2017-12-22

A Unified Hybrid Evolutionary Multi-Objective Optimization Algorithm for Decision Making

Peng, Xiufeng

UNIVERSITY OF CALGARY

A Unified Hybrid Evolutionary Multi-Objective Optimization Algorithm for Decision Making

by

Xiufeng Peng

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN COMPUTER SCIENCE

CALGARY, ALBERTA

December, 2017

Abstract

Multi-objective optimization involves the simultaneous optimization of two or more objectives. The objectives to be optimized are often conflicting to each other, whereas there is no single optimal solution in the search space that are superior to other solutions when all objectives are considered. Therefore, a set of trade-of optimal solutions, also known as Pareto-optimal solutions, are required to give decision makers an informed decision-making process within the acceptable time frame. Evolutionary algorithms, also known as genetic algorithms, are well suited to address the multiplicity of objectives in solutions in its search procedure and therefore used for solving multi-objective optimization problems. There have been a number of research conducted on using genetic algorithms to solve multi-objective optimization problems in the past decades, and many variations of multi-objective genetic algorithms in literature. However, the application of multi-objective genetic algorithm to solve real world problems is not documented much or cited due to the challenges and complexities presented in real world situations. Furthermore. real-world situations often require the Pareto-set to be obtained in a timely and efficient manner for decision makers, and subject expertise is required to be incorporated in the initialization and search process interactively. Furthermore, the visualization of Pareto-optimal sets is an important aspect in the decision-making process for decision makers to use and understand the impacts of choosing the solutions from the Pareto-optimal sets.

This research presents an innovative unified hybrid framework with a novel multi-objective genetic algorithm with an integrated expert module that can be applied directly in helping decision makers to solve real world multi-objective optimization problems. The validity and effectiveness of the proposed algorithm are verified by conducting experiments with three well cited benchmark data sets and comparing with previous studies in the literature. The experiments

conducted with benchmark datasets proved that it can find a much better spread of solutions and better convergence near the true Pareto-optimal front compared to other research on Pareto evolutionary algorithm.

The implementation of the proposed framework uses parallel and asynchronous design to achieve high performance computing and faster convergence of the algorithm. Sets of unique parallel, asynchronous parallel genetic and K-mean operators are used to reach a global optimality through population diversity.

An expert module is included in the framework to integrate domain subject matter experts' knowledge, experiences and preferences. The integration makes more it realistic and practical to apply this framework to solve real-world problems. A graphical reporting submodule for data visualization on the generated Pareto-optimal set is included in the expert module to visualize the results and outcomes for decision makers. The proposed framework was applied to solve two real-world multi-objective optimization problems with expected Pareto-optimal solution sets.

Acknowledgements

I would like to thank my supervisor, Dr. Reda Alhajj, for his time, care, encouragement and supervision, especially during the difficult and stressful times during my study. The research would not be possible without Dr. Reda Alhajj's support and advice.

I would also like to thank Dr. Jon Rokne, for his support throughout the entirety of the research, especially in collaboration with industries. Apart from the unique algorithm developed in this research, the outcome framework of this research is also designed to be applied to solve real world multi-objective optimization problems. Thanks to Dr. Rokne's extensive industrial experience, the implementation of the framework was completed successfully.

Additionally, I would like to extend thanks to all of the defense committee for their time in reviewing this work and valuable feedback.

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols, Abbreviations and Nomenclature

MOO	*Multi-Objectives Optimization*
GA	*Genetic Algorithm*
x	*Decision vector*
y	*Objective vector*
X	*Decision space*
Y	*Objective space*
DM	*Decision Making*
MOHKGA	*Multi-objective Hybrid K-means Genetic Algorithm*
EA	*Evolutionary Algorithm*
S	*Population Size*
G	*Maximum Number of Generations*
p_r	*Crossover Probability*
Rm	*Mutation Rate*
FFGA	*Fonseca and Fleming's genetic algorithm*
NPGA	*Niched Pareto Genetic Algorithm*

NSGA	*Non-dominated Sorting Genetic Algorithm*
SPEA	*Strength Pareto Evolutionary Algorithm*
VEGA	*Vector Evaluated Genetic Algorithm*
MCDA	*Multi-Criteria Decision Analysis*
NLP	*Non-Linear Programming Problem*
$d(C_i,C_j)$	*Distance between Cluster* C_i *and* C_j
$d_{max}(X_n)$	$MAX_k\{d(X_n, C_k)\}$
DB	*DB Index*
S(*i*)	*Silhouette Index*
SD	*SD Index*
TWCV	*Total Within-Cluster Variation*
ID	*Inter-cluster Density*
APN	*Average Proportion of Non-Overlap*
AD	*Average Distance*
ADM	*Average Distance between Means*

Epigraph

“Perform good deeds, never worry about the future”

-Tao Feng <<Tian Dao>>

Chapter One: INTRODUCTION

1.1 Background

Multi-objective optimization is to get a number of optimized objectives simultaneously. The objectives are often conflicting with each other; the improvement of one objective function often results in the degradation of other objective values. The solution to this type of problem usually involves searching for simultaneous optimization of several conflicting and competing objectives. However, there exists a set of solutions that no single best solution in the search space is better than the other with all objectives considered. This set of solutions are known as Pareto-optimal solutions. By using the generated Pareto-optimal solutions, the decision maker will take trade-offs of all the objectives considered in solving real-world problems based on their knowledge, experiences and expertise.

There are a number of methods to solve multi-objective optimization problems. Classical methods of solving multi-objective optimization problems use the algorithm to convert multi-objective optimization to a single-objective optimization problem by emphasizing one particular Pareto-optimal solution at a time, which requires multiple-runs to produce the alternative solutions for every objective. Real-world multi-objective optimization problems usually are multi-dimensional and multi-modal, and sometimes there is no clear definition of certain objective functions, classical methods are usually inadequate to produce the Pareto-optimal solutions for real world problem solving due to the high computational complexity [1]. Because there does not exist a single solution that simultaneously optimizes each objective in a nontrivial

multi-objective optimization problem, there have been various research and algorithms that were applied to solve the multi-objective optimization problems [2-8].

The solutions of multi-objective optimization problems are considered equally good without additional subjective preference information, which are also referred as non-dominated, Pareto optimal, Pareto efficient or non-inferior solutions. Among the Pareto optimal solutions, decision makers determine the more applicable or favorable solution by taking trade-offs and sacrifices.

Due to limitations of classical methods in searching for the Pareto-optimal sets for difficult problems with non-convex, discontinuous, and multi-modal solutions spaces, as discussed in section 1.4.1., Genetic algorithm are well suited to solve multi-objective optimization problems. Genetic algorithm can find a set of multiple non-dominated solutions in a single run by simultaneously searching different regions of a solution space. Over the past decade, the application of genetic algorithm (GA) in solving multi-objective optimization (MOO) problems and data mining processes for knowledge discovery has become a popular research area. There were a number of algorithms developed in the research to address the multiple and often conflicting objectives optimization problems [3-27].

In this research, a novel hybrid and unified multi-objective genetic algorithm based framework is proposed. The applicability and effectiveness of the proposed framework are demonstrated by conducting experiments on three well-known benchmark datasets, the results are then compared with other well-known multi-objective genetic algorithms as comparative studies for validation and verification. Then the framework is applied to solve two real-world problems.

1.2 Multi-objective Optimization Problems (MOOP)

A multi-objective optimization problem is an optimization problem that involves multiple objective functions. The solution is to get the optimized value of more than one objective function simultaneously. For a nontrivial multi-objective optimization problem, there does not exist a single solution that is better or dominates other solutions with all objectives considered. It is also an area of multiple criteria decision-making processes as multi-objective optimization has been applied in many fields of science, including engineering, economics and logistics, where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives.

In mathematical terms, a multi-objective optimization problem formulated as a set of n decision variables, and a set of k objective functions and a set of m constraints. Objective functions and constraints are functions of the decision variables [101].

If $k = 1$, the problem becomes a single objective optimization problem, and the feasible set is reduced to one solution where one meets the objective function $f(x)$ that gives the maximum or minimum value of the objective function $f(x)$. Whereas, in multi-objective optimization, there does not typically exist a feasible solution that optimizes all objective functions simultaneously as the objectives are conflicting and cannot be optimized simultaneously.

When the objectives $k \geq 2$, $f(x)$ is only partially ordered. For example, in the computer design engineering, the goal is to get the most powerful computer ($f_1(x)$) with lowest cost ($f_2(x)$). The objectives are conflicting to each other, therefore, depending on the decision marker's

requirements, an intermediate solution (Pareto front) might be an appropriate trade-off between the two conflicting objectives, as illustrated in listing below.

Definition 1: Multi-Objective Optimization

x: decision vector

y: objective vector

X: decision space

Y: objective space

***Objectives**:*

Max|Min: $y = f(x) = (f_1(x), f_2(x), \ldots, f_k(x))$

Constraints/subject to: $s(x) = (s_1(x), s_2(x), \ldots, s_m(x)) \leq 0$

where $k \geq 2$

Definition 2: Feasible Set

Feasible set X_f is defined as the set of decision vectors x that satisfy the constraints s(x).

$X_f = \{ x \in X \mid s(x) <= 0 \}$

Definition 3: Feasible Region

Feasible Region Y_f is the region in the objective space

$Y_f = f(X_f) = \cup_{x \in Xf} \{ f(x) \}$

1.3 Decision Making Processes with Multi-Objective Optimization

There are two processes required to solve real-world multi-objective optimization problems.

- Search Process

  This process is to search the Pareto-optimal solutions. Real-world problems often have large and complex search space, and the requirements for fast outputs for decision makers. Result visualization on the Pareto-optimal sets is an important part at the end of the search process because decision makers need to understand result sets and visualize the impacts of choosing different solutions.

- Decision Making Process

  Decision making process is based on past experiences, judgment and intuition and can be error prone. The human mind is not capable of perceiving in all details more than seven parameters, on an average, at a time [1,149]. In scientific decision making is to take the trade-offs among all the possible optimal solutions. The application of genetic algorithm to search the Pareto-optimal set is, therefore, becoming more useful. The decision maker has to deal with vast data, number of alternatives and different decision situations before taking any decision. The issue becomes taking trade-offs in conflicting interests. Consequently, one of the most important and difficult aspects of any decision problem is to achieve an equilibrium among multiple and conflicting objectives [1]. With the help of generated set of Pareto-optimal solutions, human decision makers make the trade-offs between conflicting objectives and choose the

final solutions, in combination with their knowledge, experiences, preferences and expertise.

Prior to the start of search process, decision makers set up the objectives of the targeted problems and/or initial conditions, including their preferences info, initial parameters, stopping constraints and thresholds. This usually required profound domain knowledge as the search process is performed with the objectives given. Solving a multi-objective optimization problem is sometimes understood as approximating or computing all or a representative set of Pareto optimal solutions. When decision making is emphasized, the objective of solving a multi-objective optimization problem is referred to supporting decision makers in finding the most preferred Pareto optimal solution according to their subjective preferences [1,85,90,92].

In this research, the focus is to design and implement a framework for MOOP that are capable of handling the real-world MOOP with large and highly complex search spaces during the search process, with the visualization of Pareto-optimal sets for the decision-making process.

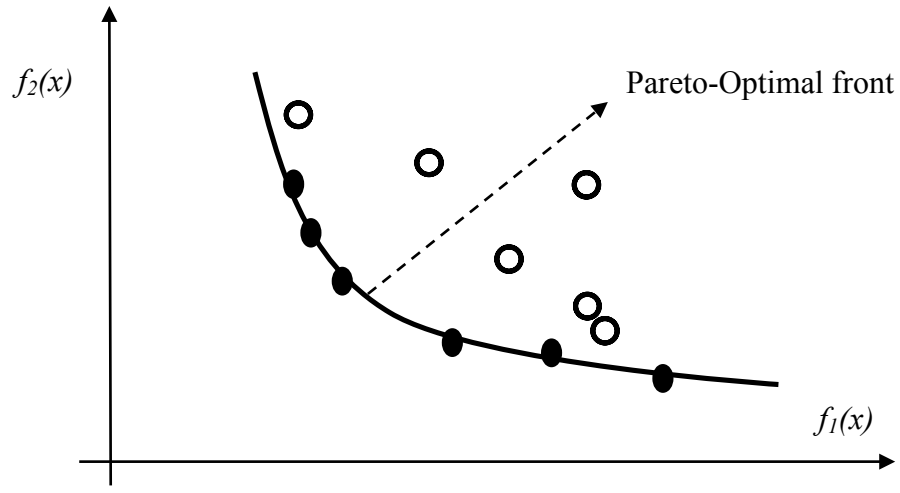


Figure 1.1 Pareto optimality

1.4 Multi-Objective Optimization Methods and Algorithms

1.4.1 Classical Methods

Over the past four decades, there are large amount of research and applications of using classical methods (non-GA based method) on multi-objective optimization problem. They can be classified into the following classes:

- Non-preference methods

  These methods do not assume any information about the importance of objectives, a heuristic is used to find a single optimum solution, without consideration of multiple Pareto-optimal solutions.

- Posteriori methods

  Posteriori methods use preference information of each objective and iteratively generate a set of Pareto-optimal solutions.

- A priori methods

  Priori methods use more information about the preferences of objectives and they usually find one preferred Pareto-optimal solution.

- Interactive methods

  These methods use the preference information progressively during the optimization process.

Some representatives of classical methods are discussed in section 2.1. The limitations of each method are also discussed in detail in this section.

1.4.2 Genetic Algorithms

Genetic algorithm is the one of most commonly used methods to solve multi-objectives optimization problems. Genetic algorithms are search and optimization procedures that are inspired by the process of natural selection which is the differential survival and reproduction of individuals through evolution. Charles Darwin used the term "natural selection", and compared it with artificial selection. Natural selection is the differential survival and reproduction of individuals due to differences in phenotype. It is a key mechanism of evolution, the change in heritable traits of a population over time. Because random mutations arise in the genome of an individual organism, and offspring can inherit such mutations, variation exists within all populations of organisms. Individuals with certain variants of the trait may survive and reproduce more than individuals with other, less successful, variants. Therefore, the population evolves over generations [90,91,93].

The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. Individuals that gives a reproductive advantage

may become more common in a population. Over time, this process can result in populations that specialize for particular ecological niches (microevolution) and may eventually result in speciation (the emergence of new species, macroevolution). Natural selection is a key process in the evolution of a population [71-74].

In comparison with classical optimization techniques, genetic algorithm is different, as the objectives are often multiple and conflicting. With these constraints and objectives, the design problem is nonlinear programming problem (NLP) [75,76,82].

In a genetic algorithm, a population of candidate solutions, also called as individuals, creatures, or phenotypes, are evolved toward better solutions in optimization search space. Each candidate solution has a set of properties (its chromosomes or genotype). The solutions can be mutated and altered after each generation; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible [96,97]. The processes of evolution are used to construct genetic algorithms with genetic operations requires a genetic representation of the solution domain, a fitness function to evaluate the solution domain and a standard representation of each candidate solution is as an array of bits [98]. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming; a mix of both linear chromosomes and trees is explored in gene expression programming [119,112,114].

Genetic algorithm (GAs) is a method for solving optimization problems based on a natural selection process that mimics biological evolution. Genetic algorithms (GAs) are often well-suited for optimization problems involving several, often conflicting objectives. With pre-defined set of iterations and goals, the algorithm continuously and repeatedly generates a population of better individual solutions. At each iteration, the genetic algorithm randomly selects individuals from the current population within the predefined goal settings and uses them as parents to produce the children for the next generation, in other words, it generates a population of points at each iteration and selects the next population by computation which uses random number generators. Over successive generations, the population "evolves" toward an optimal solution. In comparison, the classical, derivative-based, optimization algorithm generates a single point at each iteration and selects the next point in the sequence by a deterministic computation.

1.5 Applications of Multi-Objective Genetic Algorithm

Multi-objective optimization has been applied in many fields of science, including engineering, economics and logistics, where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. For example, in the banking industry, a common problem for investors is to choose a portfolio when there are two conflicting objectives: highest investment returns with lowest risks. This problem shows the best combinations of risk and expected return that are available, and in which indifference curves show the investor's preferences for various risk-expected return combinations. In resource management, including power, cell network, water etc., every user's objective is to have sufficient utilization of the resources but with minimal cost, while resources themselves are limited and often difficult to

make up to demand, and another objective is to make them environment friendly while minimizing the cost of making the resources available. These objectives are conflicting to each other. The decision maker would need to find a Pareto optimal solution that balance the total resource supply and demand. In engineering and economics, many problems involve multiple objectives which are not describable as the-more-the-better or the-less-the-better; instead, there is an ideal target value for each objective, and the desire is to get as close as possible to the desired value of each objective. For example, computer design problem typically involves taking trade-offs between performance and cost [105,106]. In economics, government might want to conduct open market operations so that both the inflation rate and the unemployment rate are as close as possible to their desired values.

Genetic algorithm are well suited to solve multi-objective optimization problems in comparison with traditional algorithms. Traditional search and optimization methods often have difficulties solving the nonlinearities and complex interactions among problem variables in real world situation, especially when the search space has more than one optimal solution. [87,88, 89]. Genetic algorithm can be applied to solve problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear as GAs do not require derivative information or use gradient information in its search process. They use direct search procedures, hence allowing them to be applied to a wide variety of optimization problems. Additionally, in each iteration, there are multiple parallel solutions encoded in a population, hence in combination with the use parallel programming and modern multi-processor computers, can do a computationally quick overall search. The crossover operator of GA may exploit structures of good solutions with

respect to different objectives to create new non-dominated solutions in unexplored parts of the Pareto front. In addition, most multi-objective GA do not require the user to prioritize, scale, or weigh objectives. Therefore, GA have been the most popular heuristic approach to multi-objective design and optimization problems. Jones et al. [138] reported that 90% of the approaches to multi-objective optimization aimed to approximate the true Pareto front for the underlying problem. A majority of these used a meta-heuristic technique, and 70% of all meta-heuristics approaches were based on evolutionary approaches. Alternative and complementary algorithms include evolution strategies, evolutionary programming, simulated annealing, Gaussian adaptation, hill climbing, and swarm intelligence (e.g.: ant colony optimization, particle swarm optimization) and methods based on integer linear programming. The suitability of genetic algorithms is dependent on the amount of knowledge of the problem.

There is a number of research on genetic algorithm applications in multi-objective optimization with the goal to find a representative set of Pareto optimal solutions, and/or quantify the trade-offs in satisfying the different objectives, and/or finding a single solution that satisfies the subjective preferences of a human decision maker (DM), as illustrated in conference proceedings and domain-specific books, journals and proceedings [131,132, 135,136 *et al*]. Multi-Objective Genetic Algorithms (MOGA) have the following advantages in comparison with the traditional algorithms. MOGA can obtain a set of non-dominated solutions opposed to a single solution and it has flexibility in handling a wide range of types of variables, objective functions, and constraints, such as nonlinear, discontinuous as discussed in Section 2.12.

Multi-Objective Genetic Algorithm are also applied in the knowledge discovery and data mining research. Data mining methods are designed for extracting previously unknown

significant relationships and regularities out of huge heaps of details in large data collections [20,21,22, 45]. If MOGA procedure can find solutions close to the true Pareto-optimal set, the solutions can be further analyzed for properties which are common. Such a systematic approach can be used in deciphering important and hidden properties. The finding of multiple trade-off and optimal solutions using a multi-objective optimization, and then analyzing the solutions to discover useful knowledge can be part of knowledge discovery process [130].

Data clustering can be used in MOGA for fast convergence to a global optimal solution as the size of populations can be reduced though clustering. Clustering is the task of grouping a set of objects in such a way that objects in the same group are more to each other than to those in other groups (clusters). Clustering analyses requires multiple objectives to be optimized. The objectives of clustering are to first have clear separation of clusters in data sets, which can lead to a larger number of clusters. On the other hand, another objective of clustering is to have smaller groups of quality clusters. These objectives are conflicting to each other and therefore, it is naturally a multi-objective optimization problem. Without any previous knowledge about the data, it is hard to decide on the number of clusters, and there are always some trade-offs between the quality of a clustering result and the number of clusters. Cluster analysis is an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure, in combination with the use of genetic algorithm with generated data set and model parameters, can achieves the optimal result set [6, 130].

1.6 An Innovative Hybrid Unified Framework

Multi-objective optimization is important in real-world practical problem solving, but not much attention has been paid so far in this respect among the GA research [209,210]. In real

world situations, multi-objective optimization problems become more challenging as there is no one algorithm that can be used to fit all problems and be applicable to all situations that can generate the Pareto-set in a timely and efficient manner for decision makers. Multi-objective evolutionary algorithms usually have following challenges, which include

- Missing global optimum
- Slow converge to global optimum
- $O(MN^3)$ computational complexity (where M is the number of objectives and N is the population size) in the search process.
- Lack of elitism or slow elitism selection approach in the search process.
- Lack of efficient stopping criterion in the search process.
- Lack of result visualization in the search process for decision makers.
- Lack of integration with human decision makers in the decision-making process.

This research proposed an innovative unified and comprehensive multi-objective genetic algorithm to try to address the aforementioned challenges. The generated Pareto-optimal solution sets can give researcher and decision makers the best overview of the problems at hand and recommended solutions with expected impacts.

The main contribution of this research is a novel genetic algorithm and a unified framework that provides a fast, comprehensive and general-purpose approach to solve real-world multi-objective optimization problems. This framework can also be applied to date clustering in data mining and knowledge discovery as illustrated in the experiments. The proposed research has the following contributions.

1. Better convergence performance with a fast search algorithm and K-mean operators computing through the use of modern advancement of computational technology, including the parallel and asynchronous programming with $O(MN^2)$ computational complexity.
2. A set of hybrid genetic algorithm with local search algorithm and K-mean operator that can create a global optimization population by combining the parent and offspring populations and selecting the best N solutions (with respect to fitness and spread).
3. A framework that can find a better spread of solutions and better convergence near the true Pareto-optimal front compared to other researches on Pareto evolutionary algorithm.
4. An innovative and unified framework that can be applied directly in helping to solve real world multi-objective optimization problems.

The applicability and effectiveness of the described framework were verified by clustering validity analysis with extensive testing and experimental datasets from a variety of domains ranging from very general to very specific like gene expression data.

This framework is then applied to solve two real-world multi-objective problems. The first application was on the blood bank utilization optimization for Calgary Health Region, Alberta, Canada. It provides a recommended set of actions to optimize the blood bank inventory. The second application was on shopping optimization application for Microsoft Canada Imagine Cup 2013 competition. It provides a recommend set of optimal routes for shopper to find the most cost-effective way of shopping.

The structure of this thesis is organized as follows. Chapter Two is an overview of related works in multi-objective genetic algorithm research. Chapter Three is devoted to the methodology, algorithm, setup and development of the entire framework. Chapter Four reports the actual results on experimental datasets to test the applicability, accuracy, performance, and efficiency of the framework, including the experimental testing on benchmark datasets, and the solution sets for two real world situations. Chapter Five concludes the outcome of this research, including the discussion of the strengths and limitations of this research and the developed framework with future possible enhancements with the rapid advancement of the modern computing technologies.

