i	0.25	0.50	1.0	2.0	4.0	6.0	8.0	10.0	12.0	14.0	16.0

This is a least squares problem with a global optimal value $f(x^*) \approx 3.0748 \times 10^{-4}$ located at $x^* \approx (0.192, 0.190, 0.123, 0.135)$.

27. Miele and Cantrell Problem

$$\min_x f(x) = \quad (\exp(x_1) - x_2)^4 + 100(x_2 - x_3)^6 + (\tan(x_3 - x_4))^4 + {x_1}^8 \tag{52}$$
$$\text{subject to} \qquad -1 \leq x_i \leq 1, i \in \{1,2,3,4\}. \tag{53}$$

The number of local minima is unknown but the global minimizer is located at $x^* = (0,1,1,1)$ with $f(x^*) = 0$.

28. Neumaier 2 Problem

$$\min_x f(x) = \quad \sum_{k=1}^{n} \left(b_k - \sum_{i=1}^{n} {x_i}^k\right)^2 \tag{54}$$
$$\text{subject to} \quad 0 \leq x_i \leq n, i \in \{1,2,\ldots,n\}. \tag{55}$$

We consider a case when $n = 4$ and $b = (8,18,44,114)$. The global minimum is $f(1,2,2,3) = 0$.

29. Powell's Quadratic Problem (PWQ)

$$\min_x f(x) = \quad (x_1 + 10x_1)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 \tag{56}$$
$$\text{subject to} \qquad -10 \leq x_i \leq 10, i \in \{1,2,3,4\}. \tag{57}$$

This is a unimodal function with $f(x^*) = 0$, $x^* = (0,0,0,0)$. The minimizer is difficult to obtain with accuracy as the Hessian matrix at the optimum is singular.

30. Shekel 5 Problem

$$\min_x f(x) = \quad -\sum_{i=1}^{5} \frac{1}{\sum_{j=1}^{4} \left(x_j - a_{ij}\right)^2 + c_i} \tag{58}$$
$$\text{subject to} \quad 0 \leq x_j \leq 10, j \in \{1,2,3,4\}, \tag{59}$$

with constants a_{ij} and c_j given in Table 6 below. There are five local minima and the global minimizer is located at $x^* = (4.00,4.00,4.00,4.00)$ with $f(x^*) \approx -10.1532$.

31. Shekel 7 Problem

$$\min_x f(x) = \quad -\sum_{j=1}^{7} \frac{1}{\sum_{i=1}^{4} \left(x_j - a_{ij}\right)^2 + c_i} \tag{60}$$
$$\text{subject to} \quad 0 \leq x_j \leq 10, j \in \{1,2,3,4\}, \tag{61}$$

Table 6: Data for Shekel problem family

	i	a_{ij}				c_i
		$j=1$	2	3	4	
S5	1	4	4	4	4	0.1
	2	1	1	1	1	0.2
	3	8	8	8	8	0.2
	4	6	6	6	6	0.4
	5	3	7	3	7	0.4
S7	6	2	9	2	9	0.6
	7	5	5	3	3	0.3
S10	8	8	1	8	1	0.7
	9	6	2	6	2	0.5
	10	7	3.6	7	3.6	0.5

with constants a_{ij} and c_j given in Table 6. There are seven local minima and the global minimizer is located at $x^* = (4.00, 4.00, 4.00, 4.00)$ with $f(x^*) \approx -10.4029$.

32. Shekel 10 Problem

$$\min_x f(x) = \quad -\sum_{j=1}^{10} \frac{1}{\sum_{i=1}^{4}(x_j - a_{ij})^2 + c_i} \tag{62}$$

$$\text{subject to} \quad 0 \le x_j \le 10, j \in \{1,2,3,4\} \tag{63}$$

with constants a_{ij} and c_j given in Table 6. There are 10 local minima and the global minimizer is located at $x^* = (4.00, 4.00, 4.00, 4.00)$ with $f(x^*) \approx -10.5364$.

33. Wood's Problem

$$\min_x f(x) = \quad 100(x_2 - x_1^2)^2 + (1-x_1)^2 + 90(x_4 - x_3^2)^2 + (1-x_3)^2 \tag{64}$$

$$+10.1[(x_2-1)^2 + (x_4-1)^2] + 19.8(x_2-1)(x_4-1)$$

$$\text{subject to} \quad -10 \le x_i \le 10, i \in \{1,2,3,4\}. \tag{65}$$

The function has a saddle near $(1,1,1,1)$. The only minimum is located at $x^* = (1,1,1,1)$ with $f(x^*) = 0$.

34. Levy and Montalvo 2 Problem

$$\min_x f(x) = \quad 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1}] \tag{66}$$
$$+(x_n-1)^2[1+\sin^2(2\pi x_n)])$$
$$\text{subject to} \qquad -5 \le x_i \le 5, i \in \{1,2,\ldots,n\}. \tag{67}$$

There are approximately 15^n minima and the global minimizer is known to be $x^* = (1,1,\ldots,1)$ with $f(x^*) = 0$. Our tests were performed with $n = 5$ and 10.

35. Salomon Problem

$$\min_x f(x) = \quad 1 - \cos(2\pi\|x\|) + 0.1\|x\| \tag{68}$$
$$\text{subject to} \qquad -100 \le x_i \le 100 \tag{69}$$

where $\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$. The number of local minima (as a function of n) is not known, but the global minimizer is located at $x^* = (0,0,0,\ldots,0)$ with $f(x^*) = 0$. Our tests were performed with $n = 5$ and $n = 10$.

36. Shekel's Foxholes

$$\min_x f(x) = \qquad -\sum_{j=1}^{30} \frac{1}{c_j + \sum_{i=1}^n (x_i - a_{ji})^2} \tag{70}$$
$$\text{subject to} \quad 0 \le x_i \le 10, i \in \{1,2,\ldots,10\}. \tag{71}$$

Our tests were performed with $n = 5$ and 10. The constants c_j and a_{ji} are given in Table 7. The number of local minima is not known, but the global minima are presented in Table 8.

37. Hartman 6 Problem

$$\min_x f(x) = \quad -\sum_{i=1}^4 c_i \exp\left[-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2\right] \tag{72}$$
$$\text{subject to} \qquad -0 \le x_j \le 1, j \in \{1,\ldots,6\}, \tag{73}$$

with constants a_{ij} and c_i given in Table 9 and constants p_{ij} in Table 10. There are four local minima, $x_{local} \approx (p_{i1},\ldots,p_{i6})$ with $f(x_{local}) \approx -c_i$. The global minimum is located at
$x^* \approx (0.201690, 0.150011, 0.476874, 0.275332, 0.311652, 0.657301)$
with $f(x^*) \approx -3.322368$.

Table 7: Data for Shekel's foxholes problem

j	c_j	a_{ji}									
		$i=1$	2	3	4	5	6	7	8	9	10
1	0.806	9.681	0.667	4.783	9.095	3.517	9.325	6.544	0.211	5.122	2.020
2	0.517	9.400	2.041	3.788	7.931	2.882	2.672	3.568	1.284	7.033	7.374
3	0.100	8.025	9.152	5.114	7.621	4.564	4.711	2.996	6.126	0.734	4.982
4	0.908	2.196	0.415	5.649	6.979	9.510	9.166	6.304	6.054	9.377	1.426
5	0.965	8.074	8.777	3.467	1.863	6.708	6.349	4.534	0.276	7.633	1.567
6	0.669	7.650	5.658	0.720	2.764	3.278	5.283	7.474	6.274	1.409	8.208
7	0.524	1.256	3.605	8.623	6.905	4.584	8.133	6.071	6.888	4.187	5.448
8	0.902	8.314	2.261	4.224	1.781	4.124	0.932	8.129	8.658	1.208	5.762
9	0.531	0.226	8.858	1.420	0.945	1.622	4.698	6.228	9.096	0.972	7.637
10	0.876	7.305	2.228	1.242	5.928	9.133	1.826	4.060	5.204	8.713	8.247
11	0.462	0.652	7.027	0.508	4.876	8.807	4.632	5.808	6.937	3.291	7.016
12	0.491	2.699	3.516	5.874	4.119	4.461	7.496	8.817	0.690	6.593	9.789
13	0.463	8.327	3.897	2.017	9.570	9.825	1.150	1.395	3.885	6.354	0.109
14	0.714	2.132	7.006	7.136	2.641	1.882	5.943	7.273	7.691	2.880	0.564
15	0.352	4.707	5.579	4.080	0.581	9.698	8.542	8.077	8.515	9.231	4.670
16	0.869	8.304	7.559	8.567	0.322	7.128	8.392	1.472	8.524	2.277	7.826
17	0.813	8.632	4.409	4.832	5.768	7.050	6.715	1.711	4.323	4.405	4.591
18	0.811	4.887	9.112	0.170	8.967	9.693	9.867	7.508	7.770	8.382	6.740
19	0.828	2.440	6.686	4.299	1.007	7.008	1.427	9.398	8.480	9.950	1.675
20	0.964	6.306	8.583	6.084	1.138	4.350	3.134	7.853	6.061	7.457	2.258
21	0.789	0.652	2.343	1.370	0.821	1.310	1.063	0.689	8.819	8.833	9.070
22	0.360	5.558	1.272	5.756	9.857	2.279	2.764	1.284	1.677	1.244	1.234
23	0.369	3.352	7.549	9.817	9.437	8.687	4.167	2.570	6.540	0.228	0.027
24	0.992	8.798	0.880	2.370	0.168	1.701	3.680	1.231	2.390	2.499	0.064
25	0.332	1.460	8.057	1.336	7.217	7.914	3.615	9.981	9.198	5.292	1.224
26	0.817	0.432	8.645	8.774	0.249	8.081	7.461	4.416	0.652	4.002	4.644
27	0.632	0.679	2.800	5.523	3.049	2.968	7.225	6.730	4.199	9.614	9.229
28	0.883	4.263	1.074	7.286	5.599	8.291	5.200	9.214	8.272	4.398	4.506
29	0.608	9.496	4.830	3.150	8.270	5.079	1.231	5.731	9.494	1.883	9.732
30	0.326	4.138	2.562	2.532	9.661	5.611	5.500	6.886	2.341	9.699	6.500

Table 8: Global optimizers for Shekel's foxholes problem

n	$f(x^*)$	x^*
5	-10.4056	(8.025, 9.152, 5.114, 7.621, 4.564)
10	-10.2088	(8.025, 9.152, 5.114, 7.621, 4.564, 4.771, 2.996, 6.126, 0.734, 4.982)

Table 9: Data for Hartman 6 problem

i	c_i	a_{ij}					
		$j=1$	2	3	4	5	6
1	1.0	10.0	3.0	17.0	3.5	1.7	8.0
2	1.2	0.05	10.0	17.0	0.1	8.0	14.0
3	3.0	3.0	3.5	1.7	10.0	17.0	8.0
4	3.2	17.0	8.0	0.05	10.0	0.1	14.0

Table 10: Data for Hartman 6 problem

i	p_{ij}					
	$j=1$	2	3	4	5	6
1	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
4	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

38. Storn's Tchebychev Problem

$$\min_x f(x) = p_1 + p_2 + p_3, \tag{74}$$

where

$$p_1 = \begin{cases} (u-d)^2 & \text{if } u < d \\ 0 & \text{if } u \geq d \end{cases} \qquad u = \sum_{i=1}^{n} (1.2)^{n-i} x_i$$

$$p_2 = \begin{cases} (v-d)^2 & \text{if } v < d \\ 0 & \text{if } v \geq d \end{cases} \qquad v = \sum_{i=1}^{n} (-1.2)^{n-i} x_i$$

$$p_3 = \sum_{j=0}^{m} \begin{cases} (w_j - 1)^2 & \text{if } w_j > 1 \\ (w_j + 1)^2 & \text{if } w_j < -1 \\ 0 & \text{if } -1 \leq w_j \leq 1 \end{cases} \qquad w_j = \sum_{i=1}^{n} \left(\frac{2j}{m} - 1\right)^{n-i} x_i,$$

for $n = 9$: $x_i \in [-128, 128]^n$, $d = 72.661$, and $m = 60$

for $n = 17$: $x_i \in [-32768, 32768]^n$, $d = 10558.145$, and $m = 100$.

