Knowledge Discovery for Design Optimization Using Correspondence Analysis

by

Yu Guo

B.Eng., Sun Yat-sen University, 2013

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Applied Science

in the
School of Mechatronic Systems Engineering
Faculty of Applied Science

SIMON FRASER UNIVERSITY

Fall 2015

Approval

Name:	**Yu Guo**
Degree:	**Master of Applied Science**
Title:	***Knowledge discovery for design optimization using Correspondence Analysis***
Examining Committee:	**Chair:** Jason Wang Assistant Professor

G. Gary Wang
Senior Supervisor
Professor

Krishna Vijayaraghavan
Supervisor
Assistant Professor

Flavio Firmani
Internal Examiner
Lecturer
School of Mechatronic Systems Engineering

Date Defended/Approved: September 28th 2015

Abstract

In recent years, researchers have made great efforts in tackling the High-dimensional, Expensive (computationally), Black box (HEB) design problems. The high dimensionality and lack of knowledge of the problem usually demand a large number of samples for optimization, which is often impractical due to the total time required to compute the required number of expensive simulations. In this thesis, Correspondence Analysis (CA) is introduced to discover as much information as possible about the black box to minimize the number of samples. The discovered information such as the promising subdomains, important variables, and symmetric variables is used to assist the resampling in an existing optimization algorithm. While being independent from the optimization algorithm, the approached method is applied to the Trust Region based Mode Pursuing Sampling (TRMPS2), a global optimization method developed for HEB problems. The CA based TRMPS2 method (CA_TRMPS) is shown to yield better optima with higher efficiency than TRMPS2. Tests on mathematical benchmark functions and application to a real-world engineering problem show the promise of the proposed approach.

Keywords: Computationally-expensive Black box; Large-scale; HEB; Correspondence Analysis; Singular Value Decomposition; Optimization; Visualization

I would like to dedicate this thesis and everything I do
to my beloved parents and boyfriend

Acknowledgements

First, I sincerely thank my senior supervisor Dr. G Gary Wang. He introduced and guided me through several brand new areas and helped me to find my interests. He cares very much about his students, so as one of his students, he provided me with a lot of opportunities and supports during my study. As an international student, who lacks of professional training on writing, I am so lucky to be his student. He pointed this out at the very early stage of my study and pushed me to learn how to write. He encouraged me to face my own weakness and practice more to improve myself. He also spent a lot of time on revising my writing over and over again. Now, I am more comfortable with writing than before and it all credits to his endless supports.

Second, I would like to thank Dr. Krishna Vijayaraghavan, Dr. Jason Wang and Dr. Flavio Firmani, for taking the time to serve on my supervisory committee, committee chair and examiner. Thank you all for spending time to read and criticize my thesis, which help me a lot in polishing my work.

Finally, I would like to thank all the people in my lab. They have been so kind to me, which made me very easy to adjust to the new environment. I enjoyed working with all of you and I have learned a lot while we were discussing with each other.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ANN	Artificial Neural Network
ANOVA	Analysis of Variance
ARSM	Adaptive Response Surface Method
BPL	Bone Phosphates of Lime
CA	Correspondence analysis
CA_TRMPS	Correspondence Analysis based TRMPS2
CT	Contingency Table
FA	Factor Analysis
HDMR	High Dimensional Model Representation
HEB	High-dimensional, Expensive (computationally), Black box
HSCT	High-Speed Civil Transport
LHS	Latin Hypercube Sampling
MDS	Multidimensional Scaling
PCA	Principal Component Analysis
RBF	Radial Basis Function
RDP	Relative Distance Plane
SVD	Singular Value Decomposition
TRMPS2	Trust Region based Mode Pursuing Sampling
VDS	Visual Design Steering

Chapter 1. Introduction

1.1. Overview

Optimization [1] is the research field that studies the design of algorithms for finding the best solutions to problems. The global optimization [2] utilizes mathematical and statistical methods to find the global optimal according to certain criteria. Optimization is not a standalone mathematical topic, since it is always concerned with real cases in engineering practices. In the field of engineering, optimization is often referred to as design optimization [3], which means to achieve design goals with optimization techniques.

1.1.1. Black-box Optimization

The field of black-box optimization [4] is a subset of optimization, in which the black-box problem, or system, is only viewed in terms of its inputs and outputs, with no or little knowledge of the internal workings. The goal of a black box optimization problem is usually to find out either the mechanism inside the black box or the optimal outputs of the system, and to make predictions on further inputs.

Many engineering problems can be categorized as black-box problems. Many of these problems in real practices can be more difficult than the textbook cases. Real world engineering design problems often contain many variables and a large number of constraints, and evaluating the output can be extremely computationally expensive. For example, the wing configuration design of a high-speed civil transport (HSCT) aircraft [5] includes 26 variables, four objectives, and four technical constraints. Running the simulation to evaluate the design is very time consuming. Such problems are the so-called High-dimensional, Expensive (computationally), and Black box (HEB) problems

[6]. Apart from HSCT mentioned in the last paragraph, HEB problems widely exist in engineering design such as the ones archived in Ref. [7]-[9].

In HEB problem, the high dimensionality increases the size and complexity of the search space. This property makes modeling and searching of the space very challenging, so that HEB problems requires more sample points. On the other hand, the number of sample points cannot be too large due to the high computational cost of each sample point. Therefore, in this thesis, we want to find a new way to discover some information of the black-box problems with existed sample points, in order to help to narrow down the search space so that the total number of sample points will be reduced. In this work, we introduced Correspondence Analysis algorithm as a data analysis tool to reveal the structure of black-box problems. It will be discussed in detail in the following sections.

1.1.2. Correspondence Analysis

Correspondence analysis [10] is a multivariate statistical technique designed to explore the associations between sets of categorical variables. As an exploratory data analytic technique [11], it is very useful for revealing patterns within datasets, such as the correspondence between different categories and the associations of values within a category. Mathematically, it is usually done on a two-dimensional contingency table (CT) [12], and reveals the internal information of rows and columns. The usefulness and flexibility of correspondence analysis makes itself a popular method for data analysis. In recent years, it has been applied in various research fields, including social science, biological statistics, medical statistics, and criminology [13].

Unlike regression modeling [14] or other optimization techniques, there is no need to reconstruct a specific model from a large data set in correspondence analysis. Instead, it only requires minimal data without any prior knowledge of the underlying black-box function. In addition, correspondence analysis can usually work well with small data sets. The characteristics of correspondence analysis make itself a useful tool for solving HEB optimization problems. It can be used to find out the correspondence between input variables and outputs, and then assist the global optimization process for a black-box function. This research will further exploit its capability of data analysis and apply it in optimization problems. The way of interpolating the plot of CA is also unique and can reveal much more related information in the optimization context.

1.2. Scope

The scope of the work presented is limited to data analysis and visualization [15] to assist engineering design optimization. The focus of this research is on mining and visualizing the useful information in the design space [16] and their relations to performance spaces [17] for optimization problems. By applying correspondence analysis, which is a popular data analysis technique, relationships among design space and between design space and performance space will be visualized at the first stage of optimization. Scatter plot [18] is the central representation used to illustrate this task.

Visualization is applied to the early stage of the optimization when little information is known about the HEB problem. It allows much less sampling points for the optimization algorithm. The improved CA_TRMPS optimization algorithm, which is developed based on trust region based mode pursuing sampling (TRMPS2) [19] will be presented in this thesis as an example to show the improvement of the optimization algorithm after using the information discovered by CA. However, besides TRMPS2, this method can be applied to various optimization methods in the early stage. The focus is on visualizing the effect of many variables on a single objective without constraints. Thus multi-objective and multidisciplinary optimization is beyond the scope of this thesis.

1.3. Research Goals

The goal of this research is to develop a new method for HEB optimization problems, in support of finding out the subdomains that are most likely to produce the optimal values. The method will make use of correspondence analysis to help discover the symmetry and the importance among different dimensions of the design space, as well as visualize and exploit the data structure. The method will give out suggestions of the subdomains that are most likely to produce optimal values. The suggestions can be used to modify and refine the optimization process, so that a much faster convergence rate can be achieved. Moreover, this research will test this method in various idealized and practical engineering problems, and will also compare it with current optimization method.

The objectives of this new method are threefold. First, we want it to decide which input variables are more important, and which are less important. Second, we want to find out the promising subdomain for each of the variables, from which the optimal output is more likely to appear. Finally, we want it to discover the symmetric variables [20]. The definition of symmetric variables is that if a group of variables are symmetric, any permutation of them will result in the same function. By identifying the symmetric variables, the same value can be applied to all of the variables for the sampling points. This can reduce large number of combination of potential values for each variable. In a sense, it achieves similar results as the dimension reduction techniques. Therefore, it can save a large number of sampling points when comes to high dimension.

1.4. Structure

The thesis consists of six chapters. The flowchart in Figure 1.1 shows the basic structure of this thesis. The first chapter is the introduction chapter, providing an overview of the whole research. The second chapter is a literature survey. It reviews the current situation of HEB optimization problems, and gives an introduction to correspondence analysis. Next, in the third chapter, the methodology of CA technique is described in detail. How a modified version of CA is applied in the context of engineering optimization is provided with details of how to construct the contingency table and explain the new way to interpret the plot. In the plot, information such as promising subdomain, significant variables and symmetric variables can be identified intuitively. The proposed technique is tested on simple functions, benchmark functions, and real engineering problems to demonstrate its correctness. The convergence of the algorithm is also tested and shown in this chapter. An automatic clustering algorithm to assist the discovery of the symmetric variables is introduced and tested on two functions, which are presented in Chapter 4 examples. In the following chapter, CA is applied to an existed optimization algorithm to verify that the discovered information can help to increase the efficiency and accelerate the convergence rate of the algorithm. By using the same amount of samples, the results of the improved algorithm are compared with the results of original algorithm to show the improvement. In Chapter 6, the proposed method is successfully applied to a practical engineering problem – phosphate production scheduling problem by revealing the properties of the problem. The last chapter provides the conclusion and the potential feature work that could be developed based on this thesis.

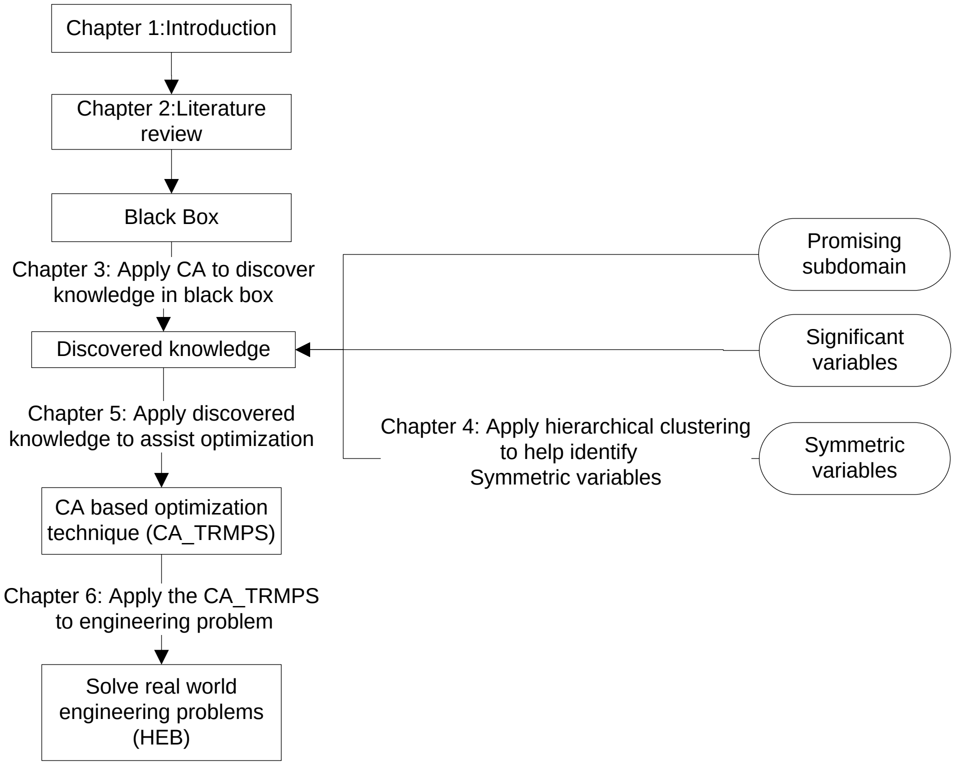

Chapter 1:Introduction
Chapter 2:Literature review
Black Box
Chapter 3: Apply CA to discover knowledge in black box
Discovered knowledge
Chapter 5: Apply discovered knowledge to assist optimization
CA based optimization technique (CA_TRMPS)
Chapter 6: Apply the CA_TRMPS to engineering problem
Solve real world engineering problems (HEB)
Promising subdomain
Significant variables
Symmetric variables
Chapter 4: Apply hierarchical clustering to help identify Symmetric variables

Figure 1.1 Structure for the thesis

Chapter 2. Literature Survey

This chapter provides an introduction to the current situation of HEB problems and the technique of correspondence analysis. The chapter generally contains three parts. The first part gives an overview of current methods applied in HEB optimization. The second part introduces correspondence analysis and reviews its history. The third part will discuss the possibility and the advantage of applying correspondence analysis in HEB optimization.

2.1. Challenges for HEB problems

The major challenges of HEB problems come from the large scale of dimensions and high computational expense for function evaluation within the black box [21][22]. On one hand, the high dimensionality of inputs and outputs requires sampling with exponential difficulty. For a simulation function with *n* variables, if we sample *k* points in each variable, k^n total sampling points will be needed. Hence, k^n total runs of simulations will be required. However, on the other hand, the high computational expense requires that the number of simulations (function calls) should be minimized, and it is usually expensive to perform simulation on a large data set. In most cases, HEB problems are expected to be solved with only small samples. Therefore, the key to HEB problems is, “how can we solve them with the least sampling data, though most of the current solutions still require a reasonably large data set?”

2.2. Current methods for HEB Optimization

2.2.1. Meta-modeling

To tackle the HEB problems, researchers try to simplify the problem from different aspects. Some of them try to reduce the computational cost by building a meta-model to approximate the black-box function. Meta-models can be polynomial functions [23], Radial Basis Function (RBF) [24], and Kriging [25] or other functions. The samples generated from a meta-model are called cheap points because the function values are calculated much faster on the meta-model than on the actual simulation. On the contrary, the exploring inputs whose outputs values are obtained by the actual black box are called expensive points. The basic idea of meta-modeling is to approximate the black box function with a meta-model by iteratively improving the meta-model. Each time, we generate cheap points on an approximated model to find out the meta-model optimal point, and then re-sample the expensive points around the meta-model optimal point and test them on the black box. By comparing the values of cheap points and expensive points, we can improve the meta-model, and then repeat the previous step. In this way, the total number of runs for sampling expensive points is reduced.

High Dimensional Model Representation (HDMR) [26][27] is one of the famous techniques used for high dimensional model. However, if the function has complicated variable correlations, HDMR still requires a large number of samples to obtain a reasonable model. To deal with the challenges from high dimensionality, several methods are applied, including reducing design space [28][29], screening significant variables [30]-[32], decomposing design problems into sub-problems[33][34], mapping[35][36], and visualizing the variable/design space [36]. These strategies tackle from different aspects of the so called “curse of dimensionality.” Some of them may overlap and are thus not completely independent.

2.2.2. Dimensionality reduction

While meta-modeling tries to handle the HEB problem by replacing the expensive calculations with much simpler meta-models, some other approaches seek for the possibilities of reducing the dimensionality of the black box. After the dimensionality of the black box is reduced, the problem can be further solved with regular methods.

One of the methods for dimensionality reduction is screening [30]-[32]. Its goal is to retain the important dimensions and discard the unimportant ones. Screening process such as Analysis of variance (ANOVA) [38], Principal component analysis (PCA) [39] and sensitivity analysis [40] identifies and retains important input variables and interaction terms, whereas removes less important ones or noises in the problems of interest so that the complexity or dimensionality of the problems is reduced to save computational cost. Screening likely pays a price of losing modeling accuracy of problems because of removed dimensionalities. It may as well miss the important regions.

Besides screening, mapping is another method that aims at dimensionality reduction. Rather than directly discarding the unimportant dimensions as screening, it transforms a set of correlated variables into a smaller set of uncorrelated variables that retain most of the original information. The most popular mapping methods include PCA [41], Relative Distance Plane (RDP) mapping [42], and Artificial Neural Network (ANN) [43]. The methods preserved the inherent data “structure” in a lower-dimensional space even though part of the information was lost. But it seems that no one has examined the possibility of mapping optimization problems from an original higher-dimensional space to a lower-dimensional space while preserving the optimum. If this is doable, both the problem size and the optimization complexity can be reduced simultaneously. The challenge is how to ensure the optimum obtained in the lower-dimensional space is the true optimum for the higher-dimensional space.

2.2.3. Space Reduction

Apart from meta-modeling and dimensional reduction techniques, there is another approach named space reduction that tackles the HEB problems in a totally different aspect. Space reduction means shrinking a design space during the optimization process so that the optimization effort is reduced in the optimization subdomain. A common space reduction approach starts with sampling a limited number of points and evaluating function values at these points. Then the design space is reduced based on feedback information from modeling on these sample points. The revised design space is again segmented using smaller increments, and the objective function is determined for new points. In this way, the focus of modeling can be in a more attractive region. The method such as adaptive response surface method (ARSM) [44], the fuzzy clustering based approach [45] and a rough set based method space [46] has been developed in the past years. The advantage of these kinds of technique is that it can reduce the effort on modeling and optimization, but at the same time it may miss the global optima or important subdomain.

2.2.4. Visualization

Finally, visualization is a user-interactive way to present a problem in a visual form, allowing users to get insight into the problems, find key trends and relationships among variables in a problem, and make decisions by interacting with the data. Winer and Bloebaum [47][48] developed a Visual Design Steering (VDS) method as an aid in multidisciplinary design optimization. It is very helpful when there is little knowledge about the data and the exploration goals are implicit since users are able to directly participate in the exploration processes, shift and adjust the exploration goals if necessary. However, VDS for high-dimensional optimization problems will be less intuitive due to the limited ability to represent data of the high dimensionality in a 2D or 3D space. Hence, it can be hard to find out the relationship among the data, since a large amount of information is lost during the process of visualization.

2.3. Correspondence Analysis

2.3.1. History

The invention of correspondence analysis dates back to the early 20th century. Karl Pearson is [49] generally considered as the founder of the technique, who developed the correlation coefficient of a two-way contingency table using linear regression in around 1904. However, the original algebraic derivation of correspondence analysis was not derived until 1935, when Hirschfeld [50] further developed the formula.

Due to the development of Singular Value Decomposition (SVD), CA was further developed by Fisher [51]. Fisher integrated dimensionality reduction technique -- SVD with CA and use it on the hair and eye colour dataset of children from Scotland to show the structure of data. CA became popular in France by the work done by Jean-Paul Benzecri [52]-[54], who gave a geometric form to CA and made it easier to understand. Before 1970s, CA was relatively unknown in English speaking countries due to the language problem. The popular paper by Hill [55] first coined the method's name correspondence analysis and introduced it to the English speaking researchers. As he showed the mathematically similarity between CA and already popular methods such as principal components analysis, CA was quickly accepted by the researchers and widely spread in the world.

2.3.2. Advantages of CA Comparing to Other Similar Data Analysis Techniques

CA is similar to a number of other statistical techniques, including Factor Analysis (FA) [56], PCA [57], and Multidimensional Scaling (MDS) [58].

CA is similar to FA [59] as they are both exploratory methods to decompose the variance into a lower dimension and both reduce the dimension by losing minimal information, which can use fewer dimensions to explain the most of original dataset. The difference between CA and FA is that FA is applied on the correlation matrix [60], which requires a large number of samples to obtain. FA finds the factors from the covariance matrix [61] by using several techniques such as PCA, Alpha factoring [62], and Image factoring [63], and use the linear combination of the factors to describe the variables. CA however employs a mathematical transformation to the original data in the contingency table with no assumptions about the form of the correlation matrix. Therefore, correspondence analysis has highly flexible data requirements. The only strict data requirement is a rectangular data matrix with non-negative entries.

CA and PCA are both dimensionality reduction techniques [64]. They both create orthogonal components to describe the original dataset and the distances between the points in high dimensional space is well preserved in the low dimensional space. CA is similar to PCA, but it is better suited for analyzing contingency tables, which examine the associations among variables, while PCA extracts which variables explain the largest amount of variance in the data. The second difference between the two techniques is the way to decompose the data matrix. CA may be defined as a special case of PCA applied on the rows and columns of a table. In addition, CA and PCA are used under different circumstances. While PCA pays more attention to the most influential variables, CA focuses more on the similarity of the variables, which can be used to detect and represent underlying structures in a data set. CA's primary goal is to transform a table of numerical information into a graphical display, in which each row and each column is depicted as a point. The converted points can reveal the similarities and associations among variables, the closer two points are, the more similar those two variables are. However, total variance is decomposed in PCA to extract the most importance variables, which represents the largest amount of data.

MDS and CA both aim to help in the understanding of particular types of data by displaying the data graphically [65]. The objective of MDS is to determine a set of the points and the number of the points in the set to give a "good" fit for the observed proximities. The dissimilarity between two variables is predefined in the proximity matrices. The MDS method aims at finding the coordination for the set of points where each point will represent one variable and distance between two points reflects the dissimilarity between two variables. The larger an observed dissimilarity between two variables, the further apart should the points be in the graph. The coordinate of the points will change according to the fitness of the model by a formula that describes how well the proximities and the distances in the geometrical model match. An optimization method is usually applied to find the best coordinates. The disadvantage of MDS is that the proximity matrix needs to be calculated and for different type of problem, different measurement should be applied. For the problem without prior knowledge, it is hard to determine which measurement should be used to reveal the dissimilarity. Also, the optimization method for finding the optimal coordinate for the points is computational expensive, which makes it inefficiency for the high-dimensional problem. CA however represents the chi-square distance and the coordinate of the points is determined after the dimensionality reduction without knowing any other information of the problem. The operation on matrix decomposition is much faster than optimization and makes CA more efficient to calculate.

2.3.3. Applying Correspondence Analysis in HEB Optimization

The strength of applying correspondence analysis in a HEB problem is that it can reveal the structure of the data set without prior knowledge and represent it in low dimensions without losing essential information. Unlike other Data Analysis techniques, which require a large number of samples, CA can still generate accurate results with small datasets. This property is really important for the HEB problems as the computations are extremely expensive and the number of runs of the black box is expected to be minimized as much as possible. Another distinct advantage of correspondence analysis over other methods is that it can be used to find the relationship within dependent variables or within independent variables, as well as the inter relationship between dependent variables and independent variables. The duality of displaying all the relations, which is not present in other multivariate approaches, will facilitate analysis and detection of relationships. It provides graphic outputs that are easier to grasp the pattern than series of numbers. The multivariate treatment of the data through simultaneous considering multiple categorical variables can reveal relationships that would not be detected in a series of pair wise comparisons of variable. What's more, both qualitative and quantitative variables can be analyzed using correspondence analysis, since the continuous data can be first discretized into discrete data.

The basic idea of applying CA in HEB optimization is to split the design space and ranges of function values into several intervals, discover the symmetry of different domains of variables, and then determine the most important ones. It can also indicate the promising area where the optimal will most likely to appear. In addition, we also introduce certain visualization techniques as useful tools to uncover the data pattern. During optimization process, as the important variable is identified, less sampling step is need for other dimensions, which can reduce significant number of points when comes to high dimension. In addition, when the promising subdomains are identified, fewer sampling points are needed in other subdomains. The symmetric structure can help reduce the different combination of the value in each dimension. For example when x_1 and x_2 have symmetric distributed among the function value, then assign x_1 = a and x_2 =

b will result the same function value as x_1 = b and x_2 = a. Therefore, it will save the computation of a large number of combinations.

2.4. Summary

This chapter provides a brief review of current HEB optimization technologies and their disadvantages. After that, a data analysis method – Correspondence Analysis - is introduced and compared among other methods. Based on those, we start to discuss the possibility of applying CA to assist solving the HEB problem. The next chapter will provide the methodology of applying CA to HEB optimization in details and analyze the results.

Chapter 3. Methodology

This section first gives an introduction to the numerical optimization, space and regions for an optimization problem, and then provides the steps needed for revealing the structure of the black box problem in the optimization context. The method is developed based on CA, but it is different from original CA and the interpretation of the graph is also different. The proof of applying CA and the guideline of choosing the parameters are also provided. The test results of the methodology on mathematical functions are shown in the last part of this chapter.

3.1. Numerical Optimization

3.1.1. Standard Form

Numerical optimization is usually in a standard form as Eq. (1)

$$\begin{gathered} min\, f(\boldsymbol{x}) \\ s.t.\{g(x) \leq 0, h(x) = 0\} \end{gathered} \tag{1}$$

where $\boldsymbol{x}$ is a vector of n dimension composed by n variables and we call each variable in the vector as one design variable. For each $\boldsymbol{x}$, we can also consider it as one input of the system. $\boldsymbol{x}$ can be tuned to achieve different output of the model. The objective functions are denoted as $f(\boldsymbol{x})$, which are used to measure the performance of the inputs. The results of $f(\boldsymbol{x})$ are considered as the output of the system. $g(\boldsymbol{x})$ and $h(\boldsymbol{x})$ is a set of inequality and equality constraints, respectively. These determine the model's limitation and screen out infeasible inputs. If the goal is to find the maximal value of $f(\boldsymbol{x})$, we can easily change it to a minimization problem by simply use $\{\min -f(\boldsymbol{x})\}$ as the objective function.

The scope of this thesis is limited to the single objective optimization with no constraints. It means that for every input $\boldsymbol{x}$, there is one scalar output $f(\boldsymbol{x})$. But when it comes to the multi-objective optimization problem, we can aggregate multiple objectives into an overall performance function by using the weighted sum of each objective value as it is shown in Eq.(2)

$$f(x) = \sum_{i=1}^{m} w_i f_i(x) \tag{2}$$

where m is the number of the objective functions to be aggregated and w_i is the weight of the i^{th} objective function on the overall objective function.

3.1.2. Definitions of Terminologies

Two kinds of spaces that will be constantly referred to in this paper are defined in the following paragraphs.

Design space: the design space of the design variable with d dimensions is all possible designs that may be generated. The matrix $\boldsymbol{X}$ is used to represent the design space with s number of rows and t number of columns. Each row corresponds to one design and denoted as $\boldsymbol{x}$. The i^{th} element (column) in the row, which corresponds to the i^{th} design variable, is denoted as $\boldsymbol{x}_{\mathrm{i}}$. The vectors $\mathbf{lb}$ and $\mathbf{ub}$ are the boundaries that user defined for all the design variables in a specific optimization problem. For example, the i^{th} design variable is limited by the lower bound $\boldsymbol{lb}_i$ and the upper bound $\boldsymbol{ub}_i$. The number of variables and the boundaries of each design variable collectively determine the whole design space.

Performance space: the performance space represents the range of function values, which are generated from $f(\boldsymbol{x})$. Unlike the design space, the boundary of the performance space is unknown to the user. Instead, the goal of the optimization is to find the lower bound of the performance space and its corresponding design. The upper bound is usually unknown either, however, it is not interesting from the optimization perspective.

Function value bin: The points are grouped by their relative performance (function value). As the optimal function value is unknown to the user, it is impossible to tell whether a point is good or not. However, after we run the simulations and have a pool of tested points, we can say one point is relatively better than the other point according to their function values. The points can then be sorted and classified into different function value bins (groups) according to their function values. The relative ranking (represented by the percentage) is denoted as $R(f(\boldsymbol{x}))$, for example, a point $\boldsymbol{x}_1$ ranks 23th among the 50 points, then $R(f(\boldsymbol{x_1}))$ = 23/50 = 46%. Once lower bound r_l and upper bound r_u of the percentiles are given, a group of points for a certain function value bin can be determined.

$$F_B = \{x \in X | r_l \leq R(f(x)) \leq r_u\} \tag{3}$$

For example, the top 46% of the designs form one function value bin, then the points with function value smaller or equal to point $\boldsymbol{x}_1$ will all be selected to that function value bin.

In this thesis, the number of function value bins will be indicated and we want every function value bins has the same amount of samples, then, the ranking's lower bound and upper bound of each function value bin can be calculated. According to Eq. (4), the function value bins are determined as:

$$F_{Bi} = \left\{x \in X \middle| \frac{i-1}{n} \leq R\big(f(x)\big) \leq \frac{i}{n}\right\}, i \in [1, n] \tag{4}$$

where F_{Bi} is the i^{th} function value bin and n is the total number of function value bin, $\frac{i-1}{n}$ and $\frac{i}{n}$ are the lower bound and the upper bound of percentile for the i^{th} function value bin respectively.

Dimensional bin:

Dimensional bin is defined in design space without knowing the performance of the points. The points are grouped into different dimensional bin according to different conditions in the design space. Here we consider the dimensional bin separately for each variable. The general form of the dimensional bin is shown in Eq. (5):

$$D_{Bd} = \{x \in X | x_{dl} \leq x_d \leq x_{du}\} \tag{5}$$

The dimensional bins for the d^{th} variable (D_{Bd}) are classified according to the lower bound x_{dl} and the upper bound x_{du} of the d^{th} variable. All the points whose values of the d^{th} dimensions are between the boundaries will be put into the same dimensional bin. In this thesis, the number of dimensional bins for each variable (n) is given and the dimensional bins are defined according to the Eq. (6):

$$D_{Bdi} = \left\{x \in X \middle| \frac{(i-1)(ub_d - lb_d)}{n} \leq x_d \leq \frac{i(ub_d - lb_d)}{n}\right\}, i \in [1, n] \tag{6}$$

For the i^{th} dimensional bins for the d^{th} variables (D_{Bdi}), the range of the d^{th} variables of the points will always be $\frac{(ub_d - lb_d)}{n}$, where ub_d is the upper bound of the d^{th} variable in the design space and lb_d is the lower bound. In other word, the design space of the d^{th} variable is cut into n intervals of the same length. For a point, if the value of the d^{th} variable falls into the i^{th} interval, then the point is belong to the i^{th} dimensional bins for the d^{th} variables (D_{Bdi}).

3.2. Theory of Correspondence Analysis

Mathematically, CA is usually performed on a two-dimensional contingency table [66], and reveals the internal information of rows and columns. A contingency table is a table in a matrix format that displays the frequency distribution of variables. The i^{th} row and j^{th} column of the contingency table is denoted as n_{ij}. The basic contingency table, i.e., the two-dimensional contingency table, comes with two variables. Suppose there are n objects being classified according to two variables **A** with s values (**A**$_1$ to **A**$_s$) and **B** with t values (**B**$_1$ to **B**$_t$), each object defined by a value of **A** and a value of **B**. The two-way contingency table will display the frequency distribution of **A** and **B**, where n_{ij} are the number of objects within both values **A**$_i$ and **B**$_j$.

CA uses a scattered plot to represent the rows and columns of a contingency table as points in a graph. This allows it to reveal the information inside the table more intuitively. There are three ways of representation [67], including the row profile representation, column profile representation, and biplot representation. In row profile representation, the rows of the table are viewed as vectors of points with coordinates given by the columns of the table. These row profiles are constructed by dividing each cell value of the table by its row sum n_{i+}, where n_{i+} is the total of the observed values of the i^{th} row, i.e. $n_{i+} = \sum_{j=1}^{t} n_{ij}$. The coordinate of the i^{th} row profile $\mathbf{V_{i+}}$ is therefore $\begin{bmatrix} \frac{n_{i1}}{n_{i+}} & \frac{n_{i2}}{n_{i+}} & \cdots & \frac{n_{it}}{n_{i+}} \end{bmatrix}$. The standard base $\boldsymbol{e_j}$ will be the coordinate of the point of the j^{th} column. The column points are called vertices because their positions are all on the vertices of a unit cube. The row points are called the profiles as we compare the rows and use columns as reference. In the row profile representation, the principle coordinate and the standard coordinate refer to “coordinate positions of the profile” and “coordinate position of the vertices” respectively. Similarly, in the column profile representation, the comparison among columns is considered and the row points are used as references. Column points or profiles and masses are also similarly defined. The row profile in this case is in standard coordinate and the column profiles are in the principle coordinate. Finally, for the biplot, both row points and the column points are scaled and put in principle coordinates, which allows for interpreting the row-row relationship and column-column relationship at the same time. This work uses the row profile representation for visualization.

SVD [68] is employed by CA to extract the principal axes and used to find an optimal subspace for plotting both the row and column points, weighing them by their respective masses. CA provides a low-dimensional explanation for the interaction or dependence between the rows and columns of the CT. The basic steps of CA are shown below:

***Algorithm*：**

1. The contingency table, **N** (row, column), is converted to correspondence matrix **P** by dividing matrix **N** with its grand total, this will normalize the matrix and ensure the analysis independent of the size of the matrix.

$\mathbf{P} = \frac{1}{n_{++}}\mathbf{N}$, where $n_{++} = \sum_{i=1}^{s}\sum_{j=1}^{t} n_{ij}$ is the table grand total

2. Calculate the diagonal matrix of row and column masses $\mathbf{D_r}$ = diag($\boldsymbol{r}$) and $\mathbf{D_c}$ = diag($\boldsymbol{c}$) and ($\mathbf{P}$-$\boldsymbol{rc}^T$): $\boldsymbol{r}$ is the row masses vector with $r_i = \sum_{j=1}^{J} p_{ij}$ and $\boldsymbol{c}$ is the column masses vector with $c_i = \sum_{i=1}^{I} p_{ij}$.

