THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Data-Driven Analysis of Variables and Dependencies in Continuous Optimization Problems and Estimation of Distribution Algorithms

Krishna Manjari Mishra

B. Comp. Sc., M. Comp. Sc.

A thesis submitted for the degree of Doctor of Philosophy at The University of Queensland in 2014

School of Information Technology and Electrical Engineering

Abstract

Optimization problems occur widely in various domains and are of great practical importance. Better solutions to optimization problems translate into increased production or efficiency, or less waste or error. As a result, the development of optimization algorithms is a large and active area of research. Although there is a large amount of theory and techniques for solving optimization problems, there are also many open issues and questions in understanding the relationship between optimization algorithms and the problems that they are applied to.

In any optimization problem, the influence of each solution variable on the objective function, as well as the interactions between variables are very important. For example, if the variables in a problem are independent, then it might be efficiently solved by decomposition. If important dependencies exist between variables, an algorithm that is able to capture these dependencies may be able to exploit them to find good solutions. Alternatively, if a variable has a strong influence on the objective function, then an algorithm may be better off focusing its search on this variable, compared with another variable that has very little influence on the objective function.

Estimation of Distribution Algorithms (EDAs) are a class of black-box optimization algorithms that learn and sample from a probabilistic model over the solution variables to carry out the search process. The explicit model in an EDA clearly specifies how the algorithm handles problem variables during the search process. A major focus in EDA research has been the incorporation of dependency modeling using models of varying complexity (e.g. probabilistic graphical models). The intuition behind this work is that if the algorithm model is capable and successful at capturing the structure of the problem, it will produce good performance on that problem. Experimental results have confirmed this intuition, for example EDAs that model dependency information have outperformed EDAs that do not model dependencies on certain problems.

While EDAs have shown good performance results, little work has analyzed the dynamics of EDA models in practice. In fact, it is not clear what kind of problem structure EDAs can successfully model, or to what extent it is necessary to successfully model problem structure in order to achieve good performance. To provide insight into these issues, a more detailed analysis of EDAs applied to specific problems is needed.

In this thesis, an experimental methodology is proposed to analyze the features of variables in continuous optimization problems and continuous EDAs. The approach is based on sampling points from the problem fitness landscape and/or the history of points visited by an EDA during the search process. These samples are then analyzed in three different ways, to identify key structural variables, variable dependencies and important variables. The techniques are used to analyze a variety of test problems and EDAs optimizing the same problem set. The results confirm that the interaction between variables is complex, varies across problems and gives useful insights into the performance of EDAs on these problems. The results are categorized into different problem/algorithm behavior types.

In continuous EDAs, a Gaussian distribution over continuous variables is commonly used, with several different covariance matrix structures ranging from diagonal, i.e. Univariate Marginal Distribution Algorithm (UMDA$_c$) to full, i.e. Estimation of Multivariate Normal Algorithm (EMNA$_{\text{global}}$). The modeling of key structural variables and correlations are already captured in standard EDAs. In contrast, so-called important variables are not identified by an EDA. In the final part of the thesis, a modified screening EDA (sEDA) is presented which identifies important variables and uses these to control the degree of covariance modeling in the Gaussian EDA model. Compared to EMNA$_{\text{global}}$, the algorithm provides improved numerical stability and can use a smaller selected population. Experimental results are presented to evaluate and compare the performance of the proposed algorithm to UMDA$_c$ and EMNA$_{\text{global}}$. In its first formulation, sEDA requires a large number of function evaluations for high dimensional problems. To address this issue, a modified version of (sEDA-lite) is also proposed. Experimental results on a large set of high dimensional artificial and real-world representative problems evaluate the performance of the new algorithm and compare it with the sEDA and EDA-MCC (EDA framework with Model Complexity Control), a related, recently proposed algorithm.

Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the General Award Rules of The University of Queensland, immediately made available for research and study in accordance with the Copyright Act 1968.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Publications during candidature

- Krishna Manjari Mishra and Marcus Gallagher. A Modified Screening Estimation of Distribution Algorithm for Large-Scale Continuous Optimization. The Tenth International Conference on Simulated Evolution And Learning (SEAL 2014), Dunedin, New Zealand.
- Krishna Manjari Mishra and Marcus Gallagher. Variable screening for reduced dependency modelling in Gaussian-based continuous estimation of distribution algorithms. In 2012 IEEE Congress on Evolutionary Computation (CEC). 2012 IEEE World Congress on Computational Intelligence (IEEE-WCCI 2012), Brisbane, Australia.
- Krishna Manjari Mishra and Marcus Gallagher. To Investigate the Key Structural Variables, Variable Correlations and Variable Importance in Continuous Estimation of Distribution Algorithms. 2014. Manuscript in preparation.

Publications included in thesis

“No publications included”.

Contributions by others to the thesis

No Contributions by others.

Statement of parts of the thesis submitted to qualify for the award of another degree

None.

Acknowledgements

A dream comes true.

Pursuing my Ph.D. from The University of Queensland, one of the world's most renowned universities, has been the most extraordinary experience ever.

There are many people without whose support, guidance. assistance and love, I would have floundered.

I sincerely express my gratitude to my supervisor, Associate Professor Dr. Marcus Gallagher for his continuous support of my Ph.D. study and research. He has always been an inspiration to me owing to his enthusiasm and immense knowledge of the subject matter. His generous and incomparable guidance and persistence has helped me complete my research and the production of my thesis.

I would also like to thank my associate supervisor, Dr Ian Wood for dealing in such a supportive manner with every query and challenge I confronted during my Ph.D. work. I have a genuine respect and admiration for his talents in Mathematics and Statistics and have valued all of his expert advice.

In addition to my supervisors, I would like also to thank Dr. Daniel Angus, Dr. Ruth Schulz, Dr. Nadarajah Mithulananthan and Dr. Xuelei Hu for their encouragement and insightful comments. I thank my fellow lab mates, Rachael Morgan, Bo Gao and Kirill Makukhin for the numerous lively discussions of many topics and for generously helping me throughout my course. This work has been supported by a University of Queensland Research Scholarship for which I am extremely grateful.

I extend a special thanks to all my friends who supported me at all times throughout my journey towards the Ph.D.

I am also most grateful to my parents, Mrs. Gaytri Mishra and Mr. Biswanath Mishra as well as my in-laws, Mrs. Meera Acharya and Mr. Dhaneswar Acharya. They have always supported me and encouraged me with their best wishes, and love.

I also express my sincere appreciation to my brothers, Shidharth and Arvind, who always extend their hands whenever I need them.

Finally, I would like to thank my husband, Dr. Sibasis Acharya and my little daughter Miss. Anishka Acharya. They have always supported me, cheered me up and stood by me through all of the good and the bad times.

Acknowledging all of the gifts and contributions to my academic journey, I silently pray to the Almighty for giving me enough courage and strength to persevere.

Keywords

optimization problems, metaheuristic algorithms, estimation of distribution algorithms, screening technique, sensitivity analysis, eigenanalysis

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 080101, Artificial Intelligence and Image Processing 100%

Fields of Research (FoR) Classification

FoR code: 0801, Artificial Intelligence and Image Processing 100%

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

EDA	Estimation of Distribution algorithm
CiaS	Circle in a Square
MWP	Multi Source Weber Problem
UMDA$_c$	Univariate Marginal Distribution Algorithm
EMNA$_{\text{global}}$	Estimation of Multivariate Normal Algorithm
CMA-ES	Covariance Matrix Adaptation-Evolution Strategy
CMR	Covariance Matrix Repairing
ECMR	Efficient Covariance Matrix Repairing
ECMR0	Efficient Covariance Matrix Repairing-zero
ED-EDA	Eigen Decomposition of EDAs
EDA-MCC	EDA framework with Model Complexity Control
EEDA	Eigenspace EDA
AVS	Adaptive Variance Scaling
SDR	Standard-Deviation Ratio triggering
AMS	Anticipated Mean Shift
PCA	Principal Component Analysis
sEDA	screening EDA
BBOB	Black-Box Optimization Benchmarking
sEDA-lite	screening EDA-lite

Symbols

$\mathbf{x}^*$	Optimal solution
$f(\mathbf{x}^*)$	Best fitness solution
$f(\mathbf{x})$	Candidate solution
n	Dimension of the problem
S	Search space
$\leq$	Less than equal to
$\forall$	For all
$\mathbf{x} = (x_1, \ldots, x_n)$	Individual
$\mathbb{R}^n$	Real space in n dimensional Problem
$lb(i)$	Lower bound
$ub(i)$	Upper bound
$f'(x)$	Slope of the fitness
z^i	The center of circle i
r	Radius of the circle
$C(z^i, r)$	The circle with center z^i and radius r
$\mathbf{w}^i$	Coordinates of point of the circle in the unit square
n_c	No. of Circles in the unit square
X	A set of population or sampling points in matrix format
p	No. of facilities
(x_i, x_{p+i})	The coordinates values of pth facility in an individual vector
A	Customer
(a_{1j}, a_{2j})	Coordinate values of the customer
t	Time index
N	Selected Population
$p(\mathbf{x})$	Probability distribution
μ	Mean of the selected population
σ	Standard deviation of the selected population
Σ	Covariance matrix
r_{ik}	Correlation coefficient between ith and kth variable
D	$n * n$ matrix, whose diagonal element contains the eigenvalues of the covariance matrix Σ

$\mathbf{v}$	Eigenvector
λ	Eigenvalues
c_i	Coefficient of the eigenvector
$E_i(\mathbf{x})$	Elementary effect values
Δ	Some fixed resolution or increment size
$\overline{E}_i(\mathbf{x})$	Estimated mean of the elementary effect values
$std(E_i(\mathbf{x}))$	Estimated standard deviation of the elementary effect values
$\overline{E}_i^*(\mathbf{x})$	Estimated absolute value of mean of the elementary effect
τ	Truncation selection
$\%$	Percentage
M	Population size
P	Initial population
M_{sel}	Selected population for sEDA and sEAD-lite
P_{tot}	Population after perturbation
M_{tot}	Size of the population after perturbation
M_{tot}^{sel}	Size of the selected population after perturbation
$\vec{m}$	mean of the M_{tot}^{sel}
η	Model selection parameter
p_o	Pareto optimal solutions
B	Best variables modeled using covariance matrix
θ	Sparse matrix
$\tilde{m}$	Median of the selected population
i,j,k	Index
$f(\mathbf{x}^{\mathbf{k}})$	Fitness values of kth individual
$\mathbf{M}$	Transformation matrix
$\vec{o}$	Offset values
$\frac{d_n}{f(x_n)}$	Performance ratio
μ_t	Mean at time t
Σ_t	Covariance matrix at time t

Chapter 1

Introduction

1.1 Overview

Optimization is ubiquitous. Optimization is necessary in every aspect of our daily life because it leads us to obtain an optimal solution from a limited amount of resources.

The wide application of optimization in various domains leads to a significant research topic in modern times. For example, optimization plays a significant role in economics and business, since it mainly deals with minimizing cost and maximizing profit from a specific amount of resources [34]. Network communication needs optimization for the efficient delivery of communication from one place to another through different media, hence it requires the optimal use of physical media, such as wire, fiber optics etc. in different locations [114]. Optimization also underpins the planning section [16], the time tabling [138], the transportation network [106], the evacuation plans during natural disaster [158] and many more.

In general, for all types of optimization problems, a number of different decision alternatives (variables), with specified available constraints are used to maximize or minimize the evaluation criteria of the objective function. The input variables are used for solving the objective function which returns either a single value, termed as the single objective function, or the function which solves more than one objective simultaneously, which is termed as multi objective function. Depending upon the type of variables involved, the optimization problems are categorized as combinatorial (for discrete variables) or continuous (for continuous variables).

There exists a number of algorithms to solve the optimization task. Among these, evolutionary computation has gain the interest of the researchers over the last few decades. Evolutionary algorithms (EA) are nature inspired, that is they deal with population based methods at each iteration with a set of candidate solutions. At each iteration, the algorithm chooses the individuals having best fitness values and these are used for generating the set of population for the next generation using some genetic operators (crossover or mutation) or some probability distribution. The Evolutionary algorithms have many more categories, one of which is the genetic algorithm which was initiated by Holland [82]. In Genetic algorithms, the best selected population is used to generate the set of the next population using the recombination operators or the Mutation operators. Choosing this operator is a strenuous task to do. In the last decade, researchers tried to control the limitations of the genetic operators by introducing Estimation of Distribution Algorithms (EDAs), the essence of which is based on probability theory. The discussion of EDAs is presented in Section 2.6 of Chapter 2.

1.2 Research Gap, Methodology and Justification

The aim of this dissertation is to develop an understanding and to analyze the properties of optimization problem variables and how these variables are modeled in Estimation of Distribution algorithms (EDA). The difficulty of an optimization problem clearly relates to the problem variables, their interactions and the resulting sensitivity of the objective function. If a problem contains a variable that has little or no effect on the objective function value, then it may be inefficient to include this variable in the search. Alternatively, if the variables in a problem are independent, then a high dimensional problem may be more efficiently solved by decomposing it. Hence the performance of an algorithm will be related to the understanding of problem variables, and if their important properties can be captured efficiently and effectively.

EDAs are a class of black-box optimization algorithms that learn and sample from a probabilistic model over the solution variables to carry out the search process. The explicit model in an EDA specifies how the algorithm handles problem variables during the search process. A major focus in EDA research has been the incorporation of dependency modeling using models of varying complexity (e.g. probabilistic graphical models). The intuition behind this work is that if the algorithm model is capable and successful at capturing the structure of the problem, it will produce good performance on that problem.

While EDAs have shown good performance results, little work has analyzed the dynamics of EDA models in practice. In fact, it is not clear what kind of problem structure EDAs can successfully model, or to what extent it is necessary to successfully model problem structure in order to achieve good performance. To provide insight into these issues, a more detailed analysis of EDAs applied to specific problems is needed.

A unified, data-driven approach based is used in this thesis for understanding and analyzing the variables in problems as well as algorithms. Here, three main types of methodologies are investigated: firstly, for identifying dependent variables, secondly, for analyzing the structure of the population/selected sample points and finally for determining which variables the objective function is most sensitive to. These methodologies reveal the characteristics of the variables in optimization problems and algorithms, which leads to insight into which EDA model produces the best algorithm for a given problem. Furthermore, by comparing the results based on data sampled from the problem and data produced by the EDA, the analysis shows whether or not the EDA captures the same structure found from sampling the problem. Overall, the thesis makes a significant contribution to the understanding of Gaussian EDAs and provides a general framework for understanding the relationship between optimization problems and algorithms that could be extended and applied to other problems and algorithms in the future.

1.3 Blueprint of the Thesis

The blueprint of the remainder of the thesis is as follows. Chapter 2 introduces the continuous optimization problems and discusses different types of examples of continuous optimization problems. The Estimation of Distribution Algorithms and their different types are discussed here. This chapter emphasizes on the continuous Gaussian based EDAs and points out the issues related to them. It also discusses on various remedies to overcome these issues. Finally it explores the research gap found in EDAs. Chapter 3 is mainly concerned with the explanation of methodologies considered in the thesis. It examines correlation coefficients, eigenanalysis and the Morris method for identifying and analyzing correlated variables, Key structural variables and important variables in Continuous optimization problems and EDAs. In Chapter 4, the methodologies studied in Chapter 3 are applied to different continuous optimization problems to analyze and understand the nature of variables. Different categories are made based on the analysis and examples have been set for each category. In Chapter 5, the methodologies studied in Chapter 3 are applied to the examples of each category developed in Chapter 4 and analyzed with regard to whether EDA is capturing the right information about the nature of variables or not. Chapter 6 develops an algorithmic framework and its modified form which control the covariance modeling of EDAs. The Comparison of the newly developed framework with different EDAs is also covered in this chapter, while Chapter 7 concludes the work of the thesis.

1.4 Scope and Limitation

The main contribution of this thesis is a framework that can be used for the data-driven analysis of variables in blackbox optimization problems and in particular EDAs. A second contribution is two modified EDAs which incorporate part of the analysis framework and apply it to address issues regarding covariance modeling in EDAs for high dimensional problems. The analysis is applicable to any continuous black-box optimization problem. It is particularly applicable to Gaussian-based continuous EDAs and to some extent, to other black-box optimization algorithms. This thesis examines a range of continuous optimization problems and Gaussian-EDAs.

The framework focuses on measuring three general features: variable dependencies, key structural variables and variable importance. Each of these factors could be investigated with a variety of data analysis techniques from machine learning and statistics; however this thesis focuses on the use of one relatively simple, standard technique to analyze each factor. Nevertheless, these techniques are well-suited to the data produced by problem sampling and Gaussian EDAs.

The main focus of this thesis is not on trying to demonstrate the superior performance of any algorithm on any specific problem, but rather to develop analysis to better understand the relationship between optimization problem and algorithm instances. Reasonable, but not exhaustive effort is spent on tuning algorithm parameters. These are used to implement and evaluate the proposed analysis framework (Chapters 4 and 5) and proposed modified EDAs in Chapter 6. For the new algorithms, results are compared with those from the literature to evaluate the relative performance.

1.5 Original Contributions

The main novel contributions of this thesis are:

1. The development of a new methodological framework for analyzing the nature of variables in black-box optimization problems and algorithms. The framework and techniques are based on sample data and are generally applicable, however they are particularly well-suited to the analysis of continuous, Gaussian-based EDAs. The framework incorporates three ideas:

   - An approach to identify key structural variables using eigenanalysis in optimization problems and EDAs.
   - Visualization of the correlation graph to determine correlation between variables in optimization problems and EDAs.
   - Using an existing sensitivity analysis method to determine the important variables in optimization problems and EDAs.

2. A comparative experimental study of artificial and real-world representative optimization problems and EDAs applied to these problems. This study evaluates the proposed framework and illustrates it's utility and application.

3. Two variants of a modified Gaussian EDA, incorporating the screening technique from the proposed methodological framework. The algorithm incorporates knowledge of the objective function value and uses it to control dependency modeling, providing a new way of resolving issues associated with estimating the full covariance matrix in $\text{EMNA}_{\text{global}}$.

Chapter 2

Continuous Black-Box Optimization, Example Problems and Estimation of Distribution Algorithms

2.1 Overview

This chapter presents a brief review of black-box optimization problems, particularly continuous optimization problems and defines a number of artificial test problems and real world problems that will be studied in this thesis. In addition to this, Estimation of Distribution Algorithms (EDAs) are reviewed and discussed. The key research issues related to Gaussian-based EDAs and the existing methods for solving these issues are described. A particular focus of the chapter is the analysis of the nature and management of variables and their interactions, in problems and in EDAs.

2.2 Continuous Optimization Problems

The "practical" goal of (global) optimization can be defined as *"There exists a goal (e.g. to find as small a value of f() as possible), there exist resources (e.g. some number of trials), and the problem is how to use these resources in an optimal way"* [150].

The global optimization problem is to find $\mathbf{x}^*$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \in S,$$

where S is the set of feasible solutions. If $S \subseteq \mathbb{R}^n$, then it is a continuous optimization problem, where n is the dimensionality of the problem, $f(\mathbf{x})$ is the fitness or objective function and $\mathbf{x} = (x_1, \ldots, x_n)$ is an individual or candidate solution vector. If S is an n dimensional box-constrained subset of $\mathbb{R}^n$ then the domains of variables are defined by their lower and upper rectangular cartesian bounds, i.e. $lb(i) \leq x_i \leq ub(i)$, where lb is the lower bound, ub is the upper bound and the value of i ranges from 1 to n. Optimization problems can also involve some additional constraints [113, 52, 83]. The problem is stated above as a minimization problem, which can easily be turned into a maximization problem by negating the objective function.

Continuous optimization problems require finding a vector of variables, subject to a set of constraints or boundary values. The constraints depend upon the system of the problem, which may be physical, societal, legal etc. Test optimization problems that are used in research often have simple or no constraints, but in real world problems, constraints are often present and are defined very specifically with respect to a problem. A solution vector is feasible if it obeys all the constraints else considered as infeasible. The goal of optimization is to maximize or minimize the objective function using all the evaluation criteria and produce the best possible solution.

In a convex optimization problem, the objective and constraint functions are convex. Solving a convex optimization problem is very simple and effective. But recognizing a problem as convex is very difficult. A number of technologies are exist for transforming problems into convex form [25].

If optimization problems are convex, then the single local optimum is the global one, but when it is non-convex, and have a number of local optima, it is very difficult to obtain the global optimum values. Hence, optimization problems having larger local optima are generally hard to solve. It is also difficult to solve in high dimensions. Incorporating all constraints while solving the problem is also a challenging task.

The main mathematical approach for optimizing a continuous function is gradient-based. It may be gradient ascent or gradient descent, depending upon the solution type. Algorithms such as steepest descent, Newton and quasi-Newton variants, require the objective function to be once or twice differentiable. These are local optimization algorithms [141, 127].

Although gradient-based techniques are useful for solving many optimization problems, there are many other problems where gradient information is not available and/or the objective function contains local optima, etc. global optimization and metaheuristic.

2.3 Test Problems

Given that optimization problems occur in so many different fields, it is not surprising that a large number of different algorithms have been developed to solve them. According to the "no free lunch concept" [156, 38], no algorithm is superior to any other algorithm when averaged over the set of all possible problems, provided they are all performing well in all of the problems. The practical implication of this result is that the performance of algorithms is problem dependent. That is, the performance of an algorithm strongly depends upon the problem it is applied to. Therefore to evaluate the general performance of an algorithm, it is important to evaluate it on a wide variety of different test problems [45, 110, 47].

A large number of benchmark problems and problem instances have been used in the optimization literature for the experimental evaluation of algorithms. Based on the properties of the problems, the test cases have been categorized by various researchers.

The artificial test functions are mathematical formulations, constructed artificially to have certain properties. For example, some problems are defined so that they are unimodal and have smoothness in their structure, where other functions have features such as multimodality, ridges, symmetry, noise

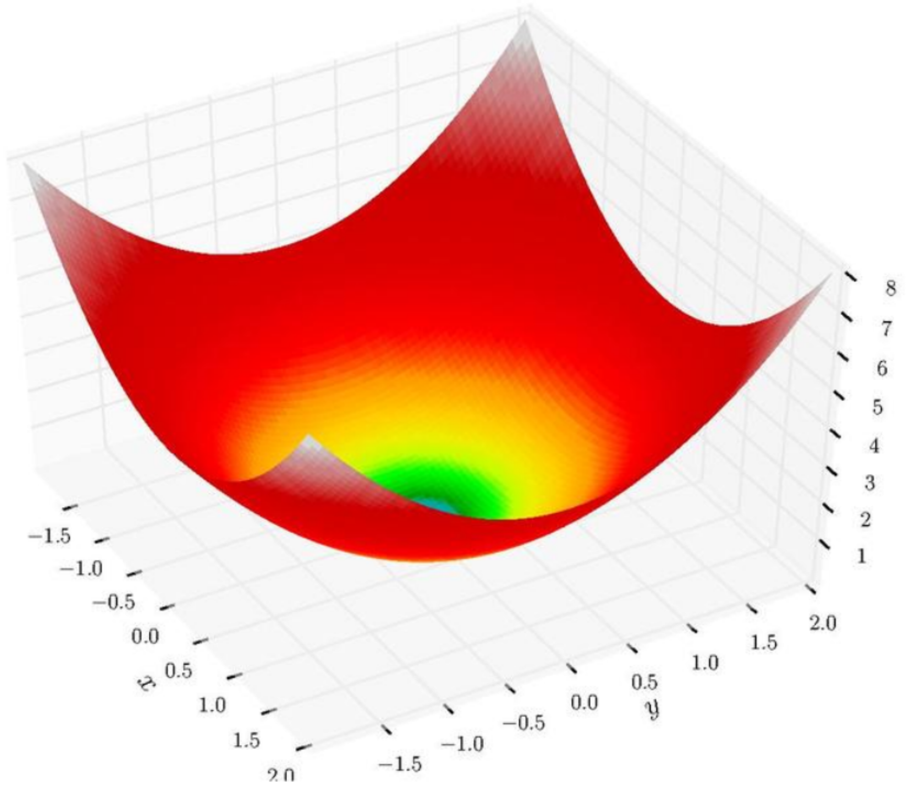


Figure 2.1: 3D Landscape Sphere Function
`http://en.wikipedia.org/wiki/Test_functions_for_optimization`

etc. The locations and values of local and global optima for these functions are typically known, which is useful in examining the performance of the algorithms. These problems are also often scalable in dimensionality. In evolutionary computation, many researchers have used the test functions which were introduced in the so-called De Jong test suite [94]. These problems are popular, since they are easy to implement and the results are easy to interpret. Successively, a number of researchers have introduced a large number of different test instances [50, 53, 157, 79].

A 3D landscape figure of Sphere function has been shown in Figure 2.1.

Problem instance generators provide an alternative source of test functions. Typically due to randomization, problem generators can produce a large number of problem instances, which avoids algorithms being tuned to some specific problems during experimentation. For continuous problems, one example is the Max-set of Gaussians (MSG) generator [59]. A list of test problem generators can be found on the MSG generator webpage: `http://staff.itee.uq.edu.au/marcusg/msg.html`.

Another class of problems used in research are based on real world problems. Typically less is known about real-world problems instances (e.g. landscape structure, number of local optima, value of global optimum) and they can be complex (e.g. incorporating nonlinear constraints) and difficult to implement (e.g. requiring domain expert knowledge). On the other hand, there are advantages in using real world problems. Most importantly, they indicate how an algorithm might perform in practice, since the eventual goal is to solve real optimization problems. They also often have features that make them difficult to solve for many algorithms, e.g. the fitness functions are not differentiable and non-convex. In larger dimensions some real-world problems have proven to be difficult for many

Table 2.1: Artificial Test Problems used in the Thesis

Problems	Formulation	Domain	Global Optimum Value
Sphere	$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	$[-600,600]^n$	0.0
Ackley	$f(\mathbf{x}) = -20.exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ $-exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	$[-600,600]^n$	0.0
Rastrigin	$f(\mathbf{x}) = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$	$[-5.12,5.12]^n$	0.0
Rosenbrock	$f(\mathbf{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2$ $+(x_i - 1)^2]$	$[-10,10]^n$	0.0
Griewangk	$f(\mathbf{x}) = 1/4000 \sum_{i=1}^{n} x_i^2$ $-\prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]^n$	0.0
Rotated Ellipsoid	$f(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	$[-10,10]^n$	0.0
2D Elliptical	$f(\mathbf{x}) = x_1^2 + 10x_2^2$	$[-5,5]^2$	0.0
SumCan	$f(\mathbf{x}) = 1/\left(10^{-5} + \sum_{i=1}^{n} \lvert x_i \rvert\right)$	$[-0.16,0.16]^n$	100000
Schwefel Problem 2.26	$f(\mathbf{x}) = -\sum_{i=1}^{k} x_i sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-500,500]^n$	-418.9829

algorithms to find a good solution.

A number of researchers are providing guidelines for framing test suites to meet the goals of EA research [154, 9, 13, 66, 89]

2.4 Problems used in the Thesis

In this thesis, both artificial and real world problems are considered. The artificial test functions are widely used in various optimization research and since their shape is known, they lend themselves easy for analysis.

Additionally, real world problems are also considered to show how the analysis can be used in real problems. The circles in a square packing problems and location-allocation problems are discussed in Sections 2.4.2 and 2.4.3 respectively.

2.4.1 Artificial Test problems

A set of artificial test functions used in the various parts of the thesis are tabulated in Table 2.1. The Sphere and Ellipse functions are continuous convex, unimodal and additively separable unconstrained minimization test functions. These two functions are simple and used in analyzing and comparing the efficiency of the algorithms. The Ackley, Rastrigin, Griewangk and Rosenbrock functions are multimodal with a large number of local minima. The multimodal nature reflects the characteristics of real world problems as well. The Rosenbrock function is also known as a banana function. The global optimum is inside a long, narrow, parabolic-shaped flat valley. To find the valley is trivial; however, convergence to the global optimum is difficult and hence the problem has been repeatedly

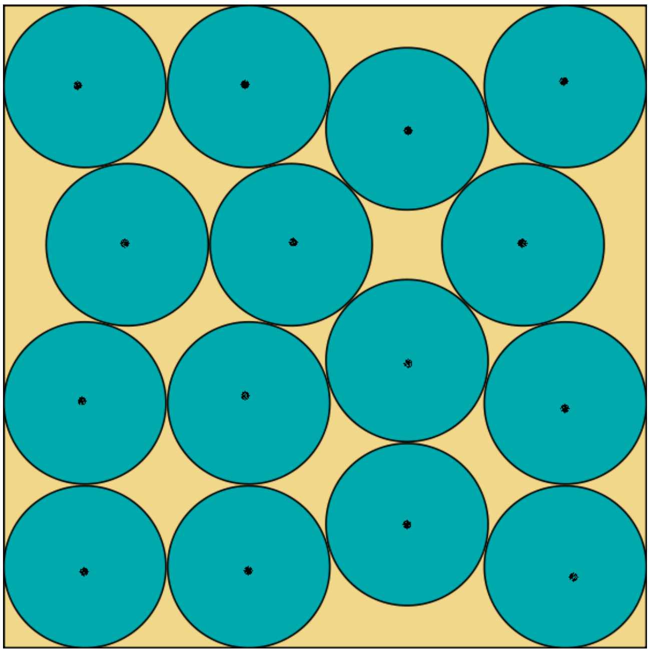


Figure 2.2: Example of 15 equal sized circles packing in a unit square
`http://en.wikipedia.org/wiki/File:Circles_packed_in_square_15.svg`

used in assessing the performance of optimization algorithms. Rotated Ellipsoid function is a non-separable function. All the functions are minimization problems except the SumCan function.

The problems defined in Table 2.1 is used throughout the thesis. In addition to this, another set of problems, the Real-Parameter Black-Box Optimization Benchmarking (BBOB) experiment set [76] are used in Chapter 6 to findout the model parameters for 2 different proposed algorithm (sEDA and sEDA-lite). In addition to these sets of problems, another set of problems from [41] has been used in Chapter 6 to compare different algorithms.

2.4.2 Circles in a Square Packing Problems

Packing problems are a class of optimization problems which involve packing objects together (often inside a container), as densely as possible. Circle packing has received a considerable amount of attention in the mathematics and operation research literature. A survey of circle and sphere packing problems has been found in [81].

For the Circles in a Square (CiaS) problem, a unit square is defined in a 2D Euclidean space and a pre-specified number n_c of equal-sized circles are given. The objective of the problem is to find an optimum solution for packing, i.e. to position the circles and compute the radius of the circles such that the circles occupy the maximum possible area within the square. The circles should not overlap each other and should not cross the boundary of the unit square.

Pictorially, an example of 15 equal sized circles in a unit square is shown in Figure 2.2.

Mathematically, the problem can be stated, as follows:

Let, $z^i = (x_i, y_i) \in \mathbb{R}^2$ be the center of circle i and $C(z^i, rad)$ the circle with center z^i and radius rad, then the problem is

$$rad_n = max \quad rad \tag{2.1}$$

$$C(z^i, rad) \subseteq D \qquad i = 1, ..., n \tag{2.2}$$

$$C^{int}(z^i, rad) \cap C^{int}(z^j, rad) = \varnothing; \forall i \neq j \tag{2.3}$$

where C^{int} denotes the interior of a circle [1, 28].

Alternatively, the problem can be reformulated as finding the positions of n_c points inside the unit square such that their minimum pairwise distances are maximized. In this case the problem (and constraints) can be restated as :

$$d_n = max \min_{i \neq j} \left\| \mathbf{w}^i - \mathbf{w}^j \right\|_2, \tag{2.4}$$

$$\mathbf{w}^i \in [0,1]^2, i = 1, \cdots, n_c \tag{2.5}$$

This problem cannot be solved using analytical approaches or via gradient-based mathematical optimization. It is a specific class of geometric packing problems.

It is known that a solution to (2.4) can be transformed into a solution to (2.1) using the following relation:

$$rad_n = \frac{d_n}{2\left(d_n + 1\right)} \tag{2.6}$$

The second formulation (2.4) is convenient for evaluating algorithms because generating feasible candidate solutions simply requires placing a set of n points within the unit square. The optimization problems are over $2n_c$ continuous variables (the coordinates of each point $\mathbf{w}^i$ in the unit square).

The research on Circle in a Square packing problem was initiated in 1967 by Kravitz, who first proposed the optimal solutions for n_c = 1-19 [98]. Since then, a number of researchers have used different types of solution strategies and have tried to determine the optimal solutions for an increasing number of circles [149, 122, 108, 18, 105, 146].

A solution strategy based on simulated annealing for the optimal packing of arbitrary-shaped polygons has been validated against existing analytical results for packing circles in a square. A constant penalty for equal circles in a square and variable penalty method in packing arbitrary polygons has been also introduced to improve the solution quality [149].

In addition to this, some rules which improved the efficiency of the Simulated annealing solution strategy have also been introduced. An iterative solution strategy based on some energy function for finding the optimal solution of circles ranging from 20 to 50 is introduced in [122]. The packing of uniform-sized circles inside a circular container and a solution strategy based on a billiards simulation algorithm has also been proposed [108]. Graham et al. in 1998 used two solution strategies, which were based on strategies proposed by [122] and [108] to improve the solution quality of circles ranging from 20-50, but later they proved that the strategy proposed in [108] was more effective.

Later in 2000, Boll et al. improved the solution quality for n_c = 32, 37, 48 and 50 by using a

2-phase solution strategy [18].

Additionally, some researchers used branch and bound solution strategy to exploit the special structure of the problem and the properties fulfilled by some of its solutions. Raber et. al. included a combination of strategies such as, a special partitioning strategy, symmetry avoiding strategy, the corner edge rules and the volume reduction strategy, to get optimal solutions for the problems $n_c \leq 35$, $n_c = 38$ and $n_c = 39$ and their results proved to be optimal within the given tolerance [105].

Szabo et al. who conducted the survey on the uniform-sized circle packing problem inside a square provided a theoretical and computational approach to this problem, and have produced approximate results up to $n_c = 200$ [146].

CiaS problem contains a large number of local optima, which was proved by Grosso et al., who solved the equal circles in a circular container and estimated the number of a local optima by using local optimizer. By using a traditional method, at least 4000 local optima for $n_c = 25$ and more than 16000 local optima for $n_c = 40$ have been determined [71].

Castillo discussed several circle packing problems with some industrial applications, such as circular cutting, container loading, cylinder packing, facility dispersion and communication networks as well as facility and dashboard layout [28]. The CiaS packing problem can be considered to be a simplified version of such real world problems.

Gallagher in 2009 analyzed the landscape properties of the CiaS problem and provided some guidelines to the characteristics and difficulty of the problems [56].

In the CiaS problem, there is a high degree of symmetry and most of the algorithms find it difficult to get the optimal solution. Research also has introduced some symmetry breaking constraints to the CiaS problem and produced good results for low dimensional cases [36].

For larger values of n_c, finding the optimal solution is generally difficult as well as time consuming. A large list of optimal packings of circles which ranges from 2 up to 10000, along with references and other related resources has been maintained by `http://www.packomania.com/`.

CiaS problems have a number of properties which are identified to be useful for evaluating algorithms ([154, 56]).

2.4.3 Location-Allocation Problems

Location allocation problems are a general class of optimization problems that are found in many real-world domains and which have been widely studied in operation research [132, 115, 148, 147]. A continuous location-allocation problem is also known as the Multisource Weber Problem (MWP). The location-allocation framework mainly consists of three essential components, which are as follows:

- Facilities - the type of goods and services used for the allocation.
- Locations - placing of facilities based on the territorial space, taking into account the user requirements and geographical sites.
- Customers - who are distributed in a geographical area, where the demands for goods and services are originated.

The objective of this problem is to place the facilities in such locations that the demand of each customer is satisfied by the facilities at minimum cost. There are a number of applications of this location-allocation problem, such as, location of warehouses, distribution centers, communication centers, production facilities, disaster management etc. [4, 112, 2].

The mathematical formulation of the location-allocation problem is as follows: the location of p facilities in continuous space in order to serve customers at m fixed points as well as the allocation of each customer to the facilities so that total transportation costs are minimized [137, 26]. Let us assume that there is no restriction on the capacity of the facilities. The location-allocation model is formulated as follows:

$$Minimize \quad \sum_{j=1}^{m} \min_{i} d_j \left(X_i, A_j\right)$$

where $i = 1 \cdots p$, X is the vector containing the 2D coordinates of the facilities. For the p facilities problem, there are $2p$ number of variables. Hence, for X_ith facility the coordinates values are (x_i, x_{p+i}).

A is the vector consisting of the known coordinates of the customers in the problem. For A_jth customer and X_i facility, the coordinate values are represented as (a_{1j}, a_{2j}) and (x_i, x_{p+i}) respectively. $d\left(X_i, A_j\right)$ is an appropriate distance metric, such as the Euclidean distance from the location of facility X_i to the location of a customer at fixed point A_j. In general, location-allocation is a difficult global optimization problem which has been shown to be NP-hard [111].

Due to its practical applicability a number of algorithms have been used to solve the problem.

In 1963, Cooper solved certain classes of location-allocation problems by using both exact external equations and heuristic methods [35]. Since exact methods are limited to small problems and takes a significant amount of calculation time for larger problems, he used heuristic methods for larger problems. An important extension of the location-allocation problem was the introduction of weights on the customers. For example, let the suburbs of a city represent the customers and the population of each suburbs be the weights of the customer. Hence the placing of a facility (e.g. a shopping mall) in the suburbs depends upon the weights of the customer. This has became an interest among the researchers [72] because, the larger weights of the customers have the greater possibility of using the facility than the smaller one. Later on, a number of different location-allocation models are presented by various researchers [11, 137].

The Branch and bound algorithm is implemented for the problem size of the order of 25 fixed points and 1 to 5 facilities which provide an exact solution to the problem [99]. The main difficulty with continuous space location-allocation problems is the excessive run time for obtaining the required results. A number of algorithms have been developed to increase the efficiency of the problems [123].

Algorithms have been developed which are used to solve multi-source Weber problems using rectangular distance between facilities [107]. It consists of two steps. The first step is a set reduction method, that is equivalent to the p-median of a weighted connected graph. In the second stage a technique for solving the p-median problem has been used for obtaining the optimal locations and allocations.

Table 2.2: 51 Customer Problem

x coord.	y coord.	Wts.	x coord.	y coord.	Wts.	x coord.	y coord.	Wts.
depot			1.2400	6.6900	1	7.3100	1.6100	1
1.3300	8.8900	1	3.1300	1.9200	1	6.3700	7.0200	1
1.8900	0.7700	1	8.8600	8.7400	1	7.2300	7.0500	1
9.2700	1.4900	1	4.1800	3.7400	1	1.6800	6.4500	1
9.4600	9.3600	1	2.2200	4.3500	1	3.5400	7.0600	1
9.2000	8.6900	1	0.8800	7.0200	1	7.6700	4.1700	1
7.4300	1.6100	1	8.5300	7.0400	1	2.2000	1.1200	1
6.0800	1.3400	1	6.4900	6.2200	1	3.5700	1.9900	1
5.5700	4.6000	1	4.5300	7.8700	1	7.3400	1.3800	1
6.7000	2.7700	1	4.4600	7.9100	1	6.5800	4.4900	1
8.9900	2.4500	1	2.8300	9.8800	1	5.0000	9.0000	1
8.9300	7.0000	1	3.3900	5.6500	1	6.6300	5.2300	1
8.6000	0.5300	1	0.7500	4.9800	1	5.8900	8.0600	1
4.0100	0.3100	1	7.5500	5.7900	1	1.1300	5.2500	1
3.3400	4.0100	1	8.4500	0.6900	1	1.9000	8.3500	1
6.7500	5.5700	1	3.3300	5.7800	1	1.7400	1.3700	1
7.3600	4.0300	1	6.2700	3.6600	1	9.3900	6.4400	1

Methods based on the dynamic changing of the demand have also proposed [44]. A dynamic p-median problem is formulated where new facilities are built according to change in demand. Drezner put forward a new algorithm for the problem of two new facilities and Euclidean distance. This has proved to be a very good solution for a number of real world problems such as hospitals, schools and restaurant etc.

Location allocation problems can also be solved by using artificial neural network [64]. The main idea was to reduce the solution space by introducing some of the constraints explicitly in the calculation of the partition function. This algorithm has been tested on real size problems with excellent results.

A lot of work in this location-allocation process has been contributed by [61, 132, 62, 26] .

The genetic algorithm for continuous location-allocation problems was first used in 1996 [84]. Houck et al. implemented the problem by using the floating point (real numbers) representation. The facilities are represented by (x, y) pairs. The upper and lower bounds of x and y values are used to direct the search to possible values of x and y, respectively. Hence the individuals consists of p(x, y) pairs representing the locations of the facilities.

$Individual \rightarrow (x_1, y_1, x_2, y_2, \cdots, x_p, y_p)$.

These individuals are created randomly and evaluated using an evaluation function to determine their fitness. Crossover and mutation has been done to produce its offspring. It has shown promising and efficient solutions for larger problems.

The same idea has been used in this thesis with continuous Estimation of Distribution Algorithms. A probabilistic model has been used using the individuals to estimate the next generation.

In this thesis the widely used 51-Customer problem with a unit weight value for all the customers is considered. The 51-customer data values are shown in Table 2.2. The data A_i for the 51-Customers has been found in [48]. The best solutions for the 51-Customer problems listed in the PhD thesis of Krau are also known as global solutions to this problem [97].

For the comparison purposes of different heuristic methods in multisource Weber problem, 51 Customer problem has been used in [26, 30, 132]. This dataset has also been used as a test problem in vehicle routing problems, the coordination of production and distribution planning centers and other related problems [51, 29, 14].

2.5 Metaheuristics

In general, finding an exact global solution to an optimization problem is intractable. Therefore, many optimization algorithms have been developed that aim to provide good solutions in a reasonable amount of time.

Approximate algorithms are widely used to solve optimization problems. Approximate algorithms aim to find a good solution in a limited computation time without any guarantee of "efficiency" or "goodness" in the solutions [17]. Due to this, the use of approximate methods to solve optimization problems has received more and more attention in the last few decades.

Metaheuristics are a general class of approximate optimization algorithms, that typically make few assumptions about the problem and are generally and widely applicable [63, 17, 49, 109]. A metaheuristic is a way of solving very general computational problems by combining several heuristic techniques. Well-known examples of metaheuristics include evolutionary algorithms, other nature-inspired techniques (e.g. ant colony optimization and particle swarm optimization), simulated annealing and tabu search.

A Metaheuristic has the following characteristics [17]:

- Strategies guide the search space.
- Goal is to efficiently explore the search process.
- Adaptable to a wide range of problems.
- Easy to incorporate domain specific knowledge.
- Many metaheuristics contain some element of randomness.
- Easy to implement.

Metaheuristics are often applied to problems when :

- there is limited knowledge of the fitness function which means it is difficult to evaluate the gradient or higher derivative information.
- the fitness function may be discontinuous or noisy.
- the fitness function is suspected to contain any nasty feature e.g. many local minima, plateaus, ravines etc.

- the search space is high dimensional.

Evolutionary algorithms are a type of population-based metaheuristic. Population-based metaheuristics perform search processes which describe the evolution of a set of points in the search space [39]. Evolutionary algorithms are inspired by the natural evolution of species. In the natural world, the evolution of species is carried out by means of selection, recombination and random changes. Computers can use these criteria to optimize a function. The basic components of the algorithm are a population of individuals, a set of random operators to modify the individuals and a selection procedure over the individuals [102].

Evolutionary algorithms have been widely used and have shown impressive performance on many challenging problems. However, there are also challenges involved in their application. For example, evolutionary algorithms depend on the careful choice of several parameters (e.g. crossover and mutation operators and probabilities, selection operators, size of the population) to achieve good performance. Parameter selection is a very challenging problem (e.g. see [69, 104]).

Examples of evolutionary algorithms are the Genetic Algorithm, Genetic Programming, Evolutionary Programming, Evolution Strategy etc. [7, 10]. Among all these, Evolution Strategy (ES) is specifically designed for continuous optimization problems [8, 94]. The main components of ES include the coding, which is the representation of the individuals, fitness function, parent selection mechanism, definition of operators, selection and other algorithm parameters. It mainly uses self-adaptive mutation rates. Mutation is carried out by a random vector drawn from a Gaussian distribution. Covariance matrix adaptation is an example of evolution strategy (CMA-ES). It is a very efficient optimizer which adapts the covariance matrix of the search distribution in response to points sampled during search to try and achieve rapid progress in finding improved solutions over previously found solutions. In the CMA-ES, a set of new population is generated through the multi-variate normal distribution. It uses the mean of the selected population. The covariance matrix is built by considering how the mean of the population has shifted the previous generation. This information in addition to the cumulation shifts of the earlier generations obtain the evolution path. CMA-ES is factorized into the multiplication of a covariance matrix and is called global step size, which is also computed on the basis of the evolution path. The adjustment of the step size is based on a different adaptation principle. This path length helps in improving the convergence speed and global search capabilities at the same time [73, 74].

2.6 Estimation of Distribution Algorithms

EDAs are a recent development in the field of evolutionary algorithms, which is a population-based paradigm [102, 121]. In EDAs a population is approximated with a probability distribution and new candidate solutions are obtained by sampling this distribution. The aim is to avoid the use of arbitrary operators (such as mutation and crossover) in favor of explicitly modeling and exploiting the distribution of promising individuals. EDAs have been developed for both discrete and continuous problems. Reviews of EDAs can be found in [102, 125, 129]. The thesis focuses on continuous EDAs.

The general pseudo code framework for an EDA is presented in Algorithm 2.1.

Algorithm 2.1 General pseudocode framework for an EDA

```
Initialization: set $t = 0$, Generate initial population uniformly in search space
Evaluate $f(\mathbf{x})$ for each individual $\mathbf{x}$ in the current population
Select promising individuals
Build probabilistic model $p(\mathbf{x})$ based on selected individuals
Generate new population by sampling from $p(\mathbf{x})$
$t = t + 1$
Goto step 2 until a stopping criterion is met
```

The main components of EDAs are described as follows:

1. Population and its size

   In EDAs the initial population is typically generated randomly in the search space using uniform distribution within a pre-specified interval. The range of the interval is in a continuous rectangular space. The size of the population plays an important role in solving a problem with EDAs. If the population size is not large enough for some types of EDAs, the solutions found may not be of high quality. In addition, numerical issues may arise (discussed further in Section 2.7).

   Except for the initial generation, the population of the successive generation is generated using the probability density estimation model.

2. Fitness function

   The sample (population) of individuals on each generation of the algorithm is evaluated using the objective function.

3. Selection

   The main operator in EDAs is the population selection operator. The selection operator is used for selecting the best individuals from the population set. It is believed that the best fitness value is always achieved through selection. The general purpose of selection is to allow the search to progress in the direction of the better solutions in the current population.

   There are a number of selection operators like tournament selection, roulette wheel selection, truncation selection etc. Another approach is the Elitism selection, where the population of the next generation consists of more promising individuals of the current generation. Truncation selection is usually used in EDAs and is used in this thesis.

   Selection pressure plays an important role in achieving the best fitness values. If the selection ratio is very greedy, then fewer number of individuals are selected, which makes the search process more difficult and can lead to premature convergence. On the other hand, if the selection pressure is very low, then the individuals are spanned more widely, exploring across the search space, which in turn can take lots of time to achieve the desired result.

4. Probability Distribution

A probability density estimator is used to model the promising regions of the search space. The parameters of the model are typically estimated using maximum likelihood estimation, which is subsequently used for generating the candidate solutions for the next generation. The family and structure of the model used in an EDA is typically fixed (e.g. a set of Bernoulli distributions to generate the bit strings in the PBIL algorithm [12], or a factorized Gaussian distribution over a continuous search space in the UMDA$_c$ algorithm [102]).

5. Stopping Rule

   Stopping rule depends upon a number of criteria. An EDA stopped when,

   - it evaluates all the given number of function evaluations.
   - it reaches its global optimum.
   - the difference between the global optimum value and the evaluated fitness value is very small.
   - the model parameters are not showing any significant changes over a large number of generations.

Finding a good setting of the parameter values is important for getting an optimal solution, but defining the optimal values for the parameters is very difficult. The values of parameters of EDAs depend upon the optimization problems as well as type of algorithms involved. Considerable research has been done on the selection of different parameter values used in EDAs [57, 160].

It is possible to categorize continuous EDAs into three different types depending on the density estimation model used. One is based on parametric estimation, e.g. Gaussian distribution, which is the most widely used model [102, 43, 22]. Other models considered include semi-parametric, e.g. mixture models [58] and non-parametric, e.g. histograms and kernel density estimators [55, 159, 40, 23].

A Gaussian distribution over the variables is most commonly used in continuous EDAs, with several different covariance matrix structures ranging from diagonal (e.g. in the Univariate Marginal Distribution Algorithm (UMDA$_c$)) to full (e.g. Estimation of Multivariate Normal Density Algorithm (EMNA$_{\text{global}}$)).

The thesis focuses mainly on two simple EDAs, UMDA$_c$ and EMNA$_{\text{global}}$, although the contributions are likely to be expendable to other models. In addition to this, two new Gaussian based EDAs are developed and implemented in Chapter 6.

A number of advanced Gaussian based EDAs have been developed for enhancing the performance of the EDAs. For example the EDA framework with Model Complexity Control (EDA-MCC) [41] was proposed, which uses EMNA$_{\text{global}}$ for each subset of the variables, to perform well in high dimensional problems. A number of advanced Gaussian based EDAs like Adaptive-Variance scaling, Standard-Deviation Ratio triggering, Anticipated Mean Shift, which results in AMaLGAM uses the maximum likelihood estimates of EDAs [20]. These algorithms are further discussed in Section 2.7.

Table 2.3: Step 4 of UMDA$_\text{c}$

Step 4:
Build probabilistic model $p(\mathbf{x}) \leftarrow (\sigma^t, \mu^t)$

Although these advanced Gaussian based EDAs are performing better than the traditional EDAs, at a fundamental level the probabilistic model used is the same as in UMDA$_\text{c}$/ EMNA$_\text{global}$. Therefore the contribution of the thesis is relevant and applicable to these algorithms as well.

2.6.1 Univariate Marginal Distribution Algorithm

One type of parametric estimation based on Gaussian distributions is the Univariate Marginal Distribution Algorithm (UMDA$_\text{c}$). UMDA$_\text{c}$ in continuous domain which was introduced by Larrañaga et al. [101], is an example of a univariate algorithm where all the variables are independent of each other. It uses diagonal covariance matrix for estimating the population for the next generation. In UMDA$_\text{c}$ new individuals or candidate solutions are generated using a univariate Gaussian (Normal) Distribution. Here truncation selection has been used to retain the best individuals. These best selected individuals are used for calculating the mean and the standard deviation, which is defined in equation (2.7) and (2.8) respectively. These model parameters are used for estimating a new set of population for the next generation (see Table 2.3).

For continuous problems, UMDA$_\text{c}$ uses a product of Gaussian (Normal) distributions:

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i); \;\; p(x_i|\mu_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}(\frac{x_i-\mu_i}{\sigma_i})^2}$$

The two parameters to be estimated at each generation and for each variable are the mean, μ_i, and the standard deviation, σ_i [101]. Their respective likelihood estimates are:

$$\hat{\mu}_i = \bar{X}_i = \frac{1}{N} \sum_{j=1}^{N} x_{i,j} \tag{2.7}$$

$$\hat{\sigma}_i = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (x_{i,j} - \bar{X}_i)^2} \tag{2.8}$$

where $i = 1, \dots, n$ and N is the size of the selected population.

The pseudo code of UMDA$_\text{c}$ is same as the general pseudo code for EDA (Algorithm 2.1) except for step 4, where UMDA$_\text{c}$ builds the univariate probabilistic model.

UMDA$_\text{c}$ is one of the earliest and simplest EDAs. The only user-parameters to be specified are the population size and selection threshold. The algorithm has been studied theoretically [67, 68, 162, 163] as well as experimentally [102, 60, 116, 117]. The updates to the factorized model parameters are efficient and implementation is straightforward. However, the factorized model, by definition, makes no attempt to model dependencies between problem variables.

Table 2.4: Step 4 of EMNA$_{\text{global}}$

Step 4:
Build probabilistic model $p(\mathbf{x}) \leftarrow (\Sigma^t, \mu^t)$

2.6.2 Estimation of Multivariate Normal Algorithm

Estimation of Multivariate Normal Algorithm (EMNA$_{\text{global}}$) is a Gaussian-based, where all the variables are dependent on each other. In EMNA$_{\text{global}}$, the mean and covariance matrix are computed by maximum likelihood estimates [100]. At each generation, the vector of means, $\mu = (\mu_1, ..., \mu_n)$, and the variance-covariance matrix, Σ, whose elements are denoted by σ_{ik}^2 are estimated, where $i, k = 1, \ldots, n$. Their maximum likelihood estimates are calculated in the following way:

$$\hat{\mu}_i = \bar{X}_i = \frac{1}{N} \sum_{j=1}^{N} x_{i,j} \qquad i\text{=}1,...,n \tag{2.9}$$

$$\hat{\sigma}_i^2 = \frac{1}{N} \sum_{j=1}^{N} (x_{i,j} - \bar{X}_i)^2 \qquad i\text{=}1,...,n \tag{2.10}$$

$$\hat{\sigma}_{ik}^2 = \frac{1}{N} \sum_{j=1}^{N} (x_{i,j} - \bar{X}_i)(x_{k,j} - \bar{X}_k) \quad i\text{ , }k\text{=}1,...,n \;\; i \neq k \tag{2.11}$$

The pseudo code for EMNA$_{\text{global}}$ is the same as UMDA$_{\text{c}}$, except for step 4, which is replaced by the covariance matrix Σ, defined in equation (2.11) (see Table 2.4).

It is clear that the EMNA$_{\text{global}}$ model is more flexible than the UMDA$_{\text{c}}$ model (which it includes as a special case). Specifically, variable dependencies can be captured via pairwise covariance parameters. Experimentally, EMNA$_{\text{global}}$ has been shown to give improved performance over UMDA$_{\text{c}}$ on at least some specific problems where significant dependencies are known to exist [102, 60, 116, 117]. However, these results also show that the relative success of EMNA$_{\text{global}}$ is related to several factors, such as the "nature" of the dependencies between variables as well as the algorithm parameters chosen.

The main limitations of EMNA$_{\text{global}}$ are that, since it has more covariance parameters to estimate from the selected population, the computational cost is $O(n^2)$ which is higher than the computational cost for UMDA$_{\text{c}}$ i.e., $O(n)$. It also needs more storage space. Estimation of the full Σ matrix from the selected population can also lead to instabilities [43] (see below).

2.7 Issues Related to EDAs and their Remedies

EDAs have successfully solved many optimization problems which include various domains such as Artificial Intelligence [86, 87], Networking [164], Bioinformatics [5, 136, 6], Planning and Scheduling [165, 152, 31, 90], Industrial Design and Management [145, 91], Biology [134, 135, 130], Software [131] and Composite Materials [70]. However, researchers have also identified a number of issues in EDAs that sometimes cause difficulties.

In general, the success of an EDAs depends on how effective the model is at representing where the promising regions are in the search space to explore in subsequent generations. Another way to think of this is in terms of the direction in which the model should evolve to generate better individuals with higher probability. If this direction is not well-represented by the selected population, progress can be poor. In some landscapes, EDAs do not perform very well due to the premature shrinking of the variance at an exponential rate. For example, in slope-like regions of the search space, and on elliptical landscapes, premature convergence is likely to occur. In these cases, the selected solutions do not lead to the correct direction of descent in terms of sampling from the EDA model. Research has been conducted on these issues [20, 74, 19, 161, 129].

Another approach is the Variance scaling which stops the premature convergence of EDAs. This principle has already been proved to be very effective [19, 161]. If the variance of the estimated probability distribution is small, the EDA will not explore more, hence enlargement of variance leads to good exploration of EDAs in their search space [19]. According to Yuan and Gallagher, the variance obtained from the normal probability distribution of a continuous Gaussian EDA is scaled by a factor of 1.5 [161], which is used for finding an optimal solution for the Rosenbrock function. Although the results for the Rosenbrock function is satisfactory but the scaling factor cannot be used universally for all the problems. It requires further investigation on the nature of problems.

A simple and effective Adaptive-Variance-Scaling (AVS) is proposed for the use of an EDA by Bosman et al. [19]. Here the covariance matrix of the model at each generation is multiplied by a scaling factor. The best value obtained from the current generation is compared with the previous generation. If it improves, the current size of the variance allows for progress. In this way an enlargement of the variance may lead to better solutions in the next generation. It also leads to an increase in scaling factor. If the best fitness does not improve, it is assumed that search is "around an optimum". To facilitate convergence, the scaling factor becomes less than 2, allowing the variance of the search distribution to be reduced [24]. Correlation triggered also known as Standard Deviation Ratio (SDR) is done which is a test based on correlation between density and fitness to distinguish the situation where variance scaling is required and where variance scaling is not required. In later stages, it takes into account the mean shift for shifting the sampled solution. The direction which the solutions are moved to obtain better fitness is determined by the difference of the means in two subsequent generations. This is known as mean shift. This mean shift with a certain fraction of value is used for moving the new sampled solution of the current generation. This process is known as Anticipated Mean Shift (AMS) [19]. The combination of AVS, SDR and AMS leads to a technique known as AMaLGaM [21]. AMaLGaM uses the schemes of all three where it adaptively changes both the covariance matrix and the mean-shift, which leads to the prevention of premature convergence of the model.

Generally in practice, numerical issues can arise with the EDA model estimation problem [42]. The covariance matrix, which is estimated by using maximum likelihood in EMNA$_{\text{global}}$ is positive semi-definite by its definition. But in practice this is not guaranteed because of finite precision computation. There will be computation error or numerical issues arising when the sample used to estimate the model does not adequately span all dimensions of the search space, which is especially likely when the sample size is relatively small compared to the problem dimensionality.

Various methods and techniques have been proposed in order to avoid the ill-conditioned of the covariance matrix [42, 43]. Covariance Matrix Repairing (CMR) method was proposed by Dong et al. in 2007, where a positive value is added to the diagonal of the covariance matrix (Σ). The chosen value is the absolute value of the minimum eigenvalue (<0 if Σ is ill-conditioned) of Σ. Eigenanalysis has been done for repairing the ill-conditioned covariance matrix in various Multivariate Gaussian-based EDAs. A unified approach called Eigen Decomposition of EDAs (ED-EDA) has been proposed which uses different eigenvalue tuning strategies, to avoid the ill-conditioned of the covariance matrix [43]. One approach is the uniform eigenvalue scaling, where the covariance matrix is scaled by a certain value. This method expands or shrinks the covariance matrix. The AVS technique uses this approach. Another approach is non-uniform eigenvalue scaling. The first strategy of this approach is used in the eigenspace EDA (EEDA) algorithm by resetting the minimum eigenvalue. Hence, only one selective eigenvalue is enlarged while the others remain unchanged. Thus the shape of the distribution changes mostly in the direction of the smallest eigenvector. Another strategy is using minimum eigenvalues for shifting the covariance matrix as in the CMR technique. The experimental results in the literature also show that the values from different covariance repairing methods avoid the numerical difficulties and give good results with respect to best solution found as well as using a smaller sized population as compared to classical EDAs [42, 43].

For enhancing the performance of EDAs and overcoming the issues of ill-conditioned covariance matrix in high dimensional problems, different types of statistical methods have been adopted. For example, more recently, Dong et al.[41] proposed the EDA Model Complexity Control (EDA-MCC) to scale up continuous EDAs using a sparse covariance matrix, which is used for solving high dimensional problems very efficiently with less computational cost as well as less population sizes. EDA-MCC uses $\text{EMNA}_{\text{global}}$ for each subset of the variables. A set of artificial test problems were used for comparing EDA-MCC with UMDA_c, $\text{EMNA}_{\text{global}}$ and EEDA [151]. EDA-MCC is not performing better than the traditional EDAs in low dimensional problems where it has enough population size. EDA-MCC shows significantly good results in high dimensional problems. Other existing statistical methods have also been applied to control the amount of covariance/dependency modeling in EDAs. In [93], regularization techniques were adopted into EDAs. The resulting algorithm shows the ability to solve high dimensional problems with a comparable quality of solutions using much smaller populations. Both these algorithms are incomparable; authors of these algorithms use different set of problems with different bounds for comparison purposes.

A strong competitor of EDA in continuous optimization is CMA-ES. The review of relationship between EDAs and CMA-ES has been given in [129, 78, 77].

2.8 Why understanding and analyzing variables is important for optimization problems and EDAs

The difficulty of an optimization problem clearly relates to the sensitivity of the objective function and the interactions between problem variables. If a problem contains a variable that has little or no effect on the objective function value, then it may be inefficient to include this variable in the search. Alternatively, if the variables in a problem are independent, then a high dimensional problem may be more efficiently solved by decomposing it. Hence the performance of an algorithm will be related to how it handles the dependencies and importance of variables in optimization problems.

Fitness landscape analysis is an area of research that has developed theory and practical techniques for describing fitness landscapes (refer to [126, 140, 143, 49]) for recent reviews). A major theme in this work is the development of landscape features with the aim of characterizing landscapes in terms of these features. This could lead to insights into algorithm performance and behavior. For example, if a machine learning model can be used predict the performance of an algorithm from landscape features. While this work is interesting, it is largely algorithm-independent. The main focus in this thesis is to develop problem analysis techniques that are particularly relevant for the modeling in Gaussian EDAs.

EDAs focus on explicitly modeling dependencies between solution variables. The aim is to provide good solutions by possibly trying to capture dependency information between variables, using a probability density estimator over the search space. While they have shown promising experimental performance [75, 21], there is a lack of understanding about the relationship between the EDA dependency modeling and the nature of the underlying dependencies in an optimization problem. It is not clear whether a dependence between variables in a fitness function always leads to features in the landscape structure that can be captured and exploited by an EDA (e.g. elliptical basin of attraction).