The number of local minima is not known but the global minimum is known to be as shown in Table 11. Our tests were performed with $n = 9$.

Table 11: Global optimizers for Storn's Tchebychev problem

n	$f(x^*)$	x^*
9	0	(128, 0, -256, 0, 160, 0, -32, 0, 1)
17	0	(32768, 0, -1331072, 0, 21299, 0, -180224, 84480, 0, -2154, 0, 2688, 0, -128, 0, 1)

39. Ackley's Problem

$$\min_x f(x) = -20\exp\left(-0.2\sqrt{\tfrac{1}{n}\textstyle\sum_{i=1}^{n} x_i{}^2}\right) - \exp\left(\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\cos(2\pi x_i)\right) \tag{75}$$
$$+20+e$$
$$\text{subject to} \quad -30 \le x_i \le 30, i \in \{1,2,\ldots,n\}. \tag{76}$$

The number of local minima is not known. The global minimum is located at the origin with $f(x^*) = 0$. Tests were performed for $n = 10$.

40. Exponential Problem

$$\max_x f(x) = \exp\left(-0.5\textstyle\sum_{i=1}^{n} x_i{}^2\right) \tag{77}$$
$$\text{subject to} \quad -1 \le x_i \le 1, i \in \{1,2,\ldots,n\}. \tag{78}$$

The optimal value $f(x^*) = 1$ is located at the origin. Our tests were performed with $n = 10$.

41. Griewangk Problem

$$\min_x f(x) = 1 + \tfrac{1}{4000}\textstyle\sum_{i=1}^{n} x_i{}^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) \tag{79}$$
$$\text{subject to} \quad -600 \le x_i \le 600, i \in \{1,2,\ldots,n\}. \tag{80}$$

The function has a global minimum located at $x^* = (0,0,\ldots,0)$ with $f(x^*) = 0$. Number of local minima for arbitrary n is unknown, but in the two dimensional case there are some 500 local minima. Tests were performed for $n = 10$.

42. Levy and Montalvo 2 Problem

Same as problem number 34, but tests were performed with $n = 10$.

43. Langerman 2 Problem

$$\min_x f(x) = \quad -\sum_{j=1}^{5} c_j \cos\left(\pi d_j\right) \exp\left(-d_j/\pi\right), \tag{81}$$

$$\text{subject to} \qquad 0 \le x_i \le 10, i \in \{1,2,\ldots,n\}, \tag{82}$$

where $d_j = \sum_{i=1}^{n} \left(x_i - a_{ji}\right)^2$. The test used $n = 10$. The constants c_j and a_{ji} are given in Table 12. Dimension of problem is 2. The global minimum located at $f(x^*) = -1.08$

Table 12: Data for modified Langerman problem

j	c_j	a_{ji}									
		$i=1$	2	3	4	5	6	7	8	9	10
1	0.806	9.681	0.667	4.783	9.095	3.517	9.325	6.544	0.211	5.122	2.020
2	0.517	9.400	2.041	3.788	7.931	2.882	2.672	3.568	1.284	7.033	7.374
3	0.100	8.025	9.152	5.114	7.621	4.564	4.711	2.996	6.126	0.734	4.982
4	0.908	2.196	0.415	5.649	6.979	9.510	9.166	6.304	6.054	9.377	1.426
5	0.965	8.074	8.777	3.467	1.867	6.708	6.349	4.534	0.276	7.633	1.567

44. Neumaier 3 Problem

$$\min_x f(x) = \qquad \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1} \tag{83}$$

$$\text{subject to} \quad -n^2 \le x_i \le n^2, i \in \{1,2,\ldots,n\}. \tag{84}$$

The case considered here is $n = 10$. The number of local minima is not known, but the global minima can be expressed as:

$$f(x^*) = -\frac{n(n+4)(n-1)}{6}, \quad x_i^* = i(n+1-i).$$

The global minima for some values of n are presented below.

Table 13: Global minima for Neumaier 3 problem

n	10	15	20	25	30
$f(x^*)$	-210	-665	-1520	-2900	-4930

45. Paviani Problem

$$\min_x f(x) = \sum_{i=1}^{10} \left[(\ln(x_i - 2))^2 + (\ln(10 - x_i))^2 \right] - \left(\prod_{i=1}^{10} x_i \right)^{0.2} \tag{85}$$

$$\text{subject to} \quad 2 \leq x_i \leq 10, i \in \{1, 2, \ldots, 10\}. \tag{86}$$

This function has a global minimizer at $x_i^* \approx 9.351$ for all i, with $f(x^*) \approx -45.778$.

46. Rastrigin Problem

$$\min_x f(x) = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) \right] \tag{87}$$

$$\text{subject to} \quad -5.12 \leq x_i \leq 5.12, i \in \{1, 2, \ldots, n\}. \tag{88}$$

The total number of minima for this function is not exactly known but the global minimizer is located at $x^* = (0, 0, \ldots, 0)$ with $f(x^*) = 0$. For $n = 2$, there are about 50 local minimizers arranged in a lattice like configuration. Our tests were performed with $n = 10$.

47. Rosenbrock Problem

$$\min_x f(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right] \tag{89}$$

$$\text{subject to} \quad -30 \leq x_i \leq 30, i \in \{1, 2, \ldots, n\}. \tag{90}$$

Our tests were performed with $n = 10$. This function is known as the extended Rosenbrock function. It is unimodal, yet due to a saddle point it is very difficult to locate the minimizer $x^* = (1, 1, \ldots, 1)$ with $f(x^*) = 0$.

48. Salomon Problem

Same as problem number 35, but tests were performed with $n = 10$.

49. Schwefel Problem

$$\min_x f(x) = -\sum_{i=1}^{n} x_i \sin\left(\sqrt{|x_i|} \right) \tag{91}$$

$$\text{subject to} \quad -500 \leq x_i \leq 500, i \in \{1, 2, \ldots, n\}. \tag{92}$$

The number of local minima for a given n is not known, but the global minimum value $f(x^*) \approx -418.9829n$ is located at $x^* = (s, s, \ldots, s)$, $s \approx 420.97$. Our tests were performed with $n = 10$.

50. Shekel's Foxholes

Same as problem number 36, but tests were performed with $n = 10$.

51. Sinusoidal Problem

$$\min_x f(x) = \quad -[A\prod_{i=1}^{n}\sin(x_i - z) + \prod_{i=1}^{n}\sin(B(x_i - z))] \tag{93}$$

$$\text{subject to} \qquad 0 \leq x_i \leq 180, i \in \{1,2,\ldots,n\}. \tag{94}$$

The variable x is in degrees. Parameter A affects the amplitude of the global optimum; B affects the periodicity and hence the number of local minima; z shifts the location of the global minimum; and n indicates the dimension. Our tests were performed with $A = 2.5, B = 5, z = 30$, and $n = 10$ and 20. The location of the global solution is at $x^* = (90 + z, 90 + z, \ldots, 90 + z)$ with the global optimum value of $f(x^*) = -(A + 1)$. The number of local minima increases dramatically in dimension, and when $B = 5$ the number of local minima is equal to:

$$\sum_{i=0}^{\lfloor n/2 \rfloor} \left(\frac{n!}{(n-2i)!(2i)!} 3^{n-2i} 2^{2i} \right). \tag{95}$$

52. Sphere Problem

$$\min_x f(x) = \qquad \sum_{i=1}^{n} x_i^2 \tag{96}$$

$$\text{subject to} \quad -100 \leq x_i \leq 100, i \in \{1,2,\ldots,n\}. \tag{97}$$

A very simple unimodal function with its global minimum located at $x^* = 0$, with $f(x^*) = 0$. This function has no interaction between its variables.

53. Storn's Tchebychev Problem

Same as problem number 38, but tests were performed with $n = 17$.

54. Sinusoidal Problem

Same as problem number 51, but tests were performed with $n = 20$.

55. Step Problem

$$\min_x f(x) = \qquad \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2 \tag{98}$$

$$\text{subject to} \quad -100 \leq x_i \leq 100, i \in \{1,2,\ldots,n\}. \tag{99}$$

The global minimal is located $x^* = f(0.5, ..., 0.5) = 0$, $f(x^*) = 0$.

56. StepInt Problem

$$\min_x f(x) = 25 + \sum_{i=1}^{5} \lfloor x_i \rfloor \tag{100}$$

$$\text{subject to} \quad -5.12 \leq x_i \leq 5.12, i \in \{1, 2, \ldots, n\}. \tag{101}$$

The global minimal is located $x^* = f(0, ..., 0) = 0$, $f(x^*) = 0$.

57. Sum Squares Problem

$$\min_x f(x) = \sum_{i=1}^{n} i x_i^2 \tag{102}$$

$$\text{subject to} \quad -10 \leq x_i \leq 10, i \in \{1, 2, \ldots, n\}. \tag{103}$$

The global minimal is located $x^* = f(0, ..., 0) = 0$, $f(x^*) = 0$.

58. Trid 6 Problem

$$\min_x f(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1} \tag{104}$$

$$\text{subject to} \quad -6^2 \leq x_i \leq 6^2, i \in \{1, 2, \ldots, n\}. \tag{105}$$

The global minimal is located $f(x^*) = -50$.

59. Trid 10 Problem

$$\min_x f(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1} \tag{106}$$

$$\text{subject to} \quad -100 \leq x_i \leq 100, i \in \{1, 2, \ldots, n\}. \tag{107}$$

The global minimal is located $f(x^*) = -200$.

60. Zakharov Problem

$$\min_x f(x) = \sum_{i=1}^{n} (x_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4 \tag{108}$$

$$\text{subject to} \quad -5 \leq x_i \leq 10, i \in \{1,2,\ldots,n\}. \tag{109}$$

The global minimal is located $x^* = f(0,...,0) = 0$, $f(x^*) = 0$.

61. Matyas Problem

$$\min_x f(x) = 0.26(x_1 + x_2) - 0.48x_1x_2 \tag{110}$$

$$\text{subject to} \quad -10 \leq x_i \leq 10, i \in \{1,2,\ldots,n\}. \tag{111}$$

The global minimal is located at $x^* = f(0,...,0) = 0$, $f(x^*) = 0$.

62. Epistatic Michalewicz 2 Problem

$$\min_x f(x) = -\sum_{i=1}^{n} sin(y_i)(sin(iy_i^2/\pi))^{2m} \tag{112}$$

$$\text{subject to} \quad 0 \leq x_i \leq \pi, i \in \{1,2,\ldots,n\}. \tag{113}$$

$$y_i = \begin{cases} x_i cos(\theta) - x_{i+1} sin(\theta), i = 1,3,5,...,\leq n \\ x_i sin(\theta) - x_{i+1} cos(\theta), i = 2,4,6,...,\leq n \\ x_i, i = n \end{cases}$$

$\theta = \pi/6, m = 10$ The global minimal is located at $f(x^*) = -1.8013$.

63. Quadric Problem

$$\min_x f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2 \tag{114}$$

$$\text{subject to} \quad -100 \leq x_i \leq 100, i \in \{1,2,\ldots,n\}. \tag{115}$$

The global minimal is located at $x^* = f(0,...,0) = 0$, $f(x^*) = 0$.