3. Calculate the matrix **S**, which represents the standardized residue for matrix **N**. $\mathbf{S} = \mathbf{D_r}^{-1/2}(\mathbf{P}\text{-}\boldsymbol{rc}^T)\mathbf{D_c}^{-1/2}$

4. Calculate the SVD for matrix $\mathbf{S}=\mathbf{U\Sigma V}^\mathbf{T}$, where $\mathbf{U}^\mathbf{T}\mathbf{U} = \mathbf{V}^\mathbf{T}\mathbf{V} = \mathbf{I}$

5. The plotting coordinates for the rows, $\mathbf{V}_{i+}$, and columns, $\mathbf{V}_{+j}$, with respect to the principal axes are computed by:

   $\mathbf{V}_{i+} = \mathbf{D_r}^{-1/2}\mathbf{U\Sigma}$ and $\mathbf{V}_{+j} = \mathbf{D_c}^{-1/2}\mathbf{V}$

The SVD is the fundamental step for CA. This matrix decomposition technique expresses rectangular *m*×*n* matrix **W** in the form of $\mathbf{W}=\mathbf{U\Sigma V}^\mathbf{T}$, where $\mathbf{U}^\mathbf{T}\mathbf{U} = \mathbf{V}^\mathbf{T}\mathbf{V} = \mathbf{I}$. **U** is an *m*×*m* orthonormal matrix, **Σ** is an *m*×*n* rectangular diagonal matrix with non-negative real numbers on the diagonal, and $\mathbf{V}^\mathbf{T}$ is an *n*×*n* orthonormal matrix. The left most matrix **U** is composed of the *m* columns eigenvectors of $\mathbf{WW}^\mathbf{T}$, which is also called the left-singular vector of **W**. Similarly, the right most matrix **V** is composed of the *n* columns eigenvectors of $\mathbf{W}^\mathbf{T}\mathbf{W}$, which is called the right-singular vector of **W**. The diagonal values σ of **Σ** with elements $\sigma_1 > \sigma_2 > \sigma_3 > . . . >\sigma_k > 0$ are the singular values of **W**, where **W** has rank *k*.

Chi-square Test

Two other important concepts in CA are Chi-square test and inertia. In order to understand inertia, we need to introduce the Chi-square test first. Generally, Chi-square test is a statistical hypothesis test, which validates the hypothesis by first assuming the hypothesis is valid and then measuring the differences between the expected values and the observed ones. If the difference between expected values and observed values is sufficiently small, the hypothesis cannot be rejected (or is considered valid). Otherwise, the hypothesis is rejected. Here, we use it to measure the independence, variance, or correlation between rows and columns as well as to determine whether there is a significant association between two variables. For example, if we want to find out whether the rows are independent, we first assume that the rows are independent and use the Chi-square test to test this hypothesis. The value of the test-statistic is denoted by χ^2 and defined as **(7)**:

$$\chi^2 = \sum \frac{(observed - expected)^2}{expected} = \sum_{i}^{rows} \sum_{j}^{columns} \frac{(n_{\mathrm{ij}} - \mathrm{E}[n_{ij}])^2}{\mathrm{E}[n_{ij}]} \qquad (7)$$

As we can see in the equation above, the observed value for each cell in the table is n_{ij} and the expected value is denoted as $E[n_{ij}]$. Chi-square test requires our knowledge of the expected values of rows and columns. In order to find such information, some concepts of the contingency table are first defined:

1. **Row profile:** The row profiles $\mathbf{V_{i+}}$ which has been introduced before is in the form of $\begin{bmatrix}\frac{n_{i1}}{n_{i+}} & \frac{n_{i2}}{n_{i+}} & \cdots & \frac{n_{it}}{n_{i+}}\end{bmatrix}$.

2. **Correspondence matrix:** Correspondence matrix is defined as the original table (or matrix) **N** divided by its grand total n_{++}. Therefore, $p_{ij} = \frac{n_{ij}}{n_{++}}$

3. **Row sum:** The row sum of the i^{th} row is the sum of observed values in that row which is denoted as n_{i+} and $n_{i+} = \sum_{j=1}^{t} n_{ij}$

4. **Row mass:** The row mass of the i^{th} row (r_i) is the row sum of the i^{th} row divided by its grand total. It is denoted as $\frac{n_{i+}}{n_{++}}$. We can obverse that the row mass equals to the row sum of the i^{th} row in the correspondence matrix.

5. **Row mass vector:** The row mass vector (***r***) is a vector of the row mass of different rows in the form of $\begin{bmatrix}\frac{n_{1+}}{n_{++}} & \frac{n_{2+}}{n_{++}} & \cdots & \frac{n_{t+}}{n_{++}}\end{bmatrix}$

6. **Column mass vector:** Similarly, the column mass vector ($\boldsymbol{c}^{T}$) is a vector of the column mass of different columns in the form of $\begin{bmatrix} \frac{n_{+1}}{n_{++}} & \frac{n_{+2}}{n_{++}} & \cdots & \frac{n_{+s}}{n_{++}} \end{bmatrix}$, where n_{+j} is the total of the j^{th} column and $n_{+j} = \sum_{i=1}^{t} n_{ij}$.

7. **Average row profile:** The average row profile ($\overline{\boldsymbol{V_{\iota+}}}$) is the centroid of the row profile which equals to the weighted average of all row profiles, where the weight of each row profile is the row mass. Therefore $\overline{\boldsymbol{V_{\iota+}}} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{s}} \boldsymbol{r_i V_{i+}} = \frac{n_{i+}}{n_{++}} \times \begin{bmatrix} \frac{n_{i1}}{n_{i+}} & \frac{n_{i2}}{n_{i+}} & \cdots & \frac{n_{it}}{n_{i+}} \end{bmatrix}$ and it is in the form of $\begin{bmatrix} \frac{n_{+1}}{n_{++}} & \frac{n_{+2}}{n_{++}} & \cdots & \frac{n_{+t}}{n_{++}} \end{bmatrix}$. It is easy to find out that average row profile is actually equal to the column mass vector.

8. **Average column profile:** Similarly, the average column profile is the weighted average of all column profiles, in the form of $\begin{bmatrix} \frac{n_{1+}}{n_{++}} & \frac{n_{2+}}{n_{++}} & \cdots & \frac{n_{s+}}{n_{++}} \end{bmatrix}$ and is equal to the row mass vector.

9. **Diagonal matrix**: diagonal matrix of row masses is a diagonal matrix whose values on the diagonal are the row masses. It is denoted as $\mathbf{D_r}$**,** and $\mathbf{D_r}$ = diag($\boldsymbol{r}$). Diagonal matrix of column masses is similarly defined and $\mathbf{D_c}$ = diag($\boldsymbol{c}$).

Assume that all rows are homogeneous which means that the values from each column follow the same distribution as those in another column. Therefore, the expected values of the i^{th} row should be equal to the average row profile times the mass of the i^{th} row and its grand total. The expected value of the i^{th} row is in the form of (8)

$$\mathrm{E}[n_{i+}] = \overline{\boldsymbol{V_{\iota+}}} \times \boldsymbol{r}_i \times n_{++} = \begin{bmatrix} \frac{n_{+1}}{n_{++}} & \frac{n_{+2}}{n_{++}} & \cdots & \frac{n_{+t}}{n_{++}} \end{bmatrix} \times \frac{n_{i+}}{n_{++}} \times n_{++} \qquad (8)$$

The expected value of the cell in the i^{th} row and j^{th} column then as (9) indicates:

$$\mathrm{E}\left[n_{ij}\right] = \overline{\boldsymbol{V}_{\boldsymbol{ij}}} \times \boldsymbol{r}_i \times n_{++} = \frac{n_{+j}}{n_{++}} \times \frac{n_{i+}}{n_{++}} \times n_{++} = n_{+j} \times \frac{n_{i+}}{n_{++}} \tag{9}$$

where n_{ij} and E[n_{ij}] represent the observed value and the expected value of the cell in i^{th} row and j^{th} column respectively.

If all rows are homogeneous, the observed (actual) values of each row are equal to its expected values. However, for those more generic cases, the rows are usually not homogeneous. Then Pearson's Chi-square test is introduced to measure the differences between the expected values and the observed (actual) ones. A smaller chi-square test value means it is more probable that the rows in the contingency table are homogeneous. Otherwise, the hypothesis is rejected and the rows are not homogeneous. The Chi-square test for the contingency table is then calculated by **(10)** as follows:

$$\begin{aligned}
\chi^2 &= \sum_{i}^{rows} \sum_{j}^{columns} \left(n_{\mathrm{ij}} \frac{1}{\sqrt{\mathrm{E}\left[n_{ij}\right]}} - \mathrm{E}\left[n_{ij}\right] \frac{1}{\sqrt{\mathrm{E}\left[n_{ij}\right]}} \right)^2 \\
&= \sum_{i}^{rows} \sum_{j}^{columns} \left(n_{\mathrm{ij}} \frac{1}{\sqrt{\frac{n_{\mathrm{i+}} n_{\mathrm{+j}}}{n_{++}}}} - \frac{n_{\mathrm{i+}} n_{\mathrm{+j}}}{n_{++}} \frac{1}{\sqrt{\frac{n_{\mathrm{i+}} n_{\mathrm{+j}}}{n_{++}}}} \right)^2 \\
&= \sum_{i}^{rows} \sum_{j}^{columns} \left(\frac{n_{\mathrm{ij}}}{n_{++}} \frac{1}{\sqrt{\frac{n_{\mathrm{i+}} n_{\mathrm{+j}}}{n_{++} n_{++}}}} - \frac{n_{\mathrm{i+}} n_{\mathrm{+j}}}{n_{++} n_{++}} \frac{1}{\sqrt{\frac{n_{\mathrm{i+}} n_{\mathrm{+j}}}{n_{++} n_{++}}}} \right)^2 \times n_{++}
\end{aligned} \tag{10}$$

For simplicity, **(10)** can be written as (11) only using the concept of correspondence matrix, column masses, row masses and grand total.

$$\chi^2 = \sum_{i}^{rows} \sum_{j}^{columns} \left(p_{\mathrm{ij}} \frac{1}{\sqrt{r_{\mathrm{i}} c_{\mathrm{j}}}} - r_{\mathrm{i}} c_{\mathrm{j}} \frac{1}{\sqrt{r_{\mathrm{i}} c_{\mathrm{j}}}}\right)^2 \times n_{++} \tag{11}$$

Using the matrix representation, (11) can also be written as **(12)**. As we can see **(12)** is similar to the formula in the third step of the algorithm as we presented above:

$$\chi^2 = [D_r^{-\frac{1}{2}}(P - rc^T)D_c^{-\frac{1}{2}}]^2 \times n_{++} \tag{12}$$

Chi-square distance

For the purposes of correspondence analysis, the differences between the distributions of the given rows are measured by the so-called χ^2 distances, which are weighted Euclidean distances between normalized rows (calculated by dividing row values by their respective row sum), with weights inversely proportional to the square roots of the column mass. In symbols, the χ^2 distance between row *i* and row *k* is given by the expression:

$$d_{ik}^2 = \sum_{j=1}^{t} \frac{\left(\frac{n_{ij}}{n_{i+}} - \frac{n_{kj}}{n_{k+}}\right)^2}{\frac{n_{+j}}{n_{++}}} \tag{13}$$

As their name suggests, χ^2 distances are closely related to the χ^2 statistic of the previous section. To show how they are related, consider the average row profile - termed the *centroid of row profiles* in correspondence analysis—whose values are simply the column mass vector: $\boldsymbol{c} = \begin{bmatrix} \frac{n_{+1}}{n_{++}} & \frac{n_{+2}}{n_{++}} & \cdots & \frac{n_{+t}}{n_{++}} \end{bmatrix}$. The distance between row profile and the average row profile is:

$$
\begin{aligned}
d_{ic}^2 &= \sum_{j=1}^{t} \frac{\left(\frac{n_{ij}}{n_{i+}} - \frac{n_{+j}}{n_{++}}\right)^2}{\frac{n_{+j}}{n_{++}}} = \sum_{j=1}^{t} \frac{\left(\frac{n_{ij}}{n_{i+}} - \frac{n_{+j}}{n_{++}}\right)^2}{\frac{n_{+j}}{n_{++}}} = \sum_{j=1}^{t} \frac{\left(\frac{n_{++}n_{ij} - n_{i+}n_{+j}}{n_{++}n_{i+}}\right)^2}{\frac{n_{+j}}{n_{++}}} \\
&= \sum_{j=1}^{t} \frac{\left(\frac{n_{++}n_{ij} - n_{i+}n_{+j}}{n_{++}n_{++}}\right)^2 \times \frac{n_{++}n_{++}}{n_{i+}n_{i+}}}{\frac{n_{+j}}{n_{++}}} \qquad (14) \\
&= \sum_{j=1}^{t} \frac{\left(\frac{n_{\mathrm{ij}}}{n_{++}} - \frac{n_{\mathrm{i}+}n_{+\mathrm{j}}}{n_{++}n_{++}}\right)^2}{\frac{n_{i+}}{n_{++}}\frac{n_{+j}}{n_{++}}} \times \frac{1}{\frac{n_{i+}}{n_{++}}}
\end{aligned}
$$

The Equation (14) can also be written as (15) by using the row mass and column mass:

$$
d_{ic}^2 = \sum_{j=1}^{t} \left(p_{\mathrm{ij}} \frac{1}{\sqrt{r_{\mathrm{i}}c_{\mathrm{j}}}} - r_{\mathrm{i}}c_{\mathrm{j}} \frac{1}{\sqrt{r_{\mathrm{i}}c_{\mathrm{j}}}}\right)^2 \times \frac{1}{r_{\mathrm{i}}} \qquad (15)
$$

Inertia

Drawing an analogy with the physical concept of angular inertia, correspondence analysis defines the *inertia* of a row as the product of the row and the square of its distance to the centroid, $r_i d_{ic}^2$. Comparing the expression for d_{ic}^2 in (15) with definition of the χ^2 statistic in (11), it follows that the total inertia of all the rows in a contingency matrix is equal to the χ^2 statistic divided by n_{++}, a quantity known as Pearson's *mean-square contingency*, denoted ϕ^2:

$$
\phi^2 = \sum_{i=1}^{s} d_{ic}^2\, r_i = \sum_{i=1}^{s}\sum_{j=1}^{t} \left(p_{\mathrm{ij}} \frac{1}{\sqrt{r_{\mathrm{i}}c_{\mathrm{j}}}} - r_{\mathrm{i}}c_{\mathrm{j}} \frac{1}{\sqrt{r_{\mathrm{i}}c_{\mathrm{j}}}}\right)^2 = \frac{\chi^2}{n_{++}} \qquad (16)
$$

The inertia of CA, which is denoted by ϕ^2, can be easily obtained from the Chi-square test-statistic as (17) shows. It is calculated as the Pearson's Chi-square test value for the entire contingency table, and then divided by the table's grand total. The inertia can reveal the same information as Chi-square test but it can be represented with only rows in correspondence matrix, column masses, and row masses.

$$\phi^2 = \frac{\chi^2}{n_{++}} = [D_r^{-\frac{1}{2}}(P - rc^T)D_c^{-\frac{1}{2}}]^2 \tag{17}$$

Graphical Representation

Correspondence analysis provides a means of representing a table of χ^2 distances in a graphical form, with rows represented by points, so that the distances between points approximate the χ^2 distances between the rows they represent. This representation can be computed by using the standardized residuals matrix (**S**) of the contingency table. As it is shown in the second step of CA, every cell is subtracted by the expected value as the procedure of (**P-*rc*^{T}**) indicates. It ensures that the average row profile (centroid) will always be the origin point in the plot, which also represents the difference between expected values and the observed values. Furthermore, every row is scaled by $\mathbf{D_r^{-1/2}D_c^{-1/2}}$, which integrates the weights in the inertial formula to the formula of Euclidean distance.

$$s_{ij} = \frac{p_{\mathrm{ij}} - r_{\mathrm{i}}c_{\mathrm{j}}}{\sqrt{r_{\mathrm{i}}c_{\mathrm{j}}}} \tag{18}$$

Therefore, the Euclidean distance $d_{iO}'^2$ from all points to the origin point is actually indicating the inertial value as (19) shows.

$$\sum_{i=1}^{s} d_{iO}'^2 = \sum_{i=1}^{s}\sum_{j=1}^{t}(s_{ij} - 0)^2 = \sum_{i=1}^{s}\sum_{j=1}^{t}\left(\frac{p_{\mathrm{ij}} - r_{\mathrm{i}}c_{\mathrm{j}}}{\sqrt{r_{\mathrm{i}}c_{\mathrm{j}}}}\right)^2 = \phi^2 \tag{19}$$

In visualization, we plot all row profiles as points in a high dimensional plane, with the average row profile lying at the coordinate center. The summation of Euclidean distance from each point to the coordinate center is the total inertial, which measures the homogeneity of the rows in contingency table. Similarly, the Euclidean distance of two points measures the homogeneity or similarity of those two rows. The rows are considered to be more similar if their inertial value or the distance is smaller.

SVD provides us a technique to represent the rows and columns of the matrix in different spaces while preserving the distance. The row profiles are projected onto a different space so the coordinates of the points presenting the rows are changed. However, the distance information is preserved during SVD [69], therefore, the information such as inertia is still preserved. The distance between the row (column) points can still review the homogeneity of the rows (columns) even though the coordinates change.

Interpretation of CA:

The coordinates of the row and column profile points are calculated from the above algorithm and then are plotted in the same graph. The row profile points and the column profile points are separated into two groups and analyzed separately according to the following interpretation strategy.

1. Interpretation of the distance within points of a same group

The distance of two points in the graph is actually equal to the χ^2 distances. The smaller χ^2 distances value reflects more similarity of two row profiles. Therefore, if two points of a same group are close to each other on the graph, they will have a similar profile. For example, if the i^{th} and j^{th} row profile points are close to each other, it means that the values in the corresponding row profiles are similar.

2. Interpretation of the distance between points and the origin of the same group

The sum of all distances between points and the origin is equal to the value of inertial which represents the homogeneity of all the rows. In this case, a larger distance between a point and the origin represents a more distinguishable feature. On the contrary, if the distance is small, there will be less difference among the values in the profile. Otherwise we can say the rows are more homogenous.

3. Interpretation of projection between points belonging to different groups

To analyze the correlation between a point (P_A) in one group and all the points in the other group, one calculates the dot product between the vectors of every two points. For the vector V_A of point $P_{A,}$ it starts from the origin and ends at P_A. For the point (P_B) in the other group, the vector V_B is from the origin to P_B. A large dot product of V_A and V_B indicates that the feature represented by point B is more relevant to the feature represented by point A.

The row profiles show the distribution among the columns. For a certain row profile, row A for example, it has different values across the columns. Obviously, if row A has the same relevance among all the columns, the values in different columns should be the same. Otherwise, if row A at the Bth column has a larger value, it means that row A is more relevant to the Bth column. Therefore, when we analyze each row in the correspondence matrix, a larger value at a certain column means that the row is more relevant to that column.

In the graph, the coordinate of the row points and the column points is obtained by applying SVD on the correspondence matrix ($\mathbf{S}=\mathbf{U\Sigma V}^{\mathbf{T}}$). The plotting coordinates for the rows are $\mathbf{V}_{i+} = \mathbf{D}_{\mathbf{r}}^{\mathbf{-1/2}}\mathbf{U\Sigma}$ and the plotting coordinates for the columns are $\mathbf{V}_{+j} = \mathbf{D}_{\mathbf{c}}^{\mathbf{-1/2}}\mathbf{V}$. Therefore, the product of rows and columns can be represented as (20):

$$\mathbf{V}_{\mathrm{i}+} \cdot \mathbf{V}_{+\mathrm{j}} = (\mathbf{D}_{\mathrm{r}}^{-\frac{1}{2}}\mathbf{U\Sigma})(\mathbf{D}_{\mathrm{c}}^{-\frac{1}{2}}\boldsymbol{V})^{\boldsymbol{T}} = \mathbf{D}_{\mathrm{r}}^{-\frac{1}{2}}(\mathbf{U\Sigma V}^{\boldsymbol{T}})\mathbf{D}_{\mathrm{c}}^{-\frac{1}{2}} = \mathbf{D}_{\mathrm{r}}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}_{\mathrm{c}}^{-\frac{1}{2}} \tag{20}$$

In other words, the correspondence matrix can be represented as the product of rows and columns and times the coefficients ($\mathbf{D}_{\mathbf{c}}^{\mathbf{-1/2}}$ and $\mathbf{D}_{\mathbf{r}}^{\mathbf{-1/2}}$). The dot product between two points (for example P_A and P_B) from different groups is actually the product of row A and column B. The value of the dot product is the value s_{AB} in the correspondence matrix, which can show the relevance of row and column.

The information discovered above can reveal several properties of the data set. A popular plot of CA to reveal the brand associations will be used as an example. The original data in the matrix format are obtained from [70] and are shown in Table 3.1. We have modified the data by adding the last row. We can observe that the attributes of Coke and Pepsi are similar. So are Diet Coke and Diet Pepsi. Coke and Pepsi are older and more traditional while Diet Coke and Diet Pepsi are more feminine according to the matrix. The attribute named feminine represents the percentage of women drinking a certain kind of drinks. The Coke Norm has the same values for all the attributes.

Table 3.1 Contingency table for brand association example

	Feminine	Health-conscious	Innocent	Older	New experiences	Rebellious	Sleepy	Traditional	Weight-conscious
Coke	6	2	9	65	23	26	9	92	1
Diet Coke	57	59	23	22	9	5	24	15	76
Coke Zero	22	55	13	5	52	31	9	3	65
Pepsi	9	2	10	38	16	18	14	54	0
Diet Pepsi	61	58	43	9	16	4	30	3	76
Pepsi Max	10	31	7	6	50	45	6	4	41
Coke Norm	55	55	55	55	55	55	55	55	55

1. From Figure 3.1, we can discover that Coke and Pepsi are similar and so are Diet Coke and Diet Pepsi, because the corresponding points are close to each other in the graph.

2. The distance from Coke Norm to the origin point is smaller than the other brands. Therefore, Coke Norm doesn’t have as much distinguishable features as the other brands.

3. The larger projection of points in the row profile onto a certain point in the column profile, the stronger the relationship is. Thus, the association between Coke and (being) Traditional is substantially stronger than the association between Pepsi and traditional, because the projection of Coke onto the vector of Traditional is larger than Pepsi. Coke and Pepsi tend to be older and more traditional, while Diet Coke and Diet Pepsi are more feminine.

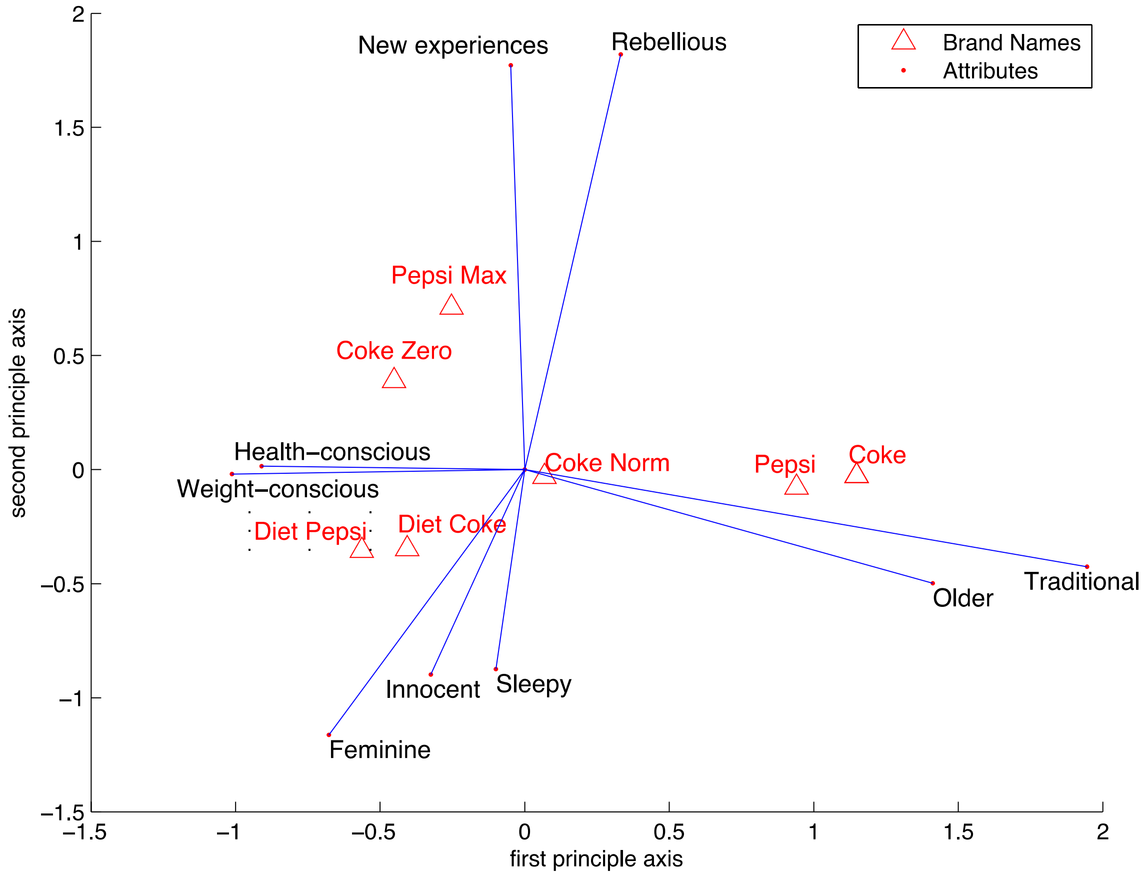


Figure 3.1 Correspondence Analysis of brand association

3.3. Generating the Contingency Table

As is done in CA, we are going to solve the HEB problems using a two-way contingency table. For simplicity, we first define a black box function with input of *d* variables and output of one function value. The following steps demonstrate the details of constructing a two-way contingency table, which will be used in the analysis and visualization later:

Step (1) Sample *n* sets of inputs using Latin hypercube sampling (LHS) [71], and calculate the corresponding function values (outputs).

Step (2) Divide the range of function values into *t* bins with each bin containing the same number of points. These bins are denoted as *function value bins*. $\mathbf{B_j}$ represents the set of points that fall into the j^{th} function value bin.

Step (3) For each input variable, divide the domain into *s* bins with equal lengths, named *dimensional bins*. $\mathbf{A_{ki}}$ represents the set of points that fall into the i^{th} dimensional bin of the k^{th} variable.

Step (4) Create an *s* by *t* matrix for each input variable. For the k^{th} variable, the matrix is denoted as $\mathbf{M_k}$, and a_{kij} is the number of points that lie in the i^{th} dimensional bin of the k^{th} variable and in the j^{th} function value bin of the output.

$$\mathbf{M_k} = \begin{bmatrix} a_{k11} & a_{k12} & \cdots & a_{k1t} \\ \cdots & \cdots & & \cdots \\ a_{ks1} & a_{ks2} & \cdots & a_{kst} \end{bmatrix} \tag{21}$$

Step (5) In the last step, we have obtained *d* number of *s* × *t* matrices. We can stack these matrices vertically and get

$$\mathbf{W} = \begin{bmatrix} \mathbf{M_1} \\ \cdots \\ \mathbf{M_d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a_{111}} & \boldsymbol{a_{112}} & \cdots & \boldsymbol{a_{11t}} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{a_{1s1}} & \boldsymbol{a_{1s2}} & \cdots & \boldsymbol{a_{1st}} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{a_{d11}} & \boldsymbol{a_{d12}} & \cdots & \boldsymbol{a_{d1t}} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{a_{ds1}} & \boldsymbol{a_{ds2}} & \cdots & \boldsymbol{a_{dst}} \end{bmatrix} \tag{22}$$

The final contingency table will be:

Table 3.2 Contingency table for HEB problems

	B_1	…	B_j	…	B_t	Total
A_{11}	a_{111}	…	a_{11j}	…	a_{11t}	a_{11+}
…	…	…	…	…	…	…
A_{1s}	a_{1s1}	…	a_{1sj}	…	a_{1st}	a_{1s+}
…	…	…	…	…	…	…
A_{d1}	a_{d11}	…	a_{d1j}	…	a_{d1t}	a_{d1+}
…	…	…	…	…	…	…
A_{ds}	a_{ds1}	…	a_{dsj}	…	a_{dst}	a_{ds+}
Total	a_{+1}	…	a_{+j}	…	a_{+t}	dn

A small example below will help to describe how a contingency table is generated. Suppose that $F(x, y) = x + y, (x, y \in [0,1])$ is the internal function of a black box, with two variables *x* and *y*. We divide the range of function values into *t* = 3 function value bins, and divide the domain of each input variable into *s* = 2 dimensional bins.

Step (1) Randomly generated *n* = 12 points using Latin hypercube sampling.

Table 3.3 randomly generated points

Points id	*x*	*y*	*F(x,y)*
1	0.6286	0.1045	0.7331
2	0.9485	0.8204	1.7689
3	0.1723	0.4047	0.5770
4	0.01370	0.4495	0.4632
5	0.5087	0.7288	1.2375
6	0.8681	0.05630	0.9244
7	0.4514	0.9665	1.4179
8	0.7621	0.2995	1.0616
9	0.1638	0.6345	0.7983
10	0.6762	0.5325	1.2087
11	0.2994	0.9028	1.2022
12	0.4136	0.2343	0.6479

Step (2) The domains of *x* and *y* are equally divided into *s*=2 bins (intervals) respectively. In this example, both *x*'s and *y*'s domains lie in the interval [0, 1], so the dimensional bins are [0, 0.5] and [0.5, 1] for both *x* and *y*. First, we process the data only in terms of variable *x*: let $\mathbf{A}_{x1}$ be a set that represents the points whose *x* values lie in the first bin of variable *x* ([0, 0.5]). It includes points 3, 4, 7, 9, 11, 12. $\mathbf{A}_{x2}$ is another set in which the points have their *x* values lie in the second bin of variable *x*. It includes points 1, 2, 5, 6, 8, 10. Then, we process the data again, but this time only in terms of variable *y*. Similarly, we get set $\mathbf{A}_{y1}$ that includes points 1, 3, 4, 6, 8, 12, and set $\mathbf{A}_{y2}$ that includes 2, 5, 7, 9, 10, 11.

Step (3) The range of function value (*F*) is divided into *t*=3 bins, each bin containing the same number of points. In this example, we can set the three bins as [0, 0.7331], (0.7331, 1.2022] and (1.2022, 1.7689]. Points 4, 3, 12, 1 with smaller function values are classified into set $\mathbf{B}_1$, and four points 9, 6, 8, 11 are in $\mathbf{B}_2$, while the other points 10, 5, 7, 2 with larger function values are classified into $\mathbf{B}_3$.

Step (4) Create a two-way contingency table **W** of **A** and **B**. **A** has four possible values, $\mathbf{A}_{x1}$, $\mathbf{A}_{x2}$, $\mathbf{A}_{y1}$, $\mathbf{A}_{y2}$, and **B** has three possible values, $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$. The value in each cell is the number of points that fall into the corresponding function value bin and dimensional bin. For example, $\mathbf{A}_{x1}$ has 3, 4, 7, 9, 11, 12 and $\mathbf{B}_1$ has 1, 3, 4, 12, therefore, the cell in the first row and the first column is $|\ \mathbf{A}_{x1} \cap \mathbf{B}_1| = |3, 4, 12| = 3$.

Thus after this step, we get the contingency table **W**:

Table 3.4 Contingency table of the example

	B_1 (bin 1 of F)	B_2 (bin 2 of F)	B_3 (bin 3 of F)	Row mass
A_{x1} (bin 1 of x)	3	2	1	6
A_{x2} (bin 2 of x)	1	2	3	6
A_{y1} (bin 1 of y)	4	2	0	6
A_{y2} (bin 2 of y)	0	2	4	6
Column mass (Average row profile)	8	8	8	24

The fifth row of Table 3.4 contains the column sum of the table. It reveals that one third of the points are in each of the function value bin according to the function value and because we classify the points to three bins with equally number of points, the element in the average row profile (fifth row) will always be the same. As the number of sample points that fall into each interval with the same length of the domains will be the same according to the properties of LHS, we divide the range of each variable into bins of equal length. Therefore the total number of points in each dimensional bin will be the same, which leads to the same value of each element in the fourth column. The row profiles are used to compare the difference between the various ranges of the variables and their distribution of the function values.

3.4. Process the Contingency Table

After we get the contingency table, it is necessary to process the table for further analysis. The steps are shown as follows.