In addition, as discussed in the previous section, there are also specific examples where basic EDAs do not perform well. If the full covariance matrix is estimated, numerical issues arise with the implementation of the algorithm, particularly in high dimensional problems. Hence, it is important to understand the impact of variable dependencies, their influence on the fitness function and its structure, as well as the relationship this has with the EDA model and performance.

2.9 Summary

This chapter has reviewed continuous black-box optimization, including different types of problems used in algorithm evaluation. Continuous Gaussian EDAs have also been introduced and several key issues identified that are relevant to this thesis. The chapter also highlighted the importance of understanding and analyzing variables for optimization problems and EDAs. The main contributions of the thesis are in the development and application of data-driven techniques to provide an improved understanding of the relationship between problem variables and EDAs. The next chapter develops the framework for this analysis.

Chapter 3

Methodology

3.1 Overview

This chapter presents the core methodology and techniques used in this thesis for analyzing the properties of the variables in continuous optimization problems and continuous EDAs. The different types of data-driven analysis for variables in optimization problems and Gaussian EDAs that are considered in this thesis are discussed here. Methods from different domains are unified to describe the properties of variables. As an example, correlation coefficients are used for calculating the correlation between the variables, whereas eigenanalysis is used for analyzing the key structural variables to the problem/model, and lastly, the Morris method is used for identifying important variables to the objective function.

3.2 Properties of Variables

As discussed in Chapter 2, to better understand the performance of EDAs, it is important to understand the interaction between the variables in a given optimization problem and the way that variables are handled and modeled in EDAs. It is proposed that three general variable properties need to be considered:

1. **Variable dependencies**. This is the interaction between subsets of variables given an optimization problem. As discussed in Chapter 2, this property has received a large amount of attention in the EDA literature. EDAs have different models for solving different problems which can capture different dependency structures. For example, EMNA$_{\text{global}}$ uses a full covariance matrix for modeling dependent variables, whereas UMDA$_c$ assumes each variable is independent (as per the reviews of EDAs discussed in Section 2.6 of Chapter 2). A number of algorithms have been developed, which perform modeling based mainly on the degree of correlations between the variables [41].

2. **Key structural variables**. This is intuitively defined as the subset of problem variables that together capture a significant part of the shape of some given distribution of interest over the search space. For example, if the EDA model distribution is to be studied, the key structural

variables are the variables that are most important in determining the directions in which the search will proceed when a sample is drawn from the model distribution.

3. **Important variables**. The aim is to determine, which problem variables have the most influence over the value of the objective function. Alternatively, for which variables is f most sensitive to changes in values?

Note that for each of the above property types, the intuitive notions used need to be defined precisely to produce a practical method of analysis. Furthermore, various techniques may be applied to study these properties. Each technique will involve different assumptions and offer different advantages and disadvantages. The techniques implemented to analyze these properties in this thesis are described in detail in the following subsections.

3.3 Data Sampling and Requirements

The methodology and analysis techniques developed here have the general assumption that a reasonably large sample of candidate solutions together with their corresponding objective function values is available or can be obtained with a reasonable amount of time or other resources. When analyzing problems (discussed in Chapter 4), the experiments here use a uniformly generated sample of points with truncation selection applied. Uniform Random Sampling is the standard approach in the Black Box optimization for analyzing problems and algorithms [47]. When analyzing EDAs, the sample is the selected population, often analyzed repeatedly over several generations of the algorithm. The methodology can be applied by other researchers to different data samples from optimization problems depending on the goal of the work. For example, if there is interest in the behavior (e.g. convergence) of an algorithm on a given problem in a specific, localized region of the search space, then sampling could be restricted to only points from this region. Alternatively, Markov Chain Monte Carlo algorithms can be used when sampling is desired from some probability distribution of interest over the search space [65]. Generally, it has been suggested, how MCMC can be used, but doing this is out of scope of the thesis.

All experiments in this thesis are executed on Windows General Purpose computation server (PE R720 and PE 2950) using MATLAB 2012 and MATLAB 2013. Statistical toolboxes have been used in the algorithms.

3.4 Sample Correlation Coefficients

Correlation is a standard statistical measure of the degree of linear dependency between variables. While there are many possible ways of estimating correlation from a data sample, Pearson's correlation coefficient is the most popular choice. For a set of data obtained from sampling or a set of population generated from EDA, the correlation coefficient for $(x_i, x_k), i = 1, \ldots, n$ and $k = 1, \ldots, n$, is calculated as:

$$r_{ik} = \frac{N(\sum_{j=1}^{N} x_{i,j}x_{k,j}) - (\sum_{j=1}^{N} x_{i,j})(\sum_{j=1}^{N} x_{k,j})}{\sqrt{[N\sum_{j=1}^{N} x_{i,j}{}^2 - (\sum_{j=1}^{N} x_{i,j})^2][N\sum_{j=1}^{N} x_{k,j}{}^2 - (\sum_{j=1}^{N} x_{k,j})^2]}}$$

where n is the dimensionality of the problem and N is the set of data obtained from sampling or a set of population generated from EDA.

The r_{ik} has a value ranging between -1 and 1. If r_{ik} has a positive value, then it shows that x_i and x_k tend to increase and decrease together, and if it has a negative value then x_i and x_k tend to move in opposite directions [80].

3.4.1 Use of Correlation Coefficient in Problems and EDAs

Since the correlation always finds the linear dependency between a pair of variables, the absolute value of the correlation coefficient is examined and this quantifies how one variable is dependent on the other variable. The correlation coefficient between two variables is always a single number, but for larger values of n, the correlation coefficient can be calculated and presented in a matrix format, with each r_{ik} being the correlation coefficient between two consecutive variables. The correlation coefficient between variables form a symmetric matrix [3].

A threshold value must be chosen to set the variables correlated and non-correlated. A threshold value of 0.3 has been set here for measuring the high correlation between variables [41]. In a 2-dimensional problem, the correlation between two variables has been determined by checking the correlation coefficient values. The variables x_1 and x_2 are correlated to each other if the correlation coefficient is greater than 0.3 otherwise they are non-correlated. For the dimension greater than two, the correlation is found by visualizing the correlation coefficient matrix, which is represented in the form of a color image. Although the correlation is calculated for pairwise variables, the correlation among more than two variables can also be established, when some blocks in the image have colors with high intensity (above the threshold value).

Figure 3.1 shows the visualization of correlation between variables in a 10D Rosenbrock function. 300000 sample points are generated using the uniform random sampling (detailed discussion in section 4.2 of Chapter 4). A selection ratio 0.01 is set. The set of selected sample points are used for measuring correlation between variables. Figure 3.1 shows that 1st and 2nd variable are correlated to each other because the intensity of the color (yellow and green) shown in the figure is greater then 0.3 as well as correlated to all other variables in the problem because the intensity of the color varies from green to light blue, which can be seen by examining the intensity of colors in the figure. But for rest of the variables, there is not so strong correlation between variables because the intensity of the color is less than 0.3.

3.5 Eigenanalysis

Eigenanalysis is a fundamental technique based on linear algebra [85]. It deals with the understanding of problems and finding a solution based on the coordinate system. This is also known

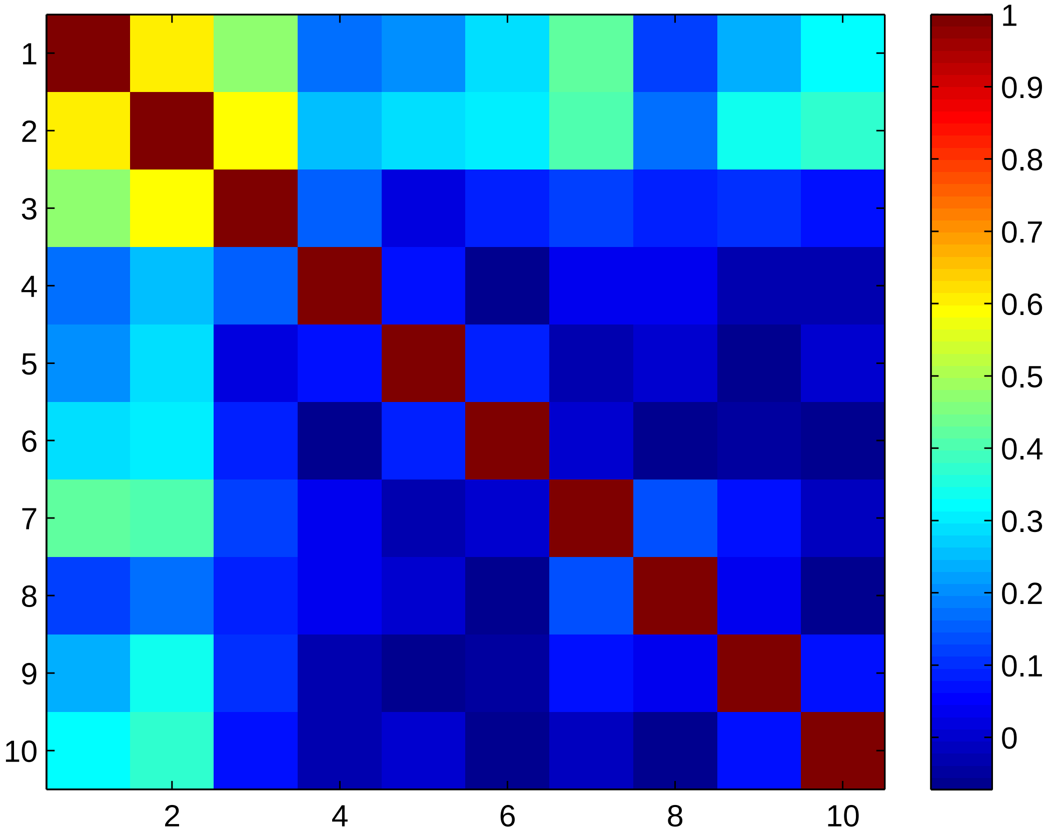


Figure 3.1: Visualization of correlation between variables in a 10D Rosenbrock Function

as "*the method of simplifying coordinates*". Eigenanalysis is a well known technique which has already been used in various research fields [153, 119, 96]. The eigenanalysis has been conducted on a symmetric matrix. Hence a set of data has been used for calculating the covariance matrix, which in turn is used for performing eigenanalysis.

To illustrate the computation of eigenanalysis, let us consider a matrix X consisting of a set of selected sample points, of size $N \times n$.

The entries of the covariance matrix (Σ) of X is calculated as follows,

$\Sigma_{i,k} = \frac{1}{N}\sum_{j=1}^{N}\left(x_{i,j} - \bar{X_i}\right)\left(x_{k,j} - \bar{X_k}\right)$

where $\Sigma_{i,k}$ is the covariance value between variable x_i and x_j, and

$\bar{X_i} = \frac{1}{N}\sum_{j=1}^{N} x_{i,j}$, where i = 1 to n and $j = 1, \ldots, N$ and $\bar{X_k} = \frac{1}{N}\sum_{j=1}^{N} x_{k,j}$, where k = 1 to n and $j = 1, \ldots, N$.

The Σ can be defined as

$$\Sigma = \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} & \cdots & \Sigma_{1,n} \\ \Sigma_{2,1} & \Sigma_{2,2} & \cdots & \Sigma_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ \Sigma_{n,1} & \Sigma_{n,2} & \cdots & \Sigma_{n,n} \end{bmatrix}$$

The covariance matrix Σ, is used to calculate the eigenvectors and eigenvalues, i.e. $\Sigma\mathbf{v} = D\mathbf{v}$, where D is a $n \times n$ matrix, whose diagonal element contains the eigenvalues of the covariance matrix Σ, denoted as $\lambda_1, \ldots, \lambda_n$, where $\lambda_1 > \lambda_2 > \ldots > \lambda_n$. $\mathbf{v}$ is a $n \times n$ matrix which contains n column vectors, each having n elements. Each vector (for example $\mathbf{v_i}$) is known as an eigenvector and each $\mathbf{v_i}$ of the covariance matrix (Σ) is associated with its corresponding eigenvalue (for example λ_i). Each eigenvector has n number of vector coefficients.

While performing eigenanalysis, a new coordinate system has been formed, such that the greatest variance of the data lies on the first eigenvector, the second greatest variance on the second eigenvector, which is orthogonal to the first one and so on. Hence λ_1 captures most of the variability of the data. λ_1 is the first eigenvalue, also known as dominant eigenvalue since the variance is greater than all other eigenvalues. The eigenvector associated with λ_1 is known as first eigenvector or with dominant eigenvector.

If there is equal spread between λ's, then, not 1 axis is dominant in this case. Hence the first eigenvalue will not give much adequate information about the key structural variables. If there is much difference between the λ_1 and λ_2, then λ_1 is considered as the first eigenvalue which captures most of the variability in data. The first eigenvector is placed in a new coordinate system which has the greatest variance of data. The length of the eigenvectors is of no consequence but its direction is important. Hence to keep eigenvectors standard, the length of the eigenvector is set to 1. The first eigenvector is defined as follows: $\mathbf{v}_1 = (c_1x_1, c_2x_2, \ldots, c_nx_n)$, where $c_1, c_2, \ldots, c_n$ are the coefficients of first eigenvector along $x_1, x_2, \ldots, x_n$ axis respectively.

For example, the Figure 3.2 shows the eigenanalysis on a 2D Rotated Ellipsoid function. $\mathbf{v}_1$ and $\mathbf{v}_2$ are the two eigenvectors, which are shown in Figure 3.2. λ_1 associated with $\mathbf{v}_1$ captures most of the variance of data. Hence, λ_1 is the dominant eigenvalue. For this example (refer Figure 3.2), the values of the components of $\mathbf{v}_1$ are $c_1 = 0.5277$ and $c_2 = 0.8494$.

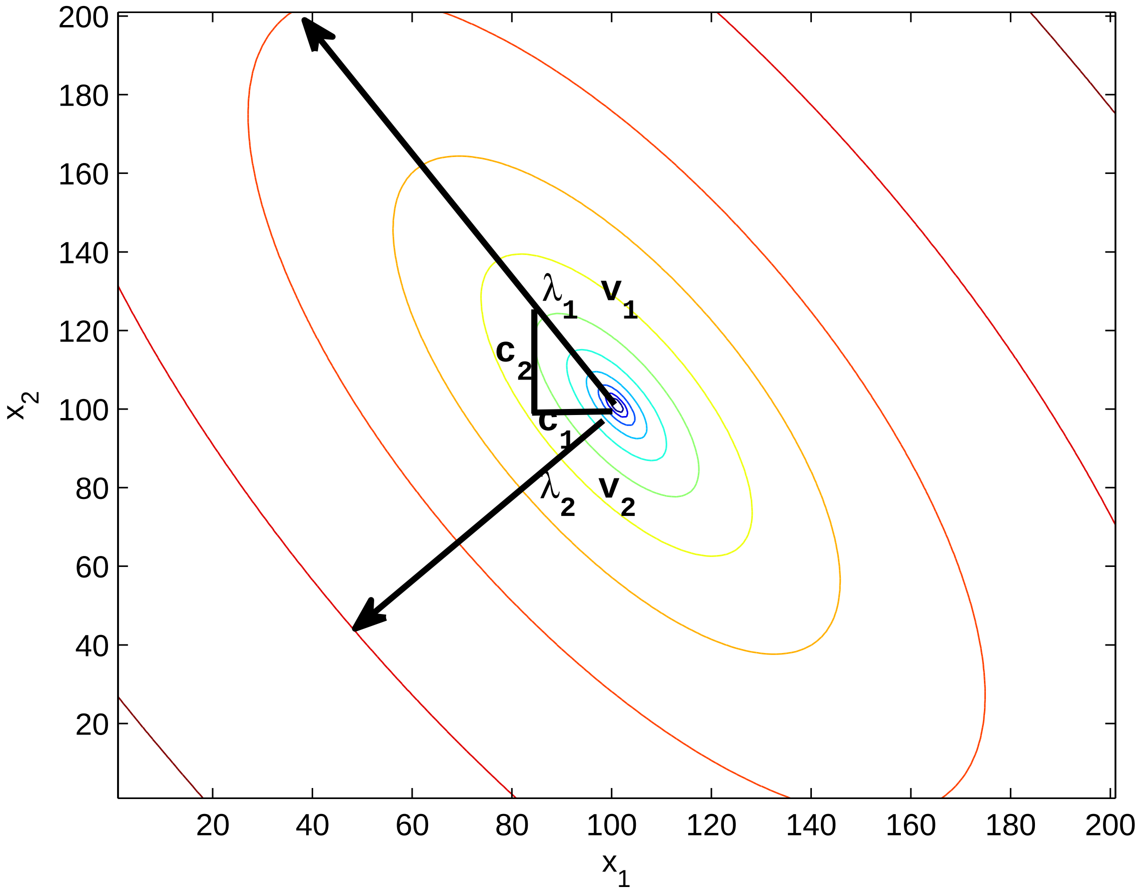


Figure 3.2: Explanation of Eigenanalysis, using 2D Rotated Ellipsoid function

One of the main application of eigenanalysis is principal component analysis (PCA). PCA is utilized to find out the subspace on which to project the data that captures most of the information about the structure of the problem [124]. This is used in dimensionality reduction in multivariate analysis and its application includes image processing, visualization, pattern recognition etc. [120, 32, 103]. PCA can be achieved through the eigendecomposition of the covariance matrix, which results in eigenvalues and its associated eigenvectors called as principal components ([92, 155, 88]).

3.5.1 Calculation of Key Structural Variables using Eigenanalysis in Problems and EDAs

The key structural variables of a problem can be calculated using the coefficients of $\mathbf{v}_1$, which can be represented as $(c_1, \ldots, c_n)$. If $c_i = 1$, where i is any value between 1 and n, and the other c_j's $j \neq i$, must be close to zero, then the vector is aligned/parallel to the x_i direction in the original space, which captures most of the relevant information about the structure of the problem. Hence, x_i is the key structural variable to the problem.

But in most of the cases, the value of c_i is not equal to 1. Therefore a set of larger vector coefficients has been chosen, whose associated variables mainly contain the key information about the structure of the problem, hence referred to key structural variables.

Therefore to calculate a threshold value for the vector coefficients for identifying key structural variables in problems and EDAs, the unit eigenvector has been analyzed. The explanation is as follows:

The coefficients of the unit eigenvector are perpendicular to each other. For an n-dimensional problem, the equation is stated as follows:

$$c_1^2 + c_2^2 + \ldots + c_n^2 = 1 \tag{3.1}$$

where c_i are the coefficients of the vector. When all the coefficients are sharing similar values, equation 3.1 becomes,

$$1 = n \times c_i^2 \tag{3.2}$$

where c_i are the coefficients of the vector. Thus, the value of c_i is equal to $\frac{1}{\sqrt{n}}$. Therefore, the variables associated with the coefficients which are greater than or equal to $\frac{1}{\sqrt{n}}$, where n is the dimensionality of the problem, are the key structural variables to the problem.

Hence, the threshold value for the 2D Rotated Ellipsoid function, shown in Figure 3.2, is $\frac{1}{\sqrt{2}}$ which is used for identifying key structural variables. From the figure, the value of c_1 is greater than the threshold value, therefore the first eigenvector is pointing more towards the x_2 axis than the x_1 axis. Hence, x_2 is the key structural variable to the 2D Rotated Ellipsoid function. Techniques have been mentioned in section 2.7, for example, EEDA.

3.6 Morris Method

Input variables which are providing considerable impact to the objective function are termed as important variables. This process can be easily achieved through the screening techniques. A number of techniques for identifying variable importance have been developed and applied in the field of experimental design and particularly in model-based engineering design. A simple technique proposed by Morris in 1991 [118] is based on measuring the mean and standard deviation of perturbations of individual variables for a given problem, calculated via so-called elementary effect terms.

The elementary effect for the ith variable, $E_i(\mathbf{x})$, is defined as follows. Let Δ be a pre-determined amount to perturb each variable. For a given $\mathbf{x}$

$$E_i(\mathbf{x}) = \frac{f(x_1, x_2, \ldots, x_{i-1}, x_i + \Delta, x_{i+1}, \ldots., x_n) - f(\mathbf{x})}{\Delta} \tag{3.3}$$

where $\mathbf{x}$ = $(x_1, x_2, \ldots., x_n)$ is a given starting or "baseline" vector in the solution space. The perturbations, Δ are by default assumed to come from a set of points over a given parameter/variable space obtained using a full factorial sampling grid of some fixed resolution or increment size. In other words, for each variable x_i, over some fixed range and increment size, the value of x_i is changed and f is recalculated, producing a sample or set of values of $E_i(\mathbf{x})$ [54].

Given a set of elementary effect values, the mean, $\overline{E}_i(\mathbf{x})$, and standard deviation, $std(E_i(\mathbf{x}))$ over the sample can be calculated. It has been predicted that, in some cases there may exist cancellation effects of the ith variables, legacy of calculating the mean of the elementary effect, which ultimately provides a null effect to the objective function. Hence, a new version, i.e. the mean of the absolute value of the elementary effect term defined as $\overline{E}_i^*(\mathbf{x})$, where

$$E_i^*(\mathbf{x}) = abs(\frac{f(x_1,x_2,...,x_{i-1},x_i+\Delta,x_{i+1},....,x_n)-f(\mathbf{x})}{\Delta}) \tag{3.4}$$

[27]. This is the enhanced version of the Morris method and has proved to be very effective in very high dimensional problems [95]. A high value of $\overline{E}_i^*(\mathbf{x})$ $\left(where, \overline{E}_i^*(\mathbf{x}) = \frac{1}{n}\sum_i^n E_i^*\right)$ then indicates a variable that has a large average influence on the value of f. A high value of $std(E_i(\mathbf{x}))$ indicates that the variable x_i has a fluctuating influence on the value of f [118]. Either or both may be important. In engineering design, these values typically are examined manually to inform decisions, e.g. about which variables to include in a model [54].

A wide variety of applications in various domains use the Morris method for screening techniques [128, 33]. Due to its wide application, considerable research has been done in this field to enhance the original Morris method [27, 37, 139].

3.6.1 Calculating Elementary Effect using Mean of the Selected Sample Points and Identifying the Important Variables to the Problems and EDAs Model

For calculating the elementary effect, the value of Δ has to be specified, such that the value of $x_i \pm \Delta$ is in search space S. One possibility of the value of Δ is the mean of the selected sample size, which is calculated for each dimension x_i. The total sample points are then generated by creating new solution vectors where the mean value is substituted in turn for each problem variable (e.g. $\mathbf{x^i} = x_i, \ldots, x_{i-1}, m_i, x_{i+1}, \ \ldots, \ x_n$), which are again evaluated using the fitness function f. This fitness function is used for calculating the Elementary effect, that is the perturb of fitness function before and after substituting the mean value. In general, for a set of N selected sample points from a n dimensional problem, the method is generating n number of N sample points. The fitness function of the ith number of N sample points is subtracted from the original fitness function for calculating the elementary effect of x_i, where the value of $i = 1, \ \ldots, \ n$.

Mean and standard deviation of the elementary effect are calculated. This screening of important variables can be done manually by plotting the values of the mean and standard deviation of the elementary effect in $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ plane. A high value of mean indicates the variable has high importance to the objective function whereas a high value of standard deviation indicates that the variable is involved in interaction or curvature effects. Therefore, the variable which is most influential or important to the objective function has a high value of mean as well as a high value of standard deviation of the elementary effect. It should be noted that using mean and standard deviation of the elementary effects provides only a coarse summary of the (possibly complex) relationship between the variables and objective function in a given problem. When the Morris method has been previously used [54], only a general interpretation of this relationship is attempted.

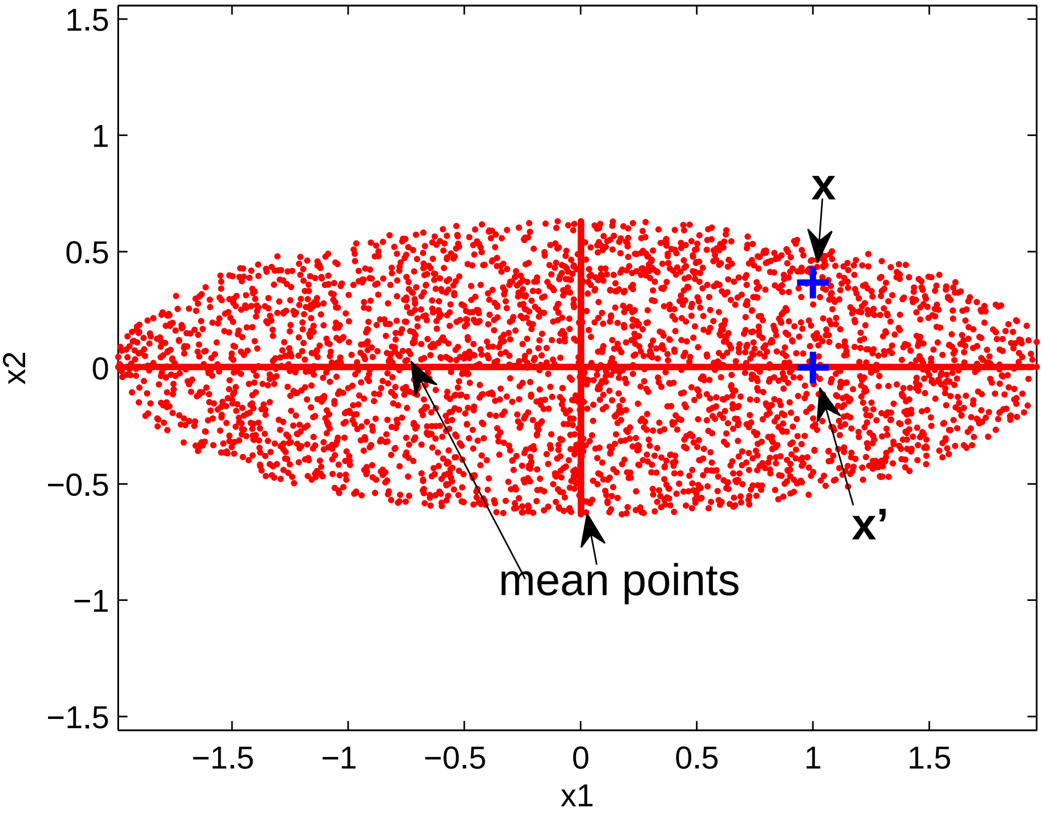


Figure 3.3: Depiction for Calculating the Elementary Effect in 2D Ellipsoid Function

The main goal of this screening exercise is to rank the factors in order of importance, and to identify the subset of the most important. However, the main limitation of this method is that the individual interactions among factors cannot be estimated. This method can provide whole interactions but is unable to give any specific information about how much they are interacted [133].

Figure 3.3 explains the calculation of elementary effect in a 2D Ellipsoid function. The mean of the selected population is calculated. The value at each variable is replaced with their corresponding mean values. For example, x is the point in the selected population, and the mean of the selected population along x_1 axis is x'. The difference between the fitness function which involved x and x' is the elementary effect for the point x in the x_1 direction in the search space. Similarly, the elementary effect of the sample point is calculated along the x_2 direction. Hence for a single sample point there are 2 set of elementary effect values, one along the x_1 axis and another along the x_2 axis. A similar procedure has been performed for each sample point in the search space for evaluating the elementary effect values. The elementary effect values are arranged according to the axes and are also arranged in a matrix format. The estimated mean and standard deviation of the elementary effect have been calculated which when combined give information about the important variable to the objective function.

3.6.2 Example for Determining Important Variables

Table 3.1: Selected population

x_1	x_2	x_3	x_4	x_5	x_6	Fitness
0.7793	0.3630	0.8382	0.0313	0.0894	0.0504	1.4532
0.3371	0.8022	0.0745	0.7597	0.2858	0.3576	1.5494
0.1385	0.5326	0.6615	0.4538	0.1956	0.8678	1.7377
0.9253	0.8166	0.1832	0.3396	0.0450	0.4432	1.8704
0.9102	0.0815	0.1560	0.2427	0.9190	0.5384	2.0528

Let us take a matrix of selected population X of size 5×6. The X is represented in Table 3.1 with its fitness values (here the Sphere function has been taken). The mean, **m**, of the selected population is as follows: $mean(x_1) = 0.6181$, $mean(x_2) = 0.5192$, $mean(x_3) = 0.3827$, $mean(x_4) = 0.3654$, $mean(x_5)$ = 0.3070, $mean(x_6)$ = 0.4515.

This mean vector is used for expanding the population. Table 3.2 shows the expanded population. The mean value is substituted for each problem variable x_i, which is again evaluated using the fitness function. The fitness values of this expanded population are shown in Table 3.3. Finally, in Table 3.4, the fitness values found from Table 3.3 for each section of perturbed values of variables are subtracted from the fitness values found from Table 3.1. This is called the Elementary effect values. The estimated mean ($\overline{E}_i(\mathbf{x})$) and standard deviation ($std(E_i(\mathbf{x}))$) of the elementary effect values are listed in Table 3.5. In this table the enhanced version of mean of elementary effect $\overline{E}_i^*(\mathbf{x})$ is also listed.

3.7 Summary and Further Work

In this chapter, three different but complementary methods have been proposed to analyze and understand three fundamental properties of variables in black-box optimization problems and EDAs: dependencies; key structural variables and variable importance to f. For analyzing the interdependency between the variables, correlation coefficients will be considered. Eigenanalysis will be used for determining the key structural variables of both the optimization problems and algorithms. Finally, the Morris method from sensitivity analysis will be used for identifying important variables for the objective function. These methods address the research gap discussed in Section 2.8 of Chapter 2.

Discussion of the implementation of these methods on optimization problems and EDAs is given in Chapters 4 and 5, together with experimental results and analysis. A graphical summary of the framework, analysis and contributions of subsequent Chapters (4-6) of the thesis is shown in Figure 3.4.

Table 3.2: Expand population using mean (**m**)

x_1	x_2	x_3	x_4	x_5	x_6
0.6181	0.3630	0.8382	0.0313	0.0894	0.0504
0.6181	0.8022	0.0745	0.7597	0.2858	0.3576
0.6181	0.5326	0.6615	0.4538	0.1956	0.8678
0.6181	0.8166	0.1832	0.3396	0.0450	0.4432
0.6181	0.0815	0.1560	0.2427	0.9190	0.5384
0.7793	0.5192	0.8382	0.0313	0.0894	0.0504
0.3371	0.5192	0.0745	0.7597	0.2858	0.3576
0.1385	0.5192	0.6615	0.4538	0.1956	0.8678
0.9253	0.5192	0.1832	0.3396	0.0450	0.4432
0.9102	0.5192	0.1560	0.2427	0.9190	0.5384
0.7793	0.3630	0.3827	0.0313	0.0894	0.0504
0.3371	0.8022	0.3827	0.7597	0.2858	0.3576
0.1385	0.5326	0.3827	0.4538	0.1956	0.8678
0.9253	0.8166	0.3827	0.3396	0.0450	0.4432
0.9102	0.0815	0.3827	0.2427	0.9190	0.5384
0.7793	0.3630	0.8382	0.3654	0.0894	0.0504
0.3371	0.8022	0.0745	0.3654	0.2858	0.3576
0.1385	0.5326	0.6615	0.3654	0.1956	0.8678
0.9253	0.8166	0.1832	0.3654	0.0450	0.4432
0.9102	0.0815	0.1560	0.3654	0.9190	0.5384
0.7793	0.3630	0.8382	0.0313	0.3070	0.0504
0.3371	0.8022	0.0745	0.7597	0.3070	0.3576
0.1385	0.5326	0.6615	0.4538	0.3070	0.8678
0.9253	0.8166	0.1832	0.3396	0.3070	0.4432
0.9102	0.0815	0.1560	0.2427	0.3070	0.5384
0.7793	0.3630	0.8382	0.0313	0.0894	0.4515
0.3371	0.8022	0.0745	0.7597	0.2858	0.4515
0.1385	0.5326	0.6615	0.4538	0.1956	0.4515
0.9253	0.8166	0.1832	0.3396	0.0450	0.4515
0.9102	0.0815	0.1560	0.2427	0.9190	0.4515

Table 3.3: Calculate fitness values

	x_1	x_2	x_3	x_4	x_5	x_6	Fitness
Perturbed for x_1	*0.6181*	0.3630	0.8382	0.0313	0.0894	0.0504	1.2279
	0.6181	0.8022	0.0745	0.7597	0.2858	0.3576	1.8178
	0.6181	0.5326	0.6615	0.4538	0.1956	0.8678	2.1005
	0.6181	0.8166	0.1832	0.3396	0.0450	0.4432	1.3962
	0.6181	0.0815	0.1560	0.2427	0.9190	0.5384	1.6063
Perturbed for x_2	0.7793	*0.5192*	0.8382	0.0313	0.0894	0.0504	1.5909
	0.3371	*0.5192*	0.0745	0.7597	0.2858	0.3576	1.1754
	0.1385	*0.5192*	0.6615	0.4538	0.1956	0.8678	1.7236
	0.9253	*0.5192*	0.1832	0.3396	0.0450	0.4432	1.4731
	0.9102	*0.5192*	0.1560	0.2427	0.9190	0.5384	2.3157
Perturbed for x_3	0.7793	0.3630	*0.3827*	0.0313	0.0894	0.0504	0.8970
	0.3371	0.8022	*0.3827*	0.7597	0.2858	0.3576	1.6903
	0.1385	0.5326	*0.3827*	0.4538	0.1956	0.8678	1.4466
	0.9253	0.8166	*0.3827*	0.3396	0.0450	0.4432	1.9832
	0.9102	0.0815	*0.3827*	0.2427	0.9190	0.5384	2.1749
Perturbed for x_4	0.7793	0.3630	0.8382	*0.3654*	0.0894	0.0504	1.5857
	0.3371	0.8022	0.0745	*0.3654*	0.2858	0.3576	1.1058
	0.1385	0.5326	0.6615	*0.3654*	0.1956	0.8678	1.6653
	0.9253	0.8166	0.1832	*0.3654*	0.0450	0.4432	1.8886
	0.9102	0.0815	0.1560	*0.3654*	0.9190	0.5384	2.1274
Perturbed for x_5	0.7793	0.3630	0.8382	0.0313	*0.3070*	0.0504	1.5394
	0.3371	0.8022	0.0745	0.7597	*0.3070*	0.3576	1.5620
	0.1385	0.5326	0.6615	0.4538	*0.3070*	0.8678	1.7937
	0.9253	0.8166	0.1832	0.3396	*0.3070*	0.4432	1.9626
	0.9102	0.0815	0.1560	0.2427	*0.3070*	0.5384	1.3024
Perturbed for x_6	0.7793	0.3630	0.8382	0.0313	0.0894	*0.4515*	1.6545
	0.3371	0.8022	0.0745	0.7597	0.2858	*0.4515*	1.6254
	0.1385	0.5326	0.6615	0.4538	0.1956	*0.4515*	1.1885
	0.9253	0.8166	0.1832	0.3396	0.0450	*0.4515*	1.8778
	0.9102	0.0815	0.1560	0.2427	0.9190	*0.4515*	1.9667

Table 3.4: Elementary Effects $E_i(\mathbf{x})$=Perturbed Fitness - Original Fitness

x_1	x_2	x_3	x_4	x_5	x_6
-0.2253	0.1378	-0.5561	0.1326	0.0862	0.2013
0.2684	-0.3740	0.1409	-0.4436	0.0125	0.0760
0.3628	-0.0141	-0.2911	-0.0724	0.0560	-0.5492
-0.4742	-0.3973	0.1129	0.0182	0.0922	0.0074
-0.4464	0.2629	0.1221	0.0746	-0.7503	-0.0860

Table 3.5: Estimated Mean and Standard Deviation of the Elementary effect

	x_1	x_2	x_3	x_4	x_5	x_6
$\overline{E}_i(\mathbf{x})$	-0.1029	-0.0769	-0.0943	-0.0581	-0.1007	-0.0701
$\overline{E}_i^*(\mathbf{x})$	0.3554	0.2372	0.2446	0.1483	0.1994	0.1840
$std(E_i(\mathbf{x}))$	0.3955	0.2985	0.3151	0.2284	0.3645	0.2876

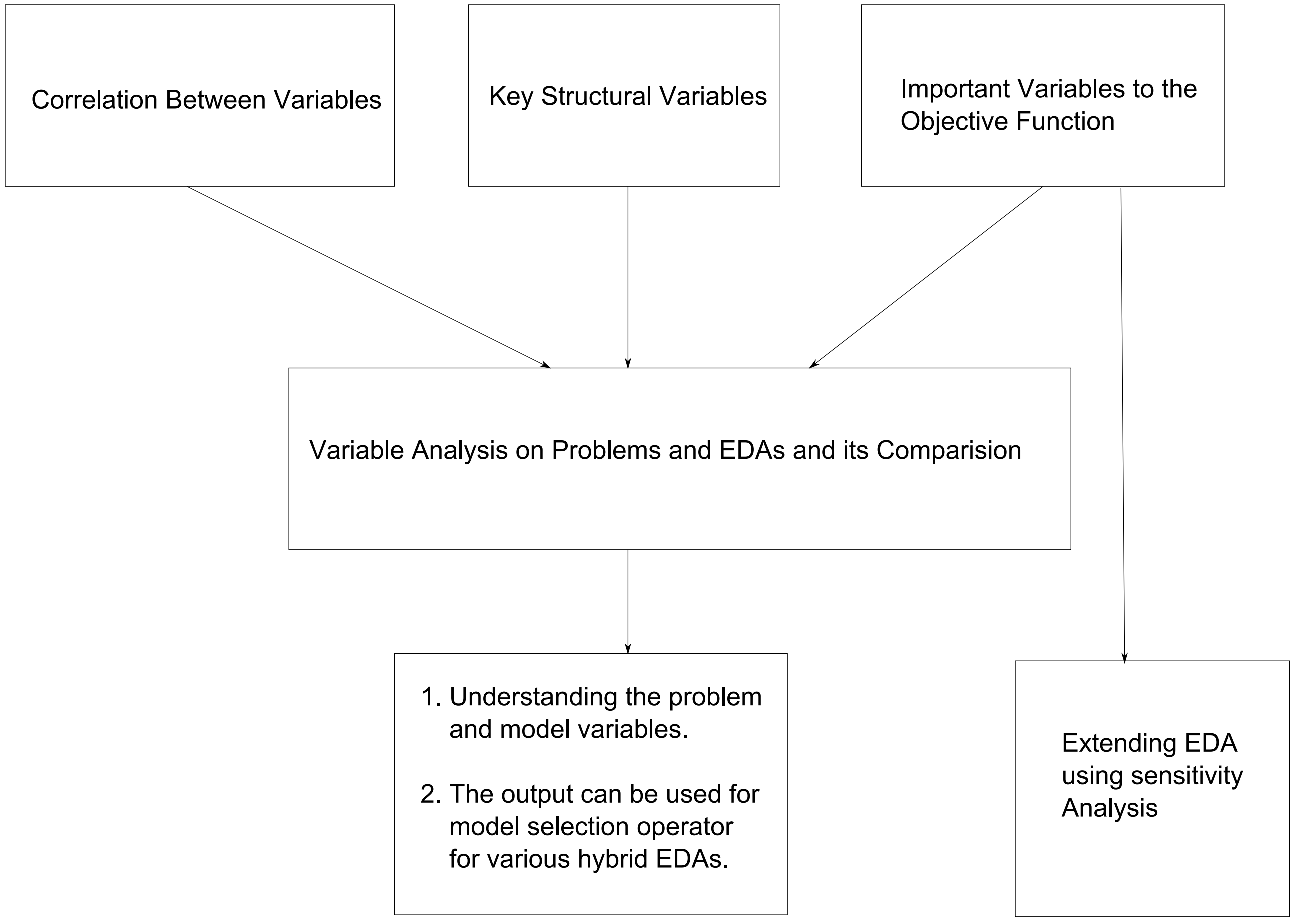

Correlation Between Variables
Key Structural Variables
Important Variables to the Objective Function
Variable Analysis on Problems and EDAs and its Comparision
1. Understanding the problem and model variables.
2. The output can be used for model selection operator for various hybrid EDAs.
Extending EDA using sensitivity Analysis

Figure 3.4: Understanding and Analyzing Variables in Problems and EDAs

Chapter 4

Variable Analysis in Continuous Optimization Problems

4.1 Overview

In order to understand an optimization problem, in particular how the properties of a problem relate to algorithm performance, understanding and analyzing the properties of the variables is important. In the previous chapter, it was proposed that three types of analysis can be used for analyzing Optimization problems and Gaussian based EDAs, namely dependencies, key structural variables and important variables. In this current chapter, analysis of variables on a set of optimization problems using these three techniques have been done. Different classifications have been completed based on these properties and problems are placed under these classifications, which will be used further for comparison purposes with EDAs in Chapter 5. Experiments and analyses have been done on both artificial and real world problems.

4.2 Uniform Random Sampling

The analysis proposed in Chapter 3 uses a sample of candidate solutions and their fitness values. In this chapter, the fitness landscape of a number of optimization problems have been analyzed directly via uniform random sampling. For the problems considered, boundaries are specified and the points are randomly generated within the bounds. For the solution vectors, variable values are generated independently.

For an n dimensional problem, the set of M points is represented as follows:
($\mathbf{x}^1$, $\mathbf{x}^2$, $\ldots$, $\mathbf{x}^M$), where $\mathbf{x}^i = (x_1, x_2, \ldots, x_n)$.

The fitness values are then calculated for each set of sample points. Let τ be the selection ratio. Let $N = (M * \tau)$[1] set of best points have been selected based on the fitness function and stored in matrix X, where X= ($\mathbf{x}^1$, $\mathbf{x}^2$, $\ldots$, $\mathbf{x}^N$), and $\mathbf{x}^i = (x_1, x_2, \ldots, x_n)$.

This set of selected points arranged in a matrix format (X) is used for analyzing different properties of the continuous optimization problems using different methodologies.

[1] Round if $(M * \tau)$ is not an integer.

4.3 Optimization Problems and Parameter Settings

In these experiments, a set of artificial test problems as well as real world problems was used, which has discussed in Chapter 2. The set of Artificial test problems consists of 10D Sphere function, 2D Elliptical function, 10D Rosenbrock function and 10D Rotated Ellipsoid function. These functions are described in Section 2.4.1. The analysis of other problems from the list are discussed in section 5.8.1.

In addition to this, Circle in a Square packing (CiaS) problems with n_c equal to 5 and 25 have been analyzed. This problem has been defined in Section 2.4.2. Furthermore, CiaS are constrained optimization problems with a hard box-bound, that is the feasible solution must be inside and including the boundary of the unit square. Therefore, while implementing CiaS problems, the values of $\mathbf{w^i}$ of Equation 2.5 have been modified. That is, $\forall\ \mathbf{w^i} = (w_1,\ w_2),\ i = 1,\ \ldots,\ n_c$, if $w_1 < 0$, then set $w_1 = 0$ or if $w_1 > 1$, then set $w_1 = 1$, where n_c is the number of circles in the unit square. This process has been repeated for w_2 also.

Finally, 51-customer location-allocation problems having the number of facilities 5 and 25 were analyzed here (see Section 2.4.3 for more details of location-allocation problems).

For any n dimensional problem, $10000 \times n$ number of sample points were generated using uniform random sampling. All sample points were evaluating the fitness function and the best selected sample points with the selection ratio (τ) = 0.01 were chosen for analyzing the properties of the variables in continuous optimization problems.

These selected sample points were used for analyzing the correlation between variables, and key structural variables as well as important variables to the objective function in continuous optimization problems.

4.4 Correlation between Variables in Continuous Optimization Problems

The selected sample points are used for calculating correlation coefficients between the variables in the optimization problems discussed in Section 2.4. The correlation and the implementation of correlation in problems have been discussed in Section 3.4. The classification of problems is based on the values of the estimated correlation coefficients (r). Problems can be classified into two types, that is, problems having:

- weak correlations between variables

- correlations between variables.

This classification is clearly dependent on the threshold value used (here abs(r)=0.3, refer to section 3.4). Correlation values are also visualized directly using matrix plots.

4.4.1 Weak Correlations between Variables

Problem variables which have all correlation coefficient values less than 0.3 (threshold value of the correlation coefficient discussed in Section 3.4.1) are considered as non-correlated variables. Examples of non-correlated variables in problems are as follows:

- 2D Elliptical function

  For the 2D Elliptical function, the correlation coefficient between variables is 0.0146, which indicates that there is weak correlation between the variables ($r_{12} < 0.3$). This is to be expected since, from the definition of the function, the variables are independent. In the limit of a large sample, the estimated value will approach 0.

- 10D Sphere function

  Figure 4.1 shows the correlation graph for the 10D Sphere function. The plot is a color representation of the values in the correlation matrix. As can be seen, the matrix is symmetric, and the values on the diagonal will always be equal to 1 (correlation of a variable with itself). The plot shows that all of the r values are less than 0.3, which suggests that there is weak correlation between the variables.

  The Sphere function can be used as a baseline/reference problem, since it is known that the true correlation matrix (in the limit of a large sample) will be the identity matrix.

4.4.2 Correlations between Variables

In some problems, there exists some correlation between variables. Out of these, some problems do not show any correlated patterns between them while in some cases, a correlation pattern has been formed between the variables.

- 10D Rosenbrock Function

  The correlation between variables for the 10D Rosenbrock function is visualized in Figure 4.2. Overall there is no strong correlation between any variables. The plot is similar to the 10D Sphere function; however there are some subtle differences, namely there is some correlation between x_2 and x_3, and a little more correlation between other variables.

- 10D Rotated Ellipsoid Function

  Figure 4.3 shows the correlation graph for the 10D Rotated Ellipsoid function, which indicates that there exists a correlation between the variables and in addition to this, the correlation between consecutive variables forms a pattern, where x_i is in correlation with x_{i+1}, where $i \leq n-1$. The structure follows from the definition of this fitness function. The result shows that sampling, selection and analysis has recovered this structure to a good approximation.

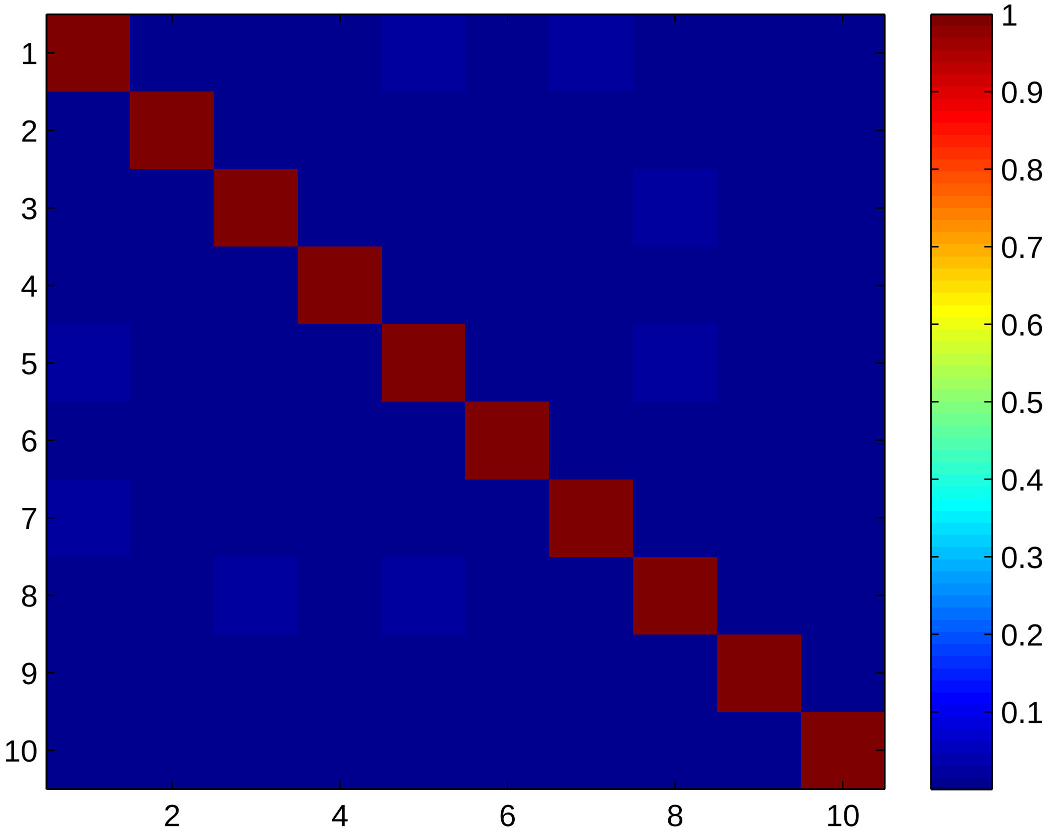


Figure 4.1: Correlation Graph for the 10D Sphere Function.

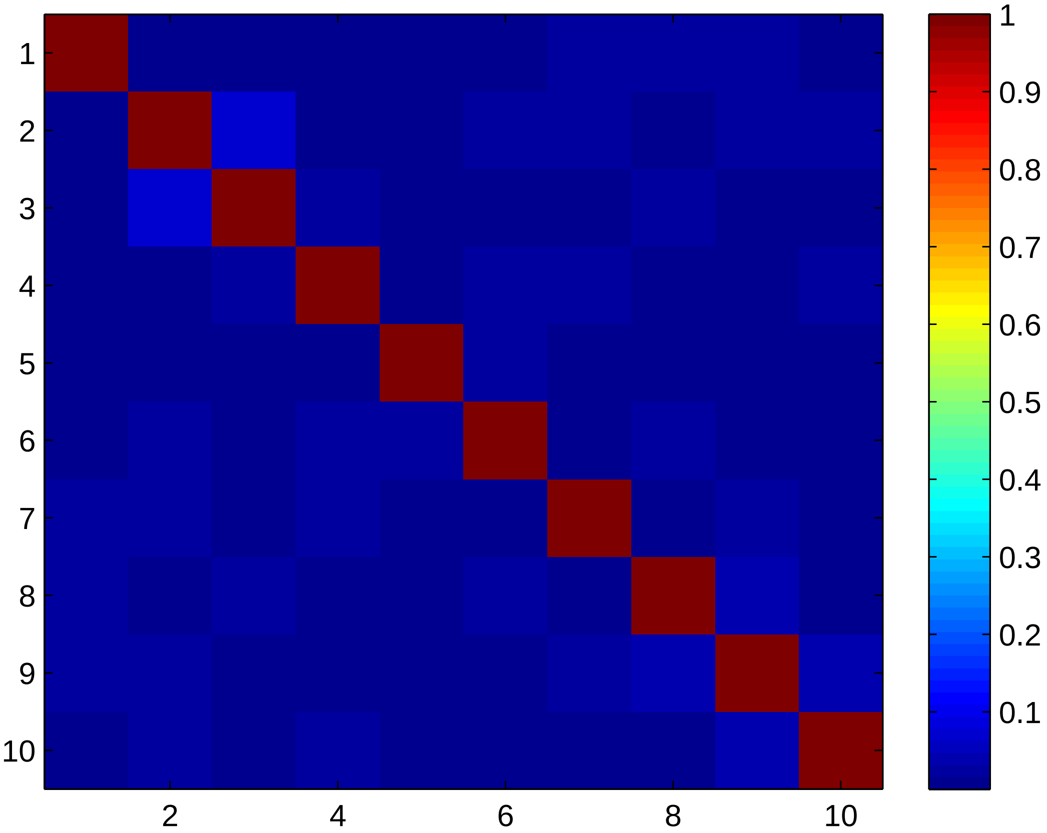


Figure 4.2: Correlation Graph for the 10D Rosenbrock Function.

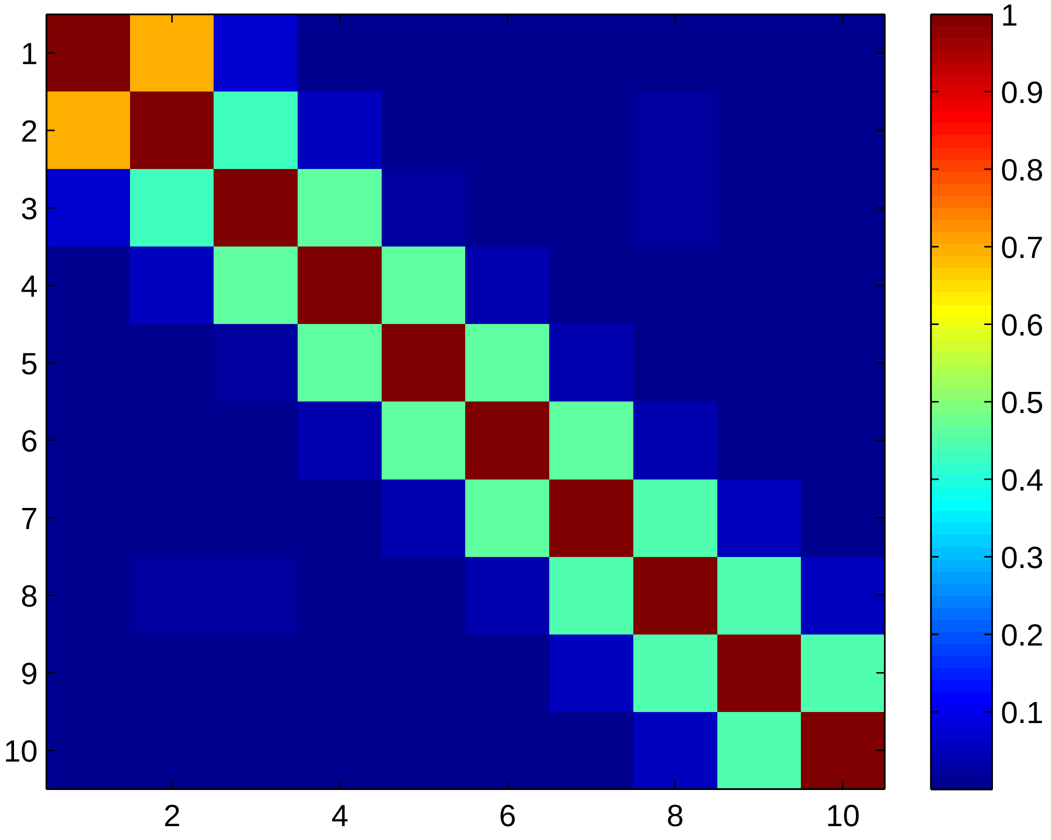


Figure 4.3: Correlation Graph for the 10D Rotated Ellipsoid Function.

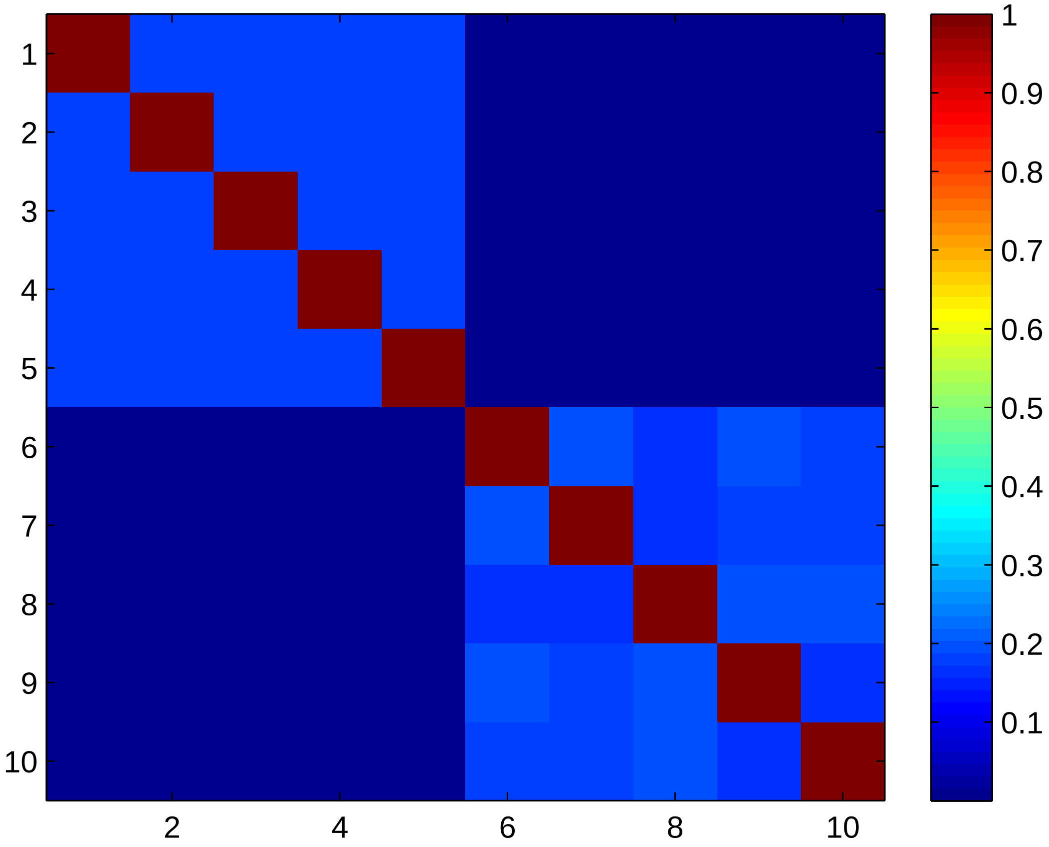
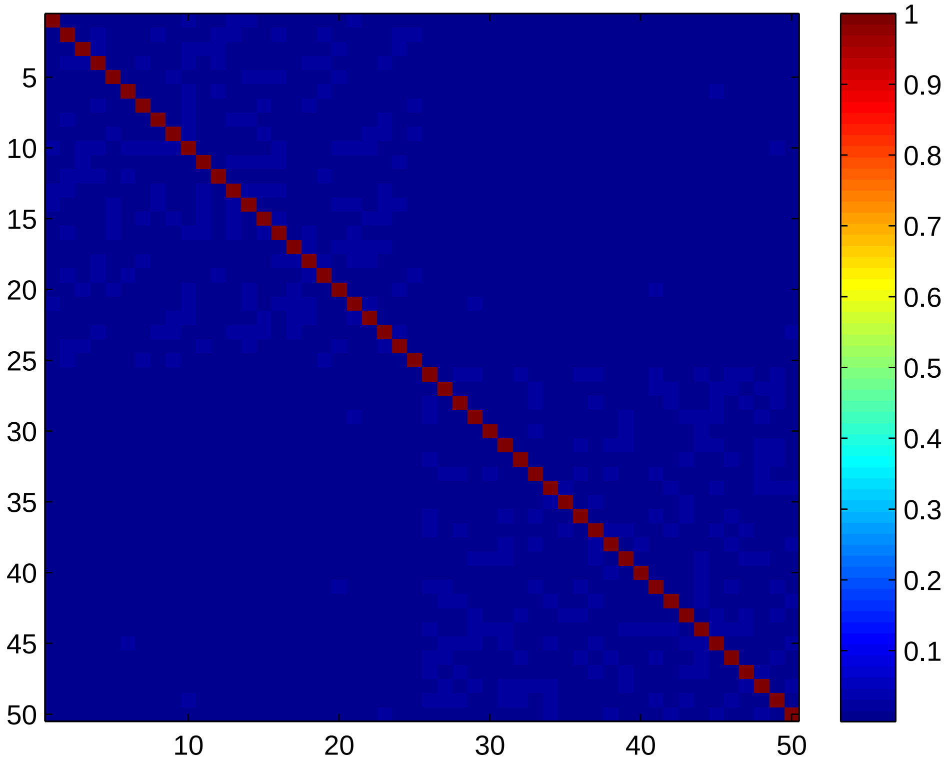
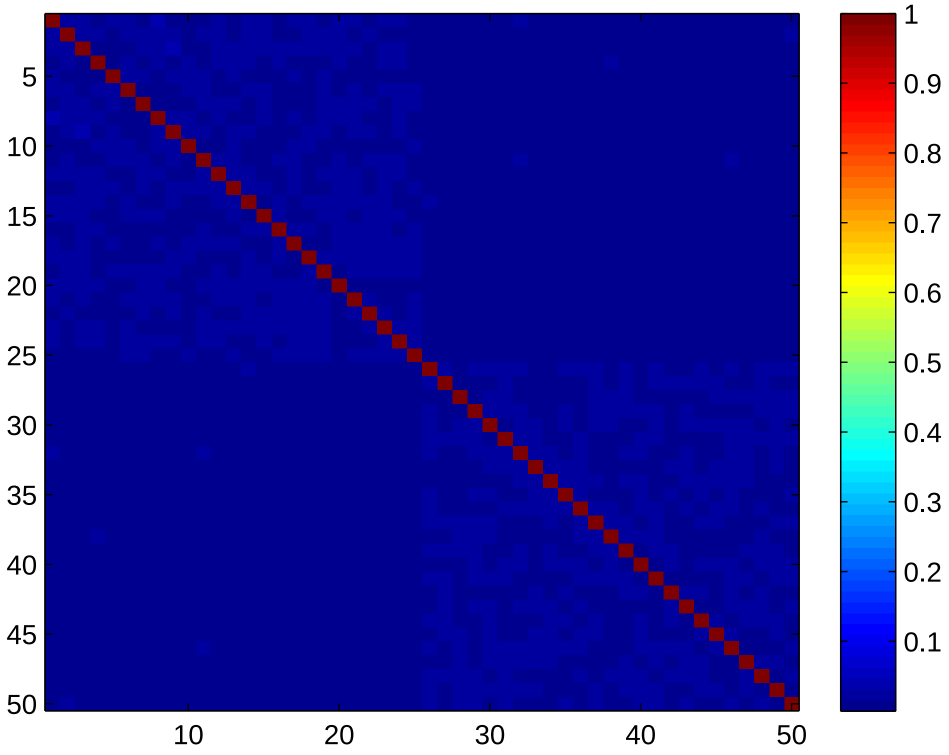


Figure 4.4: Correlation Graph for the 10D CiaS Problem.