64. Chung Reynolds Problem

$$\min_x f(x) = \quad x_1^2 + 2x_2^2 - 0.3cos(3\pi x_1 + 4\pi x_2) + 0.3 \tag{116}$$
$$\text{subject to} \qquad -100 \leq x_i \leq 100, i \in \{1,2,\ldots,n\}. \tag{117}$$

The global minimal is located at $x^* = f(0,...,0) = 0$, $f(x^*) = 0$.

65. Bohachevsky 3 Problem

$$\min_x f(x) = \quad x_1^2 + 2x_2^2 - 0.3cos(3\pi x_1 + 4\pi x_2) + 0.3 \tag{118}$$
$$\text{subject to} \qquad -100 \leq x_i \leq 100, i \in \{1,2,\ldots,n\}. \tag{119}$$

The global minimal is located at $x^* = f(0,...,0) = 0$, $f(x^*) = 0$.

66. Langermann 5 Problem

Same as problem number 43, dimension of problem is 5. The global minimum located at $f(x^*) = -1.5$

67. Langermann 10 Problem

Same as problem number 43 and 66, dimension of problem is 10. The global minimum located at $f(x^*) = NA$

68. Dixcon-Price Problem

$$\min_x f(x) = \qquad (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2 \tag{120}$$
$$\text{subject to} \quad -10 \leq x_i \leq 10, i \in \{1,2,\ldots,n\}. \tag{121}$$

The global minimal is located at $x^* = f(0,...,0) = 0$, $f(x^*) = 0$.

69. Beale Problem

$$\min_x f(x) = \quad (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1 + x_1x_2^2)^2 + \tag{122}$$
$$(2.625 - x_1 + x_1 + x_1x_2^3)^2 \tag{123}$$
$$\text{subject to} \qquad -4.5 \leq x_i \leq 4.5.$$

(125)

The global minimal is located at $f(x^*) = 0$.

70. Epistatic Michalewicz 5 Problem

Same as 62 but dimension is 5. The global minimal is located at $f(x^*) = -4.687658$.

71. Foxholes Problem

$$\begin{aligned} \min_x f(x) = & \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})6} \right]^{-1} \\ \text{subject to} \quad & -65.536 \leq x_i \leq 65.536. \end{aligned} \tag{126}$$

(128)

The global minimal is located at $f(x^*) = 0.998$. Data for a_{ij} is in table 14

72. Booth Problem

$$\min_x f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \tag{129}$$

$$\text{subject to} \quad -10 \leq x_i \leq 10, i \in \{1, 2, \ldots, n\}. \tag{130}$$

The global minimal is located at $x^* = f(0, ..., 0) = 0$, $f(x^*) = 0$.

73. Colville Problem

$$\begin{aligned} \min_x f(x) = & \quad 100(x_1^2 - x_2^2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2 - 1 \\ \text{subject to} & \quad -10 \leq x_i \leq 10, i \in \{1, 2, \ldots, n\}. \end{aligned}$$

The global minimal is located at $x^* = f(0, ..., 0) = 0$, $f(x^*) = 0$.

74. Odd Square Problem

Table 14: a Parameter of FoxHoles Function

j	a_{ij}	$i = 1,2$
1	-32	-32
2	-16	-32
3	0	-32
4	16	-32
5	32	-32
6	-32	-16
7	-16	-16
8	0	-16
9	16	-16
10	32	-16
11	-32	0
12	-16	0
13	0	0
14	16	0
15	32	0
16	-32	16
17	-16	16
18	0	16
19	16	16
20	32	16
21	-32	32
22	-16	32
23	0	32
24	16	32
25	32	32

$$\min_x f(x) = \quad -(1.0+0.2d/(D+0.1))\cos(D\pi)e^{-D/2\pi} \tag{133}$$

$$d = \quad \sqrt{\sum_{i=1}^{n}(x_i-b_i)^2} \tag{134}$$

$$D = \quad sqrtn(max|x_i-b_i|) \tag{135}$$

$$\text{subject to} \quad -15 \le x_i \le 15, i \in \{1,2,\ldots,n\}. \tag{136}$$

The global minimal is located at $x^* = f(0,...,0) = 0$, $f(x^*) = 0$.

75. Epistatic Michalewicz 10 Problem

same as 62 and 70 but dimension is 10. The global minimal is located at $f(x^*) = -9.66152$.

Table 15: Benchmarks Tested with Discrete Valued Cockroach Swarm Optimization Algorithm

	Range	f_{bias}	Properties	Functions	Description
F1	[-32,32]	-140	M,Rt,Sh,N,Sc	Shifted Rotated Ackley	$f_1(x) = -20exp(-0.2\sqrt{\sum_{i=1}^{D}\frac{z_i^2}{D}}) - exp(\sum_{i=1}^{D} cos\frac{2\pi z_i}{D})$ $+20+e+f_{bias}$ z=(x-o)*M
F2	[-5,5]	-330	M,Sh,S,Sc	Shifted Rastrigin	$f_2(x) = \sum_{i=1}^{D}(z_i^2 - 10cos(2piz_i)+10) + f_{bias}$, $z = (x-o)$
F3	[-5,5]	-330	M,Sh,Rt,N,Sc	Shifted Rotated Rastrigin	$f_3(x) = \sum_{i=1}^{D}(z_i^2 - 10cos(2piz_i)+10) +$ $f_{bias}, z = (x-o)*M$
F4	[-0.5,0.5]	90	M,Sh,Rt,N,Sc	Shifted Rotated Weierstrass	$f_4(x) = \sum_{i=1}^{D}(\sum_{k=0}^{kmax}[a^k cos(2\pi b^k(z_i+0.5))])$ $-D\sum_{k=0}^{kmax}[a^k cos(2\pi b^k 0.5)] + f_{bias}$ $a-0.5, b=3, k_{max}=20, z=(x-o)*M$
F5	$[-\pi,\pi]$	-460	M,Sh,N,Sc	Schwefel 213	$f_5(x) = \sum_{i=1}^{D}(A_i - B_i(x))^2 + f_{bias}$ $A_i = \sum_{j=1}^{D}(a_{ij}sin\alpha_j + b_{ij}cos\alpha_j)$, $B_i = \sum_{j=1}^{D}(a_{ij}sinx_j + b_{ij}cosx_j)$ $\alpha = [\alpha_1, \alpha_2, ..., \alpha_D]$, α_j are random number generated from $[-\pi,\pi]$

U= Unimodal, M=Multimodal, S=Separable, N=Non-Separable, Sc=Scalable, Sh=Shifted, Rt=Rotated, Ns=Noise in fitness, D=Dimension, Global optimum $x^* = 0$, $f(x^*) = f_{bias}$, $x = [x_1, x_2, ..., x_D], o = [o_1, o_2, ...o_D]$, M=D*D orthogonal matrix.

Table 16: Benchmark Functions for Stochastic Constriction Cockroach Swarm Optimization for Multidimensional Space Functions

Problems	Range	Minimum
Ackley: $f(X) = +20 + e - 20e^{-0.2\sqrt{\sum_{i=1}^{n} \frac{x_i^2}{n}}} - e^{\sum_{i=1}^{n} cos\frac{2\pi x_i}{n}}$	[-10,10]	0
Levy: $f(x) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 \left[1 + 10\sin^2(\pi y_i + 1)\right] + (y_n - 1)^2(1 + \sin^2(2\pi x_n))$ where $y_i = 1 + \frac{1}{4}(x_i - 1)$	[-10,10]	0
Quadric: $f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	[-10,10]	0
Rastrigin: $f(x) = 10n + \sum_{i=1}^{n} \frac{{x_i}^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-5.12,5.12]	0
Rosenbrock: $f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	[-5,10]	0
Schwefel: $f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	0
Sphere: $f(x) = \sum_{i=1}^{n} {x_i}^2$	[-10,10]	0
Sum Squares: $f(x) = \sum_{i=1}^{n} i x_i^2$	[-10,10]	0
Griewank: $f(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0

Table 17: Benchmark Test Functions for Improved Cockroach Swarm Optimization (ICSO).

No	Range	D	C	Functions	Description
1	[-100,100]	30	US	Step	$f(x)\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$
2	[-100,100]	30	US	Sphere	$f(x)=\sum_{i=1}^{n}x_i{}^2$
3	[-10,10]	30	US	SumSquares	$f(x)=\sum_{i=1}^{n}ix_i^2$
4	[-100,100]	2	MS	Bohachevsky1	$f(x)=x_1^2+2x_2^2-0.3cos(3\pi x_1)-0.4cos(4\pi x_2)+0.7$
5	[-100,100]	2	MN	Bohachevsky2	$f(x)=x_1^2+2x_2^2-0.3cos(3\pi x_1)(4\pi x_2)+0.3$
6	[-100,100]	2	MN	Bohachevsky3	$f(x)=x_1^2+2x_2^2-0.3cos(3\pi x_1+4\pi x_2)+0.3$
7	[0,180]	20	UN	Sinusoidal20	$f(x)=-[A\prod_{i=1}^{n}\sin(x_i-z)+\prod_{i=1}^{n}\sin(B(x_i-z))]$ $A=2.5,B=5,z=30$
8	[-100, 100]	30	UN	Quadric	$f(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2$
9	[-100,100]	2	UN	Easom	$f(x)=-cosx_1cosx_2.exp(-(x_1-\pi)^2)exp(-(x_2-\pi)^2)$
10	[-10,10]	2	UN	Matyas	$f(x)=0.26(x_1+x_2)-0.48x_1x_2$
11	[-5,10]	10	UN	Zakharov	$f(x)=\sum_{i=1}^{n}(x_i)^2+(\sum_{i=1}^{n}0.5ix_i)^2+(\sum_{i=1}^{n}0.5ix_i)^4$
12	[-10,10]	24	UN	Powell	$f(x)=\sum_{i=1}^{n/k}(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+$ $(x_{4i-2}-x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4$
13	[-10,10]	30	UN	Schwefel2.22	$f(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$
14	[-30,30]	30	UN	Rosenbrock	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$
15	[-5.12,5.12]	30	MS	Rastrigin	$f(x)=\sum_{i=1}^{n}x_i{}^2-10cos(2\pi x_i)+10$
16	[-100,100]	2	MN	Schaffer1	$f(x)=0.5+\frac{\sin^2\sqrt{x_1^2+x_2^2}^2-0.5}{[1+0.001(x_1^2+x_2^2)]^2}$
17	[-100, 100]	30	MN	Schaffer2	$f(x)=(x_1^2+x_2^2)^{0.25}\left(\sin^2\left(50(x_1^2+x_2^2)^{0.1}\right)+1\right)$
18	[-600,600]	30	MN	Griewangk	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}cos(\frac{x_i}{\sqrt{i}})+1$
19	[-32,32]	30	MN	Ackley	$f(x)=-20exp^{(}-0.2\sqrt{\sum_{i=1}^{n}\frac{x_i^2}{n}})-exp^{(}\sum_{i=1}^{n}cos\frac{2\pi x_i}{n})+20+e$
20	[-5,5]	2	MN	Three hump camel back	$f(x)=2x_1^2-1.05x_1^4+1/6x_1^6+x_1x_2+x_2^2$
21	[-5, 5]	2	MN	Six Hump Camel Back	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$
22	$[-128,128]^n$	9	UN	Storn's Tchebychev	$f(x)=p_1+p_2+p_3,$ $p_1=\begin{cases}(u-d)^2 & \text{if } u<d\\ 0 & \text{if } u\geq d\end{cases}$ $u=\sum_{i=1}^{n}(1.2)^{n-i}x_i$ $p_2=\begin{cases}(v-d)^2 \text{if } v<d\\ 0 \quad \text{if } v\geq d\end{cases}$ $v=\sum_{i=1}^{n}(-1.2)^{n-i}x_i$
23	$[-32768,32768]^n$	17		Storn's Tchebychev	where $p_3=\sum_{j=0}^{m}\begin{cases}(w_j-1)^2 \\ \text{if } w_j>1\\ (w_j+1)^2 & \text{if } w_j<-1\\ 0 & \text{if } -1\leq w_j\leq 1\end{cases}$ $w_j=\sum_{i=1}^{n}\left(\frac{2j}{m}-1\right)^{n-i}x_i,$ for $n=9$: $d=72.661$, and $m=60$ for $n=17$: $d=10558.145$, and $m=100$.