Chapter Two: RELATED WORK

2.1 Traditional Methods in Multi-Objective Optimization

Because of the computational complexity in the search process of multi-objective optimization solutions, traditional methods aggregate the objectives into a single, parameterized objective function by analogy for generating the optimal solution set to decision making. The input parameters of this function are set by different optimization runs with different parameter settings. The solution generated by each run are groups at the end to achieve a set of solutions which approximates the Pareto-optimal set. Three representatives of traditional techniques are summarized below:

2.1.1 Goal Programming

Goal programming is a branch of multi-criteria decision analysis (MCDA). A criterion is a single measure by which the goodness of any solution to a decision problem can be measured. Depending on the fields of application, criteria can be cost, profit, time, distance, or performance of a system. A decision problem which has more than one criterion It is a generalization of linear programming to handle multiple, normally conflicting objective measures. For each of the objective measure, a goal or target value is set for the search is given, negative deviations from this set of target values are then minimized in an achievement function. Charnes and Cooper [1977] presented the general goal programming model as below:

Objectives

Minimize: $f(x) = \sum_{i=1}^{m}(d_i^+ + d_i^-)$

Where*:*

$f(x)$ = *objective function = Summation of all deviations*

d_i^- = *negative deviational variable from the* i^{th} *goal (underachievement)*

d_i^+ = *positive deviational variable from the* i^{th} *goal (overachievement).*

Variable	Objective	Condition
d_i^-	Minimize	$d_i^- = 0$
d_i^+	Minimize	$d_i^+ = 0$
$d_i^- + d_i^+$	Minimize of the Total	$d_i^- = 0, d_i^+ = 0$

Table 2.1 Generalized Goal Programming Model

2.1.2 Constraint Method

Constraint method in general can be described as below:

Choose one of the objectives to be optimized, give other objectives an upper bound and consider them as constraints. Different PO solutions can be obtained by changing the bounds and/or the objective to be optimized.

Maximize; $f(x) = f_h(x)$

Subject to: $e_i(x) = f_i(x) > \varepsilon_i \ (1 \leq i \leq k, i \neq h)$

$x \in X_f$

As shown in Figure 2.1, the constraint method is able to obtain solutions associated with non-convex parts of the trade-off curve. Setting $h=1$ and $\varepsilon_2 = r$ (solid line) makes the solution represented by an infeasible regarding the extended constraint set, while the decision vector related to *B* maximizes *f(x)* among the remaining solutions

The problem is that the solution to the problem largely depends on the selection of the ε vector. In particular, it must be chosen such that it lies between the minimum and maximum value of each objective function. As the number of objectives increases, the complexity increases exponentially as well [135].

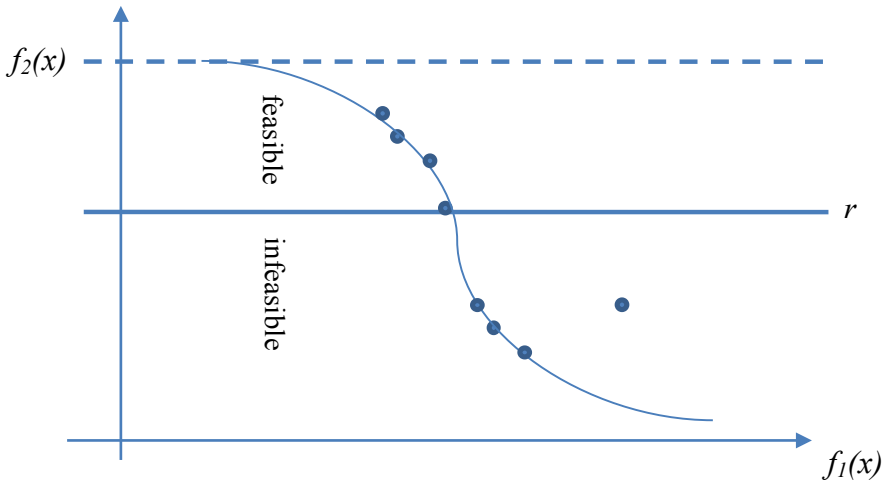


Figure 2.1 Constraint Methods in Multi-Objective Optimization

2.1.3 Weighted Sum Method

Aggregation methods is one of the classical methods that combine the objectives into a higher scalar function which is used for fitness calculation; It can produce one single solution and require profound domain knowledge from subject matter expert, which is often not available.

This method secularizes the set of objectives into a single objective by multiplying each objective with a user supplied weight. The value of the weights is based on the relative importance of each objective. The mathematical model is described below:

Maximize: $\mathrm{f(x)} = w_1 f_1(x) + w_2 f_2(x) + \cdots + w_k f_k(x)$

$Subject\ to: x \in X_f$

The weights w_i are normalized such that $\sum w_i = 1$. Different weight combination will generate set of solutions. As shown in Figure 2.2, in case of convex problems, the entire Pareto-optimal set can be found. However, for multiple mixed objective optimization problems (min-max), all the objectives need to be converted into one type, and it may not be able to find a uniformly distributed set of Pareto-optimal solutions as two different set of weight vectors not necessarily lead to two different Pareto-optimal solutions [38].

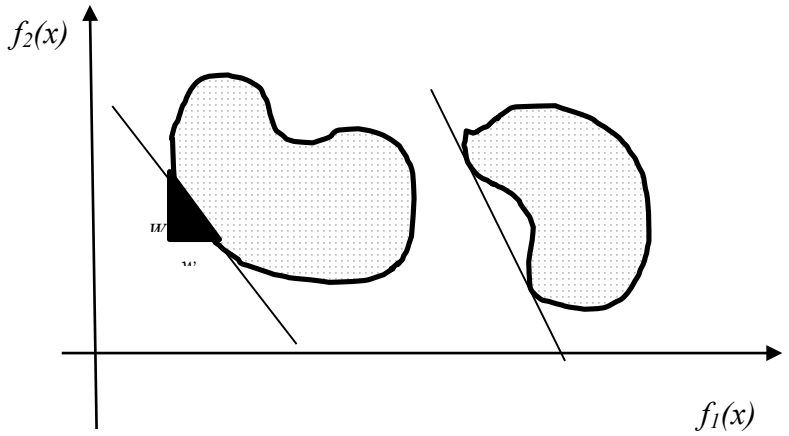


Figure 2.2 Weighted Method with Convex and Non-Convex

2.1.4 Limitations of Classical Methods

Classical methods are attractive and popular in many single objective and simple multi-objective optimization applications. However, for real world large-scale complex problems, with high dimension and modalities and lack of prior knowledge on the intrinsic of the problems, classical methods have difficulties with convergence and generate Pareto-optimal solutions. As shown in Figure 2.2, the weighted sum method may be sensitive to the shape of the Pareto-optimal front. Therefore, their application of using classical methods to complex multi-objective optimization problems is limited and restricted. Moreover, classical methods require several optimizations runs to obtain an approximation of the Pareto-optimal set, which will result in high computation overhead.

Evolutionary algorithms can overcome the aforementioned difficulties and have become established as an alternative to classical methods to solve multi-objective optimization problems.

EAs can handle large search spaces, and generate Pareto-optimal fronts with multiple alternative trade-offs a single optimization run.

2.2 Issues with Classical Methods on Multi-Objective Optimization

Classical approaches to solve optimization problems can be classified into two distinct groups: direct and gradient-based methods [83, 87, 88, 89]. In direct search methods, only objective function $O(x)$ and constraint values are used to guide the search strategy, and derivative information are not used, this usually requires more function evaluations for convergence, and another issue with this approach is that it requires changes of algorithms for different objectives.

Gradient-based method can converge to an optimal solution on linear cases but with issues in non-differentiable or discontinuous problems [83]. In addition, there are some common difficulties with most of the traditional direct and gradient-based techniques are summarized below:

- Local or sub optimal solution
- Requires specific algorithm for different optimization problem.
- Not applicable to non-linear, discrete variables
- Not efficient for parallel computing.
- Optimal solution depends on the chosen initial solution

2.3 History of Genetic Algorithms (GA)

As discussed in the previous section, genetic algorithms (GAs) can overcome the limitations of classical algorithms and have become established method for exploring the Pareto-optimal

front in multi-objective optimization problems that are too complex to be solved by classical methods, such as linear programming and gradient search.

Genetic algorithm (GA) is a metaheuristic inspired by the process of natural selection, also known as the survival of the fittest. Genetic algorithms are also referred as evolutionary algorithms (EA), or as a subset of evolutionary computation [81,82]. GA has inherent parallelism and can approximate the Pareto-optimal front in a single optimization run by crossover/recombination, mutation and selection operators in a single simulation run. Since genetic algorithms work with a population of solutions, a simple GA can be extended to maintain a diverse set of solutions with an emphasis for moving toward the true Pareto-optimal region. The goals for multi-objective optimization is to find the true Pareto-optimal sets, best uniform distribution of the solutions, and maximum spread of the obtained non-dominated front.

Genetic algorithms (GAs) have been extensively used as search and optimization tools in various problem domains, including sciences, commerce, and engineering since John Holland [86] first introduced the concept of a genetic algorithm in 1975. Over the past decade, a number of multi-objective evolutionary algorithms (MOEAs) have been suggested with a number of research papers on genetic algorithm published. A more comprehensive description of genetic algorithms can be found in the recently compiled Handbook on Evolutionary Computation, published by Oxford University Press [*Back et al 1997*] [71]. Two journals entitled Evolutionary Computation (published by MIT Press) and IEEE Transactions on Evolutionary Computation are now dedicated to publishing salient research and application activities in the area. Some of the well-cited evolutionary algorithms used in multi-objective optimization (MOGA) were reviewed in the literature reviews section.

2.4 Design of Genetic Algorithms (GA)

Genetic algorithm starts with an initial set of candidate solution and iteratively update the result sets through reproduction, mutation, recombination, and selection. Each new generation is produced by stochastically removing less desired solutions, and introducing small random changes. In biological terminology, a population of solutions is subjected to natural selection (or artificial selection) based on fitness, crossover, and mutation. As a result, the population will gradually evolve to increase in fitness and improve [76].

The solution candidates are called as individuals and the set of solution candidates is called the population. Each individual represents a possible solution, which usually encoded as bit vector or real-value vector, or other structures like trees [95]. The set of all possible solution vectors constitutes the individual space $\boldsymbol{I}$. The population is a multi-set of vectors $\boldsymbol{i} \in \boldsymbol{I}$.

In general, a GA is characterized by Table 2.1 below:

Input:	***P: population size*** ***G: maximum number of generations*** ***P_r: crossover probability*** ***R_m: mutation rate***
Output	***$P_{Set:}$ Pareto non-dominated set***

Table 2.2 Inputs and Outputs of Generalized Genetic Algorithm

Genetic Algorithm Steps:

Step 1: ***Initialization***: *An initial population is generated randomly, allowing the entire range of possible solutions, i.e., the search space. Randomization of population is essentially a stochastic process. The diversity of population is an important factor to reach global optimality. Convergence in optimization process is crucial because it specifies termination condition of the process [50].*

Set $P_0 = \emptyset, t = 0$

For $i = 1, \ldots, N$ *do*

a) Select $i \in I$ *based on* p_r

b) Set $P_0 = P_0 + \{i\}$

where I is the individual space. P_0 *can be "seeded" in areas where optimal solutions are likely to be found.*

Step 2: ***Fitness setup***: *the fitness is the value of the objective function f(x). The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation.*

For each individual $i \in P_t$

1. *determine the encoded decision vector* $x = m(i)$
2. *determine the objective vector* $y = f(x)$
3. *calculate the scalar fitness value F(i)*

Step 3: ***Selection***: *Based on the fitness value F(i), individual solutions are selected. Some selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as the former process may be very time-consuming.*

Set $P' = \emptyset$

For i= 1,…, N do

1. *Select* $i \in Pt$ *based on F(i)*
2. *Set* $P' = P' + \{i\}$

where P' is the temporary solution

Step 4: ***Crossover / Recombination***: *Within the selected solutions, the crossover operator is usually applied with a crossover probability (*$\boldsymbol{p}_c \in [0, 1]$*), which means the proportion of population members participating in the crossover operation. The remaining (1- pc) proportion of the population is copied to the offspring population.*

Set $P'' = \emptyset$

For i= 1,…, N/2 do

1. *Select two individuals* $i, j \in P'$
2. *Remove i, j from P'*
3. *Crossover i,j into* $m,n \in I$
4. *Use Pr to add m,n to P'', or re-use i,j*

where P'' is the temporary solution

Step 5: ***Mutation***: *Within the population itself, mutation happens with a probability that an arbitrary bit in a genetic sequence will be changed from its original state. After the crossover operator, the individual is then perturbed in its vicinity by a mutation operator. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable decides whether or not a particular bit will be modified. Every variable is mutated with a mutation probability* $\boldsymbol{p_m}$ *, usually set as* $1/\boldsymbol{n}$ *(****n*** *is the number of variables), so that on an average one variable gets mutated per solution. In the context of real-parameter optimization, a simple Gaussian probability distribution can be used with its mean at the child variable value.*

Set $P''' = \emptyset$

For each individual $i \in P''$ *do*

1. *Mutate i based on* $R_m \Rightarrow j \in I$
2. *Set* $P''' = P''' + \{ i \}$

Step 6: ***Termination***: *The evolutionary process will converge and stop when it reaches the predefined termination conditions. Common terminating conditions are:*

- *Satisfactory criteria reached*
- *Predefined number of generations reached*
- *Allocated resources (computation time/money) reached*
- *The highest ranked solution's fitness is reaching or has reached to a point such that successive iterations no longer produce better results*
- *Manual intervention*

Set $P_{(t+1)} = P'''$

$t = t+1$

If (t < T or any of above stopping criterion is satisfied) then

set A = p(m(Pt)

else

go to Step 2.

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously, and the pair are used for producing offspring solutions with crossover and mutation operators, new offspring solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated for next generation. Some researchers [86] [87] suggests that more than two "parents" solutions selected for crossover and mutation processes can generate higher quality chromosomes. The selection of better solutions from every generation is also known as elitism.

The elitism operator combines the old population with the newly created population and chooses to keep better solutions from the combined population. Such an operation makes sure that an algorithm has a monotonically non-degrading performance.

Each step in genetic algorithm described above can vary in design, initial settings and configurations with restrictions, for example, the population size can have different limits; crossover can involve more than two parents, selection processed can be based on probability or tournament selection. Moreover, a large number of variations in selection, crossover, and mutation operators have been proposed for different representations.

2.5 Review of Existing Multi-Objective Genetic Algorithms

Fonseca and Fleming [111] conducted a comprehensive overview of multi-objective genetic algorithms. Based on the simulation of evolutionary approaches, genetic algorithms are categorized into aggregating approaches, population-based non-Pareto approaches and Pareto-based approaches; moreover, approaches using niche induction techniques were also reviewed. There have been a number of more researches on multi-objective genetic algorithms (MOEAs) since then. Five of the most salient MOEAs have been chosen for the comparative studies reviewed in the next chapter. A summary of their main features and their differences is described as well. The thorough discussion of different evolutionary approaches to multi-objective optimization are not discussed in this paper.

2.5.1 VEGA - Vector Evaluated Genetic Algorithm

Schaffer [84] proposed an multi-objective optimization using genetic algorithm, called vector evaluated genetic algorithm (VEGA), which is a representative of the category selection by switching objectives. VEGA is the first genetic algorithm to approximate the Pareto-optimal set by a set of non-dominated solutions. In VEGA, population P_t is randomly divided into K equal sized sub-populations; $P_1, P_2, \ldots, P_K$. Then, each solution in subpopulation P_i is assigned a fitness value based on objective function Z_i. Solutions are selected from these subpopulations using proportional selection for crossover and mutation [84].

VEGA Description:

Let *N = Population Size*

K = Number of Objectives

Ns = sub-population size (Ns = N/K)

Step 1: *Initialization - Choose a random population P_t with t=0*

Step 2: *Check stopping condition: if yes, return P_t*

Else goto Step 3

Step 3: *Sort - Randomly sort population Pt*

Step 4: *Fitness Assignment - For each objective K = 1,..., k*

Step 4.1. *for i = I + (k-1)Ns, ..., kNs. Assign fitness value $f(x_i) = Z_k(x_i)$ to the i_{th} solution in the sorted population.*

Step 4.2 *Based on the fitness values assigned in Step 4.1, Select N_s solutions between the (1 + (k-1)th and (kNs)th solutions of the sorted population to create a new sub-population P_k.*

Step 5*: Combine all subpopulations P_1, ..., P_k and apply crossover and mutation operator on the combined population to create P_{t+1} of size N. Set t= t+1*

Step 6*: Combine all subpopulations P1,y,Pk and apply crossover and mutation on the combined population to create Pt+1 of size N. Set t ¼ t þ 1, go to Step 2.*

In VEGA, each subpopulation is evaluated with respect to a different objective. Fitness value and comparison is executed for each of the k objectives separately. The population of mating pool have equal size for crossover operator. Steps 2 and 3 of this algorithm are executed *k* times per generation, respectively replaced by the following algorithm [84]:

Input: P_t (population)

Output: P' (mating pool)

Step 1: *Set i =1 and mating pool P'=0*

Step 2: *For each individual i* $\in P_t$, *do* $F(i) = f_i(m(i))$

Step 3: *for j=1,……,N*

Select individual I from P_t *according to a given scheme and copy it to the mating pool*

P' = P' + {i}

Step 4: *Set i = i +1*

Step 5: *If i <=k,*

then go to Step2

Else

STOP

As shown in Fig. 2.3, the best individuals in each dimension are chosen for reproduction. Afterwards, the mating pool is shuffled and crossover and mutation are performed as usual. Schaffer implemented this method in combination with fitness proportionate selection [84], it is straightforward implementation, but tends to converge to the extreme of each objective.

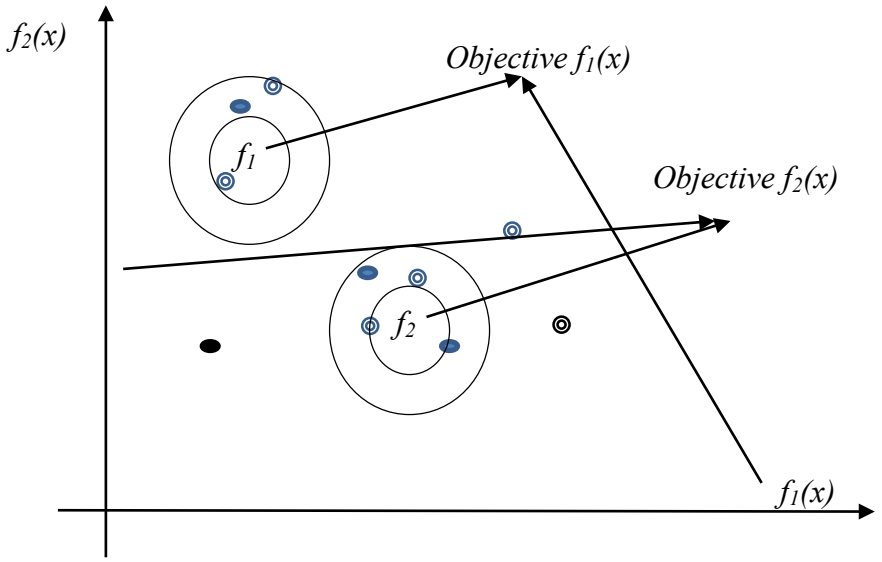


Figure 2.3 VEGA Illustration

Since VEGA, there have been a number of enhanced genetic algorithms developed [80] based on VEGA's design principal, and VEGA has been used as a strong reference in many related literatures.

2.5.2 NPGA Genetic Algorithm

The Niched Pareto Genetic Algorithm (NPGA) [110] is another well-known genetic algorithm (GA) to deal with multiple objectives optimization problems. It incorporates the concept of Pareto domination in the selection operator, and applying a niching pressure to spread its population out along the Pareto optimal tradeoff surface. The design of NPGA is summarized below.

Input: $P_{t:}$ *Population*

σ_{share}: $Niche\ radius$

$t_{dom:}$ *domination*

Output: *P' (mating pool)*

Step 1: *Set i =1 and mating pool P'=0*

Step 2: *Randomly choose 2 individuals x, y* $\in P_t$ *in a set* $P_{dom} \subseteq P_t$

Step 3: *If I(x) dominates in* P_{dom}*, and I(y) doesn't, then I(x) is the dominating individual (winner) of the tournament. P' = P' + I(x)*

If I(y) dominates in P_{dom}*, and I(y) doesn't, then I(y) is the dominating individual (winner) of the tournament. P' = P' + I(y)*

Else

Choose the winner by fitness sharing in Step 4

Step 4: *Tournament by fitness sharing*

Find the individuals that are within the σ_{share} for I(x) and I(y)

D(x|y) < σ_{share}

If D(x) < D(y), then P' = P'+I(x), else P' = P'+ I(y)

Step 4: *Set i = i+1. If i < Population size, then go to step2, else terminate.*

In NPGA, the fitness assignment can be either value based, or by tournament selection in selection operator in the objective space. Binary Pareto tournaments is illustrated in Figure below.

Two competing individuals and a set of t_{dom} individuals are compared. The competitor represented by the white point is the winner of the tournament since the encoded decision vector is not dominated with regard to the comparison set in contrast to the other competitor.

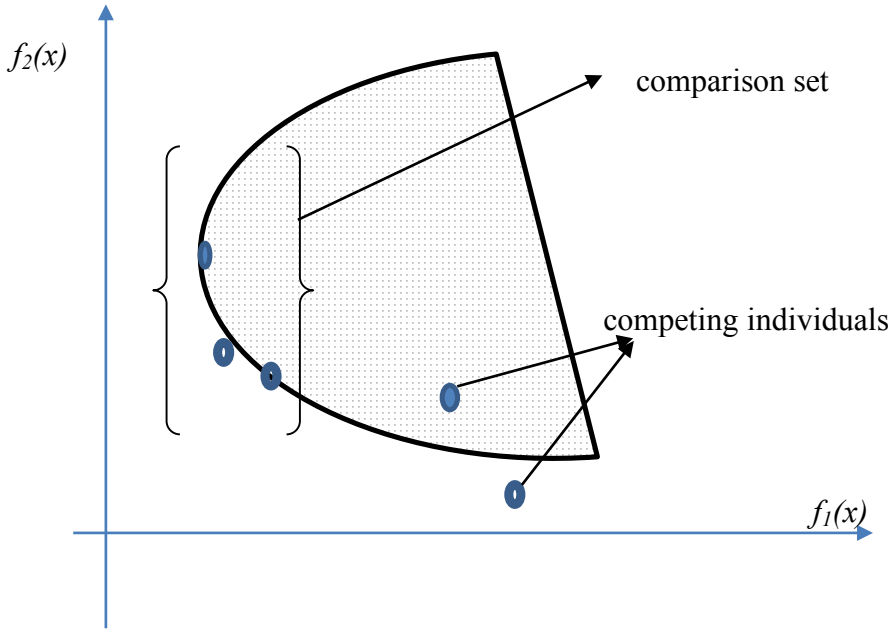


Figure 2.4 NPGA Illustration

2.5.3 NSGA Genetic Algorithm

Goldberg [79] proposed the first Pareto ranking technique. Pareto-ranking approaches explicitly utilize the concept of Pareto dominance in evaluating fitness or assigning selection probability to individuals. Individual population ranking and selection are based on a dominance rule, and then each solution is assigned a fitness value based on its rank in the population, not its actual objective function value. NSGA is described below:

Step 1: *Create a random parent population P_0 of size N. Set t=0*

Step 2: *Apply crossover and mutation to P_0 to create offspring population Q_0 of size N*

Step 3: *If the stopping criterion is satisfied, stop and return to Pt*

Step 4: *Set $Rt = Pt \cup Qt$*

Step 5: *Using the fast non-dominated sorting algorithm, identify the non-dominated fronts*

$F_1, F_2, \ldots F_k$ in Rt

Step 6: For I =1, …, k do the following steps:

Step 6.1. *Calculate crowding distance of the solutions in Fi*

Step 6.2 *Create P_{t+1} as follows:*

Case 1: if $|P_{t+1}| + |F_i| <= N$, then set $P_{t+1} = P_{t+1} \cup F_j$

Case 2: if $|P_{t+1}| + |F_i| > N$, then add the lest crowded $N - |P_{t+1}|$ solutions

from F_j to P_{t+1}

Step 7: *Use binary tournament selection based on the crowding distance to select parents from P_{t+1}. Apply crossover and mutation to P_{t+1} to create offspring population Q_{t+1} of size N.*

Step 8: *Set t =t+1 and go to Step 3.*

Goldberg's ranking technique [98] is described below:

Step 1: *Set I =1 and TP =P*

Step 2: *Identify non-dominated solutions in TP and assigned them to F_j*

Step 3: *Set TP = TPF_j If TP = 0, goto step 4, else set i = i+1 and goto step 2.*

Step 4: *For every solution $x \in P$ at generation t, assign rank r1(x,t) = I if $x \in F_j$*

where: *Fi are non-dominated fronts,*

F_1 is the Pareto front of population P in step 1.