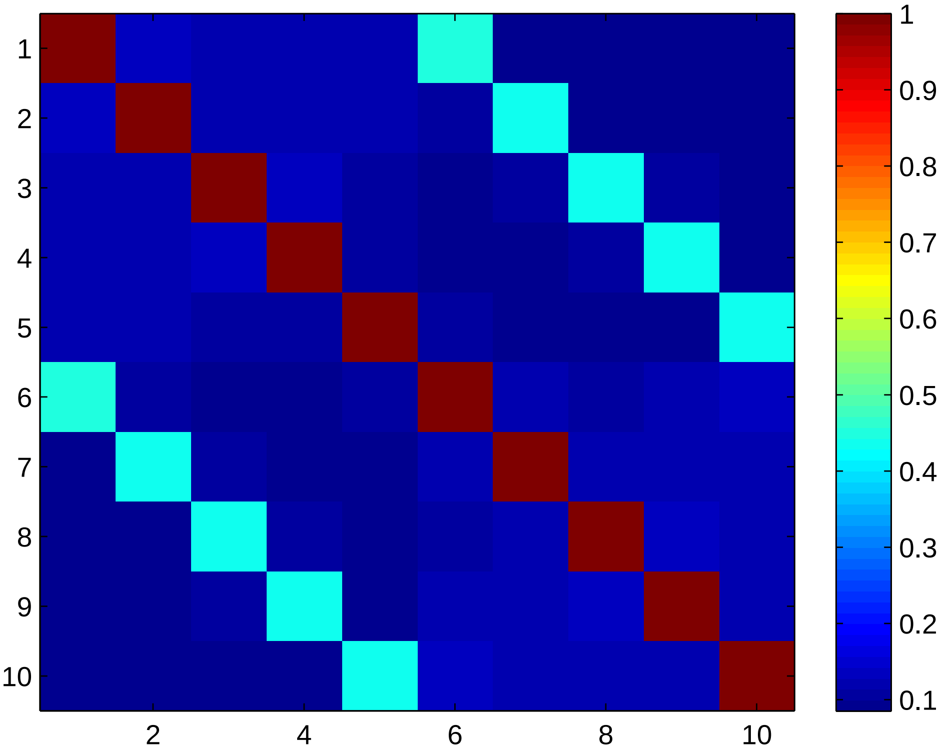


(a)

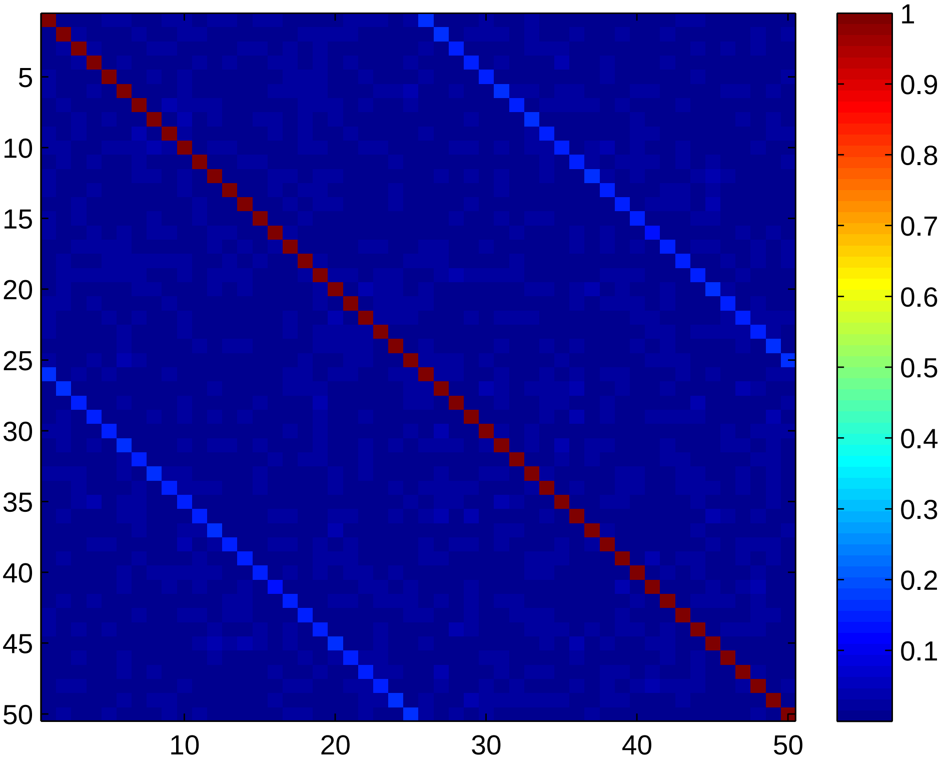


(b)

Figure 4.5: Correlation between variables in CiaS Problem ((4.5a) Correlation Graph for the 50D CiaS Problem showing less correlation between variables, (4.5b) Correlation Graph for the 50D CiaS Problem showing little bit of more correlation between variables (30000000 number of sample points with selection ratio 0.001)).



(a)



(b)

Figure 4.6: Correlation between variables in location-allocation Problem ((4.6a) Correlation Graph for the 10D location-allocation Problem, (4.6b) Correlation Graph for the 50D location-allocation Problem).

- 10D and 50D CiaS Problems

  Figures 4.4 and 4.5a show the correlation graphs for 10D and 50D CiaS problems respectively. Figure 4.4 illustrates that all the x coordinates of the circles in the square show correlation to each other, and all the y coordinate of the circles are correlated to each other, but there is negligible correlation to be found between (x, y) pairs of the variables.

  Figure 4.5a shows the correlation pattern of the 50D CiaS problem. The correlations between many pairs of x variables (top-left quadrant) and pairs of y variables (bottom-right quadrant) still appears but is weaker. This is likely to be due to the increased dimensionality of the problem relative to the sample size used. Figure 4.5b shows the correlation graph for the 50D CiaS problem with 30000000 number of sample points with selection ratio 0.001, which indicates that, if a large number of sample points are supplied and the selection ratio is much more greedy, then, some off diagonal correlation between variables can be found.

- 10D and 50D Location-Allocation Problems

  Figures 4.6a and 4.6b show the correlation graphs for 10D (5 facilities) and 50D (25 facilities) location-allocation problems. Figure 4.6a shows that there is an interesting pattern of correlation among the x and y variable pairs for each facility in the customer space. For example, let us take the 3rd facility, where x_3 and x_8 are the x and y coordinates of the 3rd facility respectively. From the figure, it shows that x_3 and x_8 are correlated to each other. This is true for each facility of the problem. The same pattern of correlation is also observed in the higher dimensional instance of the problem (Figure 4.6b).

4.5 Key Structural Variables in Continuous Optimization Problems

As described in Section 3.5, eigenanalysis is used here to analyze the key structural variables for each problem. The selected sample points from the problems are used to calculate the covariance matrix which in turn is used in eigenanalysis. The eigenvalues calculated from the covariance matrix are plotted in a scree plot, which gives a visual indication of the dominant eigenvalues.

If the largest eigenvalue, λ_1 is significantly larger than all other λs, then the first eigenvector describes a significant amount of the variance in the sample. In this case, the larger coefficients of the first eigenvector are the key structural variables to the function (use of eigenanalysis for determining key structural variables and the threshold value for choosing the key structural variables have been discussed in Section 3.5.1). In other words, the first eigenvector is pointing more towards the key structural variables. Hence there are specific key structural variables of the problem.

Depending upon the information about the key structural variables obtained from the eigenanalysis, the problems may be roughly classified into the following types:

- problems having specific key structural variables.

- problems without any specific key structural variables.

4.5.1 Problems having Specific Key Structural Variables

- 2D Elliptical Function

  The 2D Elliptical function is useful to illustrate the analysis, since by definition the contours of this objective function are stretched more (by a factor of 10) along the x_2 axis than along the x_1 axis. The eigenvalues of the selected sample points on the 2D Elliptical function are $\lambda_1 = 90.85$ and $\lambda_2 = 9.145$. The first eigenvalue (λ_1) captures most of the variance, hence λ_1 gives information about the key structural variables of the problem. The vector coefficients of the first eigenvector are $c_1 \approx 1$ and $c_2 \approx 0$. In this case the first eigenvector is closely aligned with the x_1 axis, showing that the data varies more with respect to x_1 than x_2. Therefore x_1 is classified as a key structural variable in the 2D Elliptical function.

- 10D Rosenbrock Function

  The analysis of the key structural variables in the 10D Rosenbrock function is shown in Figures 4.7a and 4.7b. The scree plot of the 10D Rosenbrock function is shown in Figure 4.7a, which shows that the first eigenvalue captures most of the variance, in the selected sample. Thereafter, the coefficients of the first eigenvector are plotted in Figure 4.7b, which illustrates that c_{10} has the largest value, showing that the first eigenvector is closely aligned to the x_{10} axis. Hence, x_{10} is identified as a key structural variable for the 10D Rosenbrock function.

4.5.2 No Specific Key Structural Variables

- 10D Sphere Function

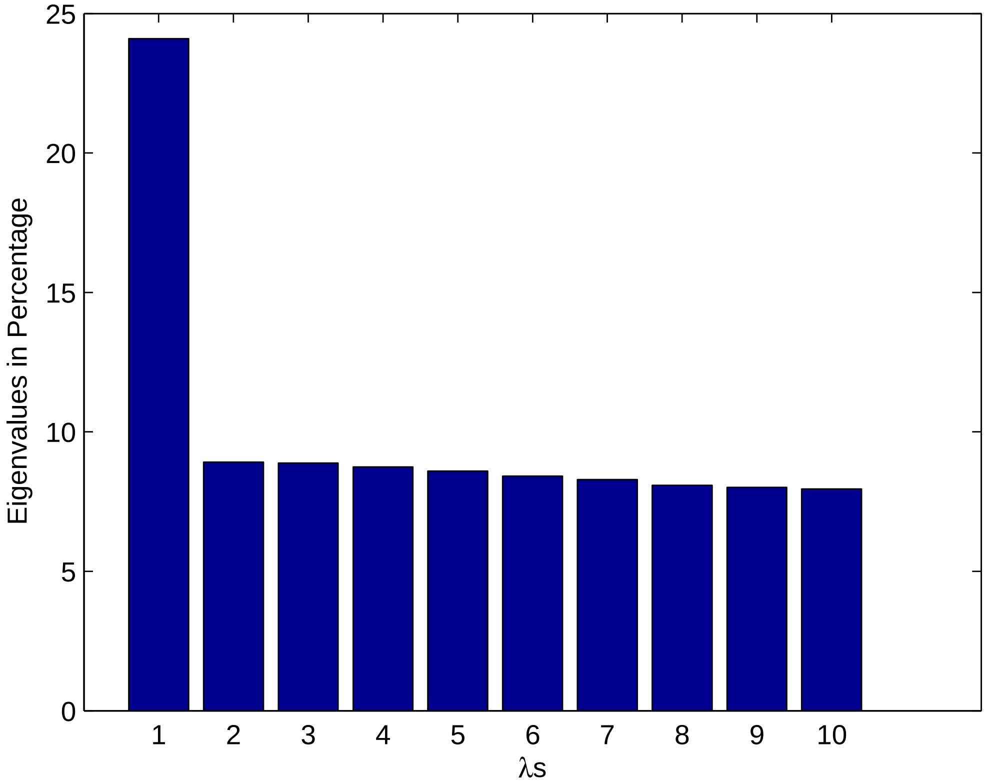
  The scree plot for the 10D Sphere function (Figure 4.8) shows that the λs have approximately equal values, which indicates that the selected sample points are spherical in shape. Here the first eigenvector does not capture significantly more variance in the selected data than any other direction in the search space. Therefore, no key structural variables are identified.

- 10D Rotated Ellipsoid Function

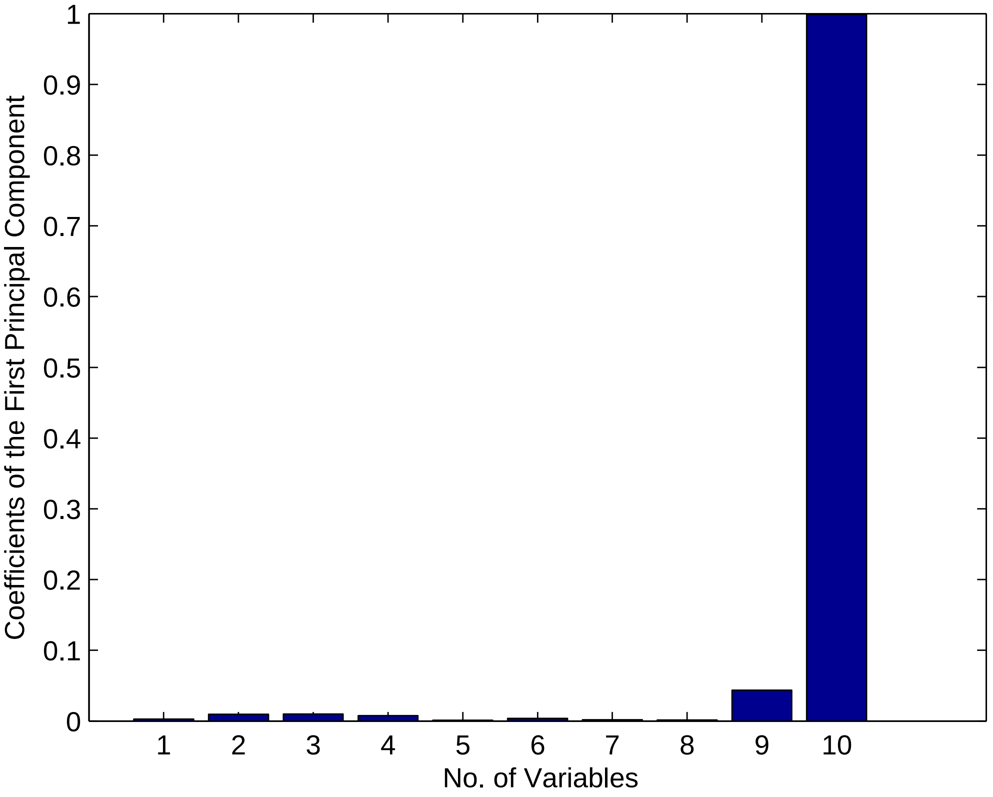
  The scree plot for the 10D Rotated Ellipsoid function (Figure 4.9) shows that the eigenvalues gradually decrease from around 19 percent of the variance to zero percent. Although some directions capture more variance than others, no single direction is dominant. Hence, there are no specific key structural variables in the selected data. However, the spread of the eigenvalues is distinctive, e.g. compared to the 10D Sphere function (see Figure 4.8), and is indicative of a particular elliptical structure.

- 10D and 50D CiaS Problem

  The scree plots for the 10D and 50D CiaS problems are plotted in Figures 4.10 and 4.11 respectively. The larger spread of eigenvalues for the 10D and 50D CiaS problem signifies that the problem is elliptical in shape. Scree plots for both the cases show that there are equal distances



(a)



(b)

Figure 4.7: Specific Key Structural Variables ((4.7a) Scree plot for the 10D Rosenbrock Function, (4.7b) Key structural Variables in the 10D Rosenbrock Function).

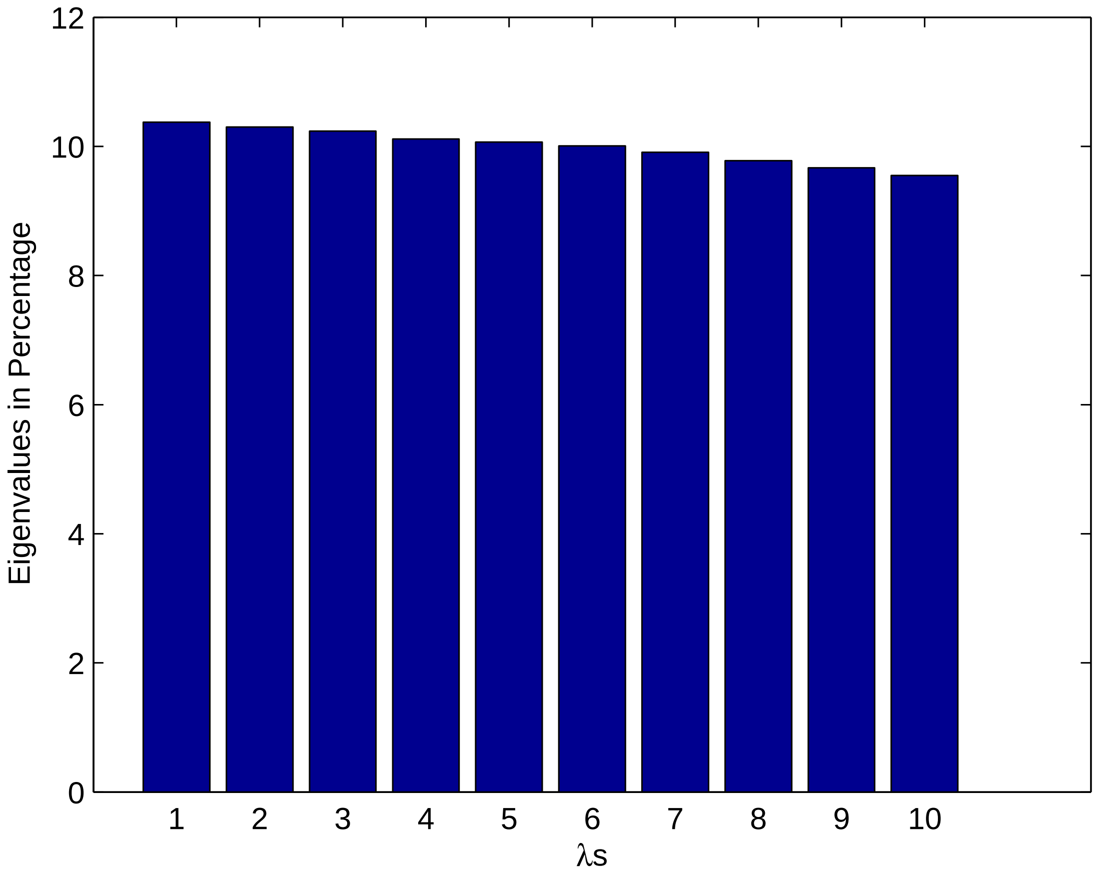


Figure 4.8: Scree plot for the 10D Sphere Function.

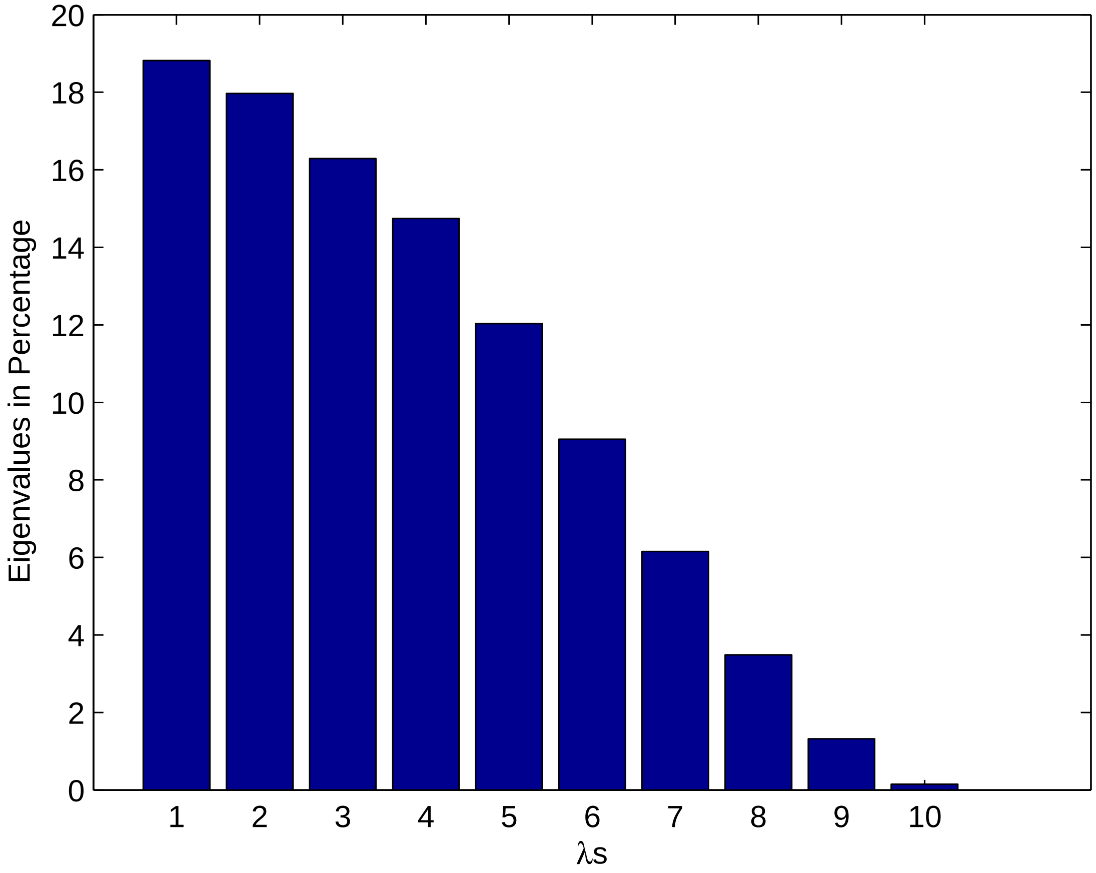


Figure 4.9: Scree plot for the 10D Rotated Ellipsoid Function.

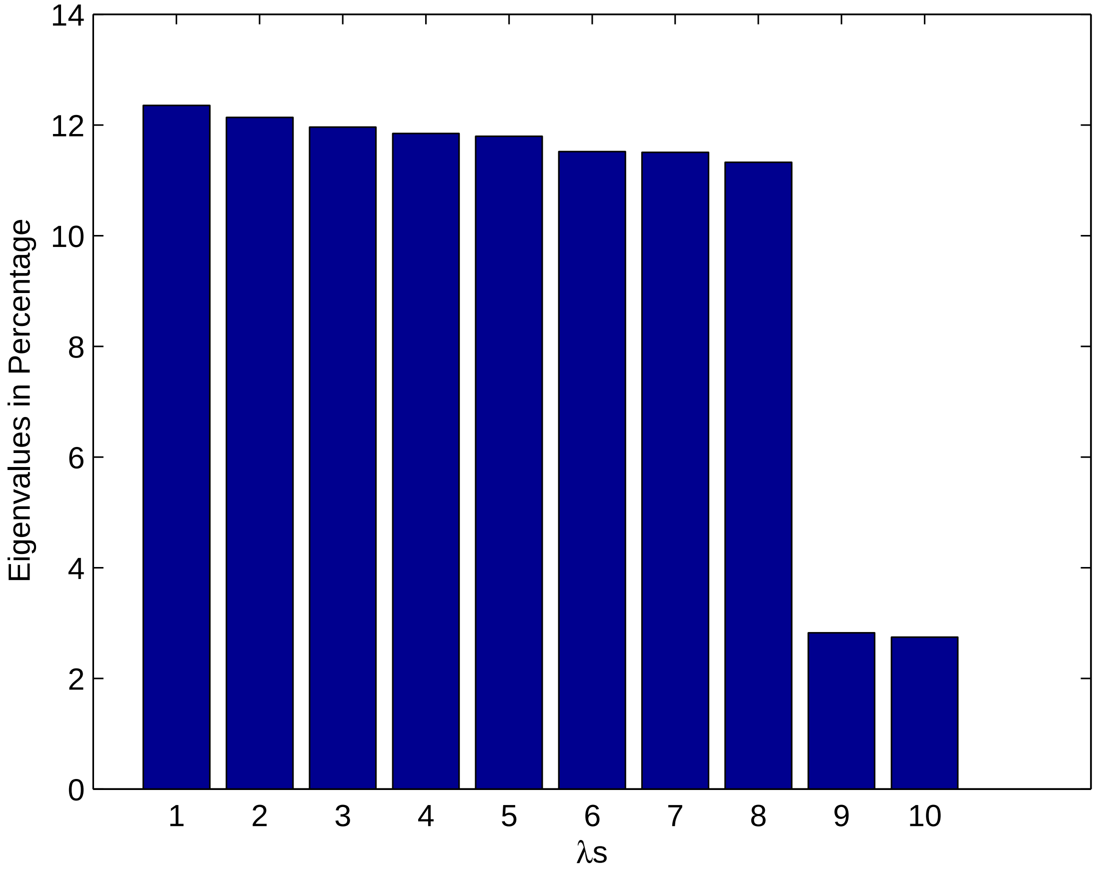


Figure 4.10: Scree plot for the 10D CiaS Problem.

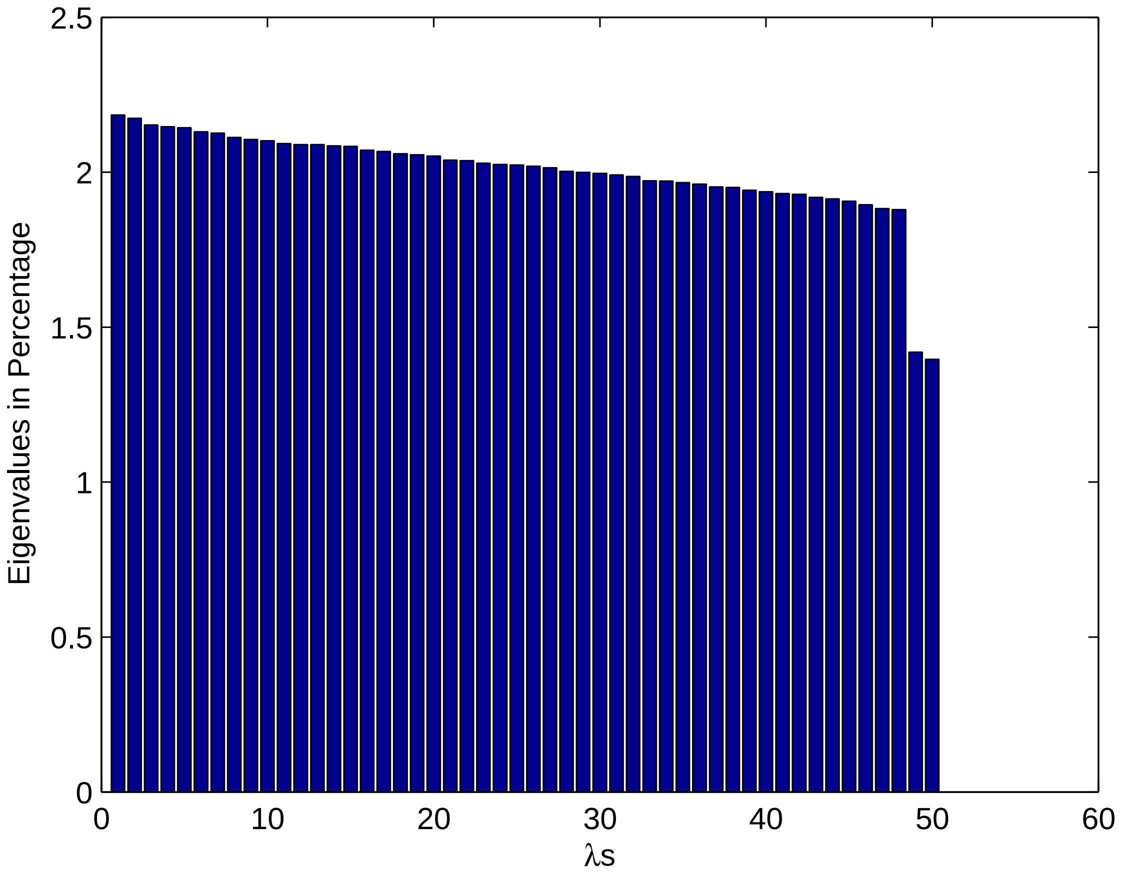


Figure 4.11: Scree plot for the 50D CiaS Problem.

between the eigenvalues from λ_1 to $\lambda_{(n-2)}$, where n is the dimensionality of the problem, which indicates that λ_1 is not able to provide enough information about the key structural variables to the problem.

- 10D and 50D Location-Allocation Problem

  Figures 4.12 and 4.13 are the scree plots for the 10D and 50D location-allocation problems respectively. Figure 4.12 and 4.13 show that there is a large spread of the eigenvalues, hence indicating that both the problems are elliptical in shape. Both the scree plots show that the first $((n/2)-1)$ eigenvalues are sharing equal amount of variances in data, where n is the dimensionality of the problem. Therefore it is very difficult to interpret the key structural variables from the first eigenvector.

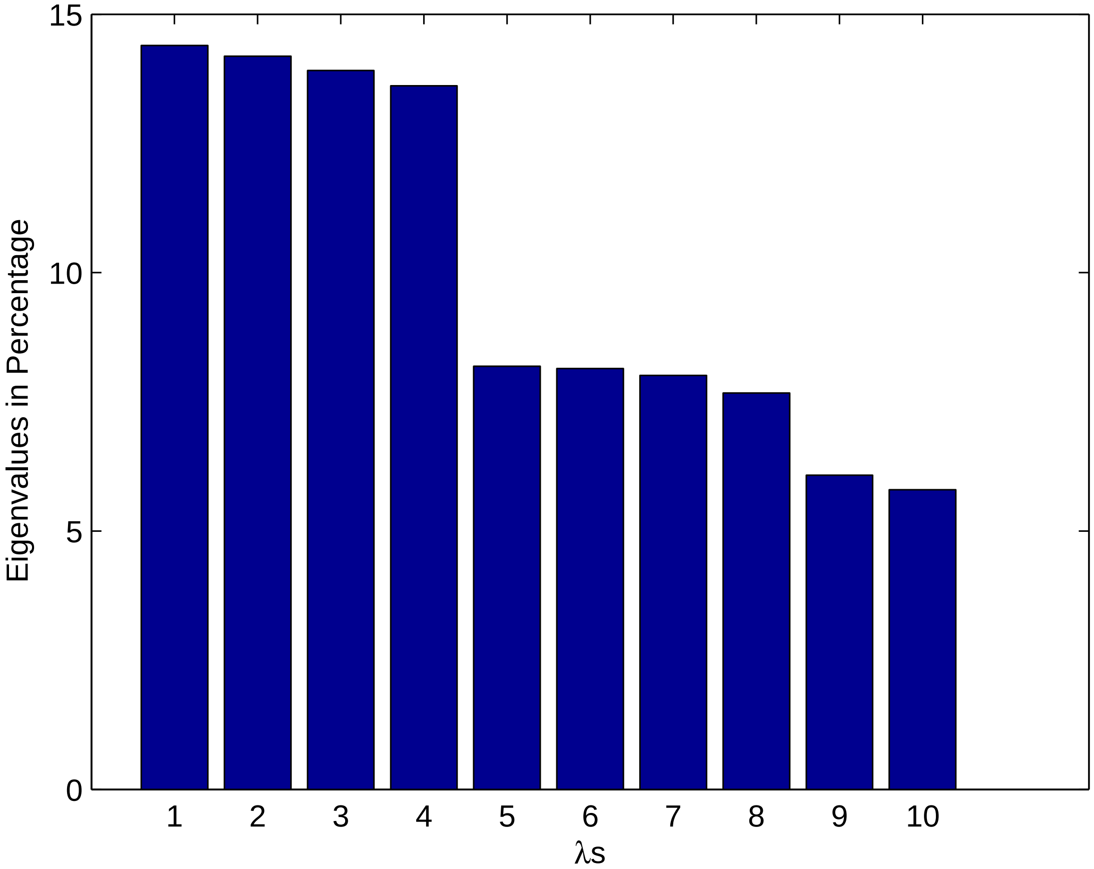


Figure 4.12: Scree plot for the 10D location-allocation Problem.

In summary, a varied distribution of eigenvalues have been observed, and in some cases, the first eigenvalues capture most of the variances.

4.6 Important Variables in Optimization Problems

The methodology for identifying "important" variables is described in Section 3.6. In this section, the technique is applied to the problem set using the selected sample points as were used for correlation

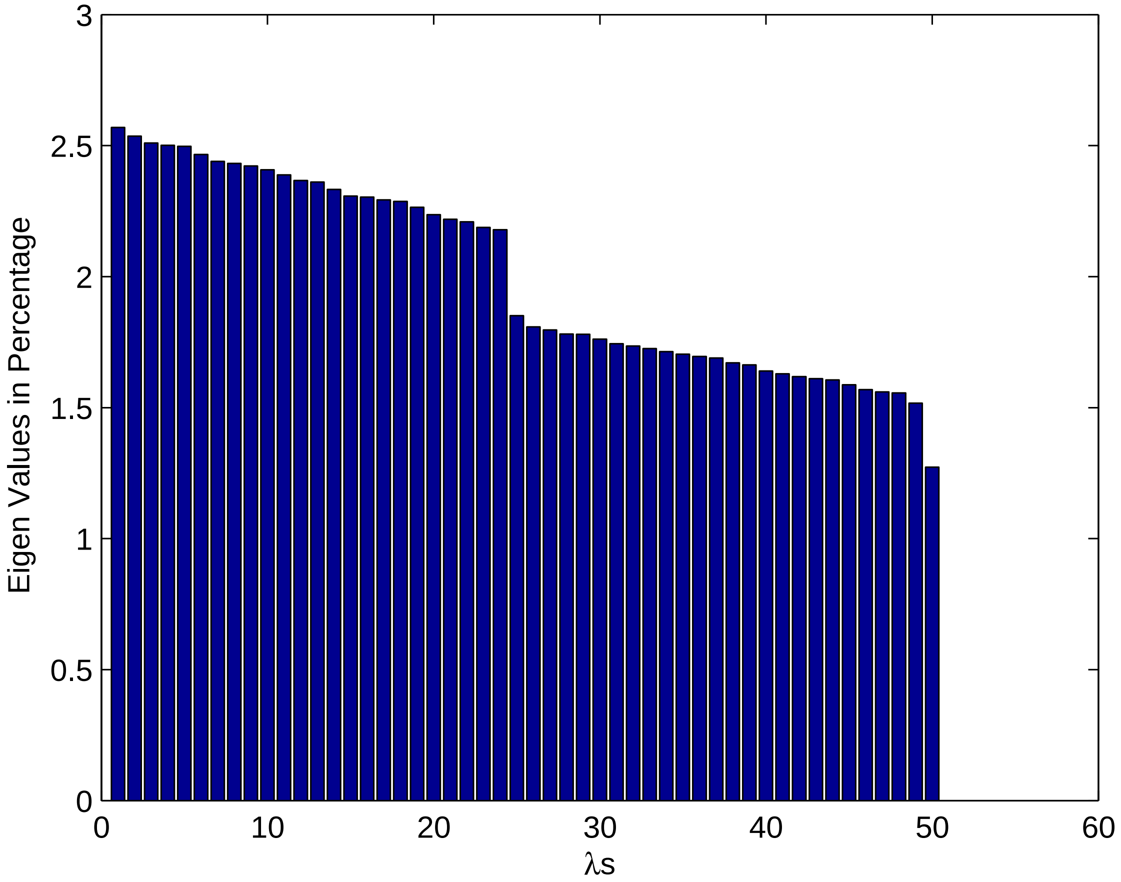


Figure 4.13: Scree plot for the 50D location-allocation Problem.

plots and key structural variables in Sections 4.4 and 4.5 above.

The estimated mean ($\overline{E}_i^*(\mathbf{x})$) and standard deviation ($std(E_i(\mathbf{x}))$ of the elementary effect values are plotted in the ($\overline{E}_i^*(\mathbf{x}) \sim std(E_i(\mathbf{x}))$ plane. Since the sample points generation are random in nature, each function is calculating the mean and standard deviation of the elementary effect for 10 different runs in order to know the exact "important variables" to the objective function.

Therefore each function consists of two figures, showing the important variables. The first figure shows the mean and standard deviation of the single run, which gives an idea about the distribution of variables in the ($\overline{E}_i^*(\mathbf{x}) \sim std(E_i(\mathbf{x}))$ plane. The second figure gives an idea about the proper order of important variables in a problem. The figures are representing the estimated values of $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x})$ for a single run and 10 runs respectively. The scale on the axes of the plots is a direct measure of the amount of variation in the objective function with respect to perturbations of each variable within the selected sample.

In this section, the classification of problems is based on the analysis of the order of $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ in the ($\overline{E}_i^*(\mathbf{x}) \sim std(E_i(\mathbf{x})))$ plane.

Based on the results, the test problems can be roughly categorized as follows:

- All/Some variables are important without any specific ordering among the variables.
- Variables are important with some specific ordering among the variables.

4.6.1 All/some Variables are Important without any order

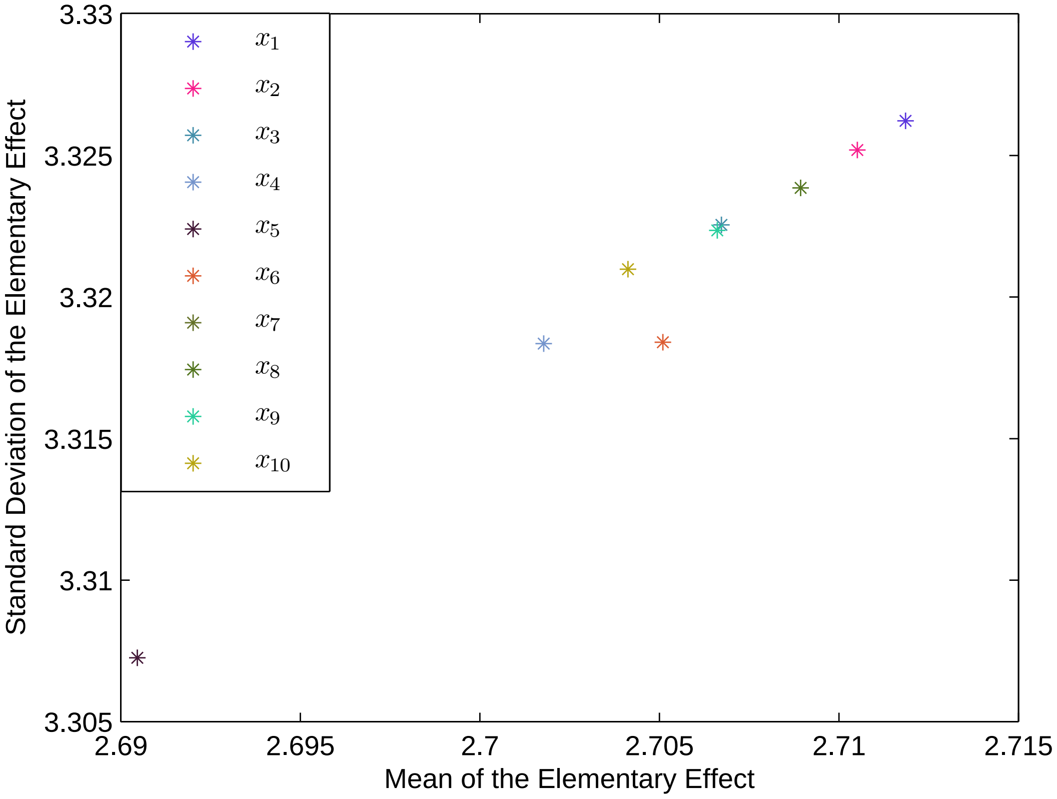
- 10D Sphere Function

  In the 10D Sphere function, the estimated $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ for all the variables are expected to be similar following from the definition of the problem. The results of the analysis are shown in Figure 4.14a. It shows that the $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ calculated for each variable are placed close to each other in a very small range in the ($\overline{E}_i(\mathbf{x}^*) \sim std(E_i(\mathbf{x})))$ plane, which indicates that the objective function has a similar sensitivity to all of the problem variables. This implies that none of the input factors that affect the output have a purely linear effect [133].

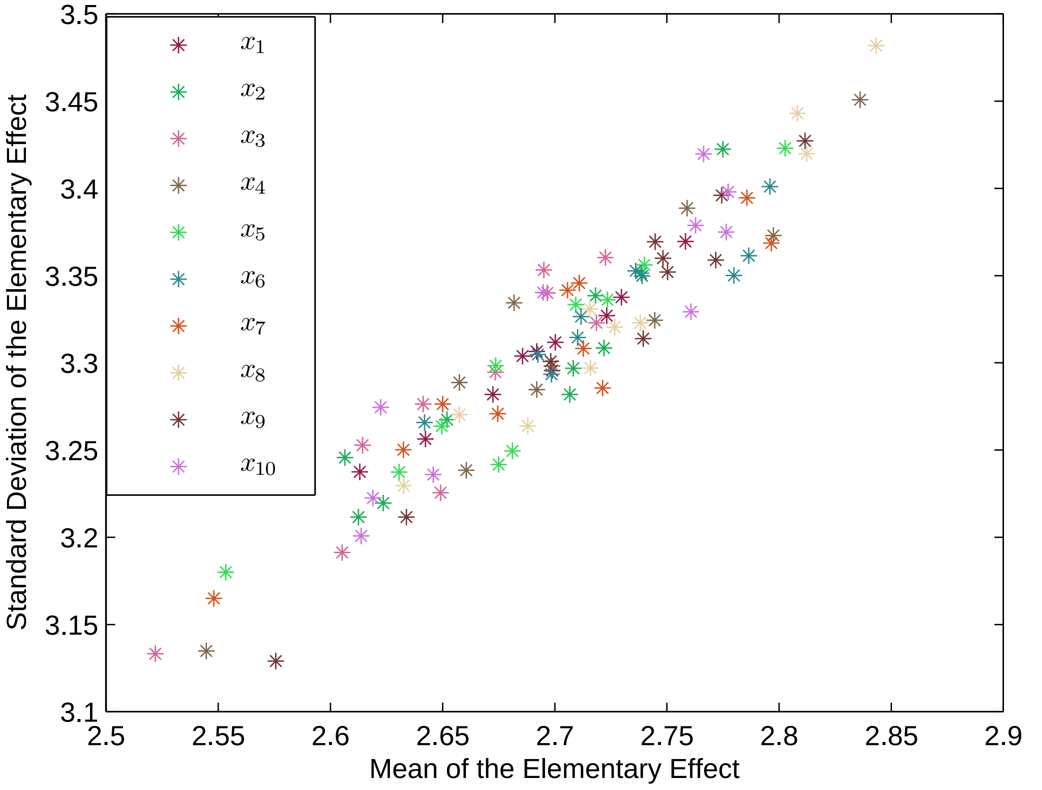
  In addition to this, to make a generalization, the values of $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ of the elementary effect values for 10 different runs are plotted in Figure 4.14b. It illustrates that, the variables are equally important to the problem for multiple runs as well, but no specific order has been found in the distribution of variables in the ($\overline{E}_i(\mathbf{x}^*) \sim std(E_i(\mathbf{x})))$ plane over the different runs.

- 10D and 50D CiaS Problem

  The estimated values of $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ calculated for the 10D CiaS problem for a single run are plotted in Figure 4.15a. In this case, $\overline{E}_i^*(\mathbf{x}) \ll std(E_i(\mathbf{x}))$. Hence, variables are not providing any overall importance to the objective function. But some of the variables are involved in interaction and/or curvature effects due to the high value of $std(E_i(\mathbf{x}))$ (only 1 or 2 variables have higher $std(E_i(\mathbf{x}))$ values in this case). Therefore, the 10D CiaS problem is sensitive to the variables which have larger values of $std(E_i(\mathbf{x}))$.

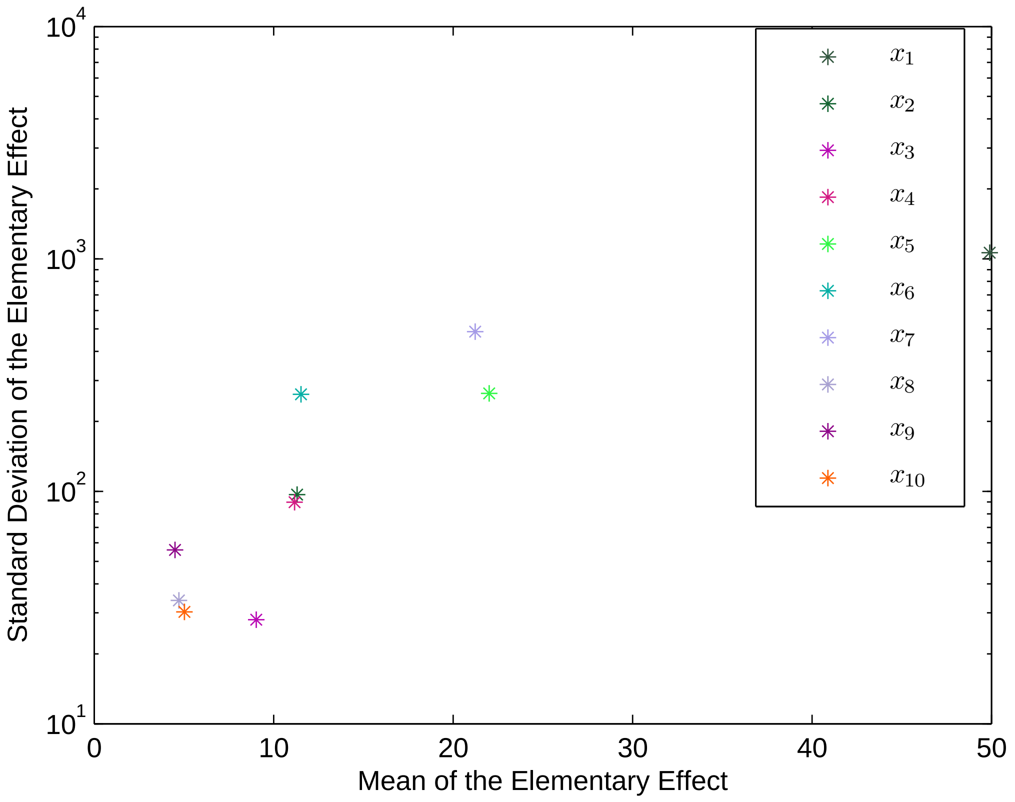


(a)

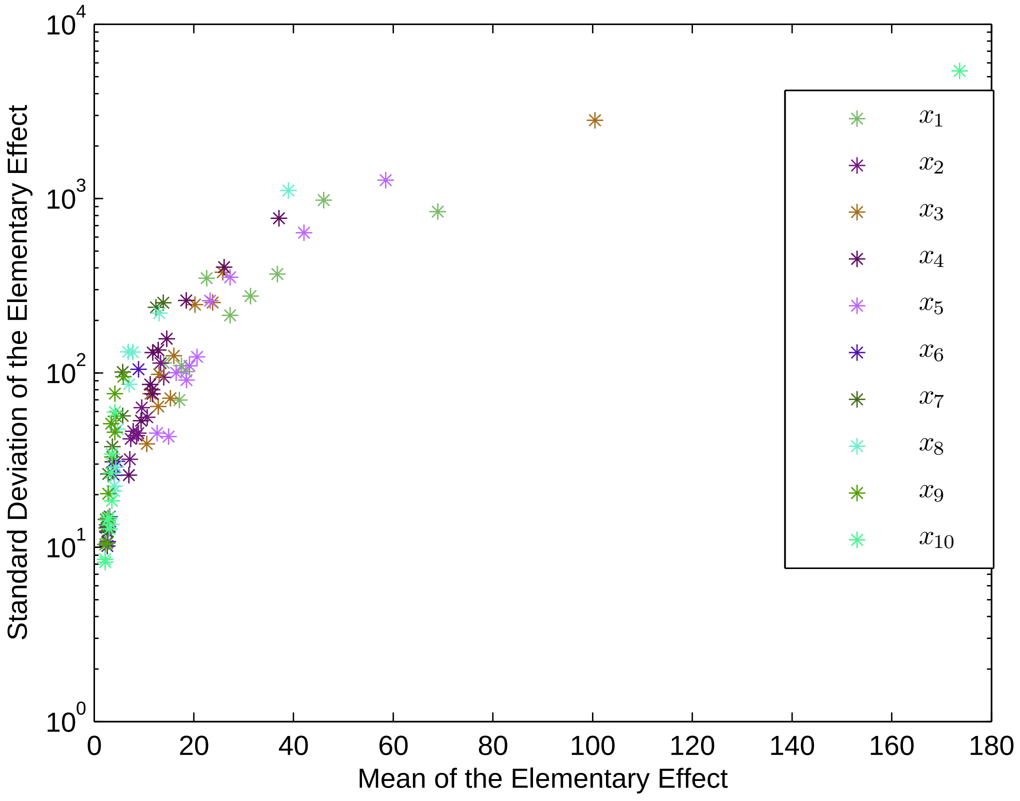


(b)

Figure 4.14: (4.14a) Important Variables in the 10D Sphere Function for a single run, (4.14b) Important Variables in the 10D Sphere Function for 10 different runs.

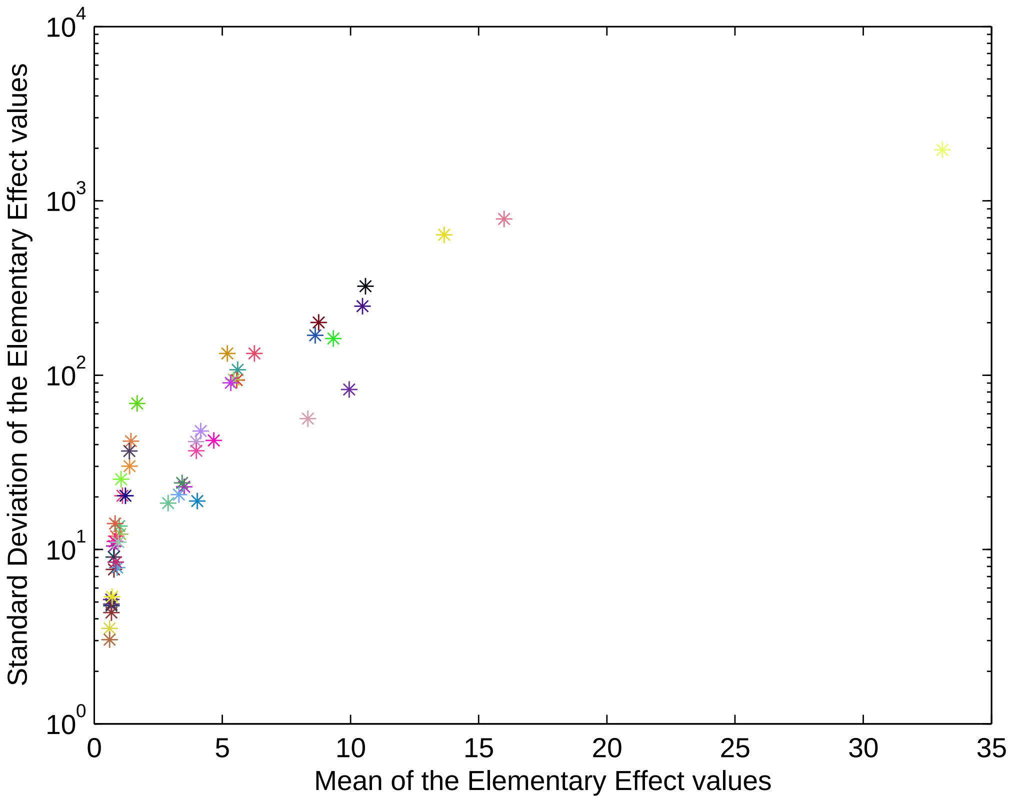


(a)

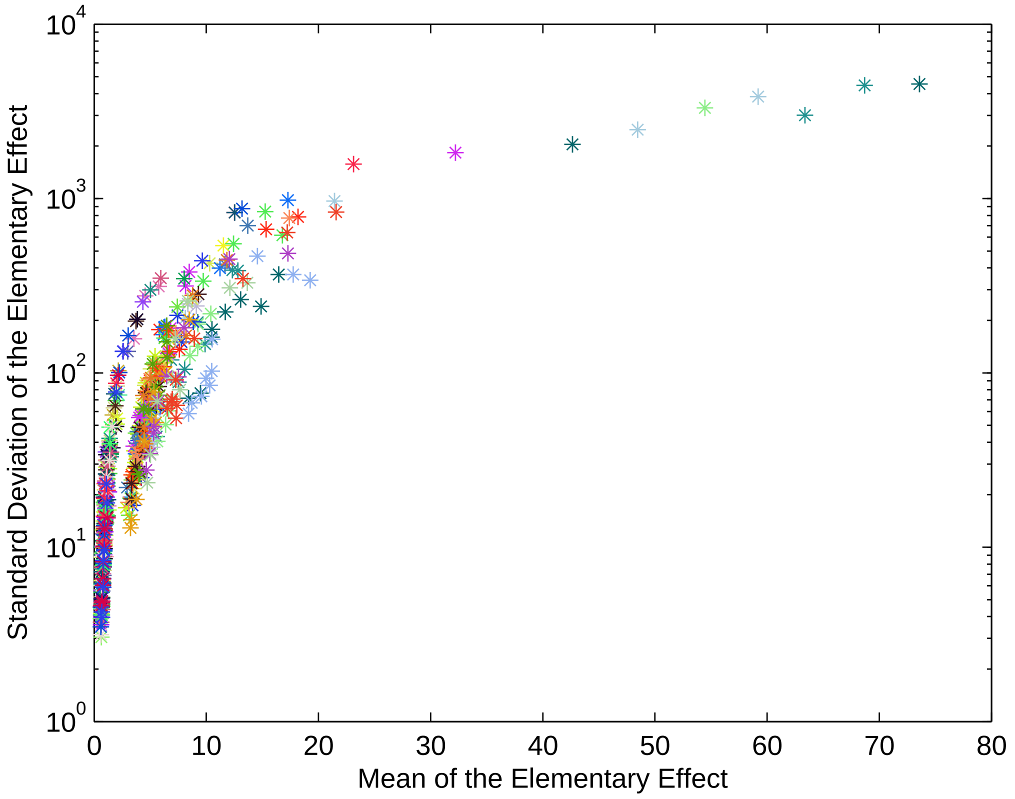


(b)

Figure 4.15: (4.15a) Important Variables in the 5 Circles in a CiaS Problem in a single run, (4.15b) Important Variables in the 5 Circles in a CiaS Problem in 10 different runs.



(a)



(b)

Figure 4.16: (4.16a) Important Variables in the 25 Circles in a CiaS Problem in a single run, (4.16b) Important Variables in the 25 Circles in a CiaS Problem in 10 different runs.

To examine the order of variables which are sensitive to the objective function, $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ of the elementary effect for 10 different runs are plotted in Figure 4.15b. Figure 4.15b shows the same distribution of variables as in the case of the 10D CiaS problem for a single run, in the $(\overline{E}_i^*(\mathbf{x}) \sim std(E_i(\mathbf{x})))$ plane. From the 10 different runs, it shows that there is no specific order of important variables found in the 10D CiaS problem.

Furthermore, the trend of distribution of variables in the 50D CiaS problem is the same as the 10D CiaS problem. The results of the analysis are shown in Figures 4.16a and 4.16b for single run and 10 runs respectively. It shows that $\overline{E}_i^*(\mathbf{x}) \lll std(E_i(\mathbf{x}))$ in the $(\overline{E}_i^*(\mathbf{x}) \sim std(E_i(\mathbf{x})))$ plane. Hence it shows that the variables are important which have higher $std(E_i(\mathbf{x}))$. They also do not show any order of the important variables in $(\overline{E}_i^*(\mathbf{x}) \sim std(E_i(\mathbf{x})))$ plane.

- 10D and 50D Location Allocation Problem

  Figure 4.17a shows that all the variables are lying at the $\overline{E}_i^*(\mathbf{x}) \approx 0$ zone.

  The analysis of important variables in the location-allocation problem is the same as the CiaS problem. In this case also, the $\overline{E}_i^*(\mathbf{x})$ of the variables are approximately equal to 0. The analysis only considers the $std(E_i(\mathbf{x}))$ of the elementary effect values to measure the important variables, which only involves the interaction or curvature of the objective function. The results for 10D and 50D location-allocation problems for single run and 10 runs are shown in Figures 4.17a, 4.17b, 4.18a and 4.18b.

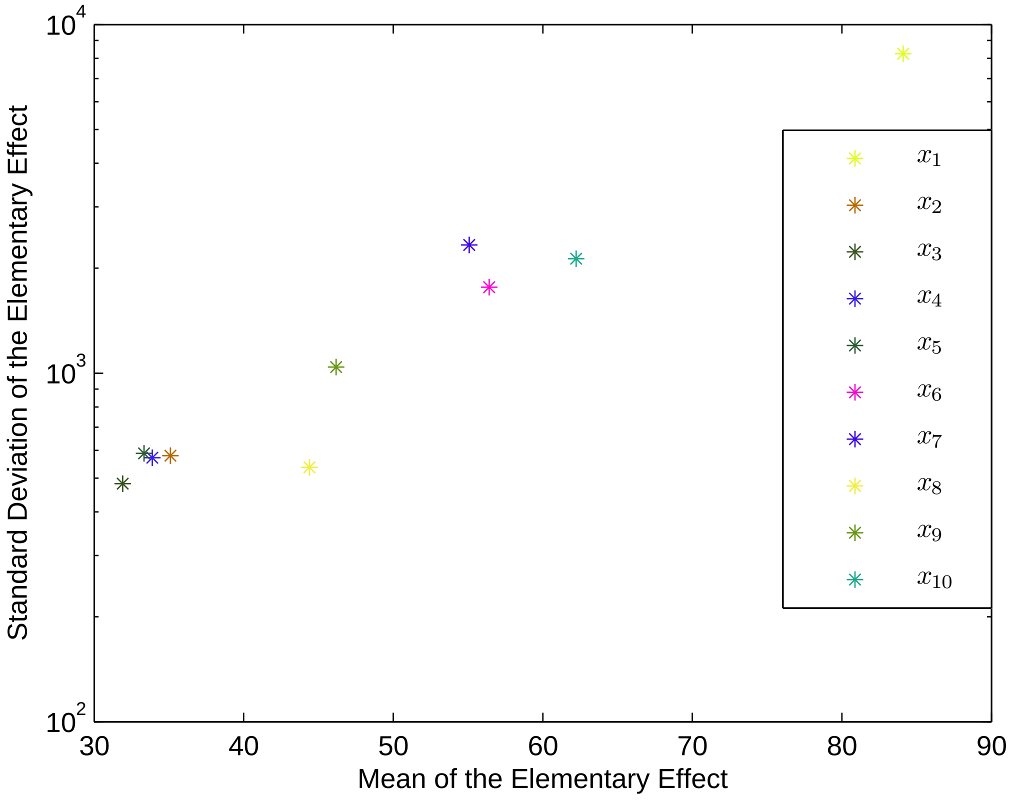
4.6.2 Variables are Important in some order

- 2D Elliptical Function

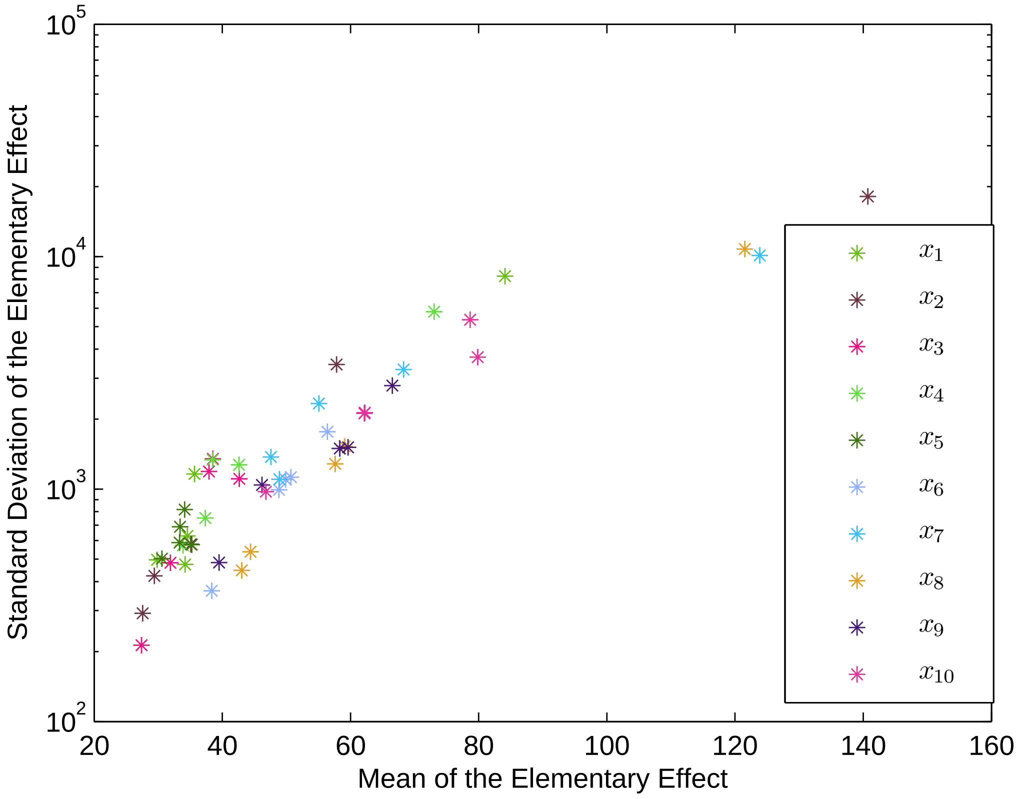
  Given the definition of this function, we know that the objective function is more sensitive to perturbations of x_2 and x_1. The results of the analysis are shown in Figures 4.19a and 4.19b. Figure 4.19a plots the values of $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ for a single run, which shows that one variable has higher value of mean and standard deviations than the other. To examine the order of importance, the mean and standard deviations of the elementary effects for 10 different runs are plotted in Figure 4.19b. Figure 4.19b shows that for each run x_2 has the largest $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ than x_1. Hence the order of importance is, x_2 is always more important than x_1 in the 2D Elliptical function.