D:Dimension, C:Characteristic, U:Unimodal, S:Seperable, N:Non-Separable.

Table 18: Benchmark Functions for Dynamic Step-Size Adaptation Roach Infestation Optimization and Modified Roach Infestation Optimization

Problems	Range	Minimum
Sphere:$f(x)=\sum_{i=1}^{n} {x_i}^2$	[-50,50]	$f(0)=0$
Rastrigin:$f(x)=\sum_{i=1}^{n} {x_i}^2-10cos(2\pi x_i)+10$	[-50,50]	$f(0)=0$
Rosenbrock:$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(1-x_i)^2]$	[-50,50]	$f(0)=0$
Ackley:$f(x)=-20exp^{(}-0.2\sqrt{\sum_{i=1}^{n}\frac{x_i^2}{n}})-exp^{(}\sum_{i=1}^{n}cos\frac{2\pi x_i}{n})+20+e$	[-50,50]	$f(0)=0$
Griewank:$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}cos(\frac{x_i}{\sqrt{i}})+1$	[-600,600]	$f(0)=0$
Michalewicz:$f(x)=-\sum_{i=1}^{n}sin(r_i)(sin\frac{ir_i^2}{\pi})^{20}$	[-pi,pi]	$f(0)=-1.8013$
Easom:$f(x_1,x_2)=-cosx_1cosx_2.exp(-(x_1-\pi)^2)exp(-(x_2-\pi)^2)$	[-100,100]	$f(0)=-1$
Hump:$f(x)=A+4r_1^2-2.1r_1^4+r_1^6/3+r_1r_2-4r_2^2+4r_2^4, A=1.0316285$	[-50,50]	$f(0)=0$

Table 19: 70 Global Optimization Test Functions.

No	Range	D	C	Functions	Description
F1	[-600,600]	30	MN	Griewangk	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}cos(\frac{x_i}{\sqrt{i}})+1$
F2	[-100,100]	30	US	Step	$f(x)\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$
F3	[-100,100]	30	US	Sphere	$f(x)=\sum_{i=1}^{n}x_i{}^2$
F4	[-10,10]	30	US	SumSquares	$f(x)=\sum_{i=1}^{n}ix_i^2$
F5	[-15,15]	10	UN	OddSquare	$f(x)=-(1.0+0.2d/(D+0.1))\cos(D\pi)e^{-D/2\pi}$ $d=\sqrt{\sum_{i=1}^{n}(x_i-b_i)^2}$ $D=\sqrt{n}(max\lvert x_i-b_i\rvert)$
F6	[-100,100]	2	UN	Easom	$f(x_1,x_2)=-cosx_1cosx_2.exp(-(x_1-\pi)^2)exp(-(x_2-\pi)^2)$
F7	$[-128,128]^n$	9	UN	Storn's Tchebychev9	$f(x)=p_1+p_2+p_3$, $p_1=\begin{cases}(u-d)^2 & \text{if } u<d\\ 0 & \text{if } u\geq d\end{cases}$ $u=\sum_{i=1}^{n}(1.2)^{n-i}x_i$ $p_2=\begin{cases}(v-d)^2\text{if } v<d\\ 0 \quad \text{if } v\geq d\end{cases}$ $v=\sum_{i=1}^{n}(-1.2)^{n-i}x_i$ where $p_3=\sum_{j=0}^{m}\begin{cases}(w_j-1)^2\\ \text{if } w_j>1\\ (w_j+1)^2 & \text{if } w_j<-1\\ 0 & \text{if } -1\leq w_j\leq 1\end{cases}$ $w_j=\sum_{i=1}^{n}\left(\frac{2j}{m}-1\right)^{n-i}x_i$, for $n=9$: $d=72.661$, and $m=60$
F8	$[-32768,32768]^n$	17	UN	Storn's Tchebychev17	$f(x)=p_1+p_2+p_3$, $p_1=\begin{cases}(u-d)^2 & \text{if } u<d\\ 0 & \text{if } u\geq d\end{cases}$ $u=\sum_{i=1}^{n}(1.2)^{n-i}x_i$ $p_2=\begin{cases}(v-d)^2\text{if } v<d\\ 0 \quad \text{if } v\geq d\end{cases}$ $v=\sum_{i=1}^{n}(-1.2)^{n-i}x_i$ where $p_3=\sum_{j=0}^{m}\begin{cases}(w_j-1)^2\\ \text{if } w_j>1\\ (w_j+1)^2 & \text{if } w_j<-1\\ 0 & \text{if } -1\leq w_j\leq 1\end{cases}$ $w_j=\sum_{i=1}^{n}\left(\frac{2j}{m}-1\right)^{n-i}x_i$, for $n=17$: $d=10558.145$, and $m=100$.
F9	$[-D^2,D^2]$	6	UN	Trid6	$f(x)=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}x_ix_{i-1}$
F10	$[-D^2,D^2]$	10	UN	Trid10	$f(x)=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}x_ix_{i-1}$
F11	[-5,10]	10	UN	Zakharov	$f(x)=\sum_{i=1}^{n}(x_i)^2+(\sum_{i=1}^{n}0.5ix_i)^2+(\sum_{i=1}^{n}0.5ix_i)^4$
F12	[-4,5]	24	UN	powell	$f(x)=\sum_{i=1}^{n/k}(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+(x_{4i-2}-x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4$
F13	[-10,10]	30	UN	Schwefel2.22	$f(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$
F14	[-20,20]		MN	Dekkers-Aarts	$f(x)=10^5x_1^2+x_2^2-(x_1^2+x_2^2)^2+10^{-5}(x_1^2+x_2^2)^4$
F15	[0,5]	3	MN	Gulf Research	$f(x)=\sum_{1}^{99}[exp(-\frac{(u_i-x_2)^{x_3}}{x_1})-0.01*i]^2$ where $u_i=25+[-50\ln(0.01\times i)]^{\frac{1}{1.5}}$
FI6	[-100,100]	2	MS	Bohachevsky1	$f(x)=x_1^2+2x_2^2-0.3cos(3\pi x_1)-0.4cos(4\pi x_2)+0.7$
F17	[-10,10]	2	UN	Matyas	$f(x)=0.26(x_1+x_2)-0.48x_1x_2$

D:Dimension, C:Characteristic, U:Unimodal, M:Multimodal S:Seperable, N:Non-Separable, Opt: Optimum.

Table 20: 70 Global Optimization Test Functions continued

No	Range	D	C	Functions	Description
F19	$[0,\pi]$	10	MS	EpistaticMichalewicz 10	$f(x)=-\sum_{i=1}^{n} sin(y_i)(sin(iy_i^2/\pi))^{2m}$ $y_i=\begin{cases} x_i cos(\theta)-x_{i+1}sin(\theta), i=1,3,5,...,\leq n \\ x_i sin(\theta)-x_{i+1}cos(\theta), i=2,4,6,...,\leq n \\ x_i, i=n \end{cases}$ $\theta=\pi/6, m=10$
F20	[-100, 100]	30	UN	Quadric	$f(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$
F21	[-100, 100]	30	UN	Chung Reynolds	$f(x)=\left(\sum_{i=1}^{n} x_i^2\right)^2$
F22	[-100,100]	2	MN	Schaffer1	$f(x_1,x_2)=0.5+\frac{\sin^2\sqrt{x_1^2+x_2^2}^2-0.5}{[1+0.001(x_1^2+x_2^2)]^2}$
F23	[-5,5]	2	MN	Six Hump Camel Back	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$
F24	[-100,100]	2	MN	Bohachevsky2	$f(x)=x_1^2+2x_2^2-0.3cos(3\pi x_1)(4\pi x_2)+0.3$
F25	[-100,100]	2	MN	Bohachevsky3	$f(x)=x_1^2+2x_2^2-0.3cos(3\pi x_1+4\pi x_2)+0.3$
F26	[-10,10]	2	MN	Shubert	$f(x)=\left(\sum_{i=1}^{5} i\cos(i+1)x_i+i\right)\left(\sum_{i=1}^{5} i\cos\left((i+1)x_2+i\right)\right)$
F27	-5,5[]	4	MN	Kowalik	$f(x)=\sum_{i=1}^{11}\left(a_i-\frac{x_1(1+x_2b_i)}{(1+x_3b_i+x_4b_i^2}\right)^2$ The values for a_i and b_i are given in Table 5
F28	[0,10]	4	MN	Shekel5	$f(x)=-\sum_{i=1}^{5}\frac{1}{\sum_{j=1}^{4}(x_j-a_{ij})^2+c_i}$ with constants a_{ij} and c_j given in Table 6 below
F29	[0,10]	4	MN	Shekel7	$f(x)=-\sum_{j=1}^{7}\frac{1}{\sum_{i=1}^{4}(x_j-a_{ij})^2+c_i}$
F30	[0,10]	4	MN	Shekel10	$f(x)=-\sum_{j=1}^{10}\frac{1}{\sum_{i=1}^{4}(x_j-a_{ij})^2+c_i}$
F31	[0,1]	3	MN	Hartman3	$f(x)=-\sum_{i=1}^{4} c_i\exp\left[-\sum_{j=1}^{3} a_{ij}(x_j-p_{ij})^2\right]$ with constants a_{ij}, p_{ij} and c_i given in Table 3
F32	[0,1]	6	MN	Hartman6	$f(x)=-\sum_{i=1}^{4} c_i\exp\left[-\sum_{j=1}^{6} a_{ij}(x_j-p_{ij})^2\right]$ with constants a_{ij}, p_{ij} and c_i given in Table 9 and 10
F33	[-5.12,5.12]	5	US	Stepint	$f(x)=25+\sum_{i=1}^{5}\lfloor x_i\rfloor$
F34	[-32,32]	30	MN	Ackley	$f(x)=-20exp^{(}-0.2\sqrt{\sum_{i=1}^{n}\frac{x_i^2}{n}})-exp^{(}\sum_{i=1}^{n} cos\frac{2\pi x_i}{n})+20+e$
F35	[0,10]	2	MN	Langerman2	$f(x)=-\sum_{j=1}^{5} c_j\cos(\pi d_j)\exp(-d_j/\pi)$, where $d_j=\sum_{i=1}^{n}(x_i-a_{ji})^2$ The constants c_j and a_{ji} are given in Table 12
F36	[0,10]	5	MN	Langerman5	$f(x)=-\sum_{j=1}^{5} c_j\cos(\pi d_j)\exp(-d_j/\pi)$, where $d_j=\sum_{i=1}^{n}(x_i-a_{ji})^2$ The constants c_j and a_{ji} are given in Table 12
F37	[0,10]	10	MN	Langerman10	$f(x)=-\sum_{j=1}^{5} c_j\cos(\pi d_j)\exp(-d_j/\pi)$, where $d_j=\sum_{i=1}^{n}(x_i-a_{ji})^2$ The constants c_j and a_{ji} are given in Table 12
F38	[0,10]	10	MN	Shekel Foxholes 10	$f(x)=-\sum_{j=1}^{30}\frac{1}{c_j+\sum_{i=1}^{n}(x_i-a_{ji})^2}$ The constants c_j anda_{ji} are given in Table 7.
F39	[0,10]	20	MN	Shekel Foxholes 20	$f(x)=-\sum_{j=1}^{30}\frac{1}{c_j+\sum_{i=1}^{n}(x_i-a_{ji})^2}$ The constants c_j anda_{ji} are given in Table 7.
F40	[-10,10]	10	MN	Aluffi Pentini	$f(x)=0.25{x_1}^4-0.5{x_1}^2+0.1x_1+0.5{x_2}^2$
F41	[-10,10]	2	MN	Berker-Lago	$f(x)=(\lvert x_1\rvert-5)^2+(\lvert x_2\rvert-5)^2$
F42	[-5,5]	2	MN	Three hump camel back	$f(x)=2x_1^2-1.05x_1^4+1/6x_1^6+x_1x_2+x_2^2$
F43	[-30,30]	30	UN	Rosenbrock	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$
F44	[-10,10]	30	UN	Dixon-Price	$f(x)=(x_1-1)^2+\sum_{i=2}^{n} i(2x_i^2-x_{i-1})^2$

D:Dimension, C:Characteristic, U:Unimodal, M:Multimodal S:Separable, N:Non-Separable, Opt: Optimum.