In NSGA, only non-dominated solutions participate in the crossover and selection operator when the combined parents and offspring population includes N non-dominated solutions. There are a number of the Pareto-based MOEAs developed since then, Srinivas and Deb[112] have used different trade-off fronts in the population and fitness sharing is performed for each front separately in order to maintain diversity. Fitness assignment is shown below:

Input: P_t *(population)*

$\sigma_{share}(niche\ radius)$

Output: *F (fitness values)*

Step 1: *Set* $P_{remain} = P_t$ *and initialize the dummy fitness value* F_d *with N*

Step 2: *Determine set* P_{nondom} *of individuals in* P_{remain} *whose decision vectors are nondominated regarding m*(P_{remain}). *Ignore them in the futher classification process, i.e.,* $P_{remain} = P_{remain} - P_{nondom}$ *(Multiset subtraction).*

Step 3: *Set raw fitness of individuals in* P_{nondom} *to* F_d *and preform fitness sharing in decision space, only within* P_{nondom}.

Step 4: *Decrease the dummy fitness values* F_d *such that it is lower than the smallest fitness in* P_{nondom} *:* $0<F_d<min\ \{F(i)|i \in P_{nondom}\}$

Step 5: If $P_{remain} \neq 0$ *then goto Step 2*

Else STOP

2.5.4 FFGA Genetic Algorithm

Fonseca and Fleming proposed a Pareto-based ranking procedure [111]. In this algorithm, the fitness assignment procedure is different from the aforementioned genetic algorithms, and an individual's rank equals the number of solutions encoded in the population by which its corresponding decision vector is dominated.

Input: P_t *- population*

$\sigma_{share} -$ *Niche radius*

Output: *F – fitness values*

Step 1: *Calculate rank R(i) of every individual i*

$R(i) = 1 + | x | x \in P_t \wedge (x>i) |$

Step 2: *Based R, Sort the current population and assign corresponding sorted fitness value F'(i) for every individual.*

Step 3: *Calculate the fitness value F(i) by averaging and sharing F'(i) in objective space.*

As illustrated in Figure 2.5, based on the sorted individuals, the ranking values are assigned from rank high (1) to low (10) accordingly. The crossover populations are implemented using stochastic universal sampling based on the ranks. FFGA use the idea of fitness sharing which was proposed by Goldberg and Richardson [79] in the search of Pareto front on multiple local optima for multi-modal functions. FFGA applies some penalties to fitness of solutions in densely populated areas in order to find the undiscovered Pareto front and to maintain diversity of population. A "niche penalty" is any group of individuals of sufficient similarity (niche radius) have a penalty added, which will reduce the representation of that group in subsequent generations [110,124]. It identifies the densely populated areas and applies penalty the solution located in such areas as shown below.

Step 1: *Calculate the Euclidean Distance D(x, y) between every pair of individuals x and y in the normalized objective space between 0 and 1.*

$$D(x,y) = \sqrt{\sum_{k=1}^{k} \left(\frac{z_k(\mathrm{x}) - z_k(y)}{z_k^{max} - z_k^{min}}\right)^2}$$

where z_k^{max} and z_k^{min} are the maximum and minimum value of the objective function $Z_k(x,y)$

Step 2: *Based on the distances, calculate the niche count for each solution $x \in P$ as*

$$Niche\ Count = \sum_{x \in p} Max\left\{\frac{\sigma_{share-D(x,y)}}{\sigma_{share}}\right\}$$

where σ_{share} is the niche size.

Step 3: *Assign the fitness of each solution* $F' = \frac{F}{Niche\ Count}$

The niche count and fitness sharing based on niche count requires computational effort because Fitness sharing requires a new parameter be selected and niche count is also an expensive calculation [124]. Some researches [79,124] proposed methods of dynamically updating the parameter σ_{share} and dynamically niche sharing to increase effectiveness of computing niche counts.

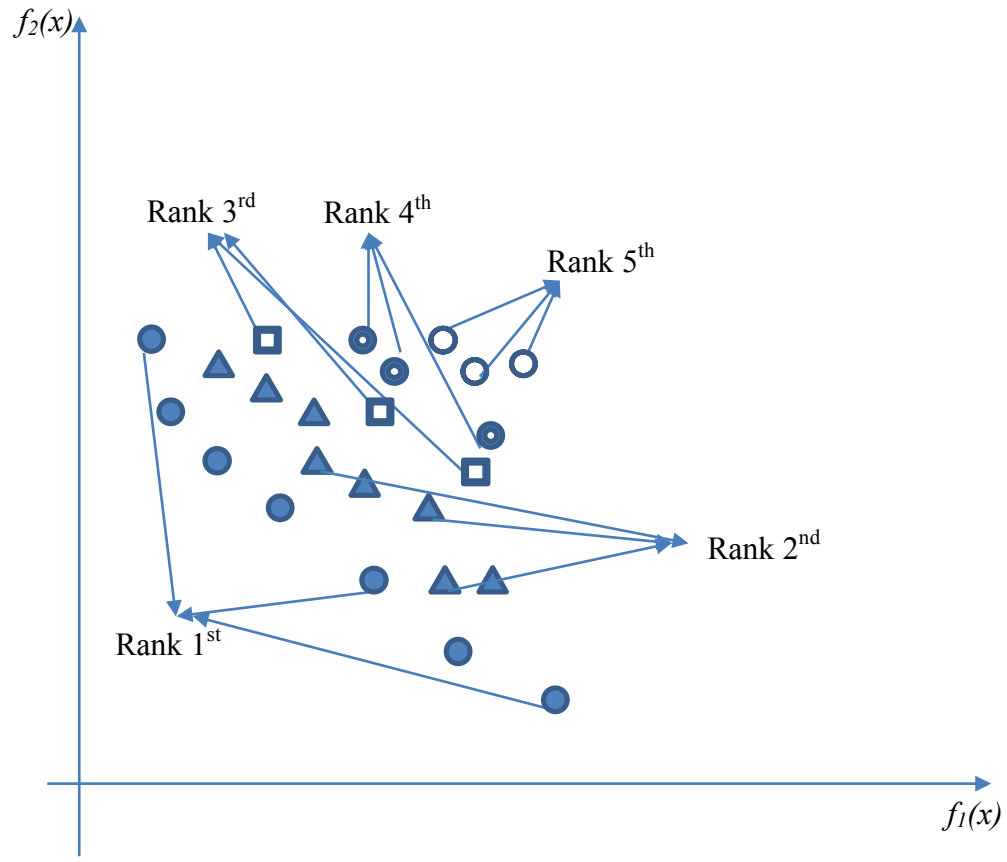


Figure 2.5 FFGA Illustration

2.5.5 WBGA Genetic Algorithm

WBGA [38] uses weighted average of normalized objectives with predefined weight. The drawback is the difficulties in nonconvex objective function space.

2.5.6 MOGA -Fonseca and Fleming's Multi-Objective Genetic Algorithm

MOGA was the first multi-objective GA that explicitly used Pareto-based ranking and niching techniques together to encourage the search toward the true Pareto front while maintaining diversity in the population. Fonseca and Fleming [111] proposed a Pareto-based ranking procedure. An individual's rank equals the number of solutions encoded in the population by which its corresponding decision vector is dominated.

Step 1: *Start with a random initial population* P_0.

Set t=0

Step 2: *If stopping criterion is met,*

then return P_0

Step 3: *Evaluate the fitness of the population:*

Step 3.1 Assign a rank r(x,t) to each solution $x \in P_t$ *with defined ranking scheme.*

Step 3.2 Assign fitness values f(x,t) to each solution based on r(x,t)

Step 3.3 Calculate the niche count nc(x,t) of each solution $x \in P_t$

Step 3.4 Calculate the shared fitness value f'(x,t) of each solution $x \in P_t$

Step 3.5 Normalize the f(x,t) by f'(x,t)

Step 4: *Use a stochastic selection method based on f' to select the parents of the mating pool.*

Apply the mutation and crossover on the mating pool

until offspring population Q_t *of size N is filled. Set* $P_{t+1} = Q_t$

Step 5: *Set t = t+1, goto Step 2.*

In this algorithm, the fitness assignment is performed with Pareto ranking, i.e., adaptive fitness sharing and continuous introduction of random immigrants. It's a simple extension of single objective GA, but can have slow convergence.

2.6 Mixed Multi-Objective Genetic Algorithms

Multi-objective genetic algorithms have wide success and applicability to many practical problems as discussed in section 2.3. Several well-known MOGA algorithms discussed have shortcomings as listed section 2.2, including the slow convergence speed to the Pareto optimal front; or missing theoretical convergence proof to the Pareto optimal front; or issues with proper stopping criterion. There were some researches attempted to address these shortcomings with mixed approaches.

Miettinen [91] suggested a weighted sum of objective functions, which is one type of a scale function. In this method, multi-objective optimization problem can be transformed into a single objective optimization problem with such a scale function. The weighted sum of objective functions is formulated with predetermined weights. If the weighted sum of objective function values of the locally optimal solution is better than that of the individual in the comparison set, it is treated as winner out of the population. However, the weighted sum of objective functions is known to be inappropriate in handling nonconvex problems and there are many issues that affect the performance of this approach [91]. Lina *et al* [149] used a hybrid evolutionary multi-objective optimization algorithm based on a probability function with a periodic increase and decrease of probability of local search. The local search module can overcome slow convergence problems. Among the popular genetic algorithm discussed above, Population-based non-Pareto approaches can produce multiple non-dominated solutions in parallel, and hence generate non-dominated solutions. But in contrast to the Pareto-based approaches, most of them do not make direct use of the concept of Pareto dominance.

2.7 Comparison of Genetic Algorithms

There are a number of studies conducted on the comparison of GAs to evaluate the correctness and performance of the GAs by using close to real world multi-objective optimization problem. An example of a NP-hard test problem, 0/1 knapsack problem, represents an important class of real-world problems. The comparison focused on the effectiveness in finding multiple Pareto-optimal solutions. The validity of the comparison requires the valid test case setup. In order obtain reliable and sound results, a test problem for a comparative and experimental study need to be chosen carefully. The tests and experiments need to be repeatable and verifiable. Additionally, in order to be applied in real-world situation, the problem should ideally represent a real-world problem. Some researchers suggested the experiments with knapsack problem, which is a multi-objective optimization problem, but difficult to solve (NP-hard). A 0/1 knapsack problem consists of a set of items, weights and profits associated with each item, and an upper bound for the capacity of the knapsack. The objectives are to find a subset of all items which maximizes the total of the profits in the subset, yet, all selected items need to have minimum size and can be fit into the knapsack, i.e. the total weight does not exceed the given capacity [4]. The two objectives are competing and conflicting to each and therefore, it is naturally a multi-objective optimization problem.

The test problems of 0/1 knapsack is defined as:

Input:

m: number of items

n: number of knapsacks

d: number objectives

$p_{i,j}$ = *profit of j in knapsack i*

$w_{i,j}$ = *weight of item j in knapsack i*

ci = capacity of knapsack i

Output:

Vector $X = (x_1, x_{2,\ldots} X_m) \in \{0,1\}^m$

Such that: $\forall_i \in \{1,2 \ldots n\} : \sum_{j=1}^{m} w_j \leq c_j$ *and* $F(x) = (f_1(x), f_2(x), \ldots f_n(x))$ *is maximum*

where: $f_i(x) = \sum_{j=1}^{m} p_{ij} * xi\, x_j$

Zitzler et al[5] did a comprehensive comparative studies on knapsack by using nine different test problems with different population size. The number of knapsacks and number of population are chosen differently. From the research in [5], random profits and weights were chosen, where $P_{i,j}$ and $w_{l,j}$ are random integers in the interval [10,100]. The knapsack capacities were set to half the total weight regarding the corresponding knapsack:

$\boldsymbol{c_j = 0.5 \sum_{j=1}^{m} w_{i,j}}$ As reported in [4], about half of the items are expected to be in the optimal solution (of the single-objective problem), when this type of knapsack capacities is used. The

test results are shown in Table 2.3 below. For comparison purpose, all GAs considered were implemented with the same selection scheme. From this test result, NSGA covers the greatest fraction of the Pareto sets achieved by the other algorithms. VEGA has second best performance in this comparison, similar to the results concerning the absolute size of the space covered. On the remaining test problems, VEGA and the weighted-sum approach show almost equal performance [5].

Algorithm		Number of knapsacks / Number of population									
A	B	2/100	2/250	2/L500	3/100	3/250	3/500	4/100	4/250	4/500	mean
Random	Weighted	0.011	0	0	O%	0	0	0	0	0	0.001
	Niched	0	0	0	0	0	0	0	0	0	0
	VEGA	0	0	0	0	0	0	0	0	0	0
	NSGA	0	0	0	0	0	0	0	0	0	0
Weighted	Random	0.98	1	1	1	1	1	0.993	0.999	1	0.997
	Niched	0.025	0.015	0	0.727	0.726	0.757	0.308	0.495	0.792	0.427
	VEGA	0	0	0	0.414	0.329	0.306	0.38	0.3	0.409	0.238
	NSGA	0	0	0	0.232	0.22	0.141	0.241	0.117	0.27	0.136
Niched	Random	1	1	1	1	0.996	1	0.996	0.999	1	0.999
	Weighted	0.925	0.95	l00%	0.129	0.201	0.146	0.408	0.265	0.045	0.452
	VEGA	0.009	0.103	0	0.124	0.142	0.076	0.478	0.233	0.086	0.139
	NSGA	0.007	0.044	0.022	0.077	0.056	0.008	0.271	0.088	0.036	0.068
VEGA	Random	1	1	1	1	1	1	0.994	1	1	0.999
	Weighted	1	0.988	1	0.438	0.546	0.474	0.343	0.483	0.347	0.624
	Niched	0.865	0.879	0.92	0.732	0.776	0.8	0.317	0.597	0.796	0.742
	NSGA	0.258	0.169	0.205	0.208	0.238	0.16	0.224	0.169	0.26	0.21
NSGA	Random	1	1	1	1	1	1	0.995	1	1	0.999
	Weighted	1	1	1	0.597	0.672	0.727	0.499	0.728	0.497	0.747
	Niched	0.938	0.975	0.988	0.88	0.897	0.952	0.511	0.844	0.91	0.877
	VEGA	0.58	0.763	0.674	0.625	0.587	0.727	0.605	0.724	0.58	0.652

Table 2.3 Comparison of GAs using Predefined Test Data [*from Zitzler, 5*]

However, there are several factors need to considered in the comparative study regarding the testing data and experiments,

a). the quantitative measures used to express the quality of the GA outcomes

b). the number of generations to reach optimal solutions.

c). the side effects caused by different selection schemes or mating restrictions in GAs

d). the initialization of parameters of the EA, particularly the niche radius

The quantitative measures can be addressed by using a set of non-dominated solutions. The Pareto-optimal set regarding all individuals generated over all generations is taken as output of an GA. The total number of Pareto-optimal solutions for all the objective space are used as a quantitative measure. In the case of convex solution space, certain solutions can be overrated. The overrating can be addressed by comparing the outcomes of the EAs directly by using the coverage relation. Given two sets of non-dominated solutions, each set the fraction of the solutions which are covered by solutions in the other set can be computed and use for comparison. Randomness is one of the key factors in genetic algorithm, especially in the initialization and selection process. To reduce the influence of random effects, the experiments need to be repeated per test problem, different randomly generated initial population need to be tested per experiment for all GAs ran on the same initial population. The performance of a particular GA on a given test problem can be evaluated by using the average of its performances over all experiments. Konak *et al* [187] presented a summary of comparison on some well-known genetic algorithms as shown in Table 2.4 below.

Algorithm	Fitness assignment	Diversity mechanism	Elitism	External population	Advantages	Disadvantages
VEGA [5]	Each subpopulation is evaluated with respect to a different objective	No	No	No	First MOGA Straightforward implementation	Tend converge to the extreme of each objective
MOGA [6]	Pareto ranking	Fitness sharing by niching	No	No	Simple extension of single objective GA	Usually slow convergence Problems related to niche size parameter
WBGA [8]	Weighted average of normalized objectives	Niching Predefined weights	No	No	Simple extension of single objective GA	Difficulties in nonconvex objective function space
NPGA [7]	No fitness assignment, tournament selection	Niche count as tie-breaker in tournament selection	No	No	Very simple selection process with tournament selection	Problems related to niche size parameter Extra parameter for tournament selection
RWGA [9]	Weighted average of normalized objectives	Randomly assigned weights	Yes	Yes	Efficient and easy implement	Difficulties in nonconvex objective function space
PESA [14]	No fitness assignment	Cell-based density	Pure elitist	Yes	Easy to implement Computationally efficient	Performance depends on cell sizes Prior information needed about objective space
PAES [29]	Pareto dominance is used to replace a parent if offspring dominates	Cell-based density as tie breaker between offspring and parent	Yes	Yes	Random mutation hill-climbing strategy Easy to implement Computationally efficient	Not a population based approach Performance depends on cell sizes
NSGA [10]	Ranking based on non-domination sorting	Fitness sharing by niching	No	No	Fast convergence	Problems related to niche size parameter
NSGA-II [30]	Ranking based on non-domination sorting	Crowding distance	Yes	No	Single parameter (N) Well tested Efficient	Crowding distance works in objective space only
SPEA [11]	Raking based on the external archive of non-dominated solutions	Clustering to truncate external population	Yes	Yes	Well tested No parameter for clustering	Complex clustering algorithm
SPEA-2 [12]	Strength of dominators	Density based on the k-th nearest neighbor	Yes	Yes	Improved SPEA Make sure extreme points are preserved	Computationally expensive fitness and density calculation
RDGA [19]	The problem reduced to bi-objective problem with solution rank and density as objectives	Forbidden region cell-based density	Yes	Yes	Dynamic cell update Robust with respect to the number of objectives	More difficult to implement than others
DMOEA [20]	Cell-based ranking	Adaptive cell-based density	Yes (implicitly)	No	Includes efficient techniques to update cell densities Adaptive approaches to set GA parameters	More difficult to implement than others

Table 2.4 General Comparison of Existing GAs [*from Konak et al, 187*]

2.8 Issues with Genetic Algorithm application in Multi-Objective Optimization

There are many variations of multi-objective genetic algorithms in the literature, the above cited GA are well-known algorithms used in many applications and their performances were tested in several comparative studies [111,187]. The implementation strategies of genetic

algorithms differ mostly in elitism in the selection operator and population diversity preservation. Because of the parallel nature and intensive iterative processes by generations, genetic algorithm has substantial high requirements on computational efforts. Therefore, the performance of GAs can not only be impacted by the design, but also by programming skills, data structure, computer hardware configuration including memory, disk IO, CPU clock time etc. Some of the main issues in genetic algorithms are discussed below.

2.8.1 Population Diversity

Maintaining a diverse population is critical in multi-objective GA in order to obtain the global optimal Pareto front solutions and hence the true global optimality. During the search process, the populations need maintain good diversity by getting uniform distribution of individuals and forming only relatively few clusters to keep the number of population under control. To generate and maintain diverse populations, fitness evaluation and assignment in the selection step plays important role. Several classical approaches proposed in the literatures are discussed below.

Fitness assignment in Pareto-based genetic algorithms is achieved by individuals' comparison. The mostly referred fitness assignment is fitness sharing, or niching techniques. Fitness sharing bases on the idea that individuals in a particular niche have to share the resources available, similar to nature. The fitness value of an individual is degraded if there are more individuals are located in its neighborhood. Neighborhood is defined in terms of a distance measure and specified by the so-called niche radius. The benefits of applying penalties to the fitness value of dense populations can help GAs to achieve better spread and diversity of population, hence reach the global optimality, other than local optimality.

Deb *et al* [79] reported the erratic behavior of genetic algorithms from the conventional combination of fitness sharing and tournament selection. NSGA uses a slightly modified version of sharing, called continuously updated sharing. It uses the partly filled next generation, other than the current generation to calculate the niche count. Horn and Nafpliotis [110] introduced this concept in the Niched Pareto GA as well.

NSGA-II [144] uses a crowding distance to obtain a uniform spread of solutions along the best-known Pareto front. The crowding distance method is described below:

Crowding distance method in NSGA-II:

Step 1*: Get the non-dominate set of individuals P_1 ... P_k form the population nin the current generation*

Step 2*: For each i= 1 to k of P_k*

Rank individual i and sort the P based on object function k

Define the crowding distance as:

$$d_k(x_{i,k}) = \frac{z_k(x_{i+1,k} - (x_{i-1,k})}{z_k^{max} - z_k^{min}}$$

where $x_{i,k}$ is the ith individual in the sorted population of objective k

Step 3*: Get the total crowding distance D =* $\sum_k d_k(x)$

From the above description, the density of the population can be measured by the crowding distance. Fitness value is not required in the definition of the density distance measure. This crowding distance measure used in a selection technique is also called the crowded tournament selection operator. By this definition, if the two solutions *x* and *y* are in the same non-dominated front, the solution with a higher crowding distance will be selected and put into the next generation. In PESA, the objective space is divided into regions or cells and the number of solutions in each cell is defined as the density of the cell, and the density of a solution is equal to the density of the cell in which the solution is located. The density index defined this way can achieve diversity similarly. Between two non-dominated solutions, the one with a lower density is preferable. PESA-II further refined the design of density definition by using region-based selection, instead of using individual solutions, cells or regions are selected during the selection process.

In addition to the density definition, probability method is also used in the selection process. A sparse cell has a higher chance to be selected than a crowded cell for the next generation. Once a cell is selected, solutions within the cell are randomly chosen to participate to crossover and mutation. Lu and Yen [148] developed an efficient approach to identify a solution's cell density in case of dynamic cell dimensions. In this approach, the width of a cell along the *k*th objective dimension $(z_k^{max} - z_k^{min})/n_k$ is used, where n_k is the number cells dedicated to the *k*th objective dimension and z_k^{max} and z_k^{min} are the maximum and minimum values of the objective function *k* in the current search. Cell boundaries are updated when a new maximum or minimum objective function value is discovered. This approach has better computational efficiency compared to the niching or neighborhood-based density techniques.

Yen and Lu [132] proposed several data structures and algorithms to efficiently store cell information and modify cell densities. From the result of the density calculation, the cell-based density approach can also obtain a global density map of the objective function space. The search can be encouraged toward sparsely inhabited regions of the objective function space based on this map.