- 10D Rotated Ellipsoid Function

  The analysis of the results for this problem is shown in Figures 4.20a and 4.20b. Figure 4.20a plots the mean and standard deviation of the elementary effects for a single run, which shows that each variable having a high value of the estimated mean also has a high value of estimated standard deviation of elementary effect, i.e. all the points lie close to the diagonal on the $(\overline{E}_i^*(\mathbf{x}) \sim std(E_i(\mathbf{x})))$ plane, which implies that variables have a non linear effect to the objective function. Thus a variable which is important in the objective function (high $\overline{E}_i^*(\mathbf{x})$) is also involved in curvature/interaction effects (high $std(E_i(\mathbf{x}))$) and vice versa.

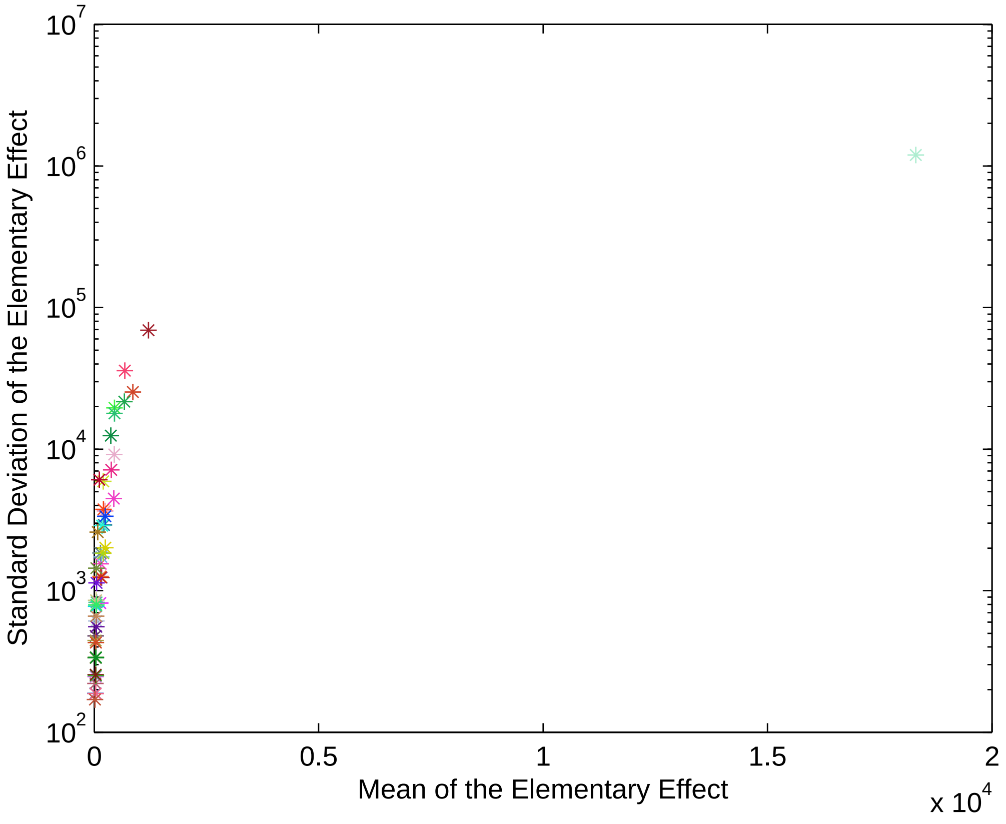


(a)

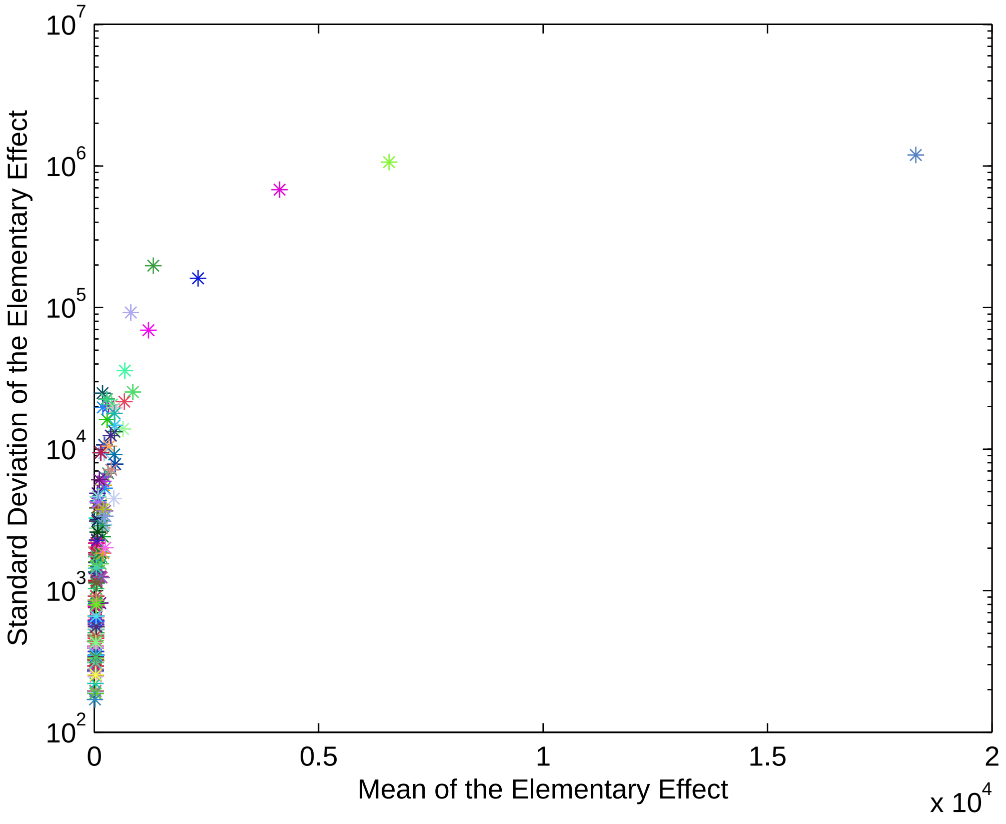


(b)

Figure 4.17: (4.17a) Important Variables in 5 Facilities in a location-allocation Problem in a single run, (4.17b) Important Variables in 5 Facilities in a location-allocation Problem in 10 different runs.

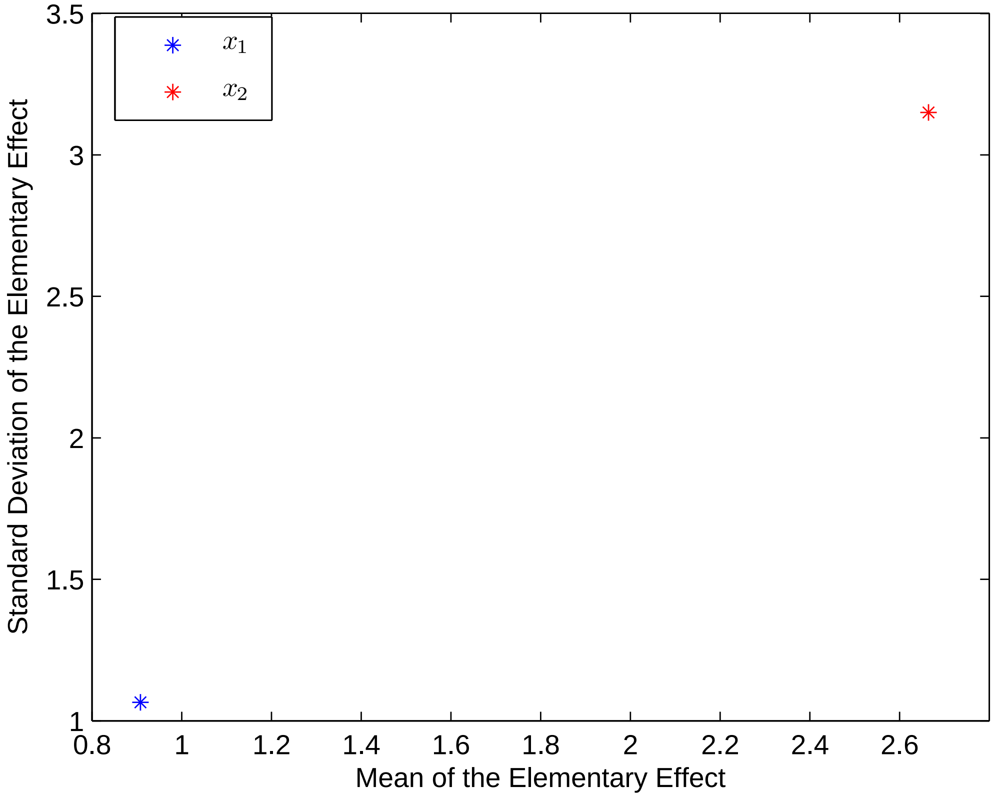


(a)

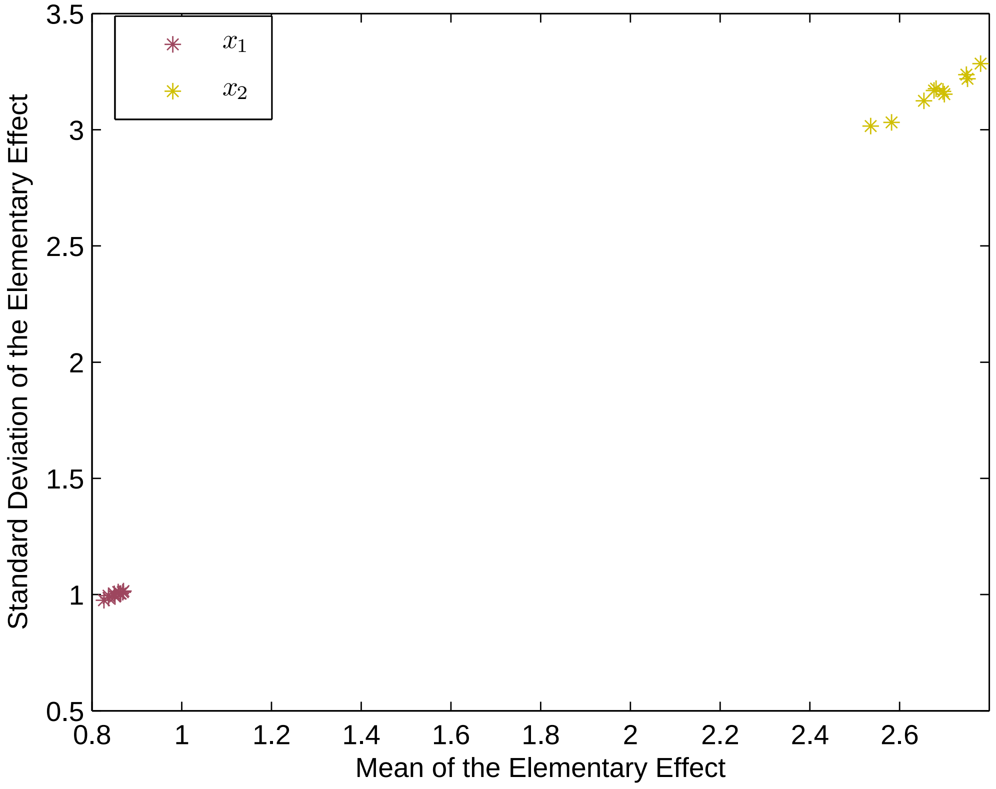


(b)

Figure 4.18: (4.18a) Important Variables in 25 Facilities in a location-allocation Problem in a single run, (4.18b) Important Variables in 25 Facilities in a location-allocation Problem in 10 different runs.

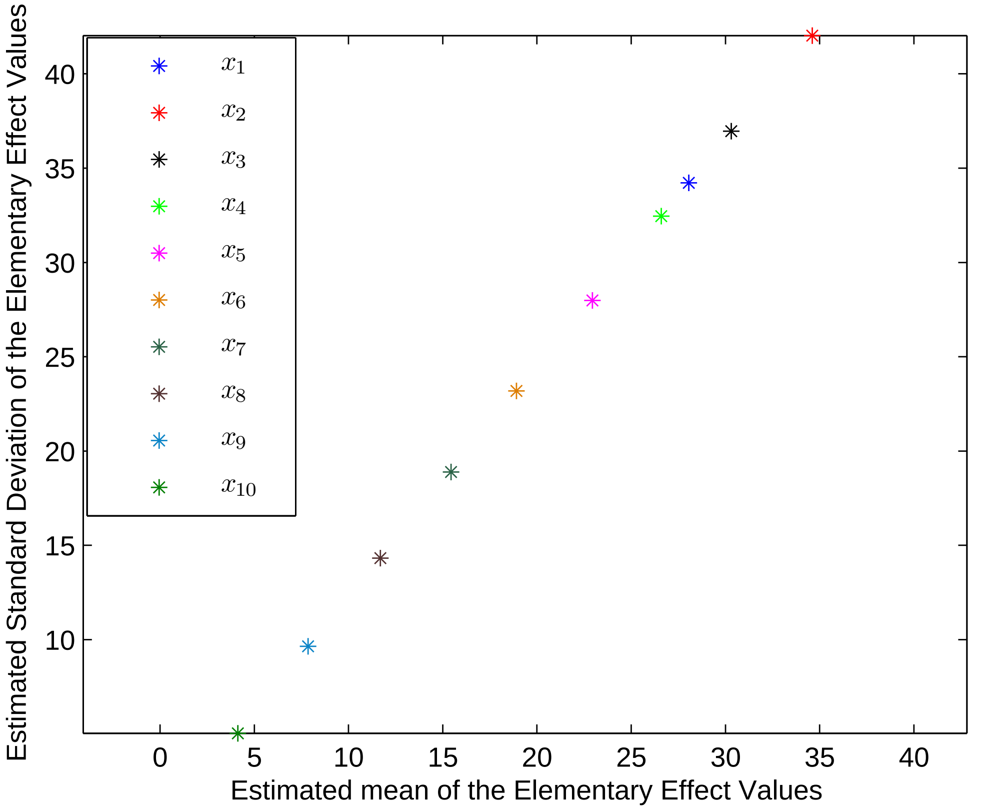


(a)

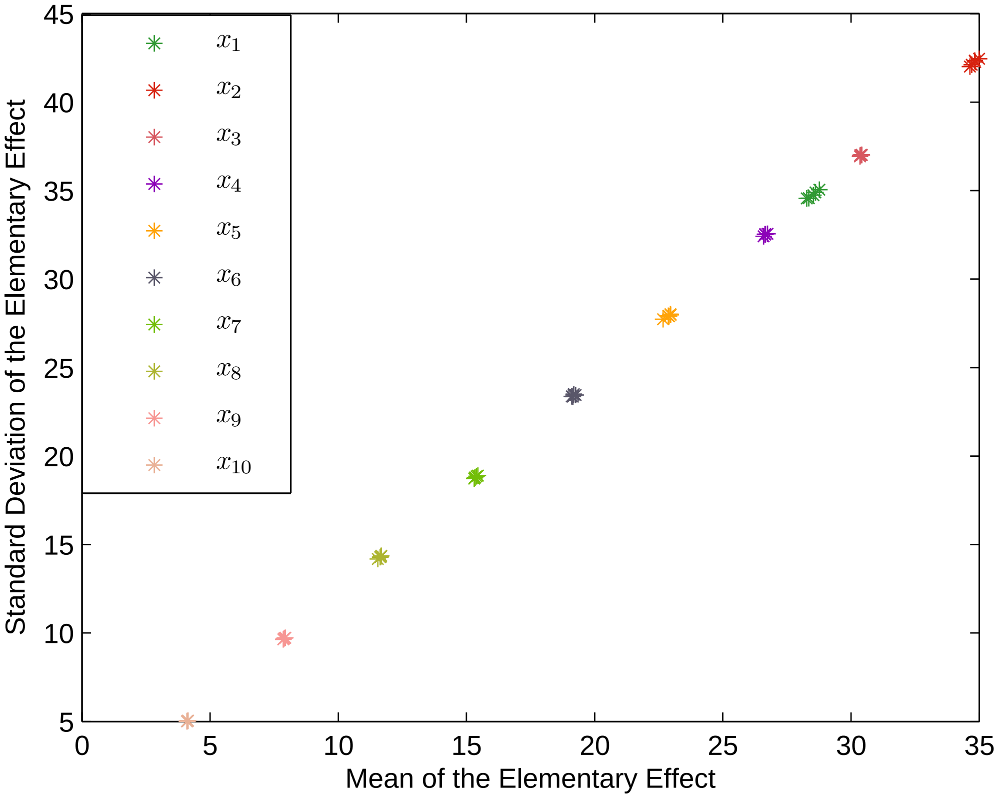


(b)

Figure 4.19: (4.19a) Important Variables in the 2D Elliptical Function in a single run, (4.19b) Important Variables in the 2D Elliptical Function for 10 different runs.

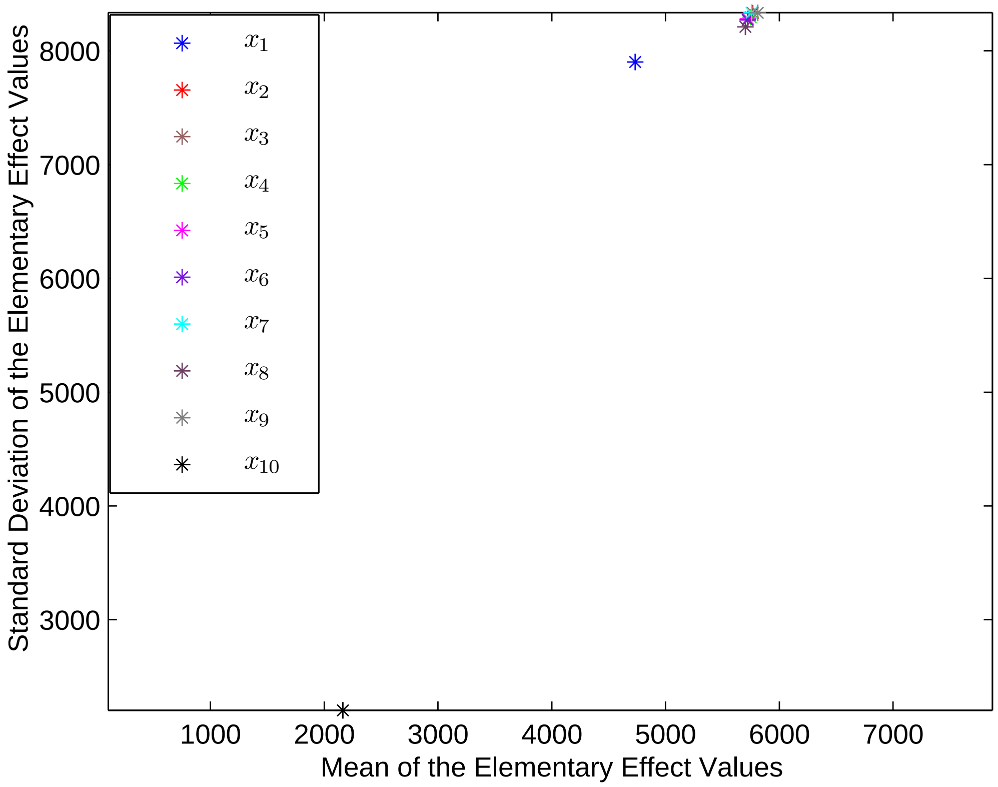


(a)

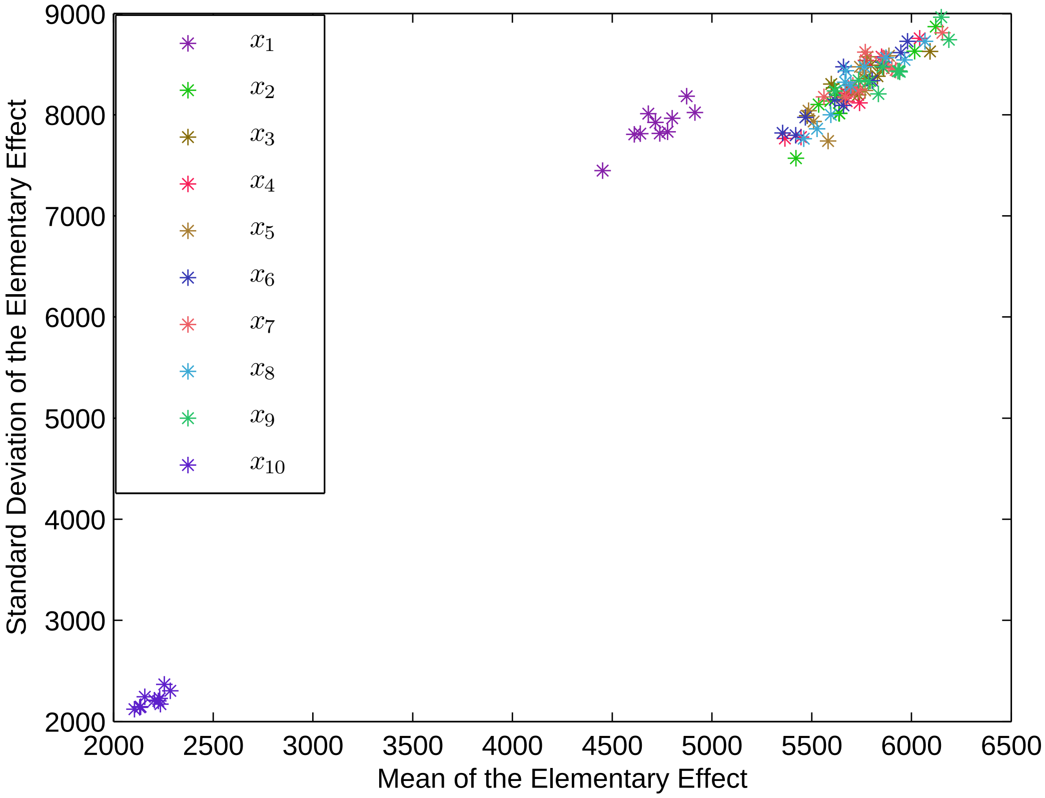


(b)

Figure 4.20: (4.20a) Important Variables in the 10D Rotated Ellipsoid Function for a single run, (4.20b) Important Variables in the Rotated Ellipsoid Function for 10 runs.



(a)



(b)

Figure 4.21: (4.21a) Important Variables in the 10D Rosenbrock Function for a single run, (4.21b) Important Variables in the 10D Rosenbrock Function for 10 different runs.

Figure 4.20b illustrates the order of important variables in the problem. Figure 4.20b shows that, in the 10D Rotated Ellipsoid function, the important variables are in order as follows: x_2 is more important than x_3, x_3 is more important than x_1, x_1 is more important than x_4, x_4 is more important than x_5 and so on.

- 10D Rosenbrock Function

  The $(\overline{E}_i^*(\mathbf{x}) \sim std(E_i(\mathbf{x})))$ graph in Figure 4.21a shows that a group of variables, having high $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$, are clustered away with larger values of mean and standard deviations from the rest of the variables. Hence these clustered variables are more important to the problem than the rest.

  To examine the order of importance between the variables, 10 different runs are considered. The $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ of the elementary effect values for 10 different runs are plotted in Figure 4.21b. Figure 4.21b shows that $x_2 \ldots x_8$ are placed in a cluster having high values of $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$. Although x_1 is placed near to the cluster variables, the spread and central tendency is less. x_{10} is the least important variable to the problem.

  Hence the order of important variables is, $x_2, \ldots,$ x_9 which are the most important, whereas x_{10} is the least important and x_1 the second least important variable in the problem.

  Explanation: From the Table 2.1, the definition of Rosenbrock function is defined as $f(\mathbf{x}) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$, can further be deduced to $f(x) = [100(x_2 - x_1^2)^2 + (x_1 - 1)^2] + [100(x_3 - x_2^2)^2 + (x_2 - 1)^2] + \ldots + [100(x_n - x_{n-1}^2)^2 + (x_{n-1} - 1)^2]$, which shows that the x_1 variable is comes in the first term only i.e. 100 times x_1^4 and one time x_1^2, where as the terms from x_2 to x_{n-1} comes more than one term in the definition, where as x_n comes in the last term with only 100 times x_n^2. Since Morris method is based on the sensibility of the objective function, it can be deduced that x_n is the least important variable where as x_1 is less important than $x_2, \ldots, x_{n-1}$.

4.7 Categorization of Problems Based on the Analysis

The analysis of variables for the different continuous optimization problems in Sections 4.4, 4.5 and 4.6 are summarized in Table 4.1. The summary shows that, although the Sphere and Elliptical functions have negligible correlation between variables, they have different key structural and important variables. Hence these functions are in different categories. In addition to this, the Rotated Ellipsoid function has a higher correlation coefficient but no specific key structural variables as well as some specific important variables. In some respect, it is different from the Ellipse function because, the Ellipse has no correlation, and specific key structural variables as well as specific important variables. The Rosenbrock function has a higher correlation coefficient than the Elliptical function, with some specific order of important variables.

The characteristics of variables in the CiaS and location-allocation problems are similar in all 3 properties that have been analyzed here, but the CiaS and location-allocation problems follow different types of correlation patterns between variables.

Table 4.1: Summary of the Analysis.

Properties		Functions							
		A	B	C	D	E	F	G	H
Correlation	Weak Correlation	×	×						
	Correlation			×	×	×	×	×	×
Key structural	Non Specific	×		×		×	×	×	×
	Specific		×		×				
Important	Non Specific	×				×	×	×	×
	Specific		×	×	×				

A 10D Sphere Function, B 2D Elliptical Function, C 10D Rotated Ellipsoid Function
D 10D Rosenbrock Function, E 10D CiaS, F 50D CiaS
G 10D Location-Allocation, H 50D Location-Allocation

It is possible to have 8 different categories, but using the examples discussed in the thesis, this analysis can be divided into 5 different possible categories:

- Category 1: Weak Correlation, No specific key structural variables and no specific order of important variables, (e.g. Sphere function)
- Category 2: Weak Correlation, Specific key structural variables and specific order of important variables, (e.g. Ellipse function)
- Category 3: Correlation, No specific key structural variables and specific order of important variables, (e.g. Rotated Ellipsoid function)
- Category 4: Correlation, Specific key structural variables and specific order of important variables, (e.g. Rosenbrock function)
- Category 5: Correlation, No specific key structural variables and No specific order of important variables, (e.g. CiaS and Location-Allocation problems)

Here the main motivation for characterizing these problems of forming categories is to determine whether the EDA model, while implemented on a problem captures the same properties identified in the categories or not. The hypothesis is that, the problem is easy, if the EDA model exiting while implementing a problem achieves the same category as the problem does. These concepts can also be used to develop new EDA algorithms for getting good optimal results.

4.8 Summary

This chapter has implemented the proposed framework for analyzing the properties of the variables of optimization problems. Specifically, correlation coefficients, eigenanalysis and variable screening were applied to samples of candidate solutions from artificial and real-world representative problems.

These techniques allow us to visualize and analyze the nature of the variables in the optimization problems. It has been shown that although some problems are similar in some properties, clear differences are also revealed by the analysis. Hence for simplification, 5 different categories, which are based on the combination of these 3 variable properties have been defined. These categories are further used for comparison purposes of the nature of variables for problems and for EDAs in Chapter 5.

Chapter 5

Variable Analysis and Comparison: EDAs

Fundamentally, the idea of using a probabilistic model to capture dependency information between problem variables and/or to search parts of the space containing high-quality solutions underlies all of the variants of Gaussian EDAs. However, relatively few studies have examined the behavior of the model experimentally. For example, the performance of an algorithm does not indicate whether or not dependency modeling was valuable (and at what points during the search), or which problem variables are important to the model during evolution. In order to understand whether or not an EDA is capturing and exploiting information regarding the influence of each variable and variable dependencies in problems, more detailed experimental analysis must be carried out.

In this chapter, the methods developed in Chapter 3 to analyze correlation, key structural variables and important variables (i.e. correlation, eigenanalysis and sensitivity analysis) are used to analyze EDA models. These same techniques were used in Chapter 4 to analyze a set of optimization test problems and these same problems are used in this Chapter for evaluating the EDAs. This allows a direct comparison to be made between the results in Chapter 4 and those in this Chapter. Since the EDA models are Gaussian, the analysis techniques are well-suited to analyzing the selected population data generated by the algorithms during execution. The results are for $\text{EMNA}_{\text{global}}$, but the methods are applicable to any continuous EDA.

5.1 General Consideration

The general aim of this work is to develop techniques to better understand and evaluate the relationships between problem variables and the optimization algorithm that is applied to a given problem. In practice, the performance of all metaheuristic optimization algorithms can be analyzed by looking at the solutions and the corresponding fitness values evaluated during the search. In EDAs, the model parameters provide a useful summary of the search. Figure 5.1 shows a diagram illustrating the main components of data used in the analysis for Gaussian EDAs (mean and covariance matrix model parameters). Previously, the mean vector of the model and the solution values (e.g. the best-so-far values) have been typically used to describe the performance of the EDA model [59]. The mean vector tracks the center of the search distribution, recording progress of the model over the search space. The covariance matrix (Σ) represents the variability of the search around the mean and determines the direction of future search progress. Σ also controls the scale of search in the space and captures pairwise dependencies. However, there has been little effort to explicitly analyze the dynamics of the covariance matrix during EDA experiments.

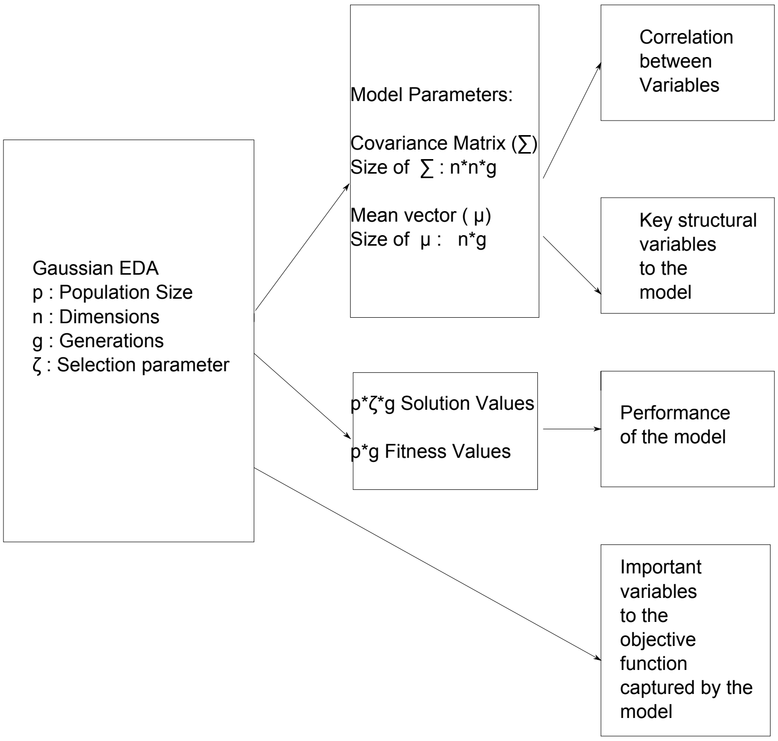


Figure 5.1: Illustrating the analysis in EDAs using current methodology.

The main advantage of using this comparison is that the best model has been chosen for a particular problem. This analysis also helps to choose the model selection parameter values of different EDA algorithms, where it chooses a sparse matrix for the modeling purposes. In this way it fulfills the research gap which has been discussed in Section 2.8.

5.2 Correlation between Variables in EDAs

EMNA$_{\text{global}}$ uses the multivariate Normal distribution as a probability distribution. For a multivariate Normal distribution, the Pearson's correlation coefficient is an appropriate choice for measuring the dependence between the variables. The calculation of correlated variables and the visualization of correlation in a heat map have already been discussed in Section 3.4.

For the analysis of the EDA, a correlation graph can be generated for the selected population at each generation. For display purposes, correlation graphs are presented as snapshots at using specific generations selected at important stages in the evolution.

If the same general trends of correlation have been followed by every generation, one correlation graph is sufficient. Multiple correlation graphs are used to show different trends of correlation throughout the generation.

5.3 Eigenanalysis for Examining Key Structural Variables in EMNA$_{\text{global}}$

The example of a multivariate Normal distribution is EMNA$_{\text{global}}$. The maximum likelihood estimates of EMNA$_{\text{global}}$ are the mean and the covariance. Eigenanalysis has been conducted with the covariance matrix of the EMNA$_{\text{global}}$.

The equal density contours of a non-singular multivariate normal distribution are ellipsoids centered at the mean. The directions of the principal axes of these ellipsoids are given by the eigenvectors of the covariance matrix $\sum$ [74]. The squared relative lengths of the principal axes are given by the corresponding eigenvalues. If the mean is zero, the eigenvectors provide a rotation of the coordinate system such that the new axes (i.e. with the eigenvectors as basis vectors) lie along the principal axes of the ellipsoids. Hence, the range of search points generated by the EDA is in proportion to the eigenvalues. This provides a complete description of the Gaussian model covariance at any generation of the EDA.

In general, the eigenvectors and eigenvalues of Σ for an EDA describe the shape of the (ellipsoid) search distribution. If the distribution is approximately spherical, the eigenvalues will all be approximately equal, with the eigenvectors indicating no specific direction to the distribution. In contrast, if the distribution has high eccentricity in some directions, this will be reflected in one or more proportionally large eigenvalues. The corresponding eigenvectors indicate the directions of greatest variance in the model. In the next generation, the population is sampled from this model and therefore will be more widely distributed in these directions. It is a time series.

In a Gaussian EDA, the population at any given generation is sampled from a Gaussian distribution. Therefore, the population and to some extent the selected population should be well-described by a Gaussian distribution. This makes the data well suited to analysis using PCA (Section 3.5).

5.4 Identifying Important Variables in EMNA$_{\text{global}}$ using the Morris Method

The Morris method for the identification of important variables is discussed in Section 3.6 and used in Chapter 4 to analyze solution samples from problems. An EDA samples and evaluates candidate solutions from a different distribution at each generation of the algorithm. Therefore, it may be the case that a different set of variables (having high estimated values of the mean ($\overline{E}_i^*(\mathbf{x})$) and standard deviation ($std(E_i(\mathbf{x}))$) of the elementary effect) are identified at each generation. Sensitivity analysis can still be carried out using the Morris method, however in this case the results data is larger and more complex.

The concepts of dominance and Pareto optimality from multi-objective optimization (see, e.g. Chapter 9 of [46]) have been utilized to determine the important variables in EDAs, since it is difficult to analyze the mean and standard deviation of all generations in a single figure. The mean and standard deviation of elementary effects as two different (aka decision-making) criteria are considered. One solution is said to dominate the other if its score is at least as high for all objectives, and is strictly better for at least one. Let $\mathbf{d}^a$ and $\mathbf{d}^b$ have n objectives as a n-dimensional vector $\mathbf{d}$. Using the $\succeq$ symbol to indicate domination, $A \succeq B$ is defined as:

$$\mathbf{d}^a \succeq \mathbf{d}^b \Leftrightarrow \forall i \ \in \{1,...,n\} \, a_i \geq b_i,$$

$$\text{and } \exists i \ \in \{1,...,n\}, a_i > b_i.$$

All non-dominated solutions possess the attribute that their quality cannot be increased with respect to any of the objective functions without detrimentally affecting one of the others. In the presence of constraints, such solutions usually lie on the edge of the feasible regions of the search space. The set of all non-dominated solutions is called the Pareto set or the Pareto front. The remaining variables (which are not included in the Pareto front) are selected based on the minimum (Euclidean) distance to the Pareto front.

5.5 Correlation between Variables, Key Structural and Important Variables in an EDA based on Artificial Test Functions

EMNA$_{\text{global}}$ is implemented on the artificial test problems (2D Elliptical function, 10D Sphere function, 10D Rotated Ellipsoid function and 10D Rosenbrock function), that were considered in Chapter 4. EMNA$_{\text{global}}$ is run over 200 generations, with a population size of 2000 and 0.5 as its selection ratio for these test functions.

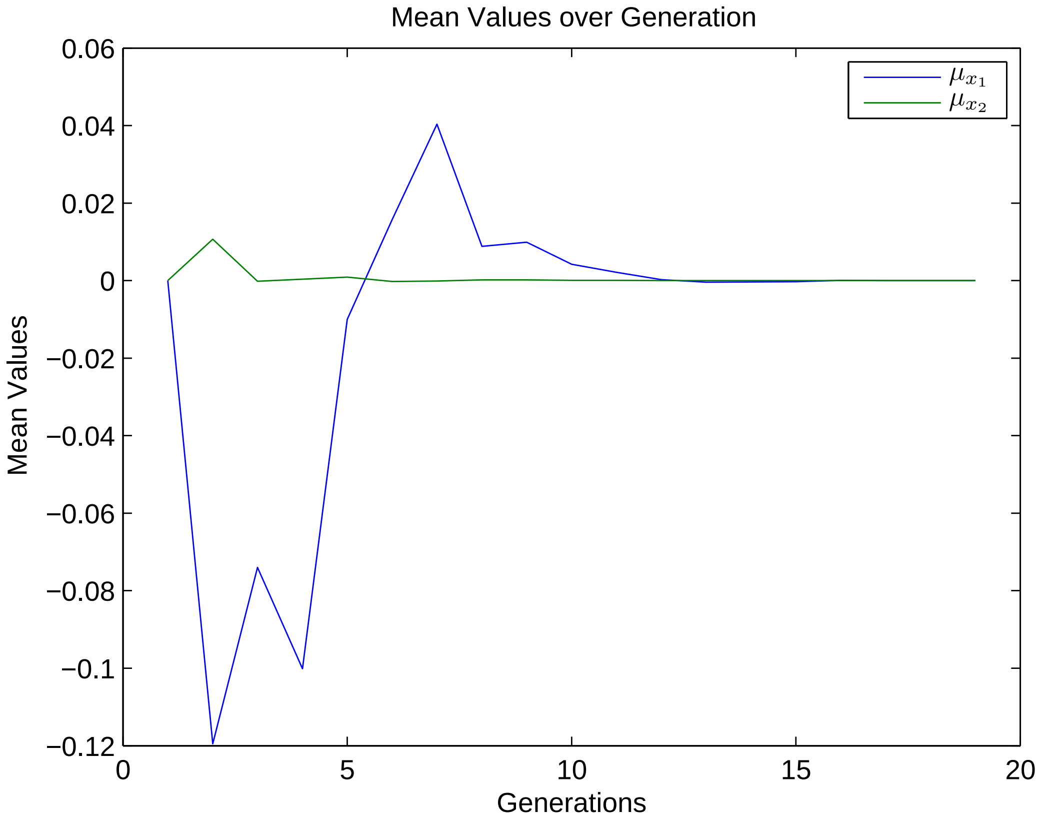
EMNA$_{\text{global}}$ model converges when the absolute difference between the maximum mean of the two consecutive generations is less than or equal to 1e-06 or terminate when it exceeds the maximum number of generations (i.e. 200 generations in this analysis). Note that the focus here is on studying the dynamics of the EDA model in the stages of the search when it changes significantly, rather than when the model is converging towards a point (as will be the case towards the end of a run with a larger number of generations). Here the analysis is a "representative" or "typical" run for the algorithm on each problem. The algorithm's behavior will vary from run to run and by looking at a number of runs, the analysis is based on a typical run.

5.5.1 2D Elliptical Function

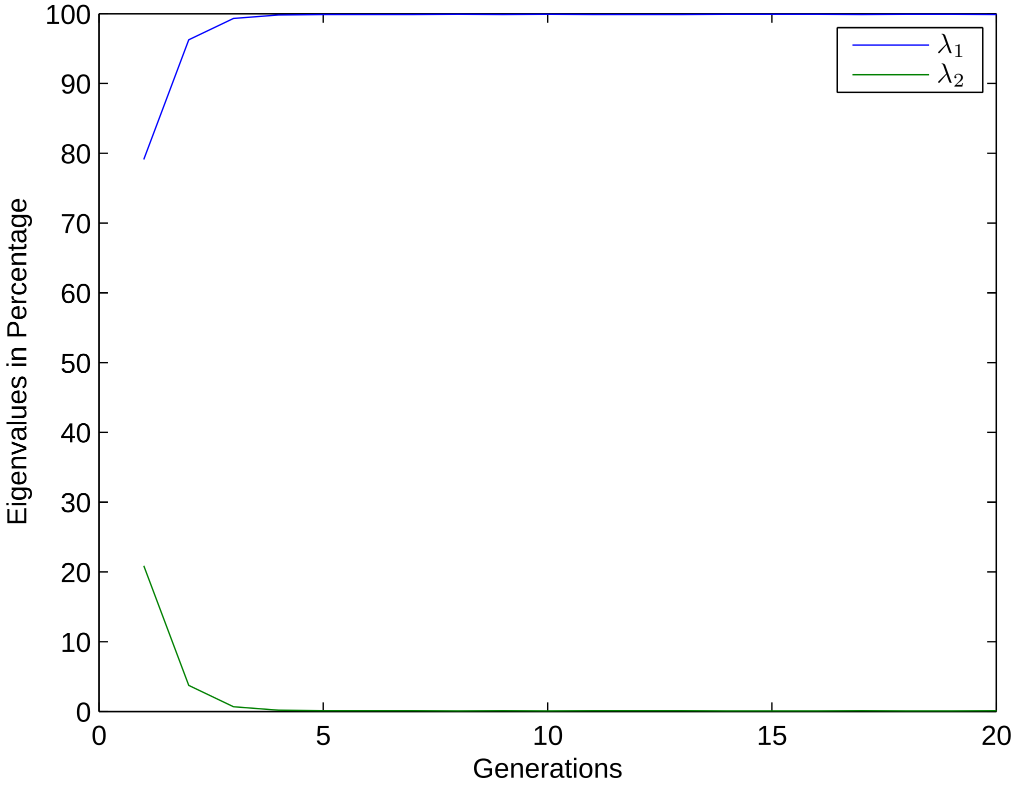
In Section 4.7, 2D Elliptical function falls into category 2, where there are no correlated variables, but some specific key structural and important variables in the problem landscape.

EMNA$_{\text{global}}$ effectively converges at the 19th generation, i.e., the absolute difference between the maximum mean of the two consecutive generations is less than or equal to 1e-06. Hence the analysis is represented for the first 19 generations. The mean graph is shown in Figure 5.2a. In this case, Gaussian model has degenerated and converged. The mean graph (Figure 5.2a) confirms that the model cannot make progress in x_2 direction.

The correlation coefficient r_{12} at any of the 19 generations for this experiment is very close to zero

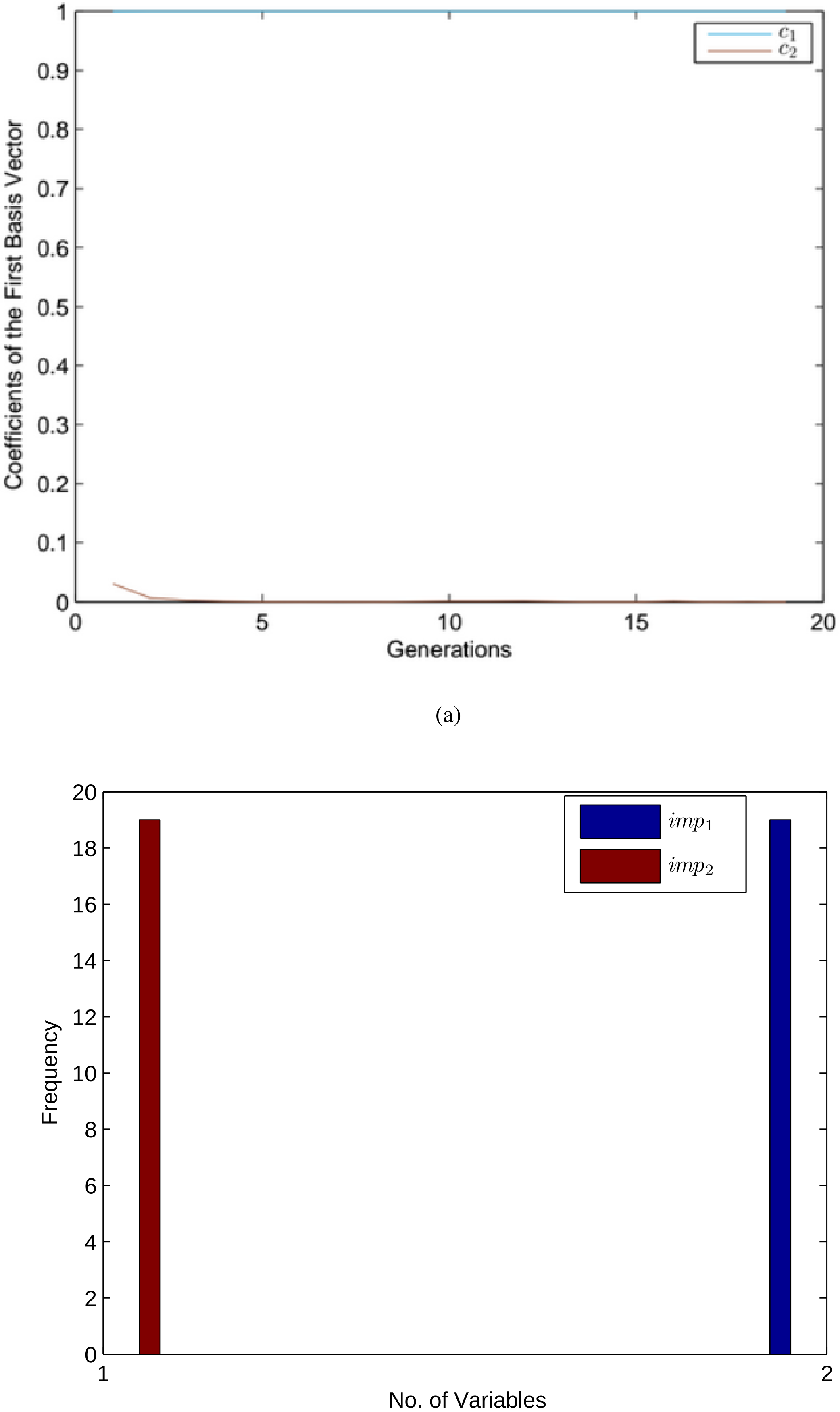


(a)



(b)

Figure 5.2: Results are based over 19 generations of EMNA$_{\text{global}}$ for the 2D Elliptical Function, (5.2a) Mean of EMNA$_{\text{global}}$, (5.2b) Eigenvalues Analysis.



(a)

(b)

Figure 5.3: Results are based over 19 generations of EMNA$_{\text{global}}$ for the 2D Elliptical Function, (5.3a) Key Structural Variables for the First Eigenvalue, (5.3b) Important Variables to the 2D Elliptical Function captured by EMNA$_{\text{global}}$.

(i.e. $\ll$ our threshold value 0.3), confirming that there is weak correlation between the variables in the EDA model.

The results for EMNA$_{\text{global}}$ for analyzing key structural variables are shown in Figures 5.2b and 5.3a. The eigenvalues of the EMNA$_{\text{global}}$ model (Figure 5.2b), show that the first eigenvalue captures almost 90 percent of the total variance (given by λ_1). The first eigenvector coefficients are plotted generation-wise in Figure 5.3a. The value of coefficient c_1 is close to 1, indicating that over the run, the first eigenvector points in a direction that is almost parallel to the x_1 coordinate axis. This is because the selected individuals (on which the EDA model is built) are distributed according to the contours of constant height (i.e. ellipsoids) for this objective function. In this sense, x_1 is the key structural variable to the EDA model. This is in contrast to the fact that f is 10 times more "sensitive" to the value of x_2 than x_1.

Finally, the important variables of 2D Elliptical function for all 19 generations are plotted in Figure 5.3b. The blue color histogram (designated as imp_1) shows that x_2 has high values of standard deviation and mean of the elementary effects. The red color histogram (designated as imp_2) shows that x_1 is second most important variable. Hence, it depicts that x_2 is the most important variable and x_1 is the least important variable.

Therefore, for a 2D Elliptical function while implementing EMNA$_{\text{global}}$, the model did not find any correlated variables, and found a specific key structural (x_1) and an important (x_2) variable throughout the generations.

Comparing these results to the results from Chapter 4, the EDA model captures the same properties as the problem appears to have been based on sampling.

Since the analysis found approximately zero correlation between the variables, UMDA$_c$ could be expected to perform well in this function.

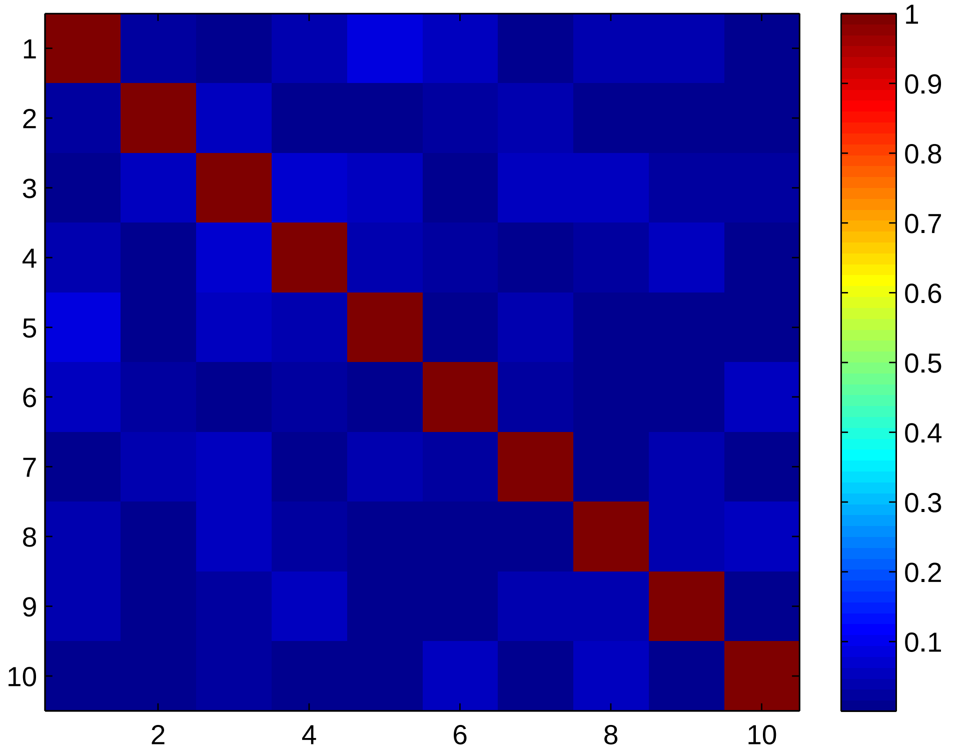
5.5.2 10D Sphere Function

The perfect example of a category 1 problem as defined in Section 4.7 is the Sphere function. Category 1 describes problems which have negligible correlation, no specific key structural variables and no specific order of important variables.

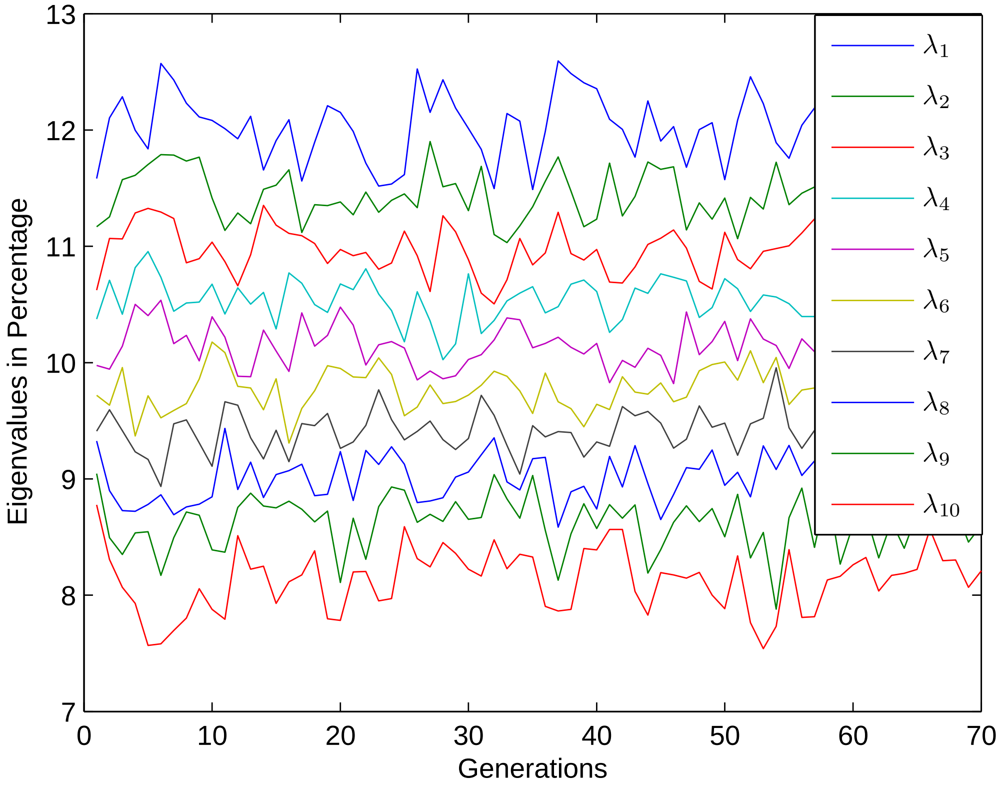
While implementing EMNA$_{\text{global}}$ on the 10D Sphere function, the model has typically converged after the 70th generation (according to the criteria defined above). Therefore, all the analysis has been done for 70 generations only.

Regarding the correlation in the 10D Sphere function, the result is shown in Figure 5.4a. Figure 5.4a shows the correlation (i.e. correlation coefficients) between the variables for this function at the 70th generation. The picture was similar across all generations, also indicating that there is very scant correlation between the variables, because the correlation coefficients over all generations are very close to zero (i.e. $\ll$ our threshold value 0.3).

The results of the analysis for EMNA$_{\text{global}}$ on the 10D Sphere function for analyzing key structural variables are shown in Figures 5.4b and 5.5a. Figure 5.4b plots the eigenvalues over 70 generations, which shows that there is not so much difference between the eigenvalues (capturing between 8 and 12% of the total variance). This is to be expected, since if the data was perfectly spherical (as the

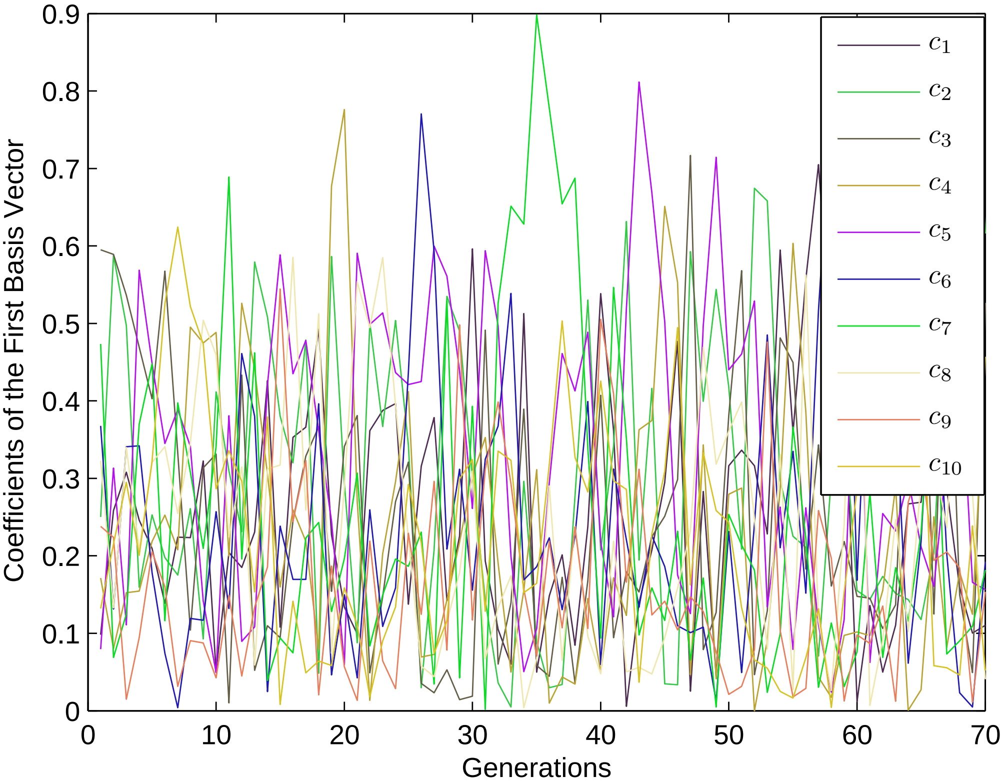


(a)

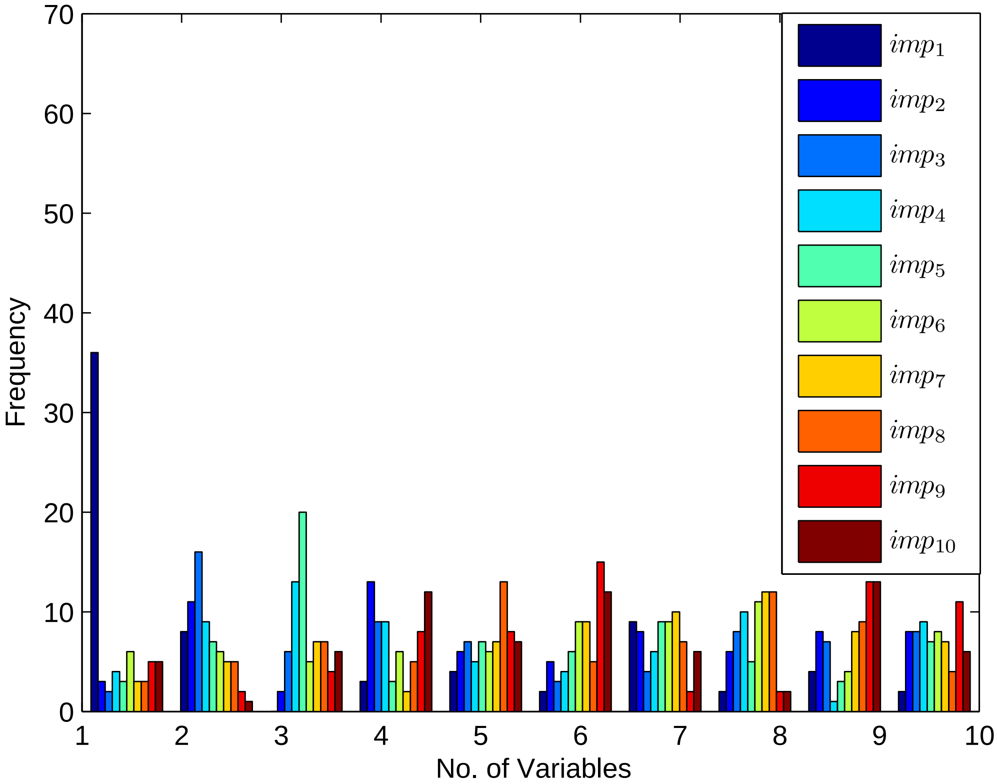


(b)

Figure 5.4: Results are based over 70 generations of EMNA$_{\text{global}}$ for the 10D Sphere Function, (5.4a) Correlation between Variables at the 70th generation, (5.4b) Eigenvalue Analysis for 70 generations.



(a)



(b)

Figure 5.5: Results are based over 70 generations of EMNA$_{\text{global}}$ for the 10D Sphere Function, (5.5a) Key Structural Variables for the First Eigenvalue, (5.5b) Important Variables to the 10D Sphere Function captured by EMNA$_{\text{global}}$.

sample size grows), the most important direction in the search space is arbitrary. The values of the coefficients (c_1 to c_{10}) for the first eigenvector (Figure 5.5a) are examined; no clear structure or trend is observable. This happens due to the properties of the Sphere function, where the EDA modeling process based on selected data does not drive the model variance to be larger in any particular direction. Hence, the key structure of variables are random in nature. Essentially, the results tell that there is no specific key structural variables to EMNA$_{\text{global}}$ in the Sphere function. However, these results provide a useful "baseline" in the sense that results on other functions can be visually compared to this case.

The spread shown in Figure 5.4b is taken as a standard for deciding the type of 10D functions. If the spread of eigenvalues of the function is the same as the Sphere function, then the function is considered as a Sphere type (the data of the selected population is spherical); otherwise, the function is an Elliptical type (the data of the selected population is elliptical).

Finally, the identification of important variables in the 10D Sphere function is shown in Figure 5.5b. The histogram shows that, for each variable, the number of times the variables are important is represented in different color formats. The results show that, no particular variable is important throughout the 70 generations. It also shows that the number of occurrences of important variables for each variable is similar. Although, x_1 is the most important for about 38 times, it does not show that x_1 is most important throughout the 70 generations. Hence, no variables are specifically important to the model.

To summarize the analysis of 10D Sphere function, while implementing EMNA$_{\text{global}}$, there is weak correlation, no specific key structural variables and the model does not identify any specific important variables to the objective function. Therefore the model agrees with the properties of problems of type category 1. Literature shows that UMDA$_c$ is performing better in this type of function ([43, 41], Chapter 8 of [15]).