Table 21: 70 Global Optimization Test Functions continued.

No	Range	D	C	Functions	Description
F45	[-10,10]	3	MN	Meyer and Roth	$f(x)=\sum_{i=1}^{5}\left(\frac{x_1 x_3 t_i}{(1+x_1 t_i+x_2 v_i)}-y_i\right)^2$ Table 4 lists the parameter values of this problem.
F46	[-1,1]	4	MN	Miele and Cantrell	$f(x)=(\exp(x_1)-x_2)^4+100(x_2-x_3)^6+(\tan(x_3-x_4))^4+{x_1}^8$
F47	[-2,2]	2	MN	Multi-Gaussian	$f(x)=\sum_{i=1}^{5} a_i \exp\left(-((x_1-b_i)^2+(x_2-c_i)^2)/{d_i}^2\right)$ Table 1 lists the parameter values of this problem.
F48	[0,n]	4	US	Neumaier 2	$f(x)=\sum_{k=1}^{n}\left(b_k-\sum_{i=1}^{n}{x_i}^k\right)^2$ $b=(8,18,44,114)$
F49	$[-n^2,n^2]$	10	UN	Neumaier 3	$f(x)=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}x_i x_{i-1}$
F50	[2,10]	10	UN	Paviani	$f(x)=\sum_{i=1}^{10}\left[(\ln(x_i-2))^2+(\ln(10-x_i))^2\right]-\left(\prod_{i=1}^{10}x_i\right)^{0.2}$
F51	[-10,10]	2	MN	Periodic	$f(x)=1+\sin^2 x_1+\sin^2 x_2-0.1\exp(-{x_1}^2-{x_2}^2)$
F52	[-100,100]	5	UN	Salomon5	$f(x)=1-\cos(2\pi\|x\|)+0.1\|x\|$
F53	[-100,100]	10	UN	Salomon10	$f(x)=1-\cos(2\pi\|x\|)+0.1\|x\|$
F54	[-100,100]	2	MN	Schaffer 2	$f(x)=(x_1^2+x_2^2)^{0.25}\left(\sin^2\left(50(x_1^2+x_2^2)^{0.1}\right)+1\right)$
F55	[0,180]	10	UN	Sinusoidal10	$f(x)=-[A\prod_{i=1}^{n}\sin(x_i-z)+\prod_{i=1}^{n}\sin(B(x_i-z))]$ $A=2.5,B=5,z=30$
F56	[0,180]	20	UN	Sinusoidal20	$f(x)=-[A\prod_{i=1}^{n}\sin(x_i-z)+\prod_{i=1}^{n}\sin(B(x_i-z))]$ $A=2.5,B=5,z=30$
F57	[0,6]	2	MN	Hosaki Problem	$f(x_1,x_2)=\left(1-8x_1+7x_1^2-\frac{7}{3}x_1^3+\frac{1}{4}x_1^4\right)x_2^2\exp(-x_2)$
F58	[-1.5,4]	2	MN	McCormick	$f(x)=\sin(x_1+x_2)+(x_1-x_2)^2-(3/2)x_1+(5/2)x_2+1$
F59	[-5,5]	2	UN	Modified Rosenbrock	$f(x)=100(x_2-{x_1}^2)^2+\left[6.4(x_2-0.5)^2-x_1-0.6\right]^2$
F60	[-10,10]	3	MN	Helical Valley	$f(x)=100\left[(x_2-10\theta)^2+(\sqrt{(x_1^2+x_2^2)}-1)^2\right]+x_3^2$ $\theta=\begin{cases}\frac{1}{2\pi}\tan^{-1}\frac{x_2}{x_1}, & \text{if} x_1\geq 0\\ \frac{1}{2\pi}\tan^{-1}\frac{x_2}{x_1}+\frac{1}{2}, & \text{if} x_1<0\end{cases}$
F61	[-1,1]	2	MS	Cosine Mixture2	$f(x)=0.1\sum_{i=1}^{n}\cos(5\pi x_i)-\sum_{i=1}^{n}x_i^2$
F62	[-1,1]	4	MS	Cosine Mixture4	$f(x)=0.1\sum_{i=1}^{n}\cos(5\pi x_i)-\sum_{i=1}^{n}x_i^2$
F63	[-1,1]	10	US	Exponential	$f(x)=\exp\left(-0.5\sum_{i=1}^{n}{x_i}^2\right)$
F64	[-4.5,4.5]	5	UN	Beale	$f(x)=(1.5-x_1+x_1x_2)^2+(2.25-x_1+x_1+x_1x_2^2)^2+(2.625-x_1+x_1+x_1x_2^3)^2$
F65	$[0,\pi]$	5	MS	EpistaticMichalewicz 5	$f(x)=-\sum_{i=1}^{n} sin(y_i)(sin(iy_i^2/\pi))^{2m}$ $y_i=\begin{cases}x_i cos(\theta)-x_{i+1}sin(\theta), i=1,3,5,...,\leq n\\ x_i sin(\theta)-x_{i+1}cos(\theta), i=2,4,6,...,\leq n\\ x_i, i=n\end{cases}$ $\theta=\pi/6, m=10$
F66	[-65.536,65.536]	2	MS	Foxholes	$f(x)=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})6}\right]^{-1}$ Data for a_{ij} is in table 14
F67	[-2,2]	2	MN	GoldStein-Price	$f(x)=\left[1+(x_1+x_2+1)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]\times\left[30+(2x_1-3x_2)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$
F68	[-10,10]	3	MS	Levy and Montalvo	$f(x)=\frac{\pi}{n}\left(10\sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]+\frac{\pi}{n}(y_n-1)^2\right.$
F69	[-10,10]	2	MS	Booth	$F(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$
F70	[-10,10]	4	UN	Colville	$f(x)=100(x_1^2-x_2^2)^2+(x_1-1)^2+(x_3-1)^2+90(x_3^2-x_4)^2+10.1(x_2-1)^2+(x_4-1)^2+19.8(x_2-1)(x_4-1)$

D:Dimension, C:Characteristic, U:Unimodal, M:Multimodal S:Seperable, N:Non-Separable, Opt: Optimum.

Appendix B: ACSO Results

Table 22: Best Performance of ACSO, RIO, DSARIO, PSO

FN	Optimum	ACSO	RIO	DSARIO	PSO
F1	0	**0.0000**	3.9382E-02	1.1299E-08	9.8573E-03
F2	0	**0.0000**	1.5500E+02	0.0000	2.300E+01
F3	0	**9.0293E-24**	3.9581E-07	1.7095E-10	1.5973E-07
F4	0	**2.7637E-23**	5.5692E-04	1.4065E-07	5.5964E-04
F5	-1.143833	-1.0011	**-1.0276**	-1.0360	-1.0151
F6	-1	**-1**	**-1**	**-1**	**-1**
F7	0	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F8	0	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F9	-50	-33.6890	-31.6910	-10.1710	**-38.0600**
F10	-210	**-137.6200**	121.4600	-45.6420	-69.0820
F11	0	**1.8832E-24**	5.9397E-04	1.9627E-10	6.8842E-04
F12	0	**9.4229E-25**	6.1341E-04	1.2968E-07	9.2828E-04
F13	0	**2.9377E-12**	5.5889E-01	8.7281E-06	6.4874E-01
F14	-24777	-22184	**-24691**	-19973	-24565
F15	0	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F16	0	**0.0000**	3.595E-06	1.6358E-09	2.1835E-05
F17	0	**1.4699E-27**	3.6530E-05	2.4759E-07	4.1298E-05
F18	0	**0.0000**	6.7898E-07	3.6719E-09	6.2938E-09
F19	-9.660152	-9.8456	**-10.0000**	-9.9811	**-10.0000**
F20	0	**4.3189E-26**	4.3762E-08	1.1521E-10	1.8292E-08
F21	0	**1.3520E-41**	2.6833E-09	1.9864E-15	6.9725E-13
F22	0	-2.6481	**-2.7553**	22.7553	-2.7499
F23	-1.03163	-9.9277E-01	**-1.0316**	-9.5735E-01	-1.0090
F24	0	**0.0000**	1.6630E-05	4.7145E-09	5.8541E-05
F25	0	**0.0000**	2.8642E-08	7.329E-09	3.3781E-05
F26	-186.73	**-8.2562**	-8.2477	-8.1464	-8.2176
F27	0.0003	3.7517E-06	2.1093E-08	6.2377E-08	**1.3628E-09**
F28	-10.15	-1.4660	**-3.7750**	-4.5643E-01	-1.9817
F29	-10.4	-7.8120E-01	-2.7559	**-12.0036**	-3.5417
F30	-10.53	-5.2573E-01	**-3.7600**	-1.3069	-2.2810
F31	-3.86	-3.8226	-3.8550	-3.8076	**-3.8602**
F32	-3.32	**-3.1067**	-3.0826	-2.4693	-2.9338
F33	0	**-6.1000E01**	-2.5000E01	5.0000	-2.2000E01
F34	0	**1.9321E-11**	5.353E-04	8.4601E-06	5.8690E-04
F35	-1.08	**-1.4090E-16**	**-1.4094E-16**	**-1.4098E-16**	**-1.4098E-16**
F36	-1.5	**-1.4060E-16**	**-1.4098E-16**	**-1.4094E-16**	**-1.4099E-16**
F37	-1.5	-1.3347E-16	**-1.4099E-16**	**-1.4099E-16**	**-1.4098E-16**
F38	-10.4056	-9.8307E-01	-1.3094	-1.4688	**-1.5441**
F39	-10.2088	-3.4013E-01	**-3.5228E-01**	-3.4584E-01	-3.4372E-01
F40	-0.3523	**-3.5211E-01**	-3.4109E-01	-3.1769E-01	-3.4459E-01
F41	0	**1.70027E-06**	2.3286E-06	3.1320E-02	3.6012E-05
F42	0	**7.5833E-25**	6.2857E-05	3.9838E-08	1.8839E-05
F43	0	1.0769E+07	3.7859E-01	8.9087	**8.2952E-02**
F44	0	1.4162E-05	7.5074E-05	4.6735E-03	**1.8000E-06**
F45	0.4E-04	**7.2454E-10**	4.0475E-07	3.8019E-06	1.8895E-07
F46	0	**1.0604E-04**	8.6098E-04	1.4066E-02	8.4200E-04
F47	1.29695	7.2141E-243	**0.0000**	4.5782E-22	**0.0000**
F48	0	1.1781E01	**3.9447**	4.5521	3.9447
F49	-210	**-7.9172E04**	-7.7673E03	-1.0906E03	01.7252E03
F50	-45.778	-1.9141E01	-5.6207E11	**-7.6812E17**	-1.6121E11
F51	0.9	**0.9**	**0.9**	**0.9**	**0.9**
F52	0	**1.0991E-14**	9.9873E-02	1.1205E-05	9.9873E-02
F53	0	**9.8233E-13**	9.9873E-02	1.0445E-05	9.9873E-02
F54	0	**7.6923E-07**	4.1395E-04	4.1395E-04	4.1625E-04
F55	-3.5	-9.499E-01	1.2494E-01	**-1.1933**	-8.0514E-01
F56	-3.5	-2.6155E-02	-4.1685E-02	**-3.1086**	-4.3874E-02
F57	-2.3458	-3.1282E03	-2.2252E02	-2.328	**-3.8513E04**
F58	-1.9133	**-8.1589**	-3.3818	-1.9022	-2.0481
F59	0	2.8211E-04	**7.2138E-06**	2.1574E-03	3.8011E-05
F60	0	1.3204E02	**9.1670E-05**	6.9628	2.3656E-04
F61	0.2	-2.6427E01	-1.0574E02	-2.0569	**-1.1730E02**
F62	0.4	-5.4858E01	**-9.2125E01**	-3.4265	-3.9841E01
F63	1	2.0836E-22	3.8070E-46	6.0731E-01	**1.1217E-69**
F64	0	7.9563E-05	3.5228E-05	1.3170E-02	**8.1844E-06**
F65	-4.687658	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F66	0.998	1.9575E+01	**0.998**	1.2550	**0.998**
F67	3	3.6434	**3.0000**	3.1334	**3.0000**
F68	0	8.5823E-04	**3.0519E-05**	1.4519E-02	1.1017E-04
F69	0	4.3013E-04	**3.2116E-05**	3.6923E-03	4.1482E-05
F70	0	**-2.5272E+14**	-6.2056E+04	-6.3024E+04	-2.1511E+06
No of good optimum		38	26	13	22

Bold values indicate good optimum values.