2.8.2 Objective function complexity

The fitness function determines the quality of the populations through the selection of non-dominate individuals. Evolution of the population takes place after the repeated application of the genetic operators with selected the non-dominate population. Coello [135] did a complete survey on the methods of objectives constraints handling in single-objective genetic algorithm, including discarding infeasible solutions, reducing the fitness of infeasible, using genetic operators to always produce feasible solutions; and transforming infeasible solutions to be feasible.

The evaluation of objective functions in the multi-objective optimization may take considerable time in solving real-life problems [135]. Reducing execution time and resource requirements of multi-objective GA using advanced data structures is the one of most interested research areas.

VEGA is the first GA used to approximate the Pareto-optimal set by a set of non-dominated solutions. Each solution in subpopulation *Pi* is assigned a fitness value based on objective function z_i in the same way as for a single objective GA. This type of using single objective GA to solve multi-objective problems is computationally efficient as it reduces the

complexity of selection operator, but the objective switching through the entire objective space tends to converge to the local optimality [116].

NSGA uses Pareto dominance in evaluating fitness or assigning selection probability to solutions. The population is ranked according to a dominance rule, and then each solution is assigned a fitness value based on its rank in the population, not its actual objective function value. SPEA used a ranking procedure to assign better fitness values to non-dominated solutions. The ranking procedure selects solution which covers the least number of solutions in the objective function space to achieve a wide, uniformly distributed set of non-dominated. However, because in multi-objective GA, the fitness assignment is based on the non-dominance rank of a solution, not on its objective function values, the implementation of penalty function strategies is not straightforward. As discussed in section 2.8.1, some GAs penalize redundancy in the population due to overrepresentation through ranking density [110,111].

One of the latest trends is parallel and distributed processing. Several recent papers [159,121] presented parallel implementation of multi-objective GA over multiple processors. But the data structure and design patterns in the implementation of algorithms are not discussed in depth.

2.8.3 Maintaining elitist solutions in the population

All non-dominated solutions, Pareto front, discovered by a multi-objective GA are considered elite solutions. Elitism means that the best solution found so far during the search always survives to the next generation, i.e., best organism(s) from the current generation to carry over to the next to guarantee that the solution quality obtained by the GA will not decrease from

one generation to the next generation. Multi-objective GA using elitist [3], tend to outperform their non-elitist counterparts. Because of the large number of possible elitist solutions in multi-objective optimization, the maintenance of elite solutions can be very complex. Multi-objective GA in general uses two strategies to implement elitism [3]:

- Maintaining all elitist solutions in the population internally, as discussed in NSGA above. All non-dominated solutions in population P_t are copied to population P_{t+1}, then filling the rest of P_{t+1} by selecting from the remaining dominated solutions in P_t. In this case, no external storage is used to store discovered non-dominated solutions. When the total number of non-dominated parent and offspring solutions is larger than initial set size of population N_P, this approach will not work and some additional measures need to be taken to reduce the elite population size.
- Storing elitist solutions in an external secondary list and re-introducing them to the population [3]. During the search space, non-dominated solutions found in the current generation so far are stored in an elitist list E and E is updated each time a new solution is created by removing elitist solutions dominated by a new solution or adding the new solution if it is not dominated by any existing elitist solution [3]. The manipulation of list E is computationally expensive. To efficiently store, update, and search in list E, there are some data structures have been proposed [33, 130]. There also might possibly exist a very large number of Pareto optimal solutions for a problem, the size of list E can grow extremely large. Trimming or pruning techniques [146] have been proposed to control the

size of E. For example, SPEA[137] used the average linkage clustering method to reduce the size of E to an upper limit N when the number of the non-dominated solutions exceeds N as follows:

Step 1: *Assign each solution x∈E to a cluster* c_j,

$C = \{c_1, c_2, \ldots. c_m\}$

Step 2: *Calculate the distance between all pairs of clusters* c_i *and* c_j *as below:*

$$d(ci, cj) = \frac{1}{|c_i|+|c_j|}\sum_{x,y} d(x, y)$$

Where $x \in ci$, $y \in cj$. *d(x,y) can be calculated in objective function space.*

Step 3: *Merge the cluster pair* c_i *and* c_j *with the minimum distance among all clusters into a new cluster*

Step 4: *If* $|C| <= N$, *goto Step 5, else goto Step2*

Step 5: *For each cluster, determine a solution with the minimum average distance to all other solutions in the same cluster (Centroid solution). Keep the centroid solutions for every cluster and remove other solutions from E.*

Other examples of elitist approaches using external populations are PESA [147], RDGA, RWGA, and DMOEA [111].

2.8.4 Fitness function evaluation for complex problems

To understand the difficulty of high-dimensional multi-objective optimization problems, some researchers suggested the use of fitness landscape metaphor [111], i.e. the ability of a searcher to find the optimal solution for that problem. In general fitness landscapes cannot be drawn because of the huge dimension of the search space, it is important to define the important features of fitness landscapes that have a direct relationship with the difficulty of the problem. fitness-distance correlation and negative slope coefficient [160, 179] are the two interesting measures of problem hardness based on the concept of fitness landscape.

For a complex problem, because of the high-dimension and multi-modalities of the search space and number of objectives, finding the optimal solution to complex high-dimensional, multimodal problems often requires very expensive fitness function evaluations. A single function evaluation may require several hours to several days of complete simulation. In a real-world situation, decision makers often require the timely output of the Pareto-optimal solutions, and therefore, it may be necessary to forgo an exact evaluation and use an approximated fitness that is computationally efficient. A number of researchers suggested a workaround by using an approximated fitness that is computationally efficient, instead of extract fitness value. The use of amalgamation of approximate models in GA to solve complex real-life problems are described in [75,95,102,107,112].

However, with the advance of modern computing technology, including parallel processing and cloud-based computing technologies, the performance problem of fitness evaluation can be much alleviated, as discussed in the methodology section.

2.8.5 Scalability problem

Genetic algorithms do not scale well with complexity. Because the chromosome encoding for the problem representations can evolve during the evolutionary process, and the complexity of solutions can evolve too. In classical genetic algorithm, initially individuals are randomly selected from an initial population set with the other initial setup parameters, fitness functions for the objectives, and termination conditions. The size of search space will increase exponentially as new generations are produced following by crossover and mutations operator. Hence the computational complex will increase accordingly. In order to make evolutionary search manageable and converge, genetic algorithm needs to improve performance by using appropriate data structure, programming skills with code reusability and design patterns. Some approaches were developed by adding modules to the system, for example, n tree-based representations of population. The most well-known of these methods is Koza's Automatically Defined Functions (ADFs) [74]. The search space was broken down into the simplest representation possible, but with the isolated parts that have evolved to represent elite solutions from further destructive mutation, particularly when their fitness assessment requires them to combine well with other parts.

There are a large and varied literature related to modularity in genetic algorithm, the use of modularity in genetic programming can help to solve some problems like number of iteration and new data abstractions, for instance in the form of ADFs [74], but the challenges and issues still remain open in most cases.

2.8.6 Benchmarks problem

It is difficult to define a true Pareto-optimal set just based on the fitness value at the end of evolutionary process, especially when the experimental data are not well-known or not well studied. The "better" solution is only in comparison to other solutions. As a result, the stop criterion is not clear in every problem. Genetic algorithms are not applicable to solve decision problems in which the only fitness measure is a linear single right/wrong measure, as there is no way to converge on this type of solution where a random search may find a solution quickly. However, for a multi-objective optimization problem which is not linear or not convex, genetic algorithm can use the ratio of successes to failures to provides a suitable fitness benchmark measure to produce the Pareto-optimal solution by fitness evaluation.

2.8.7 Data visualization problems

Data visualization itself is not a part of MOGAs process, however, in order for the MOGAs to be used in real-world problem solving, the Pareto-optimal sets need to be presented in a way that can be understood by the decision makers who may not have technical background. Therefore, data visualization is an important tool in the decision-making process. It allows business decision makers to quickly examine large amounts of data, diagnose the trends and issues efficiently, exchange ideas with key players, and influence the decisions through the experiments results visualization that will ultimately lead to success.

2.8.8 Local optima problem

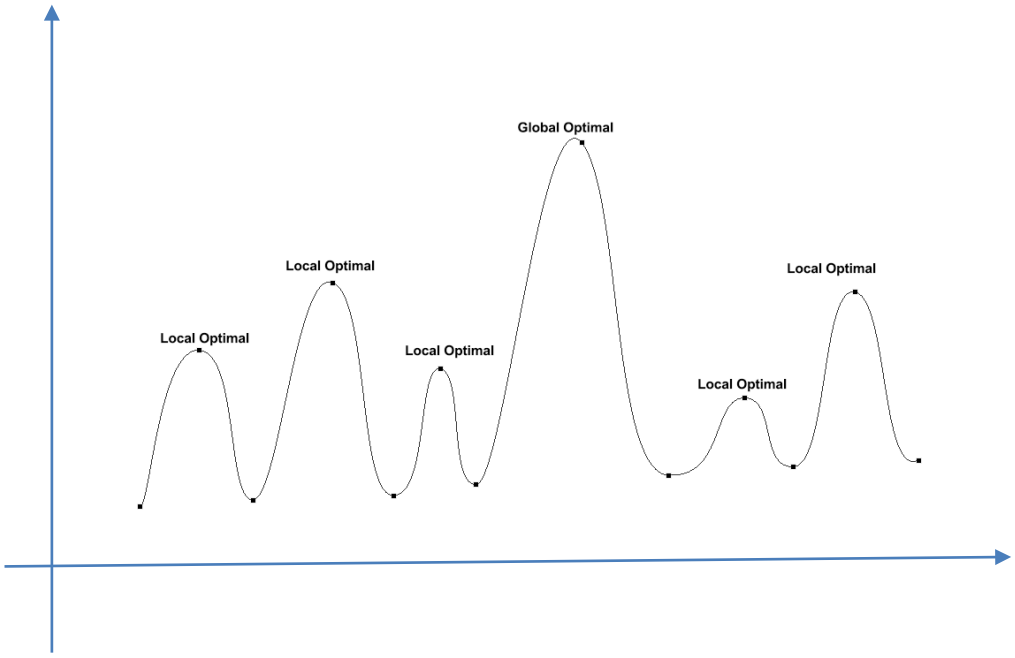


Figure 2.4 Local vs Global Optimality

Genetic algorithm is designed to find the optimal sets from the search space. If not properly designed or setup, genetic algorithm may terminate or converge towards local optima or even arbitrary points rather than the global optimum of the problem, as shown in Fig. 2.4. The search process in the n-dimensional space is by heuristic process, and termination condition are not mathematically determinative, and thus it's difficult for the algorithm to know whether the final solution sets or global optimum found is the best one, or there is better solution yet to be searched. Due to the inherent heuristic nature of the optimization process, there is no absolute optimum, or no single best solution. There are some researches to alleviate the local problems by using a different fitness function, increasing the rate of mutation, or by using selection techniques that maintain a diverse population of solutions. However, the No Free Lunch theorem [105] proves that there is no general solution to this problem. Other techniques include re-

sampling by simply replacing part of the population with randomly generated individuals, when most of the population is too similar to each other; adding penalty to any group of individuals of sufficient similarity (niche radius), which will reduce the representation of that group in subsequent generations, permitting other (less similar) individuals to be maintained in the population, etc.

To prevent early convergence, some research proposed to increase genetic diversity either by increasing the probability of mutation when the solution quality drops or by occasionally introducing entirely new, randomly generated elements into the gene pool [120] [121] [122], where a population of candidate solutions is employed other than maintaining a single candidate solution. This allows a diversity of potential solutions to be maintained, which increases the likelihood that a sufficient solution exists at any point in time to ensure the survival of the population in the long term. Population diversity is important in genetic algorithms because crossing over a homogeneous population does not yield new better solutions and result in early convergence to local optimality.

2.8.9 Stop/termination problem

Due to the inherent heuristic nature of the optimization process, there is no absolute optimum, or no single best solution. The "better" solution is only in comparison to other solutions. Depending on the complexity of the problem and the shape of the fitness landscape, certain termination conditions may be required stop the search process, for example, the termination conditions can be defined as maximum time of execution, threshold of fitness differences between generations.

2.8.10 Dynamic dataset problem

Given the degree and frequency of population changes that can occur during the evolution, operating on dynamic data sets is difficult, as genomes begin to converge early on towards solutions which may no longer be valid for later generations. Elitism is effective on solving dynamic dataset problems.

2.9 Multi-Objective Genetic Algorithm Application in Clustering Analyses

Data clustering is one of the key tasks in data mining, knowledge discovery and can be used of multi-objective genetic algorithms to reduce the population size through the selection of centroid representative individuals. There are a number of clustering algorithms developed in the past [2-19], but a good cluster result depends on the application subject to various criteria, both ad hoc and systematic. Clustering algorithms arise in many different applications, such as data mining and knowledge discovery, data compression and vector quantization, pattern recognition and pattern classification [10,11,12].

The objectives on clustering can be summarized below:

- ***$f_1(x)$: Maximize homogeneity within the cluster***
- ***$f_2(x)$: Maximize separateness between clusters***
- ***$f_3(x)$: Minimizing the number of clusters.***

These objectives are conflicting with each other. f_2, the maximization of separateness between clusters will result in the number of clusters increases, which is contradictory to objective f_3. Maximization of Objective f_1 also result in the maximization of objective f_3. Hence clustering itself is naturally a multi-objective optimization problem. As the number of clusters

decreases, the values of the other two objectives will be negatively affected. To reach the multi-objective optimization, some trade-offs between the quality of a clustering result and the number of clusters need to be taken for the final solution from the Pareto-optimal solution set.

2.9.1 Classical Clustering Methods

Traditional clustering techniques can be classified into hierarchical clustering [31], Centroid-based partition clustering [32, 70, 170, 183], graph-based [65] and distribution model-based [40] approaches. Typical cluster models that related with this research are summarized below.

- *Connectivity-based hierarchical clustering*

Hierarchical clustering is based on the idea of data nodes being more related to nearby objects than to objects farther away. Based on the distances between nodes, clustering algorithms calculate and form clusters with individual nodes. Hierarchical clustering methods are categorized into agglomerative (bottom-up) clustering and divisive (top-down) clustering [31]. An agglomerative clustering starts with one-point (singleton) clusters and recursively merges two or more clusters at a time until a single cluster is obtained. A divisive clustering starts with one cluster of all data points and recursively splits the most appropriate cluster until each point ends in a cluster. For both categories, the process continues until a stopping criterion is achieved [31].

Hierarchical clustering is robust with respect to input parameters, less influenced by cluster shapes, less sensitive to largely differing point densities of clusters, and it can represent nested clusters. However, the tree structure is prone to errors and it suffers from different aspects as stated by statisticians, including robustness, non-uniqueness, and inverse interpretation of the

hierarchy [31]. Segal *et al* [188] proposed probabilistic abstraction hierarchies (PAH), where each class is associated with a probabilistic generative model for the data in the class. This method improved the performance of traditional hierarchical clustering by handling the drawbacks mentioned above. It is more robust and less sensitive to noise in data.

- *Centroid-based partition clustering*

K-Means is a commonly used algorithm for partition clustering [32, 70, 170, 183]. K-means clustering aims to partition N datasets into K clusters in which each dataset belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This initial step of k-means algorithm starts with seeds positions defined [32], All data elements are assigned to the nearest seed and the process repeats on a new assignment step until no further improvement can be made, also known as “local optimum” has been found. Implicitly this process will produce a minimization of the "sum of the L2 distance squared between each data point and its nearest cluster center" [70]. This results in a partitioning of the data space into Voronoi cells. [33, 38,39] The purpose of K-Means clustering is the optimization of an objective function that is described by the equation:

$$E = \sum_{i=1}^{c} \sum_{x \in C_i} d(x, m_i)$$

where m_i is the center of cluster Ci, and d(x, m_i) is the Euclidean distance between a point x and m_i.

The object function is to minimize the distance between each point and the center of its cluster. The algorithm begins by randomly initializing a set of C cluster centers, then assigns

each object of the dataset to the cluster whose center is the nearest, and re-computes the centers. This process is repeated until the total error criterion converges.

Most k-means algorithms [32, 70, 170, 183] requires the number of clusters K to be specified in advance, and tend to cluster datasets into approximately similar size by assigning an object to the nearest centroid. This can result in the incorrectly cut borders in between of clusters, and convergence to a local minimum. In case that data sets are in n-dimensions, k-means clustering algorithm to get the optimal number of cluster can be NP-hard [32, 70, 170, 183]. If k and d (the dimension) are fixed, the problem can be exactly solved in time $O(n^{dk+1} \log n)$where n is the number of entities to be clustered. Thus, a variety of heuristic algorithms such as Lloyd's algorithm given above are generally used [39].

- *Distribution model-based clustering*

The model-based approach assumes that data are generated by a mixture of finite number of probability distributions. If a complex probability model is used, a small number of clusters may suffice, while if a simple model is used, a larger number of clusters may be needed to fit all the data appropriately. Examples of model-based approach are Bayesian method and the mixture model-based algorithm (EMMIX-GENE). The Bayesian method is a distribution model-based approach used in gene expression data analysis. Mar [40] proposed a mixture model-based algorithm (EMMIX-GENE) for the clustering of tissue samples and presented a case study involving the application of EMMIX-GENE to breast cancer data.

The Bayesian method has the advantage that it can identify the number of distinct clusters but it has the disadvantage of relying on the assumption that the modeled time series are stationary [40]. The model-based approach assumes that data are generated by a mixture of finite

number of probability distributions. In this approach, each cluster represents a probability distribution and a likelihood-based framework can be used. However, the assumption of Gaussian distribution in real dataset can lead to inaccurate clustering result.

- *Density based clustering*

In density-based clustering [45,46,50], clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points.

DBSCAN [45] is a popular density based clustering method which features a well-defined cluster model called "density-reachability". It connects points within certain distance thresholds, i.e., only the points that satisfy a density criterion, in the original variant defined as a minimum number of other objects within this radius. A cluster consists of all density-connected objects (which can form a cluster of an arbitrary shape, in contrast to many other methods) plus all objects that are within these objects' range. The complexity of DBSCAN is low as it requires a linear number of range queries on the dataset and it is deterministic for core and noise points in each run, therefore there is no need to run it multiple times.

OPTICS [50] is a generalization of DBSCAN that removes the need to choose an appropriate value for the range parameter, and produces a hierarchical result related to that of linkage clustering.

DBSCAN, OPTICS and other similar density-based clustering only work well with clusters that have some kind of density drop in order for the algorithm to detect cluster borders.

Moreover, because of the intrinsic cluster structures in real life data, DBSCAN and OPTICS cannot detect the clusters' borders accurately.

Mean-shift[134] is a clustering approach where each object is moved to the densest area in its vicinity, based on kernel density estimation. It is a non-parametric feature-space analysis technique for locating the maxima of a density function. Eventually, objects converge to local maxima of density. Similar to k-means clustering, mean-shift can detect arbitrary-shaped clusters similar to DBSCAN. Mean-shift algorithms require expensive iterative procedure and density estimation.

- *Graph based clustering*

Self-Organizing Maps (SOM) [10] maps centroids into 2D plane for better visualization and analyses. It provides a straightforward visualization, and therefore it is popular in vector quantization for clustering. Vector Quantization is a special case of the SOM and is essentially the same as the k-means algorithm. It is a neural network approach that uses competitive unsupervised learning and eventually the winner-takes-all approach.

The priori condition that SOM need to have is the shape and size of a network of clusters to fit the data into, i.e., the size of the two-dimensional grid and the number of nodes have to be predetermined [28,58,59]. This can be problematic if this type of information is unknown before the clustering starts.

2.9.2 Genetic Algorithm in Clustering

As discussed in Section 2.9.1, the objectives of clustering are multi-modal and self-conflicting. The classical clustering methods works well when the datasets to be clustered are

well-understood and the parameters and objective functions are pre-defined, however, lack of the prior knowledge of the datasets are common an, especially when the datasets are large, multi-dimensional and multi-modal, therefore, use of classical methods can lead to the in-accurate or not solution (NP-Hard) [35, 178,181,182]. In this case, genetic algorithm is more suitable and application to get the best clustering solutions because it does not require the prior knowledge about the datasets, and it can produce the approximation of the best sets of solutions, also known as the Pareto-optimal sets of number of clusters.

2.9.3 Application of Clustering

In computational biology and bioinformatics, clustering is employed to find the relationships among genes and discover the hidden knowledges. By clustering, genes/samples groups and intrinsic relationship between them can be discovered. In data distribution direction, observations are gauged to fit certain probabilistic distribution such as Gaussian or Mixed Gaussian, and clustering process is statistical manipulations on distributions [157-164].

Clustering different samples based on gene expression is one of the key issues in problems like class discovery, normal and tumor tissue classification, and drug treatment evaluation [164].

Clustering analysis can also be used to find direct gene-sample correlations [179]. BiCluster [76] enables gene/condition correlation analysis that can lead to molecular classification of disease states, identification of co-fluctuation of functionally related genes, functional groupings of genes, and logical descriptions of gene regulation, among others.

2.9.4 Clustering validation

The criteria used to evaluate the outcome and performance of clustering algorithms are compactness of the clusters and their separateness. These criteria should be validated and optimal clusters should be found. Clustering validity criteria used for the validation include Dunn index, Davies-Bouldin (DB) index, Silhouette Coefficient, C index, SD index and S_Dbw index etc. [24].

The SD validity index definition is based on the concepts of average scattering for clusters and total separation between clusters. The average scattering for clusters is defined as [24].

$$Scattering(n_c) = \frac{i}{n_c}\sum_{i=1}^{n_c} | \sigma(v_i) \, / \sigma(x) |$$

where $\sigma(v_i)$ is the average standard deviation (average of the Euclidian distance between all the points) of cluster centers; and $\sigma(x)$ is the average standard deviation of all the data points.

The total separation between clusters is defined as:

$$Separation(n_c) = \frac{Dmax}{Dmin}\{\sum_{k=1}^{n_c}\sum_{i=1}^{n_c}|v_k - v_z|\}^{-1}$$

where $D_{max} = max(||v_i - v_j||) \quad \forall i, j \in \{1, 2,3,\ldots, n_c\}$ is the maximum distance between cluster centers and $D_{min} = min(||v_i - v_j||) \quad \forall i, j \in \{1, 2,\ldots, n_c\}$ is the minimum distance between cluster centers.

The SD index is calculated using the following equation [24]:

$$SD(n_c) = \alpha * Scattering(n_c) + Separation(n_c)$$

where α is a weighting factor.

Scattering(n_c) indicates the average compactness of clusters. *Separation*(n_c) indicates the total separation between the *n* clusters. A weighting factor *α* is needed to incorporate both terms in SD definition to balance out the two terms. The number of clusters that can minimizes the index is an optimal value.

S_Dbw[24] is based on the clusters' compactness (intra-cluster variance) and the density (Inter-cluster Density) between clusters. Inter-cluster density is defined as follows:

$$DensityInter(n_c) = \frac{1}{n_c(n_c-1)} \sum_{k=1}^{n_i} \sum_{i=1}^{n_c} \left| \frac{density(u_{ij})}{Max(density(v_i, v_j)} \right|$$

where v_i and v_j are centers of clusters c_i and c_j; and u_{ij} is the middle point of the line segment defined by the clusters' centers v_i and v_j. The term density(u) is given by following equation:

$$Density(u) = \sum_{i=1}^{n_{ij}} f(x_i, u)$$

where n_{ij} is the number of tuples that belong to clusters c_i and c_j, i.e., $x_l \in c_i$, and $c_j \in S$. Function f(x,u) is defined as:

$$f(x,u) = f(x,u) \begin{cases} 0, if\ (d(x,u) > stdev \\ 1, otherwise\ \ above \end{cases}$$

where stedev is the average standard deviation of cluster.