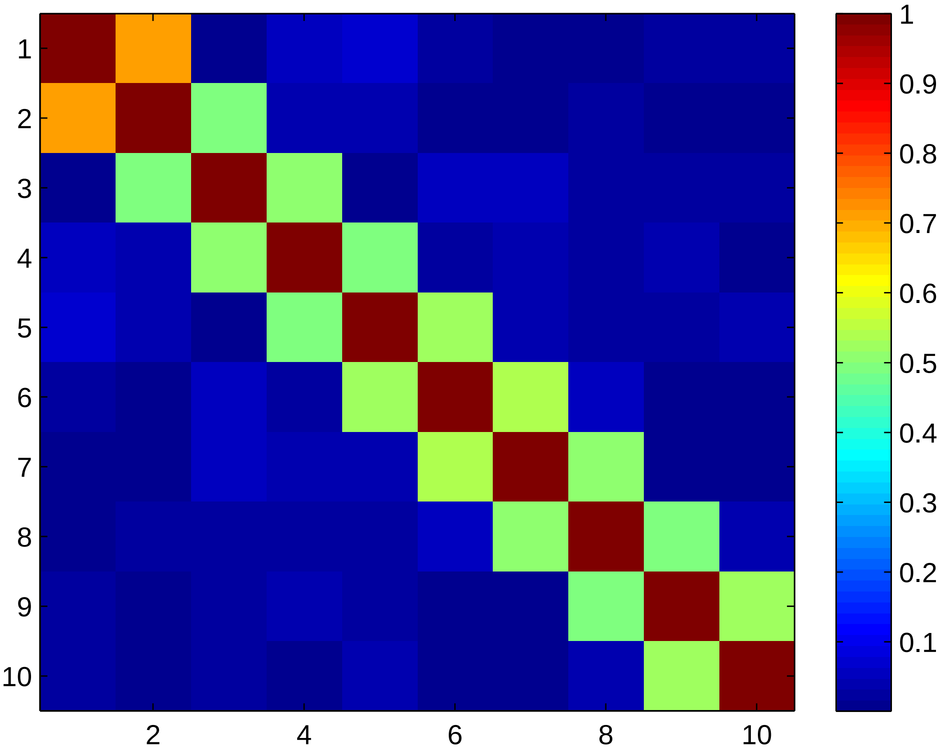
5.5.3 10D Rotated Ellipsoid Function

The 10D Rotated Ellipsoid is a category 3 type function (discussed in Section 4.7). While EMNA$_{\text{global}}$ is implemented on this function, the model typically converges at the 95th generation.

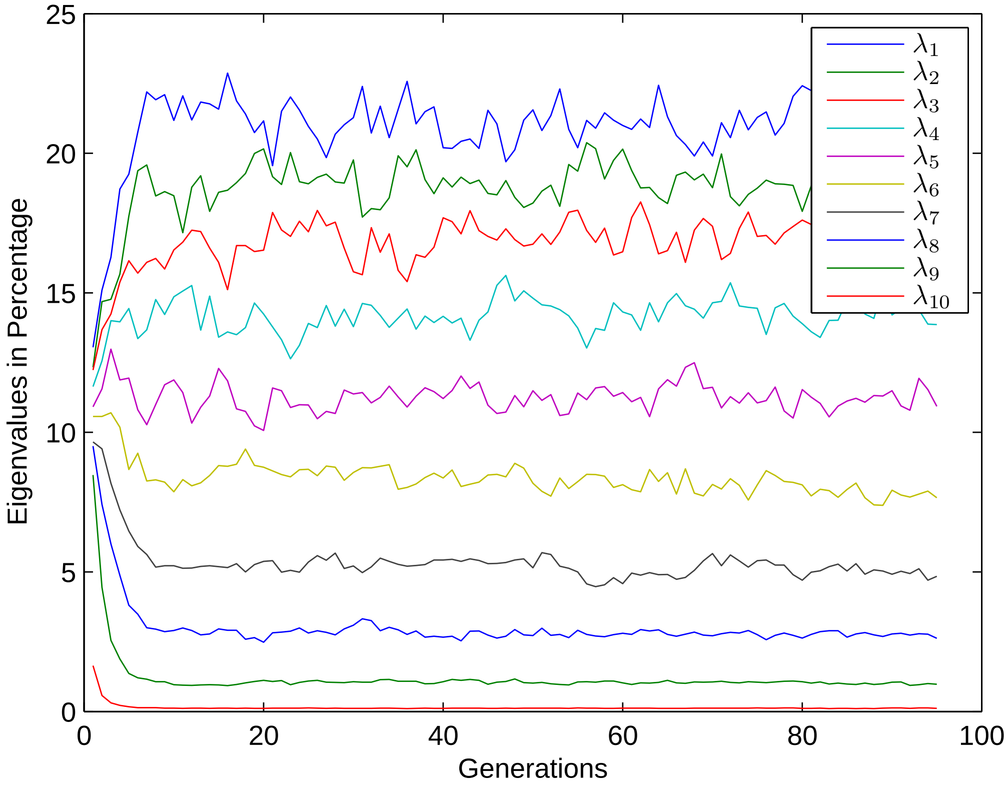
The correlation coefficients at any generation for this experiment (Figure 5.6a) follow a specific pattern, which shows that x_1 is dependent on x_2, x_2 is dependent on x_1 and x_3, x_3 is dependent on x_2 and x_4 and so on.

The eigenvalues of the EMNA$_{\text{global}}$ model over all 95 generations (Figure 5.6b) show a range of values, but this range is larger than it is for the Sphere function (Figure 5.4b) suggesting that the Rotated Ellipsoid Function is elliptical in shape. However, the first eigenvalue here does not account for a dominating amount of variability and therefore will not give strong information about the key structural variables to the EDA model.

For identifying the important variables to the objective function captured by the EDA model, the values are plotted in the form of a histogram (Figure 5.7b). Examining the number of occurrences of important variables for each variable, the results show that x_2 is typically the most important variable, with x_3 typically the second most important. Following this, x_1 and x_4 are next in importance to the

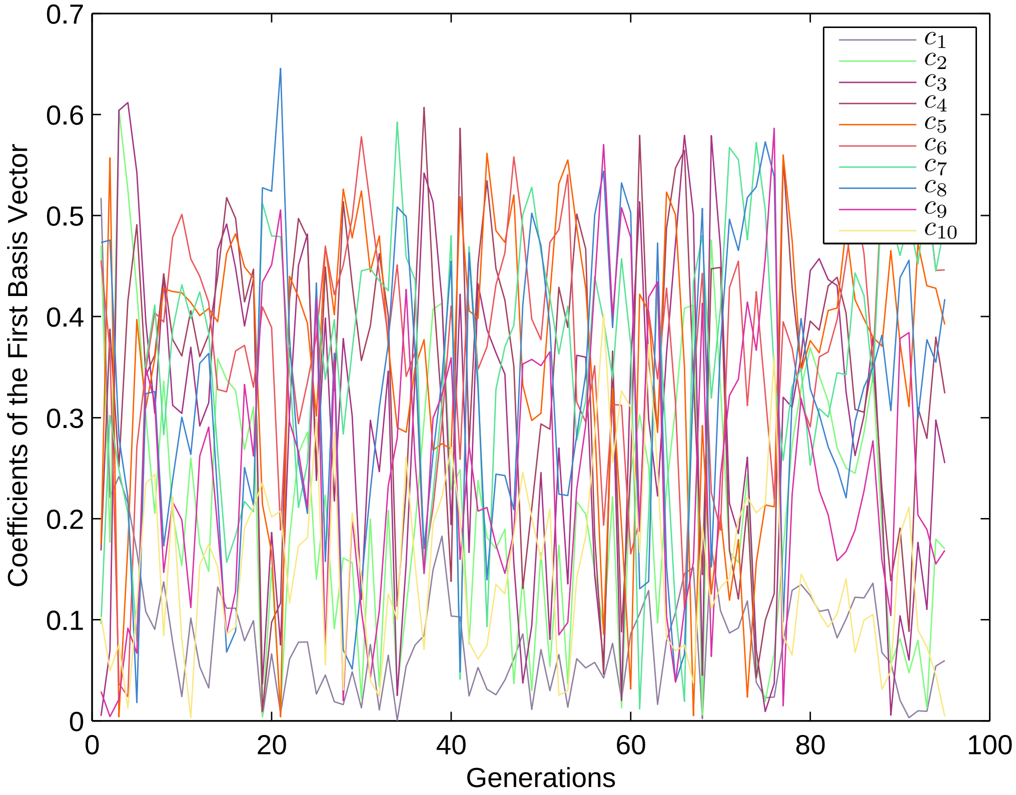


(a)

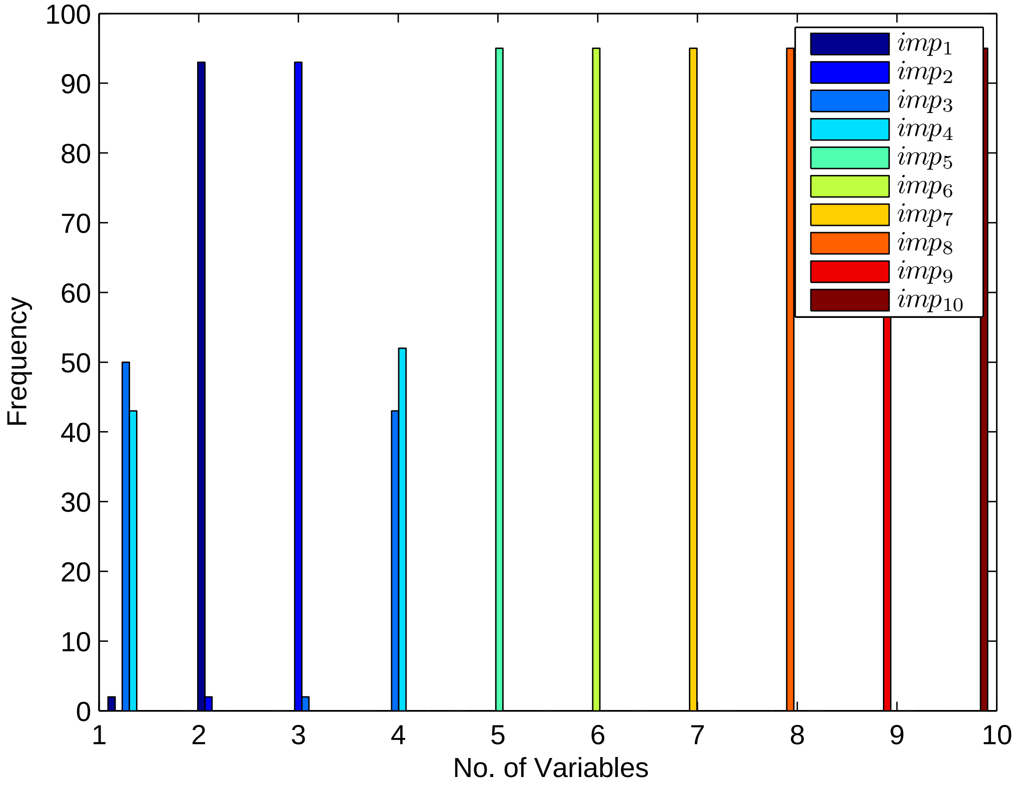


(b)

Figure 5.6: Results are based over 95 generations of $\mathrm{EMNA}_{\mathrm{global}}$ for the 10D Rotated Ellipsoid Function, (5.6a) Correlation between Variables at the 95th generation, (5.6b) Eigenvalue Analysis.



(a)



(b)

Figure 5.7: Results are based over 95 generations of $\text{EMNA}_{\text{global}}$ for the 10D Rotated Ellipsoid function, (5.7a) Key Structural Variables for the First Eigenvalue, (5.7b) Important Variables to the the 10D Rotated Ellipsoid Function captured by $\text{EMNA}_{\text{global}}$.

model. Finally, the importance of variables from $x_5, \ldots, x_{10}$ are consistently ordered over the 95 generations.

These results show that the EMNA$_{\text{global}}$ model captures the properties of category 3. Here, the properties of the problem variables are well matched with the model variables, for example, the correlation graph of the Rotated Ellipsoid function presented in Figure 4.3 is similar to the correlation graph (Figure 5.6a) captured by the EDA model. The key structural variables for both the problems and the EDA model are not specific. In addition, the variables follow the same order of importance as the objective function specification. An EDA with full covariance matrix will perform better than UMDA$_{\text{c}}$ as EMNA$_{\text{global}}$ is well-suited in this case. Given these analysis results, it would be expected that the features of the EMNA$_{\text{global}}$ result in good performance on the 10D Rotated Ellipsoid function.

5.5.4 10D Rosenbrock Function

The Rosenbrock function is known to be a hard problem for many black box optimization algorithms. It is a category 4 type function (see Section 4.7). The following experimental analysis will give some understanding about the correlation, key structural variables and important variables on EDA model in the 10D Rosenbrock function.

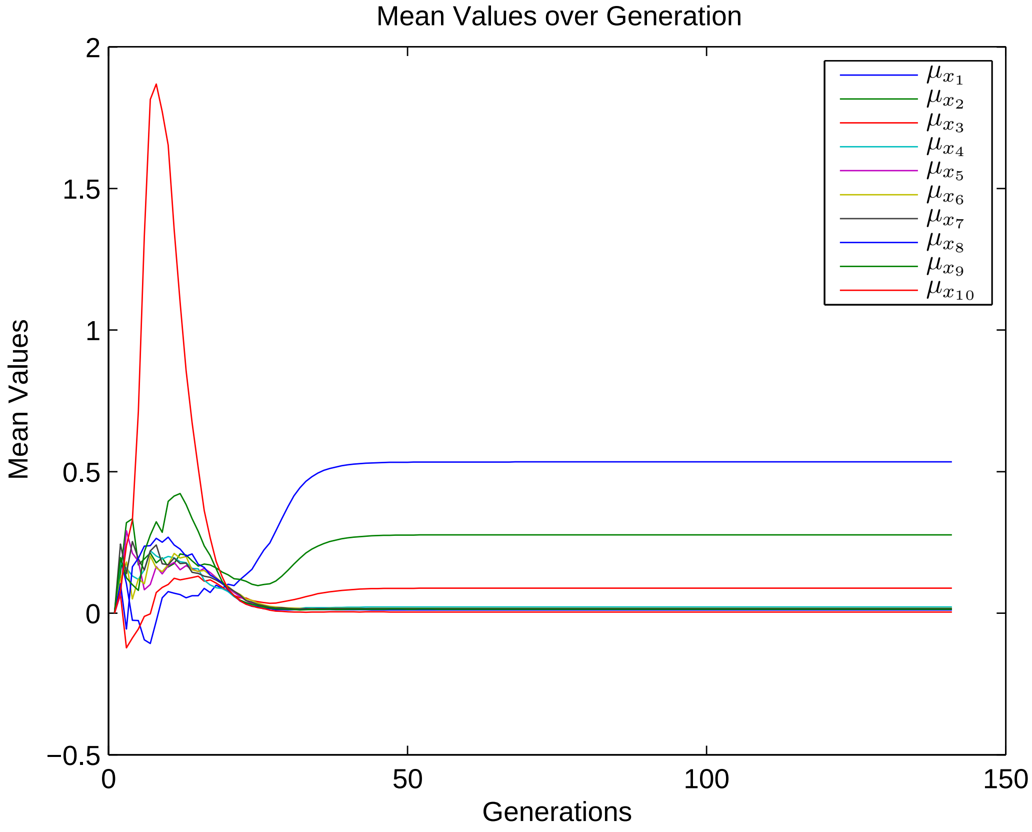
EMNA$_{\text{global}}$ converges after the 141th generation, i.e., the difference between the maximum mean between two consecutive generations is less than or equal to 1e-06. Figure 5.8a shows the dynamics of the EDA model mean vector, which indicates that the algorithm converges; however, Figure 5.8b shows that the model does not achieve the value nearest to its global optimum (global optimum = 0.0).

Firstly, the correlation between variables has been examined. Here, the correlation between variables changes over generations and settles on some pattern after about 45 generations. But this correlation structure does not produce further improvements in solution quality, because the variance and covariance of the Gaussian has shrunk towards zero (Figures 5.9a and 5.9b).

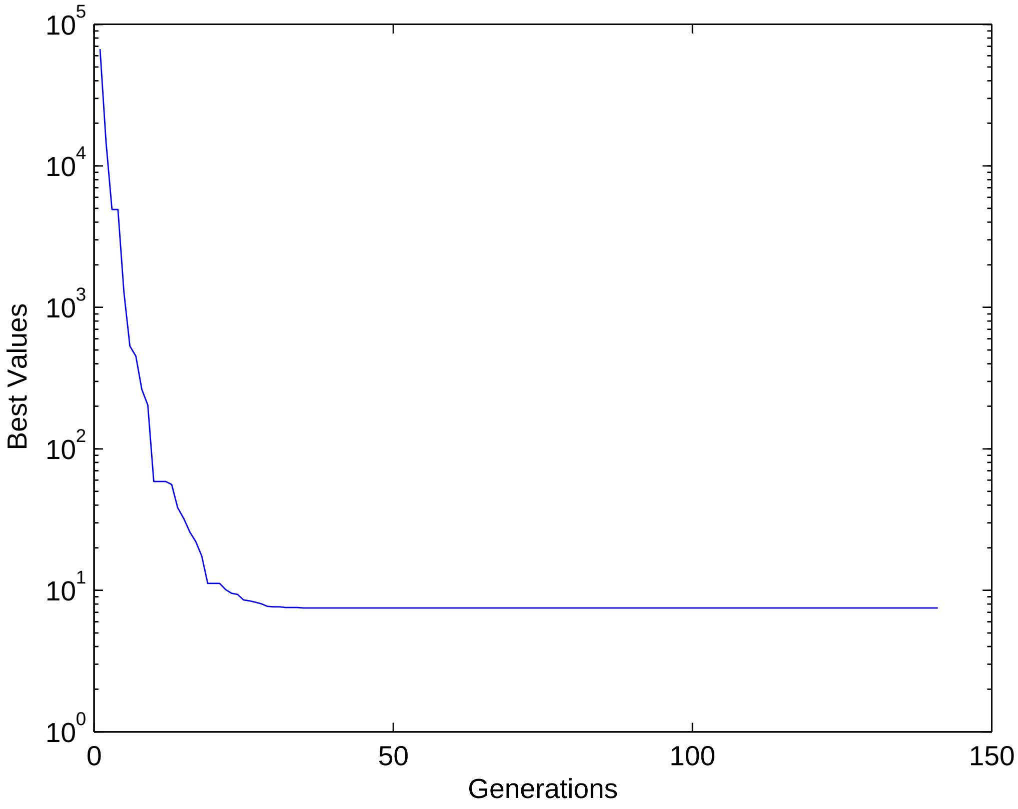
Furthermore, Figure 5.10a shows that the first eigenvalue (as indicated by the eigenvalue λ_1) captures most of the variance. The range of λ values is larger here. Examining the coefficients of the first eigenvector indicates that c_{10} is greater than all the other coefficients in first 10 generations, after that c_1, c_2 and c_3 are larger than the others (shown in Figure 5.10b), which suggests that the first eigenvector points in a direction that is nearly parallel to x_{10} in first 10 generations, after that it points to x_1, x_2 and x_3 axis. Hence, x_1, x_2 and x_3 are the key structural variables to the EDA model after the 10th generation, but before that x_{10} is the key structural variable to the model. The analysis gives us information about the progress of EMNA$_{\text{global}}$ on 10D Rosenbrock that may help in understanding the behavior of the algorithm. That is, the main direction of variance and associated coefficient values change significantly in the first 50 generations and then change very little.

Finally, Figure 5.11 shows the important variables captured by the EDA model. It shows, out of 141 occurrences/generations, that x_2 is the most important variable 122 times. x_1 is the second most important variable and x_6 is the least important variable, whereas x_{10} is the second least important variable captured by the model during the run.

However, the analysis of variables on 10D Rosenbrock function from Chapter 4 showed that there

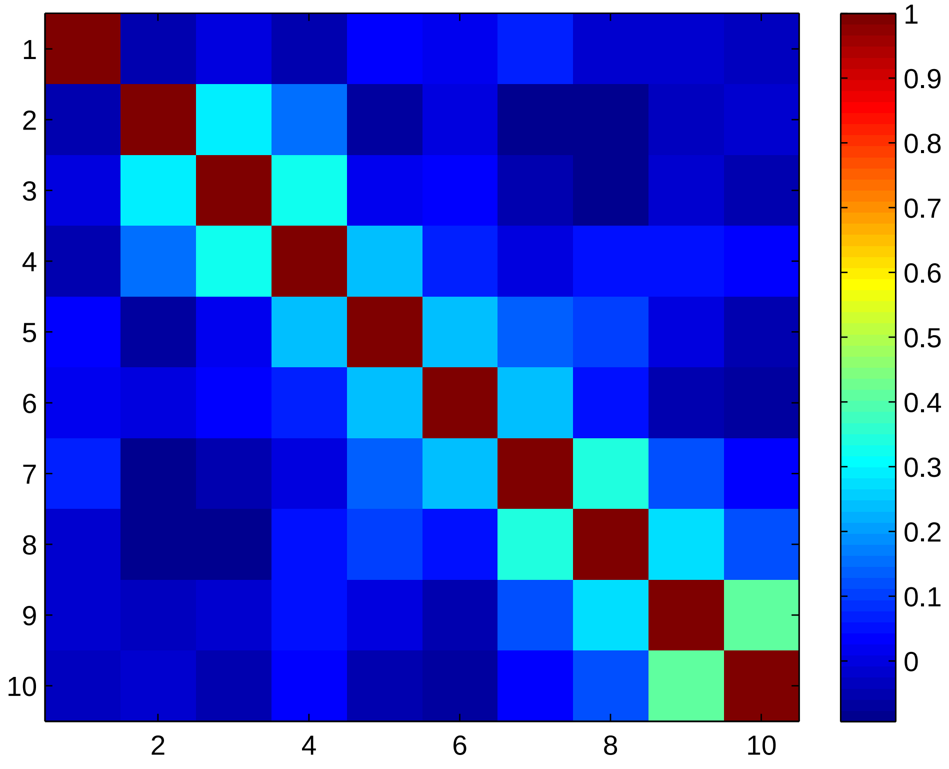


(a)

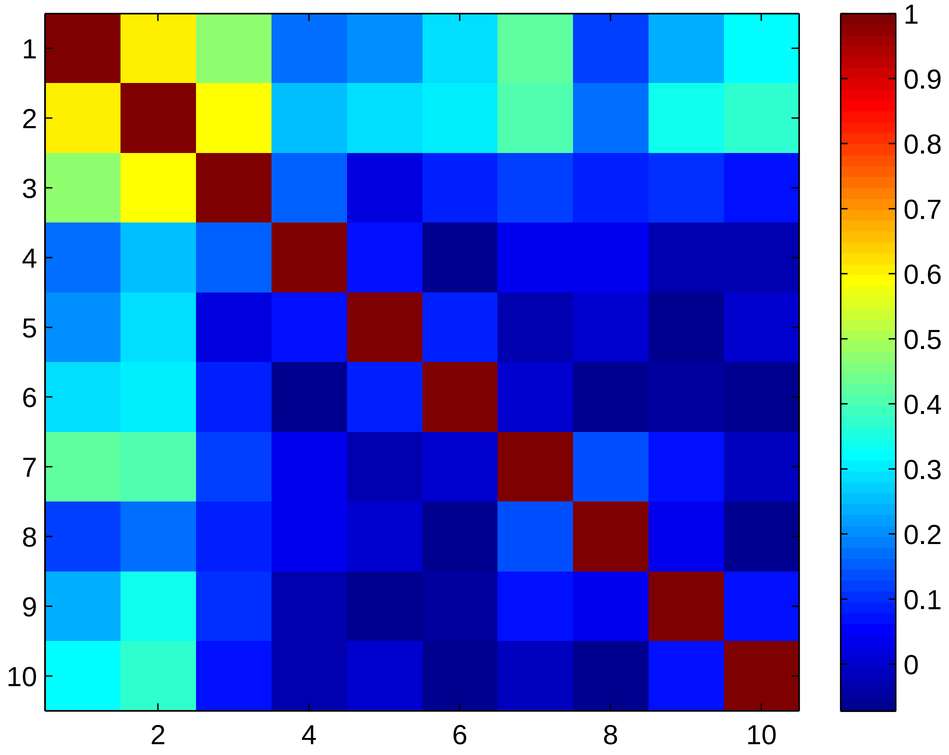


(b)

Figure 5.8: Results are based over 141 generations of EMNA$_{\text{global}}$ for the 10D Rosenbrock Function, (5.8a) Mean of EMNA$_{\text{global}}$, (5.8b) Best Values reached so far.

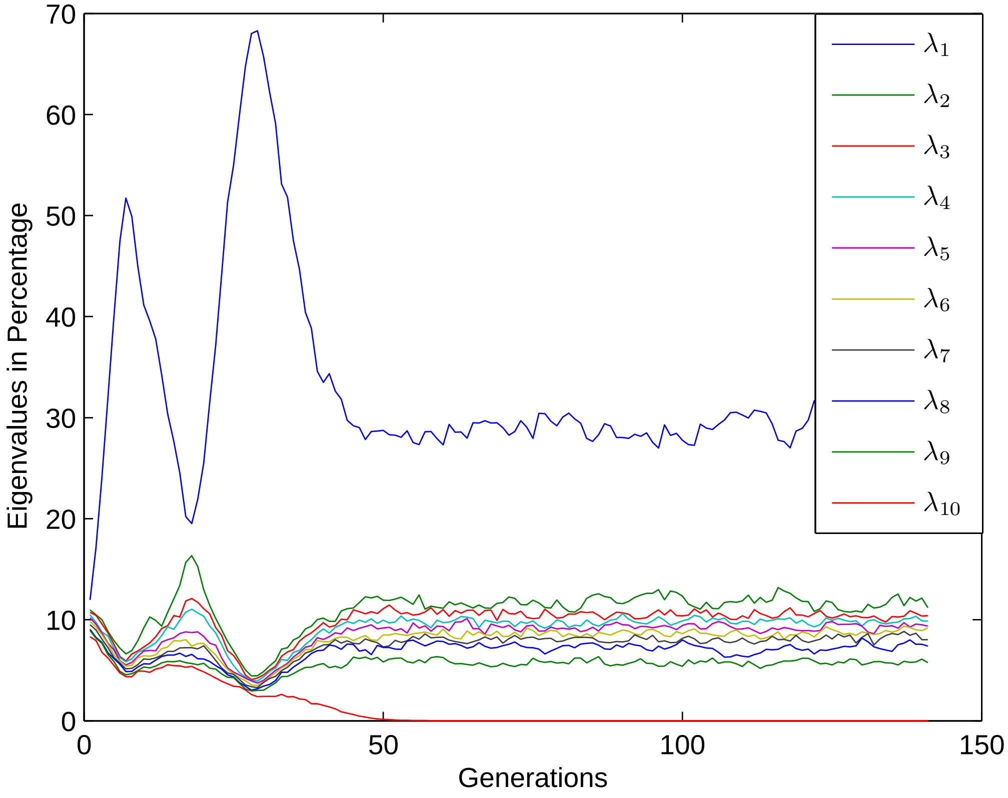


(a)

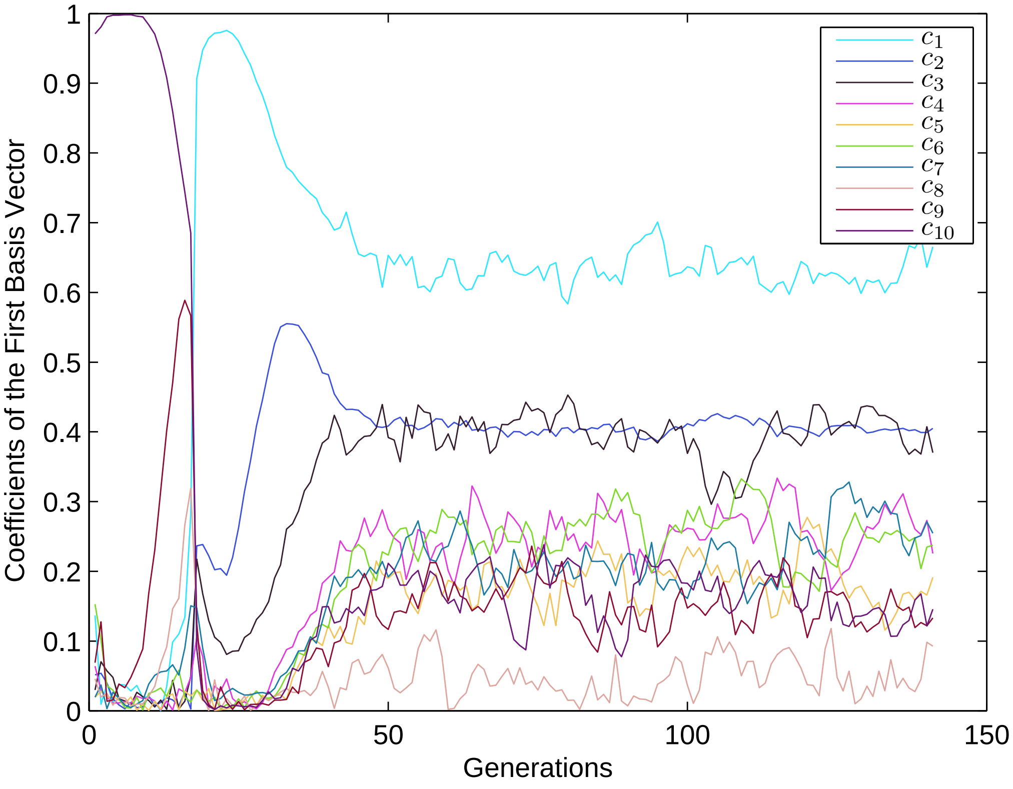


(b)

Figure 5.9: Results are based over 141 generations of EMNA$_{\text{global}}$ for the 10D Rosenbrock Function, (5.9a) Correlation between Variables at the 17th generation, (5.9b) Correlation between Variables at the 45th generation.



(a)



(b)

Figure 5.10: Results are based over 141 generations of EMNA$_{\text{global}}$ for the 10D Rosenbrock Function, (5.10a) Eigenvalue Analysis, (5.10b) Key Structural Variables for the First Eigenvalue.

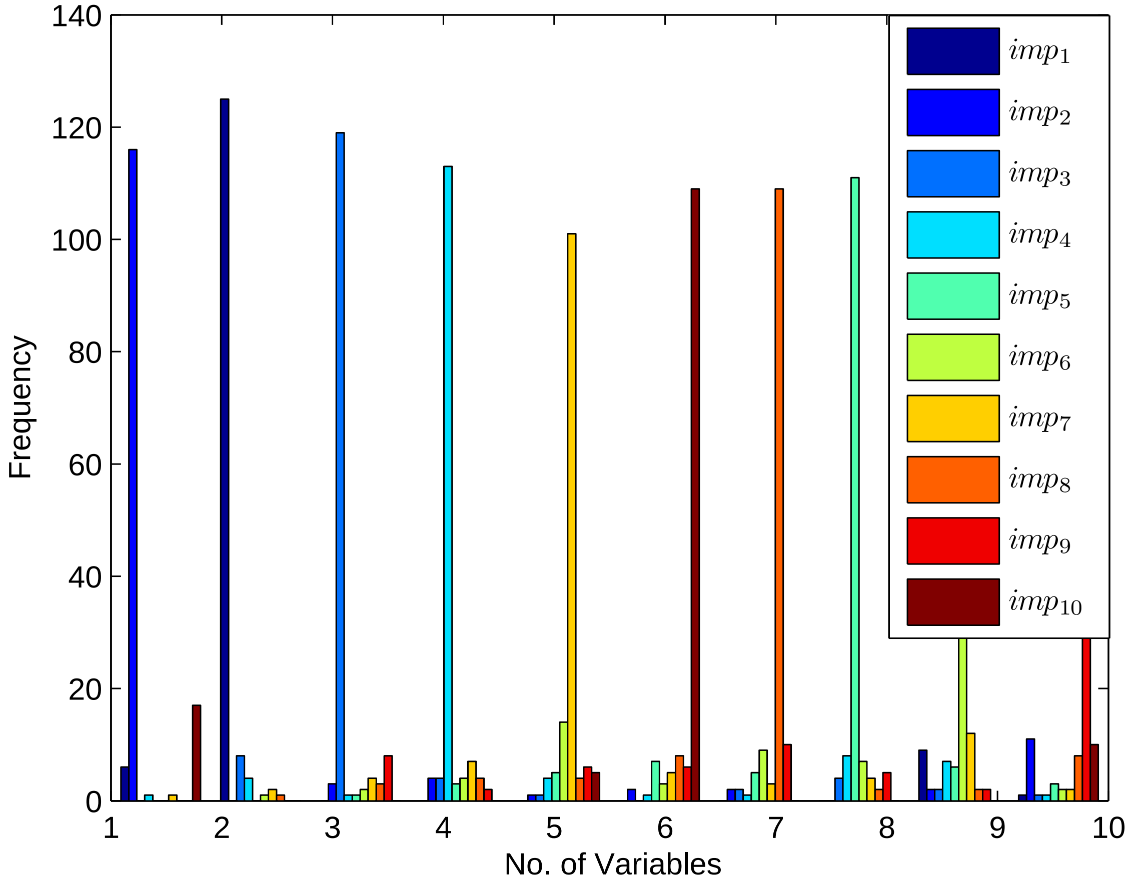


Figure 5.11: Important Variables to the 10D Rosenbrock Function captured by $\text{EMNA}_{\text{global}}$.

is a very weak correlation between variables. In addition to this, although the problem has x_{10} as its specific key structural variables, but while analyzing EDAs, only for the first few generations x_{10} ia the structural variable, but after that, it has been changed to x_1, x_2 and x_3. Regarding, the order of important variables for the problems are $x_2, \ldots, x_9$, where as, x_2 is the most important variable and x_1 is the second most important variables.

In summary, the results of the analysis of the EDA model on the 10D Rosenbrock function is contrary to the analysis found for the category 4 problem variables (see Section 4.7). Literature shows that EDAs do not perform well in this function ([43, 41], Chapter 8 of [15]).

5.6 Variable Analysis on Real World Problems

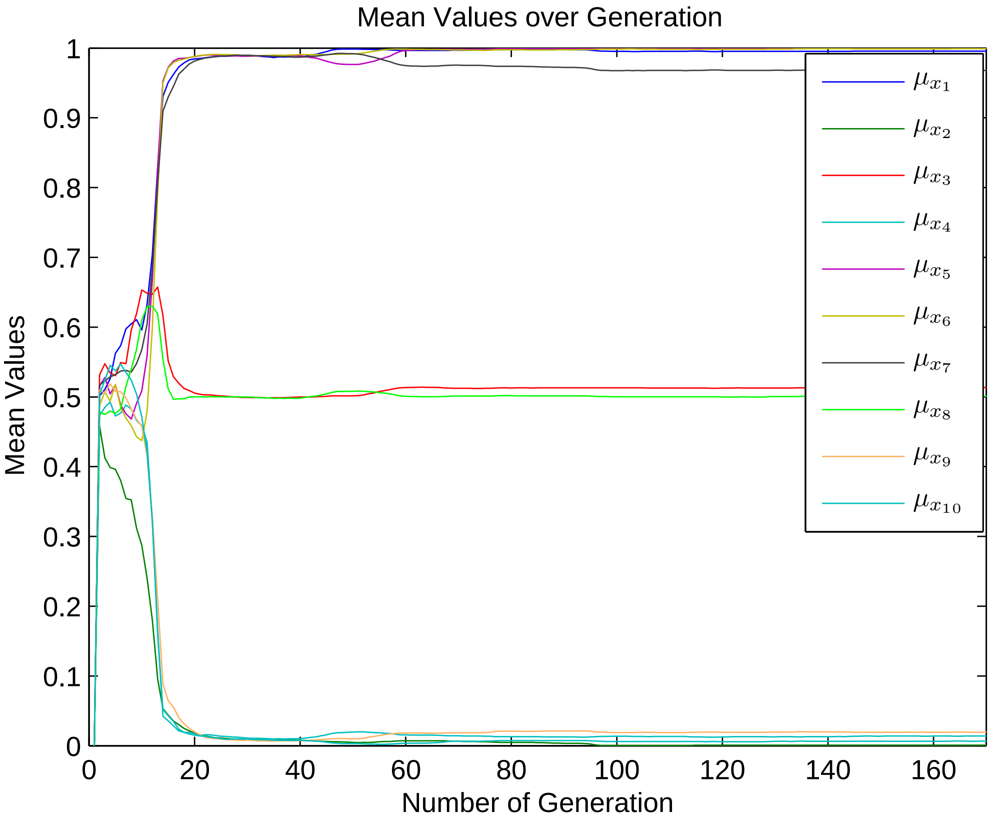
5.6.1 Circles in a Square (CiaS) Packing Problem

The analysis was applied to the Circles in a Square Packing problem, with n_c = 5, i.e. a 10D optimization problem. This was categorized in Section 4.7 with Category 5.

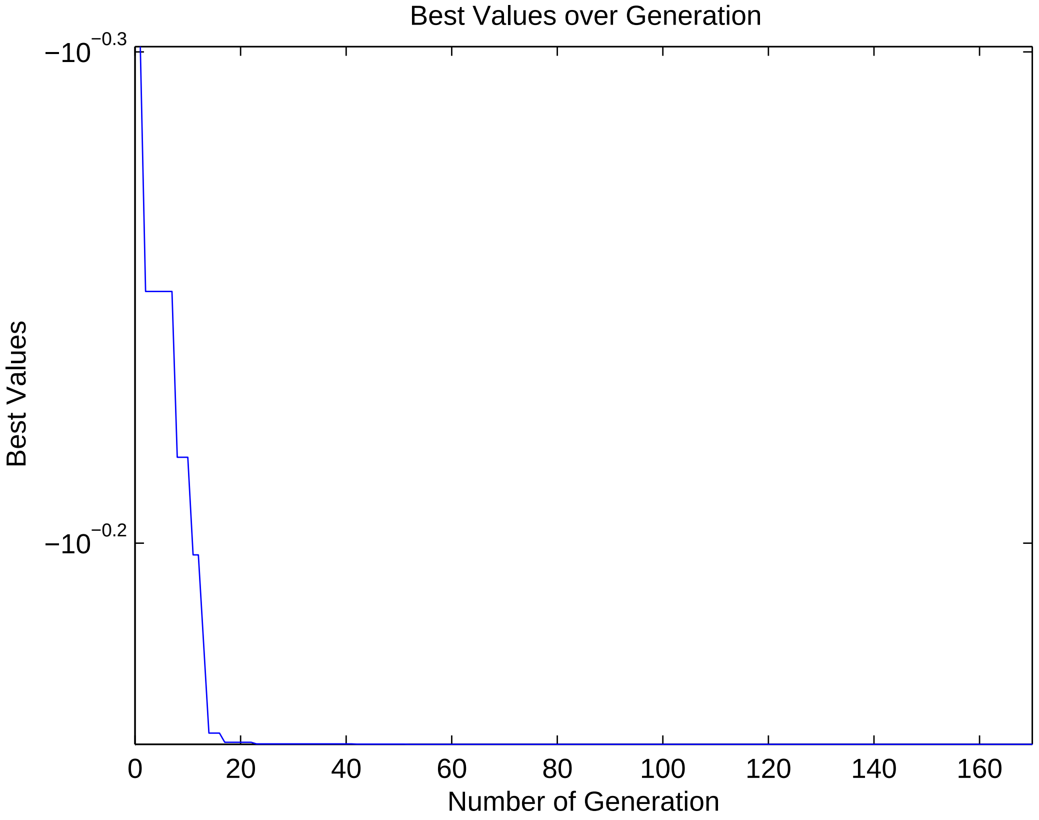
The constraints of the CiaS problem involve box-bounding, as mentioned in Section 4.3. Since Gaussian EDA models can generate component solutions values in the range of $[-\infty, +\infty]$, infeasible solutions may be generated. To address this issue, a constraint handling technique called “Boundary Repair” has been used. It is a simple way to repair the infeasible solutions using the information given from the problem (see Equation 2.5). While running the algorithms, a set of candidate solutions for each generation is generated. The candidate solutions are examined and values lying outside the feasible region are reset to the boundary value. That is, $\forall\ \mathbf{w^i} = (w_1, w_2),\ i = 1, \ldots, n_c$, if $w_1 < 0$, then set $w_1 = 0$ or if $w_1 > 1$, then set $w_1 = 1$, with identical conditions for w_2.

EMNA$_{\text{global}}$ was applied to the 5-circle CiaS problem (i.e. 10D), with a population size of 2000 and a selection ratio of 0.2, running over 200 generations. The algorithm is set to terminate when the absolute difference between any components of the model mean of two successive generations is 1e-06 or it attains all the 200 generations. The global optimum value for this problem is -0.7071 [142]. In the present case, the EDA model converges at the 170th generation. Example performance and mean results for EMNA$_{\text{global}}$ are presented in Figures 5.12a and 5.12b. Note that this is from a typical run of the algorithm; over 25 trials the average best fitness was -0.6590 with a standard deviation of 0.0297. In this run the algorithm found a solution with a relatively high fitness value (shown in Figure 5.14a) converging on the global optimum [142]. UMDA$_c$ and EMNA$_{\text{global}}$ were able to obtain reasonable performance (i.e., within 10 percent of the global optimum value for 5 circles in a CiaS problem) [57]. Note however that some small further improvement is possible for this solution (the 4th circle can be expanded slightly to touch the boundary of the unit square). The model mean values in Figure 5.12a show that, in the first 30 generations, the model moves rapidly, with slower progress until the last generation, towards the solution vector where points are positioned at each corner of the unit square and at (0.5, 0.5).

The correlated variables in the EDA model change significantly over generations. In the 20th generation, where the EDA model is moving and changing rapidly, the correlation figure (Figure 5.13a) shows that there are no strong correlations between variables at this stage. However, at the 170th

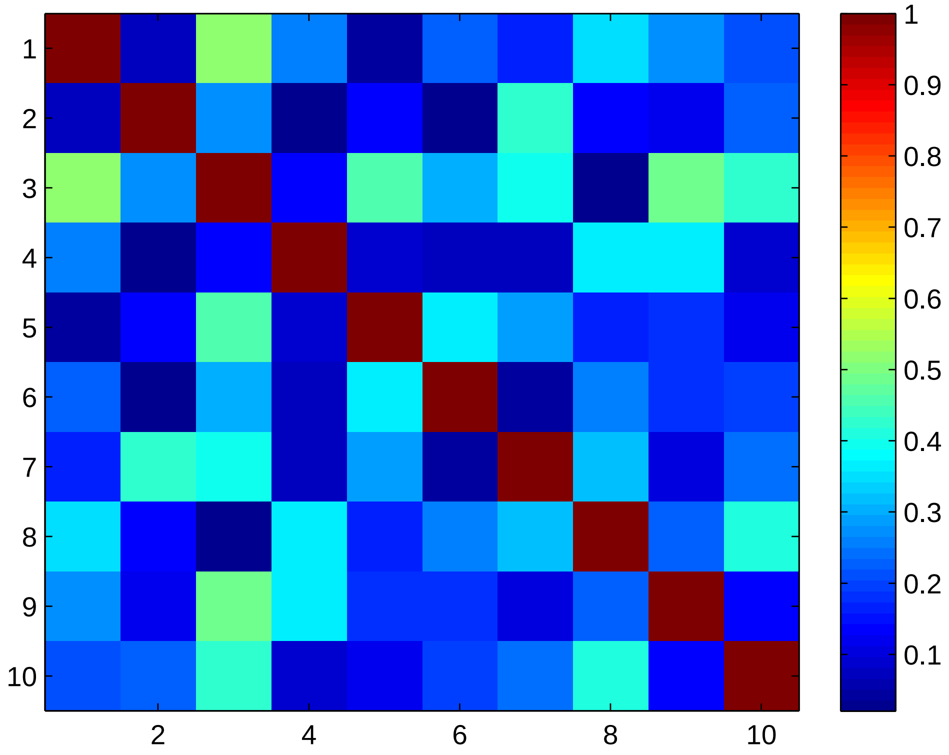


(a)

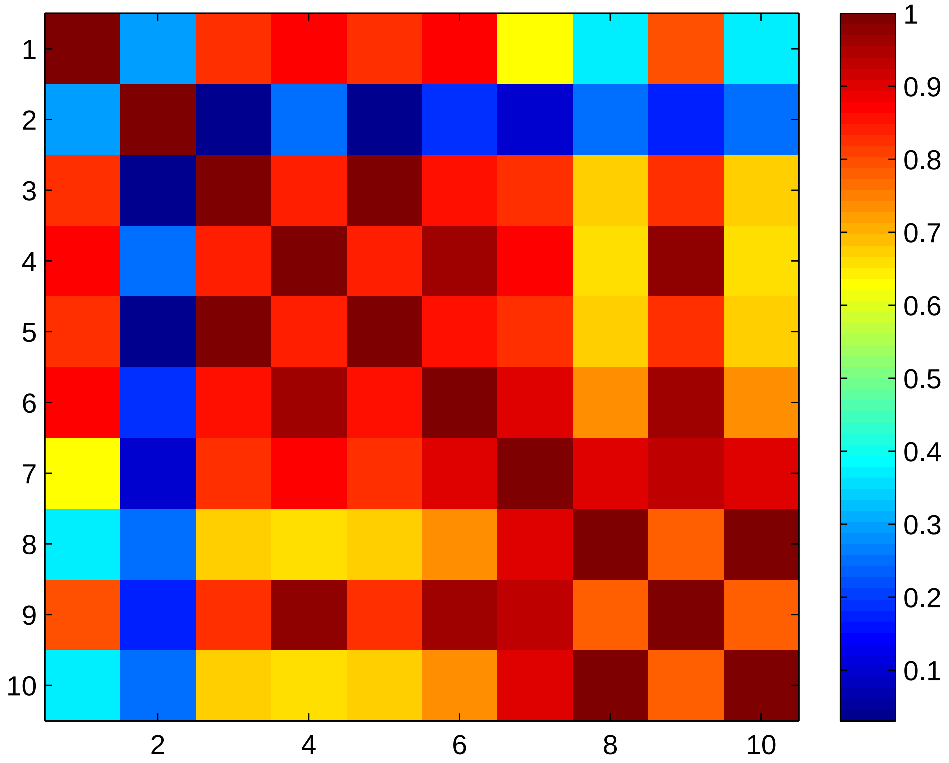


(b)

Figure 5.12: Results are based over 170 generations of EMNA$_{\text{global}}$ for the 10D CiaS Problem, (5.12a) Mean of EMNA$_{\text{global}}$, (5.12b) Best Values reached so far.

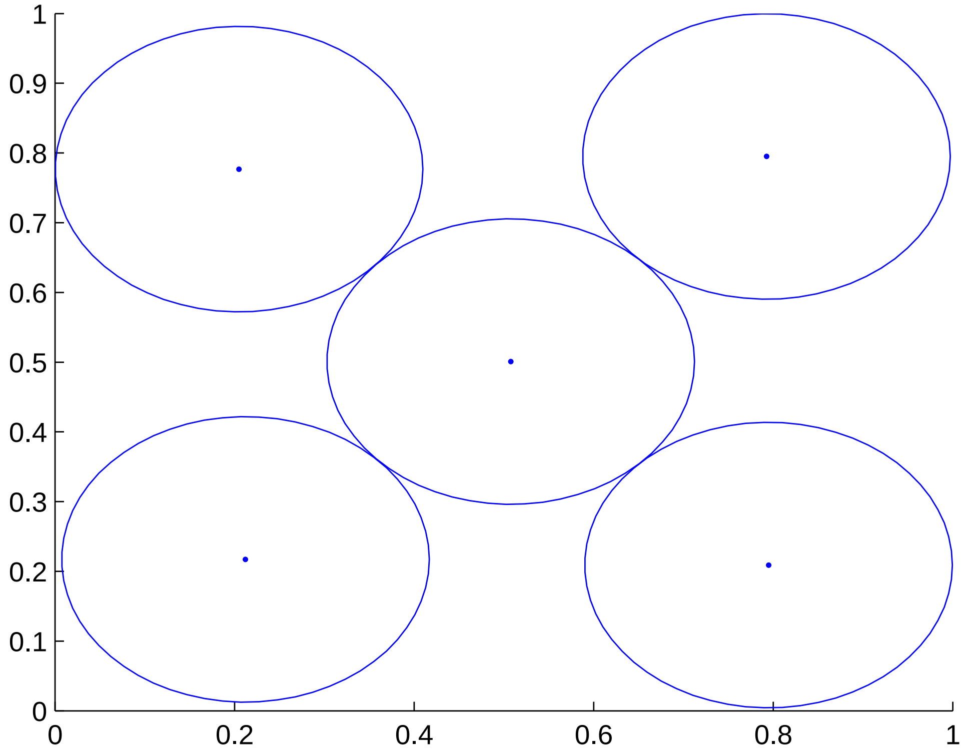


(a)

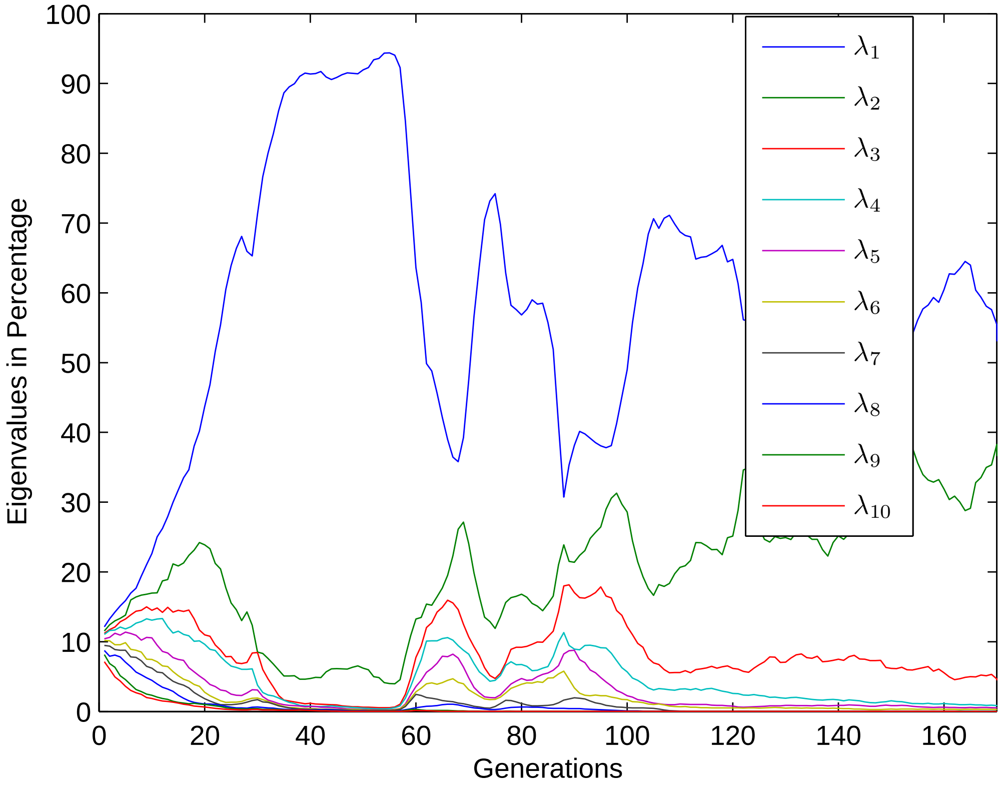


(b)

Figure 5.13: Results are based over 170 generations of EMNA$_{\text{global}}$ for the 10D CiaS Problem, (5.13a) Correlation between Variables at 20th generation, (5.13b) Correlation between Variables at the 170th generation.

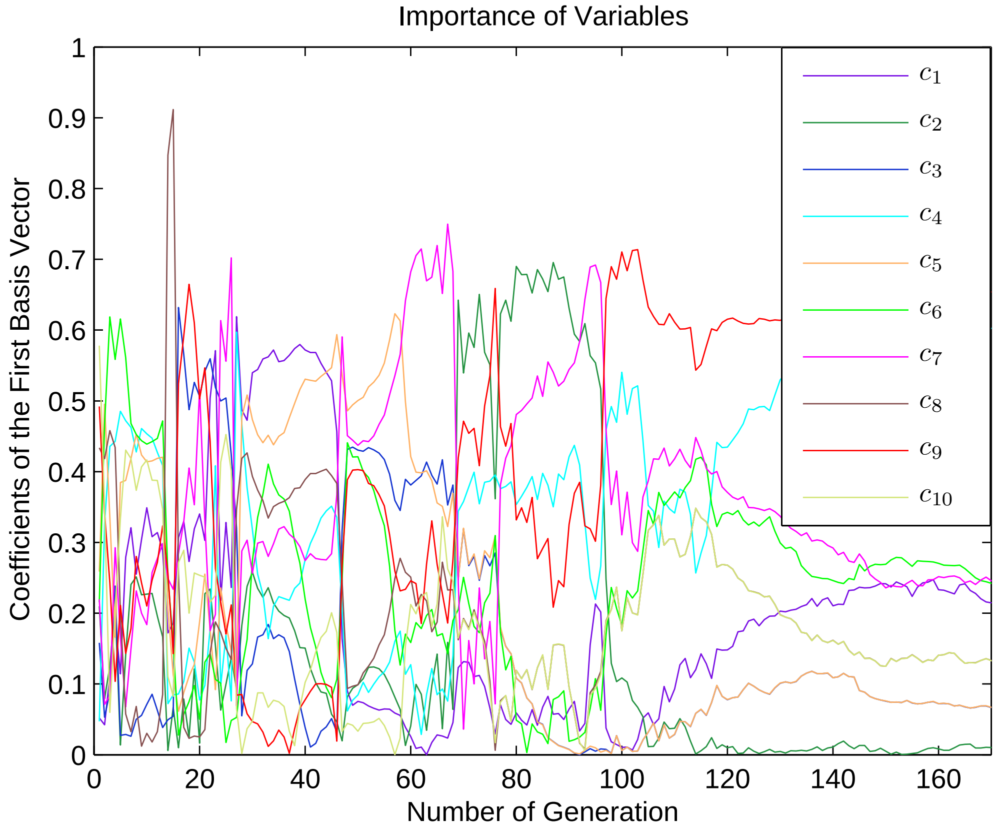


(a)

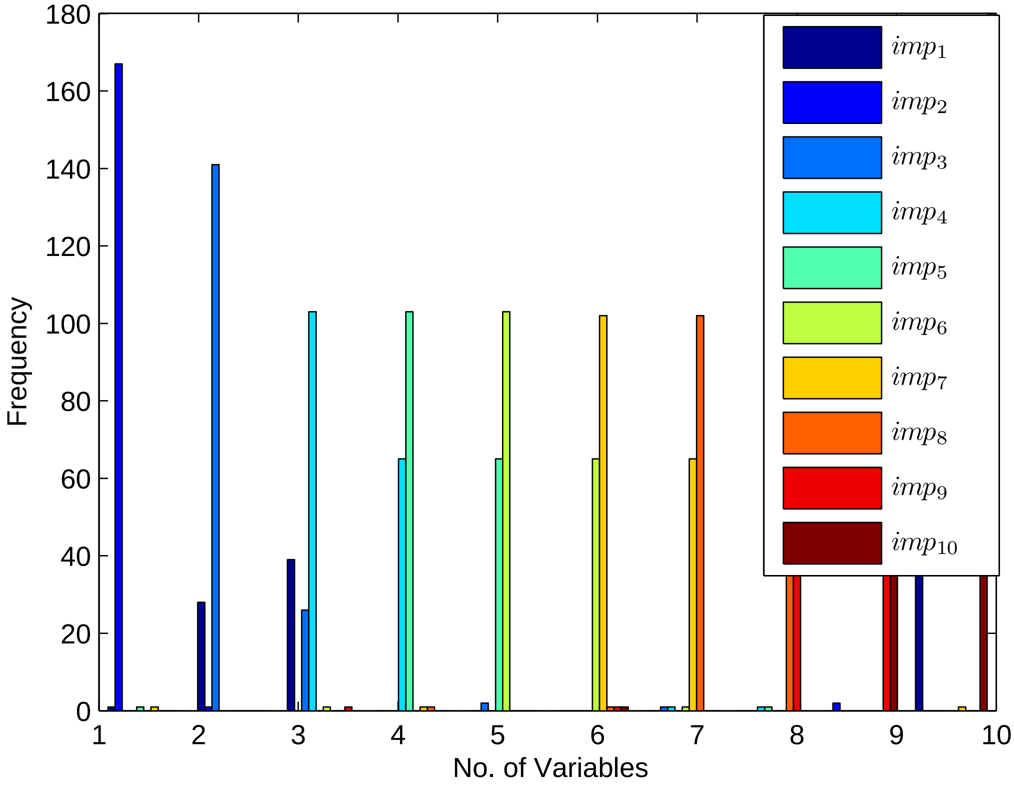


(b)

Figure 5.14: Results are based over the 170 generations of EMNA$_{\text{global}}$ for the 10D CiaS problem, (5.14a) Plotting of 5 equal circles in a unit square at the 170th generation, (5.14b) Eigenvalues Analysis.



(a)



(b)

Figure 5.15: Results are based over 170 generations of EMNA$_{\text{global}}$ for the 10D CiaS problem, (5.15a) Key Structural Variables for the First Eigenvalue, (5.15b) Important Variables to the 10D CiaS Problem captured by EMNA$_{\text{global}}$.

generation (Figure 5.13b), the second variable (i.e. the x-axis position of the second circle) is the only variable that still has weak correlation with all other variables. Other correlation values are all larger, but after 170 number of generations, the model variance has converged significantly so this structure is likely to reflect the highly local sample produced by the model.

Furthermore in this analysis, Figure 5.14b shows that the first eigenvalue captures a large amount of the variance, over most generations, hence the range of eigenvalues is larger when compared to the Sphere function, which indicates that the 5-circles CiaS problem is somewhat elliptical in shape. Over the generations, when the first eigenvalue decreases, the other eigenvalues increase smoothly, which is a visualization of the dynamics of the EDA model through the search space.

The coefficients of the first eigenvector are shown in Figure 5.15a. In the first 120 generations the values of the coefficients all change rapidly. For the remainder of the run, the values are more stable, with c_4 and c_9 having the highest values, indicating that x_4 and x_9 are key structural variables to the model in these generations. These variables correspond to the coordinates of the fourth circle in the packing solution (Figure 5.14a), showing that the EDA model was aligned towards these variables and suggesting that improvement in solution quality could be found in this direction.

The prediction of important variables can be identified by analyzing Figure 5.15b. It shows that x_{10} is the most important and least important variables most of the time, whereas, x_1 is the second most important variables to the model and x_9 is the least important and second least important variables to the model. However, in general, it can be said that the order of important variables is difficult to analyze in this case.

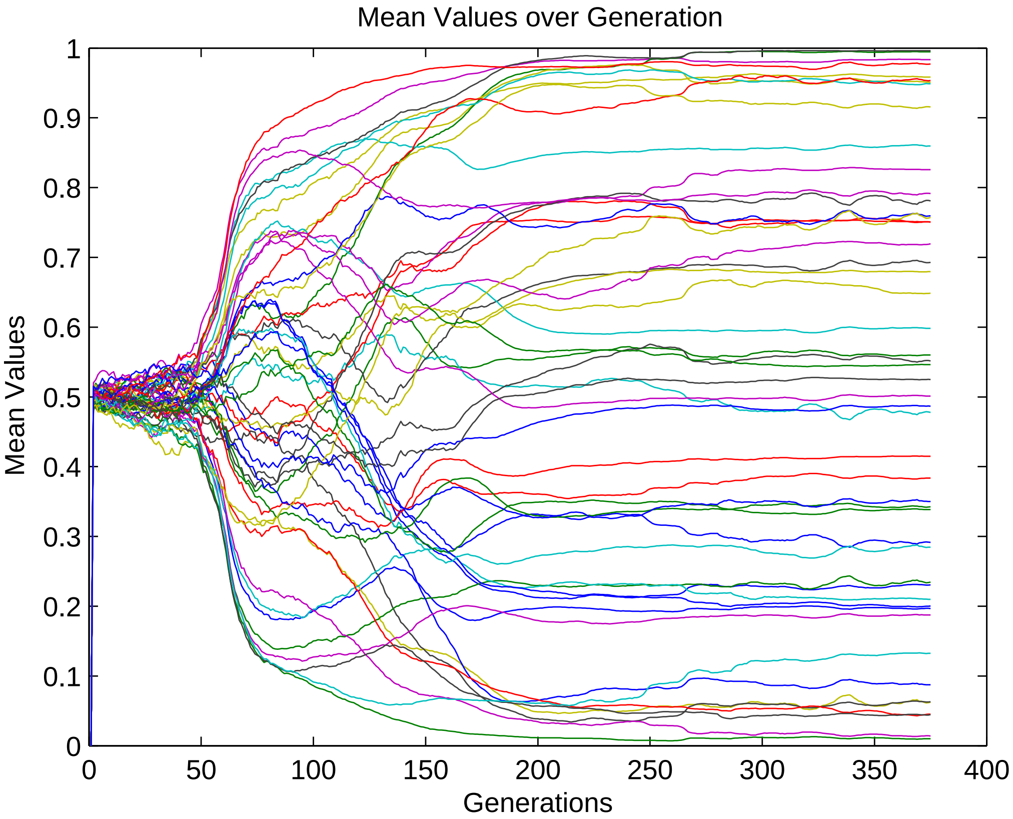
The analysis drawn from Chapter 4 for the 10D CiaS problem illustrates that the problem has correlation between variables, does not contain specific key structural variables throughout the run, and does not identify any specific order of important variables. This indicates that the problem is a type of category 5, but the analysis for the EDA model is different.

The model variables and problem variables show different patterns of correlation. While the problem variables show a pattern of correlation between variables (see Figure 4.4), the model variables do not show any pattern during the EDA run. In addition to this, λ_1 does not give any strong information about the key structural variables during the analysis of problem variables (see Figure 4.10); however, the analysis of model variables gives an indication that λ_1 is the dominant eigenvalue in most of the generations but is unable to provide specific key structural variables throughout the 170 generations. Both the model analysis and the problem analysis do not provide any specific order of important variables.