Step (1) Calculate the correspondence matrix **P** with elements p_{ij} by dividing matrix **W** by its grand total, this will normalize the matrix and ensure the analysis is independent of the size of the matrix. So we get $\mathbf{P} = \frac{1}{n}\mathbf{W}$. In our example in the last section, *n* is 24, and the Table 3.4 becomes

Table 3.5 Correspondence Table for the example

	B_1 (bin 1 of F)	B_2 (bin 2 of F)	B_3 (bin 3 of F)	Row Mass
A_{x1} (bin 1 of x)	1/8	1/12	1/24	1/4
A_{x2} (bin 2 of x)	1/24	1/12	1/8	1/4
A_{y1} (bin 1 of y)	1/6	1/12	0	1/4
A_{y2} (bin 2 of y)	0	1/12	1/6	1/4
Column Mass	1/3	1/3	1/3	1

Step (2) Calculate the diagonal matrix of row and column masses $\mathbf{D_r}$ = *diag*(***r***) and $\mathbf{D_c}$ = *diag*(***c***). Then calculate the residue matrix ($\mathbf{P}$-$\boldsymbol{rc}^T$): ***r*** is the row masses vector with $r_i = \sum_{j=1}^{J} p_{ij}$ and ***c*** is the column masses vector with $c_i = \sum_{i=1}^{I} p_{ij}$. ($\mathbf{P}$-$\boldsymbol{rc}^T$) is the residue matrix $\mathbf{M_r}$, which can measure the strength of the difference between observed and expected values. It can be represented as

$$residual\ \mathrm{M}_{\mathrm{r}_{ij}} = \ \mathrm{observed}_{\ \mathrm{ij}} - \mathrm{expected}_{\ \mathrm{ij}} \ = p_{ij} - t_{ij} \qquad (23)$$

Where

$$t_{ij} = row\ mass\ i \times column\ mass\ j$$

For the previous example, ($\mathbf{P}$-$\boldsymbol{rc}^T$), $\mathbf{D_r}$ and $\mathbf{D_c}$ will be:

$$\mathbf{D_r} = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix} \quad \mathbf{D_c} = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \tag{24}$$

$$\mathbf{P} - \mathbf{rc^T} = \begin{bmatrix} \frac{1}{8} & \frac{1}{12} & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{12} & \frac{1}{8} \\ \frac{1}{6} & \frac{1}{12} & 0 \\ 0 & \frac{1}{12} & \frac{1}{6} \end{bmatrix} - \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix} \times \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{1}{24} & 0 & \frac{-1}{24} \\ \frac{-1}{24} & 0 & \frac{1}{24} \\ \frac{1}{12} & 0 & \frac{-1}{12} \\ \frac{-1}{12} & 0 & \frac{1}{12} \end{bmatrix} \tag{25}$$

Step (3) Calculate the matrix **S**, which represents the standardized residue for matrix **W** obtained above. $\mathbf{S} = \mathbf{D_r}^{-1/2}(\mathbf{P}\text{-}\boldsymbol{rc}^T)\mathbf{D_c}^{-1/2}$, where $\mathbf{D_r}$ and $\mathbf{D_c}$ are the diagonal matrices of row and column respectively. When comparing the cells, the standardized residual makes it easy to see which cells are contributing the most to the value, and which are contributing the least. According to the example in Table 3.5, the matrix **S** can be obtained by

$$\mathbf{S} = \mathbf{D_r}^{-\frac{1}{2}}(\mathbf{P} - \mathbf{rc^T})\mathbf{D_c}^{-\frac{1}{2}}$$

$$= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \times \begin{bmatrix} \frac{1}{24} & 0 & \frac{-1}{24} \\ \frac{-1}{24} & 0 & \frac{1}{24} \\ \frac{1}{12} & 0 & \frac{-1}{12} \\ \frac{-1}{12} & 0 & \frac{1}{12} \end{bmatrix} \times \begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{24} \times 2 \times \sqrt{3} & 0 & \frac{-1}{24} \times 2 \times \sqrt{3} \\ \frac{-1}{24} \times 2 \times \sqrt{3} & 0 & \frac{1}{24} \times 2 \times \sqrt{3} \\ \frac{1}{12} \times 2 \times \sqrt{3} & 0 & \frac{-1}{12} \times 2 \times \sqrt{3} \\ \frac{-1}{12} \times 2 \times \sqrt{3} & 0 & \frac{1}{12} \times 2 \times \sqrt{3} \end{bmatrix} \tag{26}$$

Step (4) Apply SVD on matrix **S**, **S**=**UΣV**$^{\mathbf{T}}$

The detail of calculating SVD is explained in detail in [72]

As **Σ** is a rectangular diagonal matrix, it can be simplified as a square matrix of size Min{*m,n*}×Min{*m,n*}. Hence, **U** is reduced to an *m*×Min{*m,n*} matrix, and **V**$^{\mathbf{T}}$ is reduced to an Min{*m,n*}×*n* matrix. The following figure illustrates the size and the format of the matrix in SVD (with *m* > *n* in this example).

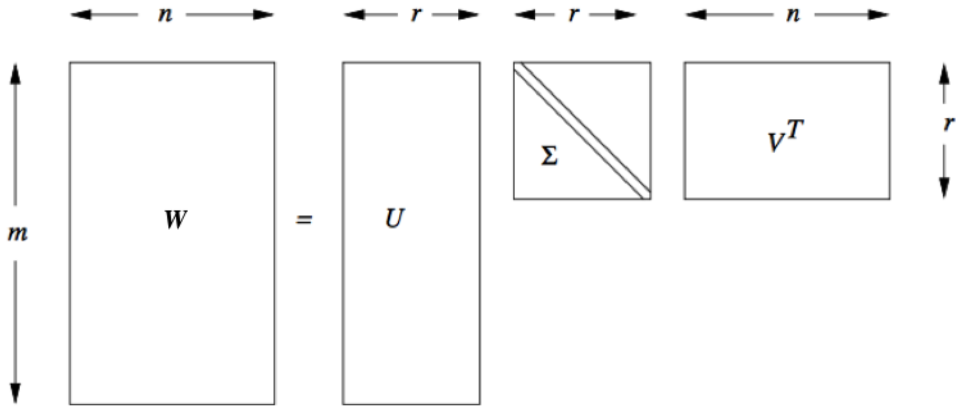


Figure 3.2 SVD format

$$\mathbf{S} = \mathbf{U\Sigma V^T} = \begin{bmatrix} \frac{\sqrt{10}}{10} & 0 \\ \frac{-\sqrt{10}}{10} & \frac{2}{3} \\ \frac{\sqrt{10}}{5} & \frac{2}{3} \\ \frac{-\sqrt{10}}{5} & \frac{1}{3} \end{bmatrix} \times \begin{bmatrix} \frac{\sqrt{15}}{6} & 0 \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} \frac{-\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} \\ 0 & 0 \\ \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} \end{bmatrix}^T \tag{27}$$

Step (5) The principle coordinates of row is **X** = **D$_r^{-1/2}$UΣ** and the standard coordinates of the column is **Y** = **D$_c^{-1/2}$V.**

$$\boldsymbol{X} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \times \begin{bmatrix} \frac{\sqrt{10}}{10} & 0 \\ \frac{-\sqrt{10}}{10} & \frac{2}{3} \\ \frac{\sqrt{10}}{5} & \frac{2}{3} \\ \frac{-\sqrt{10}}{5} & \frac{1}{3} \end{bmatrix} \times \begin{bmatrix} \frac{\sqrt{15}}{6} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{6}}{6} & 0 \\ \frac{-\sqrt{6}}{6} & 0 \\ \frac{\sqrt{6}}{3} & 0 \\ \frac{-\sqrt{6}}{3} & 0 \end{bmatrix} \tag{28}$$

$$\boldsymbol{Y} = \begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix} \begin{bmatrix} \frac{-\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} \\ 0 & 0 \\ \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} \end{bmatrix} = \begin{bmatrix} \frac{-\sqrt{6}}{2} & \frac{-\sqrt{6}}{2} \\ 0 & 0 \\ \frac{\sqrt{6}}{2} & \frac{-\sqrt{6}}{2} \end{bmatrix}$$

$$\text{so } \boldsymbol{X} = \begin{bmatrix} \frac{\sqrt{6}}{6} & 0 \\ \frac{-\sqrt{6}}{6} & 0 \\ \frac{\sqrt{6}}{3} & 0 \\ \frac{-\sqrt{6}}{3} & 0 \end{bmatrix} \text{and } \boldsymbol{Y} = \begin{bmatrix} \frac{-\sqrt{6}}{2} & \frac{-\sqrt{6}}{2} \\ 0 & 0 \\ \frac{\sqrt{6}}{2} & \frac{-\sqrt{6}}{2} \end{bmatrix} \tag{29}$$

Step (6) According to the Eckhart and Young's theory [73], the vector corresponding to the larger eigenvalue has the more information for the original data set. In Eq.(27), the first column of **U** and the first row of **V** corresponding to the largest eigenvalue $\frac{\sqrt{15}}{6}$. If we plot it in 1D, the first column of **U** and the first row of **V** will be kept for row coordinates and column coordinates respectively. Similarly, if we need to plot in a 2D space, only the first two dimensions with the largest eigenvalue will be plotted for both row coordinates and column coordinates. In this case, we only have two dimensions, because contingency table with three dimensions can be perfectly plotted in two dimensions according to [74]. Therefore, we plot both two rows and two columns.

The points representing the first to fourth rows will be $\left(\frac{\sqrt{6}}{6}, 0\right)$, $\left(\frac{-\sqrt{6}}{6}, 0\right)$, $(\frac{\sqrt{6}}{3}, 0)$ and $(\frac{-\sqrt{6}}{3}, 0)$ respectively. Similarly, the points of the first to third column will be $\left(\frac{-\sqrt{6}}{2}, \frac{-\sqrt{6}}{2}\right)$, $(0,0)$, and $(\frac{\sqrt{6}}{2}, \frac{-\sqrt{6}}{2})$.

Step (7) Plot the points according to their coordinates in the 2D row principle plot

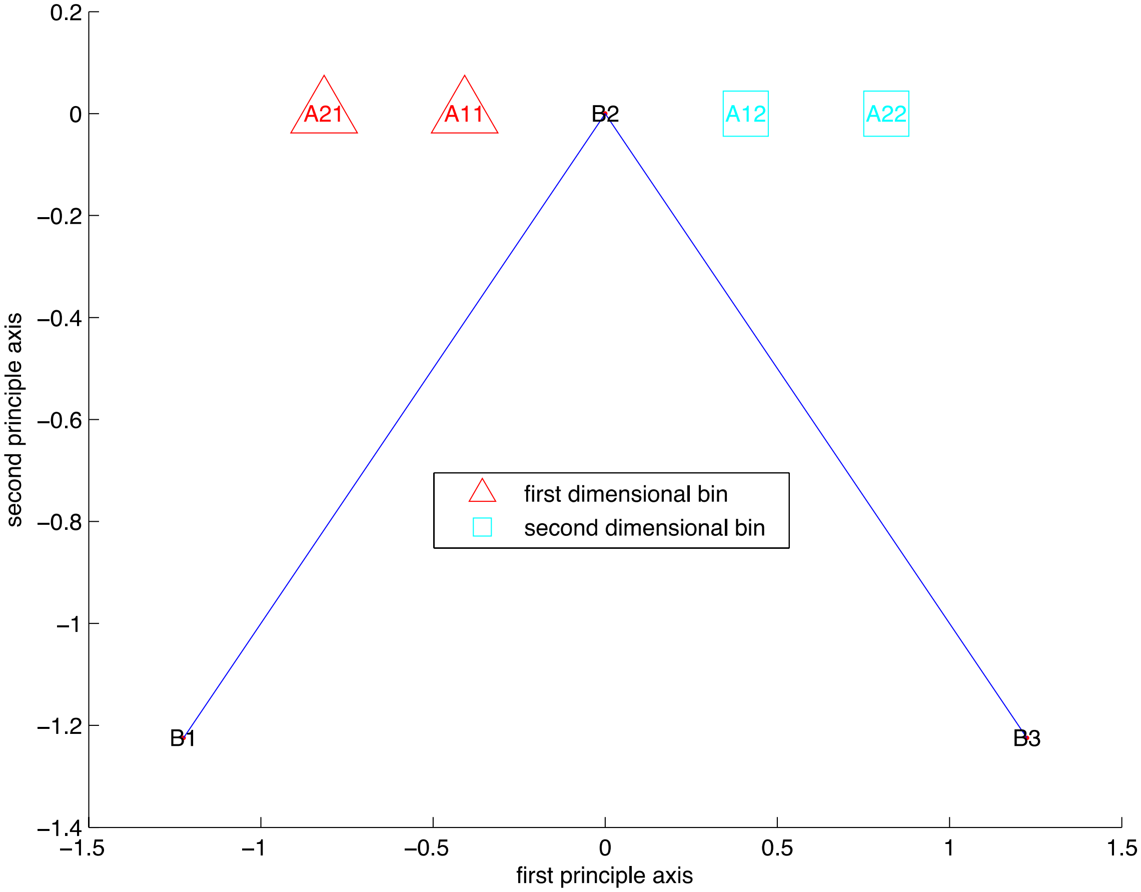


Figure 3.3 Row principle plot of the example

In the figure above, the label A_{ij} represent the j^{th} dimensional bin for the i^{th} variable. The circle and the triangle points show the first dimensional bin and the second dimensional bin for all different variables respectively. The vector from the origin (0,0) to the point represents the i^{th} function value bin is denoted as B_i. It serves to be the reference to compare the relationship among dimensional bins in the figure. The position of B_i is the same as the i^{th} row of matrix **Y**. Therefore, B_1, B_2 and B_3 should be $\left(\frac{-\sqrt{6}}{2},\frac{-\sqrt{6}}{2}\right)$, $(0,0)$, and $(\frac{\sqrt{6}}{2},\frac{-\sqrt{6}}{2})$ respectively. According to the example, the first dimension is x, therefore, the point A_{11} $(\frac{-\sqrt{6}}{6},0)$, A_{12} $(\frac{\sqrt{6}}{6},0)$, A_{21} $(\frac{-\sqrt{6}}{3},0)$ and A_{22} $(\frac{\sqrt{6}}{3},0)$ represent the first row profile A_{x1}, the second row profile A_{x2}, the third row profile A_{y1} and the fourth row profile A_{y2} in Table 3.4 respectively. The detail explanation of interpreting the figure will be discussed in the following section.

3.5. Proof

There are some presumptions of the problem for which CA can be applied to. As violating the assumptions may lead to inaccurate conclusions. In this section, we will prove that the way we representing the optimization problem can satisfy the requirements of applying CA. The assumptions of CA are listed below:

1. CA assumes that the data being analyzed is discrete; however, originally continuous variables can be categorized into discrete variables.
2. All values in the frequency table must be non-negative so that the distances between the points on the plot are always positive.
3. The statistical properties are similar across rows and columns, implying that there must not be any empty variables whose entries are all zeros.

For our approach of reforming matrix of the optimization problem, the assumptions are addressed as below:

1. We categorized continuous variables into discrete variables. For example, we discrete the range of each variable of design space into dimensional bins with equal length. And the performance space (the range of function value) is discretized into function value bins with the same number of points in each bin.

2. The elements in contingency table represent the number of the points that fall into certain dimensional bin and function value bin, therefore, the value of each element will always be non-negative.

3. As the total sample points can be divided exactly by the production of number of rows (*s*) and columns (*t*), the Latin Sampling ensures that

> there are at least *t* points sampled in each dimensional bin. Therefore, the total sum of each row will be greater or equal to *t*. Also all the elements in each row are non-negative and their sum is larger than zero, so there must be at least one element greater than zero. Similarly, the elements in each column are non-negative and the sum of each column is greater than zero (equal to the number of one third of the sample points), the columns can't have all zeros elements. As a result, we can ensure that none of the column or row has all zeros.

The distribution of data in the proposed method may vary. But CA does not make any assumption on data distribution. Therefore CA can be applied directly on the contingency table that we designed.

The next step we still need to justify that during the process of CA, the relative magnitude of each row and column remains the same. That is to say, if a row profile [a, b, c] with a>b>c shows the distribution across the function value, this a>b>c property won't change after the process. It means promising subdomains suggested by the original table won't change either after CA. This is important for the purpose of optimization. As the correspondence matrix that we construct contains some unique properties that are not in the regular correspondence matrix, we can use these properties to illustrate that during the process of CA, the matrix remains meaningful in the context of optimization.

For the standardization step in CA, the contingency table matrix **W** is transformed to correspondence matrix P by dividing every element by the grand total. Then P is standardized to matrix S, where $\mathbf{S} = \mathbf{D}_r^{-1/2}(\mathbf{P}-\boldsymbol{rc}^T)\mathbf{D}_c^{-1/2}$. The elements in column mass profile are always the same as we classify the points to function value bins with equal number of points. In addition, the properties of LHS ensure that the values in row mass profile are also the same because the length of the dimensional bins is the same for a certain variable. Therefore, every element in $\boldsymbol{rc}^T$ is the same, so does the value in the diagonal of $\mathbf{D}_r$ and $\mathbf{D}_c$. Every element in the contingency table will be subtracted and then divided by the same value, and then the magnitude relation remains the same among all the elements in matrix. Therefore, the information in row vector is preserved, which can reveal the subdomain properties of the distribution across the function value space. In other word, the process of CA doesn't change the relative magnitude of the rows and columns, and thus the outcomes from CA can be directly used to support optimization.

3.6. Interpretation of the Plot in Optimization Context

The important information we can discover from the graph is listed below:

1. Identify promising subdomain:

Intuitively, for each variable, the subdomain containing the optimum is called its promising subdomain, which is more likely containing more "good points" whose function values are smaller among all the points. Therefore, for a promising subdomain, there tends to be more points falling into the first function value bin than the other two function value bins (classified into three function value bins in total). Then in the context of CA, if the i^{th} subdomain (the i^{th} dimensional bin) for the k^{th} variable is promising, the corresponding row profile $\vec{x}[n_{ki1}, n_{ki2}, n_{ki3}]$ will tend to have larger n_{ki1} than n_{ki2} and n_{ki3}. Conversely, if a row profile $\vec{x}$ has its n_{ki1} larger than n_{ki2} and n_{ki3}, its corresponding subdomain is more likely to be promising. In general, the larger n_{ki1} is, the more promising the corresponding subdomain is. According to CA, the projection of the i^{th} row profile vector of the k^{th} variable (A_{ki}) onto the j^{th} reference vector (B_j) can reveal the value of the cell n_{kij} in the contingency table. To ensure the first element of the row profile is larger than the second and the third element, we required that the projection of A_{ki} onto B_1 should be larger than projection onto B_2 or B_3. As the length of A_{ki} is fixed, if the angle between A_{ki} and B_1 is larger than the angle between A_{ki} and B_2 or A_{ki} and B_3, a larger projection onto B_1 is also ensured. The larger projection a point has onto B_1, the more promising the subdomain is. In other words, the points with larger projection onto B_1 as well as smaller angle between B_1 are what we want. They will indicate the promising subdomains of the variables.

2. Find the significant variables

If a variable is influential to the function value, a variance in its value will result in a significant change in the performance space (function value) and we call it significant variable. As we subdivide the domain of a variable into *s* subdomains as its dimensional bins, the distribution of the function value bins of each subdomain can be used to decide how significant a variable is to the function value. For a given variable, if the distributions of function values are similar for all the three subdomains, then this variable is having a small impact on the function value, because the change of its value results in little change of the function value. If the distributions vary a lot, e.g., small function values for the first domain and large function values for the second domain, it means that the variable is influential to the function value, whereas the change of its value results in a significant change of the function value. Such property in the contingency table can be discovered by inertia. According to CA, a smaller value of the total inertial means that the rows are more likely to be homogenous. In other words, the row profiles are close to each other, having smaller variances from the average row profile. Otherwise, the row profile vectors are different from the average row profile. The row profiles for an influential variable have more variety than the row profiles for a non-influential variable. Therefore, the total inertial of row profiles for an influential variable should be larger. This can also be learned from the graph easily. Considering the i^{th} variable x_i, the total inertia λ_i is the summation of the distances from A_{ij} (j = 1,2…*s*) to the origin. If the row profile vectors of a variable are away from the origin, we can say it is a significant variable to the function. One can calculate and show the total inertial of each variable to assist discovering significant variables. The larger the inertial is, the more influence is that variable to the output function value. In other words, the more significant that variable is.

3. Find the symmetric variables:

Assume that the black box function is represented by F = F(x_1,x_2,…x_d), where *d* is the number of variables (dimensionalities). F is a partially symmetric function if and only if it is unchanged when some, but not all, of the variables are permuted. Those permuted variables are called symmetric variables. Suppose there is a set of symmetric variables within the partial symmetric function. For each subdomain, the symmetric variables will tend to have similar distributions across the function values. Therefore, the corresponding row profiles of these variables should also be similar. In the plot, the row profile vectors of such variables are usually clustered by subdomains. If we observe that there is a group of variables, where each subdomain's corresponding row profile vectors are clustered in the plot, we can say they are symmetric variables of the function. To be more precise, we can call them pseudo-symmetric variables, since these variables might not be the actual symmetric variables, based on the limited samples.

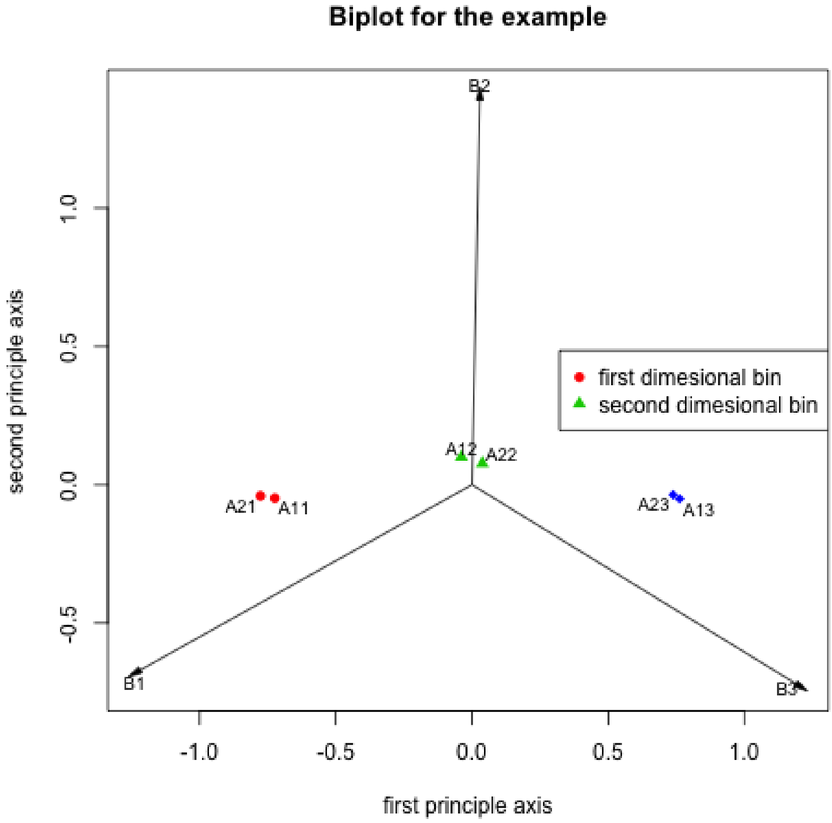


Figure 3.4 Row principle plot for "x+y" function

For the previous function f(x,y) = x+y, if we include more samples, choose three dimensional bins for each variable and three function value bins, the plot will be like Figure 3.4:

1. Identify promising subdomain

We consider the variable *x* as the first dimension and the variable *y* as the second dimension. The angle between the first dimensional bin vector of *x* (A_{11}) and the first function value bin B_1 are smaller comparing to the angle between B_1 and other dimensional bins of *x* (A_{12} and A_{13}). In addition, the A_{11} is closer to B_1 than to the other function value bins (B_2 and B_3). So is A_{21}, indicating the same for the *y* variable. It means that the first bin for both *x* and *y* are more likely to have the optimal than the other bins in their dimension. Back to the function *x+y*, as we are pursuing the minimal function value, the first dimensional bin, which indicates the smaller value, is more promising. The graph shows exactly that as the hypotheses.

2. Find the significant variables

In Figure 3.4, the total distance from the row profile vectors (A_{11}, A_{12} and A_{13}) of variable *x* to the original point are denoted as D_x. Similarly, D_y is defined as the total distance from the row profile vectors (A_{21}, A_{22} and A_{23}) of variable *y* to the original point. In the figure, D_x is similar to D_y, which means that *x* and *y* have the same influence to the output function value. As the coefficient of the *x* and *y* are the same in the function, they will have the same influence to the function value. The assumption that *x* and *y* have the same influence is true in this case.

3. Find the symmetric variables

The respective dimensional bins in *x* and *y* dimension are in the same cluster and close to each other. In Figure 3.4, it is obvious that, the first (second, third) dimensional bin A_{11} (A_{12}, A_{13}) for *x* variable is close to the first dimensional bin A_{21} (A_{22}, A_{23}) for *y* variable, therefore, we might assume that the dimensions *x* and *y* are symmetric variables. Looking back at the sample, in function *x+y*, if exchanging *x* and *y*, the function remains the same. Therefore, *x* and *y* are indeed symmetric variables in the function as CA predicted.

3.7. Guidelines of choosing parameters

Parameters *n*, *s* and *t* are to be specified by users. The guidelines of choosing the value of parameters are discussed below:

For the visualization purpose, the number of bins for each dimension is suggested to choose between three and four. More than five dimensional bins for each variable will make the figure hard to see, especially for problems of more than ten dimensions. Most importantly, as the number of dimensional bins increases, more points are needed to sample in each bin to achieve the statistical significance of the propose method. As we want to reduce the number of samples, we intend to have less dimensional bins.

The proposed method can find the mapping relationship between the dimension bins and function values to identify the promising subdomains. However, if we only cut the domain of a variable into three or four bins, the mapping relationship we found is not accurate because the range of the design space is too large. This problem can be solved because the method can be applied iteratively. If we want to find more accurate mapping on a smaller range, we can first analyze the mapping of the large range and then cut the promising range into smaller pieces and analyze it again in a smaller range. There's no need to cut the range into small pieces at the first stage. Once we find an interesting area, the smaller range of the bins can be generated for a certain subdomain. Therefore, cutting the domain into three or four dimensional bins is sufficient.

The function value bin is suggested to be three. According to the proof in [13], an N dimensional profile can be perfectly shown in N-1 dimension without losing any information, therefore, a 3-D profile can be expressed in 2-D. Three bins for function values can ensure that the matrix has three columns and can be shown in two dimensions. No information will be lost during the SVD process to show it in two dimensional plots. This will make the visualization more accurate than cutting it into more than three functional bins.

For the other parameter n in the algorithm, it should be able to be divided exactly by the production of s and t, where s is the number of dimensional bin and t is the number of function value bins.

3.8. Tests and Results

After demonstrating the details of the method, we will focus on the experiments and results to prove the capability and practicality of this method. Generally, there will be two sub-sections. In the first section, we will test the method with three simple functions. These three simple design functions are shown as an example to demonstrate correctness of three properties of the proposed method. The functions are listed in Table 3.6.

Table 3.6 Design function for proposed method

Function name	Dimension (Number of variables)	Formula	Search domain	Minimum
Sphere function	3, 30	$min\, f(x) = \sum_{i=0}^{n} x_i^2$	$0 \le x_i \le 1$	$x^* = (0, \ldots, 0)$ $f(x^*) = 0$
Low correlation function	30	$\min f(x) = x_1x_2 + 2x_3x_4 + 3x_5x_6$ $x_7\ to\ x_{30}\ are\ redundant\ variables$	$0 \le x_i \le 1$	$x^* = (0, \ldots, 0)$ $f(x^*) = 0$
High correlation function	30	$min\, f(x) = x_1x_2x_3x_4x_5x_6$ $x_7\ to\ x_{30}\ are\ redundant\ variables$	$0 \le x_i \le 1$	$x^* = (0, \ldots, 0)$ $f(x^*) = 0$

In this experiment, 90 sampling points are used for three dimension problems and 900 points are used for thirty dimension problems. Each variable will be divided into three dimensional bins, bin 1 (bin 3) corresponding to the smallest (largest) value region. The dimensional bin vector $\overrightarrow{\boldsymbol{V}_{\boldsymbol{p}_{ij}}}$ is from (0, 0) to the point $\boldsymbol{P_{ij}}$ that represents the *j*th dimensional bin of the *i*th variable. In the graph, the triangle, square and diamond marker represents the first, second and third dimensional bin respectively for certain variable. The number *i* inside the marker represents the *i*th variable which the dimensional bins belong to. Combination of markers and numbers can represent the dimensional bin $\boldsymbol{P_{ij}}$, for example, the first dimensional bin for the first variable P_{11} is represented by the triangle with number 1 in it. Similarly, we divided the points into three bins according to the function value. The function value bin vector $\overrightarrow{\boldsymbol{V_{f_i}}}$ is from (0, 0) to the point $\boldsymbol{V_i}$ that represents the i^{th} function value bin. In the graph, $\boldsymbol{V_i}$ is represented by the number *i* with parentheses around it ("(*i*)") and the vectors are shown as blue lines in all figures below.

- Sphere function

As it is shown in Figure 3.5, the first dimensional bin vector (triangle markers) for each variable has a smaller angle between the first function value bin vector $\overrightarrow{\boldsymbol{V_{f_1}}}$ than between the second and third function value bin vector $\overrightarrow{\boldsymbol{V_{f_2}}}$ and $\overrightarrow{\boldsymbol{V_{f_3}}}$. It indicates that the first dimensional bins are the most promising subdomain, which is true because when x_i is in region [0, 1/3], the function value is smaller. Also the coefficient for each item is the same, which means that all the dimensions have the same influence on the function value. Therefore, the variables should have the similar total inertia, which are shown in the graph, as the total distances from the dimensional bins to the original point for a certain variable are similar. The inertia is also shown have similar value in Table 3.7.

For the formula, every two x_i and x_j is exchangeable, so the function is a symmetric function and all the variables are symmetric variables. Thus, the corresponding dimensional bin should be close to each other. Figure 3.5 shows all the triangle are close to each other and so do squares and diamonds, which represents that the first dimensional bins for the corresponding variables are in the first cluster, the second dimensional bin in the second cluster and the third dimensional bins in the third cluster. It is the same as predicted. Figure 3.6 obtains the same results for the sphere function when the dimension increases to thirty. To make the figure more clearly for a thirty-variable problem, the numbers, which ought to be inside the marker to indicate certain variable, are omitted. The inertia of each dimension is shown in Table 3.8.