Table 23: Average Performance of ACSO, RIO, DSARIO, PSO

FN	Optimum	ACSO	RIO	DSARIO	PSO
F1	0	**2.1587E-13**	1.16661E-01	1.4953E-04	1.2597E-01
F2	0	**0.0000**	6.1320E+02	**0.0000**	4.4884E+02
F3	0	1.6125E-04	2.5735E-04	**1.1382E-04**	3.3156E-04
F4	0	**1.3373E-15**	8.7191E-04	1.0044E-04	8.7330E-04
F5	-1.143833	-1.0745	-0.8511	**-2.9387**	-0.8638
F6	-1	**-1**	**-1**	**-1**	**-1**
F7	0	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F8	0	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F9	-50	**-29.1370**	-12.7710	0.2779	-16.3650
F10	-210	**-36.7220**	31.9200	2.6811	0.8737
F11	0	**9.9540E-15**	9.9726E-03	1.3465E-04	2.8009E-03
F12	0	**4.0871E-18**	6.5472E-03	1.0354E-04	1.8167E-03
F13	0	**7.3497E-09**	4.7632	3.3592E-04	3.3316
F14	-24777	**-10233**	-13541	-49992	-13734
F15	0	5.3366E-14	**0.0000**	2.1377E-07	2.9688E-05
F16	0	**2.4425E-15**	4.1192E-04	1.7616E-04	2.8545E-05
F17	0	**8.9534E-05**	5.2121E-04	9.2077E-05	5.8413E-04
F18	0	1.6968E-04	2.5530E-04	**1.0955E-04**	2.2522E-04
F19	-9.660152	-6.6374	-9.5000	-9.0058	**-9.6277**
F20	0	**9.1595E-05**	2.3375E-04	1.3138E-04	2.5793E-04
F21	0	1.1460E-04	1.3617E-04	**4.2881E-05**	1.9027E-04
F22	-1	-1	-1.8842	**-2.2495**	-2.0333
F23	-1.03163	-4.6410E-01	**-7.1969E-01**	-3.7566E-01	-6.3740E-01
F24	0	**1.4633E-15**	9.1175E-03	1.3343E-04	3.6042E-04
F25	0	**9.6589E-16**	4.9737E-04	1.4538E-04	4.4007E-04
F26	-186.73	-5.8022	**-7.2260**	-6.5406	-7.2458
F27	0.0003	0.0003	0.00014	0.00067	**0.000098**
F28	-10.15	-2.1673E-01	**-1.0107**	-2.4993E-01	-6.0081E-01
F29	-10.4	-2.4531E-01	-6.8191E-01	-2.8590E-01	**-9.2466E-01**
F30	-10.53	-2.0936E-01	-7.4732E-01	-4.0651E-01	**-8.8047E-01**
F31	-3.86	-2.1273	**-3.7400**	-3.5107	-3.7105
F32	-3.32	-5.8408E-01	-2.2863	-1.7483	-2.4543
F33	0	**-3.2120E01**	-7.0000	7.6400	-8.6400
F34	0	**1.2732E-08**	7.3086E-01	2.8468E-04	1.3644
F35	-1.08	-6.4271E-17	-1.2332E-16	-8.6860E-17	**-1.2918E-10**
F36	-1.5	-5.8388E-17	-1.2842E-16	-9.6111E-17	**-1.3049E-16**
F37	-1.5	-4.6596E-17	**-1.3516E-16**	-8.8841E-17	-1.2696E-16
F38	-10.4056	-5.1432E-01	-1.0019	-8.9258E-01	**-1.0250**
F39	-10.2088	-2.1065E-01	**-3.2412E-01**	-2.9201E-01	-3.1824E-01
F40	-0.3523	-1.2487E-01	-1.4533E-01	-1.1179E-01	**-2.1600E-01**
F41	0	7.6633	4.5971E-04	3.9890E-01	**4.5304E-04**
F42	0	**7.9618E-05**	5.8447E-04	1.5175E-04	3.9017E-04
F43	0	4.0446E+09	1.4550E+01	**8.9798**	3.1514E+01
F44	0	1.7646E+04	5.8276E-04	3.7712E-01	**5.3484E-04**
F45	0.4E-04	**8.4621E-05**	9.7778E-05	2.8271E-03	2.2392E-04
F46	0	2.7967	**9.7615E-04**	1.2896E-01	9.8618E-04
F47	1.29695	2.6646E-09	1.7253E-34	1.9672E-16	**1.2699E-34**
F48	0	4.9627E06	**3.9447**	8.1848E01	3.9447
F49	-210	**-2.4102E04**	-1.0015E03	-5.0677E02	-1.0086E03
F50	-45.778	**-4.5881E22**	-2.7607E13	-2.2138E17	-3.4309E13
F51	0.9	1.1554	**0.9**	**0.9**	**0.9**
F52	0	**1.4758E-09**	1.0387E-01	2.3258E-04	1.0789E-01
F53	0	**6.6572E-10**	2.5987E-01	2.751E-04	2.5984E-01
F54	0	**5.2004E-05**	1.8339E-03	4.8227E-04	1.8059E-03
F55	-3.5	-2.9110E-01	**-3.7784E-01**	-3.6724E-01	-3.6488E-01
F56	-3.5	-6.3524E-03	-9.6019E-03	**-1.4117E-01**	-1.1676E-02
F57	-2.3458	-1.9594E02	-2.8143E01	-2.0993	**-3.7641E03**
F58	-1.9133	**-3.8574**	-1.8162	-1.4894	-1.8169
F59	0	9.7618E04	4.4141E-03	3.8105E-01	**3.4730E-03**
F60	0	2.6000E04	5.0687E-03	6.4212E01	**1.3925E-03**
F61	0.2	-5.8459	-1.8680E01	-1.7583	**-1.8769E01**
F62	0.4	**-2.1117E01**	-2.0972E01	-2.8178	-1.5381E01
F63	1	2.8826E-04	1.174E-04	6.1723E-01	**1.0401E-04**
F64	0	2.0765E+05	**5.3430E-04**	1.3750	4.8829-04
F65	-4.687658	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F66	0.998	4.6078E+02	**0.998**	4.3453E+02	**0.998**
F67	3	3.3689E+03	**3.0000**	2.9140E+01	4.0800
F68	0	2.8324E01	5.3490E-04	3.6745E-01	**5.2574E-04**
F69	0	3.4994E+02	4.8456E-04	7.3302	**4.0821E-04**
F70	0	**-1.4110E+13**	-3.3199E+03	-9.6055E+03	-8.6682E+04
No of good optimum		30	18	13	25

Bold values indicate good optimum values.

Table 24: Standard Deviation of Average Optima of ACSO, RIO,DSARIO, PSO

FN	ACSO	RIO	DSARIO	PSO
F1	**1.0765E-12**	9.2344E-02	2.2195E-04	7.0443E-02
F2	**0.0000**	5.2984E+02	**0.0000**	2.8334E+02
F3	2.8381E-04	2.3894E-04	**1.9129E-04**	2.8938E-04
F4	**5.211E-15**	1.1052E-04	1.2398E-04	1.1570E-04
F5	**4.4159E-01**	6.7003E-01	9.0750	6.1386E-01
F6	1.4544E-01	**2.6811E-02**	3.9507E-02	2.7422E-02
F7	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F8	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F9	**3.4156**	8.9229	5.7431	10.1810
F10	29.848	121.460	**11.8800**	73.0930
F11	**4.9703E-14**	4.5325E-02	1.8891E-04	8.3398E-03
F12	**2.0280E-17**	1.9173E-02	12.7292E-04	1.7136E-03
F13	**2.4747E-08**	4.9643	2.8583E-04	2.0864
F14	**6.9373E02**	8.4103E03	7.0350E03	6.5270E03
F15	2.6683E-13	**0.0000**	9.9333E-07	1.4844E-04
F16	**1.1892E-14**	2.8876E-04	2.8737E-04	2.8548E-04
F17	**2.1866E-04**	2.8287E-04	1.0144E-04	2.9537E-04
F18	2.8640E-04	2.9507E-04	**1.6544E-04**	2.9907E-04
F19	3.0730	3.9689E-01	8.2091E-01	**3.4992E-01**
F20	**2.1050E-04**	2.8747E-04	2.5360E-04	3.0304E-04
F21	2.2402E-04	2.0020E-04	**1.4461E-04**	2.8286E-04
F22	8.6883E-01	**5.2018E-01**	7.4135E-01	5.9152E-01
F23	3.0142	2.621E-01	**2.5677E-01**	2.7416E-01
F24	**5.0091E-15**	4.3583E-02	2.0221E-04	2.1900E-04
F25	**2.9988E-15**	2.9271E-04	1.9047E-04	2.2547E-04
F26	2.3471	8.8542E-01	1.0986	**7.2980E-01**
F27	3.1749E-04	1.9548E-04	1.4103E-03	**1.8463E-04**
F28	3.3822E-01	9.3313E-01	**8.9364E-02**	4.4564E-01
F29	1.9151E-01	4.9562E-01	**1.7102E-01**	8.7578E-01
F30	**1.6936E-01**	7.0590E-01	2.834E-01	5.8296E-01
F31	1.5484	**7.2763E-02**	1.8706E-01	9.9101E-02
F32	9.4195E-01	4.1581E-01	4.4954E-01	**3.2249E-01**
F33	1.9654E01	6.4872	**2.1579**	6.7631
F34	**5.8327E-08**	8.7925E-01	2.8049E-04	1.0682
F35	6.0609E-17	2.3589E-17	4.6158E-17	**-1.5819E-17**
F36	5.2628E-17	1.9809E-17	4.4372E-17	**1.7567E-17**
F37	4.9743E-17	**9.3102E-18**	3.9599E-17	2.5882E-17
F38	3.3458E-01	1.5669E-01	**1.4846E-01**	1.9542E-01
F39	1.0759E-01	1.5440E-02	1.7900E-02	**1.3756E-02**
F40	**1.0522E-012**	1.1236E-01	9.5984E-02	1.0183E-01
F41	3.8209E+01	3.2123E-04	3.7194E-01	**2.9985E-04**
F42	2.3721E-04	2.8022E-04	**1.7251E-04**	3.1285E-04
F43	7.1440E+09	2.2233E+01	2.1289E-02	5.1801E+01
F44	5.8577E+04	**2.9033E-04**	2.4840E-01	3.2449E-04
F45	1.6341E-04	**1.5542E-04**	5.7177E-03	2.5941E-04
F46	1.0426E+01	3.9289E-05	1.030E-01	**3.0972E-05**
F47	8.2204E-09	1.7253E-34	1.9672E-16	**1.2699E-34**
F48	1.0891E07	**6.7093E-15**	1.0394E+02	8.1018E-15
F49	2.6887E04	2.613E02	**2.3337E02**	3.0581E02
F50	2.2197E23	**3.1804E13**	1.7242E17	5.6333E13
F51	4.2083E-01	4.8190E-02	**3.3993E-16**	4.5826E-02
F52	**6.614E-09**	2.0000E-02	2.1929E-04	2.7891E-02
F53	**1.0155E-09**	9.5143E-02	2.8179E-04	1.1902E-01
F54	**8.2843E-05**	1.4509E-03	1.2317E-04	1.4764E-03
F55	1.7196E-01	**1.2494E-01**	2.2953E-01	2.0094E-01
F56	**6.4343E-03**	9.7767E-03	6.1956E-01	1.1532E-02
F57	6.7986E02	5.8096E01	**1.8136E-01**	1.0042E04
F58	2.6835	4.6137E-01	**3.2371E-01**	1.3196
F59	1.8022E05	**3.4632E-03**	2.4691E-01	3.5715E-03
F60	4.3273E04	2.1839E-02	2.6652E01	**3.4963E-03**
F61	4.9619	2.1696E01	**1.9731E-01**	2.1904E01
F62	1.8927E01	1.7100E01	**3.0635E-01**	7.4261
F63	3.0041E-04	2.6589E-04	8.5085E-03	**1.9838E-04**
F64	6.4456+05	**2.9125E-04**	1.2711	3.3238E-04
F65	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F66	1.2878E+02	**2.2662E-16**	1.5088E+02	2.2662E-16
F67	9.3311E03	**7.0799E-10**	2.7704E01	5.4000
F68	4.4115E+01	2.4954E-04	2.2470E-01	2.4292E-04
F69	9.7496E+02	**2.9911E-04**	6.8336	2.9635E-04
F70	5.1521E+13	**1.2363E+04**	1.7384E+04	4.3008E+05
No of minimum STD	24	19	21	16