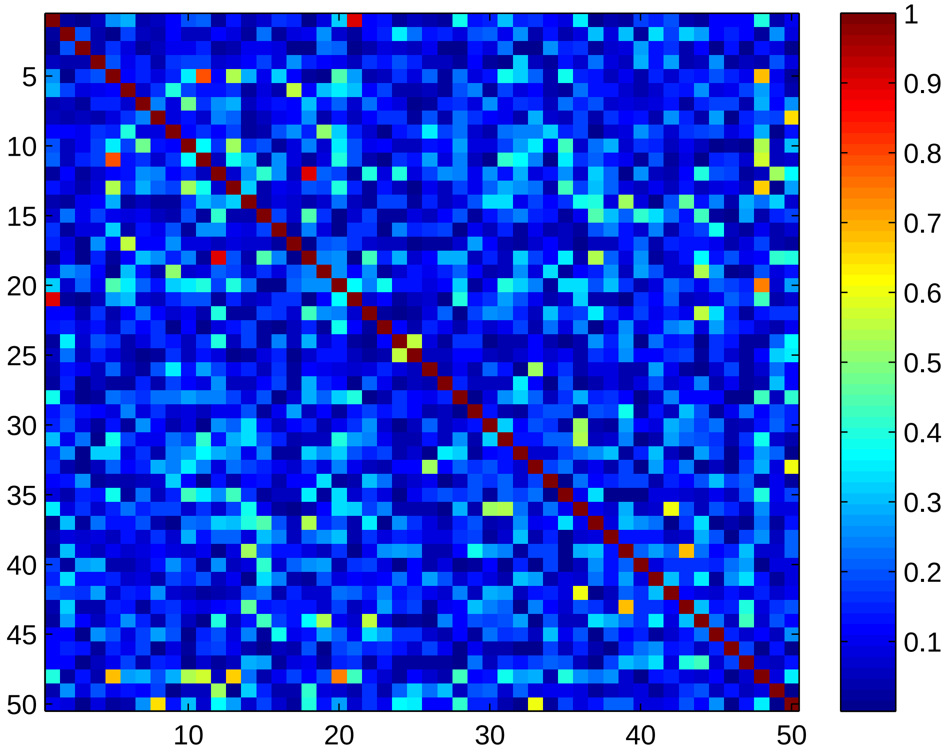
5.6.2 EDAs implemented on 25 Circles in a Square Problem

50D CiaS is a category 5 type problem (see Section 4.7). EMNA$_{\text{global}}$ was implemented on 25 circles in a square packing problem, with population of 5000 having 0.2 as its selection ratio for 500 generations. The threshold for identifying key structural variables is $\frac{1}{\sqrt{50}}$=0.1414. The algorithm will stop when the absolute difference between the mean of two successive generations is 1e-06 or after it has completed the 500 generations.

The EDA model converges after the 375th generation. The mean of the model is shown in Fig-

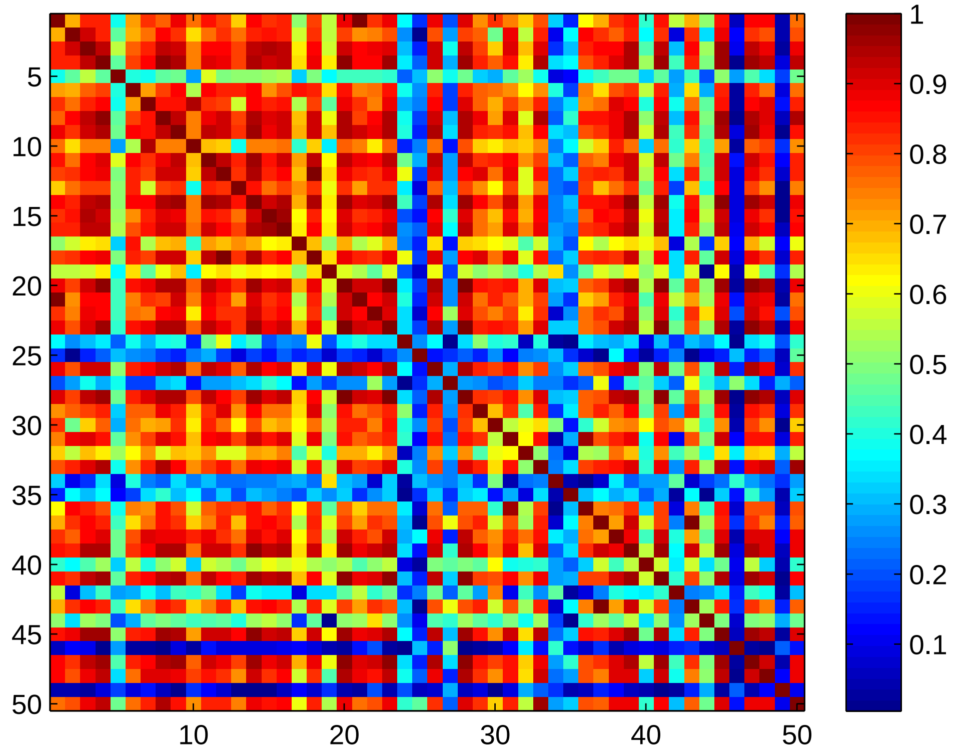


(a)

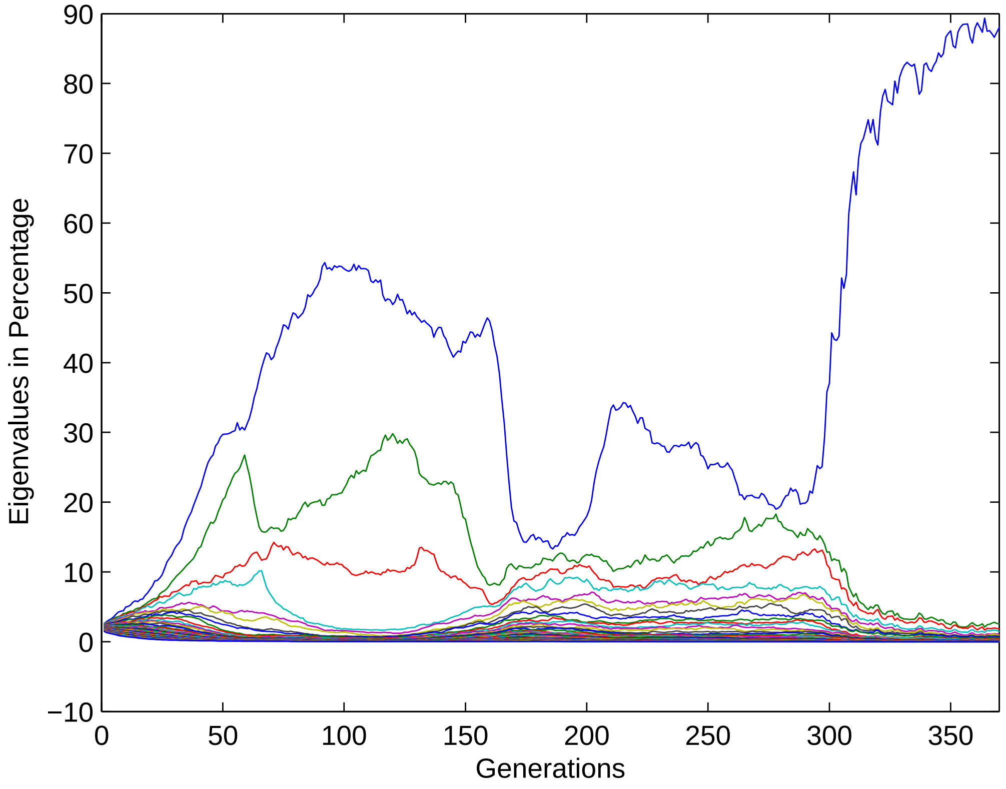


(b)

Figure 5.16: Results are based over 375 generations of EMNA$_{\text{global}}$ for the 50D CiaS Problem, (5.16a) Mean Values, (5.16b) Correlation Between Variables at the 180th generation.

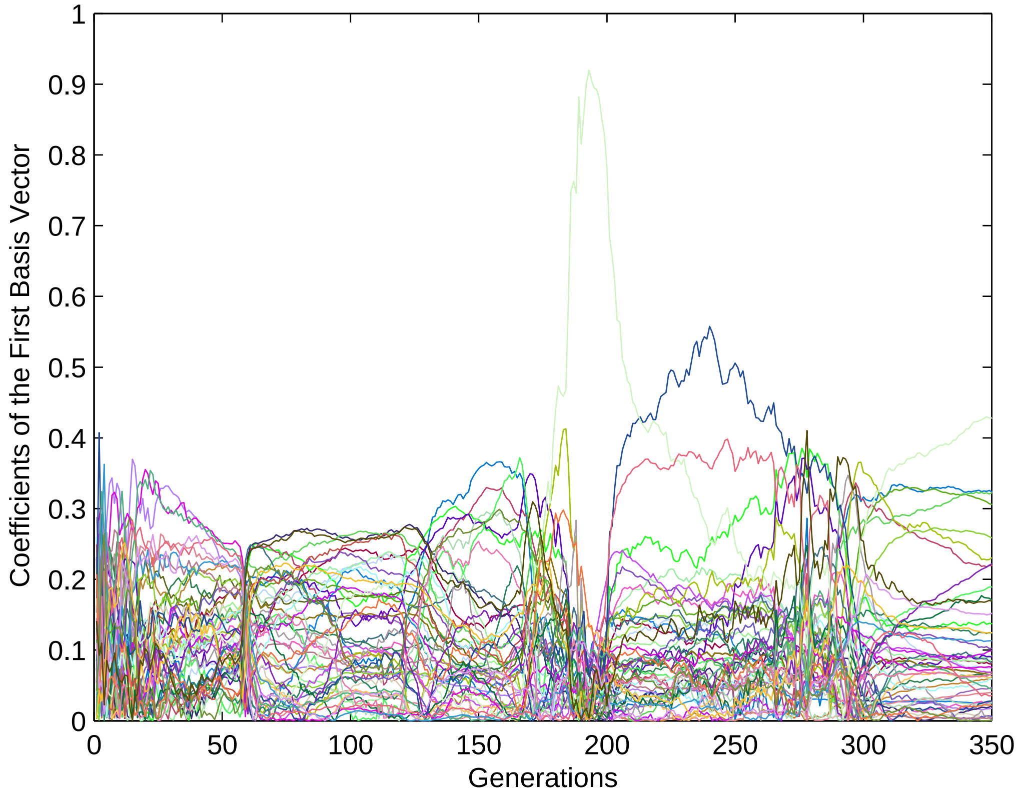


(a)

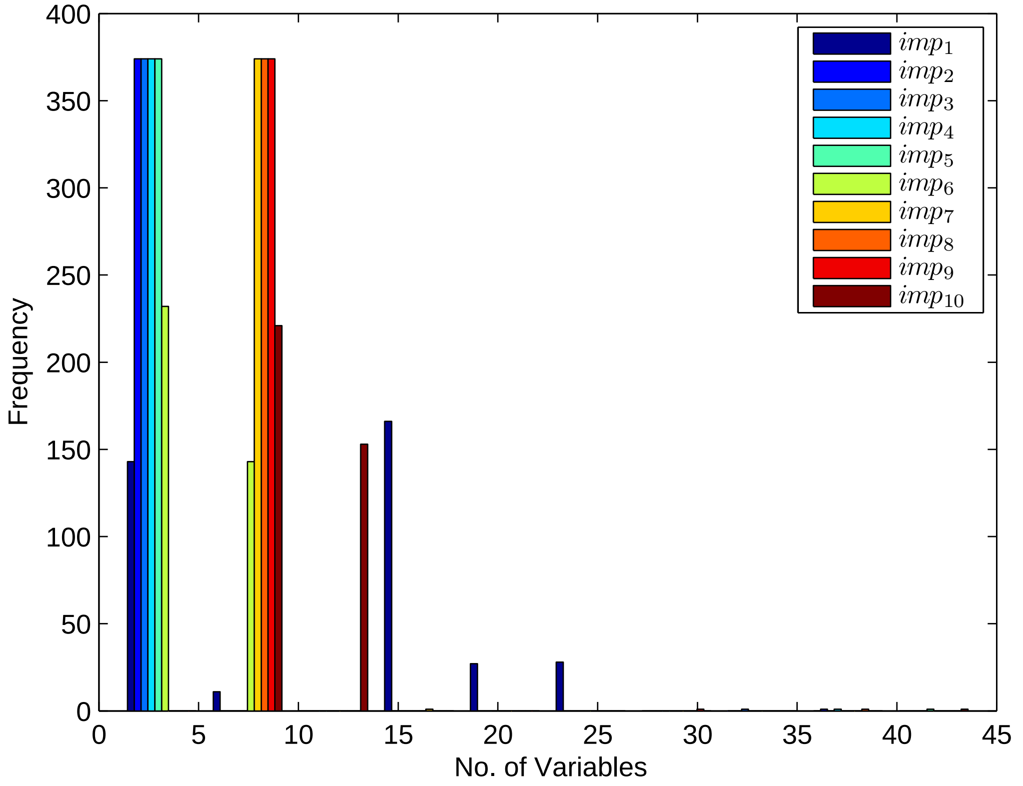


(b)

Figure 5.17: Results are based over 375 generations of EMNA$_{global}$ for the 50D CiaS Problem, (5.17a) Correlation between Variables at the 375th generation, (5.17b) Eigenvalue Analysis.



(a)



(b)

Figure 5.18: Results are based over 375 generations of $\text{EMNA}_{\text{global}}$ for the 50D CiaS problem, (5.18a) Key Structural Variables for the First Eigenvalue, (5.18b) 10 Important Variables to the 50D CiaS Problem captured by $\text{EMNA}_{\text{global}}$.

ure 5.16a. The correlation between variables changes over generations. Figure 5.16b shows that the correlations between all variables at the 180th generation are relatively small. However, 5.17a shows that this changes considerably by the 375th generation. The results of the analysis of key structural variables captured by the model are shown in Figure 5.17b. It shows that the first eigenvalue dominates the other eigenvalues in most of the generations. Drawing the vector coefficients of the first eigenvector into the Figure 5.18a does not provide any significant information of the key structural variables. Hence, it can be said that, EDA has no specific key structural variables in this case.

Finally, for identifying important variables from among the 50 variables in this case, only the first 10 most important variables are shown in Figure 5.18b. It shows that, the first most important variables are distributed among x_2 and x_{16}. The rest of the order is x_1,x_3,x_4,x_5,x_6,x_7,x_8 and x_9.

The analysis shows that the implementation of 10D CiaS in an EDA is not a category 5 type problem. 50D CiaS is showing very negligible correlation between variables, whereas the correlation analysis by the EDA model shows very strong correlation between variables. In the EDA for this problem, $\lambda_1 \gg \lambda_2$ whereas during problem analysis, $\lambda_1 \approx \lambda_2$. However, both analyses do not show any specific key structural variables.

There are no specific important variables found from the problem analysis, whereas from model analysis, it shows an order of important variables.

From this analysis, it can be predicted that a Gaussian EDA algorithm having greater amount of covariance parameter modeling will work better.

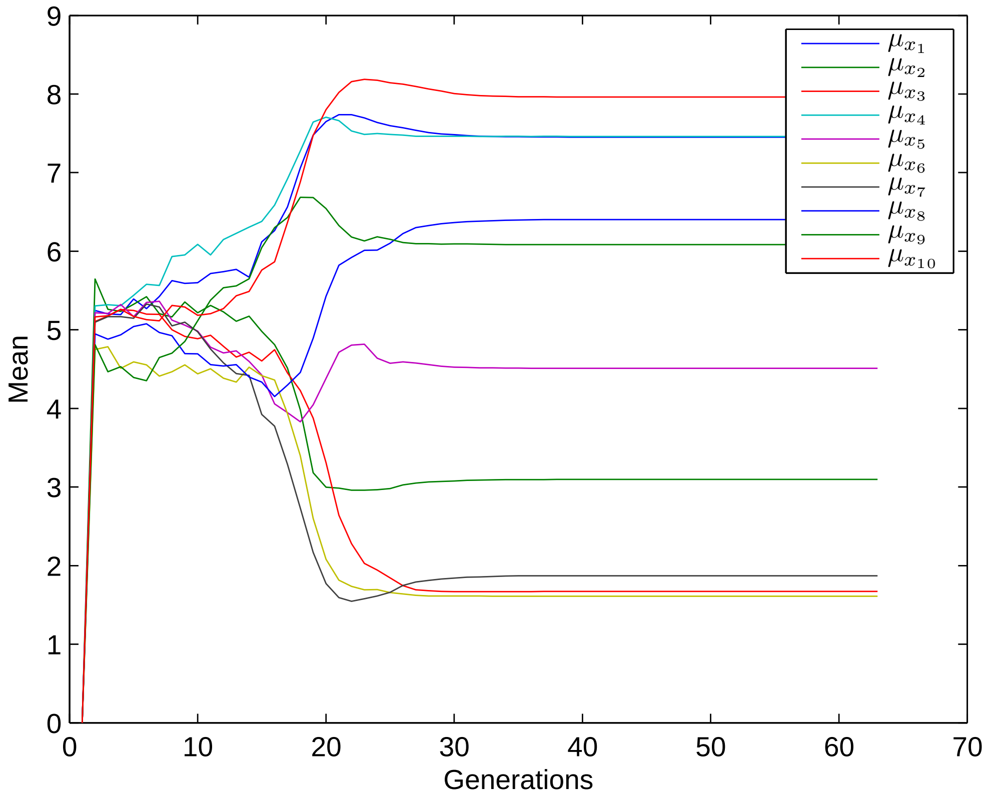
5.6.3 EDAs implemented on 5 Facilities in a 51-Customer Location Allocation Problem

The 51-customer problem on 5 facilities is a category 5 type problem. EMNA$_{\text{global}}$ was applied to the 51-customer problem on 5 facilities (i.e. 10D), with a population size of 2000, and a selection ratio of 0.2 running over 200 generations. The algorithm stops when the absolute difference between the mean of the two consecutive generations is less than or equal to 1e-06, or it evaluates 200 generations. The threshold for measuring the key structural variables is $\frac{1}{\sqrt{10}} \approx 0.3$. The global optimum value for this problem is 72.2369 [26].

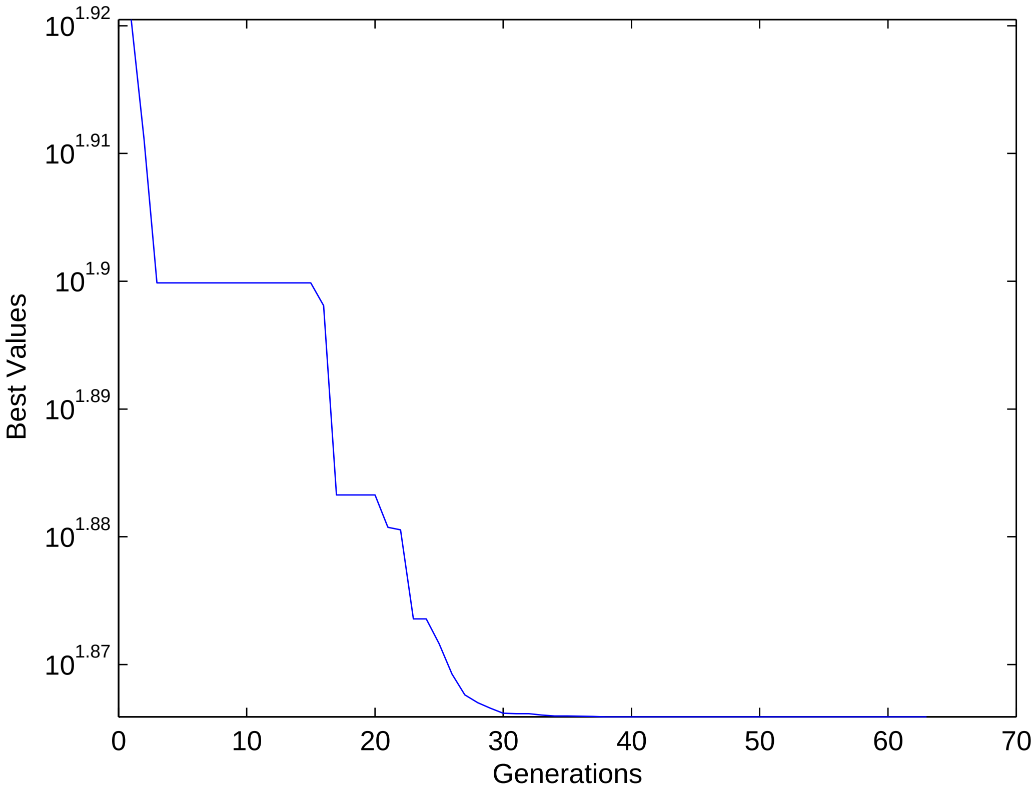
In this case, the algorithm converges at the 63rd generation. The mean values and the best values of the EDA reached so far are shown in the Figures 5.19a and 5.19b respectively. The model mean values move rapidly until about the 40th generation and then makes little progress, which suggests that the model is converged. Figure 5.19b illustrates about the best so far curve, which does not show any progress after 40th generation. However, the result is based on a typical run; the average best value over 10 runs is 74.9590 and the standard deviation is 2.7544. In this experiment the model reaches near to the global optimum with a value of 72.2470 (Figure 5.19b).

Firstly, the correlation captured by the model in this problem shows that, in the early generations, no persistent trends or patterns of variable dependency exist. At the 63rd generation, some structure appears (Figure 5.20b), which indicates that the EDA model has a lack of correlation between most of the variables.

Figures 5.19a and 5.19b indicate that around the 15th generation, the model moves rapidly and a

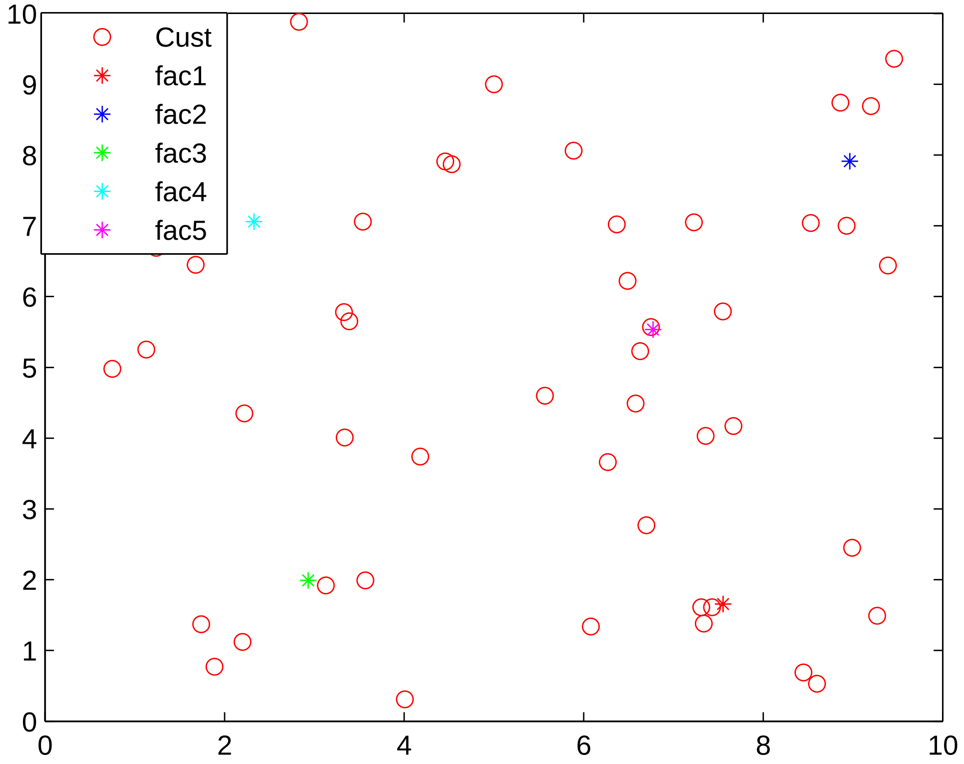


(a)

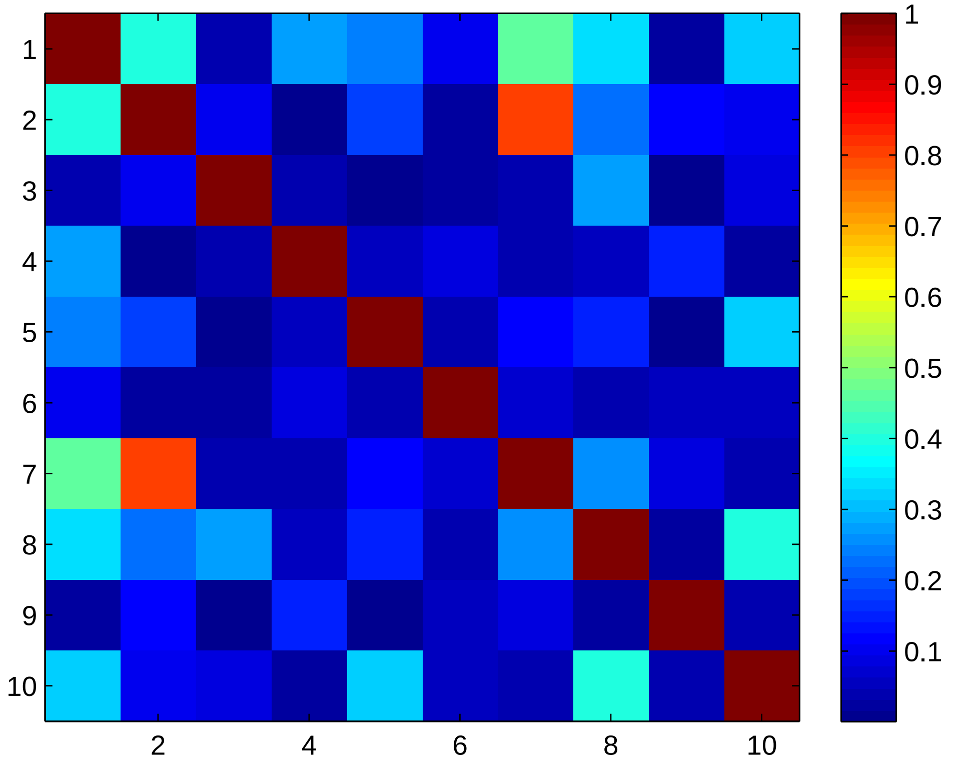


(b)

Figure 5.19: Results are based over 63 generations of EMNA$_{\text{global}}$ for the 10D Location Allocation Problem, (5.19a) Mean of EMNA$_{\text{global}}$, (5.19b) Best Values reached so far.

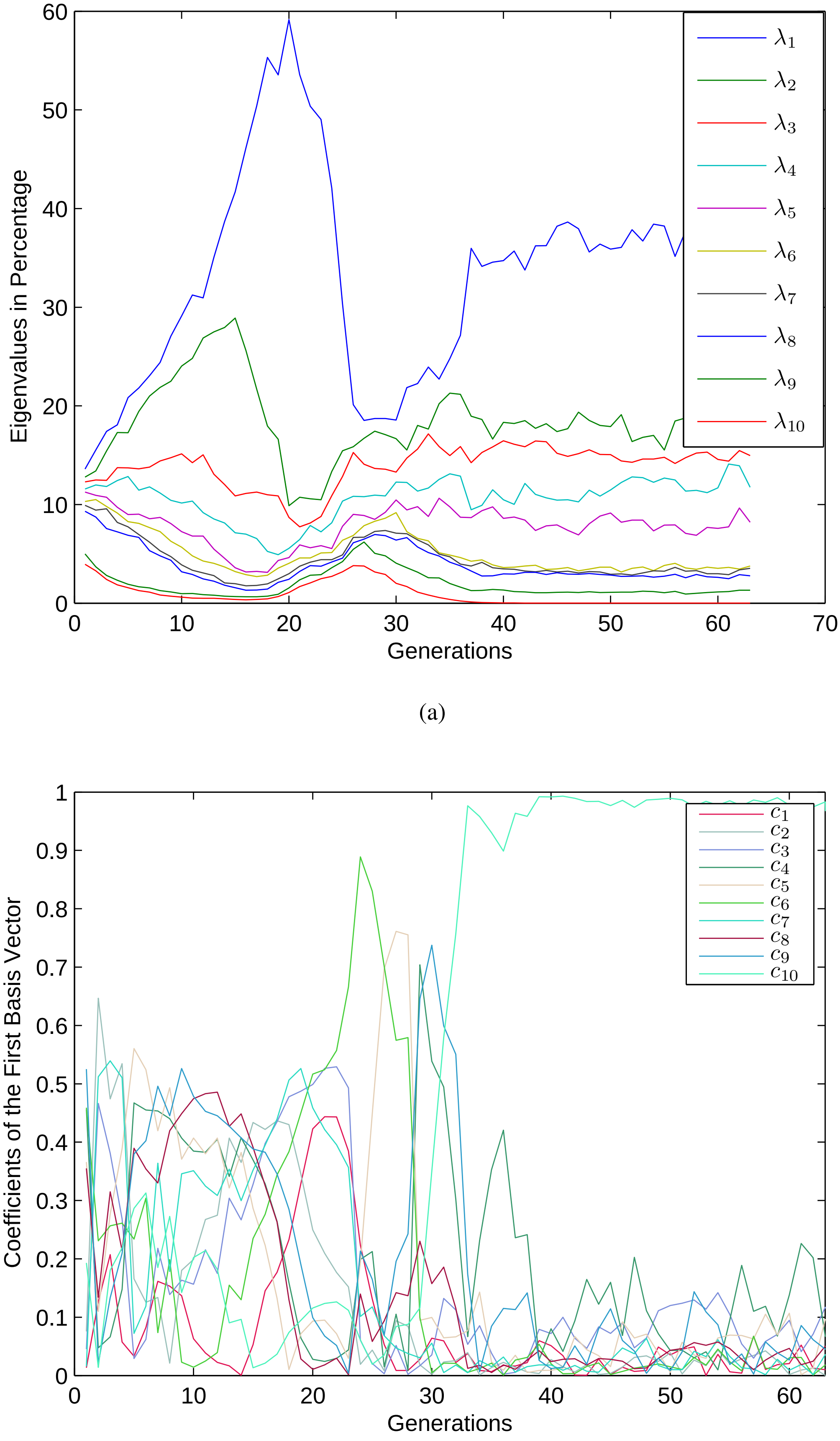


(a)



(b)

Figure 5.20: Results are based over 63 generations of $\text{EMNA}_{\text{global}}$ for the 10D Location Allocation Problem, (5.20a) Plotting of 5 Facilities in 51 Customer space at the 63rd generation, (5.20b) Correlation between Variables at the 63rd generation.



(a)

(b)

Figure 5.21: Results are based over 63 generations of the EMNA$_{\text{global}}$ for the 10D Location Allocation Problem, (5.21a) Eigenvalue Analysis, (5.21b) Key Structural Variables for the First Eigenvalue.

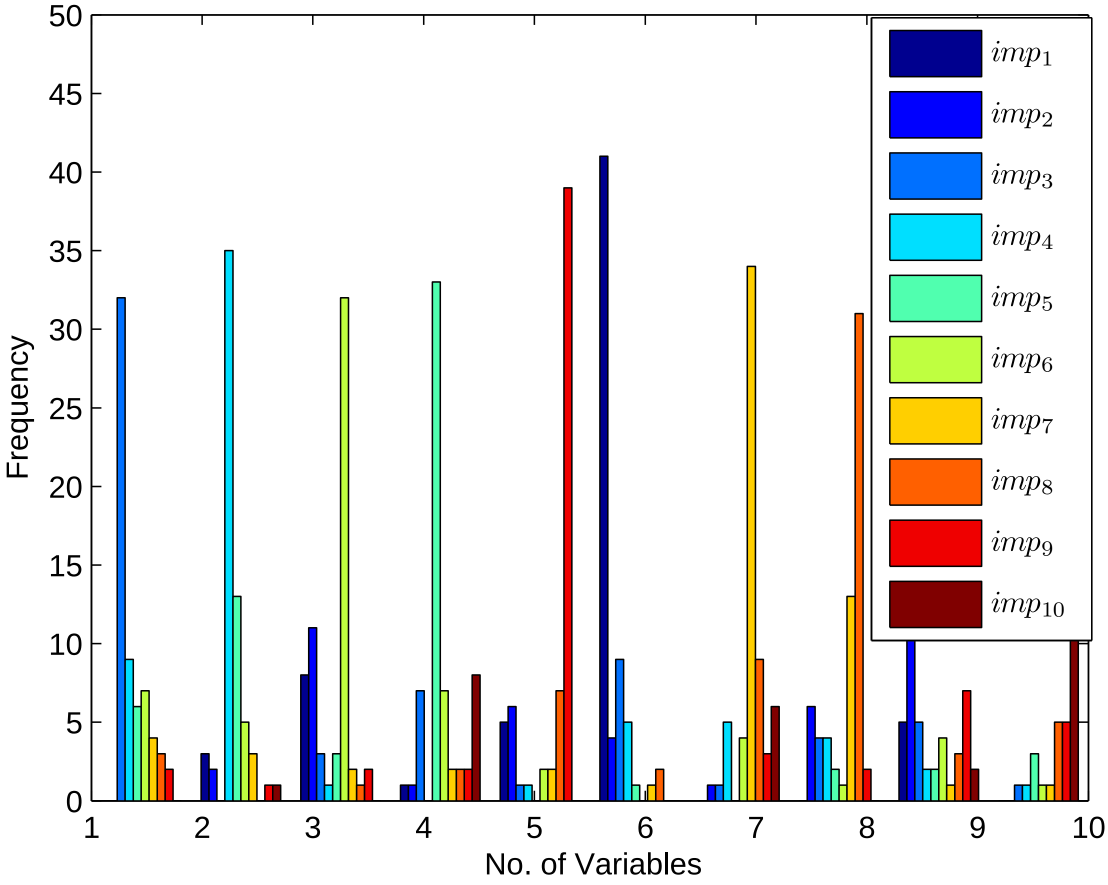


Figure 5.22: Important Variables to the 10D Location Allocation Problem captured by EMNA$_{\text{global}}$.

much better solution is found. Furthermore, Figure 5.21a shows there is some spread in eigenvalues, but the first one is somewhat larger after the 15th generation, indicating that the problem is somewhat elliptical in shape. Overall, there is a lot of rapid change of coefficient values (Figure 5.21b); however, in the first 38 generations, predicting the key structural variables in the search model is difficult, but after that, c_2 and c_7 have the higher values than the rest. So the first eigenvector points strongly in the directions of the x_2 and x_7 axis, which is the x and y coordinate of the 2nd facility in the problem (the blue dot in the Figure 5.20a represents the placing of second facility in the 51-Customer space).

Finally, Figure 5.22 shows the important variables in the form of a histogram. It shows that, for more than 40 generations, x_6 is the most important variable, and x_1 is the 3rd important variable. For more than 45 times, x_{10} is the least important variable. According to the figure, the order of importance is as follows: $x_6, x_9, x_1, x_2, x_4, x_3, x_7, x_8, x_5, x_{10}$.

From the analysis, it has been deduced that the model does not capture the properties of category 5. The figures obtained from the EDA model and the problem are different from each other. The problem variables form a pattern of correlation between them, where as the model does not capture any pattern of correlation during its run nor does it reveal any strong correlation between the variables.

During the analysis of the key structural variables of the model, λ_1 is the dominant eigenvalue, whereas during the problem analysis $\lambda_1 \approx \lambda_2$. The EDA model is able to analyze the key structural variables, but in this case no specific variables are key structural variables throughout the run (although, in this case the key structural variables is x_{10} for the last couple of runs) whereas variable analysis on problems cannot give such information.

EDA analysis gives an order of important variables (for this run the three most important variables are x_6, x_9 and x_1, but the problem analysis did not show any type of important variables.

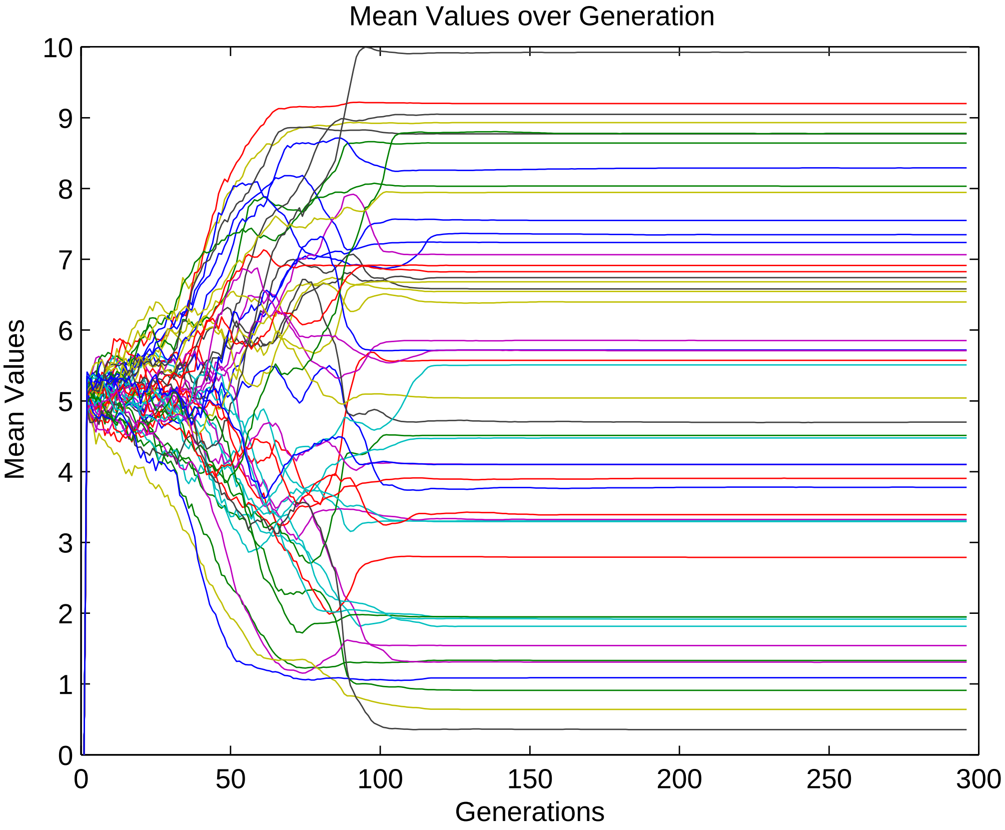
Hence, the conclusion of this analysis shows that the EDA has a weak correlation among most of the variables, has no specific key structural variables and has an order of important variables. Therefore the 10D location-allocation problem can be solved using any Gaussian EDA which has the covariance model control parameter.

5.6.4 EDAs implemented on 25 Facilities in a 51-Customer Location Allocation Problem

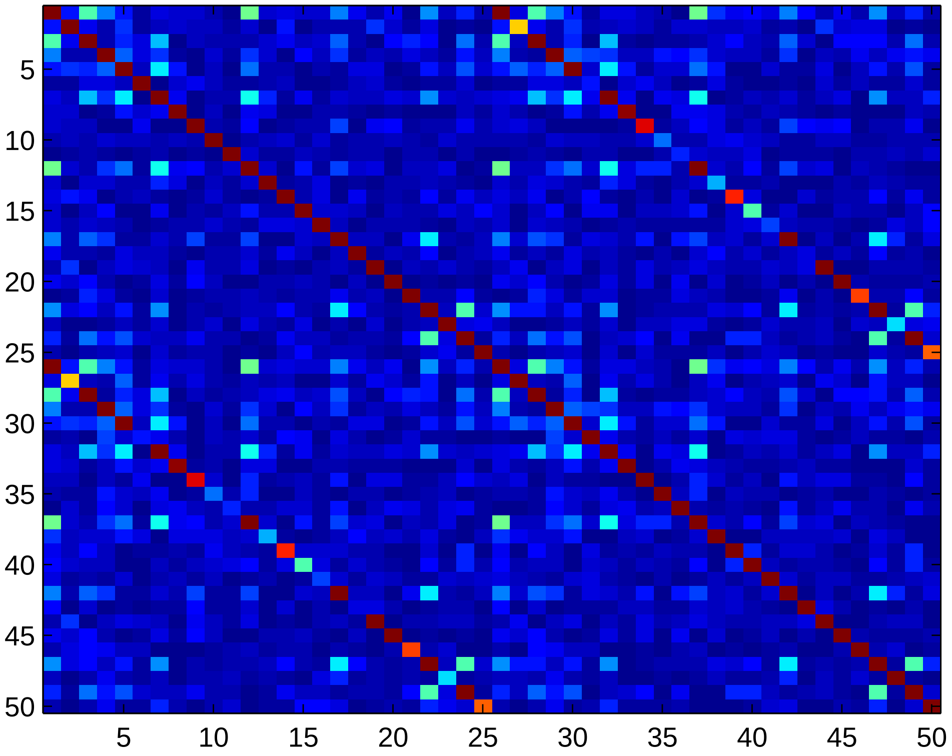
50D Location allocation is a category 5 type problem (refer Table 4.1). EMNA$_{\text{global}}$ was implemented on 25 facilities of a 51-customer location allocation problem, with 5000 population having 0.2 as its selection ratio for 500 generations. Since, this is a 50D, the threshold for identifying key structural variables is $\frac{1}{\sqrt{50}}$=0.1414. The algorithm stopped when the absolute difference between the mean of two successive generation is 1e-06 or it attained all the 500 generations.

The algorithm stopped at 296 generations. The mean of the selected population over 296 generations are plotted in Figure 5.23a. It shows that the model is converged. The correlation between variables for this problem is shown in Figure 5.23b, and displays the correlation between x and y pairs of each facility, while the problem is implemented on EMNA$_{\text{global}}$.

Furthermore, the results of the eigenanalysis are plotted in Figure 5.24a and 5.24b. Figure 5.24a shows that the λ_1 relatively captures more variance than the rest of the λ's. Hence the first eigenvalue

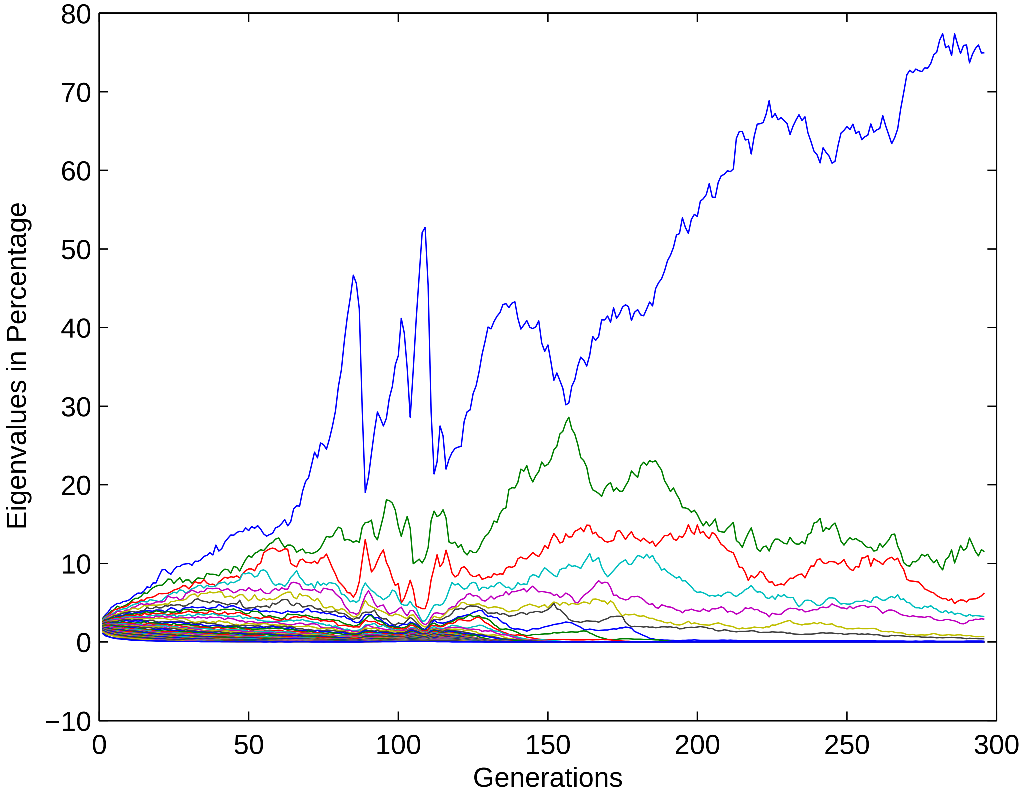


(a)

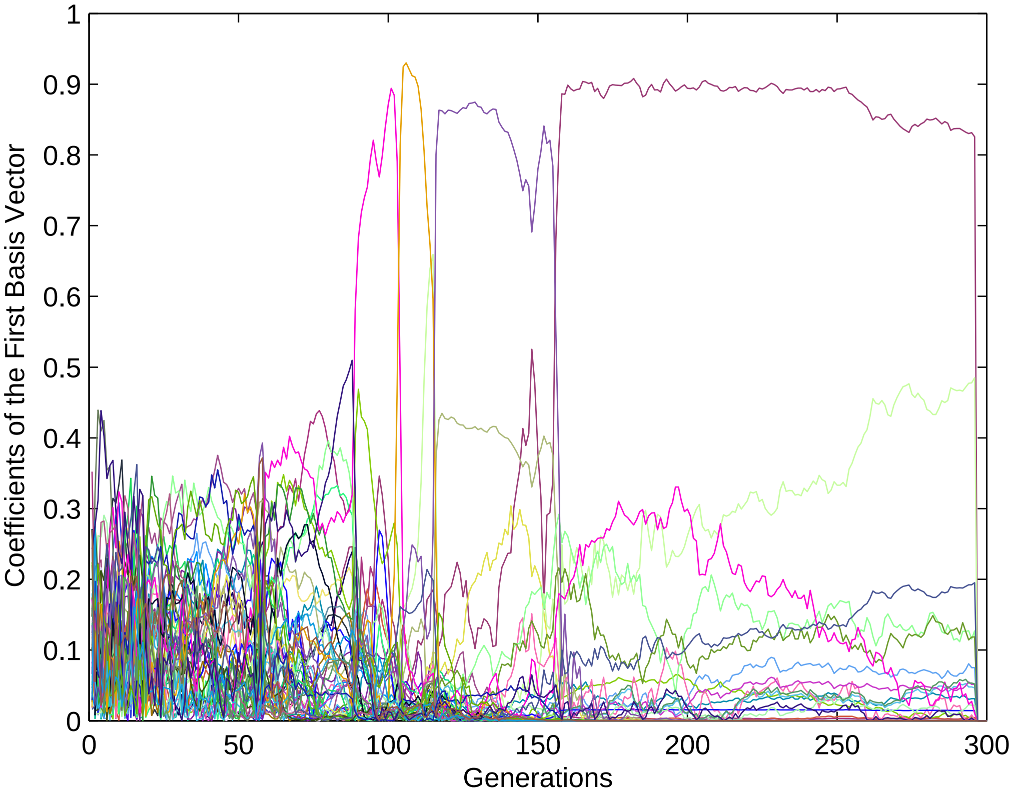


(b)

Figure 5.23: Results are based over 296 generations of EMNA$_{\text{global}}$ for the 50D Location Allocation Problem, (5.23a) Mean Values, (5.23b) Correlation between Variables.



(a)



(b)

Figure 5.24: Results are based over 296 generations of EMNA$_{\text{global}}$ for the 50D Location Allocation Problem, (5.24a) Eigenvalue Analysis, (5.24b) Key Structural Variables for the First Eigenvalue.

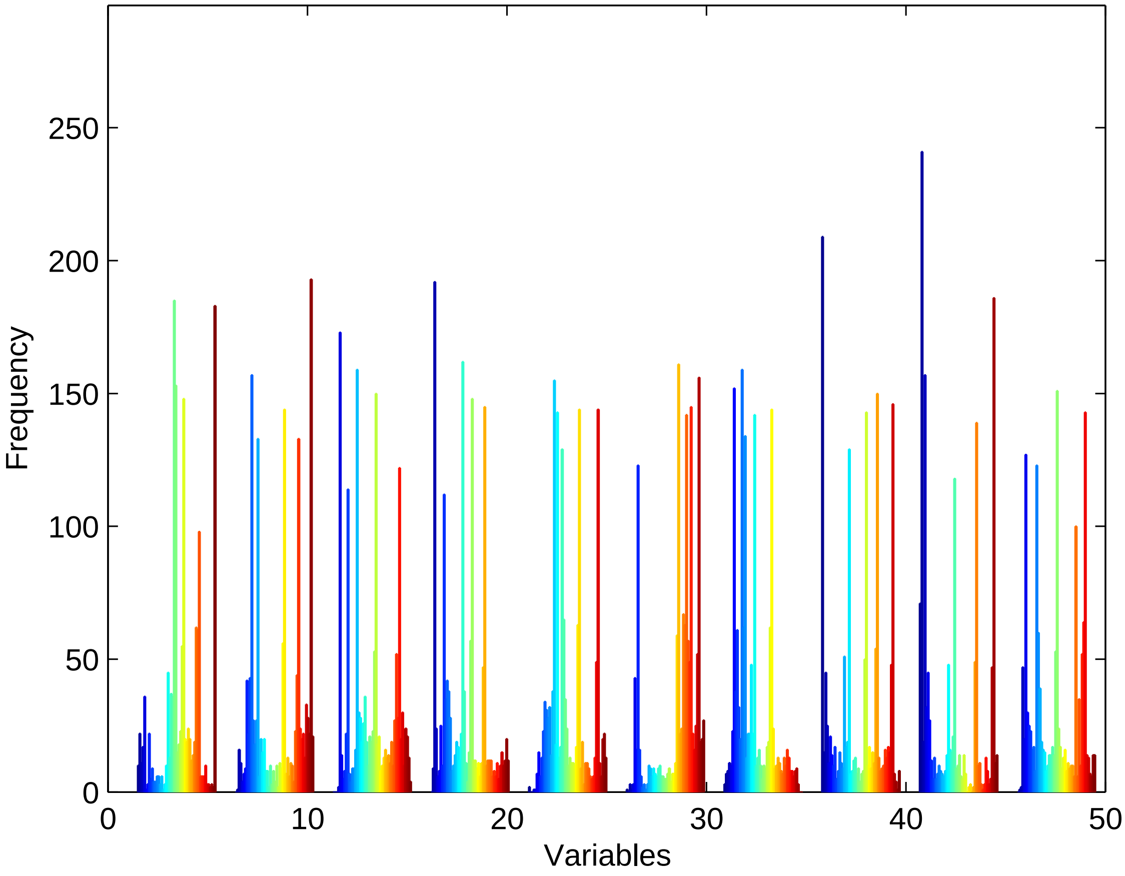


Figure 5.25: Important Variables to the 50D Location Allocation Problem captured by the EMNA$_{\text{global}}$.

gives an idea about the key structural variables to the model. Figure 5.24b shows that some of the variables are above threshold values (i.e., 0.1414) in most of the generations, indicating the existence of some specific key structural variables captured by the model.

Lastly, Figure 5.25 identifies the important variable to the objective function captured by the EDA model. Figure 5.25 shows that the blue bars are the most important variables and the red bars are the least important to the model. It shows that x_{18}, x_{37} and x_{42} are important variables, since these variables are important in most of the generations.

Comparing the analysis found by the model in this problem and the analysis found from the problem, it is clear that, the correlation between variables captured by the EDA model and the problem variables are exactly same. Regarding the eigenanalysis, the first eigenvalue gives information about the key structural variables, whereas the first eigenvalue found from the problem analysis is not suggesting any key structural variables.

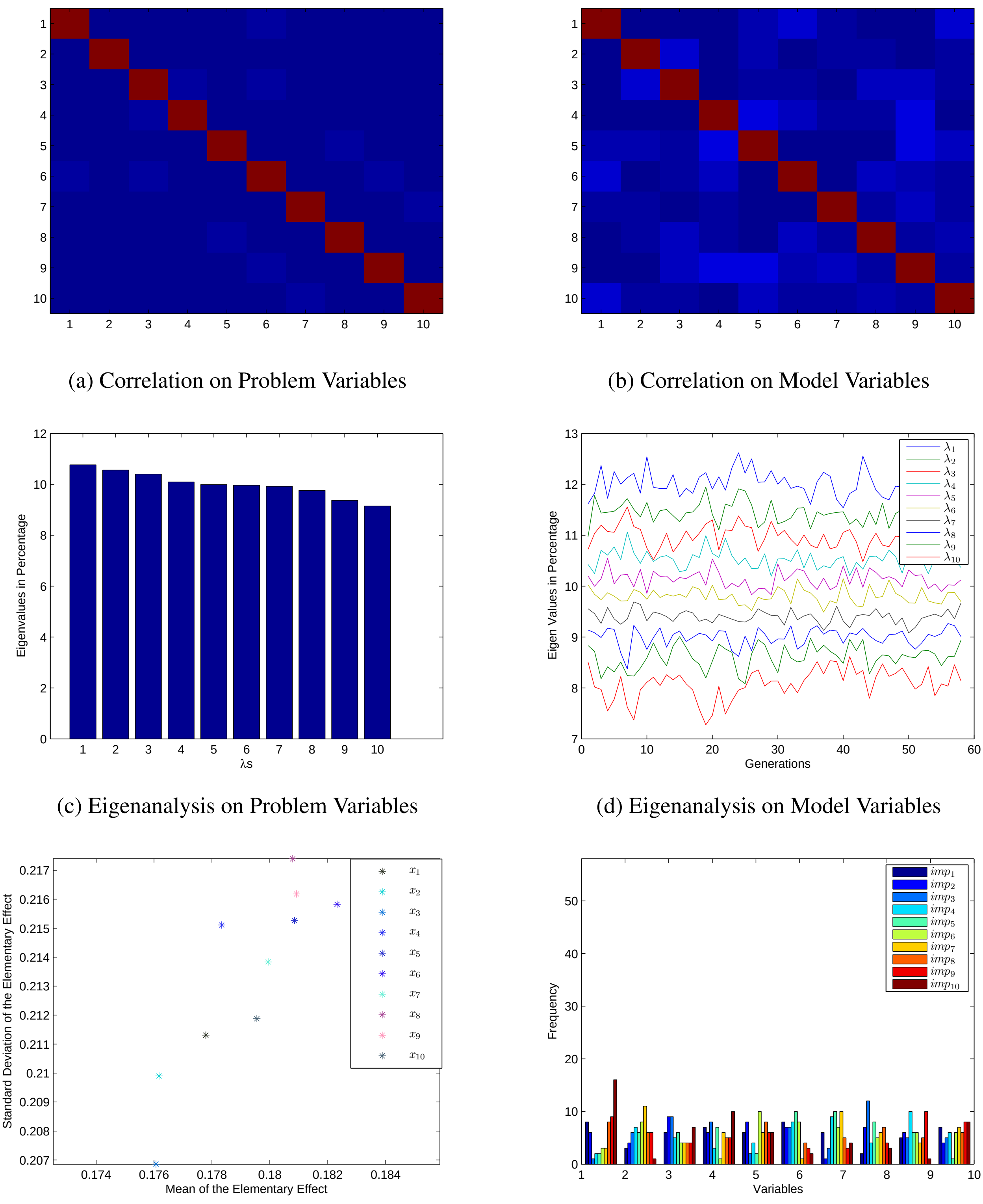
Finally, there are some specific variables which are important in most of the generations, but the problem analysis did not find any specific key structural variables.

Therefore, the EDA model does not capture exact properties of problem variables, Hence, any Gaussian based EDA with covariance model control parameter may work better in this problem.

5.7 Summary of the Analysis and Comparison

This chapter has extended the framework proposed in Chapter 3 to analyze data produced by EDAs during search (i.e. the selected population and corresponding objective function values). EMNA$_{\text{global}}$ was applied to the set of test problems analyzed in Chapter 4 and the data generated from the algorithm experiments was analyzed. From the results, it can be concluded that for some problems, the properties of problem variables appear similar for both problem and algorithm data analysis, and in some cases, the properties are different. In some cases, the EDA model variable properties do not match well the problem variable properties.

The intention here is to show in principle how the analysis might be used as a component in addressing the algorithm selection problem (either manually or as part of an automated meta-algorithm). The results in this Chapter and the following Chapter also provide some insight into whether or not the suggested algorithm is a good choice. However, the primary focus of this thesis is in the development of a general-purpose analysis framework rather than attempting to demonstrate state-of-the-art performance for specific problems.



(a) Correlation on Problem Variables

(b) Correlation on Model Variables

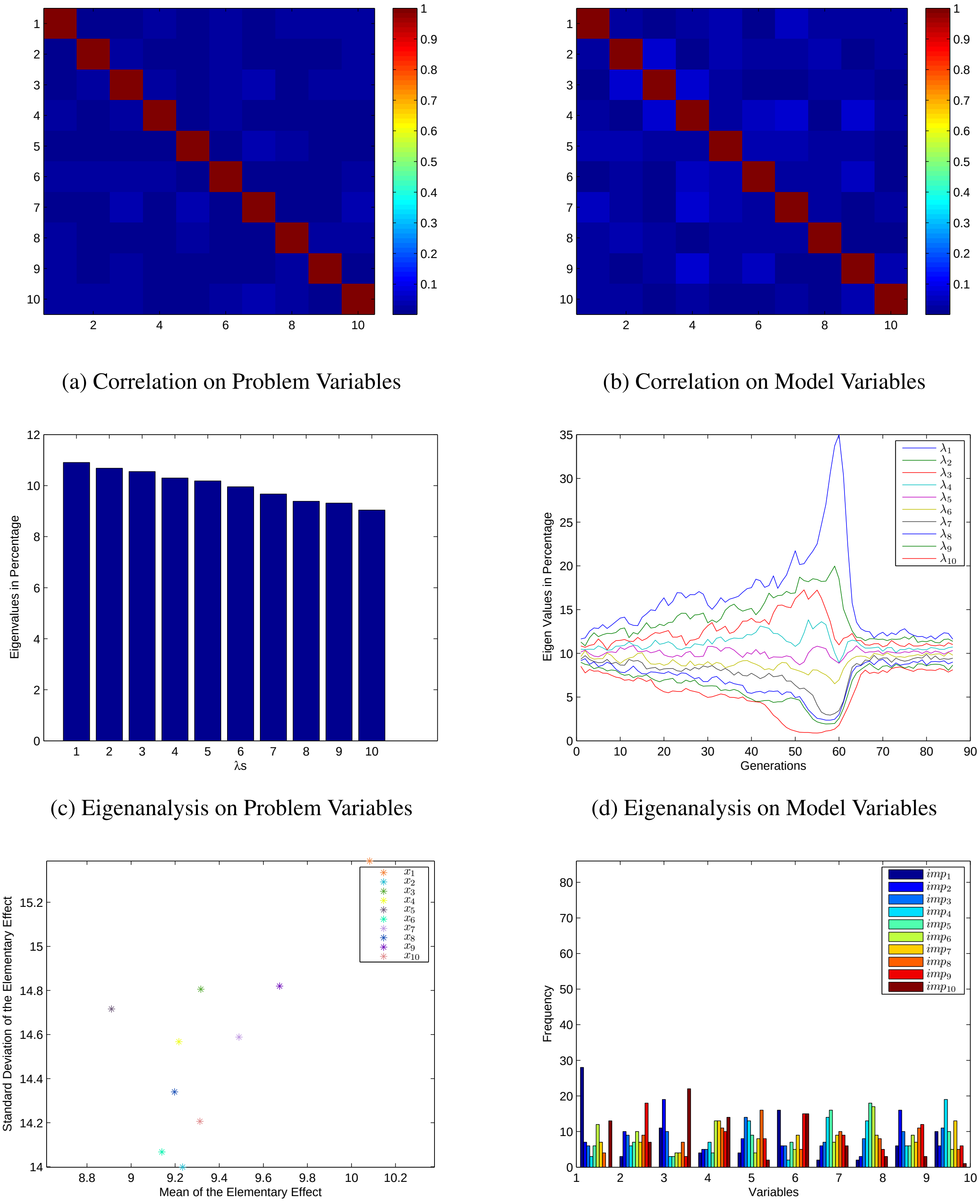
(c) Eigenanalysis on Problem Variables

(d) Eigenanalysis on Model Variables

(e) Morris Method on Problem Variables

(f) Morris Method on Model Variables

Figure 5.26: Comparision of the properties of Problem Variables and the Model Variables in a 10D Ackley Function



(a) Correlation on Problem Variables

(b) Correlation on Model Variables

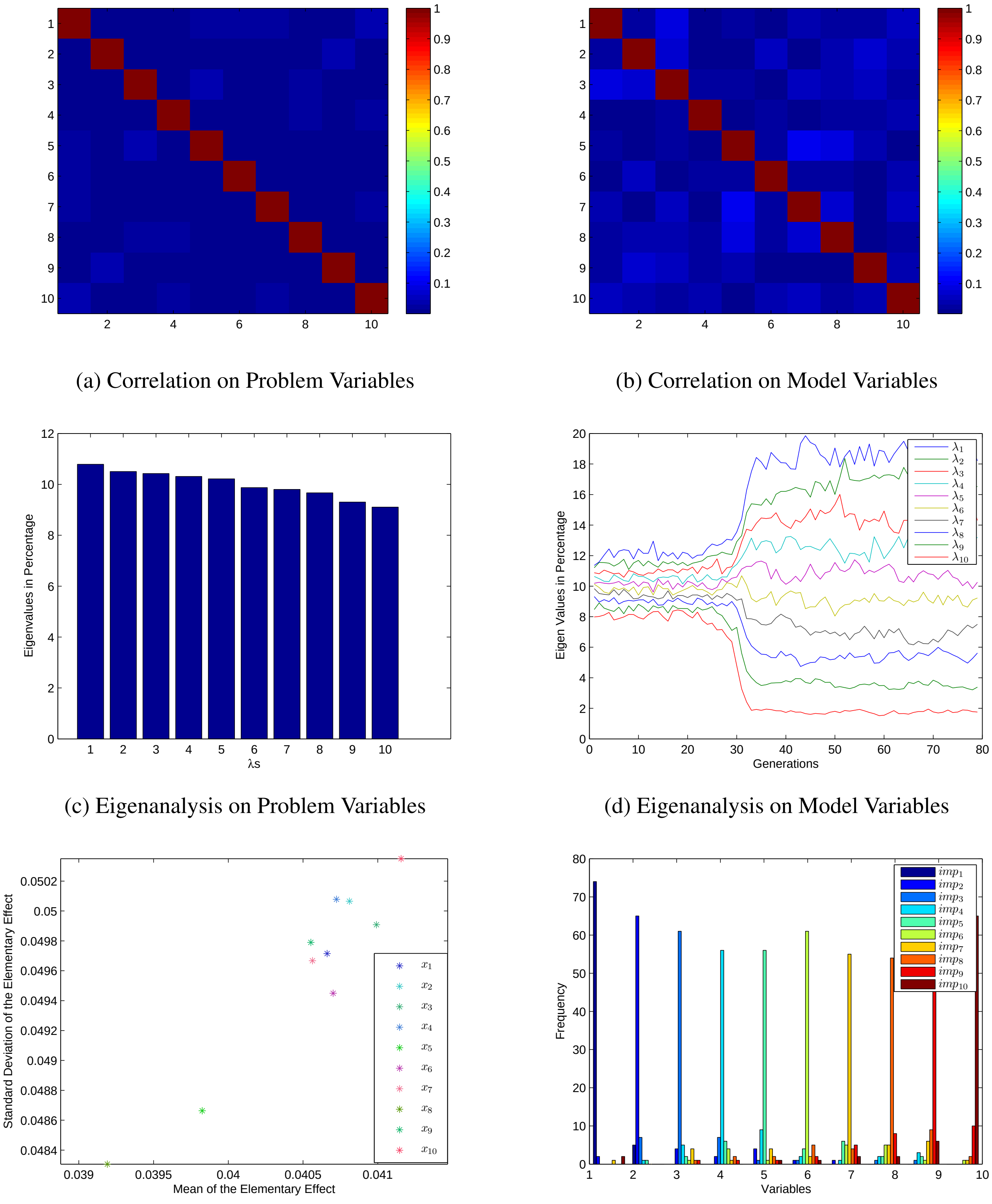
(c) Eigenanalysis on Problem Variables

(d) Eigenanalysis on Model Variables

(e) Morris Method on Problem Variables

(f) Morris Method on Model Variables

Figure 5.27: Comparision of the properties of Problem Variables and the Model Variables in a 10D Rastrigin Function.