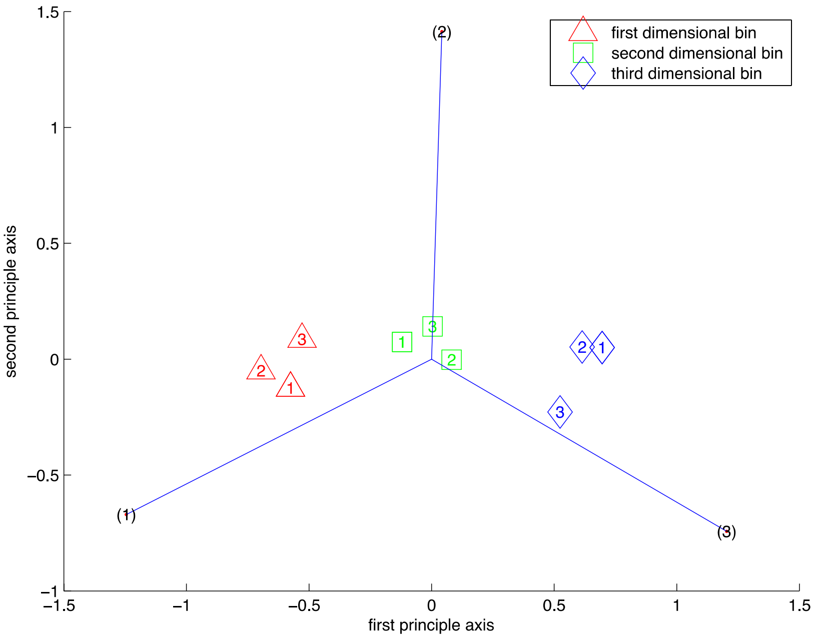


Figure 3.5 3D sphere function

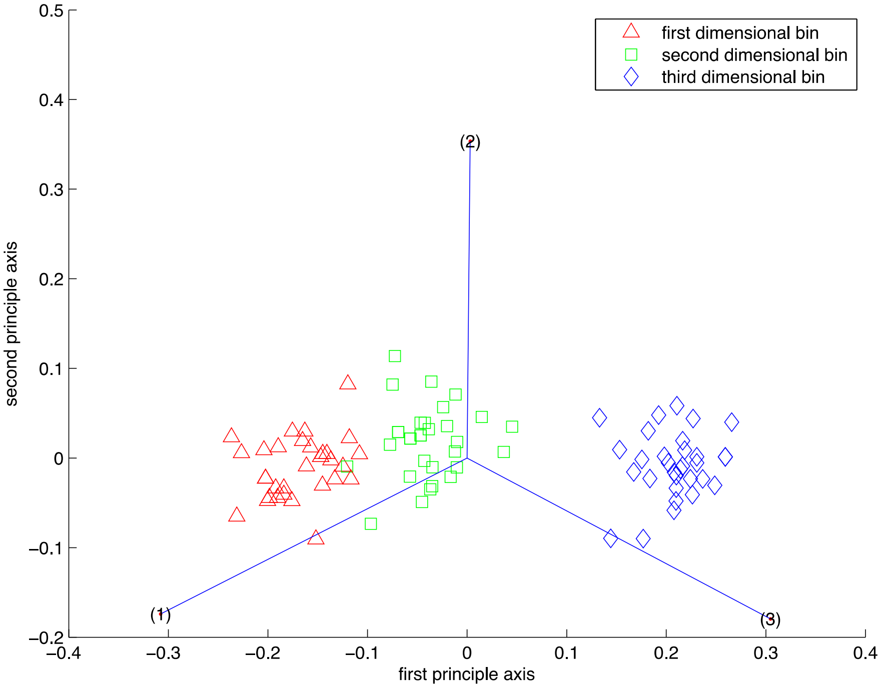


Figure 3.6 30D sphere function

Table 3.7 The total inertial for each variable in 3D sphere function

	1st variable	2nd variable	3rd variable
Total inertial	1.37866893644938	1.20988694009540	1.33733529986230

Table 3.8 The total inertial for each variable in 30D function

	30D Sphere	30D Low correlation function	30D High correlation function
1st variable	0.398642862097115	***0.493643581226953***	***0.835540255491546***
2nd variable	0.484720387718703	***0.498139313046960***	***0.927896979695465***
3rd variable	0.420543849835798	***0.921118467868212***	***0.831809864196272***
4th variable	0.485415102617012	***0.709203377429272***	***0.813766710869774***
5th variable	0.426955589348830	***1.29523758836623***	***0.862299092505617***
6th variable	0.439154947306499	***1.33954262346104***	***0.751844229877121***
7th variable	0.393949532882725	0.0684044678555329	0.110138235790603
8th variable	0.476211967266988	0.180118822034004	0.172100481075421
9th variable	0.370744490739038	0.184946899831907	0.137247146198704
10th variable	0.431989621915272	0.279793526790522	0.0976103154324033
11th variable	0.355630816839004	0.241051864992156	0.139152493789324
12th variable	0.482509857836359	0.126451547013768	0.123115098662543
13th variable	0.385117157641772	0.114347490499738	0.158984725425672
14th variable	0.404031546685630	0.137776272522066	0.164386827705905
15th variable	0.487826854466356	0.124198132249737	0.191570768262407
16th variable	0.456250775247658	0.123434677304864	0.101330362355032
17th variable	0.399769567190802	0.294090645813407	0.0657351284914572
18th variable	0.430007045071294	0.101585740544497	0.154099432484326
19th variable	0.405756470865653	0.140025530389000	0.136868664770325
20th variable	0.508793282765993	0.200307534799441	0.142035188237546
21st variable	0.330757664815854	0.0304720672422855	0.114457306157870
22nd variable	0.597539281471514	0.150190896478542	0.0979795897113270
23rd variable	0.393110034156727	0.268400820999151	0.189643729889319
24th variable	0.485983535137494	0.108182924970725	0.0894039928977514
25th variable	0.493640058748961	0.215552993256273	0.162777636339591
26th variable	0.426381691609315	0.143773185810829	0.0614536408207023
27th variable	0.362027512649674	0.286626907215220	0.0780299105633014
28th variable	0.339287513893566	0.161262112840337	0.172224377254898
29th variable	0.419983518659428	0.180981210511452	0.208957522256008
30th variable	0.360594317400335	0.0979795897113269	0.145841701998184

- Low correlation function

The coefficients of each variable in this low correlation function are not the same, therefore, the variables will have different influences on the function value. In the formula, the 5th and 6th variables have the largest coefficient, 3rd and 4th variables have the second largest coefficient, and the 1st and 2nd variables have the smallest coefficient. The larger coefficient means a larger influence on the function value. According to the function, the 5th and 6th dimensions have the most influence on the function value and the 7th variable to 30th variable have no influence on the function value. In Figure 3.7, the dimensional bins for the fifth and sixth variables ($\overrightarrow{\boldsymbol{V}_{\boldsymbol{b}_{\imath\jmath}}}$, i = 5, 6, j = 1, 2, 3), which represented by the marker with number 5 or 6 inside, are further apart. As it is shown in the second column of Table 3.8, the inertia of the fifth and sixth dimension is much larger than the other dimensions, which indicates the larger influence on the function value. It can be easily observed that besides the most significant variables (fifth and sixth variables), the third and fourth variables are more influential than the first and the second variables. The inertias of first to sixth dimensions are highlighted in bold and italic style in order to distinguish them from the other dimension with less influence on function value. The other variables (the 7th to 30th) have least inertia, which means they have least influence on the function value. In Figure 3.7, the points that represent the dimensional bins vectors $\overrightarrow{\boldsymbol{V}_{\boldsymbol{b}_{\imath 1}}}$, $\overrightarrow{\boldsymbol{V}_{\boldsymbol{b}_{\imath 2}}}$ and $\overrightarrow{\boldsymbol{V}_{\boldsymbol{b}_{\imath 3}}}$ (*l*=7, 8, …, 30) for 7th to 30th variables are close to each other, which make the numbers inside the makers unable to be seen. Therefore, in order to get the clear plot, none of the numbers will be plot. These dimensional bins for 7th to 30th variables are close to the origin, which also indicates they have the least influence on the function. Thus, the previous prediction is correct. Vector $\overrightarrow{\boldsymbol{V}_{\boldsymbol{b}_{51}}}$ and $\overrightarrow{\boldsymbol{V}_{\boldsymbol{b}_{61}}}$, $\overrightarrow{\boldsymbol{V}_{\boldsymbol{b}_{52}}}$ and $\overrightarrow{\boldsymbol{V}_{\boldsymbol{b}_{62}}}$and $\overrightarrow{\boldsymbol{V}_{\boldsymbol{b}_{53}}}$ and $\overrightarrow{\boldsymbol{V}_{\boldsymbol{b}_{63}}}$ are close to each other, which indicates they have symmetric property. We can guess according to the figure that these two variables are symmetric variables. Similarly, we can discover that the first and second variables are symmetric variables and so do the third and the fourth variables. It is correct because in the formula, those two variables are exchangeable. In the graph, the promising subdomain for each variable is still the first dimensional bin [0,1/3], which is represented by the triangle marker in the graph. This can be easily verified, as in the low correlation function, the region with smaller variable value tends to have smaller function values. Thus, the first dimensional bin is more promising.

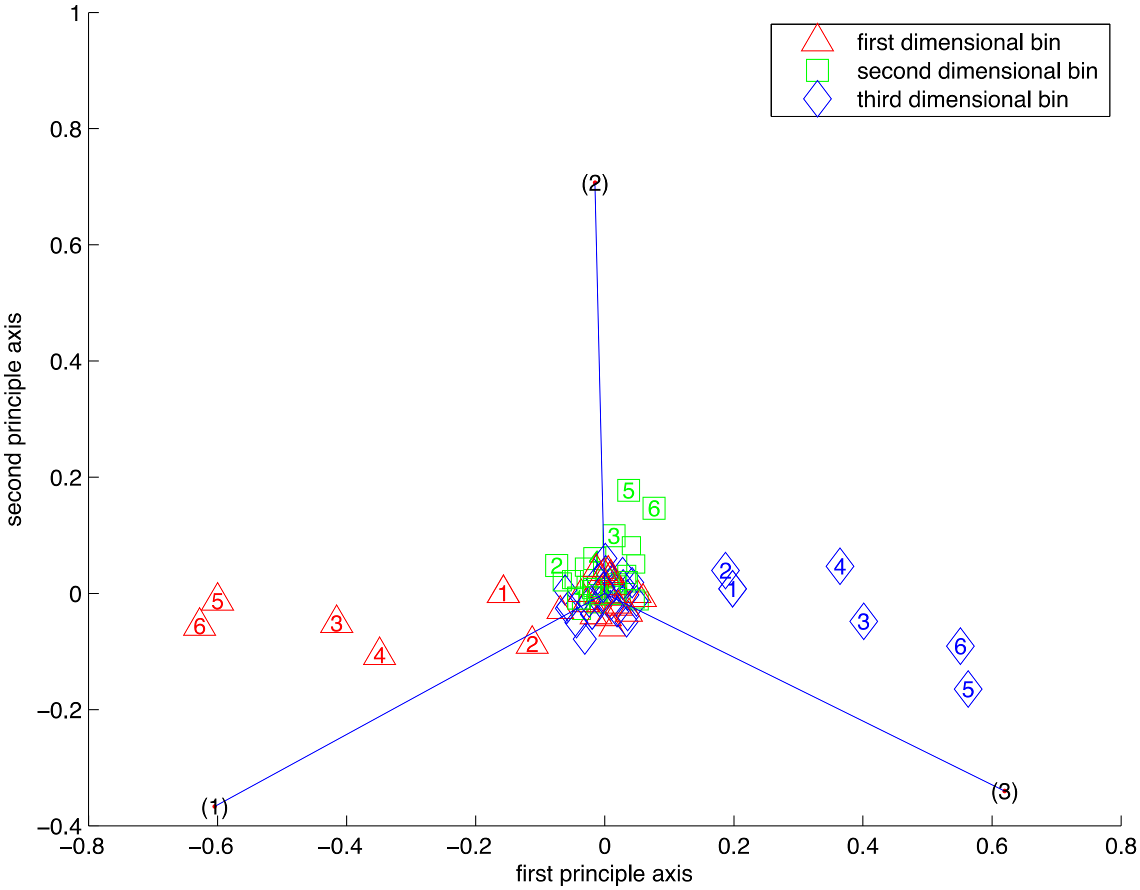


Figure 3.7 30D low correlation function

- High correlation function

For the high correlation function, the first dimensional bins for the 1st to 6th variables are shown to be the most promising subdomain. In Figure 3.8, the triangle markers with numbers 1 to 6 inside have large projection onto the first function value bin, which implies that the first dimensional bin for 1st to 6th variables is a promising subdomain. The 1st to 6th variables have similar influence on the function value while the 7th to 10th dimensions have no influence on function value, same as the formula indicates. In the figure, the dimensional bins ($P_{ij,}$ i = 7, 8, …, 30, j = 1, 2, 3) for the 7th to 30th variables that have no number inside the marker are all close to the origin. It indicates that changing in the 7th to 30th dimension won't result in much difference in function value. Therefore, the 7th to 30th variables are not significant variables. The dimensional bins for the 1st to 6th variables are separated, which suggest they have larger influence on the function value. The inertia in the third column of Table 3.8 also indicates the 7th to 30th is less influential than the 1st to 6th variables. The inertias of 1st to 6th are also boldfaced so can be seen more clearly. The first dimensional bins for the 1st to 6th variables (triangles with numbers inside) are in the same cluster and so do the second (square with numbers inside) and third dimensional bins (diamond with numbers inside). We can assume that these six variables are symmetric variables, which can be easily proved according to the formula.

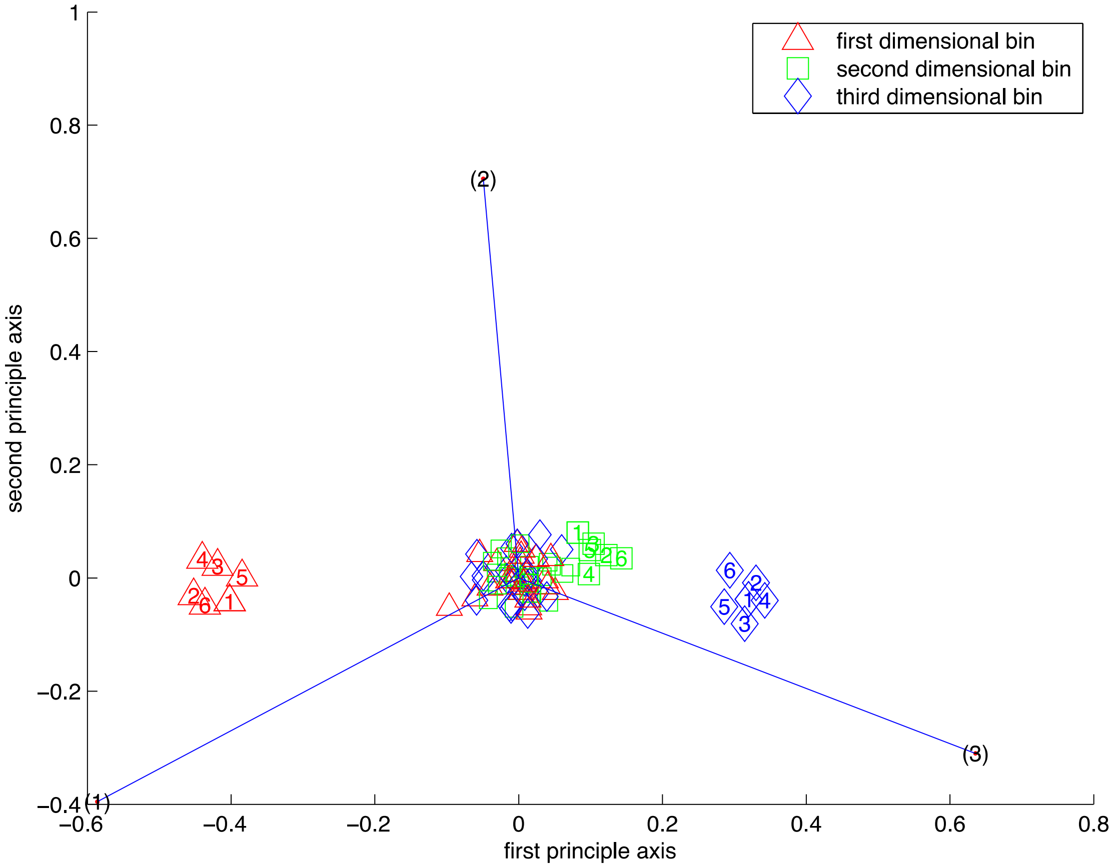


Figure 3.8 30D high correlation function

The proposed method has also been tested on a number of well-known benchmark functions and they are listed in Table 3.9 and are selected according to their significant physical properties and shapes.

Table 3.9 Benchmark test functions for proposed methods

Function name	Number of dimension	Formula	Search domain	Minimum
Ackley (Many local minimal)	2,30	$f(x) = -ae^{-b\sqrt{\frac{1}{d}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^{n}\cos(cx_i)} + a + e$ a=20,b=0.2 and c=2π	$-32.768 \leq x_j \leq 32.768,$ $i = 1,2, \ldots, n.$	$x^* = (0, \ldots, 0); f(x^*) = 0$
Rosenbrock function (Valley-shaped)	2,30	$f(x) = \sum_{i=1}^{n-1}\left[100\left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2\right]$	$-10 \leq x_j \leq 10,$ $i = 1,2, \ldots, n.$	$x^* = (1, \ldots, 1);\ f(x^*) = 0$
Griewank function (Many local minimal)	2,30	$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(x_i/\sqrt{i}\right) + 1$	$-600 \leq x_i \leq 600,$ $i = 1, \ldots, n.$	$x^* = (0, \ldots, 0); f(x^*) = 0$

In this experiment, 60 sampling points are used for two dimensional problems and 900 points are used for 30 dimension problems. Each dimension is divided into three dimensional bins, the bigger dimensional bin number corresponding to the larger values in design space for a certain variable. All the points are divided into three bins according to their function values with the same number of points in each bin.

- Ackley function

As shown in Figure 3.9, the Ackley function is characterized by a nearly flat outer region. It has many local minima and a global optimal at the center. From Figure 3.9, it is easy to see that *x* and *y* have similar influence on the function value. As it is shown in Figure 3.10, the total length of dimensional bin vectors for *x* variable (marked with number 1 inside) is similar to that of the *y* variable (marked with number 2 inside). In addition, the total inertial for *x* variable is 1.5881, which is similar to the total inertial for *y* variable (1.1028). Therefore, *x* variable and *y* variable have similar influence on the function value, which is the same as what the figure and the formula indicate. The promising subdomain for the function is $x \in [-10.92, 10.92]$ and $y \in [-10.92, 10.92]$, which correspond to the second dimensional bin for both *x* and *y* variables. In Figure 3.10, the square markers with number 1 or 2 inside have larger projection on the first function value bin. It implies the second dimensional bins for *x* and *y* variable are promising subdomains. In the formula, exchanging *x* and *y* won't change the function, therefore, *x* and *y* are symmetric variables. In Figure 3.10, squares, triangles, and diamonds are close to each other in separate clusters, which means the corresponding dimensional bins for *x* and *y* variables are in the same cluster. Therefore, *x* and *y* are symmetric variables, which can be seen from the formula.

In Figure 3.11, the dimensional bins are shown by markers without numbers in it to make the plot clear for the 30 variables problem. For the 30D Ackley function, the second dimensional bins are still the most promising subdomains, corresponding to the formula. Figure 3.11 can show this property as the second dimensional bins vectors, which are represented by square markers, have larger projection on the first function value bin vector $\overrightarrow{\boldsymbol{V_{f_1}}}$. All the variables can be permutated without changing the function. Therefore, all the variables are symmetric variables and have similar influence on the function value. This can be shown in Figure 3.11 that the distributions for the corresponding dimensional bins of all the variables are similar. The total inertia for each variable is similar according to the first column in Table 3.10. The range of the inertia is from 0.1 to 0.3, and there is not too much difference between them.

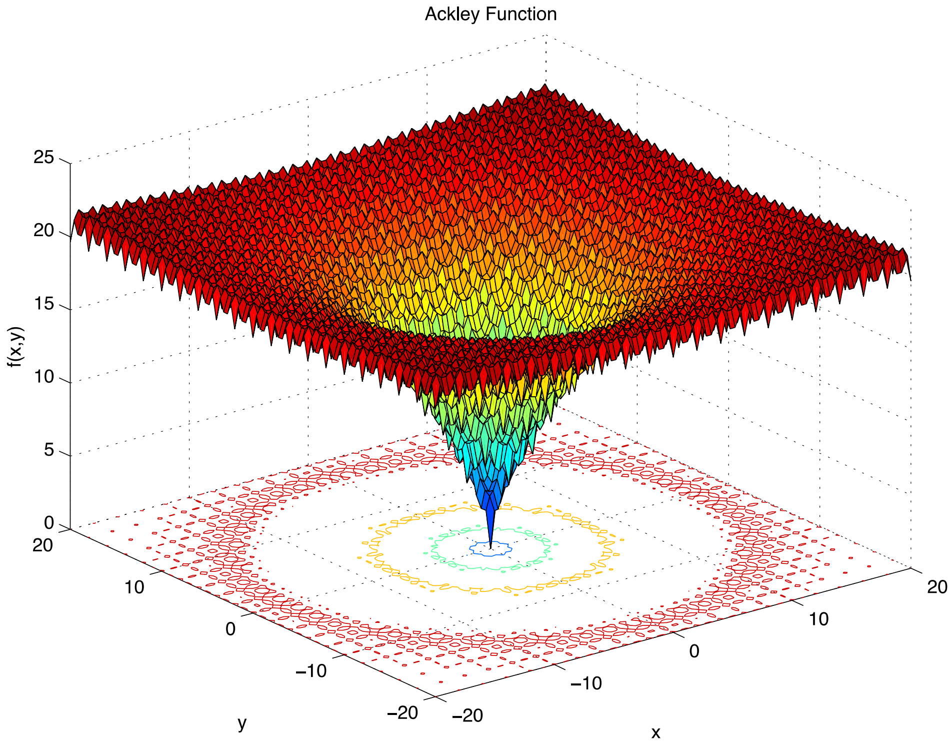


Figure 3.9 Plot of 2D Ackley function

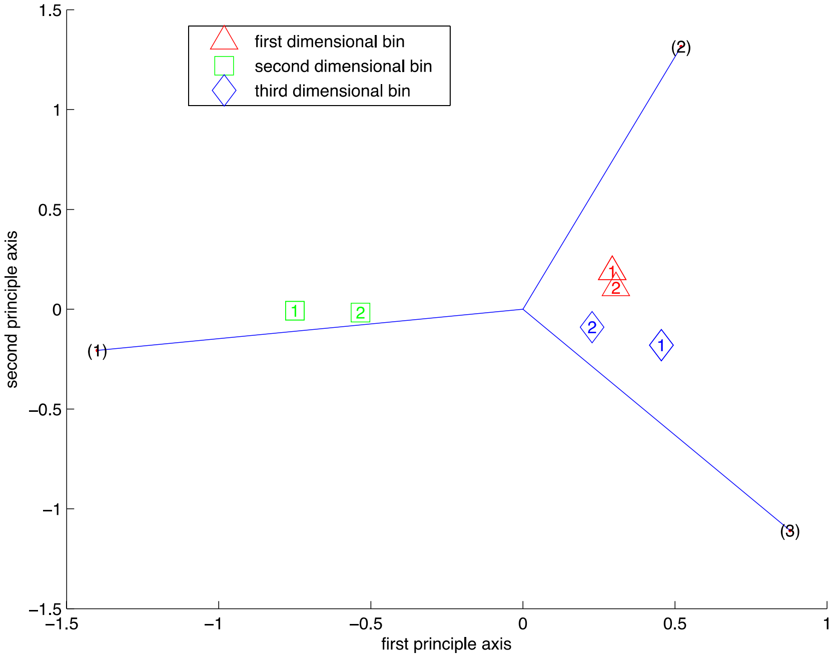


Figure 3.10 2D Ackley function

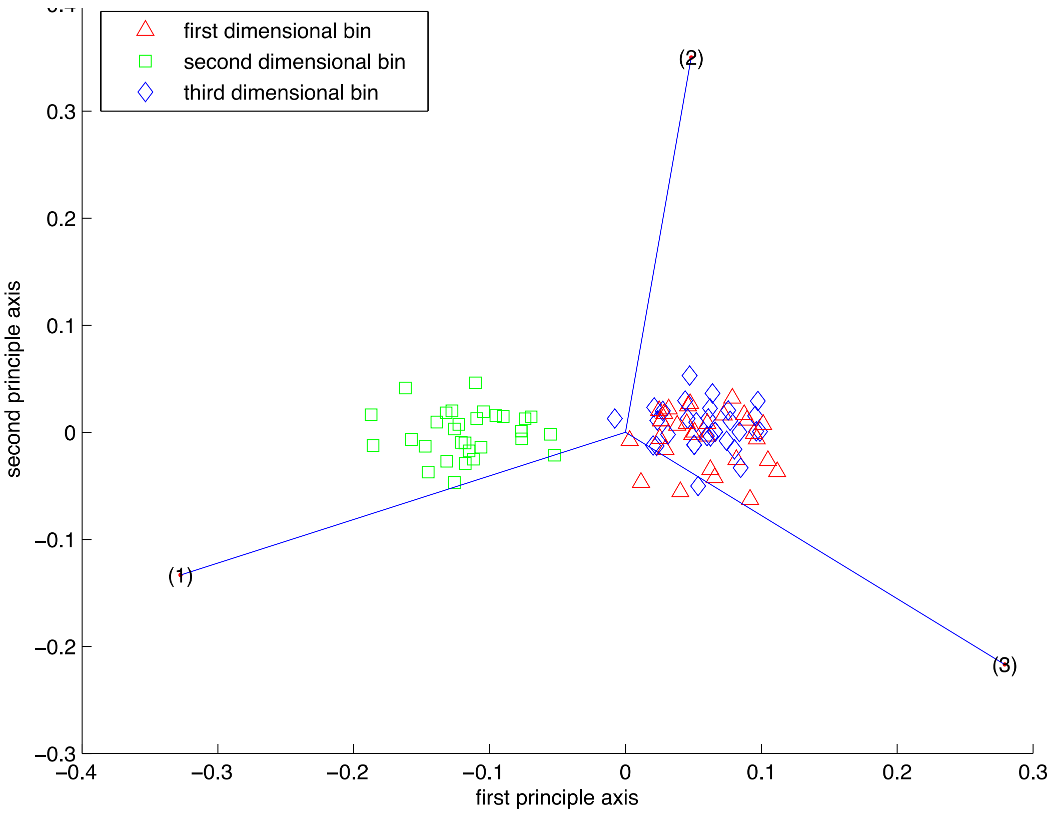


Figure 3.11 30D Ackley function

Table 3.10 The total inertial for each variable in 30D benchmark function

	30D Ackley	30D Rosenbrock	30D Griewank
1st variable	0.156997488975102	0.216043015678609	0.398081568767318
2nd variable	0.372737805959510	0.294798580763886	0.359538555375814
3rd variable	0.265529433497009	0.198129916514252	0.368322988409546
4th variable	0.143862799192994	0.202129728826809	0.349418379196463
5th variable	0.291597742964840	0.266733147156827	0.382822108640321
6th variable	0.272726605250939	0.289800216644367	0.323784588661886
7th variable	0.196088400464651	0.280217677564368	0.374801278814388
8th variable	0.334422323311321	0.202823176957569	0.231889278917752
9th variable	0.232866003856331	0.174686129089212	0.428660659088698
10th variable	0.220689530160963	0.266262389730234	0.188507563501276
11th variable	0.245673854678211	0.247515059730517	0.383666728737046
12th variable	0.239137240875665	0.213090663556394	0.309708076791997
13th variable	0.243043650991654	0.218785535790416	0.315489254481124
14th variable	0.151974033509559	0.239869024596892	0.356504070454649
15th variable	0.273431621562437	0.234730269297719	0.363230995401263
16th variable	0.269655200196547	0.247734198223954	0.352091993727200
17th variable	0.116393507641377	0.212674994456928	0.323878735246277
18th variable	0.232744460731019	0.201264426272304	0.334058089250958
19th variable	0.132519251137372	0.231119626872281	0.311731581117503
20th variable	0.233302015954688	0.275372222341321	0.196480307922627
21st variable	0.321032948389495	0.219444621862407	0.299822470966603
22nd variable	0.196743198527225	0.245048917995712	0.367368566733102
23rd variable	0.383993054640754	0.200708990093284	0.376716546266156
24th variable	0.243923067661769	0.275714373219223	0.375697317151377
25th variable	0.153815496426274	0.218165148667338	0.343528390244800
26th variable	0.248626676993008	0.223987232598559	0.371050933608585
27th variable	0.266805958627875	0.281072842423294	0.323360193888095
28th variable	0.297021726360394	0.212905686321817	0.306302034448605
29th variable	0.304453929989870	0.213295933481147	0.389919361767524
30th variable	0.316810908077524	***0.0794830110973816***	0.280178748145810

- Rosenbrock function

The Rosenbrock function is unimodal, and the global minimum lies in a narrow, parabolic valley. Even though this valley is easy to find, convergence to the minimum is difficult [75]. From Figure 3.12, it is easy to see that when *x* changes from -10 to 10, the function value changes more than *y* changes in the same region. Therefore, the first dimension *x* is more influential than the second dimension *y*. The prediction properties of CA on this function are also shown in Figure 3.13. The dimensional bins for *x* variable (marked with number 1 inside) are further apart from the original points and have larger inertia than the *y* variable. The total inertial for *x* is 2.2937 and larger than 0.6871 for *y*. Therefore, *x* is more influential than *y*. In Figure 3.13, the square marker with number 1 inside, which represents the second dimensional bins for *x*, have larger projection on the first function value bin. It implies the second dimensional bin for *x* is a promising subdomain. In Figure 3.12, we can see that the optimal point is in the second dimensional bin for *x* ($x \in [-1.5, 1.5]$), which means the predicted promising subdomains are correct. In the formula, exchanging *x* and *y* will result in a different function, which means that *x* and *y* are not symmetric variables. From Figure 3.13, no symmetric variables can be found either.

When the dimension increased to thirty, the second dimensional bins for the 1st to 29th variables are still the most promising subdomain, which we can see from the formula. In Figure 3.14, the dimensional bins for the 2nd to 29th variables are plot by markers only. The 1st and 30th variables' dimensional bins have numbers inside the markers. Figure 3.14 shows that the second dimensional bins, represented by square markers, have larger projection on the first function value bin vector $\overrightarrow{\boldsymbol{V_{f_1}}}$. According to the formula, the 30th variable only appears in the term $100(x_{29}^2 - x_{30})^2$ while other variables appear in more terms. Therefore, the 30th variable has different performance comparing to other variables. It leads to the distribution of the dimensional bins of the 30th variable on the graph being different from the pattern of other variables. Also the 30th variable is less influential than the other variables, which can be shown in Figure 3.14 that the dimensional bins for the 30th variable are closer to the origin. The total inertia for the 30th variable is also much less than the others according to the second column of Table 3.10. The inertia of the 30th variable is shown in boldface and italic style.

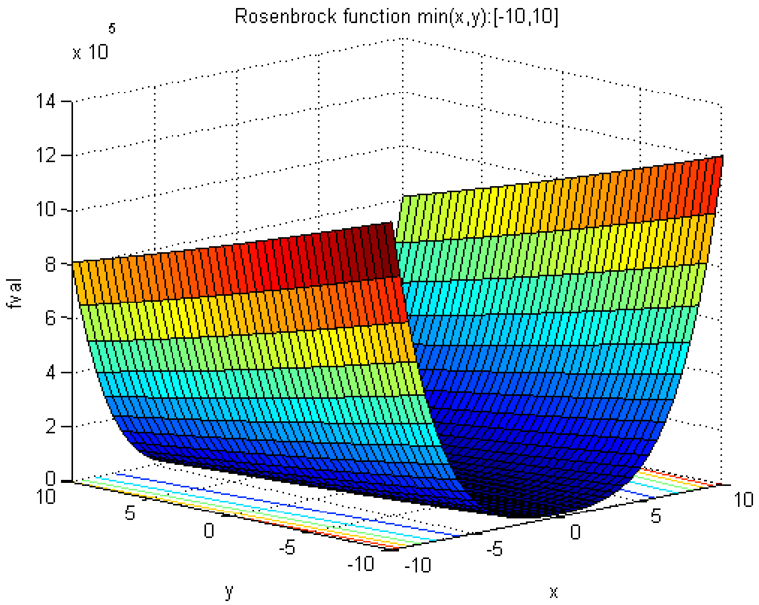


Figure 3.12 Plot of 2D Rosenbrock function

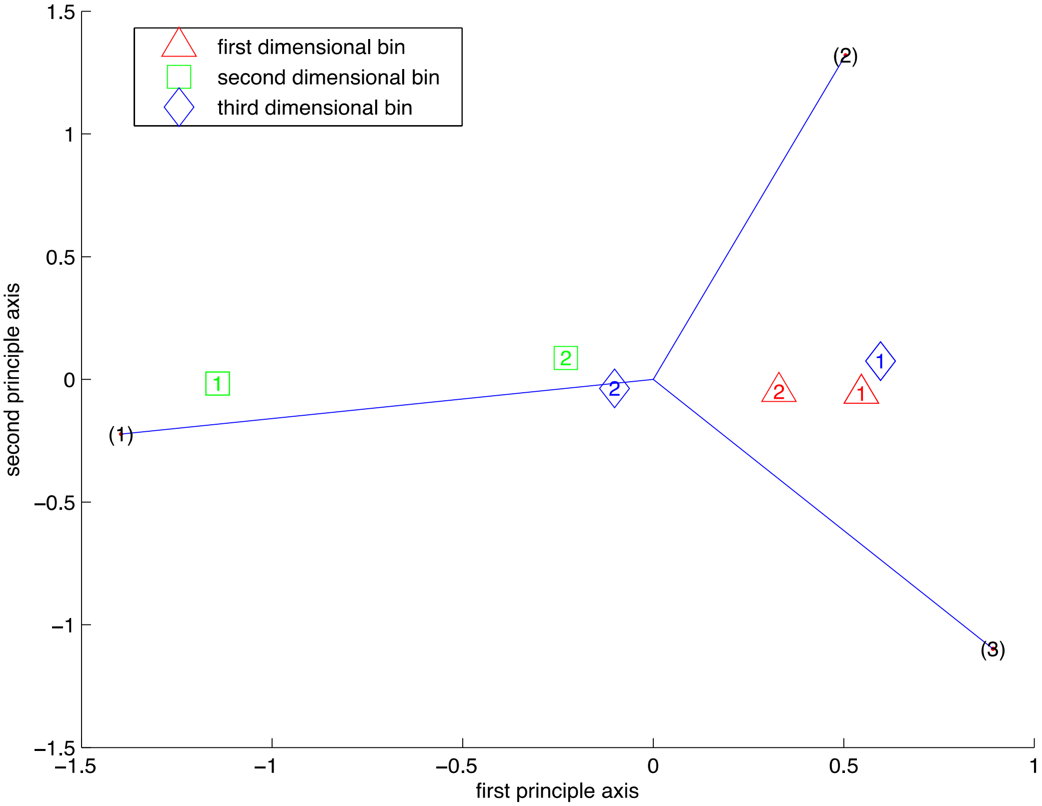


Figure 3.13 2D Rosenbrock function

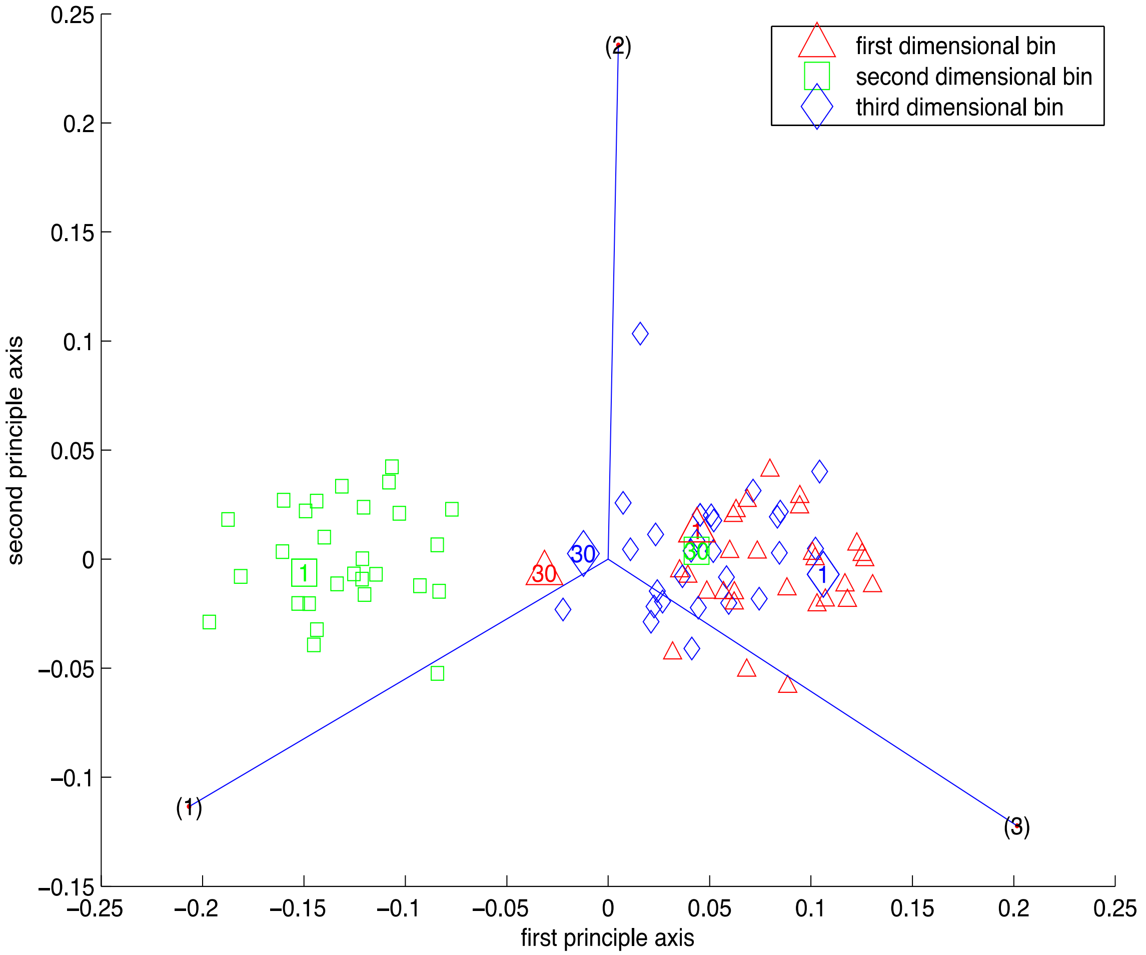


Figure 3.14 30D Rosenbrock function

- Griewank function

The Griewank function has many widespread local minima, which are regularly distributed. The complexity is shown in the zoomed-in plots. Regardless of the different shapes of Griewank function and Ackley function, the properties are similar. According to the formula, the variables are pseudo-symmetric variables in the range of [-600, 600], which also means the variables have similar influence on the function value. The second dimensional bins, which are the center of the domain, have the optimal as Figure 3.15 shows. The plots of the 2D Griewank function (Figure 3.16) and 30D Griewank function (Figure 3.17) are similar to the Ackley function and have the same interpretation as we discuss before. The total inertial for each variable is also alike as the third column of Table 3.10 shows.