Inter-cluster Density (ID) evaluates the average density in the region among clusters in relation to the density of the clusters. Intra-cluster variance measures the average scattering of clusters (*Scat*(n_c)) and has already been defined in the *SD* index part [24].

The *S_Dbw* is calculated using the following equation:

$$S_{S_{Dbw(_{nc})}}$$

$$S\_dbw(n_c) = Scattering(n_c) + DensityInter(n_c)$$

The definition of *S_Dbw* considers both compactness and separation. The number of clusters that minimizes the index is an optimal value.

The Dunn index is calculated using the following equation [191]:

$$D(n_c) = \min\{\min\{\frac{\frac{1}{|c_i||c_j|}\sum d(x,y)}{\max(\frac{\sum_{x\in c_k} d(x,c_k)}{|c_k|})}\}\}$$

where c_i represents the i-cluster of a certain partition, d(x,y) is the distance between data points x and y, where x belongs to cluster i and y belongs to cluster j , d(x, c_k) is the distance of data point x to the cluster center that it belongs to, $|C_k|$ is the number of data points in cluster K.

The main goal of the measure is to maximize the intercluster distances and minimize the intracluster distances. Therefore, the number of clusters that maximizes *D* is taken as the optimal number of clusters.

The DB index is calculated using the following equation [24]:

$$DB = \frac{1}{n}\sum_{n=1}^{n} \max\{\frac{S_n(Q_i) + S_n(Q_j)}{S(Q_i, Q_j)}\}$$

where n is the number of clusters, S_n is the average distance of all objects from the cluster to their cluster center, $S(Q_i,Q_j)$ denotes the distance between centers of clusters.

The Davies-Bouldin index is a function of the ratio of the sum of within-cluster scattering to between clusters separation. When it has a small value, it exhibits a good clustering.

The following formula is used to calculate the Silhouette index [24]:

$$S(i) = \frac{(b(i) - a(i))}{\max\{a(i), b(i)\}}$$

where $a(i)$ is the average dissimilarity of i-object to all other objects in the same cluster, Euclidian distance is used to calculate the dissimilarity; and $b(i)$ is the average dissimilarity of i-object to all objects in the closest cluster.

The above formula indicates that the silhouette value is in the interval [–1, 1]:

- Silhouette value is close to 1: means that the sample is assigned to a very appropriate cluster.
- Silhouette value is about 0: means that the sample lies equally far away from both clusters; it can be assigned to another closest cluster as well.
- Silhouette value is close to –1: means that the sample is "misclassified".

The application of Dunn index aims to identify dense and well-separated clusters. It is defined as the ratio between the minimal inter-cluster distance to maximal intra-cluster distance. For each cluster partition, the Dunn index can be calculated by the following formula [24]:

$$D = \frac{\min_{1 \le i < j \le n} d(i,j)}{\max_{1 \le k \le n} d'(k)},$$

where d(i,j) = Distance (clusters i and j)

d '(k) = Intra-cluster distance of cluster k.

d(i,j) can be any number of distance measures, such as the distance between the centroids of the clusters. Similarly, d '(k) can be measured in a variety ways, such as the *Max(Distance(any pair of elements in cluster k))*. Clusters with high Dunn index are more desirable [27] because the internal criterion seek clusters with high intra-cluster similarity and low inter-cluster similarity.

Silhouette Coefficient [24] contrasts the average distance to elements in the same cluster with the average distance to elements in other clusters. Objects with a high silhouette value are considered well clustered, objects with a low value may be outliers because it is based on tightness and separation of clusters. It finds the overall average of the ratio of the difference of each object's minimum average dissimilarity to all objects in other clusters. This index works well with k-means clustering, and is also used to determine the optimal number of clusters [27].

Depending on the shape and size of the datasets, cluster validity index should be applied carefully. For example, many evaluation indexes assume convex clusters where k-means clustering is used because it is good to find convex clusters, but for anon-convex clusters, k-means, or any evaluation criterion that assumes convexity should be avoided.

Other evaluations for clustering results are based on external evaluation benchmarks in comparison with aforementioned internal evaluation. The external benchmarks consist of a set of pre-classified items, and these sets are often created by domain subject matter expert. These types of evaluation methods measure how close the clustering is to the predetermined benchmark classes. However, this type of evaluation may not be applicable to real world data sets because classes can contain internal structure and the attributes present may not allow separation of clusters or the classes may contain anomalies, and therefore the reproduction of known knowledge may not necessarily be the intended result [27,36,37,38]. External evaluation criterion includes the following method.

The Jaccard index [24] is a statistic used for comparing the similarity and diversity of sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets: is used to quantify the similarity between two datasets. The Jaccard index takes on

a value between 0 and 1. An index of 1 means that the two datasets are identical, and an index of 0 indicates that the datasets have no common elements. The Jaccard index is defined by the following formula [24]:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

where $0 \leq J(A,B) \leq 1$ *If* $A = \varnothing$ *and* $B = \varnothing$, *then* $J(A,B) = 1$

C-index is another technique used for cluster validity. It uses the within cluster pairwise dissimilarity. Further, according to the number of pairs in the within cluster pairs, minimum and maximum summation of the number of pairwise object distance parameters are used in the calculation. However, this method is not recommended since it is likely to be data dependent. SD index is evaluated by using the average scattering for clusters and the total scattering between clusters. S_Dbw is similar to SD index, but it also considers inter-cluster density instead of total scattering in SD, and no weighting is used. Density formula uses the average standard deviation of the clusters.

Examples of other cluster validity approaches used in gene expression data analysis include Principal Component Analysis (PCA) [68] and Gap statistic [69]. PCA is a statistical method that can improve the extraction of cluster structure and compare clustering solutions [68]. Gap statistic utilizes within-cluster distance to determine the “appropriate” number of clusters in a dataset. It is good at identifying well-separated clusters, but it does not produce satisfactory results for not-well-separated data and data concentrated on a subspace.

2.10 MOGA Applications in Resources Management

There are a number of researches on using genetic algorithms (GAs) to solve multi-objective resource allocation problems [163][165][167][168] due to the limitation of dynamically programming such as performance and scalability [163][166]. Real-world multi-objective optimization problems are difficult to deal with because they are multidimensional data, which varies in precision and resolution. Genetic algorithm can provide meaningful classifications and tackle the challenges prompted by the presence of some fuzziness in data. There have been several interesting and successful applications of multi-objective GAs in solving real-world problems.

In engineering, many problems involve multiple conflicting objectives, an ideal target value for each objective, and the desire is to get as close as possible to the desired value of each objective, sacrifice and trade-offs have to be made in order to get the most desirable result. For example, Sharizi *et all* [94, 95] used multi-objective optimization in energy systems to manage a trade-off between performance and cost. Amirahmadi [137] employed SPEA multi-objective optimization on the optimal controller design to solve problems are subject to linear equality constraints that prevent all objectives from being simultaneously perfectly met, especially when the number of controllable variables is less than the number of objectives and when the presence of random shocks generates uncertainty.

In designing high performance buildings, designers often have to deal with multiple and conflicting design objectives in the same requirement, for example, minimum energy consumption but with maximum thermal comfort, minimum energy utilization efficiency with minimum construction cost. This has led to the application of multi-objective optimization

algorithms (MOOAs) that identify the Pareto optimum trade-off between conflicting design objectives [147] [148]. Nguyen *et al* [146] conducted a comprehensive review on optimization methods applied to building optimization problems, as shown in Figure 2.5. The result was derived from more than 200 building optimization studies given by SciVerse Scopus of Elsevier. [146]

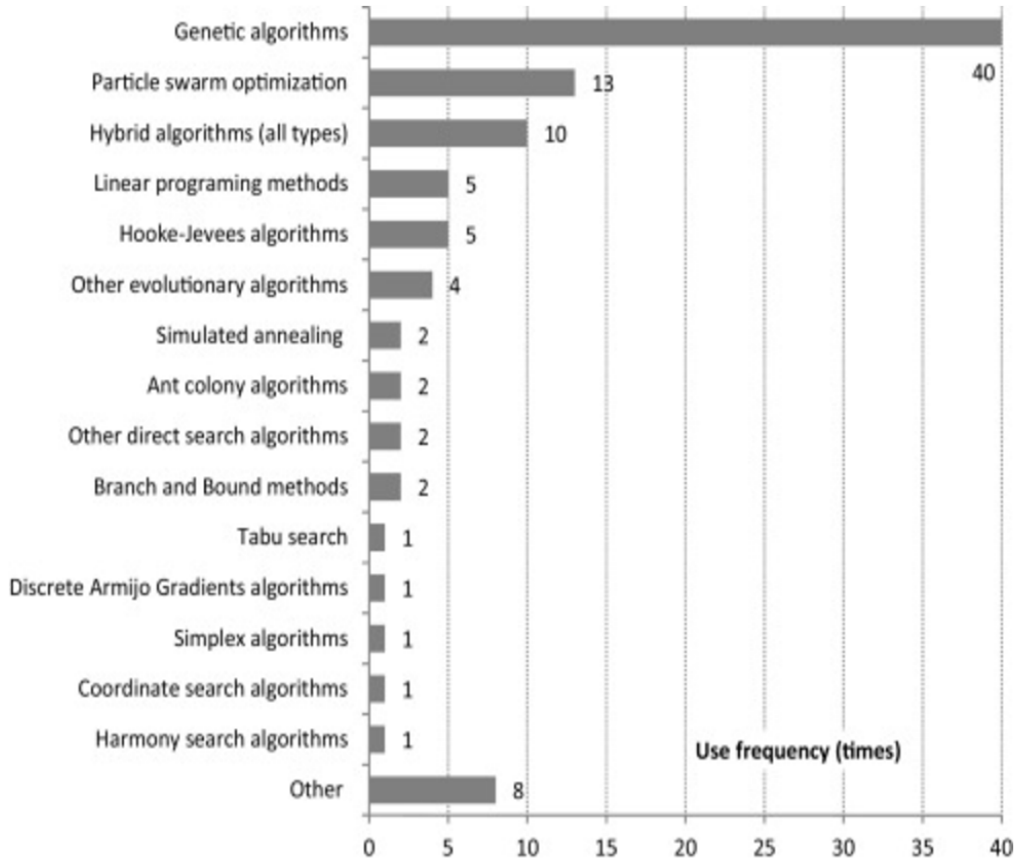


Figure 2.5 Optimization Algorithms Application in Building Design *[from Nguyen, 146]*

Resources management optimization usually involves multiple but conflicting objectives. There are only a limited number of resources available, yet there are a number of different ways in which the resources are needed and allocated. Reviewing all multi-objective evolutionary algorithm on resources management is beyond the scope of this research because of a large number of existing evolutionary optimization methods. The research on scarce resources management, for example, blood bank management, in general requires an understanding of the

subject domain concepts, the decision-making, and the interdependence of all related factors in the domain area.

Blood inventory management belongs to the scarce resource management in general as blood or red cell is in high demand by hospitals, yet the supply is limited. Red cell has shelf life of around 40 days, in order to make sure there are sufficient supply for patients, hospital authorities usually over-stock the inventory, which will inevitably result in the waste. The objectives are to maintain the maximum amount of inventory for sufficient supply to patients and minimize the waste. These two objectives are conflicting and yet need to be optimized at the same time.

There are few researches on using MOGA in blood bank management due the complexity of problem domain, Hsieh[151] used NSGA-II for blood bank supply chain model. Sivakumar *et al* [152] used a genetic program for inventory and routing structure optimization, which is also part of supply chain distribution problem. Adewumi[153] used genetic algorithm on Assignment of Blood in a Blood Banking System with some initial result. Most of existing research and literatures focused on the supply chain model but missing the other real-world situations, including emergency/epidemic situation, prediction of shortage, visualized Pareto-optimal result set for the decision makers etc. Additionally, performance of MOOAs on blood bank management optimization problems are not well researched. Therefore, a more complete multi-objective optimization framework with comprehensive reviews in solving blood bank management problem is needed for decision maker.

Chapter Three: METHODOLOGY

This research proposed a novel framework of hybrid multi-objective genetic algorithm with expert system. The Genetic algorithm module is based on the contributions from previous researches, mainly on the basis of the well-known Fast-Genetic K-means Algorithm (FGKA) [170] and the Niched Pareto Genetic Algorithm II [110] with the introduction of innovative operators in the evolutionary process. It can achieve a fast convergence to global optimal solution with integration of human domain expert knowledge and preferences.

The framework described in this research is designed to handle multiple and conflicting objectives optimization in real world situation. The framework was first applied to solve clustering problems for validity and performance experiments. Clustering itself is multi-objective optimization problem as discussed in Chapter two. Unlike the other common clustering methods that use a fixed threshold value and\or a prior specified fixed number of clusters, this framework proposed in this research doesn't require the prior knowledge of the datasets. It finds the optimal number of clusters which is a set constituting Pareto optimal solution. It proves that there is better number of clusters are superior to the generated Pareto-optimal solutions. This idea differs from traditional multi-objective algorithms that scale the objectives by assigning subjective weights to each objective function. Hence, weights are not used and assigned to each objective function in the system. Furthermore, the scalable design of this framework provides the divide and conquer concept. It can partition the large datasets into subsets to improve the performance of the computation where each subset is manageable. The clustering results produced by this framework are then validated with some well-known benchmark experimental datasets and

compared with other MOGAs. After the validation, the framework is further applied in two case studies to solve the real world multi-objective optimization problems.

3.1 Framework Description

The proposed Multi-Objective Hybrid K-means Genetic Algorithm (MOHKGA) with expert module is described below. The framework includes the following modules.

1. A novel K-Mean operator module to address the population size problem.
2. A novel parallel approach to increase the GA performance and diversity of population to achieve global optimality.
3. An innovative expert module to improve the convergence process. It provides an interactive tool for decision makers to incorporate their knowledge and expertise in the frames to improve the performance and visualized the result sets.
4. A data visualization module at the end of GA process for decision making to visualize the Pareto-optimal solutions.

The proposed hybrid framework with expert system is scalable and flexible, therefore, it is applicable to various multi-objective optimization problems. The K-Mean operator module can locally improve diversity of selected individuals and reach the global optimality. In real world application, the expert system can provide dynamic decision variable inputs for preferred termination and constraint handling.

This section is organized as follows. The architectural design of MOHKGA is discussed in Section 3.2. The chromosome representation process in MOHKGA is introduced in Section 3.3. The uniqueness of the framework is discussed in Section 3.4. Section 3.5 discusses the

experiments of framework application on some well-known datasets to prove the validation of the framework. Section 3.6 discussed two case studies where the framework was applied to solve two real world problems.

3.2 Framework Design Flow

The design of Multi-Objective Hybrid K-means Genetic Algorithm (MOHKGA) is shown in Fig. 3.1. The standard genetic algorithm operators, including initialization, fitness assignment, selection and mutation are integrated with the unique K-mean operator and the expert module. The novel algorithm used in fitness assignment, parallel processing of selection, crossover and mutation are described in details in each section below.

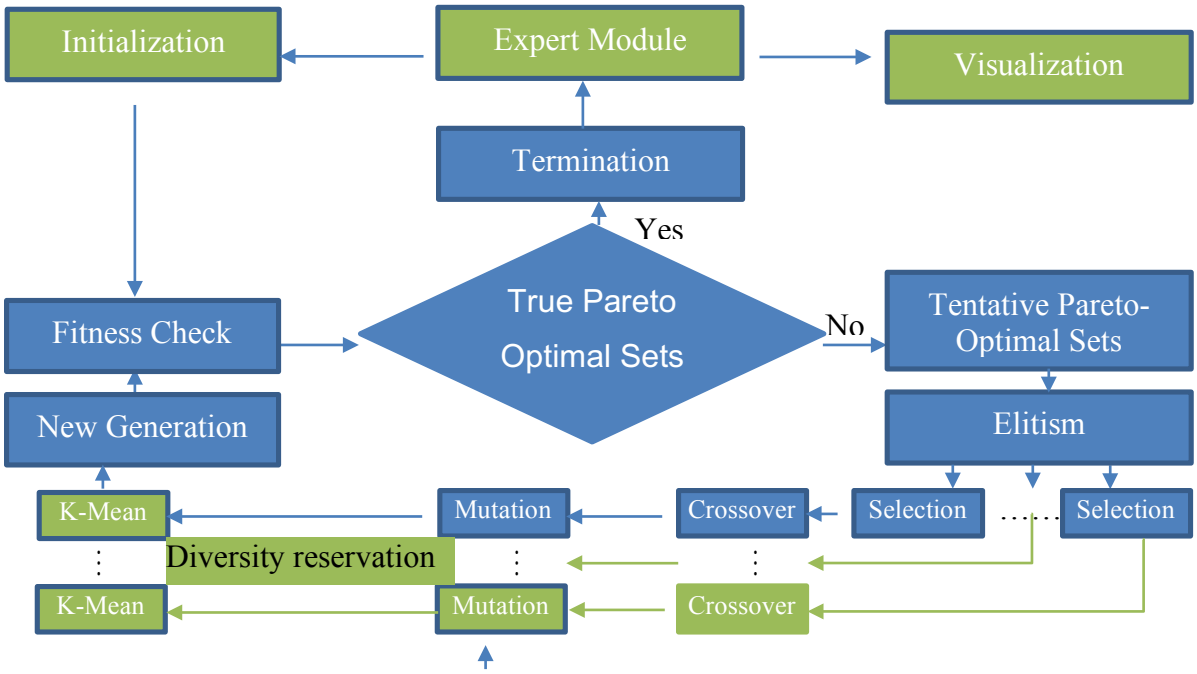


Figure 3.1 MOHKGA with Expert Module

3.3 Genetic Algorithm Operators

3.3.1 Initialization and Problems Encoding

The solutions to the multi-objective optimization problems are encoded in a data structure than can represent the potential solutions to the problem. The genetic representations with chromosome encoding has fixed size that are corresponding to the objectives. This also facilitates crossover operation in the next step. The standard representation of each candidate solution is encoded as array of string in this research. Arrays of other types and structures can be used in essentially the same way for chromosome encoding.

Initialization of MOHKGA

Step 1: *Set the initial and boundary parameters with input from expert module*

Step 2: *Set the initial population $P_0 = \{\varnothing\}$ and i=0*

Step 3*: For each i= 1 to N*

Set $P_0 = P_0 + I$ where i is selected based probability P(x) in 3.3.2

3.3.2 Selection

In this research, the modified Niched Pareto tournament selection scheme is referred for the selection process in the multi-objective genetic algorithm. The selection probability of each solution is defined by the roulette wheel selection using the linear scaling:

$$P(x) = f(x) - f_{min}(n) \,/(\sum_{x=1}^{n}(f(x) - f_{min}(n))$$

where $f_{min}(n) = min\{fx|x \in n\}$ *is the fitness value of the worst solution in the current population.*

Selection of MOHKGA

Step 1: *Two candidates for selection are picked randomly from the population,* C_1, C_2

Step 2: *Set domination flag*

D_1 *of* C_1 = *false;*

D_2 *of* C_2 = *false*

Step 3*: For each of the candidate* C_i *in* C_k

For each individual i in the comparison set

If the candidate i is dominated by the comparison set,

then delete i

If both candidates are non-dominated, they will be kept in the population.

Step 4*: Select the population for the next generation:*

Set the temporary population P'' =0. For i=1,..... n do

Select one individual $i \in Pt$*, based on the given scheme and fitness Value F(i)*

Set P'' = P' +{i}

The selection design and implementation are different from the original Niched Pareto Tournament Selection. In NSGA selection process, if neither the two candidates are dominated by the comparison set, a winner will be chosen based on the fitness sharing. In MOHKGA, the selection doesn't choose a winner. If the candidates in the comparison sets are not dominated, then they are both kept in the population for the next generation, as shown in Figure 3.2. This can result in the number of population grow and cause problem for crossover operation. K-means operator is applied to solve this problem, which is discussed in the K-means operator section. During the selection process, the ranking approaches can be directly used to assign fitness values to individual solutions, they can also be combined with other fitness sharing techniques to achieve the second goal in multi-objective optimization, finding a diverse and uniform Pareto front.

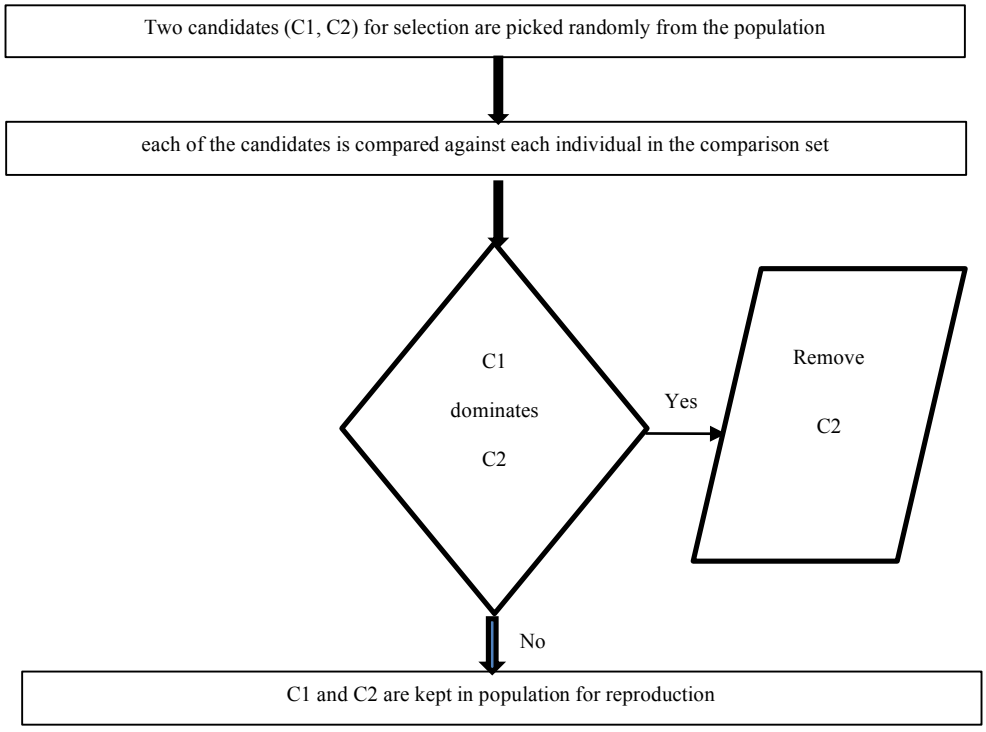


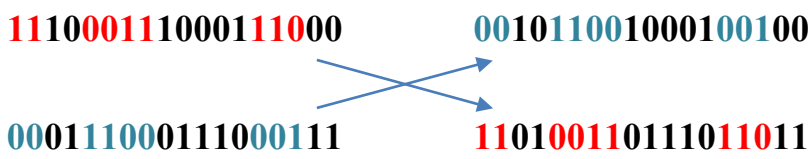
Figure 3.2 Pareto Tournament Selection of MOHKGA

3.3.3 Recombination (Crossover)

The purpose of the crossover operator is to pick two or more solutions (parents) randomly from the mating pool from the previous step. Crossover creates one or more offspring solutions by exchanging information among the parent solutions. The crossover operator is applied with a crossover probability $P_c \in [0,1]$, indicating the proportion of population members participating in the crossover operation. Two types of cross-over, point based and arithmetic cross-over are proposed in this method. It combines corresponding individuals within the two populations to produce a new individual and all the new individuals constitute a new population to be tested whether it fits in the whole population better than the existing individual. The crossover operation is carried out on the population with crossover rate P_c.