(a) Correlation on Problem Variables

(b) Correlation on Model Variables

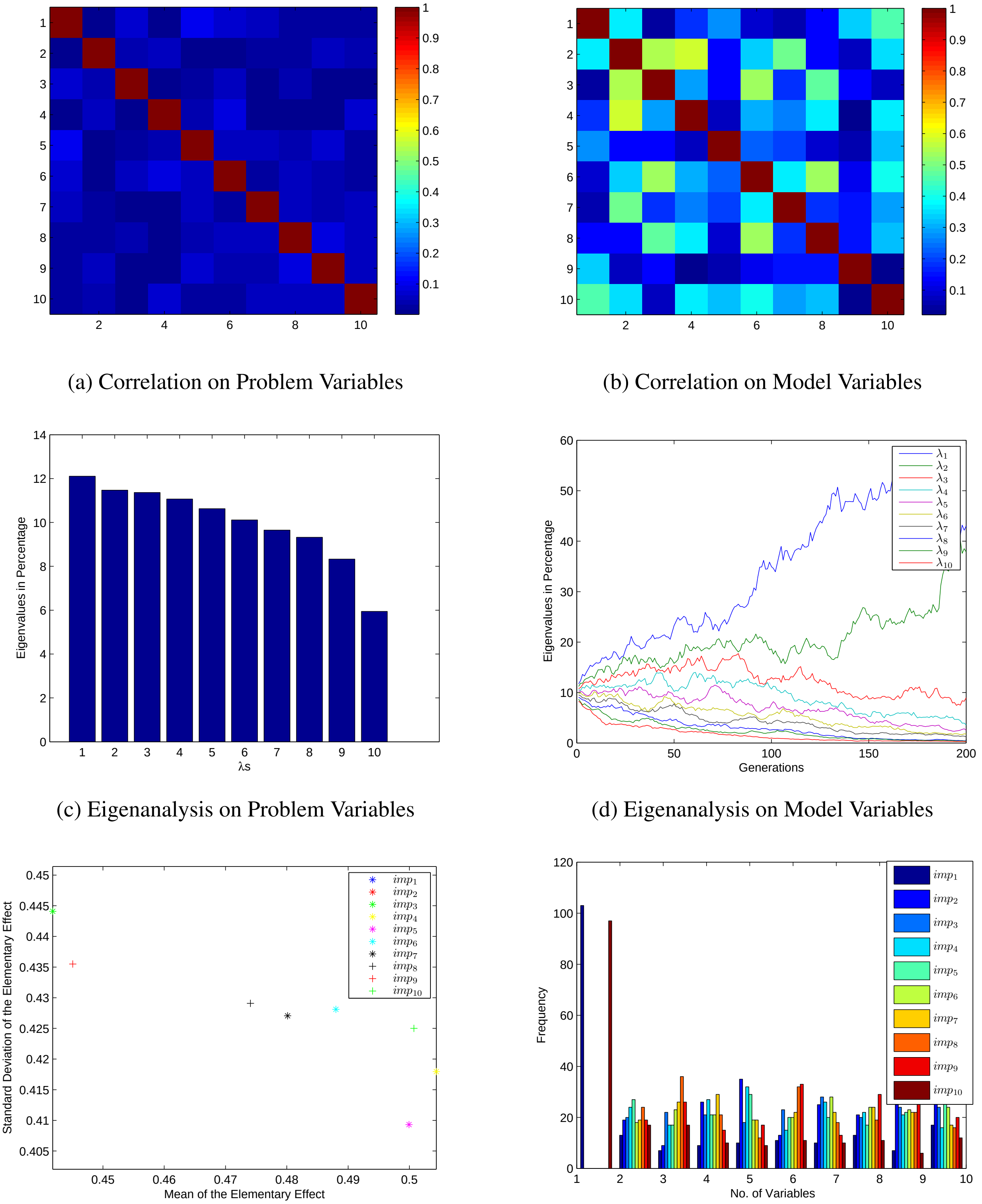
(c) Eigenanalysis on Problem Variables

(d) Eigenanalysis on Model Variables

(e) Morris Method on Problem Variables

(f) Morris Method on Model Variables

Figure 5.28: Comparision of the properties of Problem Variables and the Model Variables in a 10D Griewangk Function.



(a) Correlation on Problem Variables

(b) Correlation on Model Variables

(c) Eigenanalysis on Problem Variables

(d) Eigenanalysis on Model Variables

(e) Morris Method on Problem Variables

(f) Morris Method on Model Variables

Figure 5.29: Comparision of the properties of Problem Variables and the Model Variables in a 10D F9 Function.

5.8 Further Analysis

5.8.1 Further Analysis on Problem and EDA Model Variables in 10D Ackley Function, 10D Rastrigin Function, 10D Griewangk Function and F9 Function

In this section, analysis of problem variables and EDA variables for the 10D Ackley, Rastrigin, Griewangk and F9 functions, showing a graphical representation of the results. All the parameters for the problem and EDA analysis are set according to Chapters 4 and 5. Figures 5.26, 5.27, 5.28 and 5.29 show the graphical representation of the analysis on problems and EDAs of 10D Ackley, 10D Rastrigin, 10D Griewangk function and 10D F9 function respectively. It can be seen from the comparison tables that 10D Ackley, Rastrigin and Griewangk problems are approximately similar to Sphere function. But the notable is the Griewangk function, while implemented on EDA. The Problem analysis shows that there are no specific important variables (refer Figure 5.28e) but the EDA implementation on this problem (refer Figure 5.28f) shows that it has specific important variables. It shows that EDA is not able to capture the right information of important variables for the Griewangk function.

F9 function, (whose global optimum is not at origin) is shown in Figure 5.29. Figure shows the problem is similar to Sphere function, but while implementing the problem in an EDA, the variable properties are quite different. Figure (5.29b) shows a correlation between variables. In addition to this, figure (5.29d), indicates that the first eigenvalue has the largest variance relatively compared with other eigenvalues. hence the first eigenvalue gives information about the "Key Structural variable" captured by the EDA. Again, in figure (5.29f) shows that the first and second variables are more important than any other variable.

These results are included here for completeness.

5.8.2 Sensitivity of the Analysis on different values of τ

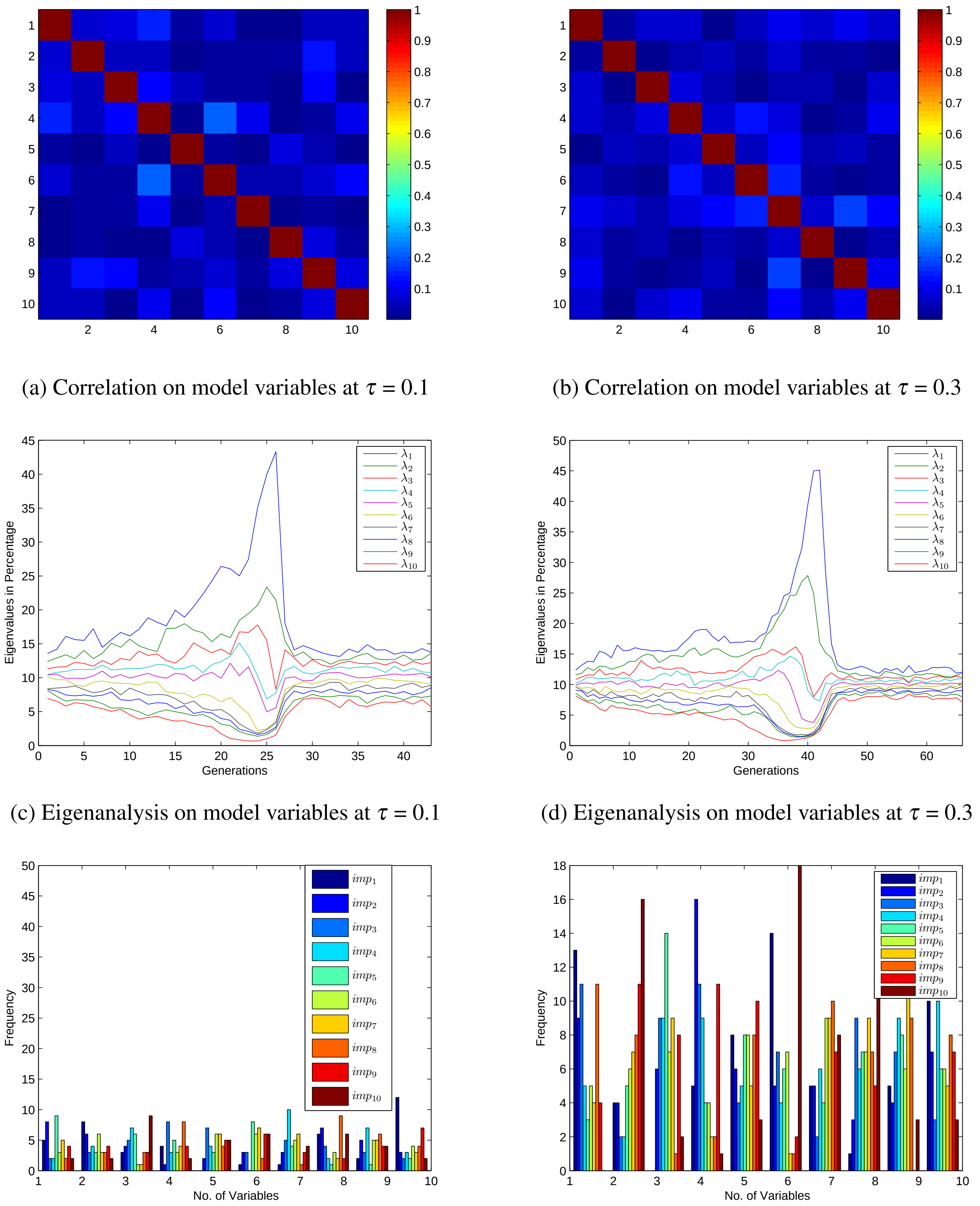
To examine EDA variables with different τ values, 10D Rastrigin function has been taken into consideration. How the nature of model variable changes with different value of selection parameter has been done here. EMNA$_{\text{global}}$ has been implemented on 10D Rastrigin function, with 2000 population for 200 generations with a number of different values of τ = 0.1, 0.3, 0.5 and 0.9. The algorithm will stop when the absolute difference between the maximum mean of the two consecutive generations is less than or equal to 1e-06 or it attains its maximum number of generations supplied (here 200 generations).

Figures 5.30a, 5.30b, 5.31a and 5.31b shows the correlation between variables when τ = .1, .3, .5 and .9 respectively. It shows that there is no significant correlation found between the variables in all these cases.

Again, Figures 5.30c, 5.30d, 5.31c and 5.31d shows the eigenanalysis when τ = .1, .3, .5 and .9 respectively. It shows that all the figures follow the same pattern of distribution of eigenvalues, but when τ=0.1, the first eigenvalue gains maximum variance at 25th generation, where as when

τ=0.3, the first eigenvalue gains the maximum variance at 40th generation, and when τ=0.5, the first eigenvalue gains the maximum variance at 60th generation, but when τ=0.9, the first eigenvalue gets the maximum variance at 200th generation.

Further, Figures 5.30e, 5.30f, 5.31e and 5.31f shows the important model variables when τ = .1, .3, .5 and .9 respectively. It shows that, all the variables are more or less likely to be important through out the generation.



(a) Correlation on model variables at $\tau = 0.1$

(b) Correlation on model variables at $\tau = 0.3$

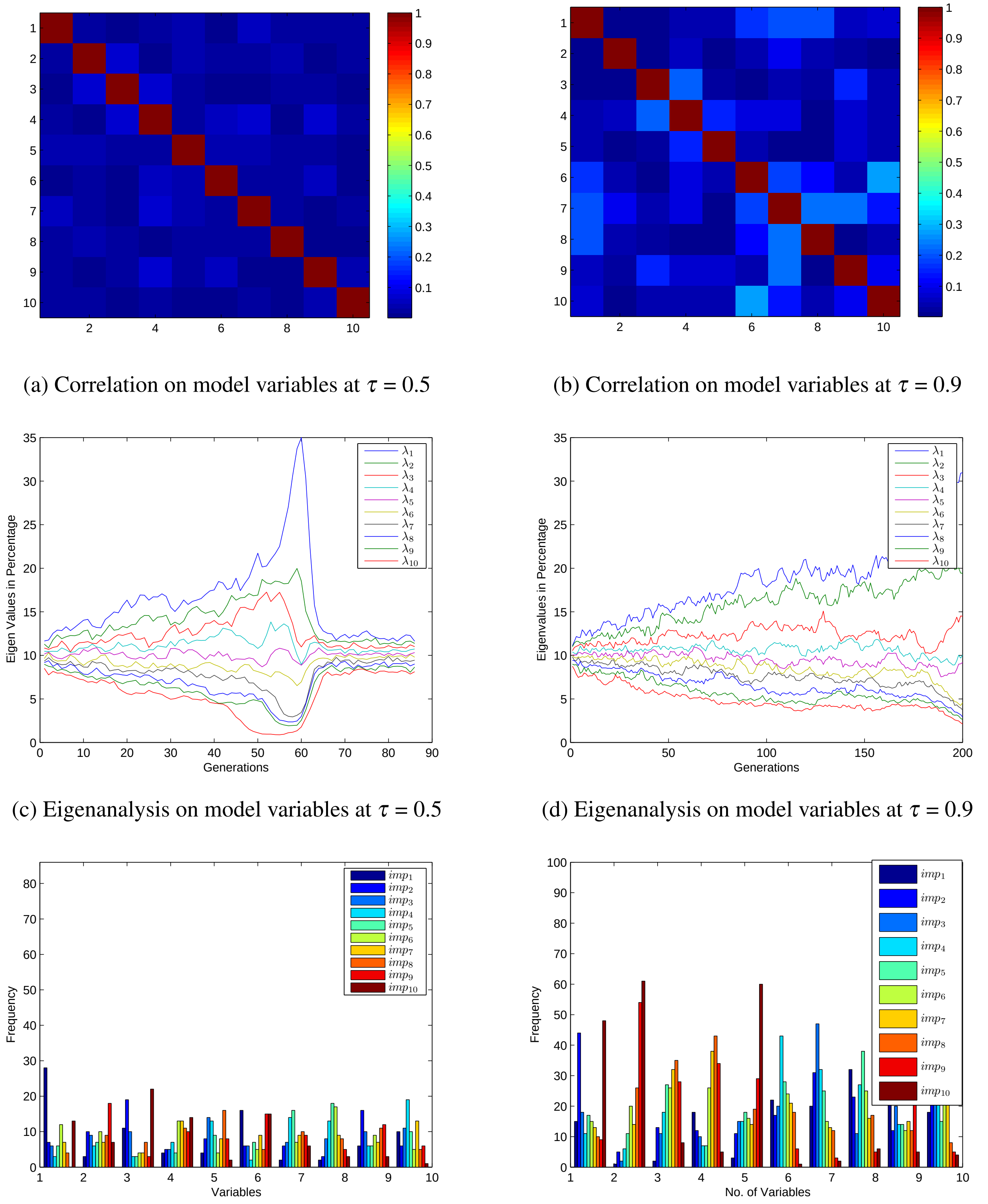
(c) Eigenanalysis on model variables at $\tau = 0.1$

(d) Eigenanalysis on model variables at $\tau = 0.3$

(e) Morris Method on model variables at $\tau = 0.1$

(f) Morris Method on model variables at $\tau = 0.3$

Figure 5.30: Comparision of the properties of EDA model variables in a 10D Rastrigin Function at τ=0.1 and 0.3.



(a) Correlation on model variables at $\tau = 0.5$

(b) Correlation on model variables at $\tau = 0.9$

(c) Eigenanalysis on model variables at $\tau = 0.5$

(d) Eigenanalysis on model variables at $\tau = 0.9$

(e) Morris Method on model variables at $\tau = 0.5$

(f) Morris Method on model variables at $\tau = 0.9$

Figure 5.31: Comparision of the properties of EDA model variables in a 10D Rastrigin Function at τ=0.5 and 0.9.

Chapter 6

Screening Estimation of Distribution Algorithms

6.1 Overview

In this chapter, a modified Gaussian-based continuous EDA is proposed, called Screening EDA (sEDA). It provides a mechanism to control the percentage of covariance parameters estimated within the Gaussian model. To achieve this, a simple variable screening technique which has already been discussed in Chapter 3 called the Morris method has been used. This technique has been used in Chapter 4 and Chapter 5 for analyzing important variables in problems and EDAs respectively. Now, in this chapter, the screening technique is incorporated into the EDA itself to try and improve the modeling. Compared to EMNA$_{\text{global}}$, the algorithm provides improvement on ill-conditioned covariance matrix and can use a smaller selected population. Experimental results are presented to evaluate and compare the performance of the proposed algorithm to UMDA$_{\text{c}}$ and EMNA$_{\text{global}}$.

Analysis and experimental evaluation of the basic sEDA implementation reveals that it does not scale well to high-dimensional problems because it requires a large number of additional fitness function evaluations per generation. Consequently, a modified version of the sEDA algorithm is also proposed, called sEDA-lite. It requires no additional fitness function evaluations for sensitivity analysis and therefore is applicable to larger problems. Experimental results on a large set of artificial and real-world representative problems evaluate the performance of sEDA-lite and compare it with sEDA and EDA-MCC (EDA framework with Model Complexity Control), a related, recently proposed algorithm [41].

This chapter is structured in the following way. In Section 6.2, a modification to the EDA algorithm is proposed, the Screening EDA (sEDA). The implementation of the sEDA and parameter values is described in detail in Section 6.3. In Section 6.4 experimental results on various artificial problems are presented evaluating the sEDA and comparing it with UMDA$_{\text{c}}$ and EMNA$_{\text{global}}$. Section 6.5 presents some unique behavior of sEDA. The issues arising in sEDA are discussed in Section 6.6. In Section 6.7, a modified Gaussian-based sEDA called sEDA-lite is proposed to address the issues of high dimensional problems in sEDA. In Section 6.8, comparison of the solutions of sEDA-lite with solutions of UMDA$_{\text{c}}$, EEDA (eigenspace EDA) and EDA-MCC on a set of artificial test problems have been done. In addition to this, the solution of sEDA-lite is also compared with UMDA$_c$, EMNA$_{\text{global}}$, sEDA on a couple of real world problems. The conclusion of this chapter is drawn in Section 6.10.

6.2 Introducing Variable Screening into an EDA

Two of the three analysis techniques used in this thesis (dependency modeling and key structural variables) come directly from the Gaussian EDA model (i.e. the covariance matrix). As discussed in Section 2.7, when a full covariance matrix is used such as in EMNA$_{\text{global}}$, numerical issues create practical difficulties, particularly when high-dimensional problems are to be solved. Several techniques (as discussed in Section 2.7) have been proposed to control the degree of dependency modeling as a trade-off between the UMDA$_c$ model and the EMNA$_{\text{global}}$ model. In this chapter, the third analysis technique, namely variable screening, is introduced into the EDA and used to control the degree of dependency modeling. The general idea is that variables that are important (to the fitness function) may benefit from dependency modeling while other variables do not need this.

6.2.1 Measuring the Strength or Importance of Dependencies Between the Variables Using Elementary Effects

The Morris method, discussed in Section 3.6, is used here for identifying variable interactions and importance. As discussed earlier, the Morris method is based on measuring the mean and standard deviation of perturbations of individual variables for a given problem, calculated via so-called elementary effects terms. In these experiments, measurement has been done on the strength or importance of dependencies between the variables using the elementary effect terms.

6.2.2 Incorporating Variable Screening (Morris method) in an EDA

To incorporate elementary effect values ($E_i(\mathbf{x})$) into an EDA, two main decisions must be made. The first is how to calculate the $E_i(\mathbf{x})$ values themselves, which will happen on each generation of the sEDA. In theory, a sampling plan such as a full factorial design could be considered, however this requires the specification of a grid of equally spaced points in the search spaced points in the search space, of some predetermined resolution, perturbing each $\mathbf{x}^i$ along vertices in the grid and evaluating f at these points. Since this would not be practical inside an EDA, an alternative, based on the selected population size M_{sel} has been proposed. Specifically, the mean of the selected population ($\mathbf{m}$) is calculated for each dimension x_i.The population, P_{tot} is then formed by creating new solution vectors where the mean value is substituted in turn for each problem variable (e.g. $\mathbf{x^i} = x_i, \ldots, x_{i-1}, m_i, x_{i+1}, \ldots, x_n$). This produces nM_{sel} new solutions, which are evaluated using f.

Given the elementary effect values and their means and standard deviations, the second decision to be made is how to use them to determine the covariance matrix structure to be used in the sEDA model. To do this, the concepts of dominance and Pareto optimality from multi-objective optimization (see Section 5.4) have been utilized.

In the sEDA, a fixed fraction η of the variables need to be selected for covariance modeling. Variables that belong to the Pareto set are selected first. If more variables are required, then those which have the minimum (Euclidean) distance to the Pareto front are selected. On the other hand, if the number of variables on the Pareto front is greater than required, then a random subset of these variables is selected.

The complete framework of sEDA is summarized in Algorithm 6.1. The critical steps of the algorithm are as described above. Three algorithm parameters must be specified for implementation: the population size M, the selection parameter τ and the variable screening/selection parameter η. sEDA uses truncation selection: a fraction τ of the population with the best objective function values are retained for building/adapting the search model[1]. The mean ($\mathbf{m}$) of the selected population is then calculated for expanding the population. This expanded population is then used to calculate elementary effects values ($E(\mathbf{x})$) and their mean ($\overline{E}_i^*(\mathbf{x})$) and standard deviation ($std(E_i(\mathbf{x}))$) (for more details refer to Section 3.6). After selecting variables with respect to the Pareto set of the mean and standard deviation of $E(\mathbf{x})$, the covariance matrix for the EDA model is formed as a sparse matrix, with non-zero covariance terms for selected variables. This is used in combination with the EDA mean vector (estimated from the selected expanded population) and the model is then used to generate the new population as in a standard EDA. Within each generation, the variables which are used for generating the population for the next generation are indexed. The process is repeated until some stopping criterion is met.

One possible intuition for the sEDA algorithm is that, it is similar to hybrid (memetic) algorithm which combines an evolutionary algorithm with a local search implementation step.

6.3 Experimental Design

In this section, the effect of varying the truncation selection parameter (τ) and the model selection parameter (η) have been investigated for sEDA. In addition to this, the number of population used and the number of generations used during the implementation of the algorithms have been discussed.

6.3.1 Selection Parameter Settings for sEDA

sEDA contains two selection parameters, τ and η. Setting these values is important for experimental results. In this set of experiments, different combinations tried for τ and η are ((0.3, 0.1),(0.3, 0.3), (0.3, 0.5), (0.3,0.7), (0.3, 1.0), (0.5, 0.1), (0.5, 0.3), (0.5, 0.5), (0.5, 0.7), (0.5,1.0)). With these values, sEDA is tested using 6 different 10D benchmarking functions, which are also the functions from the Real-Parameter Black-Box Optimization Benchmarking (BBOB) experiment set [76]. sEDA is implemented with a population size of 500 having 300000 function evaluations. The algorithm is terminated when a difference between the best fitness value found and the global optimum is less than

[1] Rounding if $M \times \tau$ is not an integer.

Algorithm 6.1 sEDA Algorithm.

```
Given: Population size $M$, dimensionality $n$, selection parameter $0 < \tau < 1$, model selection parameter $0 < \eta < 1$.
Begin (set $t = 0$)
Initialize population $P$ by generating $M$ individuals uniformly in $S$.
while stopping criteria not met do
    Evaluate $f$ for population $P$.
    Truncation selection: $P_{sel} = M_{sel}$ best individuals from $P; M_{sel} = \text{Rnd}(M \cdot \tau)$ .
    Calculate sample mean ($\mathbf{m}$) of $P_{sel}$.
    Calculate $P_{tot}$ by expanding $P_{sel}$, successively replacing variable $1, \ldots, n$ with $m_i$ to produce $M_{tot}$ number of population.
    Evaluate $f$ for new individuals in $P_{tot}$ population.
    Selection: $P_{tot} = M_{tot}^{sel}$ best individuals; $M_{tot}^{sel} = \text{Rnd}(n \cdot M_{sel} \cdot \tau)$ individuals from $P_{tot}$.
    Calculate Elementary Effect ($E(\mathbf{x})$) of the fitness function of $P_{tot}^{sel}$.
    Calculate the mean $\mu_{\mathbf{t}}$ of the $P_{tot}^{sel}$.
    Calculate mean ($\overline{E}_i^*(\mathbf{x})$) and standard deviation ($std(E_i(\mathbf{x}))$) of $E(\mathbf{x})$.
    Determine the Pareto optimal solutions $p_o$ using $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$ as two objective functions.
    Let $B = \text{Round}(n \cdot \eta)$.
    If $p_o > B$, randomly choose $B$ variables from $p_o$.
    If $p_o < B$, select/add the next $B - p_o$ variables nearest to the Pareto front.
    Build $\Sigma_t$ using covariance terms for the $B$ selected variables and variance terms only for the remaining $n - B$ variables
    $p(\mathbf{x}) \leftarrow (\mu_{\mathbf{t}}, \Sigma_t)$.
    Generate $P$ new population by sampling from $p(\mathbf{x})$.
end while
```

or equal to 1e-08, otherwise when the algorithm attains the maximum allowed number of function evaluations.

Table 6.1 lists the mean and standard deviation of the best fitness values found over 100 runs for different combination of τ and η.

Best results with minimal mean value are represented in bold font. It is discovered from Table 6.1 that, out of 6 functions, 3 functions are better when τ = 0.3 and η = 0.3.

These results suggest that the performance is not highly sensitive to the values tested for these parameters. Hence the value of τ and η have been set to 0.3 and 0.3 respectively for sEDA for further experiments in Section 6.4.

6.3.2 Evaluation of sEDA and Comparisons

Experiments were conducted on the 10D and 50D versions of the artificial test functions. The population size of all the algorithms was set to 2000, to ensure comparability with previous results from the literature (Chapter 8 of [102]), with the exception of sEDA, since it uses far more function evaluations per generation. For the 10D and 50D functions the selection parameter is set to 0.5 for UMDA$_c$ and EMNA$_{\text{global}}$.

For comparison with previous results on the 10D functions, 301850 function evaluations were set. But when it comes to 50D functions, 10 times more function evaluations are used. Algorithms are terminated when the difference between the absolute value of the best fitness found and the known global optimum is less than or equal to 1e-08 or it exceeds the given number of function evaluations. For each single test, the result is averaged over 100 independent runs.

The number of function evaluations within each generation is different for different algorithms. In UMDA$_c$ and EMNA$_{\text{global}}$ the number of function evaluations at each generation is equal to the population size whereas in sEDA the number of function evaluations at each generation is a function of population size, selection parameter and dimension of the problem. The number of function evaluations at each generation for sEDA is $M + (M \times \tau \times n)$, where M is the population size, n is the dimensionality of the problem and τ is the standard truncation ratio. Therefore the maximum number of generations used by sEDA for 10D functions with 301850 function evaluations and 50D functions with 3000000 function evaluations are 150 and 188 respectively.

Using this formula, for a given fixed number of function evaluations, sEDA will always have fewer generations compared to UMDA$_c$ and EMNA$_{\text{global}}$. Hence to make the number of generations approximately the same for all the algorithms, a lesser number of population was used for sEDA. For 10D and 50D functions, 500 and 1000 populations are used respectively by sEDA.

The parameter values for the 10D and 50D experiments for different algorithms are presented in Table 6.2.

6.4 Results

Table 6.3 and Table 6.4 summarizes the results on 10D and 50D functions using UMDA$_c$, EMNA$_{\text{global}}$ and sEDA algorithms. Best fitness values found and the number of evaluations required to reach the

Table 6.1: Solution quality comparison having different values of τ and η with 10D 6 different function from BBOB in sEDA.

Function	(τ, η)	Best fitness values	Function	(τ, η)	Best fitness values
fun_1	(0.3,0.1)	4.1859e-10±1.9377e-10	fun_4	(0.3,0.1)	7.8814e+00±1.2229e-01
	(0.3,0.3)	**4.0747e-10±2.1534e-10**		(0.3,0.3)	**7.6111e+00±2.4945e-01**
	(0.3,0.5)	4.4503e-10±1.8162e-10		(0.3,0.5)	7.6258e+00±2.6868e-01
	(0.3,0.7)	4.5039e-10± 1.7320e-10		(0.3,0.7)	7.6505e+00±2.9905e-01
	(0.3,1.0)	4.4686e-10±1.5717e-10		(0.3,1.0)	7.5280e+00±4.3890e-01
	(0.5,0.1)	6.9786e-10± 2.7656e-10		(0.5,0.1)	8.1424e+00±6.2824e-02
	(0.5,0.3)	7.8054e-10± 2.6987e-10		(0.5,0.3)	8.0293e+00±1.4390e-01
	(0.5,0.5)	7.2316e-10± 3.0482e-10		(0.5,0.5)	8.0092e+00± 1.3398e-01
	(0.5,0.7)	7.6307e-10±2.8930e-10		(0.5,0.7)	7.9782e+00±1.5023e-01
	(0.5,1.0)	7.5919e-10±2.5931e-10		(0.5,1.0)	7.9046e+00±2.8372e-01
fun_2	(0.3,0.1)	3.9609e-10 ±1.7273e-10	fun_5	(0.3,0.1)	3.7458e-09±8.2402e-10
	(0.3,0.3)	**1.9899e-10±1.4000e-01**		(0.3,0.3)	3.8316e-09±7.4032e-10
	(0.3,0.5)	4.2146e-10± 1.9201e-10		(0.3,0.5)	3.5363e-09±7.4431e-10
	(0.3,0.7)	4.9748e-02±2.1794e-01		(0.3,0.7)	3.7028e-09±8.9438e-10
	(0.3,1.0)	6.6241e-02±2.4107e-01		(0.3,1.0)	**3.4978e-09±7.7805e-10**
	(0.5,0.1)	7.5311e-10 ± 2.7304e-10		(0.5,0.1)	6.4005e-09±1.1508e-09
	(0.5,0.3)	1.7040e-06±1.7033e-05		(0.5,0.3)	6.2027e-09±1.1695e-09
	(0.5,0.5)	7.3522e-10±2.7501e-10		(0.5,0.5)	6.1331e-09±1.2245e-09
	(0.5,0.7)	7.4852e-10±2.8924e-10		(0.5,0.7)	5.9026e-09±1.0768e-09
	(0.5,1.0)	7.4683e-10 ±2.4104e-10		(0.5,1.0)	6.2874e-09 ±1.1587e-09
fun_3	(0.3,0.1)	4.2197e-10±1.8961e-10	fun_6	(0.3,0.1)	1.603e-02±2.5277e-02
	(0.3,0.3)	4.1811e-10±1.8736e-10		(0.3,0.3)	4.4690e-03±5.6286e-03
	(0.3,0.5)	7.3961e-05±7.3960e-04		(0.3,0.5)	1.9019e-03±2.7286e-03
	(0.3,0.7)	**3.9370e-10±1.4854e-10**		(0.3,0.7)	1.9154e-04±3.3700e-04
	(0.3,1.0)	8.7162e-06±8.4289e-05		(0.3,1.0)	**5.3204e-10±2.8298e-10**
	(0.5,0.1)	7.5749e-10± 2.6752e-10		(0.5,0.1)	8.1264e-02±9.9892e-02
	(0.5,0.3)	7.7859e-10±2.4938e-10		(0.5,0.3)	3.2813e-02±3.9157e-02
	(0.5,0.5)	7.8415e-10±2.8021e-10		(0.5,0.5)	1.3922e-02±1.7797e-02
	(0.5,0.7)	7.6587e-10±2.7093e-10		(0.5,0.7)	4.4019e-03±9.0104e-03
	(0.5,1.0)	7.1325e-10±2.5283e-10		(0.5,1.0)	1.2561e-09±4.9750e-10

Table 6.2: Parameter values for different Algorithms.

Algorithm	Dim	Population	No. of Gens.	No. of Func. Eval
UMDA$_c$	10	2000	150	301850
EMNA$_{\text{global}}$		2000	150	301850
sEDA		500	150	301850
UMDA$_c$	50	2000	1500	3000000
EMNA$_{\text{global}}$		2000	1500	3000000
sEDA		1000	188	3000000

final solution are recorded for each of the experiments. The best result with the minimum mean value with respect to fitness as well as number of function evaluations are highlighted in bold font. A t-test has been done between best values found by sEDA and 2 other EDAs, (with null hypothesis being that each set of experimental results is drawn from distributions with equal mean (and assumed unequal variances)).

6.4.1 Discussion for 10D and 50D Artificial Test Functions

Tables 6.3 and 6.4 show the comparison of the 3 algorithms using 6 different functions.

$Sphere\ function$: In Sphere function, all the variables are independent to each other and equally important to the problem (Sphere function is an example of category 1, discussed in Section 4.7). Although it always facilitates univariate model-based EDAs in solving the problems (proved in Section 5.5.2) Tables 6.3 and 6.4 show that the performance of UMDA$_c$, EMNA$_{\text{global}}$ and sEDA are similar in both 10D and 50D dimensions. In terms of the the number of the function evaluations, for 10D Sphere, sEDA requires lesser number of function evaluations whereas the requirements of function evaluations are more when dimension increases as compared to 2 other EDAs.

$Griewangk\ function$: This is a multimodal and non-separable function. For the 10D and 50D version, sEDA, UMDA$_c$ and EMNA$_{\text{global}}$ are performing similar in terms of best fitness values found. In terms of the number of function evaluations, sEDA requires less number of function evaluations as compare to UMDA$_c$ and EMNA$_{\text{global}}$ for 10D version, but in 50D sEDA is using significantly larger number of function evaluations than the other 2 EDAs.

$Ackely\ function$: Ackely has several local minima, hence it can be difficult to optimize. The performance of sEDA compared to UMDA$_c$ and EMNA$_{\text{global}}$ are similar in terms of minimum average best fitness values.But it has been concluded that sEDA is better in 10D because it requires a lesser number of function evaluations as compared to the other 2 algorithms. For 50D Ackely, sEDA requires significantly more function evaluations.

$Rosenbrock\ function$: The results of Section 5.5.4 give some insight into why the Rosenbrock function is a very hard problem for EDAs. The comparison of problem variables and EDA model variables suggests that any EDA having control over the covariance parameter will work better in

Table 6.3: Solution quality comparison for 10D problem. Bold font represents the best result.

Function	Alg.	Best Fitness	No. of Evaluations
Sphere	U_c	**6.91E-09±1.8E-09(-)**	1.5E+05±1.4E+03(+)
	E_g	7.67E-09±1.6E-09(+)	1.4E+05±1.8E+03(+)
	s	6.98E-09±1.9E-09	**5.5E+04±9.7E+02**
Griewangk	U_c	7.78E-09±1.6E-09(+)	1.4E+05±1.7E+03(+)
	E_g	7.32E-09±1.5E-09(+)	1.4E+05±1.5E+03(+)
	s	**6.82E-09±1.8E-09**	**6.0E+04±3.2E+03**
Ackely	U_c	8.50E-09±1.1E-09(-)	2.0E+05±1.4E+03(+)
	E_g	8.40E-09±1.1E-09(-)	2.1E+05±1.4E+03(+)
	s	**8.27E-09±1.2E-09**	**7.8E+04±9.2E+02**
Rosenbrock	U_c	8.21E+00±2.3E-02(+)	3.0E+05±0.0E+00
	E_g	7.80E+00±1.5E-01(+)	3.0E+05±0.0E+00
	s	**7.51E+00±2.6E-01**	3.0E+05±0.0E+00
Rastrigin	U_c	7.32E-09±1.8E-09(+)	2.2E+05±5.4E+03(+)
	E_g	**6.91E-09±1.8E-09(+)**	2.2E+05±4.6E+03(+)
	s	1.99E-02±1.4E-01	**7.8E+04±3.2E+03**
Rotated Ellipsoid	U_c	3.72E+00±3.5E+00(+)	3.0E+05±0.0E+00
	E_g	**7.51E-09±2.1E-09(+)**	**1.3E+04±1.5E+03(+)**
	s	6.87E-01±1.0E+00	3.0E+05±0.0E+00

U_c stands for UMDA$_c$, E_g stands for EMNA$_{\text{global}}$, s stands for sEDA.
+ sign, the value of t-test (2 tailed)>0.05, indicates statistically significant difference when compared with sEDA.
- sign, the value of t-test (2 tailed)<0.05, indicates no statistically significant difference when compared with sEDA.
No marker indicated, when the results are same.

this function. sEDA is an example of such an EDA, and while it does not converge towards the global optimum, it outperforms UMDA$_c$ and EMNA$_{\text{global}}$ with significantly better solutions. These 3 algorithms use the same number of function evaluations to achieve best fitness values.

Rastrigin function : Rastrigin function which is a multimodal and separable problem shows that the performance of sEDA is degraded significantly as compared to UMDA$_c$ and EMNA$_{\text{global}}$. For the 10D version of the function, both UMDA$_c$ and EMNA$_{\text{global}}$ performs better (since an adequate population is supplied), while for the 50D version, UMDA$_c$ performs better. Further examination on the best results of sEDA in 10D Rastrigin function shows that, out of 100 runs, the best results are within 10e-8 for 95 different runs, where as rest of the runs are getting the best result in the range 10e-2. While taking the average, the best value is in the range 10e-2. Taking average results in poor performance, hence considering this results shows the wrong interpretation of experimental results. *Rotated Ellipsoid function* : For the 10D and 50D Rotated Ellipsoid functions, the performance of EMNA$_{\text{global}}$ surpasses sEDA and UMDA$_c$. In this case, the comparison of model variables and EDA model variables are matched exactly the same (found from Section 5.5.3). The comparison also suggests that, EMNA$_{\text{global}}$ will perform better in this function.

To approximately set the value of η for 10D Griewangk, 10D Ackley and 10D Rastrigin functions, refer to Section 5.8.1.

Table 6.4: Solution quality comparison for 50D problem. Bold font represents the best result.

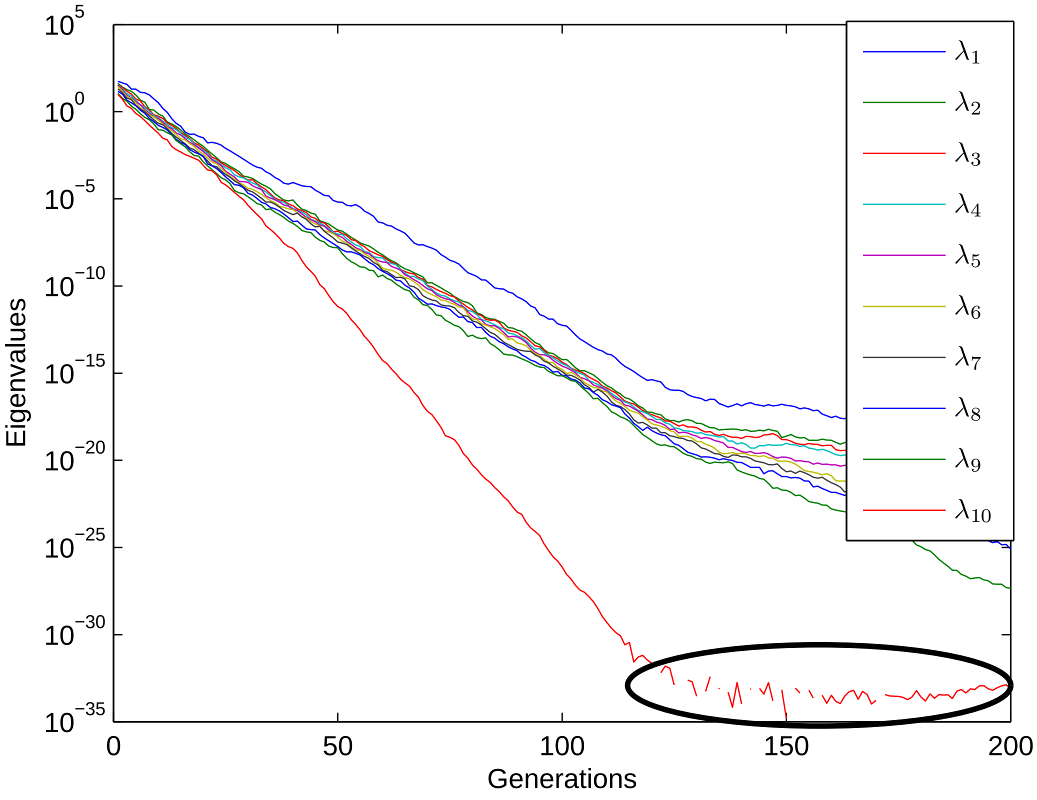
Function	Alg.	Best Fitness	No. of Evaluations
Sphere	U_c	8.88E-09±8.5E-10(-)	3.9E+05±1.7E+03(+)
	E_g	2.65E-08±1.8E-07(+)	**3.9E+05±2.6E+05(+)**
	s	**8.81E-09±9.1E-10**	1.3E+06±7.8E+03
Griewangk	U_c	8.87E-09±8.7E-10(-)	**3.5E+05±1.6E+03(+)**
	E_g	1.42E-08±3.6E-08(-)	4.6E+05±5.9E+05(+)
	s	**8.58E-09±7.2E+10**	1.1E+06±6.9E+03
Ackely	U_c	4.38E-08±1.8E-07(-)	7.2E+05±7.3E+05(+)
	E_g	9.41E-09±4.4E-10(-)	**5.3E+05±1.8E+03(+)**
	s	**9.35E-09±4.5E+10**	1.7E+06±8.2E+03
Rosenbrock	U_c	5.25E+01±1.8E+00(+)	3.0E+06±0.0E+00
	E_g	4.78E+01±2.8E-02(+)	3.0E+06±0.0E+00
	s	**4.73E+01±3.6E-01**	3.0E+06±0.0E+00
Rastrigin	U_c	**8.84E-09±8.1E-10(+)**	**5.7E+05±9.2E+03(+)**
	E_g	7.05E+00±5.0E+00(+)	3.0E+06±0.0E+00
	s	3.18E-01±5.0E-01	3.0E+06±0.0E+00
Rotated Ellipsoid	U_c	7.22E+02±2.2E+01(+)	3.0E+06±0.0E+00
	E_g	**9.52E-09±6.6E-09(+)**	**3.5E+05±2.6E+05(+)**
	s	2.09E+02±7.5E+01	3.0E+06±0.0E+00

U_c stands for UMDA$_c$, E_g stands for EMNA$_{\text{global}}$, s stands for sEDA.
+ sign, the value of t-test (2 tailed)>0.05, indicates statistically significant difference when compared with sEDA.
- sign, the value of t-test (2 tailed)<0.05, indicates no statistically significant difference when compared with sEDA.
No marker indicated, when the results are same.

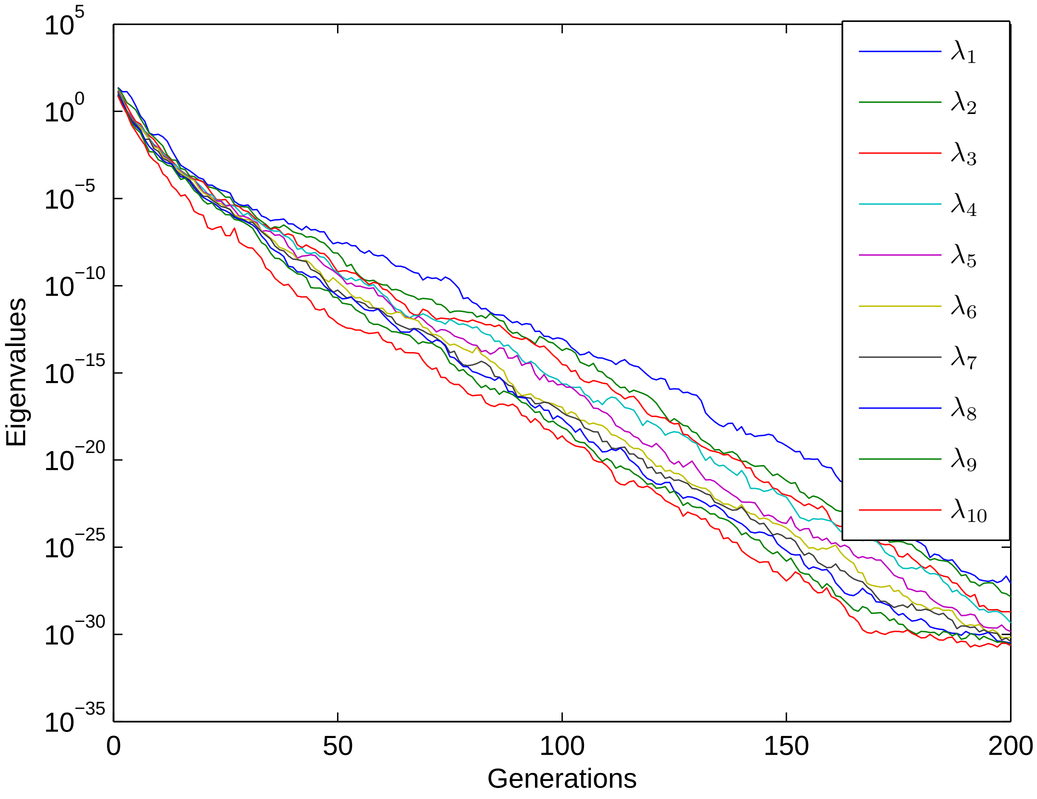
6.5 Examination of sEDA Behavior

6.5.1 No ill-conditioned Matrix

From a stability point of view, sEDA is much more stable than EMNA$_{\text{global}}$. If the population size is not large as compared to the problem dimensionality, EMNA$_{\text{global}}$ produces an ill-conditioned matrix, which is a major issue. Although sEDA uses some characteristics of EMNA$_{\text{global}}$, it does not produce any ill-conditioned matrix because there are far fewer covariance parameters to be estimated from the data. This can be experimentally shown by plotting the eigenvalues of the Σ of EMNA$_{\text{global}}$ and Σ_t of sEDA. EMNA$_{\text{global}}$ uses 0.5 as its selection ratio, while sEDA uses selection parameter (τ) and model selection parameter (η) as 0.3 and 0.3 respectively. Both algorithms are implemented on the 10D Rosenbrock test function with 200 number of generations and 100 population.



(a)



(b)

Figure 6.1: Eigenvalues of the Covariance Matrix over generations, (6.1a) EMNA$_{\text{global}}$ on 10D Rosenbrock Function, (6.1b) sEDA on 10D Rosenbrock Function.

Figure 6.1a shows a representative run of the performance of EMNA$_{\text{global}}$ where the ellipse in the figure represents the existence of a ill-conditioned covariance matrix. Figure 6.1b compares this with a run of sEDA, which shows that sEDA doesn't have any ill-conditioned matrix, even though the population size is small. The estimation of many parameters in EMNA$_{\text{global}}$ means that it is likely that one or more eigenvalues will become very small, making the covariance matrix ill-conditioned.

Previous techniques for repairing ill-conditioned covariance matrices are described in [42, 43] (for more discussion see Section 2.7). A comparison table is discussed in [43]. The methods include CMR, ECMR (Extended CMR) and ECMR0 (Extended CMR-zero) which is implemented on different artificial test functions. The explanations of algorithms and parameter details are mentioned in [43].

In the next experiments, the sEDA algorithm is compared with the 3 algorithms discussed in [43] using Rosenbrock, SumCan and Schwefel's Problem 2.26 functions (described in Table 2.1). These algorithms are covariance matrix repairing algorithms. Since these algorithms are implemented on the set of problems in [43], it has been easy to make a comparison between sEDA and the different covariance matrix repairing algorithms. This is fairly a small set of problems, here the main motive is to show that, sEDA is also comparable to these covariance matrix repairing algorithms.

Except Sumcan, the other 2 functions are minimization functions. sEDA uses same population size and number of function evaluations as described in [43], with τ=0.3 and η=0.3 as selection ratios. The comparison values shown in Table 6.5 with the average performance of 100 independent runs of each algorithm. Table 6.5 as well as discussion from [43] revealed that CMR, ECMR and ECMR0 have approximately identical performances. From Table 6.5, it can be concluded that, the performance of sEDA is better than the other 3 algorithms in terms of the best fitness values in all these functions, but sEDA requires a greater number of function evaluations in each case.

6.5.2 Role of η

As discussed in Section 6.2.2, the value of η in sEDA determines the number of variables to be selected for covariance modeling. When the value of $\eta = 1$, sEDA model becomes the full covariance matrix, whereas, when $\eta = 0$, sEDA model becomes a diagonal covariance matrix. However, the algorithm is not similar to EMNA$_{\text{global}}$ and UMDA$_c$. sEDA generates a different larger population at each generation because of the elementary effect calculation, which differentiate sEDA from EMNA$_{\text{global}}$ and UMDA$_c$. The performance comparison of sEDA while $\eta = 0$ and $\eta = 1$ is discussed here.

Hence the value of η depends on the problem to be solved and some experimentation is expected to be required to find the optimal value in any given application. The results and comparison of model variables and problem variables for various problems have been analyzed in Chapter 5.

The three problems described in Chapter 5, where the properties of model variables are exactly matched to the problem variables are 10D Sphere, 2D Elliptical and the 10D Rotated Ellipsoid functions.

sEDA is implemented on these 3 functions. The population size, number of generations as well as the number of function evaluations are the same as described in Table 6.2 for 10D problems. The selection ratio (τ) for UMDA$_c$ and EMNA$_{\text{global}}$ is set to 0.5. For sEDA, the selection ratio (τ) is

Table 6.5: Comparision of sEDA, CMR, ECMR and ECMR0. Bold font represents the best result.

Function	Alg.	Best fitness	No. of Evaluations
Rosenbrock	sEDA	**7.52E+00±3.3E-01**	3.0E+05±0.0E+00
	CMR	2.35E+02±1.2E+02(+)	**3.4E+04±2.7E+04(+)**
	ECMR	2.26E+02±9.9E+01(+)	3.9E+04±4.6E+04(+)
	ECMR0	2.16E+02±8.5E+01(+)	4.5E+04±5.9E+04(+)
Sumcan	sEDA	**8.64E+02±1.1E+03**	3.0E+05±0.0E+00
	CMR	9.88E+04±7.9E+03(+)	4.9E+04±4.1E+04(+)
	ECMR	9.73E+04±1.4E+04(+)	5.2E+04±4.7E+04(+)
	ECMR0	9.85E+04±9.7E+03(+)	**4.9E+04±4.1E+04(+)**
Schwefel Problem 2.26	sEDA	**-1.81E+04±2.1E+04**	3.0E+05±0.0E+00
	CMR	-4.93E+03±6.8E+02(+)	5.9E+04±6.2E+04(+)
	ECMR	-4.91E+03±7.0E+02(+)	5.9E+04±6.2E+04(+)
	ECMR0	-5.02E+03±6.8E+01(+)	**5.6E+04±5.7E+04(+)**

+ sign, the value of t-test (2 tailed)>0.05, indicates statistically significant difference when compared with sEDA.
- sign, the value of t-test (2 tailed)<0.05, indicates no statistically significant difference when compared with sEDA.

Table 6.6: Solution Quality Comparison for the 10D Sphere and 2D Elliptical Functions when η = 0.0. Bold font represents the best result.

Function	Alg.	Best Fitness	No. of Evaluations
10D Sphere	U_c	6.91E-09±1.8E-09(-)	1.5E+05±1.4E+03(+)
	s	**6.90E-09±2.0E-09**	**2.7E+01±4.6E-01**
2D Elliptical	U_c	2.91E-09±1.8E-09(-)	1.5E+01±1.0E+00(+)
	s	**2.72E-09±2.8E-09**	**5.5E+00±6.1E-01**

U_c stands for UMDA$_c$, s stands for sEDA.
+ sign, the value of t-test (2 tailed)>0.05, indicates statistically significant difference when compared with sEDA.
- sign, the value of t-test (2 tailed)<0.05, indicates no statistically significant difference when compared with sEDA.

Table 6.7: Solution Quality Comparison for the 10D Rotated Ellipsoid Function when η = 1.0. Bold font represents the best result.

Function	Alg.	Best Fitness	No. of Evaluations
10D Rotated Ellipsoid	E_g	7.51E-09±1.7E-09(-)	**1.3E+04±1.5E+03(+)**
	s	**7.42E-09±1.8E-09**	5.9E+04±1.0E+03

E_g stands for EMNA$_{\text{global}}$, s stands for sEDA
+ sign, the value of t-test (2 tailed)>0.05, indicates statistically significant difference when compared with sEDA.
- sign, the value of t-test (2 tailed)<0.05, indicates no statistically significant difference when compared with sEDA.

set to 0.3, whereas the model selection ratio (η) depends upon the type of the problem, since η is problem-dependent.

For the 10D Sphere and 2D Elliptical functions, the value of η for sEDA is set to 0.0 whereas for the 10D Rotated Ellipsoid function, the value of η is set to 1.0. The results are based on 100 independent runs.

Table 6.6 illustrates the results of the 10D Sphere and 2D Elliptical functions, when the value of η for sEDA is set to 0.0. It shows that for both the functions, the results of best fitness values achieved by sEDA and UMDA$_c$ are similar, where as the number of function evaluations attained by sEDA is better than UMDA$_c$.

Table 6.7 shows the comparison between EMNA$_{\text{global}}$ and sEDA when η = 1 in a 10D Rotated Ellipsoid function. The performance of sEDA is similar to EMNA$_{\text{global}}$ with respect to the best fitness values but it requires more number of function evaluations than EMNA$_{\text{global}}$ to achieve the solution.

6.6 Scaling sEDA to High-Dimensional Problems

Due to the nature of the algorithm, sEDA as described above will require a relatively large number of function evaluations when applied to high-dimensional problems. This is due to the fact that, for each generation the population size is directly proportional to the dimension of the problem. Hence, the number of function evaluations per generation is $O(nM)$.

6.7 Modified sEDA for Solving High Dimensional Problems (sEDA-lite)

To address the issue of population size, a modified version of sEDA, called sEDA-lite is also proposed here. The algorithm uses the same principles as sEDA but differs in the calculation of the elementary effect values. As discussed in Section 6.2.2, using the mean of the selected population to calculate elementary effect values in sEDA, results in a significant increase in the number of fitness function evaluations required per generation. In sEDA-lite, the median of each dimension in the selected population is used instead to calculate elementary effects. The mechanics of the calculations in sEDA-lite are the same as those in sEDA apart from the use of the median of the elementary effects values in place of the mean. Like the mean, the median is representative of the center of the selected population. However, the median is by definition located at an individual. Hence, all calculations in Equation (3.3) only use individuals in the selected population (their fitness values have already been evaluated). It follows that, for each generation of sEDA-lite, the number of function evaluations is reduced from $M + (M \times n \times \tau)$ to M.

The pseudocode for sEDA-lite is presented in Algorithm 6.2. Truncation selection is used with parameter τ. The median $\tilde{m}$ of the selected population is used to calculate elementary effect values.

Firstly, the median of the selected population has been calculated dimension-wise. Let for a n dimensional problem, the median values are denoted as $\tilde{m}_1, \tilde{m}_2, \ldots, \tilde{m}_n$, where $\tilde{m}_n$ belongs to one of the value of x_i, where $i = 1, \ldots, n$. Suppose $\tilde{m}_i$ belonging to the kth value of x_i (ie. $\tilde{m}_i$=x_i^k), where $k \in \{1, \ldots, length(M * \tau)\}$. The fitness function $f(\mathbf{x^k})$ belongs to the kth candidate is used for calculating the elementary effect for x_i. Hence the formula for calculating the elementary effect for x_i is,

$$E_{x_{ij}} = \frac{f(\mathbf{x^k}) - f(\mathbf{x_j})}{x_i^k - x_{ij}} \tag{6.1}$$

where $j = 1, \ldots, length(M * \tau)$; $i = 1, \ldots, n$ and $i \neq k$ as well as $j \neq k$. Set $E_{kk} = 0$.

sEDA-lite uses the same Pareto optimal concept as used in sEDA, to select the important and dependent variables in the problem. After selection, the covariance matrix for the EDA model is formed as a sparse matrix, with non-zero covariance terms for selected variables. This is used in combination with the EDA mean vector (estimated from the selected expanded population) and the model is then used to generate the new population as in a standard EDA. The process is repeated until some stopping criterion is met.

Table 6.8: Complexity of sEDA and sEDA-lite.

	sEDA	sEDA-lite
Model estimation	$O((n\eta)^2 M(1 * n))$ $+O(nM)$	$O((n)^2 M)$ $+O(nM)$
Solution Sampling	$O(n^2 M(1+n))$	$O(n^2 M)$

The complexities of sEDA and sEDA-lite are shown in Table 6.8. Although both the algorithms are in-between UMDA$_c$ and $EMNA_{global}$, From Table 6.8 shows that, sEDA-lite uses a less number of function evaluations than sEDA. For the Model estimation, sEDA requires more n number of function evaluations than sEDA-lite. For the Solution Sampling, sEDA requires $(1+n)$ times more function evaluations than sEDA-lite.

6.8 Experimental Design

To more generally and thoroughly evaluate and compare the performance of the algorithms, experiments have been carried out on 3 different sets of problems. The first are artificial test functions.

Algorithm 6.2 Pseudo code for sEDA-lite

```
Given: Population size $M$, dimensionality $n$, selection parameter $0 < \tau < 1$, model selection parameter $0 < \eta < 1$.
Begin (set $t = 0$)
Initialize population $P$ by generating $M$ individuals uniformly in $S$.
while stopping criteria not met do
    Evaluate $f$ for population $P$.
    Truncation selection: $P_{sel} = M_{sel}$ best individuals from $P; M_{sel} = \text{Rnd}(M \cdot \tau)$ .
    Calculate mean, $\vec{\mu}$ and median, $\vec{\tilde{m}}$ of $P_{sel}$
    Calculate $\tilde{m}$ = median($P_{sel}$), where $\tilde{m} = \tilde{m}_1, \cdots, \tilde{m}_n$.
    for $i$ = 1 to $n$ do
        for $j$ = 1 to $M_{sel}$ do
            Calculate $E_{x_{ij}}$ using Eqn.6.1, where $\tilde{m}$ is the baseline point and the perturbation value is given by $j^{th}$ individual
        end for
    end for
    Calculate mean ($\overline{E}_i^*(\mathbf{x})$) and standard deviation ($std(E_i(\mathbf{x}))$) of $E(\mathbf{x})$.
    Determine the Pareto optimal solutions using $\overline{E}_i^*(\mathbf{x})$ and $std(E_i(\mathbf{x}))$. Let this number of variables be $p_o$ .
    Let $B = \text{Round}(n \cdot \eta)$.
    If $p_o > B$, randomly choose $B$ variables from $p_o$.
    If $p_o < B$, select/add the next $B - p_o$ variables nearest to the Pareto front.
    Build $\Sigma_t$ using covariance terms for the $B$ selected variables and variance terms only for the remaining $n - B$ variables
    $p(\mathbf{x}) \leftarrow (\vec{\mu}, \Sigma_t)$.
    Generate $P$ new population by sampling from $p(\mathbf{x})$.
end while
```

The second set of problems are Circles in a Square (CiaS) packing problems and the third set are the 51 customer location-allocation problems with different numbers of facilities. Both artificial and real world problems are considered because the artificial test functions are widely used in various optimization research and since their shape is known, they are very easy for comparison. In addition to this, there are a number of advantages in using real world problems, since the results are application-oriented. The real world problems used are all scalable in terms of dimensionality. They are also known to have features that make them difficult to solve for many algorithms, e.g. the fitness functions are not differentiable and they contain a large number of local optima.