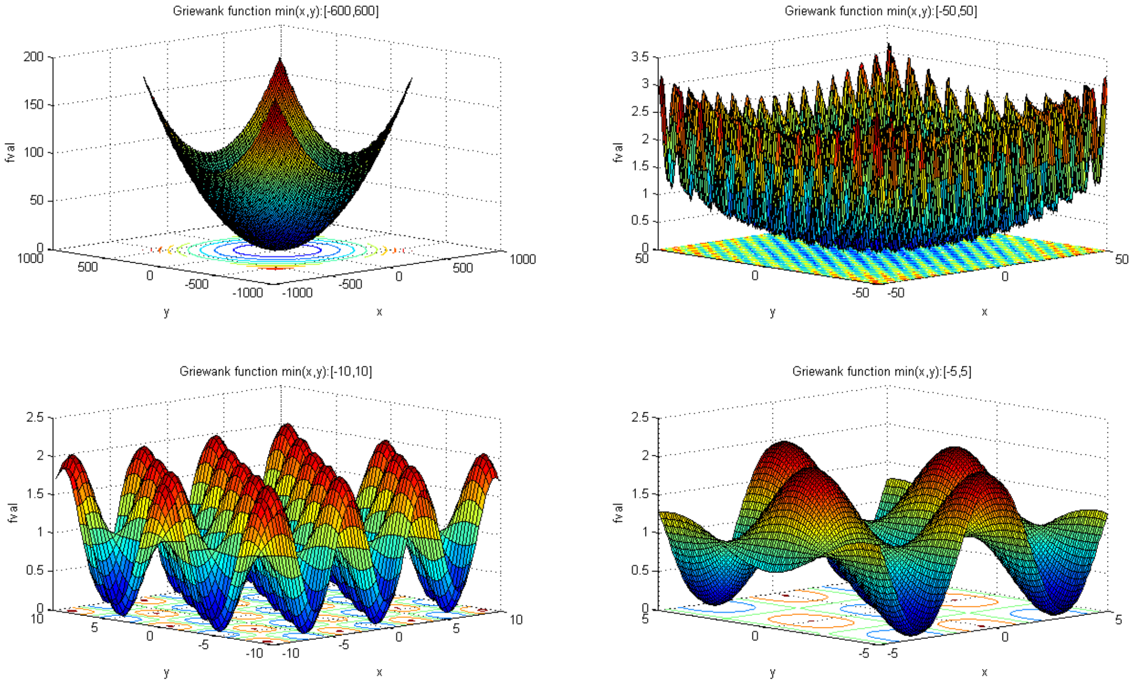


Figure 3.15 Griewank function with different variable ranges

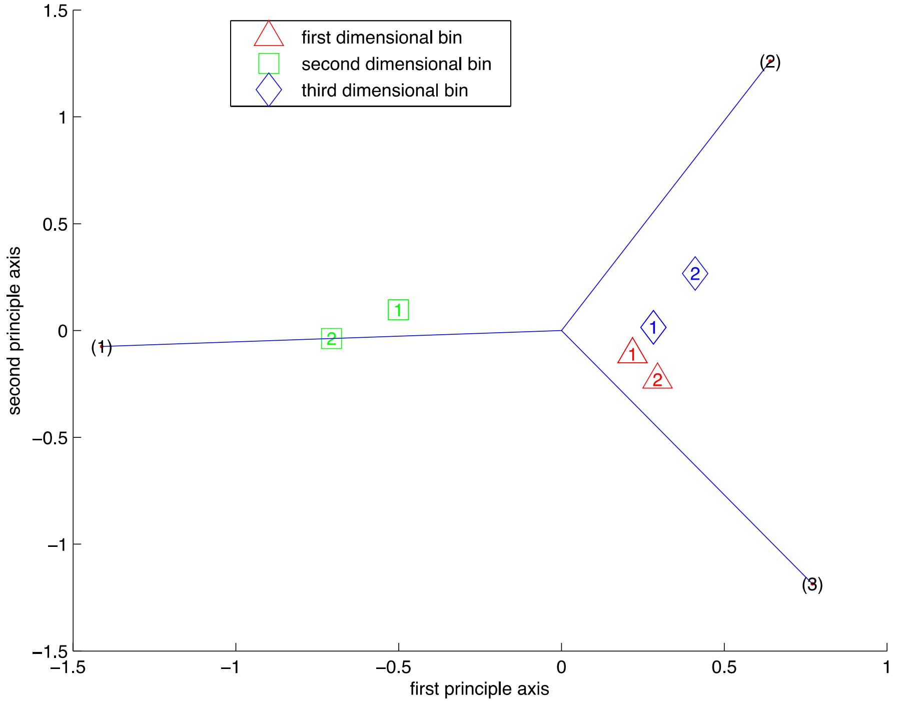


Figure 3.16 2D Griewank function

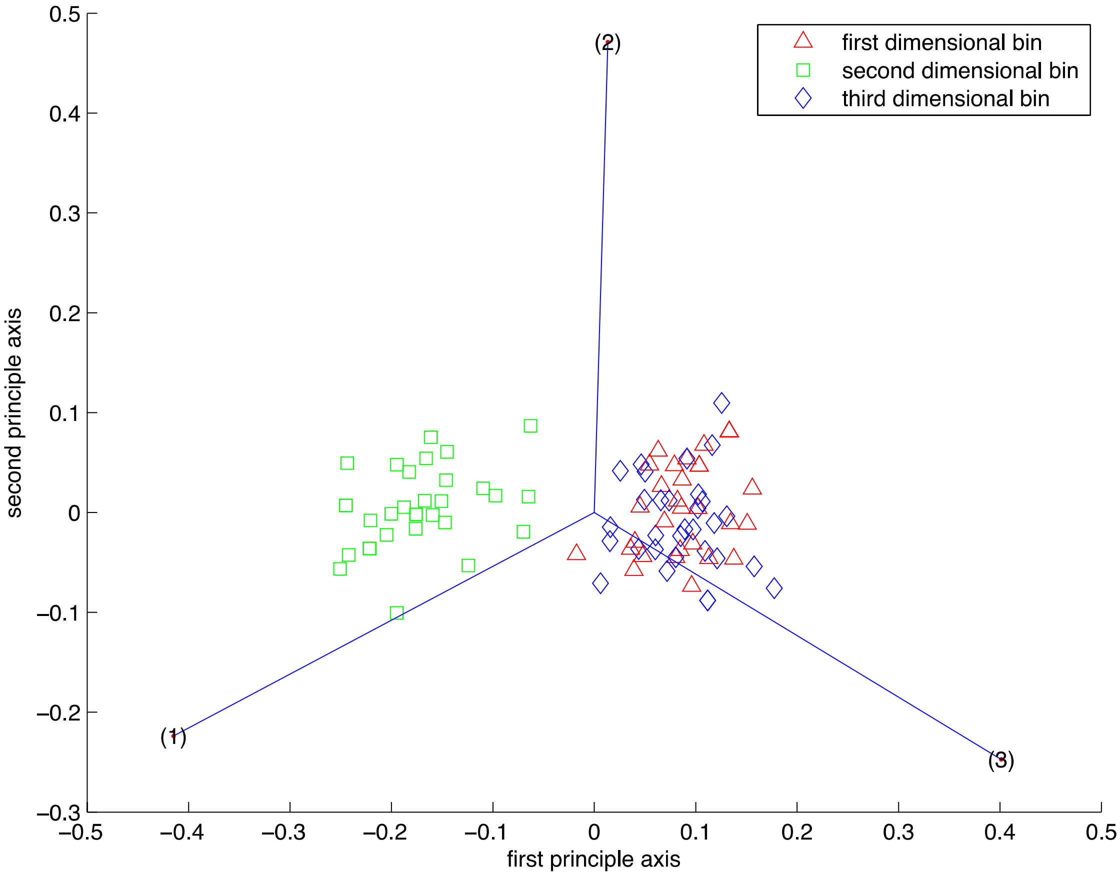


Figure 3.17 30D Griewank function

Overall, from the accuracy and efficiency perspectives, the proposed visualization method demonstrates satisfactory performance. The total time needed by the visualization is represented by the number of total function evaluations, which does not increase exponentially with the dimension.

The following part shows the convergence test for the proposed method. As we increase the number of sample points, the CA plot will remain similar, which is considered the convergence state. We desire to find the minimal sample points to reach the convergence state. The total inertial of the matrix is used to measure the relative distance of the dimensional bin points, which to a certain degree shows the similarity of the figures. When the total inertia numbers in two consecutive iterations are close to each other, it means that the distribution of the points in those two corresponding graphs is also similar. Therefore, we use the total inertia to show the convergence of the graph for the proposed method.

In the following figure, the *x*-axis represents the number of sample points and the *y*-axis shows the total inertia for the matrix. Six different problems are tested and different line colors represent different problems as the legend indicates.

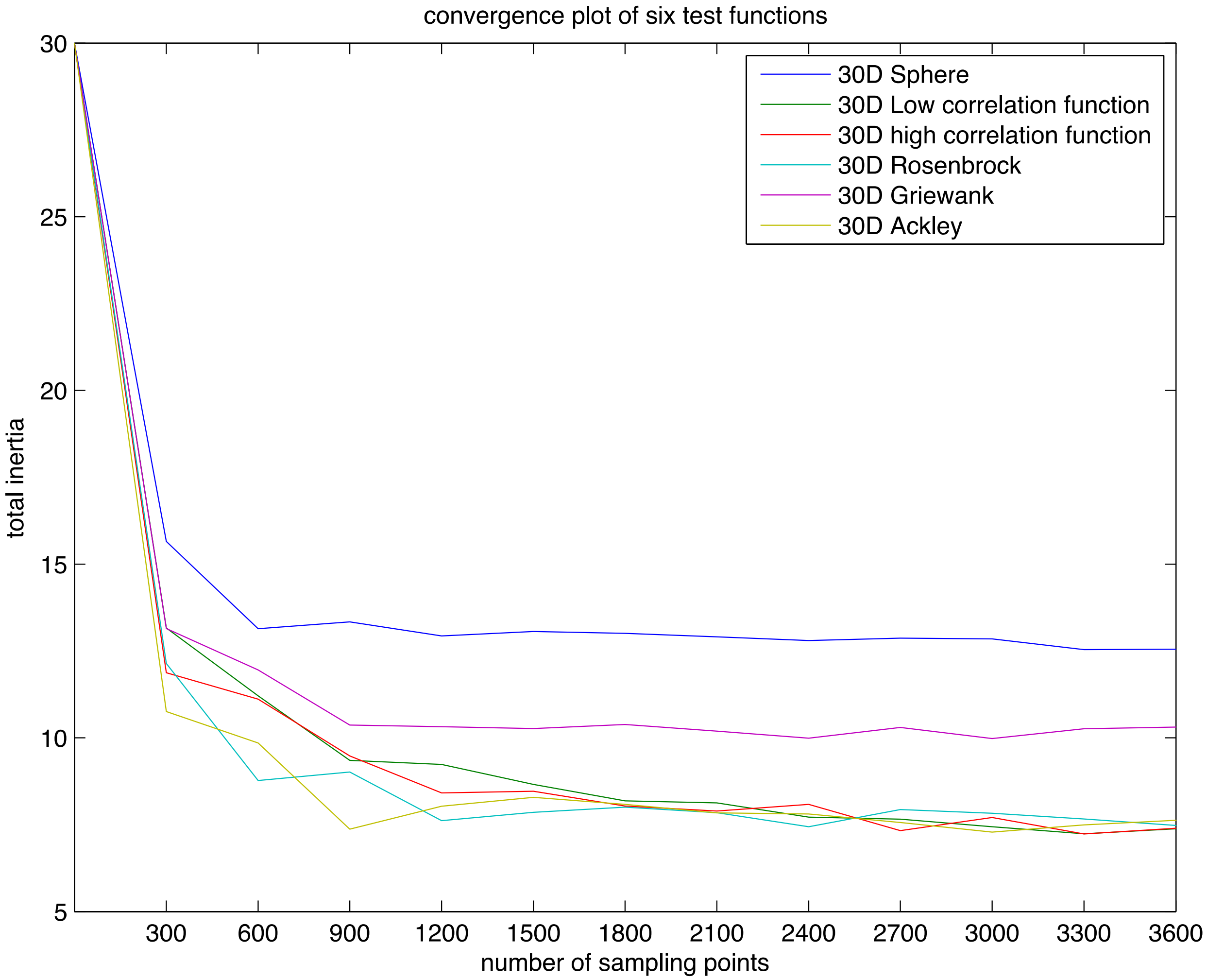


Figure 3.18 Convergence plot for six functions

As we can see from Figure 3.18, the proposed method starts to converge when 900 points are sampled for 30-variables problems. The suggested number of sample points is thus 30 times the number of variables for the problems. However, due to the different complexity of each problem, the “sufficient” number of sample points may also change. Under sampling may result in incorrect interpretation of the black box.

3.9. Summary

The process of applying CA to analyze HEB problem, the guideline of choosing parameter for the method, and the experiment results are described in this chapter. The results show that the proposed method can discover the data's properties with a limited number of samples. However, discovering the symmetric variables need users to identify the clusters. It requires human effort and sometimes inevitably causes errors. The next chapter will use hierarchical clustering method to find the clusters and discover the symmetric variables automatically.

Chapter 4. Computer assisted knowledge discovering

Information discovering is usually interactive and requires human to identify patterns or relations in a data set. This chapter aims to use the hierarchical clustering method to automatically classify similar data and find the distinct groups. A brief introduction and background of hierarchical clustering is provided and the average-linkage clustering is used for experiments in this chapter.

4.1. Hierarchical Clustering

The hierarchical clustering [76] is an agglomerative algorithm [77], which is called hierarchical because it builds the clusters from the bottom-up hierarchical approach. The clusters start from single point clusters and continuously combine the nearest clusters to form bigger clusters until the distances between the clusters are all larger than the predefined cut-off distance (ϵ).

4.1.1. Basic Terminology

Given a set of *n* points x_1, x_2,..., x_n, the terminology for the hierarchical clustering are defined as followed:

Dissimilarity (d_{ij}): The term dissimilarity [78] is often used to measure the difference between two points and usually represented by the distance between the points. The distance between a pair of point *i* and point *j* is denoted as d_{ij}. In addition, the distance is not necessarily Euclidean distance. It can be Manhattan distance [79], cosine distance [80] or Pearson's distance [81], etc.

Cluster (G): A group of points (**P**) with similar properties is considered as one cluster. A cluster G with n_G points can be expressed by a set $G = \{P_{id_1}, P_{id_2}, \dots, P_{id_{n_G}},\}$ where ***id*** is the index of point.

Linkage: The dissimilarity score between two clusters G_1 and G_2 is called linkage and denoted as $d(G_1,G_2)$. There are three kinds of linkage, single linkage (i.e., nearest-neighbour linkage) [82], complete linkage (i.e., furthest-neighbour linkage) [83], and average linkage [84]. Figure 4.1 illustrates the different ways of measuring the dissimilarity score between clusters.

Cut-off Distance: a parameter provided by the user to indicate how large the distance between groups is needed for them to be considered truly distinct.

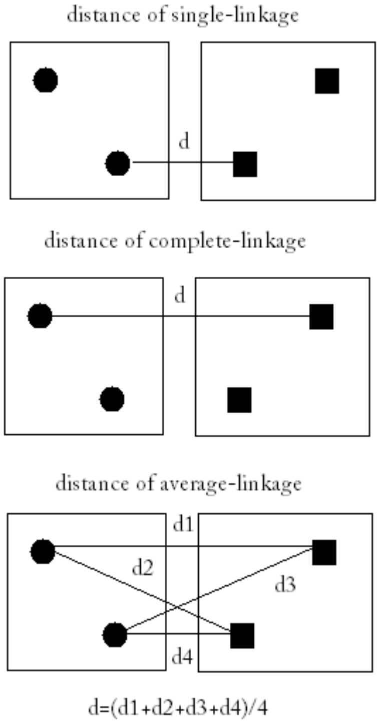

distance of single-linkage
d
distance of complete-linkage
d
distance of average-linkage
d1
d2
d3
d4
d=(d1+d2+d3+d4)/4

Figure 4.1 Illustration of three kinds of linkages and distance measures

Single linkage: the dissimilarity score is the smallest dissimilarity between pair of points with one point from cluster G_1 and one point from the other cluster G_2.

$$d_{single}(G_1, G_2) = \min d_{ij} \ (i \in G_1, j \in G_2) \tag{30}$$

Complete linkage: the definition is similar to the single linkage except that the complete linkage uses the largest dissimilarity of two points in opposite clusters.

$$d_{complete}(G_1, G_2) = \max d_{ij} \ (i \in G_1, j \in G_2) \tag{31}$$

Average linkage: the dissimilarity of two clusters is equal to the average distance from any member of one cluster to any member of the other cluster. In the formula, $n_{G_1}$and $n_{G_2}$are the number of points in G_1 and G_2 respectively.

$$d_{average}(G_1, G_2) = \frac{1}{n_{G_1} + n_{G_2}} \sum_{i \in G_1, j \in G_2} d_{ij} \tag{32}$$

4.1.2. Basic Step of Hierarchical Clustering

For example, there are N points to be clustered, the basic steps of hierarchical algorithm are shown as following:

1. At the first step, each cluster only contains one point (a.k.a. each point forms its own cluster). So if there are N points, N clusters will be formed at the first stage.

2. Calculate the dissimilarities between every two clusters.

3. Select a pair of clusters that have the smallest dissimilarity, and combine those two clusters into one bigger cluster.

4. Recalculate the dissimilarity between the new cluster and all the old clusters.

5. Continue perform steps 2 to 4 until all the points are in a single cluster or the distances between clusters are larger than the cut off distance.

The simple example is shown in Figure 4.2 to illustrate the steps of the hierarchical clustering (use single-linkage).

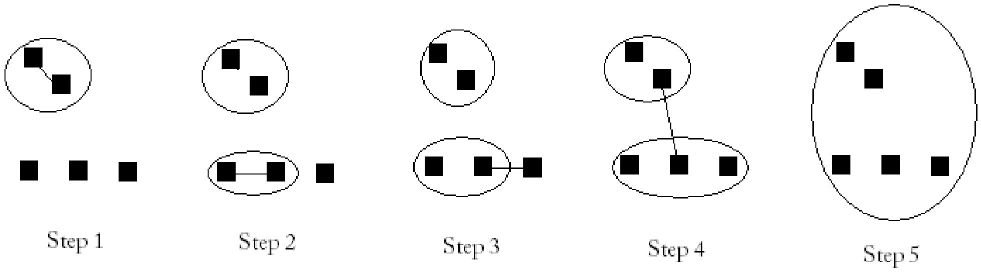


Figure 4.2 Example of single-linkage clustering

4.2. Apply Average-linkage Clustering to CA Plot

The average-linkage clustering method is applied to each of the dimensional bins. It is a compromise between the complete-linkage clustering, which is sensitivity to outliers, and the single-link clustering, which has tendency of forming bigger clusters. Therefore, the average-linkage clustering method is chosen because it is less sensitive and more robust than the other two methods. Without the clustering method, users have to discover the different groups of variables in order to identify the importance of the variables. Now with the assistance of average-linkage clustering, the group is automatically formed and can save a lot of human effort. However, the number of clusters still needs to be determined by users. Therefore, the user can look at the CA plot first and estimate the number of clusters, then input the number of clusters for average-linkage clustering to group the points. The hierarchical clustering first forms a dendrogram as it is shown in Figure 4.3. Then we can determine the cut-off distance according to the predefined number of clusters. After that, the clusters are automatically formed according to the cut-off distance. For example, in Figure 4.3, if we need to cluster the points into two groups, then the cut-off distance can be set to between 0.8 and 1.2 as the red line shows. Then the points in different sub-tree are automatically classified into separated clusters. Here, points 6, 8, 1 and 5 are in one cluster and points 9, 2, 10, 7, 3 and 4 are in the other.

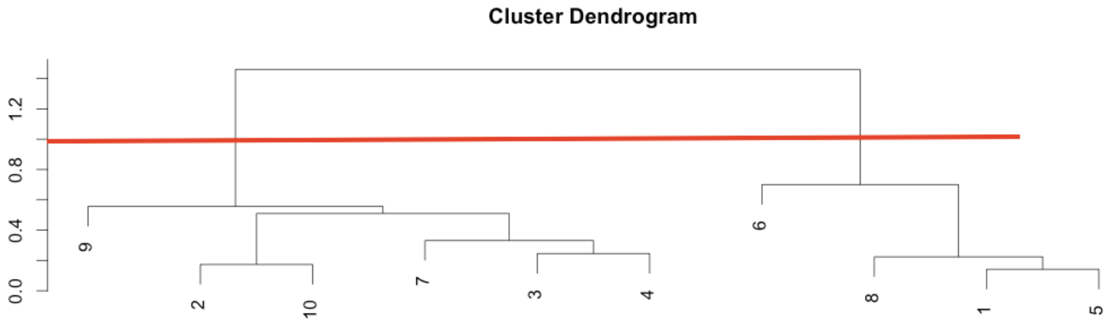


Figure 4.3 Dendrogram of hierarchical clustering

4.3. Experiment and result

For the design functions in Table 3.6, the low correlation function and the high correlation function are tested to justify the correctness of the clustering algorithm. The results are shown in Figure 4.4 to Figure 4.7.

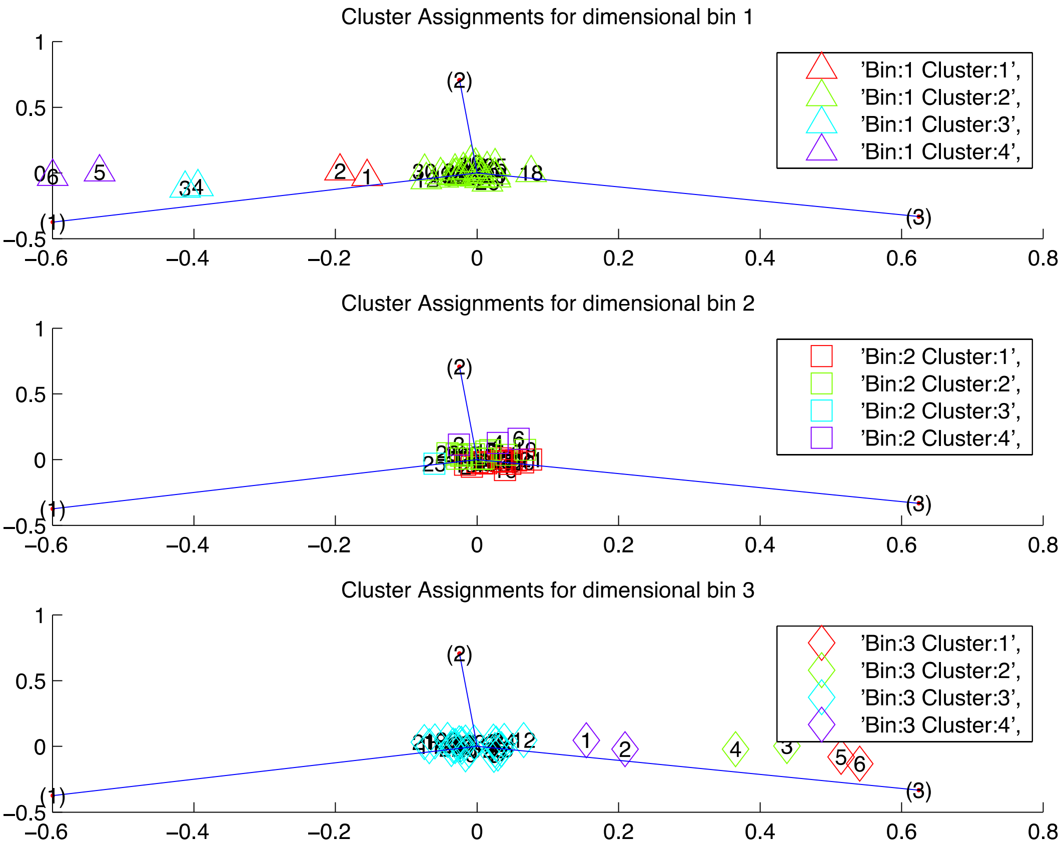


Figure 4.4 Clusters of each dimensional bin for low correlation function

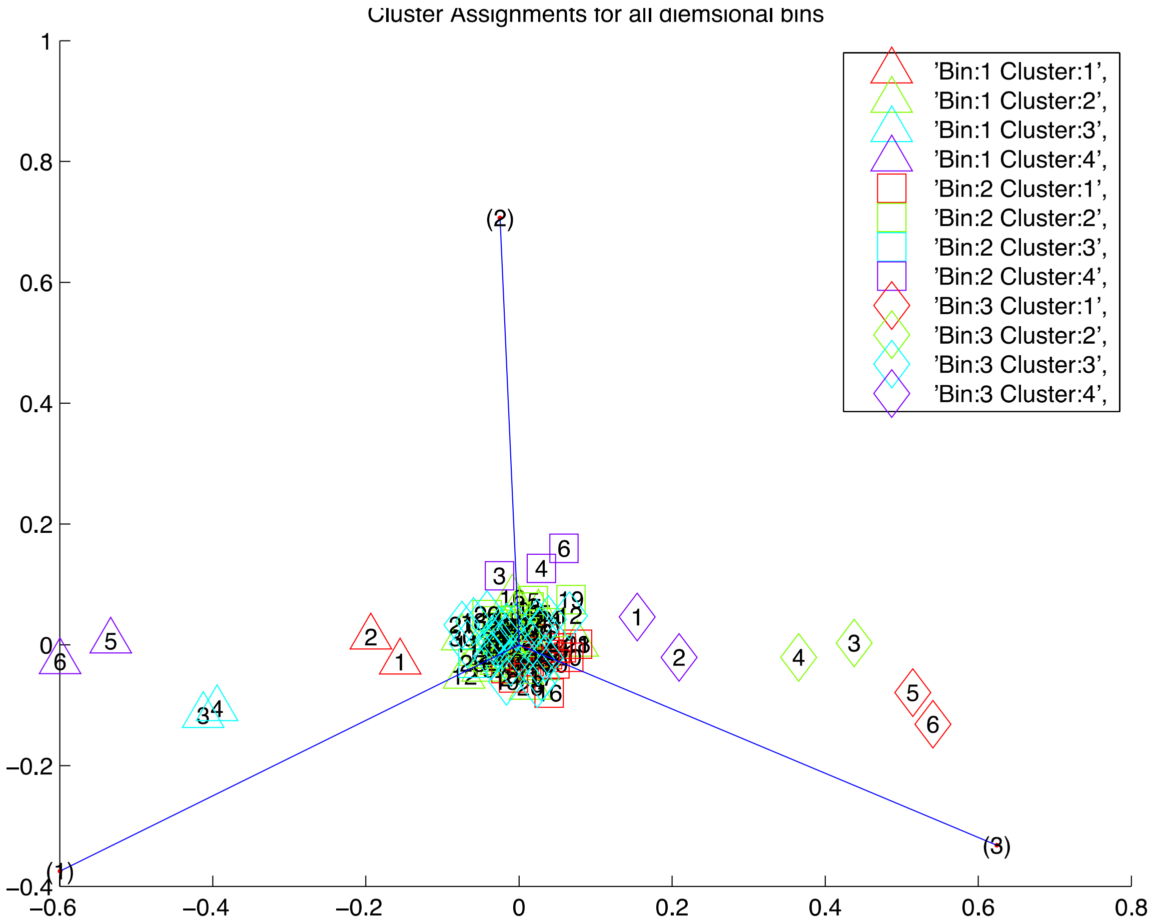


Figure 4.5 Clustering for low correlation function

For the low correlation function, we represent the clusters for each of the dimensional bin in different plots in Figure 4.4, while the combination of the clustering for all dimension bins are shown in Figure 4.5. The first, second and third dimensional bins are represented by triangle, square, and diamond symbols respectively as indicated in the legend. The number inside the symbol represents the i^{th} variable. Different clusters are separated by different colors. Inertia of three dimensional bins for a certain variable can represent the importance of that variable. As the inertia of second dimensional bins is small and similar to each other, we will focus on the first and third dimensional bins. Therefore, the clusters for the second dimensional bin (plot in the middle of Figure 4.4) can be ignored. We only need to focus on clusters in the top and the bottom plot of Figure 4.4. For example, the plot on the top of Figure 4.4 shows that the 6th and 5th variables with purple are in one cluster and the 4th and 3rd variables with blue are in other cluster. So in the plot, 1st and 2nd variables, 3rd and 4th variables, 5th and 6th variables, and 7th to 30th variables are in separated clusters. The 5th and 6th variables are both the most important variables because their dimensional bins are in the same group and are further apart from the origin than the other variables. Similarly, the 3rd and the 4th variables are the second important variables and the 1st and 2nd variables are the third important variables. The coefficients for the 5th and 6th variable, 3rd and 4th variables, and 1st and 2nd variables are three, two and one respectively. Therefore, the 5th and 6th variables, which have the largest coefficient is the most important variables, and then the 3rd and 4th variables. The information discovered by the clustering method is correct according to the formula.