Bold values indicate minimum standard deviation values.

Table 25: Success Rate of ACSO, RIO, DSARIO, PSO

FN	ACSO	RIO	DSARIO	PSO
F1	100%	0%	100%	0%
F2	100%	0%	100%	0%
F3	100%	100%	100%	100%
F4	100%	100%	100%	100%
F5	100%	92%	100%	92%
F6	100%	100%	100%	100%
F7	100%	100%	100%	100%
F8	100%	100%	100%	100%
F9	100%	100%	36%	100%
F10	100%	80%	28%	92%
F11	100%	96%	100%	88%
F12	100%	56%	100%	64%
F13	100%	0%	100%	0%
F14	100%	100%	100%	100%
F15	100%	100%	100%	100%
F16	100%	100%	100%	100%
F17	100%	100%	100%	100%
F18	100%	100%	100%	100%
F19	100%	100%	100%	100%
F20	100%	100%	100%	100%
F21	100%	100%	100%	100%
F22	100%	100%	100%	100%
F23	100%	100%	100%	100%
F24	100%	96%	100%	100%
F25	100%	100%	100%	100%
F26	100%	100%	100%	100%
F27	100%	100%	88%	100%
F28	100%	100%	100%	100%
F29	100%	100%	100%	100%
F30	100%	100%	100%	100%
F31	100%	100%	100%	100%
F32	100%	100%	100%	100%
F33	100%	100%	0%	100%
F34	100%	52%	100%	28%
F35	100%	100%	100%	100%
F36	100%	100%	100%	100%
F37	100%	100%	100%	100%
F38	100%	100%	100%	100%
F39	100%	100%	100%	100%
F40	100%	100%	100%	100%
F41	92%	100%	0%	100%
F42	100%	100%	100%	100%
F43	0%	0%	0%	0%
F44	48%	100%	0%	100%
F45	100%	100%	56%	100%
F46	68%	100%	0%	100%
F47	100%	100%	100%	100%
F48	0%	0%	0%	0%
F49	100%	100%	100%	100%
F50	100%	100%	100%	100%
F51	0%	0%	0%	0%
F52	100%	0%	100%	0%
F53	100%	0%	100%	0%
F54	100%	60%	100%	60%
F55	100%	100%	100%	100%
F56	100%	100%	100%	100%
F57	100%	100%	100%	100%
F58	100%	100%	100%	100%
F59	4%	44%	0%	56%
F60	0%	96%	0%	96%
F61	100%	100%	100%	100%
F62	100%	100%	100%	100%
F63	100%	100%	0%	100%
F64	68%	100%	0%	100%
F65	100%	100%	100%	100%
F66	0%	0%	0%	0%
F67	0%	0%	0%	0%
F68	4%	100%	0%	100%
F69	12%	100%	0%	100%
F70	100%	100%	100%	100%
Number of problems evaluated with 100% success rate	57	47	51	52

Table 26: Normalized Success Performance (nSP) for ACSO, RIO, DSARIO and PSO

FN	ACSO	RIO	DSARIO	PSO
F1	30.38	-	1.00	-
F2	17.73	-	1.00	-
F3	22.98	1.77	1.00	1.79
F4	19.98	11.68	1.00	11.84
F5	1.00	81.46	1.06	81.44
F6	1.00	1.00	1.00	1.00
F7	1.00	2.59	2.59	2.59
F8	1.30	1.00	1.00	1.00
F9	1.71	1	570.12	1.11
F10	1.00	13.73	130.85	5.12
F11	32.09	61.53	1.00	24400.91
F12	39.48	592.74	1.00	439.24
F13	17.51	-	1.00	-
F14	2.56	1.00	1.33	1.04
F15	1.30	1.02	1.00	1.04
F16	24.3	3.69	1.00	3.52
F17	40.47	2.96	1.00	2.84
F18	16.55	1.86	1.00	2.04
F19	1.00	1.00	1.00	1.00
F20	21.66	1.82	1.00	1.77
F21	7.43	1.00	1.04	1.00
F22	1.01	1.03	1.00	1.00
F23	2.42	1.02	1.00	1.00
F24	36.29	27.67	1.00	3.41
F25	30.65	3.59	1.00	3.76
F26	1.00	1.00	1.00	1.00
F27	2.02	1.00	159.59	1.00
F28	1.00	1.00	1.00	1.00
F29	1.00	1.00	1.00	1.00
F30	1.00	1.00	1.00	1.00
F31	1.00	1.00	1.00	1.00
F32	1.00	1.00	1.00	1.00
F33	2.38	1.00	-	1.08
F34	21.96	363.24	1.00	981.39
F35	1.00	1.00	1.00	1.00
F36	1.00	1.00	1.00	1.00
F37	1.00	1.00	1.00	1.00
F38	1.00	1.00	1.00	1.00
F39	1.00	1.00	1.00	1.00
F40	2.80	1.08	1.00	1.10
F41	181.14	1.00	-	1.09
F42	44.48	2.43	1.00	3.89
F43	-	-	-	-
F44	132.32	1.00	-	1.01
F45	1.58	1.04	756.45	1.00
F46	21.62	1.00	-	1.36
F47	1.00	1.00	1.00	1.00
F48	-	-	-	-
F49	1.28	1.00	1.00	1.00
F50	1.00	1.00	1.00	1.00
F51	-	-	-	-
F52	21.36	-	1.00	-
F53	18.06	-	1.00	-
F54	15.62	278.25	1.00	259.72
F55	1.00	1.00	1.00	1.00
F56	1.00	1.00	1.00	1.00
F57	1.12	1.00	1.00	1.00
F58	1.26	1.00	1.00	1.00
F59	30.66	1.63	-	1.00
F60	-	1.00	-	1.26
F61	1.00	1.00	1.00	1.00
F62	1.00	1.00	1.00	1.00
F63	2.52	0.96	-	1.00
F64	183.19	1.00	-	1.67
F65	1.00	1.00	1.00	1.00
F66	-	-	-	-
F67	-	-	-	-
F68	4994.83	1.00	-	1.05
F69	2178.46	1.04	-	1.00
F70	38.68	3.30	1.00	5.88

"-" indicates that the algorithm can not solve the problem successfully.

Appendix C: DCSO-TSP Results

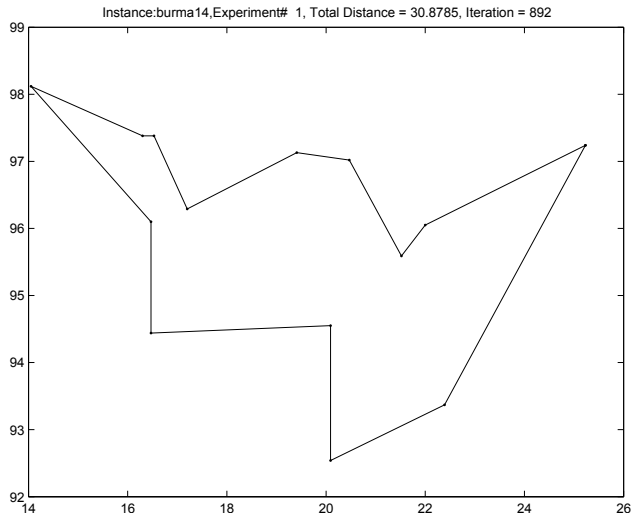

Figure 1: Graphs illustrating burma14 instance total distance.

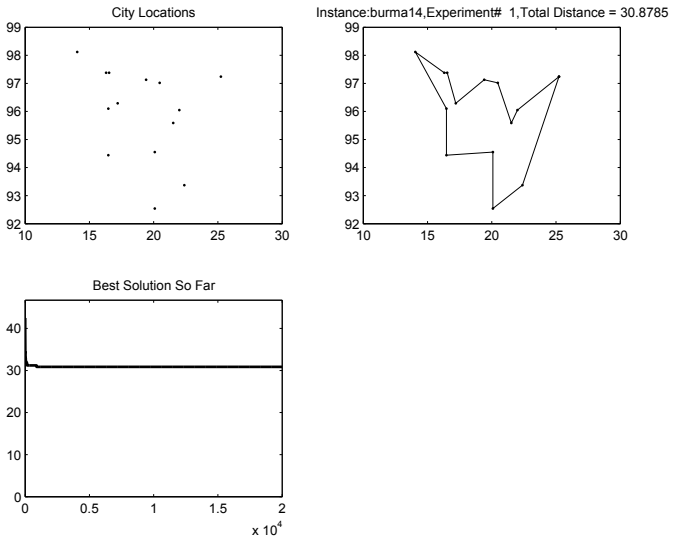

Figure 2: Graphs illustrating DCSO-TSP Tour for 14 Cities.

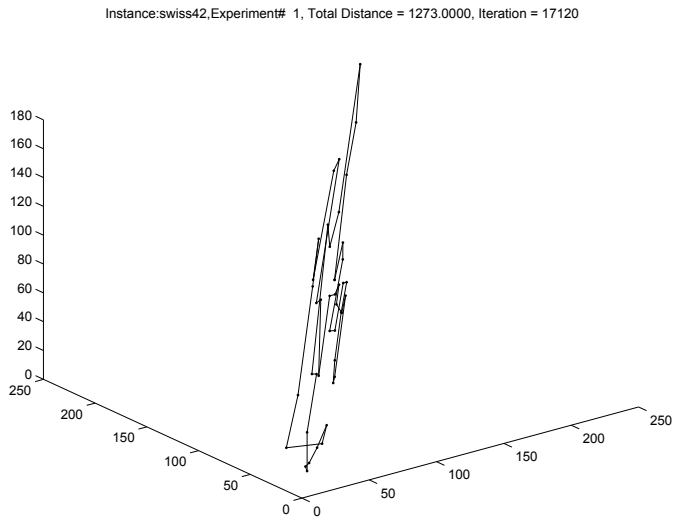

Figure 3: Graphs illustrating Swiss42 instance total distance.