The probability of crossover operation is pre-defined in the initialization step. The probability value needs to be carefully set because the high crossover rate will result in the slow convergence time, extreme high rate can lead to the non-stop state of the process, whereas the low rate can lead the premature convergence to local optimality.

Two-point cross-over are used in this research as illustrated below.



The implementation of crossover in the framework is described below:

Crossover of MOHKGA

Step 1: *Set P'' = ∅*

Step 2: *For each j in i – 1, …., N/2*

Step 3:*Choose 2 individuals i, j ∈ P'*

Step 4: *Remove i,j from p'*

Step 5: *Crossover i and j with resultant children k, l ∈ I with probability Pc*

Step 6: *Add k,l to P''. Otherwise add i, j to P''*

3.3.4 Mutation

The mutation operator makes the individual a_n is replaced by a_n' within itself according to the probability rate of mutation. The mutation probability is usually defined as $P_i = 1/N$, which represents the probability interval of a mutating individual *i*.

Mutation rate plays an important role in the genetic algorithm to keep the diversity of population generation by generation. The principle of setting the mutation rate is similar as crossover rate. Low mutation rate can lead to the loss of population diversity and local optimality. High mutation rate can lead to the difficulty of convergence due to the randomness of new individuals generated by high mutation rate.

Two-point cross-over are used in this research as illustrated below.

1110001110001110000 ⟶ **0010110010001001000**

Mutation of MOHKGA

Step 1: *Set P''' = ∅*

Step 2: *For each individual i in P"*

Step 3: *Mutate i with regard to mutation rate Pm*

Step 4: *individual i becomes individual j ∈ current generation*

Step 5: *Set P''' = P''' + { j }*

3.3.5 K-Mean Operator

K-mean operator proposed in this framework is unique and critical. Because the selection operator does not choose of the winner from every comparison set, instead, all the non-dominate individuals are kept for the next generation. Hence a large number of candidate individuals for next generation are generated. The large number of individuals can not only cause high computation complexity and slow convergence of global optimal Pareto set, but also the make the next generation operation impossible as the size of the population to be crossed over does not match. Therefore, clustering and re-group of candidate solutions into smaller size without losing elitism is necessary. The distribution of populations from every generation can be formed into clusters. The density and shape of clusters have direct impacts on the quality of final global Pareto-optimal solution set. The distribution of the tentative Pareto set solutions achieved so far is an important aspect. In the case the trade-off surface is continuous or contains many points, clustering approach can be applied directly to further reduce the number of solutions.

The K-means operator is applied to re-analyze the candidate solutions through clustering. As discussed in the Chapter two, K-mean operator calculates the centroid for each cluster and re-assigns each solution to the closest cluster. In other words, applying K-means helps in quickly rectifying any unwanted outcome from the crossover operator and reduce the size of tentative Pareto-optimal sets. Additionally, the K-means operator can speed up the convergence process by clustering with the reduced population size.

The design and implementation of K-Means operator is described below:

Step 1: *For each individual in the current population*

Step 2: *Get the centroid for each cluster*

Step 3: *for each data point in an individual*

Step 4: *Get the Euclidean distance from the data point to each centroid*

Step 5: *If a closer centroid is found*

Step 6: *Assign the data point to that cluster*

The algorithm of non-dominated set K-mean clustering is described below, and illustrated in Figure 3.3

Input: *P' = external set*

N' = maximum size of external set

Output: *P'' = updated external set*

Algorithm:

Step 1: *Initialize cluster sect C*

Each individual i ∈P' constitutes a distinct cluster C = $\cup_{i \in P}, \{\{i\}\}$

Step 2: *if |C| <= N', goto Step 5*

else goto Step 3.

Step 3: *Calculate the distance D_c of all possible pairs of clusters.*

$D_c = \frac{1}{|c1||c2|} \sum_{i1 \in c1, i2 \in c2} d(i1, i2)$

The distance in objective space is used.

Step 4: *Find minimum dc of cluster c1 and cluster c2*

And the chosen clusters group into a larger cluster C.

C = {c1,c2}$\cup \{c1 U c2\}$

Go to Step 2.

Step 5*: Per cluster, select a representative individual*

and remove all other individual form the cluster.

The centroid is the representative individual.

The representative of the clusters is the reduced non-dominated set P''

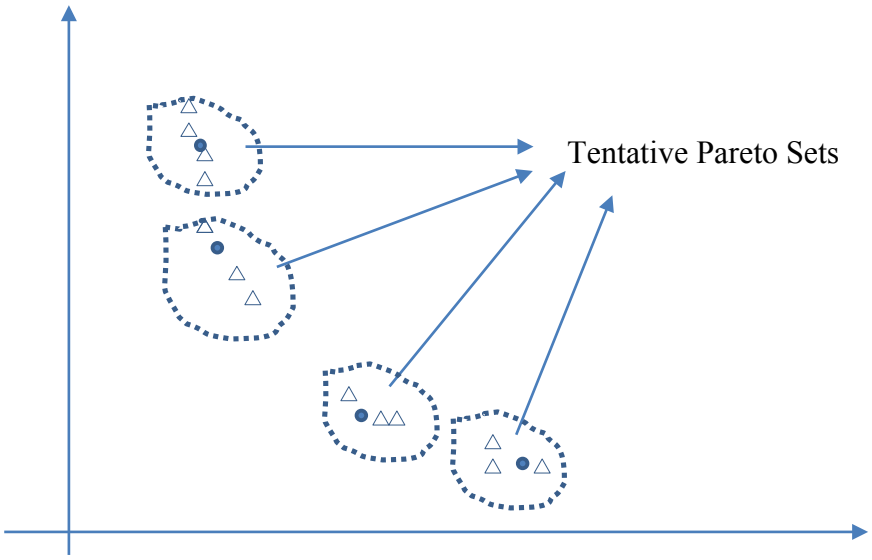


Figure 3.3 K-Mean Clustering of Tentative Pareto optimal solutions

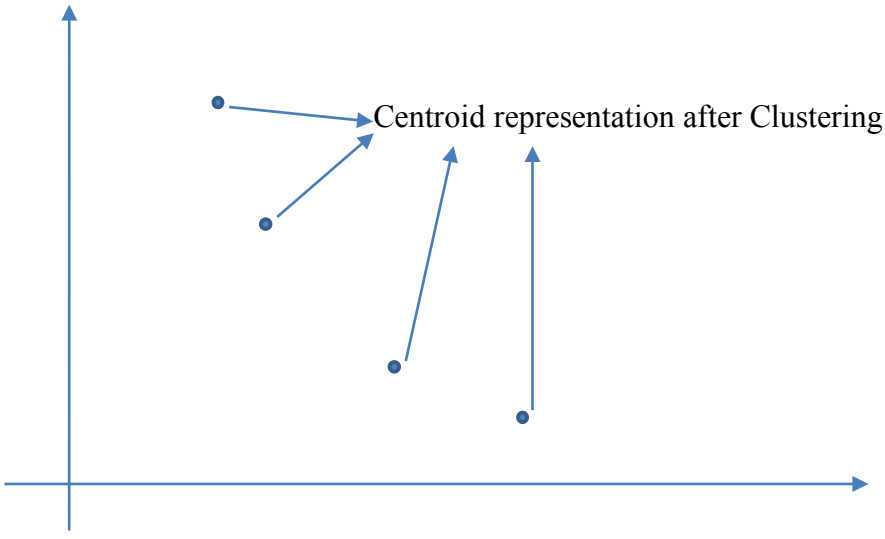


Figure 3.4 Centroids Representation of Pareto-optimal sets

3.3.6 Termination

The program terminates when one of the following conditions applies:

- It converges and generate the final Pareto-optimal sets. No-more better solutions can be generated based on the objective functions
- The predefined stopping criteria are met, including:
    - The execution time
    - The number of generations
- Manual intervention through the expert module. Users can choose the stop the execution whenever necessary.

3.4 Uniqueness of MOHKGA

In addition to use the outcome from the previous researches, MOHKGA incorporates some innovative ideas and algorithm in the algorithm. The implementation is discussed in the following sections.

3.4.1 Reducing the Non-dominate Set by K-Mean Clustering

In real-world multi-objective optimization, the number of the tentative Pareto-optimal solution set in the genetic process and the final Pareto-optimal sets can be very large, this is especially true when the selection operator keep all the non-dominate solutions for the next generation. As discussed in the methodology, the unique introduction of K-mean operator to cluster, re-group and reduce the size of populations can solve the aforementioned problems, improve the performance and also keep the population diversity. The application of K-mean cluster on the final Pareto-optimal set can make the result visualization easy in the expert module so that they can be used as recommendations for decision making.

3.4.2 Termination with Expert customized local search module

Many other genetic algorithms referred in the literature are terminated after a prefixed number of generations or when no better new individuals have entered the non-dominated set after a prefixed number of generations. The converge process of global optimality are usually slow when the problem itself is complex and hence requires high computational costs. The expert module proposed in this framework can speed up the converge process and using domain subject expert's input as input parameters to terminate the GA process.

For a large number of solution set, the computation time can be too long to be used for real-world decision-making process, therefore, the customized initialization and termination condition setup can help reduce the computation time.

3.4.3 Visualization and Reporting module

Effective data visualization is an important tool in the decision-making process. It allows decision makers to quickly examine large amounts of data, identify trends and issues efficiently, exchange ideas with key players, and influence the decisions that will ultimately lead to success.

After finding a set of representative Pareto-optimal solutions and the reduced number of solution sets, the expert module provides the result with the visualized results for decision makers. Presenting and visualizing the preferred recommendations form the obtained Pareto set is important on solving real-world practical problems during the decision-making process.

3.4.4 Integration with Subject Domain Expertise

A decision maker's preference information, if there are any, can be integrated in interactive approach during the search process. In the interactive approach, decision maker's preference information can be configured as input parameter for each generation and is presented with

solutions chosen from the current non-dominate front. The decision maker can rank the solutions, taking trade-offs according to their preferences.

3.5 MOHKGA Validation and Comparison with Other MOGAs

To prove the validity of the proposed framework, several well-known experimental datasets are used for data clustering analyses. The clustering results were analyzed by using six of the cluster validity techniques proposed in the literature, Silhoutte, C index, Dunn's index, SD index, DB index, and S_Dbw index, and then compared with the published result from other researches from the literature.

Different cluster validity indexes aforementioned are used to validate the result. As discussed in the literature, minimal SD index indicates an optimal cluster number, while maximal Dunn index shows the optimal number of clusters as it means maximal inter-cluster distances and minimal the intra-cluster distances, i.e., good separation of clusters. DB index is a function of the ratio of the sum of within-cluster scattering to between clusters separation, a small value exhibits a good clustering. Silhouette value is in the interval [–1, 1], it's value that is close to 1 means that the sample is assigned to a very appropriate cluster, whereas 0 means that the sample lies equally far away from both clusters while close to –1 means that the sample is misclassified.

3.5.1 Objectives Definition of Clustering

As discussed in Chapter two, clustering itself is A multi-objective optimization problems, and is generalized blow:

Objectives:

Minimize $\{f_1(x), f_2(x) \ldots\ldots, f_k(x)\}$

Where :

1. *with k is the number of objectives and K≥2.*
2. $f(x) = (f_1(x), f_1(x), \ldots f_k(x))^T$ *is the objective vector*
3. $x = (x_1, x_2, \ldots x_n)^T$ *is the decision vectors belong to the search space S with constraint.*

For clustering process, two conflicting objective functions are defined:

- Min (f_1), minimizing the cluster partitioning error with minimal Total Within-Cluster Variation (TWCV), i.e. to maintain the clear separation of clusters with minimal number of clusters [130].

$$TWCV = \sum_{n=1}^{N}\sum_{d=1}^{D} X_{nd}^2 - \sum_{k=1}^{K}\frac{1}{Z_k}\sum_{d=1}^{D} SF_{kd}^2$$

  where $X_1, X_2, \ldots, X_N$ are the N objects, X_{nd} denotes feature d of pattern X_n (n = 1 to N), Z_k denotes the number of patterns in cluster k, and SF_{kd} is the sum of the d-th features of all the patterns in cluster k: [130]

$$SF_{kd} = \sum_{x_{n \in G_k}} X_{nd} \ (d = 1,2, \ldots . D)$$

- Max(f_2), maximize the separateness of the clusters.

3.5.2 Experimental Datasets

The multi-objective genetic algorithm-based approach proposed in this research are tested with various initial and boundary environmental conditions. The validity of MOHKGA was

tested on the following well-known experimental datasets for clustering. The result Pareto-optimal front gives the optimal number of clusters. The results are then compared with the known result and other MOGAs in the literature for parallel analyses.

3.5.2.1 IRIS Datasets

The Iris dataset is a well-known dataset widely used in pattern recognition and clustering. It is a four-attributes dataset containing 150 instances; it has two or three clusters each has 50 instances. One cluster is linearly separable from the other two and the latter two are not exactly linearly separable from each other. Jiang *et al* [15] applied visual rendering to the Iris dataset by using a linear and reliable mapping model to visualize the k-dimensional dataset in a 2D star-coordinate space.

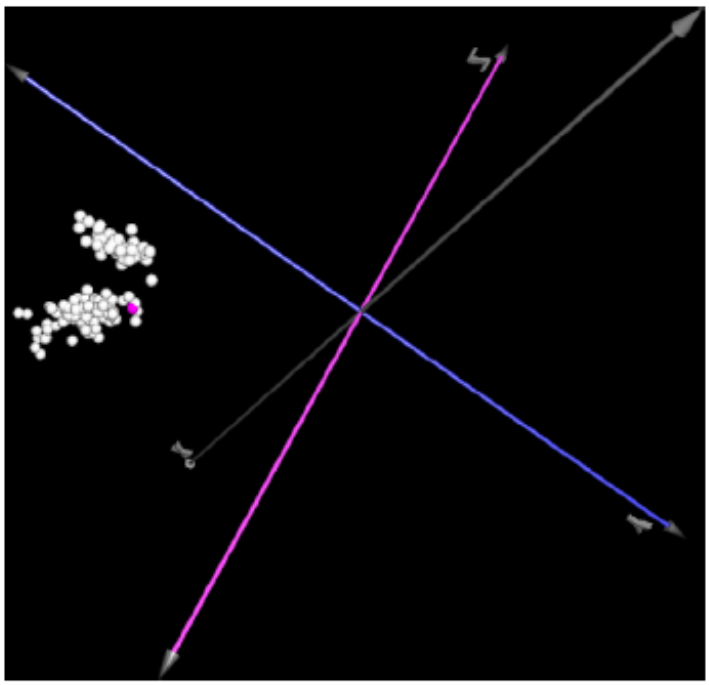


Figure 3.5 Cluster distribution of the original Iris dataset

Cole [13] also conducted tests on the Iris dataset using general genetic algorithms with the main parameters:

- *number of iterations = 1000,*
- *range of exponential mutation rate = from 10.0 to 0.000001,*
- *population size = 50,*
- *crossover probability = 1.00*

The optimal number of clusters obtained are 3 for the Davies Bouldin method and 2 for the Calinski and Harabase method.

MOHKGA experiment with this dataset with the following parameters are setup:

Parameters	**Value**
population size	*200*
number of comparison set	*20*
crossover rate	*0.8*
mutation rate	*0.01*
*Threshold**	*0.0001*

Table 3.1 Initial Setup Parameters for Iris Dataset

** was used to check if the population stops evolution after 50 generations or if the process needs to be stopped.*

Average changes in the Pareto-optimal front by running the proposed algorithm for the Iris dataset are displayed in Fig. 3.6 for different generations. It demonstrates that the system can quickly converges to an optimal Pareto-optimal front. some key TWVC values are reported in Fig.3.7 and Table 3.2 contains the TWVC values and index values.

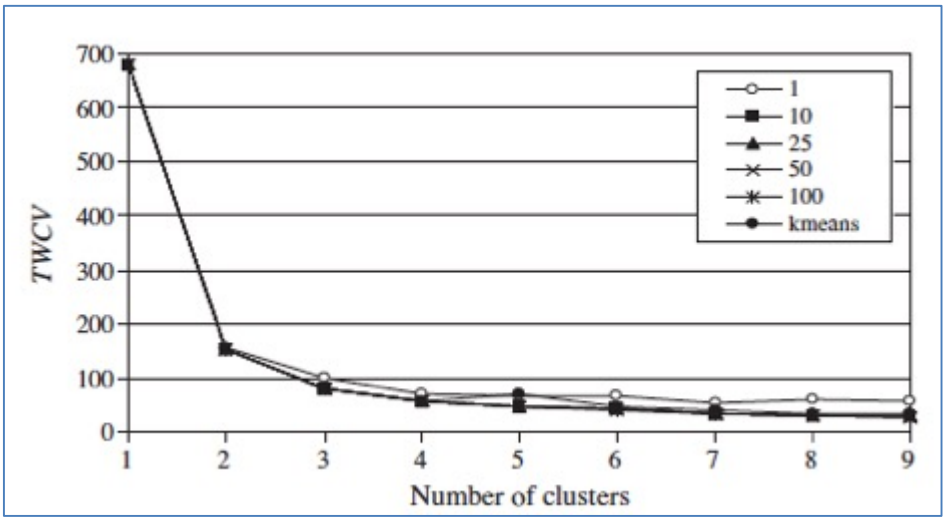


Figure 3.6 Pareto-optimal front using Iris dataset

The intermediate values during the genetic process is listed in Table 3.2. It can be observed that the program converges well towards to the true Pareto-optimal number. Figure 3.7 shows the average values of various clustering index for the comparison purpose over ten runs. Six indices were used to analyzed the output results of the Iris dataset. The results obtained are compared with the corresponding results reported by the other researchers [13, 15,16].

Generations	TWCV(k=3)	TWCV(k=8)
1	65.9482	57.2637
10	41.7086	29.2056
25	41.7086	28.3555
50	41.7083	28.1758
100	39.043	28.1758
k-means	45.5185	34.1203

Table 3.2. IRIS dataset TWCV for k = {3,8}

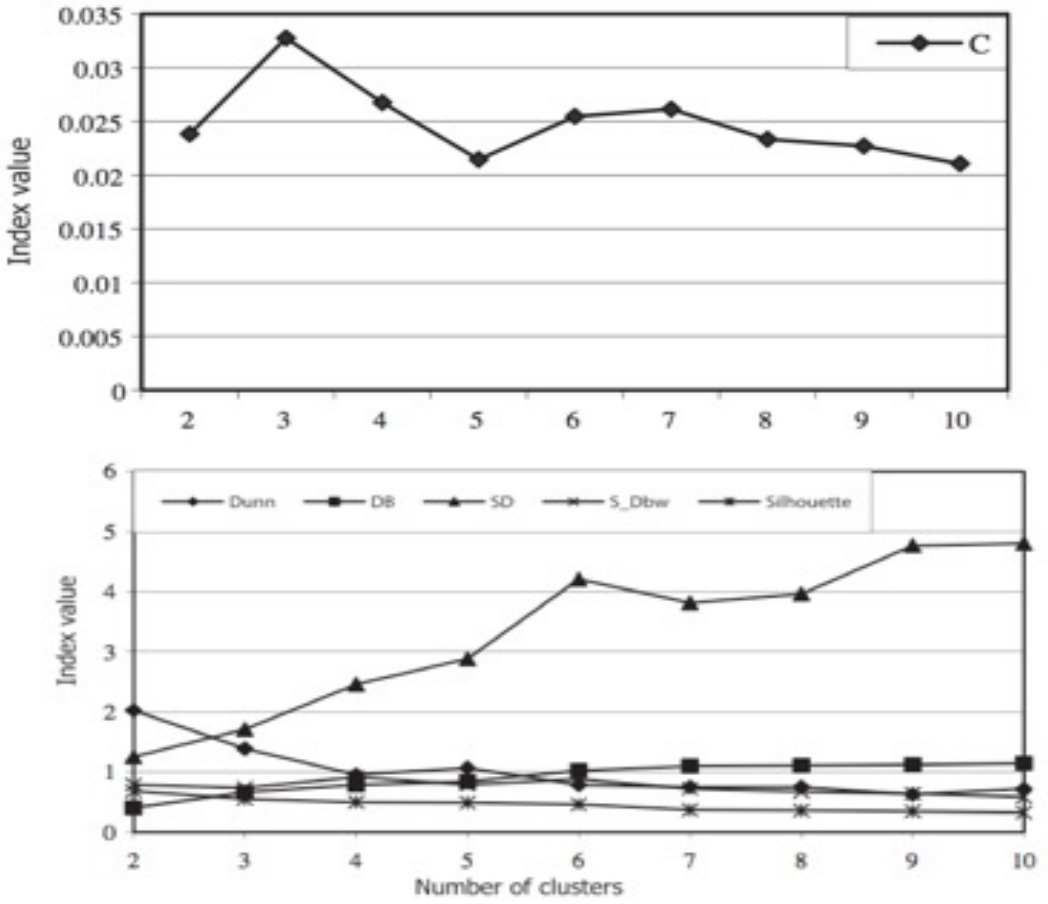


Figure 3.7 Iris dataset cluster validity C, Dunn, DB, SD, S_Dbw and Silhouette indices.

Validity Index	Best Index Value	Number of Clusters
Dunn	1	4
DB	0.5	2
SD	1.2	2
S_Dbw	1.8	3
Sil.	1.8	3

Table 3.3. Best Validity Index Value with Cluster number

As discussed in Chapter 2, SD validity index definition demonstrates the average scattering for clusters and total separation between clusters. A small value for SD index indicates compact clusters. The number of clusters that minimizes the index is an optimal value. This is in agreement with the expected result shown the Fig.3.7. The number of cluster is two with the minimum SD index value. Previous research [13] suggested the optimal number of clusters for the Iris data is 3, which ranks second for all the indexes except S-Dbw and C. This finding is consistent with the result of the DB cluster validity index published by Cole [13]. Small DB index value indicates a good clustering, but this index alone is not representative to show the optimal number of clusters. The six indices altogether indicate the optimal number of clusters for IRIS dataset, which includes properly combined compactness and separation. Clusters are more compact but less separate from each other for the number of clusters taken as 3, while clusters with number of clusters taken as 2 are better separated.

3.5.2.2 Breast Cancer Datasets

Breast cancer is known to be a heterogeneous class of cancer. Data classification and clustering of genes/tumors expression data is generally unstable. Tow datasets which are freely available at http://www.ncbi.nlm.nih.gov/geo/, GSE12093 and GSE9195 are selected to test the performance and accuracy of the framework since they are known to be a heterogeneous nature and they are well studied from the previous researches.

3.5.2.2.1 GSE12093 Dataset

The GSE12093 dataset has 76-gene signatures defined high-risk patients that benefit from adjuvant tamoxifen therapy, from 136 breast cancer samples that were treated with tamoxifen. It contains 22284 genes with 136 attributes/features. filtering standard of more than 200% coefficient of variation are used to reduce the data size and the distribution of this dataset is not sensitive to standard deviation or other filtering criteria.

For experimental purpose, the initial input parameters are shown in table 3.4.

Parameter	Value
Size of initial population size	700
The number of comparison set	10
Crossover rate	0.8
Mutation	0.01
Termination Threshold	0.1

Table 3.4 Initial Setup Parameters for GSE12093

The generated Pareto-optimal front on the GSE12093 dataset are displayed in Figure 3.4 with number of generations. We can observe the following from the algorithm output as shown in Table 3.5.