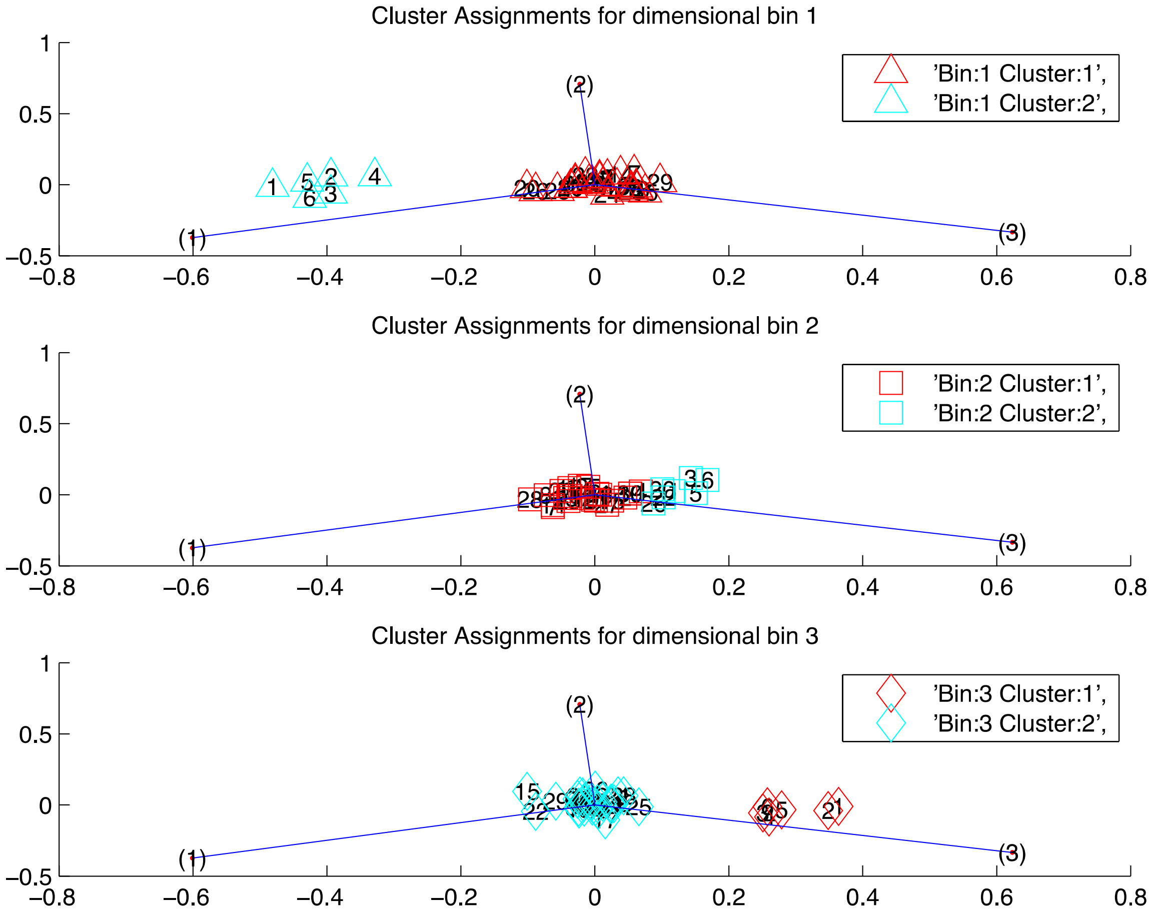


Figure 4.6 Clusters for each dimensional bin for high correlation function

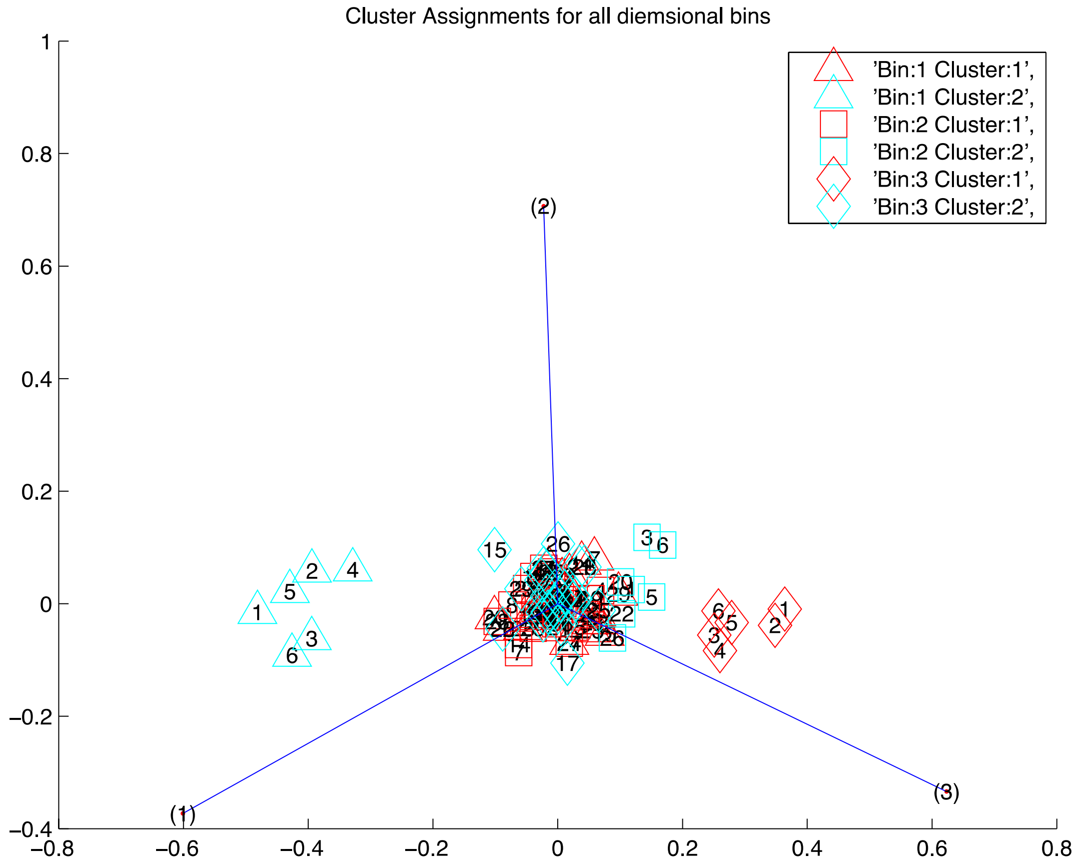


Figure 4.7 Clustering for high correlation function

Similarly, in Figure 4.6 and Figure 4.7, the 1st to 6th variables are in one cluster and all the other variables are in the other cluster. The dimensional bins of 1st to 6th variables are further apart from the origin. Therefore, the 1st to 6th variables are the most important variables and all the other variables are less important. As the high correlation function indicates, the 1st to 6th variables have influence on the function value while the 7th to 30th variables are redundant. Thus, the discovered information indicating that the 1st to 6th variables are the most important variables is consisted with the formula.

4.4. Summary

The hierarchical clustering method, which is used to discover the symmetric variables automatically, is introduced in this chapter. The average linkage clustering is applied to find the different clusters for each dimensional bin. Based on the clusters, symmetric variables can be easily discovered if they are in the same cluster among all the dimensional bins. In next chapter, CA is applied to the current optimization method to help the optimization method to find a near optimal point with fewer samples.

Chapter 5. Applying CA to optimization

This chapter will apply the discovered information as described in the previous chapter to assist the optimization process. TRMPS2 will be used as an example to show how CA can be applied to assist the optimization process. We will first give a brief introduction to TRMPS2 and describe the improved algorithm (also called CA_TRMPS in this thesis) in detail. For further details on TRMPS2, please refer to [85] and [86].

5.1. TRMPS2 Algorithm

TRMPS2 is developed based on MPS [87], which relies primarily on the initial stochastic sampling [88] and meta-modeling (a.k.a. surrogate modeling) to perform optimization. After integrating the concept of trust regions [89]-[91] into the MPS (TRMPS), TRMPS2 is more efficiency for high dimensional problems than the previous MPS algorithm. The major steps of TRMPS2 are similar to MPS and will be explained in detail in the following section.

5.1.1. Definition of Trust Region:

In TRMPS2, Dual Hyper Cubes are defined and denoted as hyper cubes S and hyper cubes B, where S is initially within B. The trust region S (TR_S) is enclosed in hyper cube S while the trust region B (TR_B) occupies the area of B minus the area of S. The trust region of 2D is shown as an example in Figure 5.1. The area of shaded region is TR_S and the white area is TR_B. The points fall into the trust region S and trust region B are denoted as $\{x_S\}$ and $\{x_B\}$ respectively. The corresponding function value set will then be $\{f_{xS}\}$ and $\{f_{xB}\}$.

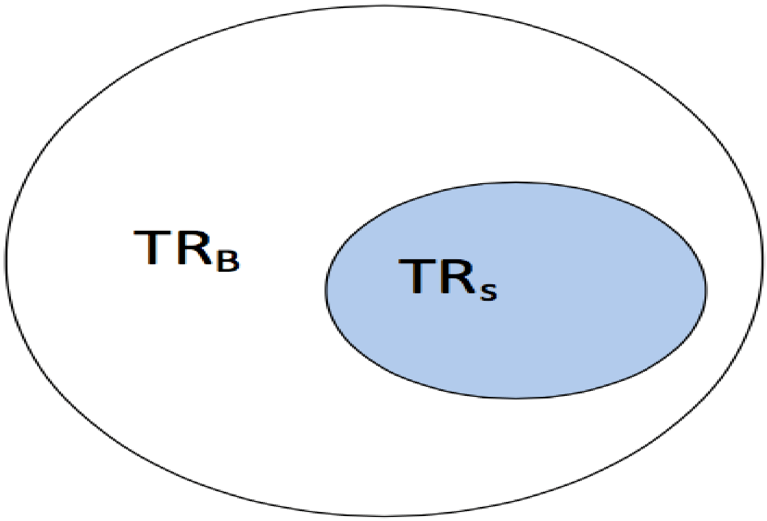


Figure 5.1 2D trust region example

5.1.2. Basic steps of TRMPS2

The steps of the algorithm are as followed:

1. Initial sampling

Uniformly sample n_0 random points in the design space to generate a set of points {x} and then evaluate the function values {f_x} with the black box function. Define the initial trust region TR_S and TR_B according to the initial space determined by the user.

2. Partition to different sets according to the region

Two sets of points {x_S} and {x_B} are then generated. {x_S} represents the points that fall into TR_S and {x_B} represents the points within TR_B. The corresponding function values for {x_S} and {x_B} are denoted as {f_{xS}} and {f_{xB}}, respectively.

3. Construct Metamodel

The expensive points {x_S} in TR_S are used to construct the metamodel by fitting {f_{xS}} onto an RBF. The RBF is shown in Eq.(33):

$$f(x) = \sum_{i=1}^{m} \alpha_i \|x - x_{si}\| \tag{33}$$

where m is the number of points in $\{x_S\}$; a_i is the weight that point x_{si} contributes to the objective function value; and x_{si} is an expensive point in $\{x_S\}$ which is used to build the model.

4. Select contour according to the metamodel point

Uniformly sample n_1 random points $\{x_{S1}\}$ in TR_S and evaluate their cheap function values $\{f_{x_{S1}}\}$ with the RBF model. As the RBF model is used to evaluate the function values of $\{x_{S1}\}$, the cost of the computation and time is not expensive anymore. Therefore, a larger number of samples can be generated in this step.

The discriminative sampling procedure as described in [92] is then used to select a subset of points $\{x_{S2}\}$ within $\{x_{S1}\}$. The point in $\{x_{S1}\}$ with smaller function value is intended to have more chances to be selected according to the mechanism of discriminative sampling procedure. Then the expensive function (black box) is used to calculate the function values $\{f_{x_{S2}}\}$ for $\{x_{S2}\}$. Place all the points of $\{x_{S2}\}$ in $\{x_S\}$.

5. Construct Quadratic model

When the number of points in $\{x_S\}$ is large enough to build the quadratic model [93], Eq.(35) is used to build the model. The minimal number of required points is n_{kk}, which can be calculated from Eq.(34). In Eq. (34), n_v is the number of variables, and n_p is set as equal to n_v.

$$n_{kk} = \frac{(n_v + 1)(n_v + 2)}{2} + 1 - n_p \tag{34}$$

If the cardinality of $\{x_S\}$ (the number of members of $\{x_S\}$) $|\{x_S\}|$ is less than n_{kk}, skip the rest of the process and go to Step 8.

Otherwise, n_{kk} points that are closest to the current optimal (denoted as $\{x_{Skk}\}$) are used to build the quadratic model q in Eq.(35), where $\boldsymbol{\beta}$ are model coefficients.

$$q = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \beta_{ii} x_i^2 + \sum_{i<j} \sum_{j=1}^{n} \beta_{ij} x_i x_j \qquad (35)$$

6. Evaluate the model

R-square [94] which can be calculated using (36) is employed in this step to test the model accuracy.

$$R^2 = 1 - \frac{\sum_{i=1}^{m} (f(x_{si}) - \widehat{f(x_{si})})^2}{\sum_{i=1}^{m} (f(x_{si}) - \overline{f(x_{si})})^2} \qquad (36)$$

where m is the number of samples. $f(x_{si})$ and $\widehat{f(x_{si})}$ are the black-box function value and approximate model function value for x_{si} respectively. The mean value of the black box function over the m samples is denoted as $\overline{f(x_{si})}$. The approximation model is more accurate when the value of R-square is closer to one. R-square value is used to calculate the overall accuracy of the model while the max difference (*Diff*), which defines the maximum relative difference between the actual and approximated function values, are used to judge the quality of the local approximation. *Diff* is calculated as:

$$Diff = \max\{|f(x_{si}) - \widehat{f(x_{si})}|\}\, i \in [1, m] \qquad (37)$$

7. Local optimization:

In this step, we first find the region *R'* defined by $\{x_{Skk}\}$, where the lower bound and upper bound for the i^{th} dimension is determined by the minimal and maximal value in the i^{th} dimension among all points.

If $R^2 \approx 1$, sample $n_v/2$ expensive points in *R'* and add to $\{x_{Skk}\}$. Construct a quadratic model using the updated $\{x_{Skk}\}$ and then recalculate R^2.

Calculate *Diff* using Eq.(37) and compare the value with a pre-determined threshold e_d. If *Diff* < e_d, use the trust region based local optimization described in [19] and find the local optimal point x_L. Then calculate the real function value of x_L using the black box. If x_L is within the region R', then terminate. Otherwise, add x_L to $\{x_S\}$.

8. Perform steps 3 to 7 for $\{x_B\}$ in TR_B.
9. Update region

If a better optimum point is discovered, then TR_S expands to avoid being trapped in the local optimal. TR_B, in contrary, contracts to exploit a smaller but more promising region. Update two regions according to Eq.(38) and Eq.(39):

$$TR_s = TR_s/k_{reduction} \tag{38}$$

$$TR_B = TR_B.k_{reduction} \tag{39}$$

Otherwise, TR_S contracts to search in a promising region and TR_B expands to find the optimal in the region further away from the current optimal. This means that when the search result is not improved, we will explore other new region or focus more on the promising region. The updated equations will become:

$$TR_B = TR_B/k_{reduction} \tag{40}$$

$$TR_S = TR_S.k_{reduction} \tag{41}$$

The last step balances the exploitation and exploration properties of the algorithm to achieve much more intelligent searching.

Go back to Step 2.

The general process of TRMPS2 is shown in Figure 5.2:

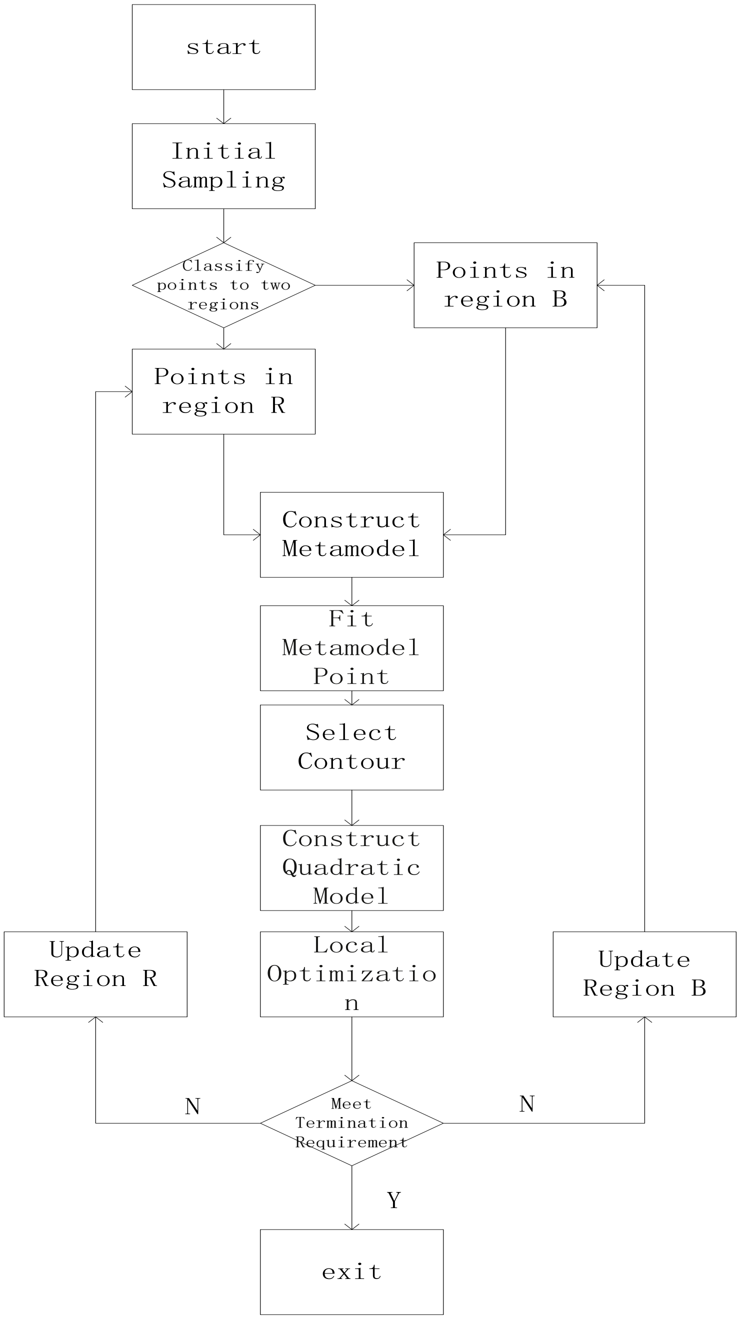


Figure 5.2 TRMPS2 Procedure

5.2. Applied CA to TRMPS2

The proposed method employs CA at the first stage (initial sampling stage) of the algorithm. As CA requires certain amount of points (usually 15*the number of variables) to show the statistic properties of the data set, the initial samples should be larger than TRMPS2. When the number of points reach 30*the number of variables, the Figure 3.18 shows that the inertia start to converge, but the user can start to recognize the same information in the figure when the parameter is larger than 10, therefore, here we chose a number 15 which is slightly larger than 10. As the total number required for finding the optimal for low dimension is small, the initial points take a large proportion of the total samples, which leaves fewer points for exploring the space intelligently. Therefore, the proposed method is more suitable for the high dimensional problem. For example, TRMPS2 only needs around 50 points to find the optimal for 2D Rosenbrock problem, then 2*15=30 initial points will cost too much. However, more than 5000 points are required for the 10D Rosenbrock problem, then 10*15=150 initial points will become reasonable. Especially when the dimensions go up to more than 50, the initial points only take a small portion of the total samples, and the information discovered in the early stage can be greatly helpful to guide the resampling. Therefore, in this thesis we focus on applying CA to the optimization algorithm to deal with high dimensional problems.

To integrate CA with the resampling process of TRMPS2, a.k.a. the fourth step of the algorithm described in the last chapter, the information, which is found by CA in the initial stage, is then used to guide the resampling by altering the points in $\{x_{S2}\}$. The resulting points form the set $\{x_{S3}\}$ and in the following step $\{x_{S2}\}$ will be replaced by $\{x_{S3}\}$. $\{x_{S3}\}$, which is generated from $\{x_{S2}\}$, has the tendency of containing the points with smaller approximate function values, therefore, $\{x_{S3}\}$ inherited the information of meta-model from $\{x_{S2}\}$. Also as $\{x_{S3}\}$ combines the information from CA and the information from meta-model, the resampling process becomes more intelligent. The proposed strategy can find a better result with the same number of expensive function evaluations.

The basic steps are similar to TRMPS2, and here, we only highlight the differences. The highlighted procedures of the proposed method are as follows:

1. Initial sampling

Use Latin Hypercube Sampling to sample n_1 points within the design space. The generated set of points is place in $\{x\}$. The initial sampling is different from TRMPS2, which uses uniform sampling, and the number of initial samples in TRMPS2 is much smaller than n_1. Besides those, others are the same in this step.

2. Apply CA to discover information from the black box

The Correspondence Analysis for the optimization method discussed in the previous chapter is used in this step. The CA plot is shown and the input dialog is appeared. It will wait for the user to input the discovered information. We will use a simple example to illustrate the format of input information.

For a ten variables mathematical function (Eq.(42)) with the redundant variables x_5-x_{10}, the CA plot for this problem is shown in Figure 5.3. The most important variables are x_3 and x_4 according to Figure 5.3, because the symbols with number '3' and '4' in them, which represents the third and the fourth variables, are far away from the origin. Similarly, the first and second variables are the second important variables. The rest of the variables are the least influential to the objective function. The promising regions for x_1-x_4 are the first interval of their design space. It is obvious that the triangles that represent the first dimensional bins have large projection onto the first dimensional bin vector, which represents the small function value.

Therefore, in summary, the discovered information will be:

(1) First dimensional bins of x_1-x_4 are promising subdomains

(2) The most significant variables are x_3 and x_4, then x_1 and x_2, all the rest variables are less important for this problem.

(3) x_3 and x_4 are symmetric variables with each other. So are x_1 and x_2.

$$f(x) = x_1x_2 + 2x_3x_4 \tag{42}$$

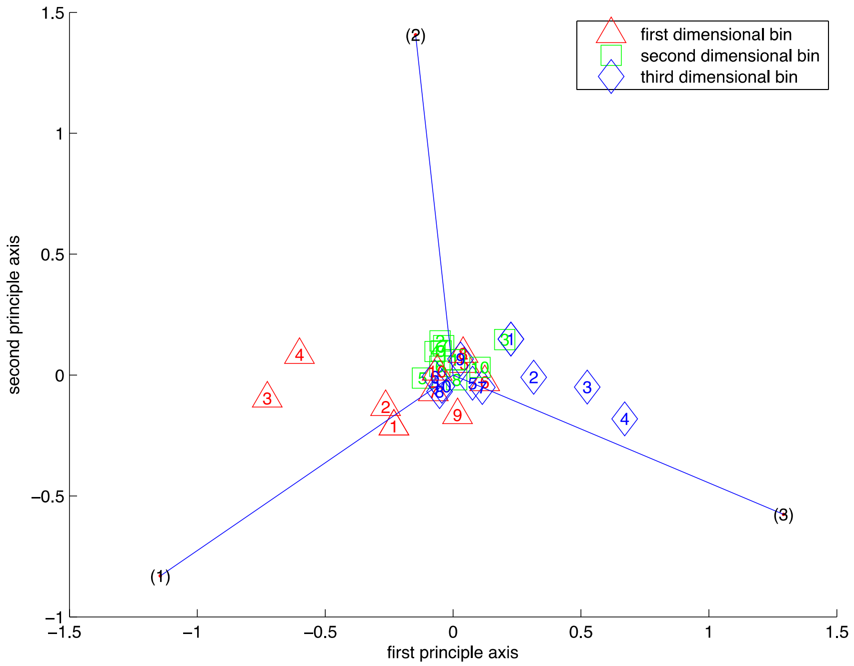
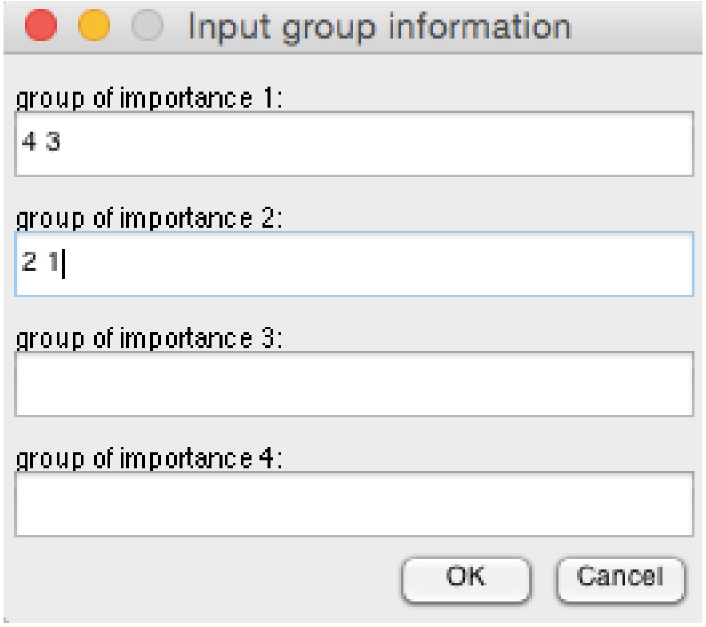


Figure 5.3 CA plot of 10D example function

The information discovered from the CA plot will be fed back to the optimization system in order to assist the resampling. The input should be according to the following format:

(1) The importance of the variables:

As shown in Figure 5.4, variables of similar importance will be put into one group; the most important variables will be put into the first line and the second most important variables will be put into the second line and so on. We don't need to input all variables, if we consider some variables are not important, we won't need to input the indices of those variables in the window. For example, the previous plot indicates the third and the fourth variables are the most important, so we input '4' and '3' in the first line. The number should be separated by one space. Because the variables in the same group have the same importance, the order of input does not matter. Therefore, input '3 4' is the same as input '4 3'. The less important variables as CA plot indicates are x_1 and x_2, thus, we put them in the second line, indicating they are less important than x_3 and x_4.



group of importance 1:
4 3

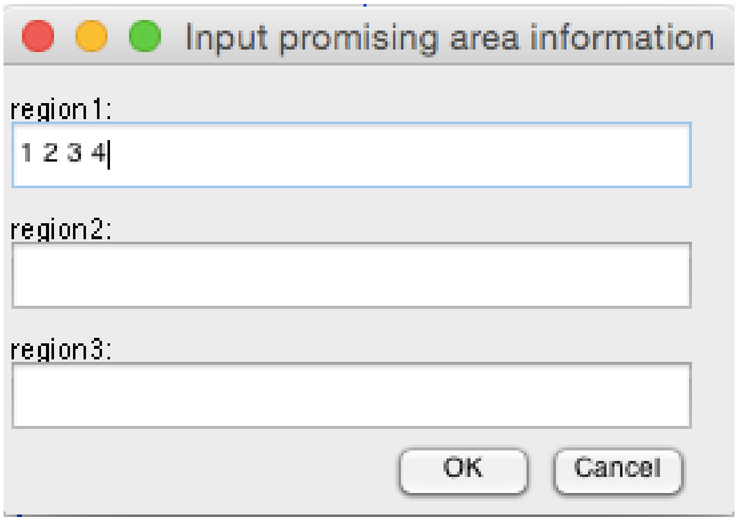
group of importance 2:
2 1

group of importance 3:

group of importance 4:

Figure 5.4 Input window of the important variables

(2) The promising subdomain:



region1:
1 2 3 4

region2:

region3:

Figure 5.5 Input window of the promising subdomain of the variables

As shown in Figure 5.5, if the first region of the i^{th} variable is the promising subdomain, then number '*i*' should be put in the first line. Similarly, variables with the second (third) interval as the promising subdomain will be put into the second (third) line. The number should be separated by a space. For the example problem, the first dimensional bins of x_1-x_4 are the promising subdomains, '1 2 3 4' will be put into the first line.

3. Altering the points in $\{x_{s2}\}$ according to the discovered knowledge

For all the points in $\{x_{s2}\}$, we alter them according to the following strategy:

(1) Important variables:

The probability of changing the value of the i^{th} variable is denoted as p_i. The base probability of changing the variables is p_0 and it becomes larger when the variable is more important. There will be different levels of importance and the total number of different levels of important variables is denoted as n_L. If the i^{th} variable is in the j^{th} level of importance, p_i can be calculated according to the Eq.(43):

$$p_i = \frac{(1-p_0)}{(2^{n_L}-1)} \times 2^{n_L-j} + p_0 \tag{43}$$

The probability for the i^{th} variable is the base probability plus the increased step. This strategy ensures that the increased step for the j^{th} level of importance is half of the $(j\text{-}1)^{th}$ level of importance. Therefore, the points in first level of the importance will have the largest increased step in probability. For example, there are three levels and the base probability is 0.3. Then the based increase step is $\frac{(1-0.3)}{(2^3-1)} = 0.1$. Therefore, the increase steps for the first to third level are $0.1 \times 2^{3-1} = 0.4$, $0.1 \times 2^{3-2} = 0.2$ and $0.1 \times 2^{3-3} = 0.1$, respectively. In this way the increased step for the j^{th} level of importance is half of the $(j\text{-}1)^{th}$ level of importance. The probabilities for the first to third level of importance are 0.7, 0.5 and 0.4 respectively.

(2) Promising subdomain

The promising subdomains can be discovered in the plot and are indicated by the user in the input window. After we obtain the information, the points in $\{x_{s2}\}$ are altered to be more likely in the promising subdomains according to the certain probability defined in Eq.(43). For the i^{th} variable, we first verify whether the existing value is within the promising subdomain. If the value is not in the promising subdomain, we randomly generate a new value within the promising subdomain. However, the new value is accepted according to certain probability p_i we calculated in the previous step. If the new value is not accepted, then we won't change the value for that dimension.

Take the previous 10D function (Eq.(42)) as an example, the promising subdomain for the problem is M_{sub} where $M_{sub} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 4 & 4 & 4 & 10 & 10 & 10 & 10 & 10 & 10 \end{bmatrix}$. The i^{th} column represents the i^{th} variable with the first row indicating the lower bound and the second row indicating the upper bound. The original boundaries for all the variables are [1,10] and the promising subdomains for x_1 - x_4 are indicated as the first dimensional bins, therefore, the regions will be [1,4] for x_1 - x_4 and unchanged ([1,10]) for other variables. In this example, the base probability p_0 is set to 0.4. As there are two groups of important variables (n_L=2), the increased step for the second important group (j=2) will be (1-0.4)/3=0.2 and the increase step for the first group of important variables will be (1-0.4)/3*2=0.4. The probability vector for all the variables will be [0.6, 0.6, 0.8, 0.8, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4], where p_i implies the probability of changing the value for the i^{th} variable. If a point P_A is [-10, -10, -10, -10, -10, -10, -10, -10, -10, -10] then the first four variables are not within the promising subdomain. Therefore, the first two variables have 0.6 chances to be replaced by a random generated number from [1,4] while the third and the fourth variables have 0.8 chance of being replaced.

5.3. Experiments

Two sets of test functions are used to test the performance of TRMPS2 and CA_TRMPS. The first set of functions are designed functions with several correlated variables and different coefficients. The functions have fifty variables and some of the variables are redundant. The other set of functions are popular benchmark functions that are used by many researchers. The functions are scalable and we choose three versions of different dimensions (30D, 50D and 100D) of those functions. All of the functions are single objective and unconstrained mathematical functions. We benchmark

CA_TRMPS against the original TRMPS2 with a limit of 5000 function evaluations for each run and 10 runs of each function to reduce random variations.

The designed functions are shown in Table 3.6. In this section, we include a highly correlated function, which is very difficult to be solved by the existing optimization algorithms and we want to show the great improvement of the algorithm after integrating with CA.

Table 5.1 Design Function for Proposed Method

Function name	Dimension (Number of inputs)	Formula	Search domain	Minimum
Square function	50	$min\, f(x) = (\sum_{i=1}^{5} x_i)^2 + 2 \times (\sum_{i=11}^{15} x_i)^2$	$0 \le x_i \le 10$	$x^* = (0, \dots, 0)$ $f(x^*) = 0$
Low correlation function	50	$\min f(x) = 2\prod_{i=1}^{4} x_i + \prod_{i=5}^{8} x_i$ $x_9\ to\ x_{50}\ are\ redundant\ variables$	$1 \le x_i \le 10$	$x^* = (1, \dots, 1)$ $f(x^*) = 4$
High correlation function	50	$min\, f(x) = \prod_{i=1}^{50} x_i$	$1 \le x_i \le 10$	$x^* = (1, \dots, 1)$ $f(x^*) = 1$

The CA plot of the design function is shown from Figure 5.6 to Figure 5.10. For the figure without clustering, the number '*i*' within the symbol indicates the *i*th variable and different symbols represent different dimensional bins as the legend indicates. The clustering algorithm is then applied to each dimensional bin separately to discover importance of the variables. For the figure with clustering, the numbers and the shape of the symbol still hold the same meaning. However, the color shows different clusters found by the clustering method.