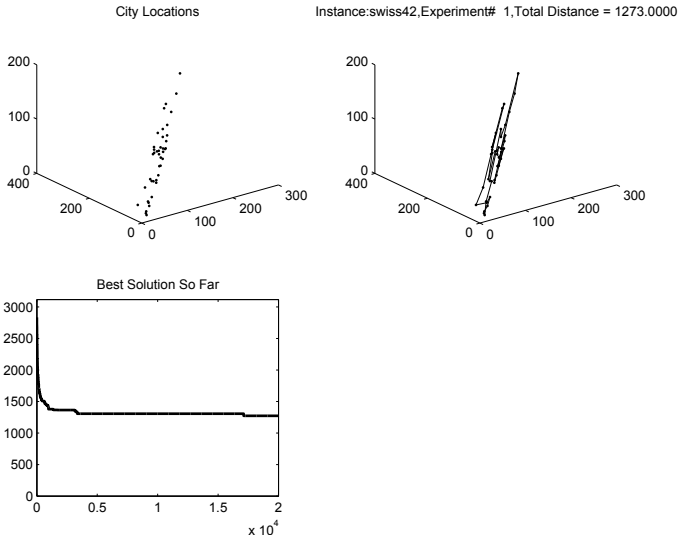

Figure 4: Graphs illustrating DCSO-TSP Tour for Swiss 42 Cities.

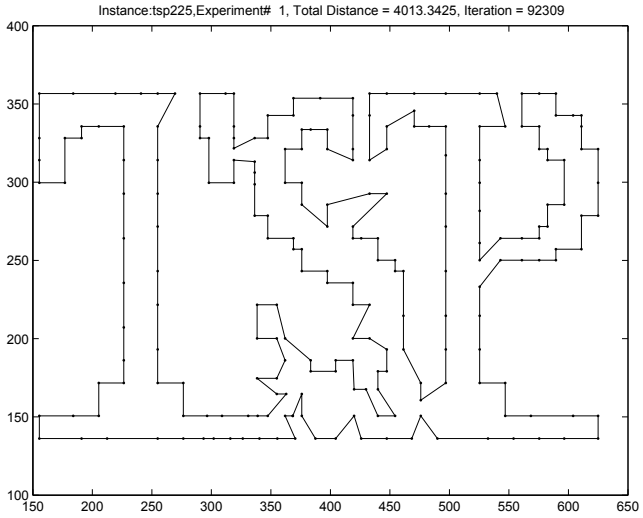

Figure 5: Graphs illustrating TSP225 instance total distance.

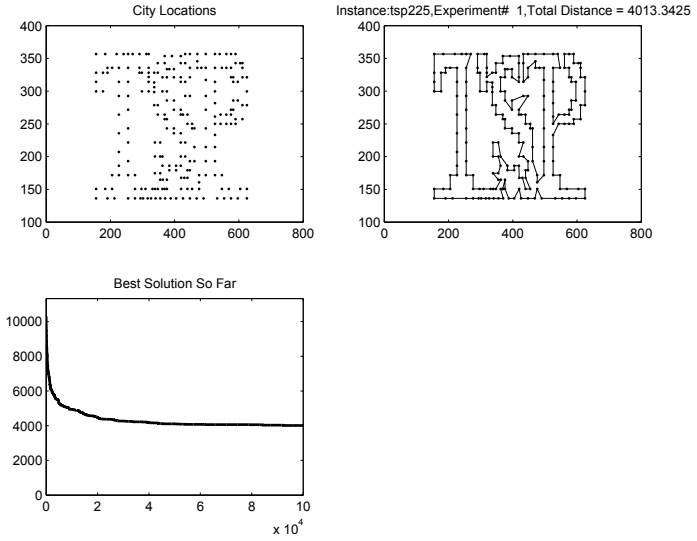

Figure 6: Graphs illustrating DCSO-TSP Tour for 225 Cities.

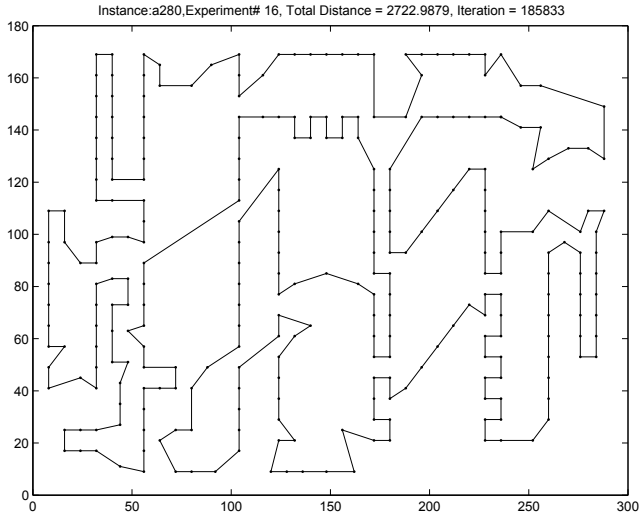

Figure 7: Graphs illustrating a280 instance total distance.

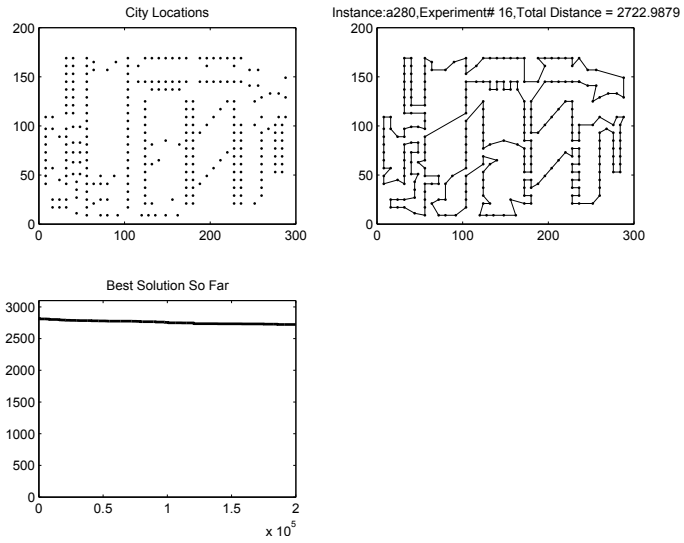

Figure 8: Graphs illustrating DCSO-TSP Tour for 280 Cities.

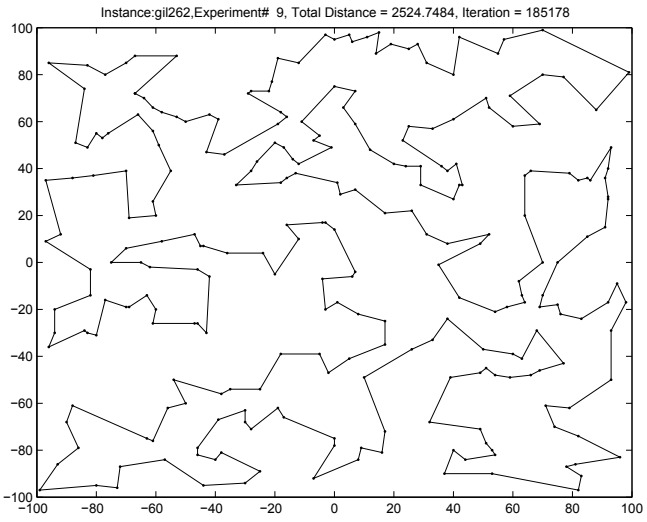

Figure 9: Graphs illustrating gil262 instance total distance.

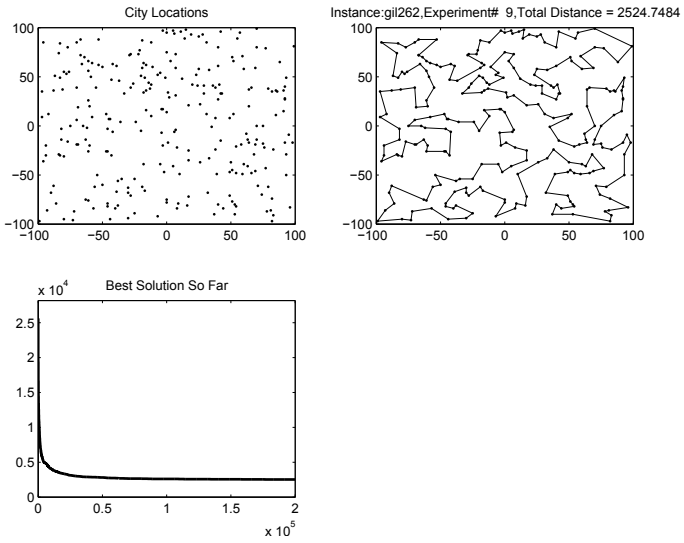

Figure 10: Graphs illustrating DCSO-TSP Tour for 262 Cities.

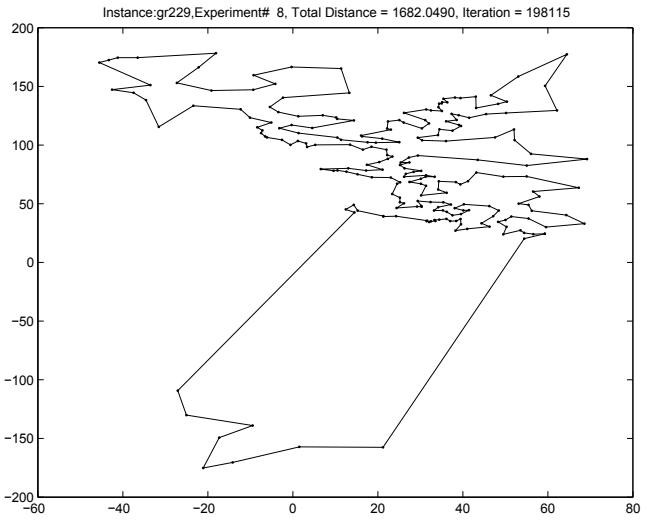

Figure 11: Graphs illustrating gr299 instance total distance.

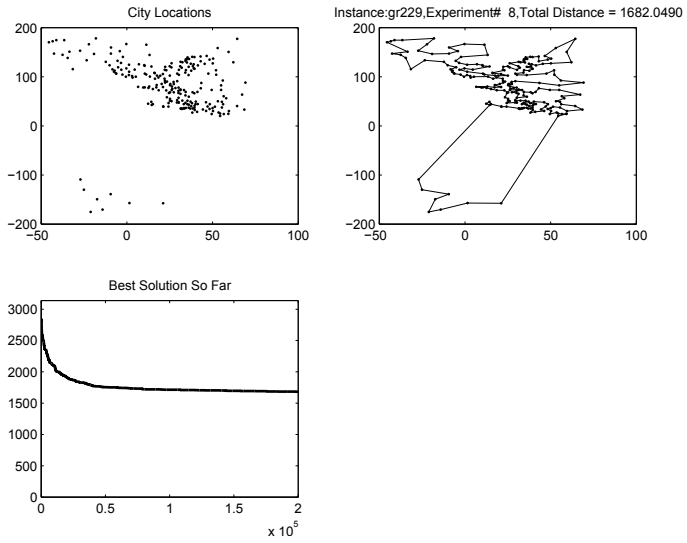

Figure 12: Graphs illustrating DCSO-TSP Tour for 299 Cities.