Generations	c1	c2	c3	c4	c5	c6
1	1.49E+12	1.79E+11	1.79E+11	1.79E+11	1.67E+11	2.86E+11
100	1.49E+12	1.79E+11	1.79E+11	1.79E+11	1.67E+11	1.33E+11
200	1.49E+12	1.79E+11	1.79E+11	1.79E+11	1.61E+11	1.30E+11
300	1.49E+12	1.79E+11	1.79E+11	1.78E+11	1.56E+11	1.27E+11
400	1.49E+12	1.52E+11	1.52E+11	1.75E+11	1.37E+11	1.23E+11
500	1.49E+12	1.33E+11	1.33E+11	1.61E+11	1.41E+11	1.20E+11
600	1.49E+12	1.09E+11	1.09E+11	1.63E+11	1.45E+11	1.15E+11
700	1.49E+12	1.48E+11	1.48E+11	1.48E+11	1.24E+11	1.09E+11

Table 3.5 TWCV with Corresponding Generations and Cluster number for GSE12093

Figure 3.8 shows that the algorithm converges quickly with the Pareto-optimal front when the number of cluster stabilizes at two or three. There is not much improvement with the increased number of generations. The actual change in the value of TWVC is not significant as shown in Figure 3.8 where the values are very close. Therefore, the true Pareto front can be considered as much smaller optimal solution sets, other than a large solution sets. As shown in Fig.3.8, the optimal number is two or three and these two solutions dominate the others.

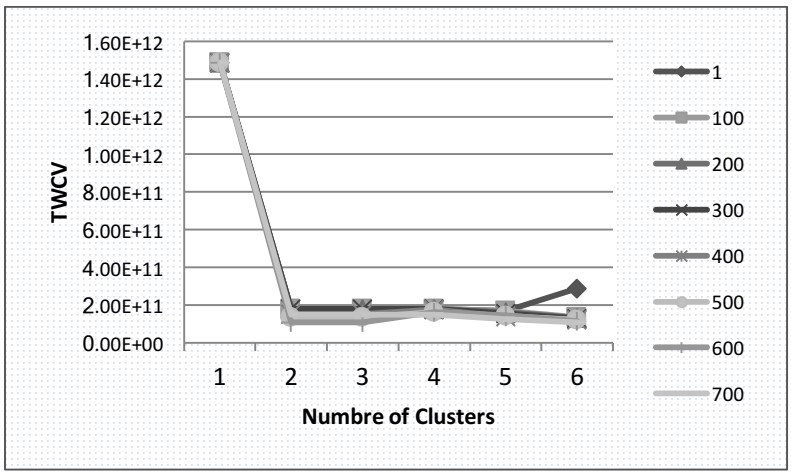


Figure 3.8 Pareto-fronts for GSE12093 dataset

Cluster validity are analyzed on the filtered GS12093 datasets to compare the results of our experiments. Three indices from internal measures (connectivity, Dunn and Silhouette index) and four from stability measures (Average proportion of non-overlap (APN), Average distance (AD), Average distance between means (ADM) and Figure of merit (FOM)) are used for the validation purpose. The test results are reported in Figure 3.9 and Figure 3.10 for internal measures indices and stability measures indices, respectively.

The minimum connectivity value indicates the optimal number of clusters of two, as shown in Fig.3.9. This is consistent with convergence of Pareto-optimal front. The variation of Dunn and Silhouette index are not significant in this case.

The higher value of AD usually indicates the optimal number of clusters, which is two. This is in consistent with the optimal number of clusters shown in Fig 3.8. All three internal measures indices and two stability measures indices show the same results, with similar trend.

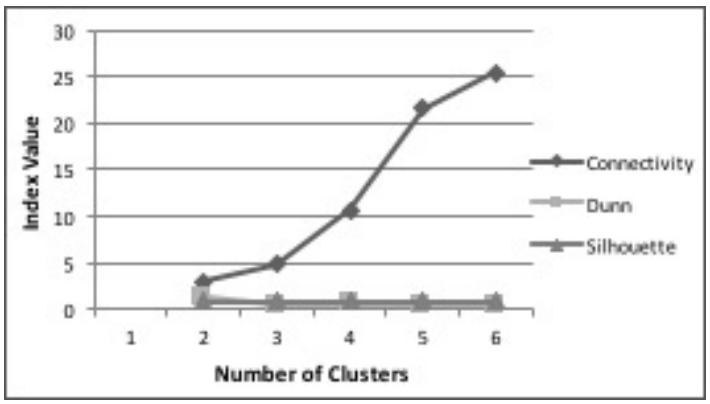


Figure 3.9 GSE12093 dataset cluster validity results using Connectivity, Dunn and Silhouette indices

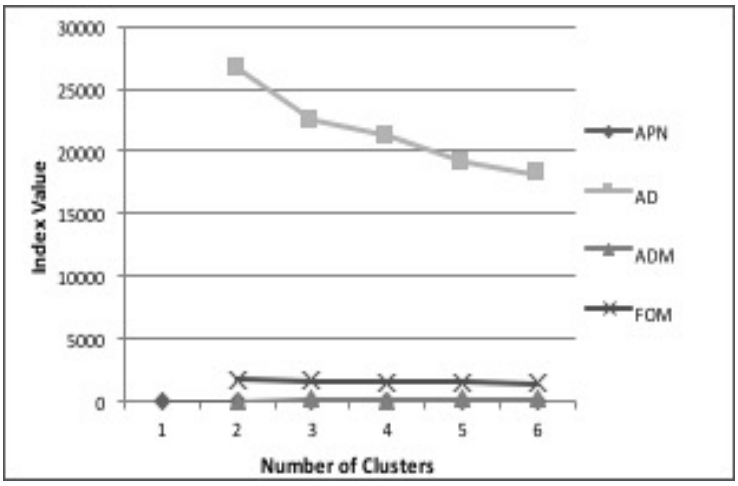


Figure 3.10 GSE12093 dataset cluster validity results using stability measures

3.5.2.2.2 GSE9195 Dataset

The GSE9195 dataset contains molecular profiling in estrogen receptor-positive (ER+) breast cancer treated with tamoxifen. Gene expression profiling is used to develop an outcome-based predictor using a training set of 255 ER+ BC samples. The data set contains 5,4675 samples with

77 attributes/features. For this research, the data was filtered with more than 1.6 standard deviation in order to reduce the data size.

The initial setup parameters are shown in Tables 3.6.

Parameter	Value
Size of initial population	150
Number of comparison set	10
Crossover rate	0.8
Mutation	0.01
Termination Threshold	0.1

Table 3.6 Initial Setup Parameters for GSE9195

The generated Pareto-optimal front from this framework on the GSE9195 datasets are displayed in Figure 3.11 for different generations. The actual results of TWVC with the corresponding generations and cluster numbers are shown in Table 3.5.

Generations	c1	c2	c3	c4	c5	c6
1	9.46E+04	8.76E+04	8.76E+04	8.60E+04	85653.4	8.48E+04
100	9.46E+04	8.76E+04	8.42E+04	8.17E+04	81402.5	8.21E+04
200	9.46E+04	8.60E+04	8.42E+04	81070.9	81211.7	8.13E+04
300	9.46E+04	8.60E+04	8.42E+04	80983.7	8.07E+04	8.10E+04
400	9.46E+04	8.78E+04	8.42E+04	8.07E+04	8.00E+04	8.02E+04
500	9.46E+04	8.42E+04	8.42E+04	80713.3	7.94E+04	7.94E+04
600	9.46E+04	8.42E+04	8.42E+04	8.05E+04	7.89E+04	7.82E+04
700	9.46E+04	8.42E+04	8.42E+04	8.05E+04	7.86E+04	7.74E+04

Table 3.7 TWCV with Corresponding Generations and Cluster number for GSE9195 Dataset

The actual change in the value of *TWVC* is not reflected in Figure 4.7 where the values are very close and all the five curves almost overlap due to the scale used.

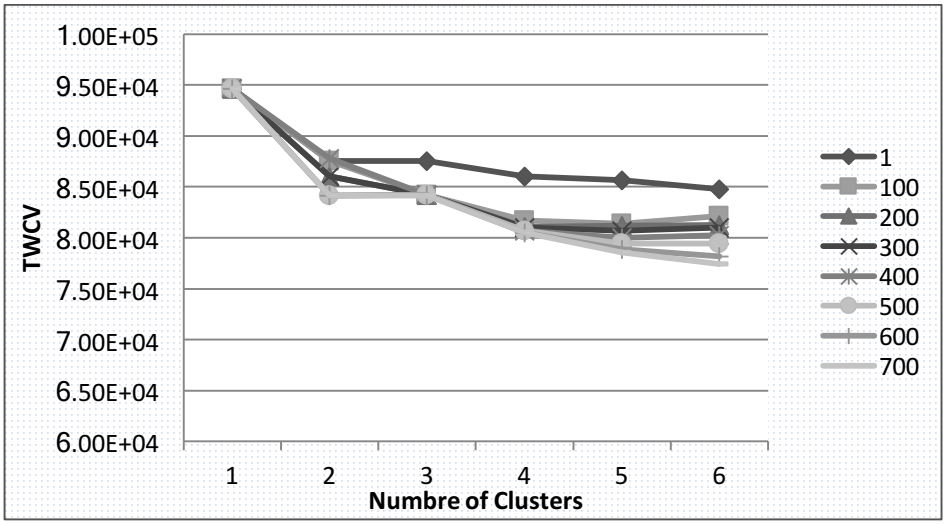


Figure 3.11 Pareto-fronts for GSE9195 dataset

As shown the in the Figure 3.11, with the increased number of generation, a refined Pareto-optimal solution are generated. Unlike the previous dataset GSE12093, a clear Pareto-optimal front can be observed from Figure 3.11.

We run the same validity process for the GSE9195 dataset. However, due to the large variances of the index values, the indices are re-grouped and shown in two figures. Figure 3.12 shows the indices with value between 0 to 6, while Figure 3.13 shows the connectivity and AD indices with larger value.

Fig. 3.11, we can observer a clear Pareto-optimal and it stabilizes when the cluster number four. From validity index as show in Fig. 3.12, ADM and Silhouette index also shows at the cluster number four, the index values start to stabilize to a point that no better value can be achieved. This is in consistent with the Pareto-optimal front in Fig. 3.11, where the Pareto front starts to show less variance on the TWCV values with the number of clusters pass over four.

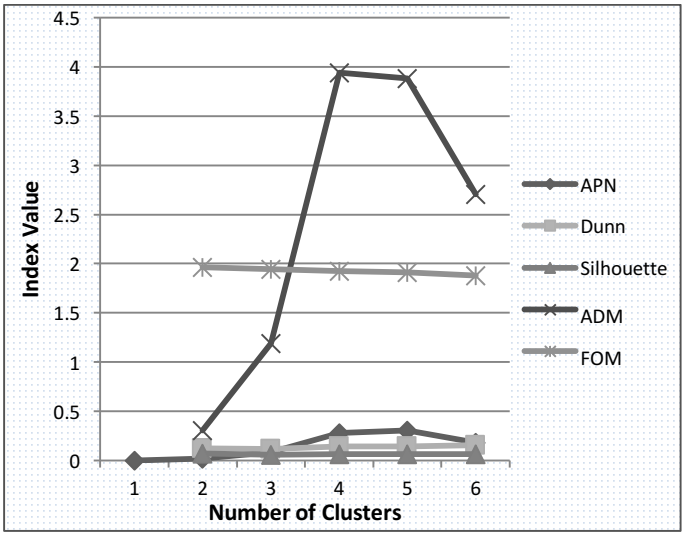


Figure 3.12 dataset cluster validity results APN, Dunn, ADM, FOM and Silhouette indices

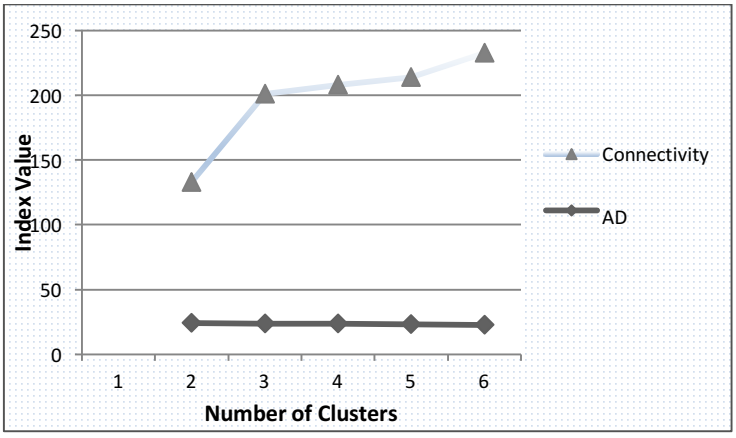


Figure 3.13 GSE9195 dataset cluster validity results using connectivity and AD

Fig. 3.13 shows the connectivity and AD index value with regards to this dataset. index, connectivity index provides a quantified value to show the degree of connectivity of individual clusters, and how well-separated clusters are. It's similar to Dunn's index but with better

indication of separation. In this case, the connectivity index value shows a large increase to at cluster number three, while AD is not sensitive for this dataset.

3.6 Applications of MOHKGA in Real-World MOOP

After the validation of MOHKGA from the experimental data sets, MOHKGA was applied to solve real-world multi-objective optimization problems. The first application was on the blood bank inventory management and optimization for Calgary Blood Services, Alberta, Canada. The outcomes are used to provide recommended solutions for decision maker to manage the blood bank more efficiently and predict the future outages based on simulation results. The implementation details and results are described in section 3.9.1. The second application of MOHKGA is to provide the mobile users a set of optimized shopping options for the products they want to buy, for example, the best prices with shortest travel time and distance, with configurable preference settings. This application and implementation was used for Microsoft Imagine Cup 2013 worldwide competition and won the first category price in Canada. The implementation details and results are described in section 3.9.2.

3.6.1 Case Study: MOHKGA on Blood Bank Management

3.6.1.1 Problem Statement and Formulation

Calgary Health Region (CHR), Alberta, Canada has four referral hospitals and ten community health care centers. All its hospitals and care centers require different level of blood services and Calgary Lab Services (CLS) is one of main authority to manage the blood bank inventory. Blood bank management is commonly known as red cell inventory management.

There are 16 types of red cells. The shelf life or storage time for each type of red cell are different, but the commonly used maximum 42 days are used in this research. In the current

management practice, even with commercial inventory management software and large amount of manpower to manage the Red Cells (RC) inventory, over the past years, the inventory of red cells is generally over-stocked on purpose in order to guarantee sufficient supply for patients use and to prepare for emergency use, as a result, the over-stocked inventory often is wasted due to the expired shelf life of red cells.

Red cells can be treated as a type of scarce, perishable good, the cost of getting them is the same as commercial goods despite some of them are from the donors other than purchase. In order to maintain the sufficient supply of red cell in stock, while keeping the waste to the minimum level, the local health authority need an expert system that can provide recommended solutions for decision making process on the optimal inventory level. More importantly, the system should also be able to predict the sustainability of current inventory level for various situations as input parameters, including flu season, epidemic situations and situations where sudden massive volume of red cells under emergency situation.

The formulation of research is described below. The main objectives in RC management are:

- f_1: Minimize the cost of total waste (TWC) of red cells (RC)

$$Min.\ \mathrm{TWC} = I + \sum_{k=1}^{K} S_k + \sum_{k=1}^{K} R_k - \sum_{k=1}^{K} I_k C_k - \sum_{d=42}^{D} I_d$$

Where:

I: Initial inventory level of all rec cells

R_k: daily replenishment of RC type k

C_k: *daily consumption rate of RC type k*

S_k: *daily savings of RC type k*

- f_2: Maximize the sustainment time with current inventory:

$$Max\ T = F(I, R, C, S)$$

Where $T \in \{t, \ldots \infty\}$ *t is the minimum inventory level for critical usage.*

In order to achieve the objective f_2, i.e., to maintain or sustain the maximum availability of RC supply, RC banks would need to maintain an excessive high level of inventory to guarantee the sufficient supply, however, the high level of inventory will inevitable result in waste and hence, the maximum f_2, which is the opposite of objective f_1.

Hence the two objectives to be optimized are conflicting and well fit the application area of multi-objectives optimization in this study.

3.6.1.2 System Design and Implementation

The system consists of n-tier architectural design as show in Figure. 3.17 below. It is a client–server software architecture pattern including the presentation (user interface), business logics (functional process logic), computer data storage and data access tiers. It was developed by John J. Donovan in Open Environment Corporation (OEC), a tools company he founded in Cambridge, Massachusetts.

The design of three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently in response to changes in requirements or technology. For example, a change of operating system in the presentation tier would only affect the user

interface code. The first tier is the presentation tier including the web graphical user interface including calendar representation of inventory levels as results of the pre-defined initial conditions. This is the top-most level of this application; the main function of the interface is to translate tasks and results from MOHKGA framework to the decision maker with understandable data visualization. The Pareto-optimal results are then encoded in the web services module in standard serialized formats over HTTP, JSON and XML. It is for the web visualization of MOHKGA generated Pareto-set, rendered with HTML5, CSS3 and JavaScript at the time of writing.

The business logic tier consists of MOHKGA framework and the integrated expert module, which generates the Pareto-optimal set based on the user's preferences form presentation tier. This layer coordinates the application, processing commands, perform the logics command, evaluation and calculation defined in MOHKGA.

The backend tier is for the storing the results in the persistent store as an archiving and tracking tool. Users preferences info are stored in this tier as well. The data tier includes the data persistence mechanisms (database servers, file shares, etc.) and the data access layer that encapsulates the persistence mechanisms and exposes the data. It provides an API to the business tier that exposes methods of managing the stored data without exposing or creating dependencies on the data storage mechanisms. Avoiding dependencies on the storage mechanisms allows for updates or changes without the application tier clients being affected by or even aware of the change.

Data security is another important factor that needs to be considered in this case study as the problem domain contains sensitive information. All data, including the input and output of

the frameworks, are encrypted between the communication of the tiers. The encryption within HTTPS is intended to provide benefits like confidentiality, integrity and identity.

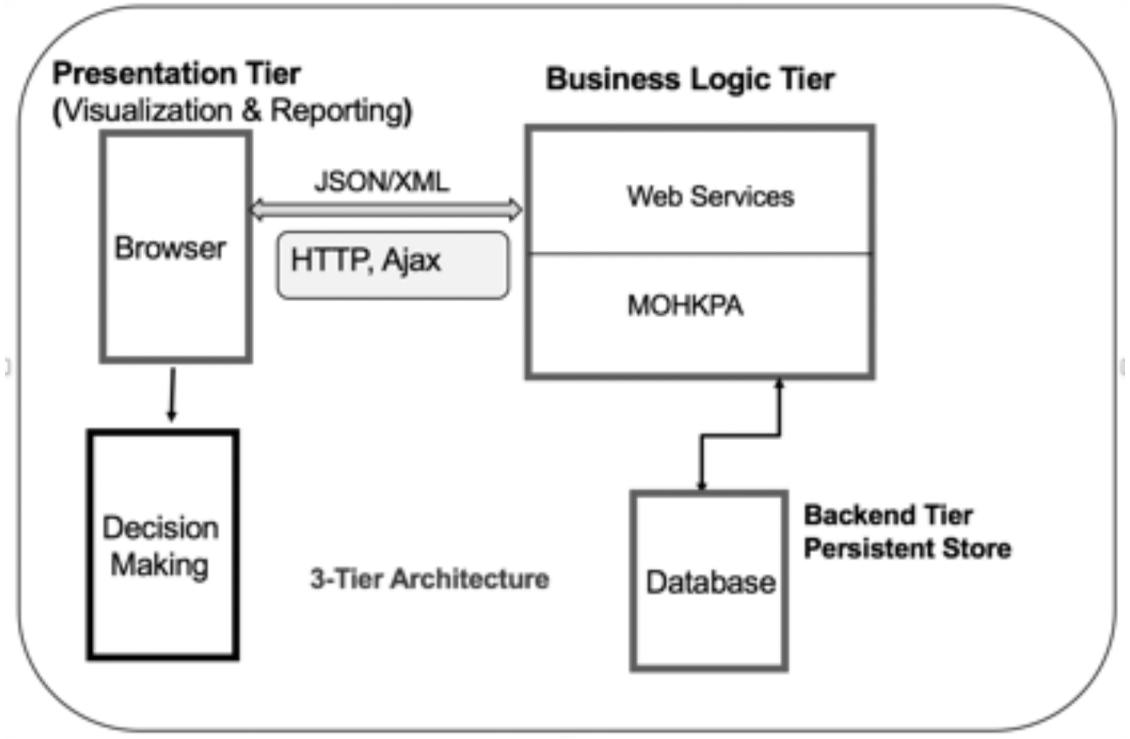


Figure 3.14 MOHKPA System Architectural Design for CLS

3.6.1.3 Assumptions

In this case study, the following assumptions are made: Because of political reasons and regulatory policy, only experimental data are used, including the decision maker's preference setting, initial inventory data, daily replenish and consumption rate, daily waste level etc. These data are not the actual data from Calgary Heath Region.

3.6.1.4 Result Analyses and Benefits

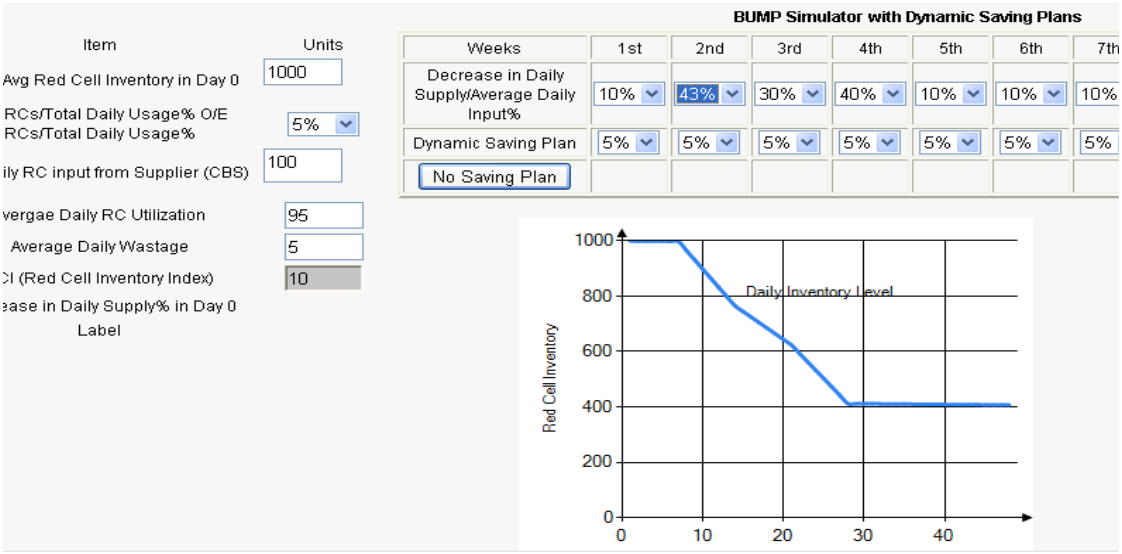


Figure 3.14 Detailed View of RC Daily Inventory Level with Various Settings

The novel expert module in this proposed framework contains an interactive user interface for decision makers or domain subject experts to integrate their experience, knowledge and expertise into the framework. As shown in Fig. 3.14, the following parameters are can be set:

- Initial conditions: The initial inventory level for the 16 different types of red cells, average daily consumption rate at the specific time/season of the year, average daily replenish rate at the specific time/season of the year, expected waste.
- Stop constraints: MOGAs usually can't control the stopping/termination in the search process on DM, as discussed in section 2.6.9. Time is an important factor in Real-world application of MOHKA in MOOPs as decision makers may require the Pareto-optimal solution sets, even may not be so refined, under emergency

situation, especially for the blood services. Decision makers can set the time to stop the program if it runs over the pre-set time threshold.