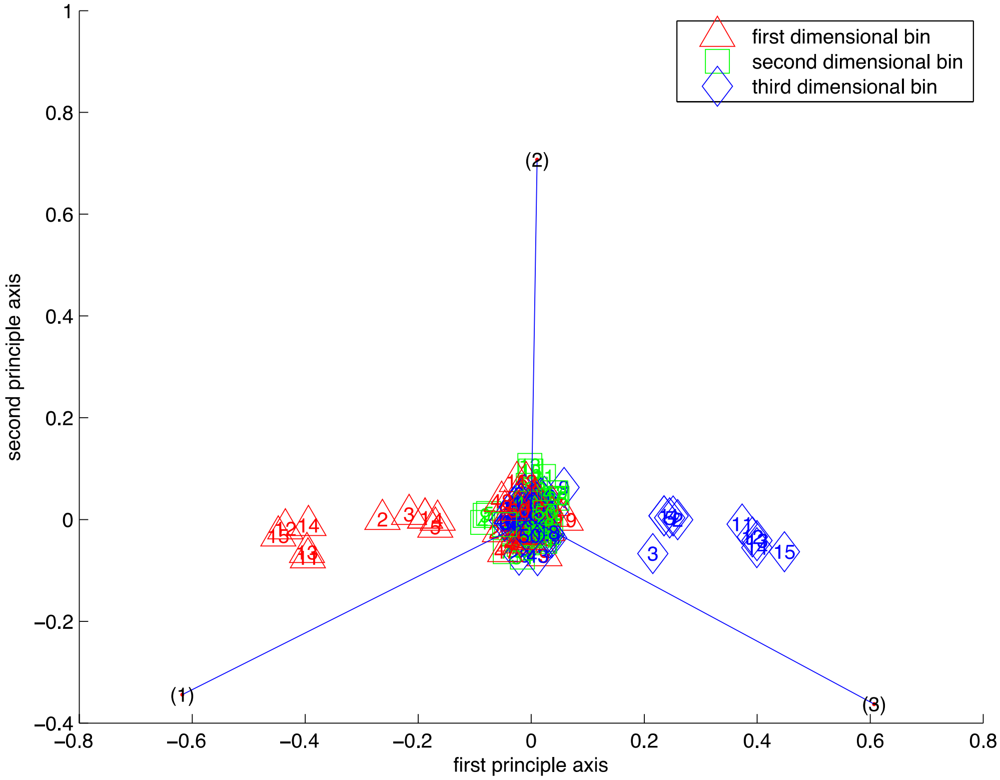


Figure 5.6 CA plot of square function without clustering

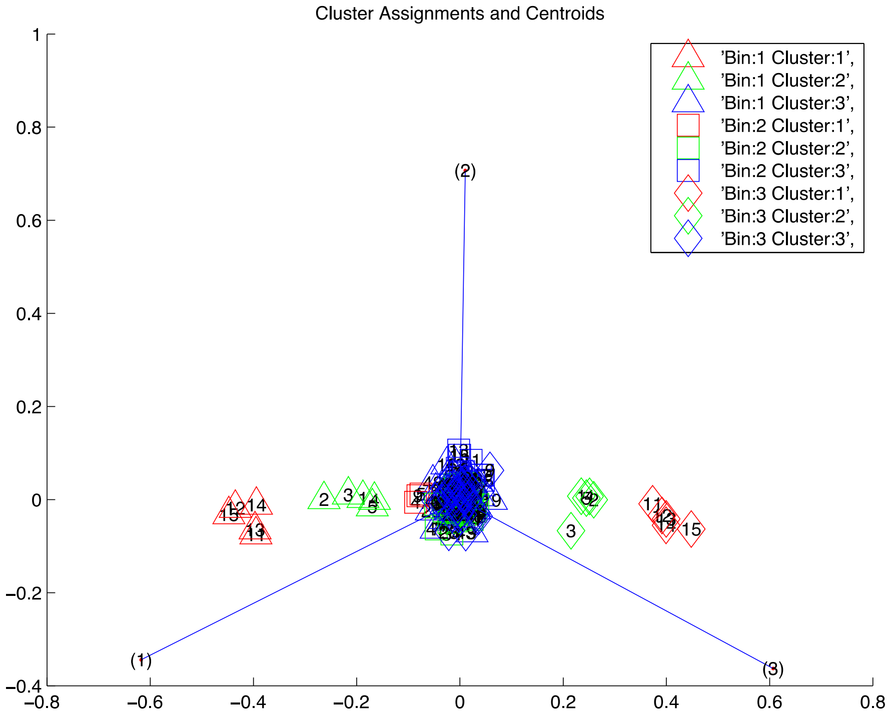


Figure 5.7 CA plot of square function with clustering

The Figure 5.6 and Figure 5.7 reveal the properties of the square function. As Figure 5.7 shows, the 11th -15th variables are in one group, which is the furthest from the origin representing the most important group of variables. The 1st to 5th variables are in the second important group. All the rest variables form anther one group, which are the least important group. Also in Figure 5.6, the first dimensional bins for the 1st to 5th variables and the 11th to 15th variables are more promising as they have the larger projection onto the first function value bin. Similarly, for the low correlated function as shown in Figure 5.8 and Figure 5.9, the first dimensional bins are the promising subdomain for the variables and the 1st to 4th variables are the most important variables and the 5th to 8th variables are less important.

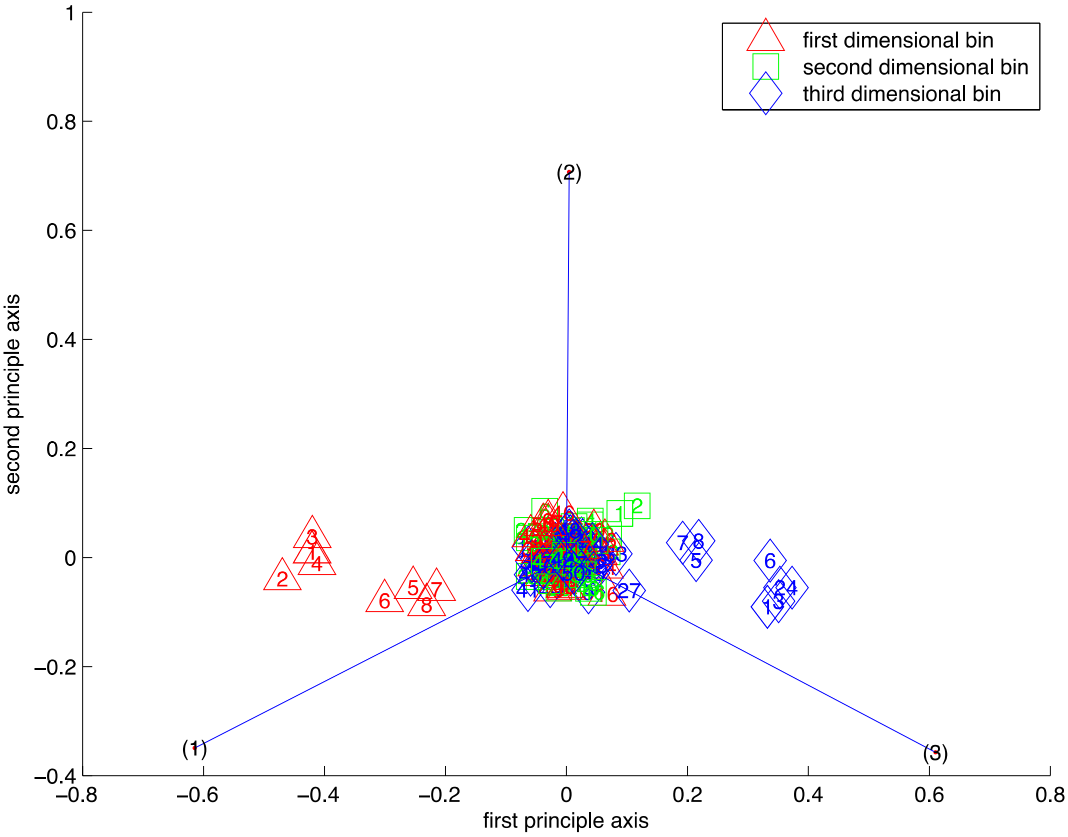


Figure 5.8 CA plot of low correlation function without clustering

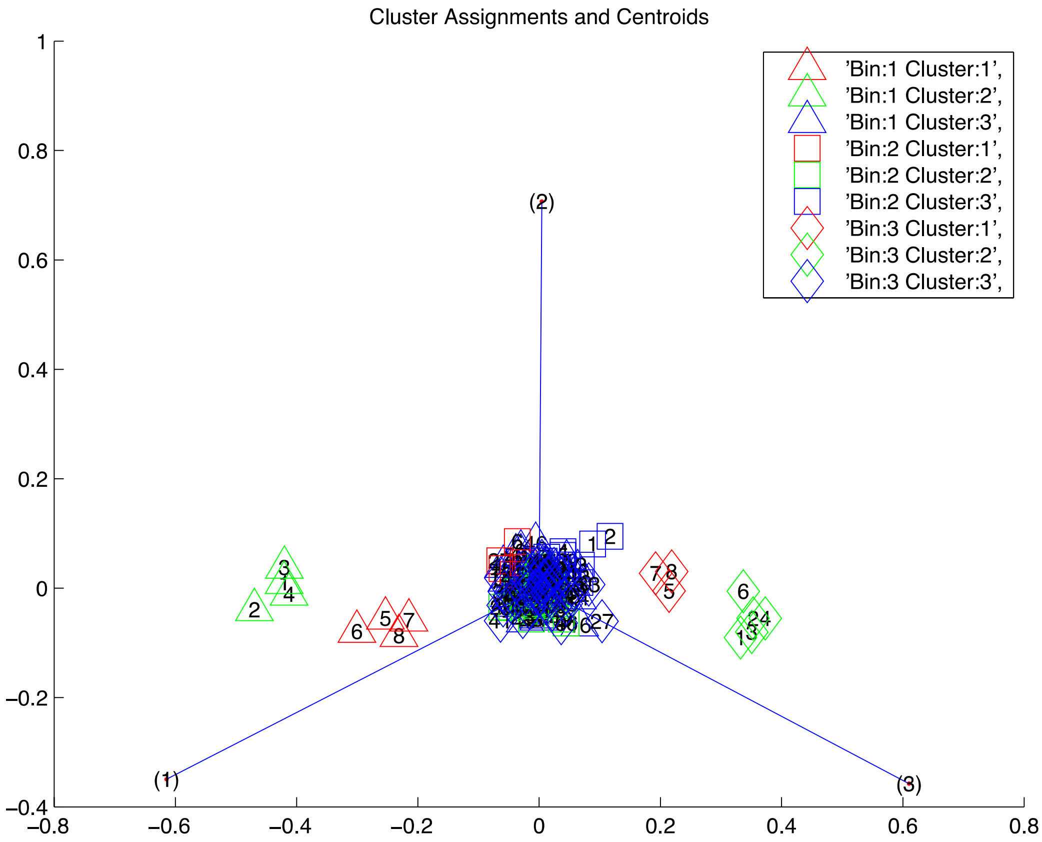


Figure 5.9 CA plot of low correlation function with clustering

For the high correlation function, it is obvious that all the variables are in the same group, so we don't need to apply the clustering algorithm on it. The result in Figure 5.10 indicates that for all the variables, the first dimensional bins comprise the promising subdomain.

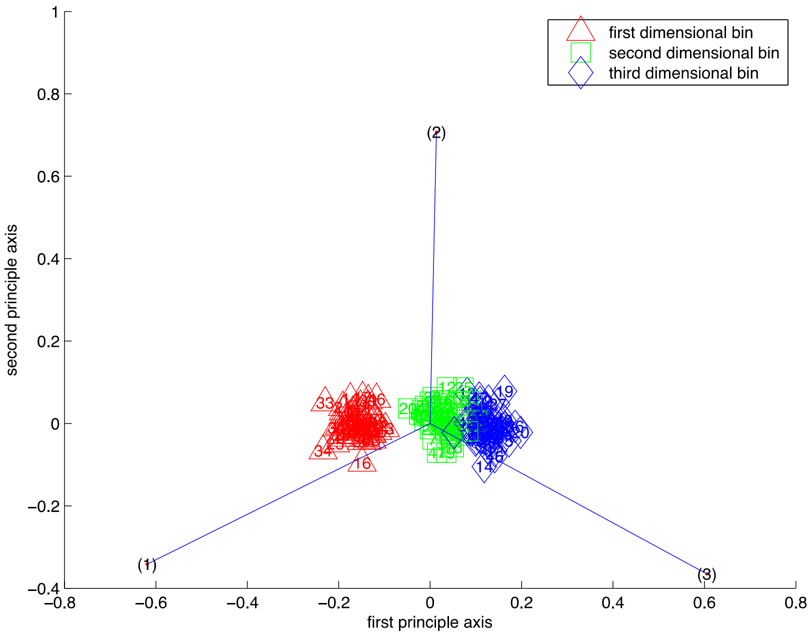


Figure 5.10 CA plot of high correlation function without clustering

All the information discovered above are integrated with TRMPS2 to develop a new optimization algorithm CA_TRMPS. The comparison of two algorithms is shown in Table 5.2. In Table 5.2, *fval* is the abbreviation of optimal function value found by the algorithm. The mean *fval* represents the average of the optimal function values for 10 runs and STD of *fval* shows the standard deviation of the optimal function values. The actual optimal function value of the function is also provided to compare how close the optimal value found by the algorithm to the analytical one.

Table 5.2 Result for the Design Functions

		CA_TRMPS		TRMPS2	
Problem	**Optimal fval**	**Mean fval**	**STD of fval**	**Mean fval**	**STD of fval**
Square function	0	88.7774	8.3077	102.6829	18.0386
Low correlation function	4	4.8456	1.2267	36.3991	39.9867
High correlation function	1	5.9157e+6	1.6452e+7	2.2047e+26	6.9638e+26

From Table 5.2, we can easily obverse that CA_TRMPS can find a better optimal point within 5000 function evaluations for all the problems. For the low correlation function, we can see that the result of CA_TRMPS is quite close to the analytical optimal, while the optimal found by TRMPS2 is eight times larger than it. The CA_TRMPS can achieve a much better result when the problem is highly correlated. The order of magnitude for the result of CA_TRMPS is six while the order of the magnitude for the results of TRMPS2 is 26, which means the CA_TRMPS can find a much better result for the high correlated problems. The standard deviation of *fval* also shows that the CA_TRMPS can yield the similar result for each run. It means that performance of the proposed method is stable and will find the similar result, however, the qualities of the results found by TRMPS2 is not that predicable because it sometimes finds a good result and sometimes not.

Table 5.3 Benchmark Test Functions for Proposed Methods

Function name	Number of dimension	Formula	Search domain	Minimum
Ackley (Many local minimal)	30, 50, 100	$f(x) = -ae^{-b\sqrt{\frac{1}{d}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^{n}\cos(cx_i)} + a + e$ a=20, b=0.2 and c=2π	$-32.768 \le x_j \le 32.768,$ $i = 1,2,\dots,n.$	$x^* = (0,\dots,0); f(x^*) = 0$
Rosenbrock function (Valley-shaped)	30, 50, 100	$f(x) = \sum_{i=1}^{n-1} \left[100\left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2\right]$	$-10 \le x_j \le 10,$ $i = 1,2,\dots,n.$	$x^* = (1,\dots,1);\ f(x^*) = 0$

According to Figure 3.11 and Figure 3.14, the second dimensional bins are the promising subdomain for all the variables in both Rosenbrock function and Ackley function. However only one group of important variables is shown in the figure, therefore, we consider all the variables are of the same importance. The information is then applied to the TRMPS2 by using the proposed strategy. The comparison of CA_TRMPS and TRMPS2 results are shown in Table 5.4.

Table 5.4 Result for the Benchmark Test Functions

		CA_TRMPS		TRMPS2	
Problem	Optimal fval	Mean fval	STD of fval	Mean fval	STD of fval
30D_Ackley	0	8.4165	1.9025	18.1402	1.0384
50D_Ackley	0	10.4630	0.9158	18.9940	0.3880
100D_Ackley	0	11.4707	0.5476	18.9557	0.1947
30D_Rosenbrock	0	27.6370	1.5146	74.8736	57.7884
50D_Rosenbrock	0	59.9041	24.5319	129.9210	136.6731
100D_Rosenbrock	0	306.7377	88.0860	500.0284	189.1272

The CA_TRMPS outran TRMPS2 for all the problems. For Rosenbrock function, CA_TRMPS finds a much better result as the dimension increases, and exhibits less variation. For the 30D Rosenbrock function, CA_TRMPS is only 37.9972 less than TRMPS2, but when it comes to 100D, it is 129.8592 less than TRMPS2. Although the results for Rosenbrock problem are getting worse for both algorithms as the dimension increases (the results of CA_TRMPS and TRMPS2 are both further from the optimal value zero), CA_TRMPS can find a relatively better result within a given number of function evaluations. The CA_TRMPS results of Ackley function are smaller than TRMPS2, but the standard deviation of CA_TRMPS is slightly worse than TRMPS2. But the difference between the standard deviation values is less than 1, so we can say the variation of the performances is quite similar.

5.4. Summary

In this chapter, CA_TRMPS is introduced to use the information discovered by CA in the early stage to guide the resampling process. The results of adaptive CA_TRMPS method show that the proposed method can find better results with a certain number of samples. In next chapter, CA is applied to real engineering problem to show that the proposed method only requires small amount of samples and can be practical for solving real problems.

Chapter 6. Case Study

In this chapter, Correspondence Analysis is applied to the phosphate production Problem. First, the background knowledge of phosphate processing is provided to give a general review of the problem. Second, how to formulate the problem is explained in detail by describing the specific optimization goal and the inputs in standard form. The results from a specific application are shown to illustrate that the Correspondence Analysis can help to discover a lot of useful information of the black box. Finally, the CA_TRMPS is used to solve this optimization problem. It is shown that after integrating the discovered knowledge, CA_TRMPS can achieve more promising results than original TRMPS2.

6.1. Background

6.1.1. Phosphate Process

The procedure of converting phosphorous compounds into different levels of phosphate, also known as the phosphate process, is widely used in industry, especially for chemical fertilizer production. The phosphate process is usually scheduled in a certain way to meet the demand, since there might be delays during the transition of raw materials between the different sections in the process. The quality of phosphate production is mainly determined by how the phosphate process is scheduled. A better scheduling for the phosphate process can make the production much more efficient.

We aim at solving the general process-scheduling problem without knowing any previous knowledge of the simulation. Once the demand (amount of output) is given, the method can automatically find the schedule of the input to satisfy the demands. This approach does not need human intervention and can work with a wide scope of optimization problems including scheduling, planning, and process design. The challenge, however, lies with the need for a high number of simulations for large-scale problems. The strategy is to randomly pick some schedules, and then use an optimization method to iteratively search for a better schedule by running the actual simulations. This method therefore needs to try a large number of schedules before it finds the near optimal one. As each simulation takes a long time to run for the given schedule, the whole optimization process would be lengthy. Especially when the dimensionality for the model of scheduling problem is high, the normal optimization methods usually require running a large number of simulations and the total time would make the approach unpractical. Therefore, more information, which can be automatically discovered during sampling step, is needed to assist the resampling, or optimization, process. The strategy needs to provide a unique way and guideline to find the good quality solutions in a limited number of evaluations, or simulations.

6.1.2. Simulation software

CADSIM Plus [94] is a chemical engineering software tool developed by Aurel System Company. It allows the user to draw a process flow sheet while creating a process simulation model. CADSIM Plus uses a complete CAD drawing interface, on which the user can create a simulation model. In other words, it helps the user to build a computer prototype of the actual chemical engineering process, instead of actually building the process physically. When the process flow sheet is completed, a specific input can be fed into the model for software to run the simulation process in an idealized condition determined by the flow sheet. Results then will be given after the simulation as the output of the simulation model.

There are two major advantages of the software. First, the design errors can be discovered earlier by running the simulation before actually building the process with real material. The cost of changing the error in the software is much less than on the actual chemical process. Second, as the simulation is running much faster than the real process, it can be used to find a better schedule for the process. The schedule can then be used as an input in the real process to obtain a better result.

6.2. Problem description

6.2.1. OCP (Phosphorus concentration) model

CADSIM Plus is used to build the model as it is shown in Figure 6.1. This model is used to produce four different grades of phosphates, which are classified by their ingredients. Here, four different grades of phosphates are K09, K10, K12 and K20.The raw materials enter the system from the inputs, be mixed and fertilized during the procedure to become intermediate material and finally turn into the final products, which are stored in the separated storage tanks (outputs).

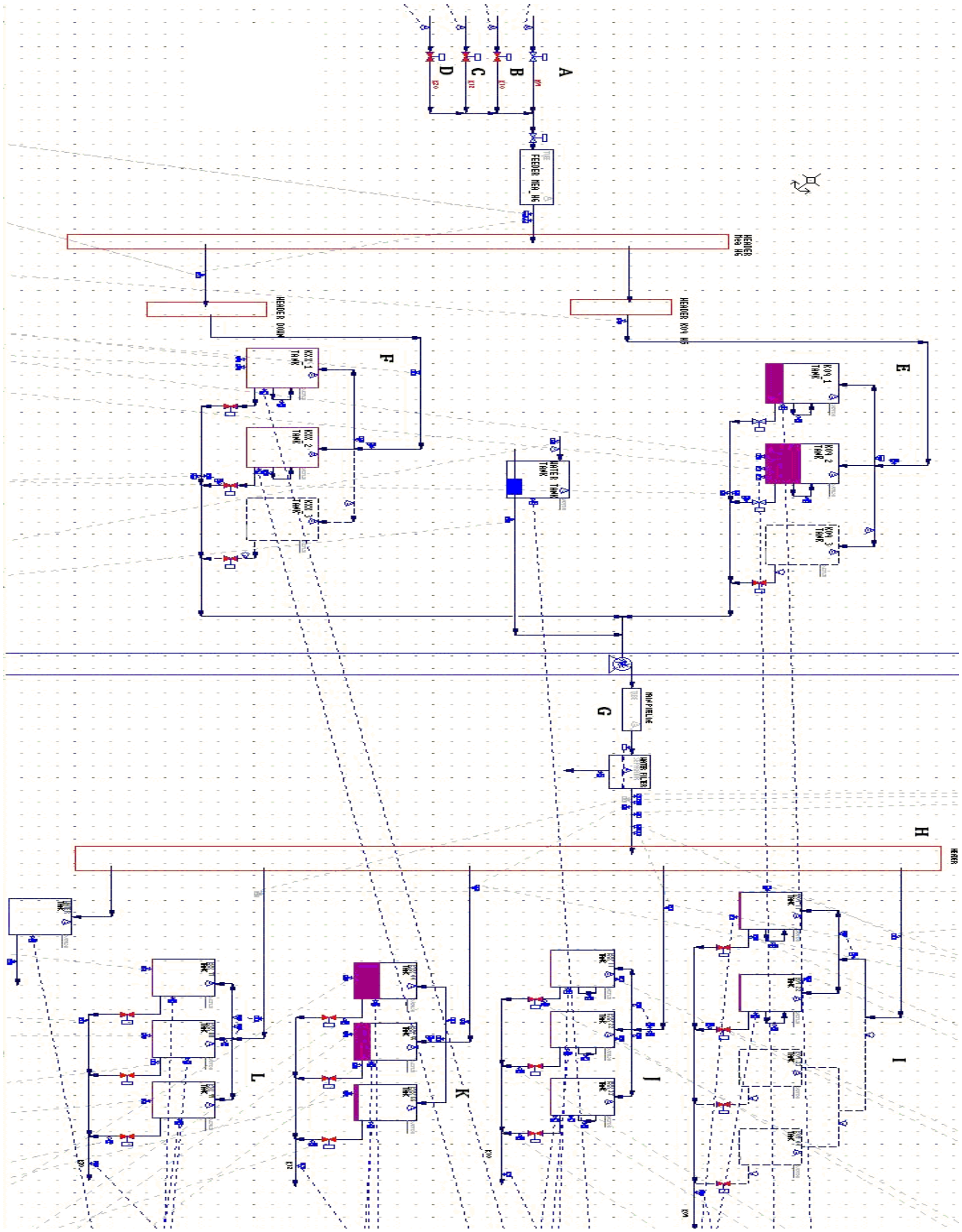


Figure 6.1 OCP model

The OCP model contains four different input pipes (Labeled as A to D in Figure 6.1) that can process raw materials for four different grades of phosphate and also has four different storage tanks (Label as I to L in Figure 6.1) to store four different grades of phosphate product. Each input pipe or output storage tank can only process a certain grade, which is shown in the first three columns in Table 6.1. In addition, different grades, which are determined by the percentages of BPL (Bone Phosphate of Lime) and the percentages of silica, are also shown in Table 6.1. The percentage of the ingredients will not change during the procedure unless two different grades of phosphates are mixed together. That is to say, if the raw materials of one grade are in the system, then only the same grade of product will be obtained from the output. However, two different grades of raw materials might result in the grades of products that are different from all the input grades. For example, if the raw materials of K09 and K20 are put into the system, the phosphate of grade K12 or K20 might be produced. The embedded logic in the system ensures that the products are classified into one of the four grades, which has the most similar percentages of ingredients to the products. For example, if a new product has 71% BPL, then it belongs to K20. The products are first classified according to the BPL and then to the silica. If its percentage of BPL is in between two grades, then it will be classified according to its silica. A new product is classified to K12 if it contains 69% BPL and 2.3% silica.

The input pipes may have different production rates and may operate at different time. Moreover each input pipe needs to be operated for a period of time and then shut down for a while for the cleaning purpose.

Table 6.1 **Formation of the grades**

Input Pipe	Storage Tank	Grade	%BPL	%Silica
A	I	K09	66	3.5
B	J	K10	68	3
C	K	K12	68	2
D	L	K20	72	2.9

During the process, the raw materials need to go through the intermediate tanks (Label as E and F), pipeline (Label as G), and separator (Label as H) before they go to the storage tanks. In this model, materials for Grade K09 go through the intermediate tank E while materials for the other three grades go through the intermediate tank F. Then all of them run through the same pipeline and are split apart according to their ingredients in the separator before they turn into the final products. After all the products are separated, different grades of phosphates go to their destined storage tanks.

As the processing time for the intermediate tanks, pipeline and the separator are different, the production rate of the inputs needs to be scheduled to avoid overflowing the tanks and mixing different grades. Therefore, our objective is to produce a certain amount of different grades of products that meet the demand without causing overflow for each tank. We need to find a schedule that determines the production rate of each input pipes and the duration of running-time and shed-time. Running-time represents the period that materials are fed into the inputs and shed-time represents the period that no material is in the input.

6.2.2. Constraints in the model

The phosphate materials are viscous, so they require certain flowing rate to run through the pipe, otherwise, they will stick to the pipes. And at the same time, there is a limited max flowing rate constrained by the maximal pressure that the pipes can bear. Therefore, in this model the minimal flow rate for each input pipe is 250m^3/h and the maximal flow rate is 280m^3/h.

6.2.3. Demands

The demands are the proposed output required by users and the example of demands for each grade is given in Table 6.2. As one can see from Table 6.2, the demand for each grade of the phosphate is specified.

Table 6.2 Demands of different grades

Output tanks	Grade	Demand
I	K09	305700m^3
J	K10	62500m^3
K	K12	24770m^3
L	K20	4900m^3

As it is shown in Table 6.2, the third column gives the total demand of each phosphate and the first column indicates which tanks the phosphate should be stored.

6.3. Optimization problem formulation

6.3.1. Objective function

Our objective is to maximize the net profit of the system, which is calculated by (total profits – total penalties). The total profit is determined by the demand and production output. If the produced amount is less than the demand, all the products can be sold. However, even if we produce more than the demand, the maximal amount we can sale is the demand amount. So to produce the amount of phosphate which is less than the demand is certainly unwanted because it doesn’t satisfy the demand. In the meantime, overproducing should also be avoided, as it will cost extra money for the raw materials and the storage. Therefore, the objective is to produce the phosphates that are as close to the demand as possible.

Moreover, the overflowing of the tank will result in environmental damage and dealing with the pollution will cost a lot of money and great effort. So avoiding overflowing the tanks is another requirement. The total penalty is calculated by summing up the penalties of all the tanks. The penalty of a tank is determined if a tank overflows and the penalty-cost is the amount of the overflow.

The overall objective function is shown in Eq.(44).

$$f(x) = \sum_{i=1}^{4} |q_i - d_i| + \sum_{j=1}^{6} o_j \quad (44)$$

The first part of the formula is the total difference of the produced amount and the demand amount of phosphates, where q_i is the produced amount of phosphate for the i^{th} grade and d_i is the associated demand. The first to fourth grades corresponds to grade K09, K10, K12 and K20 respectively. The second part of the formula is the total overflows from intermediate tanks. The intermediate tanks are ordered from one to six from the top to bottom and then left to right in Figure 6.1. o_j is the amount of the overflows for the j^{th} intermediate tank.

6.3.2. Design variables

Our purpose is to optimize the production rate of each input grades and the duration of running-time and shed-time, therefore, they will be the design variables for the optimization. The details of how to formulate them will be discussed in this section.

In this model, all the input pipes would have to run for 6 weeks, i.e., for approximately 60600 minutes. Therefore, the time horizon for the problem is 60600 minutes and we use one of continuous time models [95][96] - time-slot representation - for the time representation. According to [97], the scheduling horizon is divided into a finite number of time intervals with the same duration, which are denoted as time slots [98]. In addition, the time-slot representation only allows an event (a.k.a. begin to produce one grade of phosphate) to happen only at the boundaries of these time slots. Each time-slot contains a running-time and a shed-time. As the duration of the time slot is fixed, when running-time is known, the shed-time is automatically determined. Thus, we only need to find the period of running-time. The duration of the running-time is then defined based on the time slot. The design variables indicate how many percentages the running-time takes up in each time slot.

The time horizon is divided into six time-slots in this case. As it is shown in Figure 6.2, the first six variables indicate the duration of the running-time, where x_i represents the running time in the i^{th} time-slot. The rest of time in that time slot is shed time. The last six variables are the production rate with x_{i+6} represents the production rate in the i^{th} time-slot.

Running time						**Production rate**					
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}

Figure 6.2 design variable of the OCP model

The grade of phosphate to be produced in each time-slot is known in this real practical problem. The sequence is K09, K10, K12, K20, K09 and K09. As the sequence of the different grades won’t change the total amount in the end, in fact, the number of time-slots that one grade occupies will influence the output. Therefore, K09 occupies three time-slots and each of the other grades occupies one time-slot, as the demand of K09 is much larger than all the others. Thus, we set the sequence to be K09, K10, K12, K20, K09 and K09.

6.4. Results and Discussion

In order to prove the correctness of discovered information, we turn the black box into white box to understand how the simulation works by analyzing the logic inside the simulation model. It takes human a great effort to understand the complicated logic circuit and the intermediate mechanism. The CA approach, however, discovers useful information automatically only with the sample points.

According to the design, different grades of phosphates will only be mixed in the tanks, and won’t be mixed anywhere else. Moreover, the logic circuit, which is installed on the tanks, can control the overflowing very well. That is to say, the overflowing will not happen very often unless in very extreme cases. Finally, the product of running-time and production rate determines the total produced amount, which will not be affected by the sequence of producing. In other word, if the demand of a certain grade is large, longer running time period and larger production rate are required, otherwise, the demand can’t be satisfied. The demand for K09 is much larger than the other grades, therefore, a larger value for the design variables, which corresponding to the third dimensional bins, should be the promising subdomains. Conversely, if the demand is small, the running time and the production rate should be small to avoid overproducing. Therefore, K20, which has small demand, should have smaller values for its corresponding design variables. The first dimensional bins of those variables should be the promising subdomains. As the range of the time horizon is 0 to 60600 while the production rate ranges from 250 to 280, the time period influences more than the production rate in the product of these two factors. It also indicates that the design variables of time period (x_1 to x_6) are more important. In total 360 sample points are used to generate the CA plot for this twelve variables problem.

In Figure 6.3, the symbols represent different dimensional bins as the legend indicates while labels inside indicate the different variables. As the schedule of producing different grades is known, the meaning of the labels is determined and is shown in the following Table 6.3. There are three time slots that produce K09, therefore, the total produced amount of grade K09 will be the summation of the products in those three time slots.

Table 6.3 **Meaning of the Labels**

Label	Meaning	Label	Meaning
1	Running time for K09 (time slot 1)	7	Production rate for K09 (time slot 1)
2	Running time for K10 (time slot 2)	8	Production rate for K10 (time slot 2)
3	Running time for K12 (time slot 3)	9	Production rate for K12 (time slot 3)
4	Running time for K20 (time slot 4)	10	Production rate for K20 (time slot 4)
5	Running time for K09 (time slot 5)	11	Production rate for K09 (time slot 5)
6	Running time for K09 (time slot 6)	12	Production rate for K09 (time slot 6)

As we can easily observe in Figure 6.3, the third dimensional bins for variables 1, 2, 5 and 6 are the promising subdomains. Although the demand for K09 is much larger than K10, the projection of third dimensional bins for variables 1, 5 and 6 onto the first function value bin vector is slightly larger than the third dimensional bin for variables 2. This is because at all three time-slots K09 is produced, the duration and production rate for each time slot will be smaller than running K09 within one time slot. In addition, in Figure 6.3, the first dimensional bins of the corresponding variables (Labels 3 and 4) are the promising subdomains, which means the running time for K12 and K20 should be shorter. In this model, both K12 and K20 in deed have relatively small demands and require shorter running time. In addition, the first dimensional bin for the fourth variable has larger projection on the first function value bin vector than the third variable. This is quite reasonable as the demand for K20 is much smaller than K12, the running time of K20 will undoubtedly has more tendency to be small value.