6.8.1 Selection Parameter Settings for sEDA-lite

Like sEDA, sEDA-lite also contains two selection parameters, τ and η. A different combinations of τ and η has been tested in 6 different 50D benchmarking functions (refer to section 6.3.1 for the different combinations of τ and η and the black-box functions). sEDA-lite is implemented with a population size of 2000 having 300000 function evaluations. The algorithm is terminated when the a difference between best fitness value and the global optimum is less than equal to 1e-08 or the algorithm attains the maximum allowed number of function evaluations.

Table 6.9 lists the mean and standard deviation of the best fitness values found over 100 runs for different combination of τ and η. It has been found out that, out of 6 functions, 3 function attains the best fitness value when τ=0.3 and η=0.1.

6.8.2 Artificial Test Problems

The commonly used artificial test problems listed in Table 6.10 are taken from Dong et. al. [41]. The problems can be categorized into separable unimodal (F_1 and F_2), non-separable ($F_3, \ldots, F_{10}$) and multimodal (F_{11}, F_{12}, F_{13}) problems. For these test functions, an offset value is added to the functions to move the global optimum away from the origin. The offset values used in the test functions are the same as described in [41] except for F_4 and F_6. For these 2 functions, the offset values are generated randomly, because [41] also generates the offset value randomly. Due to the random nature of the offset values of these problems, only the algorithms whose pseudo code are available to us are implemented. The experiments on these functions are not comparable.

Again, the offset values used in [41] for these problems are different since the offset data is not available for these problems, hence the offset value is generated using the same formula as described in [41].

The problem sizes were 50D and 100D for each artificial test function discussed. The number of function evaluations was chosen as $10000 \times n$. sEDA-lite was implemented on these test functions. The population sizes in [41] were tested for sEDA-lite (i.e., 200, 500, 1000 and 2000). The selection parameter τ equal to 0.5 and 0.3 was taken since 0.5 is the selection ratio which is used in most of the literature [102] and 0.3 is the selection ratio used for evaluating the results for sEDA. The model selection parameter (τ) was set to 0.1 because it has been seen that UMDA$_c$ is performing better than EMNA$_{\text{global}}$ in all the artificial test functions discussed in [41]. The parameter values while

Table 6.9: Solution quality comparison having different values of τ and η with 50D 6 different function from BBOB in sEDA-lite.

Function	(τ, η)	Best fitness values	Function	(τ, η)	Best fitness values
fun_1	(0.3,0.1)	**8.6140e-09±9.2037e-10**	fun_4	(0.3,0.1)	4.7737e+01±3.1981e-02
	(0.3,0.3)	8.8520e-09±8.1285e-10		(0.3,0.3)	4.7747e+01±3.2180e-02
	(0.3,0.5)	8.6975e-09±9.0483e-10		(0.3,0.5)	**4.7727e+01±3.3758e-02**
	(0.3,0.7)	8.6210e-09±9.9506e-10		(0.3,0.7)	4.7727e+01±4.4864e-02
	(0.3,1.0)	2.9679e-05±2.2210e-04		(0.3,1.0)	5.1778e+01±2.3597e+00
	(0.5,0.1)	2.0836e-05±2.3387e-06		(0.5,0.1)	4.7805e+01±2.3610e-02
	(0.5,0.3)	1.9642e-05±1.9041e-06		(0.5,0.3)	4.7806e+01±2.9921e-02
	(0.5,0.5)	1.7086e-05±1.7491e-06		(0.5,0.5)	4.7797e+01±2.6589e-02
	(0.5,0.7)	1.3047e-05±1.4664e-06		(0.5,0.7)	4.7791e+01±3.2135e-02
	(0.5,1.0)	2.9791e-06±4.2177e-07		(0.5,1.0)	5.2609e+01±2.0212e+00
fun_2	(0.3,0.1)	1.6662e-03±1.4682e-03	fun_5	(0.3,0.1)	**3.0595e-07±2.9621e-08**
	(0.3,0.3)	2.4410e-03±2.7017e-03		(0.3,0.3)	5.2288e-07±2.9591e-08
	(0.3,0.5)	9.3648e-04±9.7742e-04		(0.3,0.5)	4.2262e-07±2.4349e-08
	(0.3,0.7)	**2.6012e-04±3.1926e-04**		(0.3,0.7)	3.1976e-07±1.8835e-08
	(0.3,1.0)	1.0063e+01±4.4448e+00		(0.3,1.0)	5.4330e-05±1.5653e-04
	(0.5,0.1)	5.7978e+01±2.1955e+01		(0.5,0.1)	1.8043e-04±1.0338e-05
	(0.5,0.3)	5.3092e+01±1.8768e+01		(0.5,0.3)	1.7469e-04±1.0118e-05
	(0.5,0.5)	4.9049e+01±2.1461e+01		(0.5,0.5)	1.6290e-04±9.0224e-06
	(0.5,0.7)	3.1959e+01±1.7041e+01		(0.5,0.7)	1.4100e-04±8.6207e-06
	(0.5,1.0)	7.8826e+00±5.1523e+00		(0.5,1.0)	6.6504e-05±7.3875e-06
fun_3	(0.3,0.1)	**8.2984e-09±9.2698e-10**	fun_6	(0.3,0.1)	9.2559e+00±2.3440e+00
	(0.3,0.3)	8.5845e-09±9.2434e-10		(0.3,0.3)	9.1217e+00±1.9427e+000
	(0.3,0.5)	8.4970e-09±8.5480e-10		(0.3,0.5)	8.1087e+00±1.6505e+00
	(0.3,0.7)	8.5974e-09±1.0399e-09		(0.3,0.7)	5.7582e+00±1.3445e+00
	(0.3,1.0)	3.4861e-05±3.1290e-04		(0.3,1.0)	2.5785e-07±1.6893e-06
	(0.5,0.1)	7.0779e-07±7.6088e-08		(0.5,0.1)	2.7403e+01±4.2109e+00
	(0.5,0.3)	6.7771e-07±7.2366e-08		(0.5,0.3)	2.7837e+01±4.0958e+00
	(0.5,0.5)	5.8874e-07±5.4882e-08		(0.5,0.5)	2.7658e+01±4.1012e+00
	(0.5,0.7)	4.4554e-07 ±4.1413e-08		(0.5,0.7)	2.5255e+01±3.9227e+00
	(0.5,1.0)	1.0216e-07±2.0426e-08		(0.5,1.0)	**8.7761e-09±8.0969e-10**

Table 6.10: A set of Artificial Problems obtained from [41].

	Description	Expression	Domain
F_1	Sphere	$F(\vec{x}) = \sum_{i=1}^{n} x_i^2$	$[-100,100]^n$
F_2	Shifted Sphere	$F(\vec{x}) = \sum_{i=1}^{n} z_i^2 + f_{bias_1}, \vec{z} = \vec{x} - \vec{o}$	$[-100,100]^n$
F_3	Schwefel's Problem 2.21	$F(\vec{x}) = max_i \{\lvert x_i \rvert, 1 \leq i \leq n\}$	$[-100,100]^n$
F_4	Shifted F_3	$F(\vec{x}) = max_i \{\lvert z_i \rvert, 1 \leq i \leq n\}$, $\vec{z} = \vec{x} - \vec{o}$	$[-100,100]^n$
F_5	Schwefel	$F(\vec{x}) = \sum_{i=1}^{n} [(x_1 - x_i^2)^2 + (x_i - 1)^2]$	$[-10,10]^n$
F_6	Shifted F_5	$F(\vec{x}) = \sum_{i=1}^{n} [(z_1 - z_i^2)^2 + (z_i - 1)^2]$, $\vec{z} = \vec{x} - \vec{o}$	$[-10,10]^n$
F_7	Rosenbrock	$\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-100,100]^n$
F_8	Shifted Rosenbrock	$\sum_{i=1}^{n-1} [100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2]$ $+ f_{bias_6}, \vec{z} = \vec{x} - \vec{o} + \vec{1}$	$[-100,100]^n$
F_9	Shifted Rotated High Conditioned Elliptic	$F(\vec{x}) = \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} z_i^2 + f_{bias_3}$ $\vec{z} = (\vec{x} - \vec{o}).\mathbf{M}$	$[-100,100]^n$
F_{10}	Schwefel 2.6 with Global Optimum on Bounds	$F(\vec{x}) = max\{\lvert \mathbf{A}_i\vec{x} - \mathbf{B}_i \rvert\} + f_{bias_5}$ $i = 1, \ldots, n.$	$[-100,100]^n$
F_{11}	Rastrigin	$\sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$	$[-5,5]^n$
F_{12}	Shifted Rotated Rastrigin	$\sum_{i=1}^{n} [z_i^2 - 10cos(2\pi z_i) + 10]$ $+ f_{bias_{10}}, \vec{z} = (\vec{x} - \vec{o}).\mathbf{M}$	$[-5,5]^n$
F_{13}	Shifted Expanded Griewangk plus Rosenbrock	see the details in [144]	$[-3,1]^n$

The shifted global optima are generated following the same way of [144]. Here **M** is the transformation matrix.

implementing sEDA-lite on the artificial test functions are as follows: for most of the functions the best population size was 2000 with the selection ratio 0.3 except F_1 and F_2, where the population size was 200 and selection ratio was 0.5. The algorithm stopped when the difference between the global optimum and the optimal values obtained from the algorithm was 1E-12 or it attained the maximum number of function evaluations. The results were reported based on 25 repeated trials.

6.8.3 Real World Problems

The Circle in a Square packing and location-allocation problems are used again in this section. These problems have already been discussed in Section 2.4.2 and 2.4.3 respectively.

For the (CiaS) problems, experiments were conducted on the number of circles (n_c), ranging from 2, ..., 50. The problem dimensionality n is equal to $2 \times n_c$. UMDA$_c$, EMNA$_{\text{global}}$, sEDA and sEDA-lite were implemented on this problem. After a number of trials, the value of τ is set to 0.2 for all the algorithms.

The value of η for sEDA and sEDA-lite is somewhat problem-dependent, but an approximate value can be chosen by examining the nature of the variables. Hence, the value of η was set to 0.2. The algorithm stopped after 2E+06 number of function evaluations or if the difference between the best fitness value and the global optimum was 1E-04. The population size of all the algorithms was set to 2000 except sEDA, where the population was the 50 times multiple of the dimensionality of the problem. Here the main idea for using the number of population size equals to 50 times multiple of the dimensionality of the problem is to keep approximately same number of function evaluations for sEDA with a fair amount of number of generations. The results were computed based on 25 different trials.

The 51-Customer location-allocation problem is described in Section 2.4.3. For the 51-Customer problems, experiments were conducted where the number of facilities equaled 5, 10, 15, 20, 25 and 35. UMDA$_c$, EMNA$_{\text{global}}$, sEDA and sEDA-lite were compared on these problems. The algorithms stopped after $10000 \times n$ number of function evaluations. The population size of all the algorithms was set to 2000 except sEDA, where the population was 50 times the dimension of the problem. The value of τ for all the algorithms was 0.3, whereas the value of η was 0.1 and 0.1 for sEDA and sEDA-lite respectively, since the correlation between problem variables was small here (refer to Table 4.1). The results were based on 25 repeated trials.

6.9 Results

6.9.1 Artificial Test Problems

In the present research, the results of sEDA-lite are compared with the values of UMDA$_c$, EEDA and EDA-MCC which are taken from [41]. The results of EMNA$_{\text{global}}$ are not considered, since it has been seen that, the performance of EMNA$_{\text{global}}$ is not performing better than the other 3 algorithms in the test problems. The comparative results between UMDA$_c$, EEDA, EDA-MCC and sEDA-lite are listed in Table 6.11.

In this comparison of solution quality, sEDA is not considered because a prohibitive number of function evaluations are required with the increase in dimension. However, a restricted number of function evaluations in high dimensional problems has been used, and it has already proved that the performance of sEDA is not satisfactory.

The results of the experiments for separable problems (F_1 and F_2), show that the performance of all the algorithms for 50D and 100D are similar but EEDA is not as good as the rest of the algorithms in 100D F_2 function.

Functions from F_3 … F_{10} are non-separable problems with only a few local optima. The performance of UMDA$_c$ is relatively poor in these type of functions, which have already been shown in the Table 6.11. The performance of EDA-MCC is significantly better than the rest of the algorithms on problems F_3 and F_5. Since the offset values are generated randomly, the solutions of F_4 and F_6 for EEDA and EDA-MCC are not considered here because of the unavailability of the code for these functions. The results of UMDA$_c$ have been computed on these functions and compared with sEDA-lite for the same offset values. It is observed that sEDA-lite is performing better than UMDA$_c$. The results of EDA-MCC and sEDA-lite are performing better than the UMDA$_c$ in 50D F_7 and F_8 functions. The solution comparison Table 6.11 shows that EEDA is performing better in 50D F_{10} function.

Overall, from functions F_3 to F_{10}, the performance of UMDA$_c$ is not satisfactory. In addition, sEDA-lite is performing better than UMDA$_c$, and in some cases, the performance of sEDA-lite is similar to the EDA-MCC.

Functions from F_{11} to F_{13} are multimodal functions. In these functions EDA-MCC and EEDA are not performing well. The performance of UMDA$_c$ and sEDA-lite are similar for function F_{11}; however, in the functions F_{12} and F_{13}, the performance of sEDA-lite exceeds UMDA$_c$.

The above discussion may be summarized as follows, the performance of sEDA-lite for separable problems is similar to all other algorithms and it is even better than EEDA in the 100D F_2 function. In non-separable functions, the performance of EDA-MCC exceeds that of sEDA-lite. sEDA-lite performs better than UMDA$_c$ and EEDA in some cases of non-separable functions. For multimodal problems, the performance of sEDA-lite is much better than the other algorithms in this experiment.

Table 6.11: Solution Quality Comparison. Each cell contains the Mean and Standard Deviation of the difference between the best fitness value and the global optimum. Bold font represents the best result.

Prob.	Dim	UMDA$_c$	EEDA	EDA-MCC	sEDA-lite
F_1	50	**0±0**	**0±0**	**0±0**	**0±0**
	100	**0±0**	**0±0**	**0±0**	**0±0**
F_2	50	**0±0**	**0±0**	**0±0**	**0±0**
	100	**0±0**	5.3e-10±1.4e-09(-)	**0±0**	**0±0**
F_3	50	2.6e-04±1.5e-05(+)	1.8e-08±2.4e-09(+)	**0±0(+)**	4.2e-07±2.7e-08
	100	2.6e-02±8.3e-02(-)	1.5e-03±8.5e-04(+)	**0±0(+)**	2.5e-06±4.2e-06
F_4	50	4.1e+01±2.3e+00(+)	1.4e-05±6.8e-05*	0±0*	**3.6e+01±2.1e+00**
	100	5.3e+01±2.5e+00(+)	8.1e+00±1.4e+00*	0±0*	**4.8e+01±2.8e+00**
F_5	50	1.5e+01±4.1e+00(+)	2.4e-02±3.7e-03(+)	**0±0(+)**	1.0e+01±4.0e+00
	100	1.3e+02±2.7e+01(-)	3.8e-01±4.7e-02(+)	**0±0(+)**	1.2e+02±2.1e+01
F_6	50	6.5e+01±1.4e+01(+)	1.0e-01±1.2e-02*	0±0*	**3.8e+01±1.1e+01**
	100	1.1e+03±1.6e+02(+)	7.2e+00±7.9e-01*	0±0*	**8.6e+02±1.6e+02**
F_7	50	4.8e+01±3.4e-02(+)	5.0e+01±9.2e+00(+)	**4.7e+01±2.1e-01(-)**	**4.7e+01±2.4e-02**
	100	9.7e+01±6.4e-02(-)	9.7e+01±3.7e-01(-)	**9.6e+01±7.5e-02(+)**	9.7e+01±3.1e-02
F_8	50	4.1e+02±9.1e+02(+)	5.2e+02±1.0e+03(+)	**4.8e+01±1.5e-01(-)**	**4.8e+01±1.4e+00**
	100	9.3e+02±3.1e+03(-)	4.4e+04±4.4e+04(+)	**9.6e+01±1.3e-01(+)**	1.1e+02±2.8e+01
F_9	50	4.3e+07±4.1e+06(+)	4.1e+06±1.4e+06(+)	**3.6e+06±1.5e+06(+)**	4.0e+08±4.6e+07
	100	4.3e+07±3.1e+06(+)	2.2e+07±3.7e+06(+)	**9.6e+06±2.5e+06(+)**	2.5e+09±3.7e+08
F_{10}	50	4.9e+03±1.8e+02(+)	**2.0e+03±2.0e+03(+)**	3.1e+03±3.4e+02(+)	4.5e+03±1.6e+02
	100	5.9e+03±4.3e+02(+)	4.4e+03±6.0e+02(+)	**1.9e+03±3.6e+02(+)**	5.3e+03±3.4e+02
F_{11}	50	**0±0(-)**	3.1e+02±1.3e+01(+)	2.9e+02±1.4e+01(+)	**0±0**
	100	**0±0(-)**	7.3e+02±1.5e+01(+)	7.5e+02±1.6e+01(+)	**0±0**
F_{12}	50	2.1e+00±9.5e-01(+)	3.1e+02±1.7e+01(+)	3.0e+02±1.4e+01(+)	**1.5e+00±9.9e-01**
	100	8.6e+00±2.1e+00(+)	7.3e+02±2.5e+01(+)	7.4e+02±2.3e+01(+)	**0±0**
F_{13}	50	7.8e+00±8.3e-01(+)	2.7e+01±1.1e+00(+)	2.6e+01±9.2e-01(+)	**5.9e+00±5.4e-01**
	100	1.5e+01±2.0e+00(+)	3.8e+01±2.6e+01(+)	6.5e+01±1.6e+00(+)	**1.1e+01±7.3e-01**

+ sign, the value of t-test (2 tailed)>0.05, indicates statistically significant difference when compared with sEDA-lite.
- sign, the value of t-test (2 tailed)<0.05, indicates no statistically significant difference when compared with sEDA-lite.

Values with "*": These values are directly quoted from [41] and it is incomparable to the other algorithms due to the random values of the offset

6.9.2 Results for Circle in a Square (CiaS) Problems

Here UMDA$_c$, EMNA$_{\text{global}}$, sEDA and sEDA-lite are compared with the number of circles ranging from 2, ..., 50, over a fixed number of function evaluations.

The performance of the algorithms on the CiaS problems is presented in Figure (6.2a). The x-axis denotes the problem size (n_c) while the y-axis is a performance ratio given by $d_n/f(x_n)$, where d_n is the known global optimum and $f(x_n)$ is the solution found by the algorithm.

The performance of sEDA, sEDA-lite and UMDA$_c$ are difficult to distinguish in this figure because the lines are very close to each other. Hence to clearly visualize the performance, the zoomed version of the figure is considered, where the y-axis of the figure is scaled in between 0.9 to 1.5 presented in Figure (6.2b).

In Figure (6.2a) clearly shows that, EMNA$_{\text{global}}$ is not performing better than UMDA$_c$ and sEDA-lite. Figure (6.2b) shows that up to n_c=23, sEDA is performing better but when the value of n_c increases, the performance of sEDA drops dramatically. The shape of the curve is a reflection of the fixed budget for functions evaluations imposed in the experiments. It is likely that the algorithm is no longer being given enough time to converge. The algorithm would be expected to perform better on larger problems if more function evaluations were given. In contrast, sEDA-lite seems to perform much better than sEDA, when n_c is greater than 23, with a smaller budget of function evaluations. In addition to this, sEDA-lite is performing slightly better than UMDA$_c$.

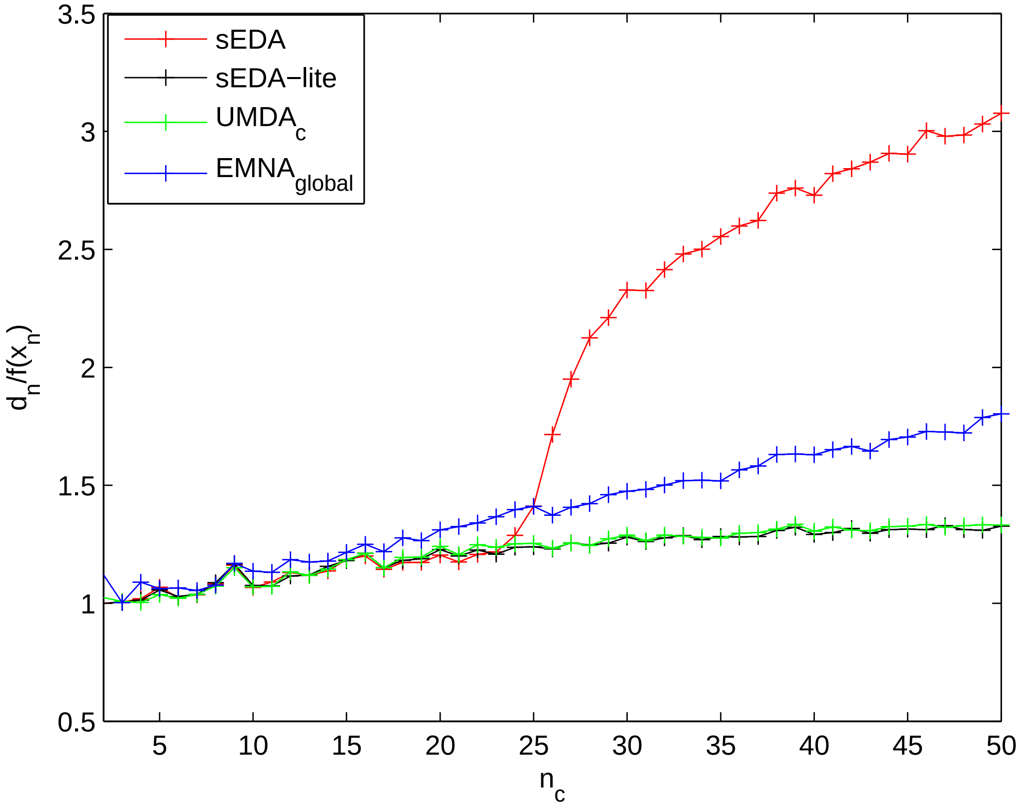
Now, from the analysis coming from Chapter 5, it suggests that, with limited covariance matrix, the modeling parameter used in sEDA helps in the better performance in CiaS problem, and here the experiments proved that, sEDA in low dimensional problems and sEDA-lite in high dimensional problems are performing better than standard Gaussian EDAs.

6.9.3 Results for 51-Customer Location-Allocation Problem

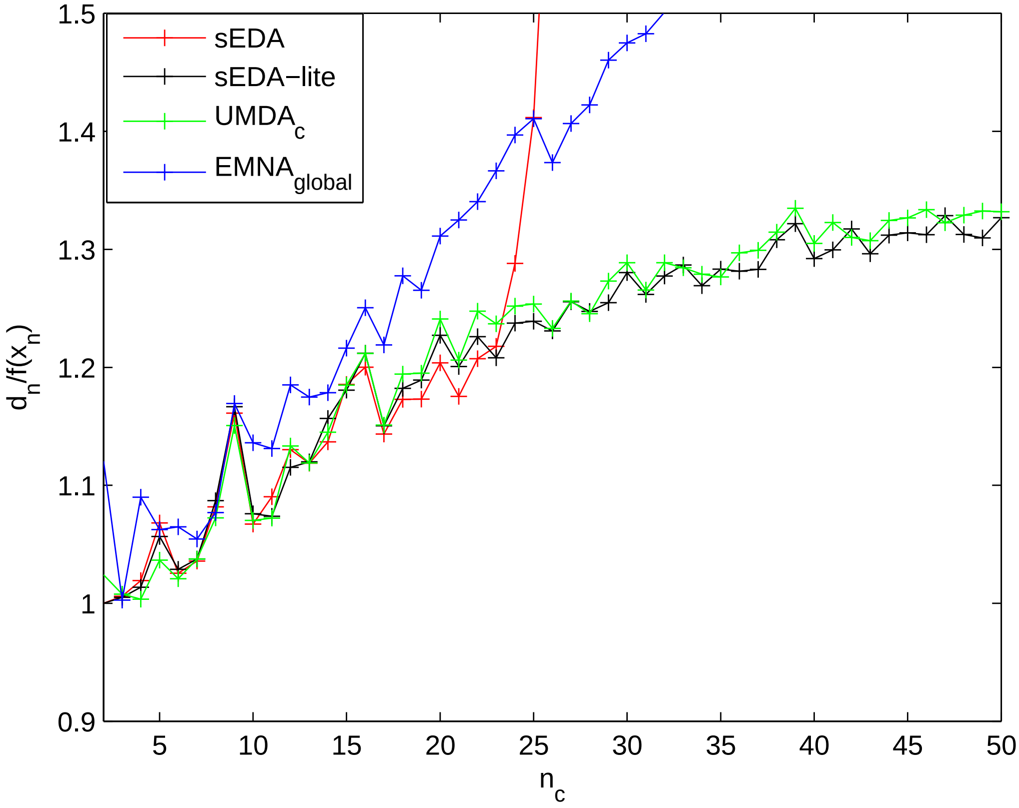
Table 6.12 shows the best performance results of UMDA$_c$, EMNA$_{\text{global}}$, sEDA and sEDA-lite on a fixed number of function evaluations over 25 runs. In low dimensional problems (e.g., when the number of facilities equals 5), the performance of sEDA is better than the rest of the algorithms (except sEDA-lite). However, the results of UMDA$_c$ is better when the number of facilities equals 10. When the number of facilities in the problem increases, sEDA-lite is performing better than the other algorithms.

Hence, in the 51 Customer problem, sEDA-lite is performing better when the number of facilities increases. The performance of EMNA$_{\text{global}}$ is inferior across these problems.

In this section of experiments on location-allocation problem, proved that sEDA in low-dimensional and sEDA-lite in high-dimensional problem are performing better .



(a)



(b)

Figure 6.2: (6.2a): Median Performance of UMDA$_c$, EMNA$_{\text{global}}$, sEDA and sEDA-lite on the CiaS problem. (6.2b): Median Performance of UMDA$_c$, EMNA$_{\text{global}}$, sEDA and sEDA-lite on the CiaS problem over the range in between 0.9 to 1.5 in the y-axis.

Table 6.12: Solution quality comparison for the 51 Customer Location-Allocation Problem. Bold font represents the best result.

Facilities	Algorithms	Best Values
5	UMDA$_c$	72.6875±0.5509(+)
	EMNA$_{\text{global}}$	77.4537±3.6498(+)
	sEDA	**72.5598 ±0.5910**
	sEDA-lite	72.5416±0.5430(-)
10	UMDA$_c$	**43.4009±0.8228**
	EMNA$_{\text{global}}$	52.5391±2.7525(+)
	sEDA	47.9302±3.8087(+)
	sEDA-lite	43.6140±0.7635(+)
15	UMDA$_c$	29.2951±0.9477(+)
	EMNA$_{\text{global}}$	42.9912±3.1708(+)
	sEDA	49.3763±3.2235(+)
	sEDA-lite	**28.9517±1.0797**
20	UMDA$_c$	21.1353±0.4938(+)
	EMNA$_{\text{global}}$	36.2924±2.5589(+)
	sEDA	48.5540±9.1548(+)
	sEDA-lite	**20.8893±0.3211**
25	UMDA$_c$	14.6532±0.7051(-)
	EMNA$_{\text{global}}$	34.4017±2.7721(+)
	sEDA	50.0931 ±16.5150(+)
	sEDA-lite	**14.2182±0.4920**
30	UMDA$_c$	10.1039±0.9703(+)
	EMNA$_{\text{global}}$	31.2373±2.1729(+)
	sEDA	49.7958±20.0151(+)
	sEDA-lite	**9.5921±0.5381**
35	UMDA$_c$	7.3644±0.6049(-)
	EMNA$_{\text{global}}$	29.9169±2.2931(+)
	sEDA	47.7707±20.7167(+)
	sEDA-lite	**7.2460±0.7566**

+ sign, the value of t-test (2 tailed)>critical values of t(.05), indicates statistically significant difference when compared with best results

- sign, the value of t-test (2 tailed)<critical values of t(.05), indicates no statistically significant difference when compared with best results

6.10 Summary

This Chapter analyzed classical EDAs over continuous variables in terms of dependency of variables and modeling and presented numerical issues as well as common difficulties with respect to performance of the model. Using different techniques and theoretical analysis, a Gaussian-based continuous EDA, called sEDA, was proposed, using a mechanism to control the amount of covariance parameters estimated within the Gaussian model, which clearly gives an idea about the number of dependent variables and importance of these variables in a problem. By adopting a simple variable screening technique from experimental design and an idea inspired by the Pareto-front in multi-objective optimization, sEDA screens out the important-dependent variables, and unimportant-independent variables. A simple implementation of this framework, sEDA, has been experimentally compared with UMDA$_c$ and EMNA$_{\text{global}}$. While considering artificial test functions, sEDA showed significantly better performance in most of the cases. Most of the results are better primarily due to better selection of variables for covariance modeling. The role of η is significant if the nature of variables in a problem is known beforehand. The main advantage of sEDA is not only finding a good solution with a lesser number of population, but it also gives an idea of variable dependency and the importance of the problem. Moreover, it doesnot show ill-conditioned covariance matrix while modeling.

In addition to this, sEDA used a large number of function evaluations in high dimensional problems. To overcome this issue, sEDA-lite was proposed in this paper. A simple implementation of this framework, sEDA-lite, has been experimentally compared with UMDA$_c$, EEDA, EDA-MCC, EMNA$_{\text{global}}$ and sEDA in various problems. In artificial multimodal test problems, the performance of sEDA-lite is better than UMDA$_c$, EEDA, EDA-MCC, whereas for real world high dimensional problems, its performance is generally better than UMDA$_c$, EMNA$_{\text{global}}$ and sEDA. The main purpose of these experiments was to demonstrate that a screening mechanism can be effectively incorporated into an EDA and used to control the covariance modeling, providing results competitive with existing techniques. Now, it has been seen that, the experiments on this chapter supports the analysis of Chapter 5.

Chapter 7

Conclusion

The focus of the thesis has been analyzing and understanding the properties of continuous optimization problems and Estimation of Distribution Algorithms (EDA) model variables. The correlation coefficient, the eigenanalysis and the Morris method have been used to analyze and better understand the correlation, key structural and important variables of the problem and the model variables; these adopted methodologies also being adaptable to a wider range of problems. A part of this investigation has resulted in the development of a new type of EDA framework called screening Estimation of Distribution Algorithm (sEDA) and its modified form for solving high dimensional problems, called sEDA-lite.

Chapter 2 provided the background of continuous optimization problems and Estimation of Distribution Algorithms (EDAs). The type of EDAs and the issues related to EDAs have been discussed. Researches undertaken for solving the EDAs' issues have been discussed. It has been shown that a number of major challenges still exist in EDAs while solving continuous optimization problems. The prior knowledge and assumption about the problem is a fundamental concept in solving a problem by any algorithm. Although a number of advanced Gaussian-based EDAs have been proposed, there has been little previous examination or attempt to analyze the behaviour of the EDA models and the relationship to the structure of the problems to which these algorithms are applied. Although general guidelines are there for adapting algorithms, it is difficult to prefer an algorithm in a given specific situation, since the preference is mainly problem dependent.

In Chapter 3, an experimental methodology is proposed to analyze the features of variables in continuous optimization problems and continuous EDAs. The first technique used is the correlation coefficient, which has been adopted from the sensitivity analysis to determine correlated variables. The second is the eigenanalysis technique where the larger vector coefficients of the first eigenvector are used for analyzing the key structural variables. The third technique is the use of the sensitivity analysis technique, namely the Morris method, for identifying the important variables to the objective function captured by the problem and the model. The results of these techniques have been presented in graphical format. These techniques are potentially very useful techniques for analyzing and comparing different continuous optimization problems and algorithms.

Chapter 4 provides the analysis of continuous optimization problems based on correlated, key structural and important variables. This helps to visualize the properties of the problems. Although some problems seem to be similar but they are different in other aspects. Five different categories have been formed based on the analysis and set an example of continuous optimization problem to this category. From the given number of problems, only 5 different categories have been formed, however, in this case 3 more categories are unexplored due to the limited number of functions used in the thesis. These categories have been further used by EDAs to analyze and compare to the problems.

Chapter 5 considered EMNA$_{\text{global}}$ and implemented all the categories of the problems which had already been discussed in Chapter 4. The analysis of model variables based on correlated, key structural and important variables has been done for each category. On some problems, the EDA modeled the structure correctly and performed well, for example, the Sphere and Rotated Ellipsoid function, whereas in some cases, it has been found that the analysis of EDA did not match with the analysis of problem, and the performance is also not satisfactory as for example, Rosenbrock function. However on some problems, it didn't model the structure correctly but still performed well, for example, in the Circle in a Square packing problem and the location-allocation problem. This shows that looking at the performance or best fitness curve doesn't indicate the behavior of the EDA model and how it works. In some problem classes (e.g., CiaS problem), a clear change is seen in the correlation, the key structural and important variables as the dimensionality increases, whereas for others, no significant change is observed.

In this chapter, the compatibility of EDAs with the problems has been analyzed and assigned the type of EDAs which better suits the problems.

Chapter 6 put forth a new type of Gaussian based EDAs called sEDA, which uses a mechanism to control the amount of covariance parameters estimated within the Gaussian model, which clearly gives an idea about the number of dependent variables and important of these variables in a problem. By adopting a simple variable screening technique from experimental design and an idea inspired by the Pareto-front in multi-objective optimization, sEDA screens out the important and dependent variables, and unimportant and independent variables. sEDA has been experimentally compared with different traditional EDAs. Although sEDA uses a smaller population, but it does not show any ill-conditioned covariance matrix while modeling. Choosing of this model parameters can be done by using the results which had discussed in Chapters 4 and 5. The main issues of sEDA is that, it uses a greater number of function evaluations in high dimensional problems. Hence, in order to solve this, a modified form of sEDA called sEDA-lite has been proposed, which requires less number of function evaluations than sEDA in high dimensional problems. The sEDA-lite has been compared with different EDAs. It has been seen that, the performance of sEDA is better in low dimensional problems where as the performance of sEDA-lite is significant in high dimensional problems. The main advantage of sEDA and sEDA-lite is not only finding a good solution but also gives an idea of variable dependency and importance of the problem.

Overall, these chapters give an idea about understanding and analyzing of variables in Black-Box optimization problems and Estimation of Distribution algorithms. As discussed above, Chapter 3 describes the methodologies used in this research. Although these methodologies have already been used in various statistical domain, but, it is the first time that they have been used simultaneously for problems and EDAs for analyzing different key properties of variables. In Chapter 4, these methodologies are used for identifying different properties of variables in black box optimization problems, which can be used to understand and to guide selection of the algorithm. Although, while categorizing the problems by different researchers, some problems fall in the same category. However this research shows that, with the addition of the existing categories, the problems can be further sub-categorized. For example, the Sphere and 10D Elliptical function is a unimodal and separable problem, but in this research, it shows that these two problems have different key structural and important variables. Hence, this research shows an in-depth understanding of variables in black box optimization problems.

In addition to this, it has always been an added advantage to know how model variables work in a given optimization problem. Therefore, methodologies described in Chapter 3 have been used in Chapter 5 for analyzing and understanding variables in EDAs.

The research shows that, if an EDA works perfectly well in a problem, why it did so. Is it actually capturing the right information of the problem variables during its implementation or are there other reasons not yet identified? This research, to some extend shows that, if the model variables capture the right information about a given problem, then it performs well.

Again, this research shows a possible method of solving problems where variables have medium correlation between them, as well as when it has been difficult to predict the number of key structural and important variables involved in a problem. This is reflected in Chapter 6, which uses the results of the nature of variables in both problems and EDAs, to figure out the model selection parameter. The analysis on previous chapters are useful in EDAs to choose the variables for modeling in between UMDA$_c$ and EMNA$_{\text{global}}$. This is an important outcome of this thesis, since this helps in choosing the value of the selection parameter for the algorithm to some extend. This is manifested in the form of sEDA and sEDA-lite in this thesis.

The future work of the thesis will be to verify the universal scope of these methodologies, which has been accomplished by implementing these methodologies on a large set of problems, as well as in some other algorithms. In addition to this, it has been used in different types of Gaussian EDAs, which has the model selection parameter.

Bibliography

[1] B. Addis, M. Locatelli, and F. Schoen. Disk packing in a square: a new global optimization approach. *INFORMS Journal on Computing*, 20(4):516–524, 2008.

[2] C.H. Aikens. Facility location models for distribution planning. *European Journal of Operational Research*, 22(3):263–279, 1985.

[3] E. Alpaydin. *Introduction to machine learning*. MIT press, 2004.

[4] N. Altay and W. G. Green III. OR/MS research in disaster operations management. *European Journal of Operational Research*, 175(1):475–493, 2006.

[5] R. Armañanzas, I. Inza, R. Santana, Y. Saeys, J. Flores, J. Lozano, Y. Van de Peer, R. Blanco, V. Robles, C. Bielza, et al. A review of Estimation of Distribution Algorithms in Bioinformatics. *BioData Mining*, 1(6):1–12, 2008.

[6] R. Armañanzas, Y. Saeys, I. Inza, M. Garcia-Torres, C. Bielza, Y. Van de Peer, and P. Larranaga. Peakbin selection in mass spectrometry data using a consensus approach with Estimation of Distribution Algorithms. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 8(3):760–774, 2011.

[7] T. Bäck. *Evolutionary Algorithms in theory and practice*. Oxford Univ. Press, 1996.

[8] T. Bäck, F. Hoffmeister, and H. Schwefel. A survey of Evolution Strategies. In *Proceedings of the 4th International Conference on Genetic Aalgorithms*, pages 2–9, 1991.

[9] T. Bäck and Z. Michalewicz. Test Landscapes. *Handbook of Evolutionary Computation, pages B*, 2:14–20, 1997.

[10] T. Bäck and H. Schwefel. An overview of Evolutionary Algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[11] M.A. Badri. Combining the analytic hierarchy process and goal programming for global facility location-allocation problem. *International Journal of Production Economics*, 62(3):237–248, 1999.

[12] S. Baluja. Population-based incremental learning. Technical report, CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, 1994.

[13] R.S. Barr, B.L. Golden, J.P. Kelly, M.G.C. Resende, and W.R. Stewart Jr. Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1(1):9–32, 1995.

[14] J.E. Bell and P.R. McMullen. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1):41–48, 2004.

[15] E. Bengoetxea, T. Miquélez, P. Larranaga, and J.A. Lozano. Experimental results in function optimization with EDAs in continuous domain. In *Estimation of Distribution Algorithms*, pages 181–194. Springer, 2002.

[16] F.P. Bernardo, E.N. Pistikopoulos, and P.M. Saraiva. Integration and computational issues in stochastic design and planning optimization problems. *Industrial & Engineering Chemistry Research*, 38(8):3056–3068, 1999.

[17] C. Blum and A. Roli. Metaheuristics in combinatorial optimziation: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.

[18] D.W. Boll, J. Donovan, R.L. Graham, and B.D. Lubachevsky. Improving dense packings of equal disks in a square. *The Electronic Journal of Combinatorics*, 7(1):46, 2000.

[19] P.A.N. Bosman and J. Grahl. Matching inductive search bias and problem structure in continuous Estimation of Distribution Algorithms. *European Journal of Operational Research*, 185(3):1246–1264, 2008.

[20] P.A.N. Bosman, J. Grahl, and D. Thierens. Enhancing the performance of maximum–likelihood Gaussian EDAs using anticipated mean shift. *Parallel Problem Solving from Nature–PPSN X*, pages 133–143, 2008.

[21] P.A.N. Bosman, J. Grahl, and D. Thierens. AMaLGaM IDEAs in noisy black-box optimization benchmarking. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pages 2351–2358. ACM, 2009.

[22] P.A.N. Bosman and D. Thierens. Continuous iterated density estimation evolutionary algorithms within the IDEA framework. In *Genetic and Evolutionary Computation Conference (OBUPM Workshop)*, 2000. (review copy).

[23] P.A.N. Bosman and D. Thierens. IDEAs based on the normal kernels probability density function. Technical Report 2000-11, Department of Computer Science, Utrecht University, (ftp://ftp.cs.uu.nl/pub/RUU/CS/techreps/CS-2000/2000-11.ps.gz), March 2000.

[24] P.A.N. Bosman and D. Thierens. Adaptive variance scaling in continuous multi-objective Estimation of Distribution Algorithms. In *Genetic and Evolutionary Computation Conference (GECCO'07)*, pages 500–507. ACM Press, 2007.

[25] S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[26] J. Brimberg, P. Hansen, N. Mladenović, and E.D. Taillard. Improvement and comparison of heuristics for solving the uncapacitated multisource Weber problem. *Operations Research*, 48(3):444–460, 2000.

[27] F. Campolongo, J. Cariboni, and A. Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental Modelling & Software*, 22(10):1509–1518, 2007.

[28] I. Castillo, F.J. Kampas, and J.D. Pintér. Solving circle packing problems by global optimization: numerical results and industrial applications. *European Journal of Operational Research*, 191(3):786–802, 2008.

[29] P. Chandra and M.L. Fisher. Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503–517, 1994.

[30] P.C. Chen, P. Hansen, B. Jaumard, and H. Tuy. Solution of the multisource Weber and conditional Weber problems by d.-c. programming. *Operations Research*, 46(4):548–562, 1998.

[31] S.H. Chen and M.C. Chen. Addressing the advantages of using ensemble probabilistic models in Estimation of Distribution Algorithms for scheduling problems. *International Journal of Production Economics*, 141(1):24–33, 2013.

[32] J.J. Collins. Visualisation of Evolutionary Algorithms using principal Component Analysis. In *GECCO Workshop Program*, pages 99–100. Citeseer, 1999.

[33] J. Comenges and F. Campolongo. An application of sensitivity analysis to fish population dynamics. *Sensitivity Analysis. John Wiley and Sons, Ltd, Chichester*, pages 367–383, 2000.

[34] T.F. Cooley and E.C. Prescott. Economic growth and business cycles. *Frontiers of business cycle research*, pages 1–38, 1995.

[35] L. Cooper. Location-Allocation problems. *Operations Research*, 11(3):331–343, 1963.

[36] A. Costa and I. Tseveendorj. Symmetry breaking constraints for the problem of packing equal circles in a square. In *ICORES*, pages 5–10, 2012.

[37] R.A. Cropp and R.D. Braddock. The New Morris Method: an efficient second-order screening method. *Reliability Engineering & System Safety*, 78(1):77–83, 2002.

[38] J.C. Culberson. On the futility of blind search: An algorithmic view of “no free lunch”. *Evolutionary Computation*, 6(2):109–127, 1998.

[39] K.A. De Jong. *Evolutionary Computation: a unified approach.* MIT Press, 2006.

[40] N. Ding, S. Zhou, and Z. Sun. Optimizing Continuous Problems using Estimation of Distribution Algorithm based on Histogram Model. *Simulated Evolution and Learning*, pages 545–552, 2006.

[41] W. Dong, T. Chen, P. Tino, and X. Yao. Scaling up Estimation of Distribution Algorithms for continuous optimization. *Evolutionary Computation, IEEE Transactions*, 17(6):797–822, 2013.

[42] W. Dong and X. Yao. Covariance matrix repairing in Gaussian based EDAs. In *Evolutionary Computation, IEEE Congress on*, pages 415–422. IEEE, 2007.

[43] W. Dong and X. Yao. Unified eigen analysis on multivariate Gaussian based Estimation of Distribution Algorithms. *Information Sciences*, 178(15):3000–3023, 2008.

[44] Z. Drezner. Dynamic facility location: The progressive p median problem. *Location Science*, 3(1):1–7, 1995.

[45] Á.E. Eiben and M. Jelasity. A critical note on experimental research methodology in EC. In *Evolutionary Computation. Proceedings of the 2002 Congress on*, volume 1, pages 582–587. IEEE, 2002.

[46] A.E. Eiben and J.E. Smith. Multimodal problems and spatial distribution. In *Introduction to Evolutionary Computing*, pages 153–172. Springer, 2003.

[47] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*, volume 2. Springer Berlin, 2010.

[48] S. Eilon, C.D.T. Watson-Gandy, and N. Christofides. *Distribution management: mathematical modelling and practical analysis*. Griffin, London, 1971.

[49] T. El-Ghazali. *Metaheuristics: from design to implementation*, volume 74. 2009.

[50] L.J. Eshelman. Real-coded Genetic Algorithms and Interval-Schemata. *Foundations of Genetic Algorithms*, 2:187–202, 1993.

[51] A. Federgruen and P. Zipkin. A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019–1037, 1984.

[52] C.A. Floudas and P.M. Pardalos. *Encyclopedia of optimization*, volume 1. Springer, 2008.

[53] D.B. Fogel and L.C. Stayton. On the effectiveness of crossover in simulated evolutionary optimization. *BioSystems*, 32(3):171–182, 1994.

[54] A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, 2008.

[55] M. Gallagher. *Multi-layer perceptron error surfaces: visualization, structure and modelling*. PhD thesis, Dept. Computer Science and Electrical Engineering, University of Queensland, 2000.

[56] M. Gallagher. Investigating circles in a square packing problems as a realistic benchmark for continuous metaheuristic optimization algorithms. In *The VIII Metaheuristic International Conference MIC*, 2009.

[57] M. Gallagher. Beware the Parameters: Estimation of Distribution Algorithms applied to Circles in a Square Packing. *Parallel Problem Solving from Nature-PPSN XII*, pages 478–487, 2012.

[58] M. Gallagher, M. Frean, and T. Downs. In W. Banzhaf and et al., editors, *Proc. Genetic and Evolutionary Computation Conference (GECCO'99)*, pages 840–846, San Francisco, CA.

[59] M. Gallagher and B. Yuan. A general-purpose tunable landscape generator. *Evolutionary Computation, IEEE Transactions on*, 10(5):590–603, 2006.

[60] M. Gallagher and B. Yuan. A general-purpose tunable landscape generator. *Evolutionary Computation, IEEE Transactions on*, 10(5):590–603, 2006.

[61] M.D.H. Gamal and S. Salhi. Constructive heuristics for the uncapacitated continuous Location–Allocation problem. *Journal of the Operational Research Society*, pages 821–829, 2001.

[62] M.D.H. Gamal and S. Salhi. A cellular heuristic for the multisource Weber problem. *Computers & Operations Research*, 30(11):1609–1624, 2003.

[63] M. Gendreau and J.Y. Potvin. *Handbook of Metaheuristics*, volume 146. Springer, 2010.

[64] H.M. Ghaziri. A neural heuristic for the multisource Weber problem. *Infor-Information Systems and Operational Research*, 37(3):226–235, 1999.

[65] W.R. Gilks. *Markov Chain Monte Carlo*. Wiley Online Library, 2005.

[66] B.L. Golden, A.A. Assad, E.A. Wasil, and E. Baker. Experimentation in optimization. *European journal of Operational Research.*, 27(1):1–16, 1986.

[67] C. González, José A. Lozano, and P. Larrañaga. Mathematical modelling of UMDAc algorithm with tournament selection. Behaviour on linear and quadratic functions. *International Journal of Approximate Reasoning*, 31(3):313–340, 2002.

[68] J. Grahl, S. Minner, and F. Rothlauf. Behaviour of UMDAc with truncation selection on monotonous functions. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pages 2553–2559. IEEE, 2005.

[69] J.J. Grefenstette. Optimization of control parameters for genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 16(1):122–128, 1986.

[70] L. Grosset, R. LeRiche, and R.T. Haftka. A double-distribution statistical algorithm for composite laminate optimization. *Structural and Multidisciplinary Optimization*, 31(1):49–59, 2006.

[71] A. Grosso, A. Jamali, M. Locatelli, and F. Schoen. Solving the problem of packing equal and unequal circles in a circular container. *Journal of Global Optimization*, 47(1):63–81, 2010.

[72] S.L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459, 1964.

[73] N. Hansen. The CMA Evolution Strategies: A tutorial. *Vu le*, 29, 2005.

[74] N. Hansen. The CMA evolution strategy: a comparing review. In *Towards a New Evolutionary Computation*, pages 75–102. Springer, 2006.

[75] N. Hansen, A. Auger, R.. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Proceedings of the 12th annual conference companion on Genetic and Evolutionary Computation*, pages 1689–1696. ACM, 2010.

[76] N. Hansen, S. Finck, R. Ros, A. Auger, et al. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. 2009.

[77] N. Hansen, S.D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

[78] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[79] N. Hansen and R. Ros. Benchmarking a weighted negative covariance matrix update on the BBOB-2010 noiseless testbed. In *Proceedings of the 12th annual conference companion on Genetic and Evolutionary Computation*, pages 1673–1680. ACM, 2010.

[80] J.C. Helton, J.D. Johnson, C.J. Sallaberry, and C.B. Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 91(10-11):1175–1209, 2006.

[81] M. Hifi and R. M'hallah. A literature review on circle and sphere packing problems: Models and Methodologies. *Advances in Operations Research*, 2009, 2009.

[82] J.H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

[83] R. Horst, P.M. Pardalos, and H.E. Romeijn. *Handbook of global optimization*, volume 2. Springer, 2002.

[84] C.R. Houck, J.A. Joines, and M.G. Kay. Comparison of genetic algorithms, random restart and two-opt switching for solving large location-allocation problems. *Computers & Operations Research*, 23(6):587–596, 1996.

[85] D.J. Inman and R.C. Singh. *Engineering Vibration*. Prentice Hall New Jersey, 2001.

[86] I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra. Feature subset selection by Bayesian network-based optimization. *Artificial Intelligence*, 123(1):157–184, 2000.

[87] I. Inza, P. Larrañaga, and B. Sierra. Feature subset selection by Bayesian networks: a comparison with genetic and sequential algorithms. *International Journal of Approximate Reasoning*, 27(2):143–164, 2001.

[88] J.E. Jackson. *A user's guide to principal components*, volume 244. Wiley-Interscience, 1991.

[89] R. Jackson, P.T. Boggs, S.G. Nash, and S. Powell. Guidelines for reporting results of computational experiments. Report of the ad hoc committee. *Mathematical programming*, 49(1):413–425, 1990.

[90] B. Jarboui, M. Eddaly, and P. Siarry. An Estimation of Distribution Algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. *Computers & Operations Research*, 36(9):2638–2646, 2009.

[91] S. Jiang, A.K. Ziver, J.N. Carter, C.C. Pain, A.J.H. Goddard, S. Franklin, and H.J. Phillips. Estimation of Distribution Algorithms for nuclear reactor fuel management optimisation. *Annals of Nuclear Energy*, 33(11):1039–1057, 2006.

[92] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[93] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga. Regularized Continuous Estimation of Distribution Algorithms. *Applied Soft Computing*, 2012.

[94] A. D. Kenneth. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, 1975.

[95] D.M. King and J.C. Perera. Morris Method of sensitivity analysis applied to assess the importance of input variables on urban water supply yield–a case study. *Journal of Hydrology*, 2012.

[96] R.J. Koessler, F.S. Prabhakara, and A.H. Al-Mubarak. Analysis of oscillations with eigenanalysis and prony techniques. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–8. IEEE, 2007.

[97] S. Krau. *Extensions du problème de Weber, PhD. Thesis*. Ècole Polytechnique de Montrèal, 1998.

[98] S. Kravitz. Packing cylinders into cylindrical containers. *Mathematics magazine*, 40(2):65–71, 1967.

[99] R.E. Kuenne and R.M. Soland. Exact and approximate solutions to the multisource Weber problem. *Mathematical Programming*, 3(1):193–209, 1972.

[100] P. Larrañaga. A review on Estimation of Distribution Algorithms. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, pages 57–100.

[101] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña. Optimization by learning and simulation of Bayesian and Gaussian networks. Technical Report KZZA-IK-4-99, University of the Basque Country, Spain, 1999.

[102] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms : A New Tool for Evolutionary Computation*. Kluwer, 2001.

[103] B.K. Lavine, C.E. Davidson, and A.J. Moores. Genetic algorithms for spectral pattern recognition. *Vibrational Spectroscopy*, 28(1):83–95, 2002.

[104] F.G. Lobo, C.F. Lima, and Z. Michalewicz. *Parameter setting in Evolutionary Algorithms*, volume 54. Springer Verlag, 2007.

[105] M. Locatelli and U. Raber. Packing equal circles in a square: a deterministic global optimization approach. *Discrete Applied Mathematics*, 122(1):139–166, 2002.

[106] M. Los and C. Lardinois. Combinatorial programming, statistical optimization and the optimal transportation network problem. *Transportation Research Part B: Methodological*, 16(2):89–124, 1982.

[107] R.F. Love and J.G. Morris. A computation procedure for the exact solution of location-allocation problems with rectangular distances. *Naval Research Logistics Quarterly*, 22(3):441–453, 2006.

[108] B.D. Lubachevsky and R.L. Graham. Curved hexagonal packings of equal disks in a circle. *Discrete & Computational Geometry*, 18(2):179–194, 1997.

[109] S. Luke. *Essentials of metaheuristics*, volume 113. Lulu Raleigh, 2009.

[110] C.C. McGeoch. Experimental analysis of optimization algorithms. *Handbook of applied optimization*, pages 1044–1052, 2002.

[111] N. Megiddo and K.J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.

[112] M.T. Melo, S. Nickel, and F. Saldanha-Da-Gama. Facility location and supply chain management–A review. *European Journal of Operational Research*, 196(2):401–412, 2009.

[113] Z. Michalewicz and M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1):1–32, 1996.

[114] M. Mihalak. *Optimization Problems in Communication Networks*. University of Leicester, 2006.

[115] N. Mladenović, J. Brimberg, P. Hansen, and J.A. Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939, 2007.

[116] R. Morgan and M. Gallagher. When does dependency modelling help? using a randomized landscape generator to compare algorithms in terms of problem structure. In *Parallel Problem Solving from Nature, PPSN XI*, pages 94–103. Springer, 2010.

[117] R. Morgan and M. Gallagher. Using landscape topology to compare continuous metaheuristics: A framework and case study on EDAs and ridge structure. *Evolutionary computation*, 20(2):277–299, 2012.

[118] M.D. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, 1991.

[119] M. Morzyński, K. Afanasiev, and F. Thiele. Solution of the eigenvalue problems resulting from global non-parallel flow stability analysis. *Computer methods in applied mechanics and engineering*, 169(1):161–176, 1999.

[120] M. Mudrova and A. Procházka. Principal Component Analysis in image processing. In *Proceedings of the MATLAB Technical Computing Conference, Prague*, 2005.

[121] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. binary parameters. In *Parallel Problem Solving from Nature - PPSN IV*, volume 1411 of *Lecture Notes in Computer Science*, pages 178–187. Springer, 1996.

[122] K.J. Nurmela and P.R.J. Östergård. Packing up to 50 equal circles in a square. *Discrete & Computational Geometry*, 18(1):111–120, 1997.

[123] L.M. Ostresh Jr. An efficient algorithm for solving the two center Location-Allocation problem. *Journal of Regional Science*, 15(2):209–216, 1975.

[124] N. Patterson, A.L. Price, and D. Reich. Population structure and eigenanalysis. *PLoS genetics*, 2(12):190, 2006.

[125] M. Pelikan, D.E. Goldberg, and F.G. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[126] E. Pitzer and M. Affenzeller. A comprehensive survey on fitness landscape analysis. In *Recent Advances in Intelligent Engineering Systems*, pages 161–191. Springer, 2012.

[127] E. Polak. *Optimization: algorithms and consistent approximations*. Springer-Verlag New York, Inc., 1997.

[128] M.V. Ruano, J. Ribes, J. Ferrer, and G. Sin. Application of the Morris method for screening the influential parameters of fuzzy controllers applied to wastewater treatment plants. *Water Science & Technology*, 63(10):2199–2206, 2011.

[129] N. Hansen D. Büche J. Ocenasek S. Kern, S.D. Müller and P. Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms–a comparative review. *Natural Computing*, 3(1):77–112, 2004.

[130] Y. Saeys, S. Degroeve, D. Aeyels, Y. Van de Peer, and P. Rouzé. Fast feature selection using a simple Estimation of Distribution Algorithm: a case study on splice site prediction. *Bioinformatics*, 19(suppl 2):ii179–ii188, 2003.

[131] R. Sagarna and J.A. Lozano. Scatter search in software testing, comparison and collaboration with, Estimation of Distribution Algorithms. *European Journal of Operational Research*, 169(2):392–412, 2006.

[132] S. Salhi and M.D.H. Gamal. A Genetic algorithm based approach for the uncapacitated continuous location–allocation problem. *Annals of Operations Research*, 123(1):203–222, 2003.

[133] A. Saltelli, K. Chan, E.M. Scott, et al. *Sensitivity analysis*, volume 134. Wiley New York, 2000.

[134] R. Santana, P. Larrañaga, and J.A. Lozano. Side chain placement using Estimation of Distribution Algorithms. *Artificial Intelligence in Medicine*, 39(1):49–63, 2007.

[135] R. Santana, P. Larrañaga, and J.A. Lozano. Protein folding in simplified models with Estimation of Distribution Algorithms. *Evolutionary Computation, IEEE Transactions on*, 12(4):418–438, 2008.

[136] R. Santana, A. Mendiburu, N. Zaitlen, E. Eskin, and J.A. Lozano. Multi-marker tagging single nucleotide polymorphism selection using Estimation of Distribution Algorithms. *Artificial Intelligence in Medicine*, 50(3):193–201, 2010.

[137] M.P. Scaparra and M.G. Scutellà. Facilities, Locations, Customers: Building Blocks of Location Models. A Survey. Technical report, 2001.

[138] A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127, 1999.

[139] G. Sin and K.V. Gernaey. Improving the Morris method for sensitivity analysis by scaling the elementary effects. *Computer Aided Chemical Engineering*, 26:925–930, 2009.

[140] K. Smith-Miles and L. Lopes. Measuring instance difficulty for combinatorial optimization problems. *Computers & Operations Research*, 39(5):875–889, 2012.

[141] J.A. Snyman. *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*, volume 97. Springer, 2005.

[142] E. Specht. Packomania. `http://www.packomania.com`, 2012.

[143] Peter F Stadler. Fitness landscapes. In *Biological evolution and statistical physics*, pages 183–204. Springer, 2002.

[144] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL Report*.

[145] J. Sun, Q. Zhang, J. Li, and X. Yao. A hybrid Estimation of Distribution Algorithm for CDMA cellular system design. In *Simulated Evolution and Learning*, pages 905–912. Springer, 2006.

[146] P.G. Szabó and E. Specht. Packing up to 200 equal circles in a square. In *Models and Algorithms for Global Optimization*, pages 141–156. Springer, 2007.

[147] E.D. Taillard. Heuristic methods for large centroid clustering problems. *Journal of Heuristics*, 9(1):51–73, 2003.

[148] M. Teboulle. A unified continuous optimization framework for center-based clustering methods. *The Journal of Machine Learning Research*, 8:65–102, 2007.

[149] V.E. Theodoracatos and J.L. Grimsley. The optimal packing of arbitrarily-shaped polygons using simulated annealing and polynomial-time cooling schedules. *Computer Methods in Applied Mechanics and Engineering*, 125(1):53–70, 1995.

[150] A. Törn and A. Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1989.

[151] M. Wagner, A. Auger, and M. Schoenauer. Eeda: A new robust Estimation of Distribution Algorithms. 2004.

[152] L. Wang and C. Fang. An effective Estimation of Distribution Algorithm for the multi-mode resource-constrained project scheduling problem. *Computers & Operations Research*, 39(2):449–460, 2012.

[153] J. Wei. Video content classification based on 3-D eigen analysis. *Image Processing, IEEE Transactions on*, 14(5):662–673, 2005.

[154] L.D. Whitley, K.E. Mathias, S.B. Rana, and J. Dzubera. Building better Test Functions. In *Genetic Algorithms : International Computation*, pages 239–247, 1995.

[155] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52, 1987.

[156] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, 1997.

[157] A.H. Wright. Genetic Algorithms for Real Parameter Optimization. In *FOGA*, pages 205–218, 1990.

[158] T. Yamada. A network flow approach to a city emergency evacuation planning. *International Journal of Systems Science*, 27(10):931–936, 1996.

[159] B. Yuan and M. Gallagher. Playing in Continuous Spaces: Some Analysis and Extension of Population-Based Incremental Learning. In R. Sarkar et. al., editor, *Proc. Congress on Evolutionary Computation (CEC)*, pages 443–450. IEEE, 2003.

[160] B. Yuan and M. Gallagher. Experimental results for the special session on real-parameter optimization at CEC 2005: a simple, continuous EDA. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 1792–1799. IEEE, 2005.

[161] B. Yuan and M. Gallagher. On the importance of diversity maintenance in Estimation of Distribution Algorithms. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pages 719–726. ACM, 2005.

[162] B. Yuan and M. Gallagher. A mathematical modelling technique for the analysis of the dynamics of a simple continuous EDA. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1585–1591. IEEE, 2006.

[163] B. Yuan and M. Gallagher. Convergence analysis of UMDAc with finite populations: a case study on flat landscapes. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 477–482. ACM, 2009.

[164] B. Yuan, M. Orlowska, and S. Sadiq. Finding the optimal path in 3D spaces using EDAs– the wireless sensor networks scenario. In *Adaptive and Natural Computing Algorithms*, pages 536–545. Springer, 2007.

[165] Y. Zhang and X. Li. Estimation of Distribution Algorithm for permutation flow shops with total flowtime minimization. *Computers & Industrial Engineering*, 60(4):706–718, 2011.