The variables of the running time are further apart from the origin, which indicates that those variables are more important than the production rate. This information is also in compliance with the previous analysis of the black box. The third dimensional bins of 7, 11, and 12 variables are closer to the first function values bin vector, so the production rate for K09 should be relatively larger than others. As the first function value bin is on the left while the third function value bin is on the right. For convenience, we consider the dimensional bins on the left side have better function value. In Figure 6.3, third dimensional bins for 7, 8, 9 and 10 are from left to right while the first dimensional bins for those variables are from the right to left, which means that the seventh variable should have large value and the tenth variable should have small value. Therefore, it reveals that the production rate of K09 should be larger than K10 and then K12 and K20. This is agreed with the demands of different grades of phosphates.

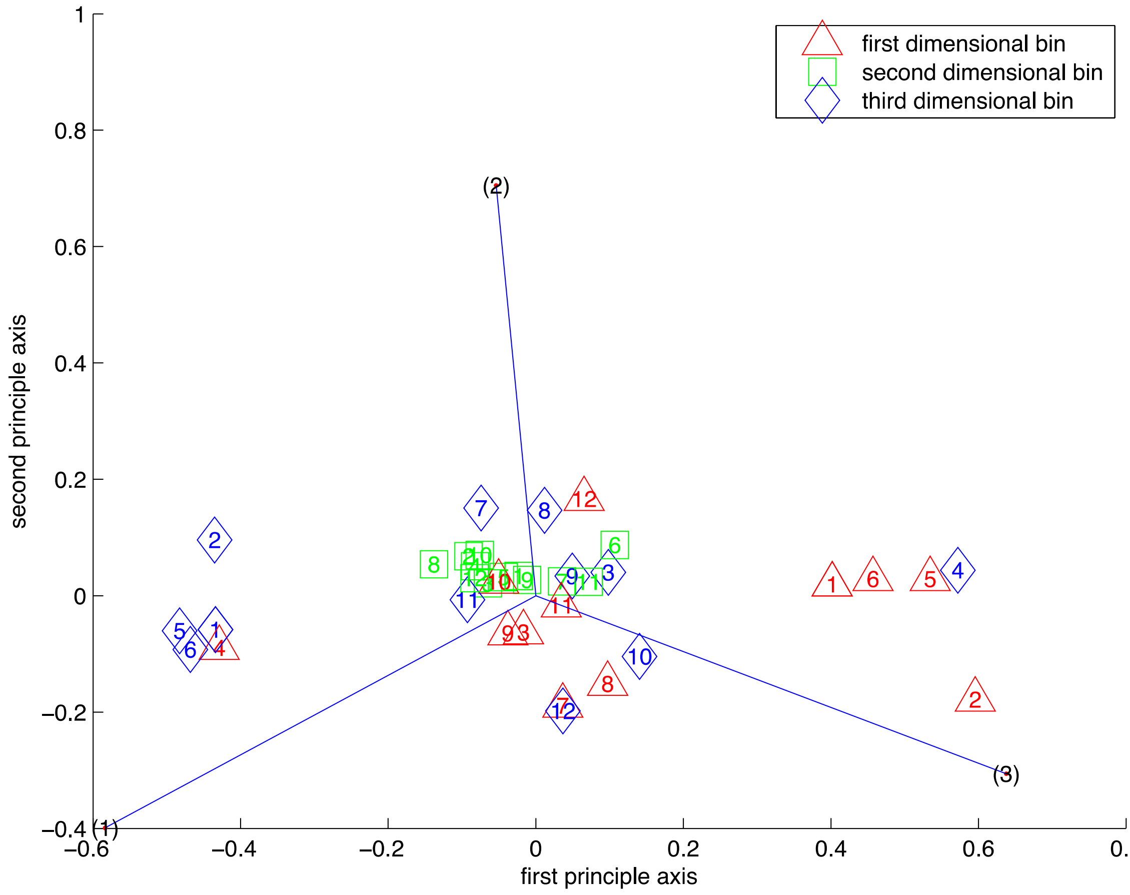
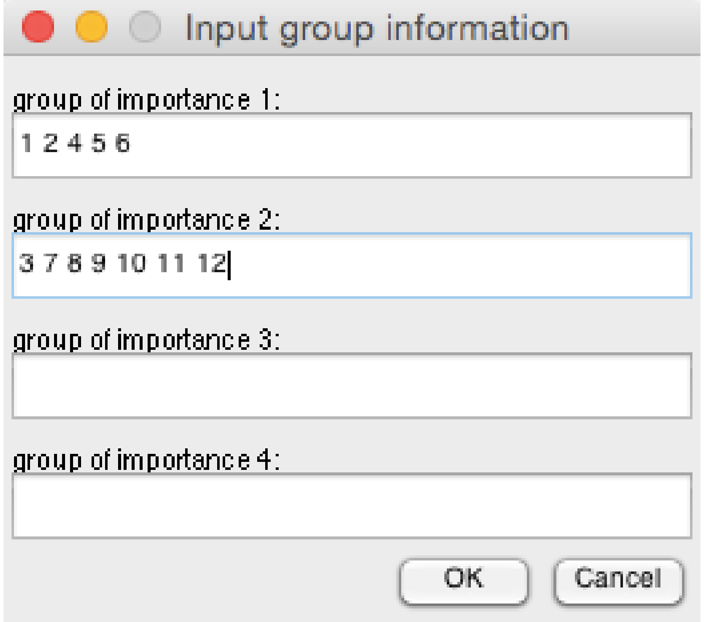


Figure 6.3 CA results of the OCP model

In conclusion, the proposed method can work well on both mathematical function and real engineering problems. It doesn't require previous knowledge of the black box and can discover the useful information with limited number of sample points.

According to the knowledge discovered above, we can conclude that the third dimensional bins for x_1, x_2, x_5, x_6, x_7 and x_{11}, the second dimensional bin for x_8 and the first dimensional bins for x_4 and x_{10} constitute promising regions. The x_1, x_2, x_4, x_5, and x_6 are obviously the important variables because they have larger inertia. The user input for the discovered important variables and the promising subdomain will be shown in Figure 6.4 and Figure 6.5.



Input group information

group of importance 1:

1 2 4 5 6

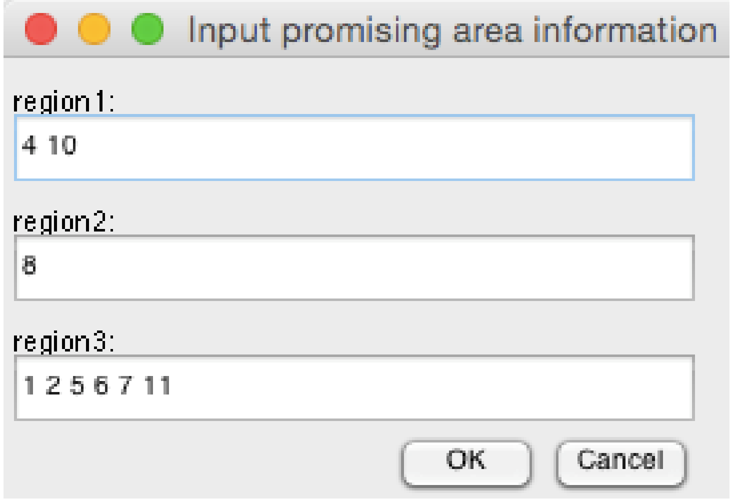
group of importance 2:

3 7 8 9 10 11 12

group of importance 3:

group of importance 4:

Figure 6.4 Input window of the important variables



Input promising area information

region1:

4 10

region2:

8

region3:

1 2 5 6 7 11

Figure 6.5 Input window of the promising subdomain of the variables

To solve this problem, both TRMPS2 and CA_TRMPS are applied separately to find the optimal result and the corresponding schedule. The results found by CA_TRMPS are then compared with TRMPS2's results to illustrate that after integrating the information into TRMPS2, better results can be found. There are 12 variables and the objective function is the same as discussed in 6.3. The same settings are applied to both methods for a fair comparison with 900 total sample points for each method.

Table 6.4 Comparing Results

	TRMPS2	CA_TRMPS
Objective value	1.730309837527980e+08	1.057506951782529e+08
Running time for K09 (time slot 1)	104.338705992196	179.153012662911
Running time for K10 (time slot 2)	133.228027946394	154.594731667562
Running time for K12 (time slot 3)	87.8411800965949	60.7942797500888
Running time for K20 (time slot 4)	170.712282645577	16.1368573288729
Running time for K09 (time slot 5)	98.9451028103020	178.602975670858
Running time for K09 (time slot 6)	120.128323311059	171.821803518285
Production rate for K09 (time slot 1)	254.951640978747	275.853009573573
Production rate for K10 (time slot 2)	267.131794841914	269.017148728942
Production rate for K12 (time slot 3)	279.348226465635	257.850396858275
Production rate for K20 (time slot 4)	278.749885567879	253.777864324671
Production rate for K09 (time slot 5)	274.076148954265	257.103844175861
Production rate for K09 (time slot 6)	270.624835261167	274.544478985383

The objective value for CA_TRMPS is 6.7280e+07 smaller than the objective value for TRMPS2, which illustrates that the CA_TRMPS is better than TRMPS2 for this problem. The schedule obtained by CA_TRMPS is also much more reasonable. As much more K09 is needed, the running time and the production rate are both relatively larger than other grades. In addition, the demand for lower grade of the phosphates is larger, therefore, the running time should be relatively longer and the production rate should be larger. The results in CA_TRMPS agree with the analysis while the results in TRMPS2 are not reasonable. The running time for K20 should be the smallest, however, in TRMPS2, it is the largest among all the running time in different time slots. The running time for K09, in contrast, is not large. The production rates also do not make sense since the lower grade of phosphates with larger demand have even smaller production rate. Therefore, the CA_TRMPS can find a better schedule within a limited number of sample points than TRMPS2.

The convergence plot for TRMPS2 and CA_TRMPS are shown in Figure 6.6, from which we can easily find that CA_TRMPS can obtain much better results after around 180 sample points. The first 180 samples are random sampling. Therefore, the results vary. However, after integrating the discovered information, the objective values drop to around 1.2*10^8, which also illustrate that the information is very essential to the optimization process.

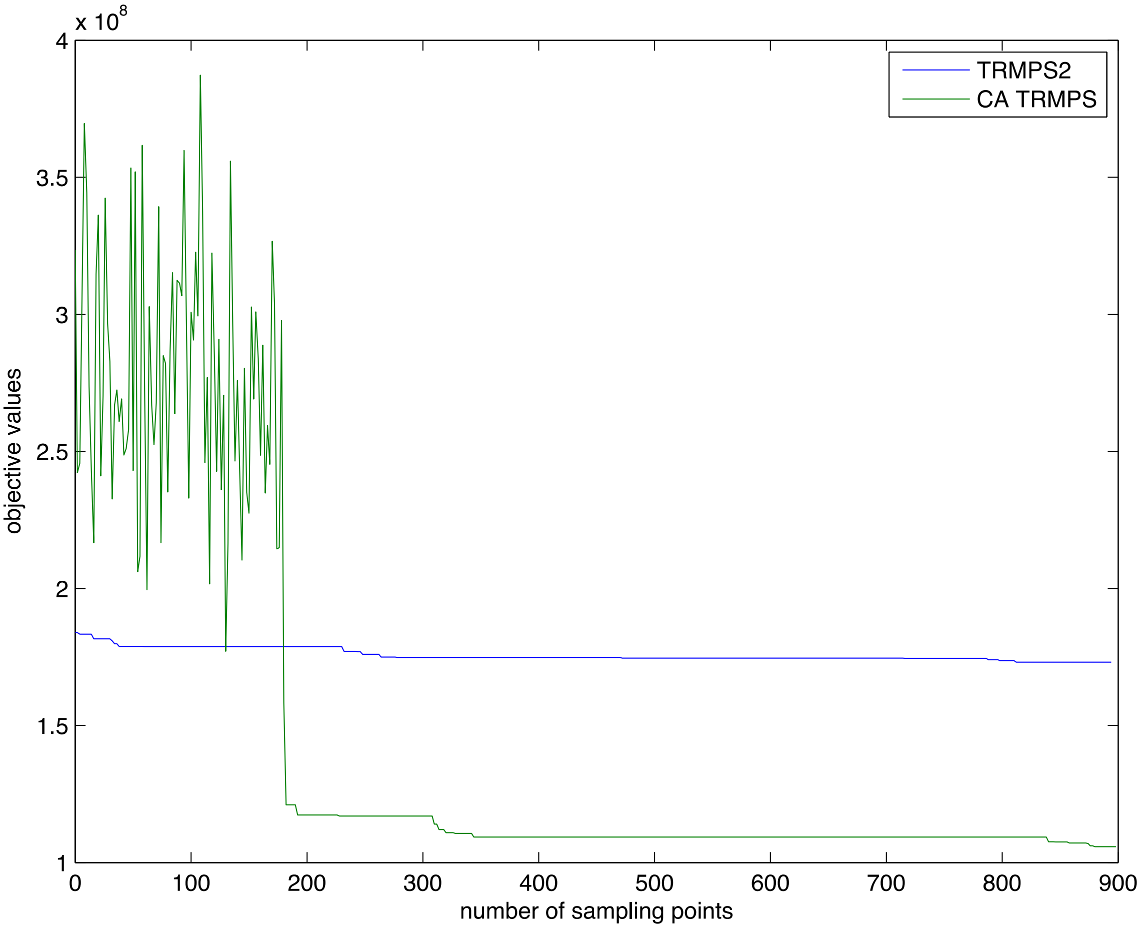


Figure 6.6 Convergence plot of TRMPS2 and CA_TRMPS

6.5. Summary

CA is shown to be able to help to discover the information in real engineering problem. The information is reasonable according to intuitive understanding and engineering insight. The next chapter will draw a conclusion for the whole thesis and summarize the distinguished features of the proposed method.

Chapter 7. Conclusion

7.1. Summary

This thesis first introduces the correspondence analysis in HEB problems with the goal of providing more transparency about the problem. By analyzing data during optimization, more information can be abstracted to assist the optimization process. In addition, the visualization in CA provides the designer with a more intuitive way of understanding the high dimensional data. The method can be used to direct the search in a smaller but more promising subdomain, mine the symmetric variables, and identify the significant variables that contribute more to the function value. New features that were developed in this method that separate this work from others are listed as follows.

In this work, one of emphases is placed on defining subdomains of the design spaces to narrow down the search, which will help to accelerate optimization progress. The current dimension reduction techniques, which are applied directly on the design space, can't guarantee that in the lower dimension, the function value is still continuous as it is in the original design space. Therefore, the contribution of the proposed method is that it only analyzes distribution of each bin in design space in relation to function value, so the dimension reduction process won't change the original design space. The presumption of the optimization is still valid. The significant variable and the symmetric variables can be identified using this method, which can significantly reduce the number of sampling points by reducing the number of different combinations of the samples. The most promising of this method is that it requires a small number of points and the further analysis of smaller region of the design space doesn't require more points. In other words, all the sampling points can be reused to obtain more accurate information of the data in the smaller region. Additionally, users may use two-dimensional plot to understand the data structure and make informed decision.

Corresponding Analysis (CA) is shown to be very useful by being applied to an optimization process and a practical engineering problem. Applying CA to the meta-modeling optimization method can reduce the sampling points and accelerate the optimization process. Fewer points are needed for the less important variables. And as for the symmetric variable, they can be treated as one factor of the function value and be assigned to a same value in the optimization process. This approach can in a sense reduce the dimensionality of the design space. After applying the discovered information to the optimization method, it shows great improvement comparing to the current optimization method especially for the high dimensional problems. For the HEB problem, the prosed method is more promising because it can reduce a large number of sampling points, which can save a significant amount of time for the expensive black-box simulation.

An application was presented for the optimization of high dimensional scheduling problem. This example shows how CA may assist in providing more information for the problem. It identifies the significant variables and discovers promising sub-domain very well. Clearly, using the discovered information can reduce a large number of sampling points. All of these show that the CA method can be beneficial for understanding the black box problems, especially for the HEB problems.

7.2. Future Work

This work applies correspondence analysis on single objective optimization. Many improvements may still be made. For instance, one limitation is that the visualization may become incomprehensible when the number of variables is more than 20. For a problem with 100 variables, the scatter plots become too dense to read and hard to analyze. To overcome this, a more scalable visualization technique is required. Finally, the proposed method can be integrated with optimization methods other than TRMPS2. Therefore, application of CA to other optimization method needs to be further tested. Different strategies of using the discovered information might result in different performance of the optimization algorithm. Therefore, strategies suitable for specific algorithms are to be developed.

References

[1] Battiti, R., Brunato, M., & Mascia, F. (2008). Reactive search and intelligent optimization (Vol. 45). Springer Science & Business Media.

[2] Storn, R., & Price, K. (1995). Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces (Vol. 3). Berkeley: ICSI.

[3] Wang, G. G., & Shan, S. (2007). Review of metamodeling techniques in support of engineering design optimization. Journal of Mechanical Design, 129(4), 370-380.

[4] Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4), 455-492.

[5] Koch, P. N., Simpson, T. W., Allen, J. K., Mistree, F. (1999) Statistical approximations for multidisciplinary design optimization: the problem of size. J Aircr 36(1):275–286

[6] Shan, S., & Wang, G. G. (2010). Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Structural and Multidisciplinary Optimization, 41(2), 219-241.

[7] Bates, R. A., Buck. R. J., Riccomagno, E., Wynn, H. P. (1996) Experimental design and observation for large systems. J R Stat Soc B 58(1): 77–94

[8] Booker, A. J., Dennis, J. E. J., Frank, P. D., Serafini, D. B., Torczon, V., Trosset, M. W. (1999) A rigorous framework for optimization of expensive functions by surrogates. Struct Optim 17(1): 1–13

[9] Brusoni, S., & Prencipe, A. (2001). Unpacking the black box of modularity: technologies, products and organizations. Industrial and Corporate Change, 10(1), 179-205. Shorter, J. A., Ip, P. C., Rabitz, H. A. (1999) An efficient chemical kinetics solver using high dimensional model representation. J Phys Chem A 103:7192–7198

[10] Lebart, L. (2013). Correspondence analysis. In Data Science, Classification, and Related Methods: Proceedings of the Fifth Conference of the International Federation of Classification Societies (IFCS-96), Kobe, Japan, March 27–30, 1996 (p. 423). Springer Science & Business Media.

[11] Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., & Tatham, R. L. (2006).Multivariate data analysis (Vol. 6). Upper Saddle River, NJ: Pearson Prentice Hall.

[12] Bartlett, M. S. (1935). Contingency table interactions. Supplement to the Journal of the Royal Statistical Society, 248-252.

[13] Greenacre, M. (2007). Correspondence analysis in practice. CRC press.

[14] Armstrong, J. S. (2011). Illusions in regression analysis. Available at SSRN 1969740.

[15] Schroeder, W. J., Lorensen, B., & Martin, K. (2004). The visualization toolkit. Kitware.

[16] Card, S. K., & Mackinlay, J. (1997). The structure of the information visualization design space. In Information Visualization, 1997. Proceedings., IEEE Symposium on (pp. 92-99). IEEE.

[17] Gielen, G., McConaghy, T., & Eeckelaert, T. (2005). Performance space modeling for hierarchical synthesis of analog integrated circuits. In Proceedings of the 42nd annual design automation conference (pp. 881-886). ACM.

[18] Elmqvist, N., Dragicevic, P., & Fekete, J. D. (2008). Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation.Visualization and Computer Graphics, IEEE Transactions on, 14(6), 1539-1148.

[19] Cheng, G. H., Younis, A., Hajikolaei, K. H., & Wang, G. G. (2015). Trust Region Based Mode Pursuing Sampling Method for Global Optimization of High Dimensional Design Problems. Journal of Mechanical Design, 137(2), 021407.

[20] Chrzanowska-Jeske, M. (2001). Generalized symmetric variables. In Electronics, Circuits and Systems, 2001. ICECS 2001. The 8th IEEE International Conference on (Vol. 3, pp. 1147-1150). IEEE.

[21] Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P. Y., ... & Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. Automatica, 31(12), 1691-1724.

[22] Li, G., Wang, S. W., Rosenthal, C., Rabitz, H. (2001) High dimensional model representations generated from low dimensional data samples. I. mp-Cut-HDMR. J Math Chem 30(1): 1–30

[23] Hill, W. J., Hunter, W. G. (1966) A review of response surface methodology: a literature survey. Technometrics 8(4):571–590

[24] Fang, H., Horstemeyer, M. F. (2006) Global response approximation with radial basis functions. J Eng Optim 38(4):407–424

[25] Martin, J. D., Simpson, T. W. (2005) Use of kriging models to approximate deterministic computer models. AIAA J 43(4):853–863

[26] Li, G., Rosenthal, C,. Rabitz, H. (2001) High dimensional model representations. J Phys Chem A 105(33): 7765–7777

[27] Rabitz, H., Alis, Ö. F. (1999) General foundations of high-dimensional model representations. J Math Chem 25:197–233

[28] Fodor, I. (2002) A survey of dimension reduction techniques. Center for Applied Scientific Computing, Lawrence Livermore National, Technical Report UCRL-ID-148494

[29] Cunningham, P. (2007) Dimension Reduction. University College Dublin, Technical Report UCD-CSI-2007-7

[30] Schonlau, M. and Welch, W. J. (2006), Screening the Input Variables to a Computer Model Via Analysis of Variance and Visualization, in Screening Methods for Experimentation in Industry, Drug Discovery and Genetics, eds. Dean, A. M. and Lewis, S. M., Springer-Verlag, 34:308–327.

[31] Tu, J., Jones, D. R. (2003) Variable screening in metamodel design by cross-validated moving least squares method. In: The 44th AIAA/ASME/ASCE/AHS structures, structural dynamics, and materials conference, Norfolk, Virginia, 7–10 April

[32] Shen, H. T., Zhou, X., Zhou, A. (2007) An adaptive and dynamic dimensionality reduction method for high-dimensional indexing. The VLDB Journal 16(2): 219-234

[33] Wang, Y., Cai, Z., & Zhou, Y. (2009). Accelerating adaptive trade‐off model using shrinking space technique for constrained evolutionary optimization. International journal for numerical methods in engineering, 77(11), 1501-1534.

[34] Marin, F. T. S., Gonzalez, A. P. (2003) Global optimization in path synthesis based on design space reduction. Mech Mach Theory 38: 579–594

[35] Bandler, J. W., Biernacki, R. M., Chen, S. H., Grobelny PA, Hemmers RH (1994) Space mapping technique for electromagnetic optimization. IEEE Trans Microwave Theor Tech 42(12): 2536–2544

[36] Saha, A., Wu, C. L., Tang, D. S. (1993) Approximation, dimension reduction, and nonconvex optimization using linear superpositions of Gaussians. IEEE Trans Comput 42(10): 1222–1233

[37] Stump, G., Simpson, T. W., Yukish, M., Bennett, L. (2002) Multidimensional design and visualization and its application to a design by shopping paradigm. In: The 9th AIAA/USAF/NASA/ISSMO sympo- sium on multidisciplinary analysis and optimization, Atlanta, GA, 4–6 September, AIAA 2002-5622

[38] Vasey, M. W., & Thayer, J. F. (1987). The continuing problem of false positives in repeated measures ANOVA in psychophysiology: A multivariate solution. Psychophysiology, 24(4), 479-486.

[39] Ding C, He X, Zha H, Simon HD (2002) Adaptive dimension reduction for clustering high dimensional data. In: The 2002 IEEE international conference on data mining (ICDM'02), Maebashi City, Japan, 9–12 December. IEEE, pp 147–154

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

[41] Dunteman, G. H. (1989) Principal components analysis. Sage, London

[42] Somorjai, R. L., Dolenko, B., Demko, A., Mandelzweig, M., Nikulin, A. E., Baumgartner, R. et al (2004) Mapping high-dimensional data onto a relative distance plane—an exact method for visualizing and characterizing high-dimensional patterns. J Biomed Inform 37:366–376

[43] Rassokhin, D. N., Lobanov, V. S., Agratiotis, D. K. (2000) Nonlinear mapping of massive data sets by fuzzy clustering and neural networks. J Comput Chem 22(4):373–386

[44] Wang, G. G., Dong, Z., Aitchison, P. (2001) Adaptive response surface method—a global optimization scheme for computation-intensive design problems. J Eng Optim 33(6):707–734

[45] Wang, G. G., Simpson, T. W. (2004) Fuzzy clustering based hierarchical metamodeling for space reduction and design optimization. J Eng Optim 36(3):313–335

[46] Wang, G. G., Shan, S. (2004) Design space reduction for multi-objective optimization and robust design optimization problems. SAE Trans 113:101–110

[47] Winer, E. H., Bloebaum, C. L. (2002) Development of visual design steering as an aid in large-scale multidisciplinary design optimization. Part I: method development. Struct Multidisc Optim 23(6):412–424

[48] Winer, E. H., Bloebaum, C. L. (2002) Development of visual design steer- ing as an aid in large-scale multidisciplinary design optimization. Part II: method validation. Struct Multidisc Optim 23(6): 425–435

[49] de Leeuw, J. (1983). On the prehistory of correspondence analysis. Statistica Neerlandica, 37, 161–164.

[50] Hirschfeld, H.O. (1935). A connection between correlation and contingency. Proceedings of the Cambridge Philosophical Society, 31, 520–524.

[51] Fisher, R.A. (1940). The precision of discriminant functions. *Annals of Eugenics*, 10, 422–429.□

[52] Benze´cri, J.P. (1973). L'Analyse des donnees: II. La Taxonomie. Paris: Dunod.

[53] Benze´cri, J.P. (1973b). L'Analyse des donnees: I. La Taxonomie. Paris: Dunod.

[54] Benze´cri, J.P. (1992). Correspondence Analysis Handbook. New York: Marcel Dekker.

[55] Hill, M.O. (1974). Correspondence analysis: a neglected multivariate method. Applied Statistics, 23, 340–354.

[56] Burt, C. (1950). The factorial analysis of qualitative data. J. Statist. Psychology, 3, 166–185.

[57] Jolliffe, I. T. (2002). Principal component analysis and factor analysis. Principal component analysis, 150-166.

[58] Borg, I., & Groenen, P. J. (2005). Modern multidimensional scaling: Theory and applications. Springer Science & Business Media.

[59] Usunoff, E. J., & Guzmán ‐ Guzmán, A. (1989). Multivariate analysis in hydrochemistry: an example of the use of factor and correspondence analyses. Groundwater, 27(1), 27-34.

[60] Dziuban, C. D., & Shirkey, E. C. (1974). When is a correlation matrix appropriate for factor analysis? Some decision rules. Psychological Bulletin, 81(6), 358.

[61] Newey, W. K., & West, K. D. (1986). A simple, positive semi-definite, heteroskedasticity and autocorrelationconsistent covariance matrix.

[62] Kaiser, H. F., & Caffrey, J. (1965). Alpha factor analysis. Psychometrika, 30(1), 1-14.

[63] Jöreskog, K. G. (1969). Efficient estimation in image factor analysis. Psychometrika, 34(1), 51-75.

[64] Karovičová, J. O. L. A. N. A., Kohajdová, Z. L. A. T. I. C. A., & Greif, G. (2002). The use of PCA, FA, CA for the evaluation of vegetable juices processed by lactic acid fermentation. Czech j. food sci, 20(4), 135-143.

[65] Oksanen, J. (1983). Ordination of boreal heath-like vegetation with principal component analysis, correspondence analysis and multidimensional scaling. Vegetatio, 52(3), 181-189.

[66] Mellenbergh, G. J. (1982). Contingency table models for assessing item bias. Journal of Educational and Behavioral Statistics, 7(2), 105-118.

[67] Michael G. and Trevor H. (1987). The Geometric Interpretation of Correspondence Analysis. Journal of the American Statistical Association, 82(398): 437-447

[68] Golub, G. H., & Reinsch, C. (1970). Singular value decomposition and least squares solutions. Numerische mathematik, 14(5), 403-420.

[69] Matt, F. (2010). Advanced Topic in Machine Learning – Dimensionality Reduction. Retrieved July 25, 2010, from http://courses.cms.caltech.edu/cs253/slides/cs253-06-dimreduction.pdf

[70] Market Research. Correspondence Analysis. Retrieved April 2, 2013, from http://mktresearch.org/wiki/Correspondence_Analysis

[71] Stein, M. (1987). Large sample properties of simulations using Latin hypercube sampling. Technometrics, 29(2), 143-151.

[72] Baker, K. (2005). Singular value decomposition tutorial. The Ohio State University, 2005, 1-24.

[73] Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. Psychometrika, 1(3), 211-218.

[74] Mike, B. (2003). A practical guide to the use of Correspondence Analysis in marketing research. Marketing Bulletin, 14(2), 16-38

[75] Picheny, V., Wagner, T., & Ginsbourger, D. (2013). A benchmark of kriging-based infill criteria for noisy optimization. Structural and Multidisciplinary Optimization, 48(3), 607-626.

[76] Johnson, S. C. (1967). Hierarchical clustering schemes. Psychometrika, 32(3), 241-254.

[77] Murtagh, F., & Legendre, P. (2011). Ward's hierarchical clustering method: Clustering criterion and agglomerative algorithm. arXiv preprint arXiv:1111.6285.

[78] Ng, M. K., Li, M. J., Huang, J. Z., & He, Z. (2007). On the impact of dissimilarity measure in k-modes clustering algorithm. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29(3), 503-507.

[79] Niedermeier, R., & Sanders, P. (1996). On the Manhattan Distance Between Points on Space Filling Mesh Indexings. Univ., Fak. für Informatik.

[80] Korenius, T., Laurikkala, J., & Juhola, M. (2007). On principal component analysis, cosine and Euclidean measures in information retrieval. Information Sciences, 177(22), 4893-4905.

[81] Eubank, R. L., LaRiccia, V. N., & Rosenstein, R. B. (1987). Test statistics derived as components of Pearson's phi-squared distance measure. Journal of the American Statistical Association, 82(399), 816-825.

[82] Gower, J. C., & Ross, G. J. S. (1969). Minimum spanning trees and single linkage cluster analysis. Applied statistics, 54-64.

[83] Krznaric, D., & Levcopoulos, C. (1998). Fast algorithms for complete linkage clustering. Discrete & Computational Geometry, 19(1), 131-145.

[84] Sokal, R. R., & Michener, C. D. (1958). A statistical method for evaluating systematic relationships. Kansas University Science Bulletin 38, I409-1438

[85] Wang, L., Shan, S., and Wang, G., 2004, “Mode-Pursuing Sampling Method for Global Optimization on Expensive Black-Box Functions,” J. Eng. Optim., 36(4), pp. 419–438.

[86] Sharif, B., Wang, G. G., and ElMekkawy, T. Y., 2008, “Mode Pursuing Sampling Method for Discrete Variable Optimization on Expensive Black-Box Functions,” ASME J. Mech. Des., 130(2), p. 021402.

[87] Duan, X., Wang, G. G., Kang, X., Niu, Q., Naterer, G., & Peng, Q. (2009). Performance study of mode-pursuing sampling method. Engineering Optimization, 41(1), 1-21.

[88] Baker, J. E. (1987, July). Reducing bias and inefficiency in the selection algorithm. In Proceedings of the second international conference on genetic algorithms (pp. 14-21).

[89] Addis, B., & Leyffer, S., 2006, “A Trust Region Algorithm for Global Optimization,” Comput. Optim. Appl., 35(3), pp. 287–304.

[90] Alexandrov, N., Dennis, J., Lewis, R., & Torczon, V., 1997, “A Trust Region Framework for Managing the Use of Approximation Models in Optimization,” Struct. Optim., 15(1), pp. 16–23.

[91] Rodriguez J., Watson, L., & Renaud, J. E., 1998, “Trust Region Augmented Lagrangian Methods for Sequential Response Surface Approximation and Optimization,” ASME Design Engineering Technical Conferences, Vol. 120(1), pp.58–66

[92] Fu, J., & Wang, L., 2002, “A Random-Discretization Based Monte Carlo Sampling Method and Its Applications,” Methodol. Comput. Appl. Probab., 4(1), pp. 5–25.

[93] Ferreira, P. M., & Liu, C. R. (1986). An analytical quadratic model for the geometric error of a machine tool. Journal of Manufacturing Systems, 5(1), 51-63.

[94] Jin, R., Chen, W., & Simpson, T. W. (2001). Comparative studies of metamodelling techniques under multiple modelling criteria. Structural and Multidisciplinary Optimization, 23(1), 1-13.

[95] Wasik, L. (1980). Aurel Systems - Products - CADSIM Plus. Retrieved July 15, 2015, from http://www.aurelsystems.com/csplus.htm

[96] Floudas, C. A., & Lin, X. (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Computers & Chemical Engineering, 28(11), 2109-2129.

[97] Shah, N., Pantelides, C. C., & Sargent, W. H. (1993). A general algorithm for short-term scheduling of batch operations-II. Computational issues. Computers and Chemical Engineering, 2, 229–244.

[98] Sundaramoorthy, A., & Karimi, I. A. (2005). A simpler better slot-based continuous-time formulation for short-term scheduling in multiproduct batch plants. Chemical and Engineering Science, 60, 2679– 2702.

[99] Pinto, J. M., & Grossmann, I. E. (1995). A continuous time mixed integer linear programming model for short-term scheduling of multistage batch plants. Industrial and Engineering Chemistry Research, 34, 3037–3051.