Figure 3.15, 3.16, 3.17 shows the intermediate output of framework run for generation 1, 100, and 150. It can be observed that the Pareto-Optimal set can be generated at generation 150. Because of the conflicting objectives in this case, the result provides a valuable recommendation to decision makers.

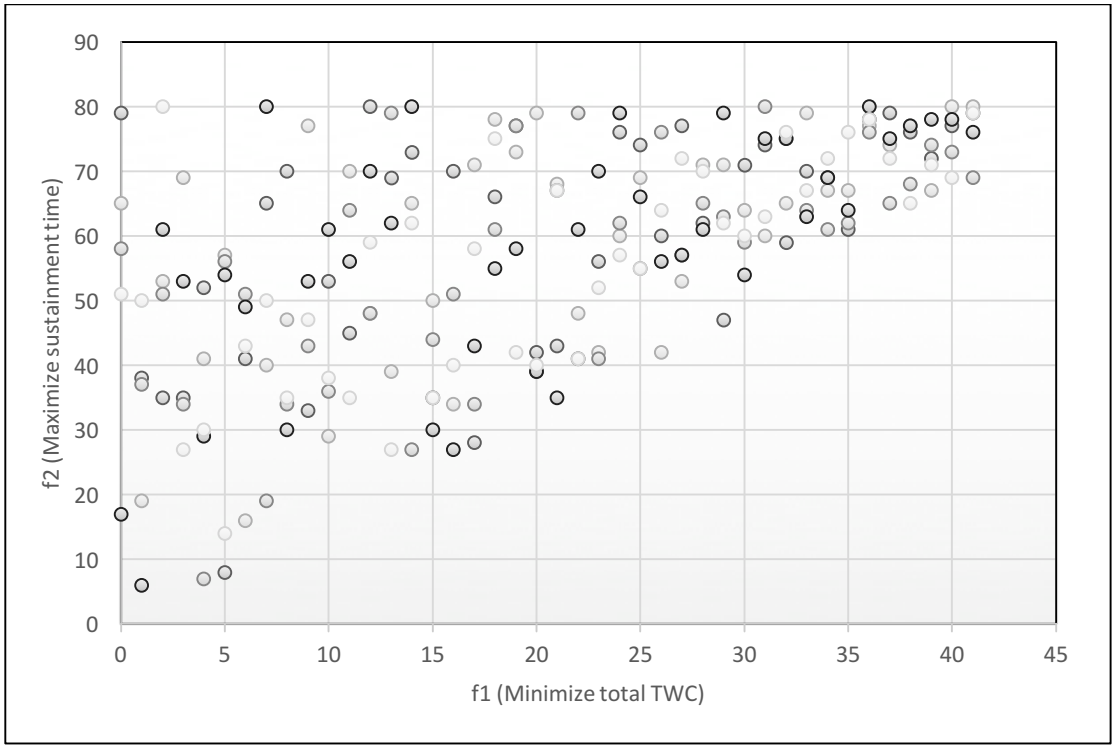


Figure 3.15 Temporary Pareto Solutions at Generation 1

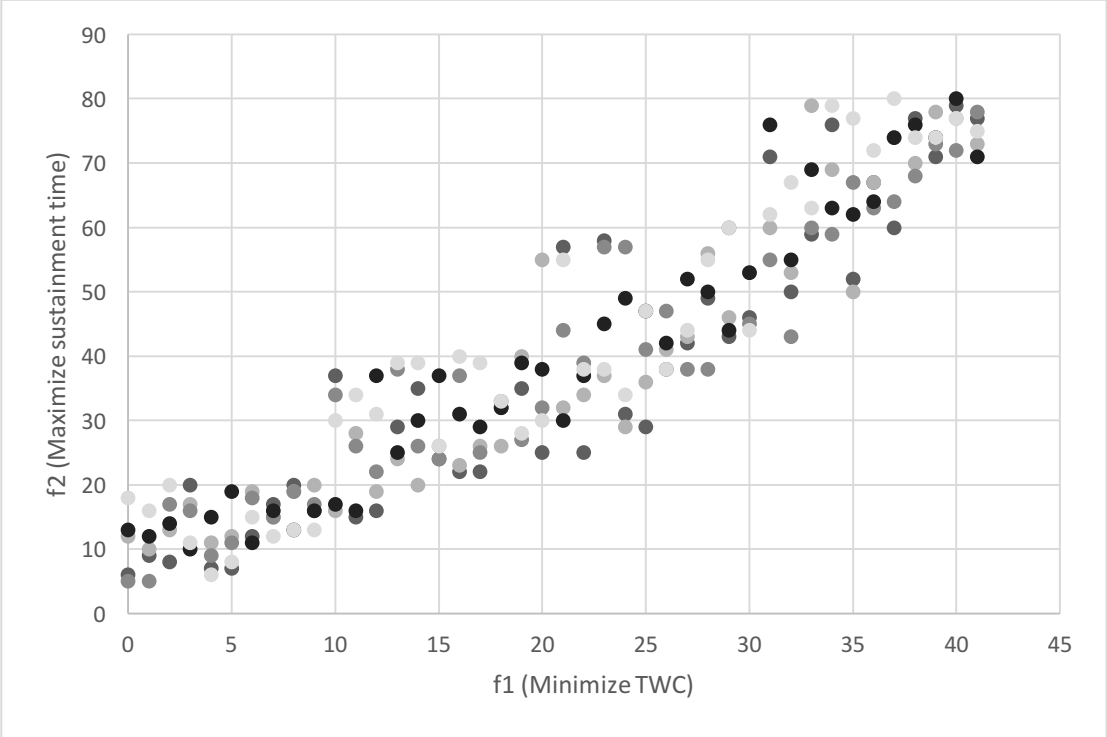


Figure 3.16 Temporary Pareto Solutions at Generation 100

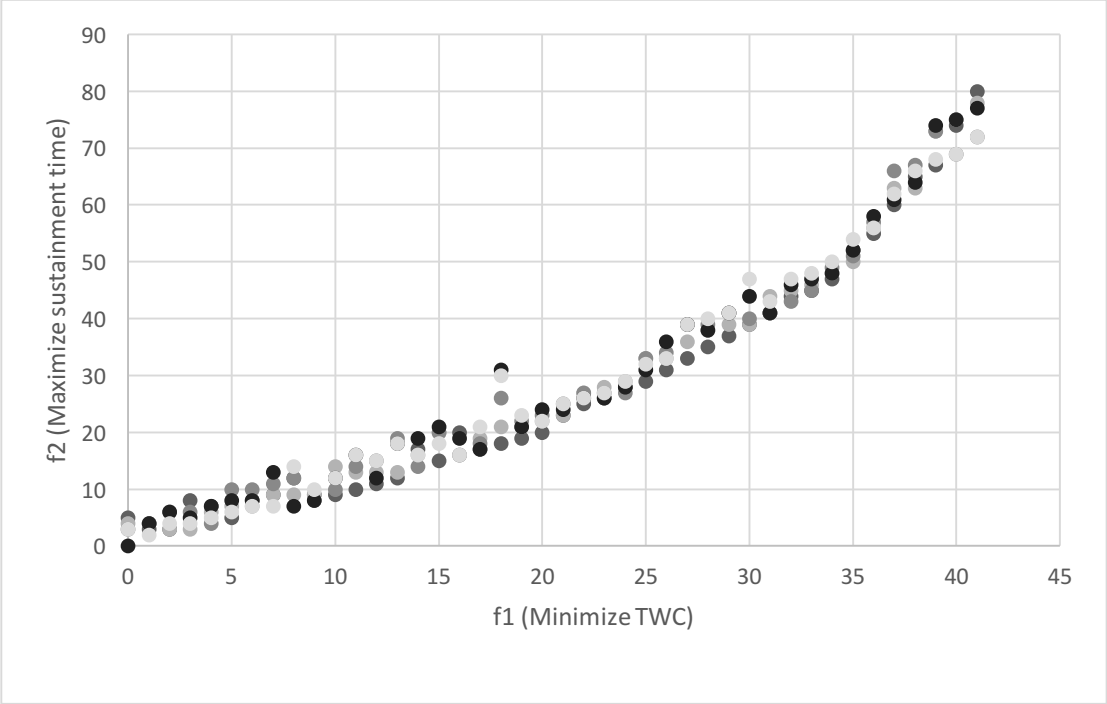


Figure 3.17 True Pareto Optimal Set at Generate 150

3.6.1.5 Result Analyses and Benefits

For practical real-world problem solving, visualization from the framework simulation results are necessary for executive decision makers because not only it can provide user friendly formats, but also for quick decision making where timing is a critical factor, including emergency situations. Furthermore, a web-based visualization provides even more values for decision makers as it can be used anywhere as long as there is an internet connection.

Fig.3.18 is the screen shot of the real-time result from the framework.

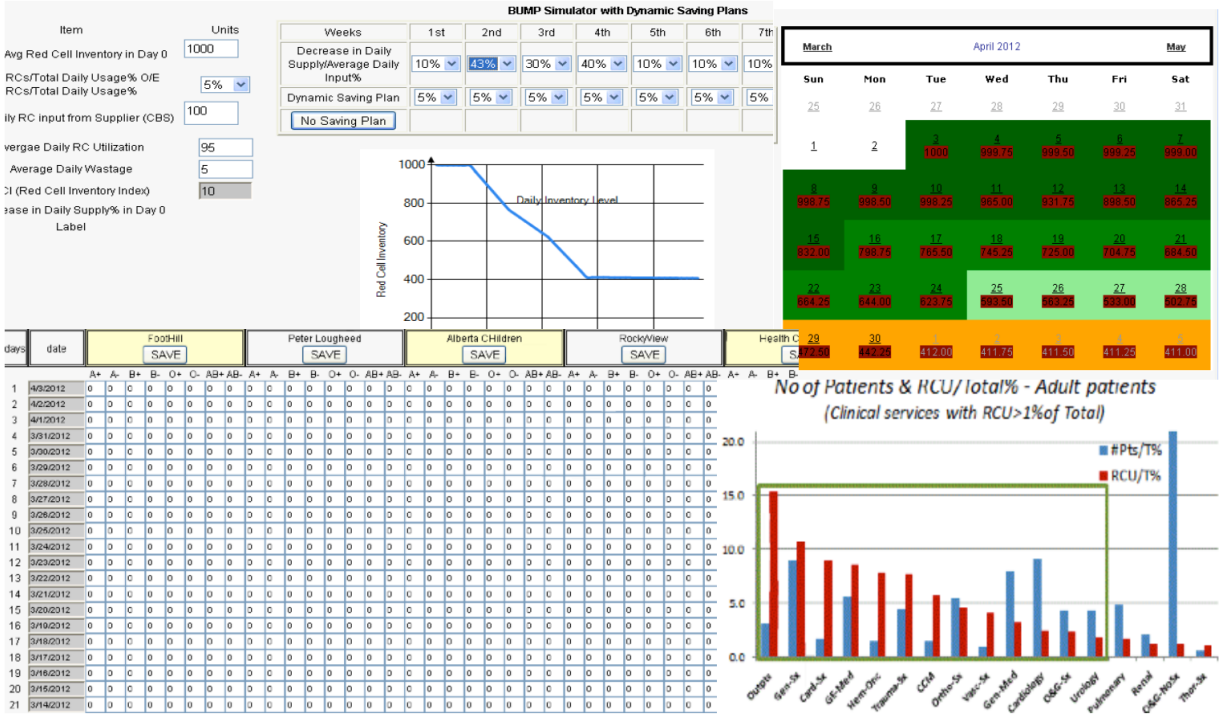


Figure 3.18 Web-based visualization of MOHKGA Output with Reporting

Daily Saving plan is one of the recommended solutions to decision makers, as shown in Fig. 3.19. Because it's user configurable settings and can be integrated in the framework during the MOOP search process. Fig.3.19 also shows that without the use of recommendation of daily saving plan. It can be observed that the inventory would quickly drop to below critical service level.

Fig.3.20 shows the calendar visualization of framework outputs with indications of over supply (Sufficient), Optimal, Amber alert under currently supply and consumption rate. Amber alert is defined as when the inventory level is below pre-defined threshold value which is defined by domain subject expert.

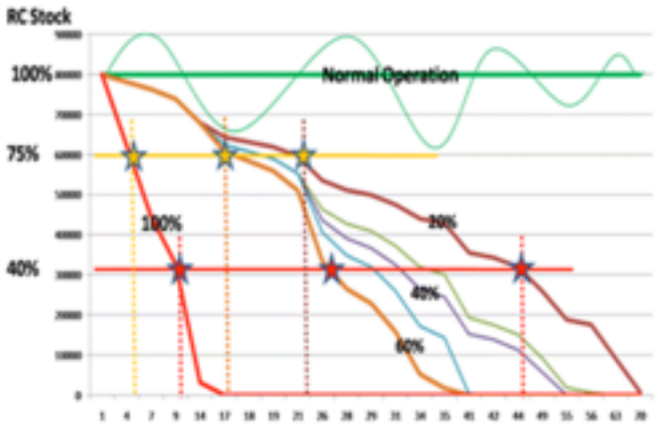


Figure 3.19 Inventory Level without Saving Plan

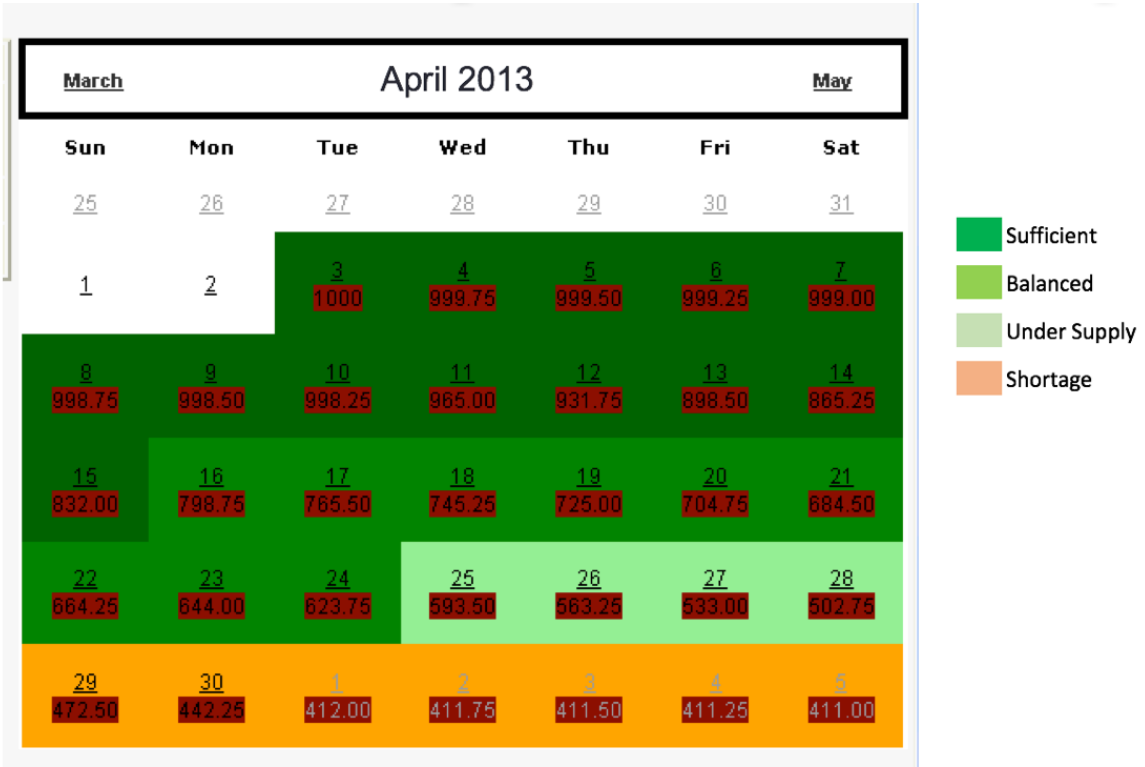


Figure 3.20 Framework Result Visualization on Calendar

3.6.2 Case Study: MOHKGA on Mobile Shopping Application

3.6.2.1 Problem Statement and Formulation

At the time of this writing, mobile devices and various mobile applications have become an essential part of people's everyday life. Shoppers usually look for the good deals when making purchases, basically, the general objectives are to

1. find all cheapest products with minimal costs
2. use minimal travel time to get the products.

The above objectives are often conflicting to each other, and trade-offs need to be made on their shopping decision making. Hence this is naturally a multi-objective optimization issue. With the use of the proposed framework as backend, an experimental mobile application, named YouSave, was created to provide recommendations to shoppers based on their current location, which can either be readily available through GPS sensor on their mobile phone, or can be inputted manually. Shoppers can then make the decision on where/what/when to purchase their intended items on their mobile device. The recommendations are from the Pareto-optimal sets generated by MOHKGA with users' preferred ways of savings, for example, save on travel time or save on costs.

3.6.2.2 Assumptions

Only certain shops and stores in the Calgary are considered in this application and goods prices information from each store does not reflect the real-time store prices. The recommendations generated by the system are only for demonstration purpose.

3.6.2.3 System Design and Considerations

The selected stores are based on the location and good prices from major stores in Calgary, Alberta, Canada. Detailed geological location information (longitude and latitude) and list of products prices information are stored in the backend database as part of MOHKA input. The outputs are in JSON data format through RESTful web services.

Input:

Users' preferences: T, C

Products: (X_1,X_2, ... X_i)

Output: *Pareto-optimal solutions (Recommended stores with trade-offs*

Constraints*: Max execution time: 5 seconds*

Objectives:

Min: T (Total time of traveling)

Min: C (Total Cost of goods)

Where T is the Traveling Time and C is the Total Cost of goods.

Web services technology has become an industry standard for connecting remote and heterogeneous resources and it overcomes the physical location constraints of conventional computing. Because of the heterogeneous nature of mobile devices, including the various hardware platform, operating systems, and programming languages, the integration of mobile computing with Web Services technology are the current industrial best practice. Mobile applications consume web services as web services provide strong interoperable capability.

YouSave consumes the web services from MOHKPA framework which contains the data of the Pareto-optimal sets. The Web Services used in the case study uses Representational state transfer (REST) or RESTful web services is a way of providing interoperability between the servers that running MOHKGA framework on the Internet. REST-compliant Web services allow requesting MOHKGA to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations. In this case, the other Web services such as WSDL and SOAP are not considered as they expose their own arbitrary sets of operations.

Through the use of RESTful Web service, the requests and response between mobile users and MOHKGA framework can be streamed in XML, HTML, JSON or some other defined format. By using a stateless protocol and standard operations, REST systems in this study can provide fast performance, reliability, and the ability to grow, by re-using components that can be managed and updated without affecting the system as a whole, even while it is running. Framework stopping time is another important factor for MOHKGA application in this application as the end user will need to have the Pareto-set solutions, i.e., the recommended stores in relatively short time in order to make their shopping decision. Therefore, MOHKGA in this case produce the recommended sores within five seconds, even there is not enough generations to produce the refined Pareto-optimal set. The expert module integrated in the framework provides the stopping condition enforcement.

The architectural design of YouSave is shown in Figure 3.21.

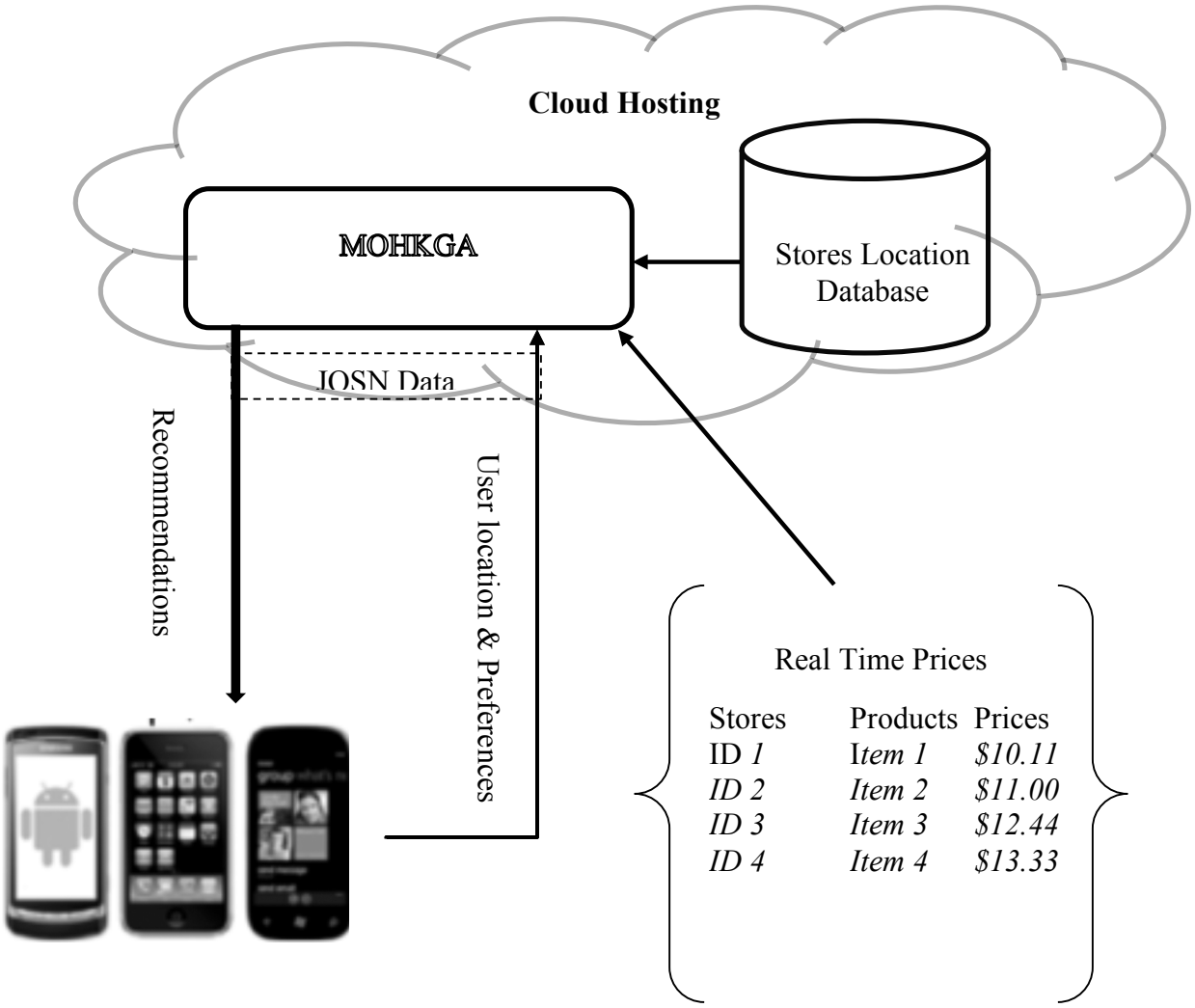


Figure 3.21 Architectural design of the Mobile Application from MOHKGA

3.6.2.4 Result Analyses and Benefits

A sample experimental results are shown in Figure 3.22 form MOHKGA based on some big box stores in the City of Calgary, Alberta.

The triangle represents shopper's current location; the round dots are stores found nearby while the green dots represents the recommended shopping stores for shopper. With the

consideration of both time and cost as objectives, the stores recommended in the Pareto set are store A and C, whereas store B is not in the Pareto set.

This application was used for Microsoft Imagine Cup 2013 competition [203] and received first place awards on the category and run-up of Canada.

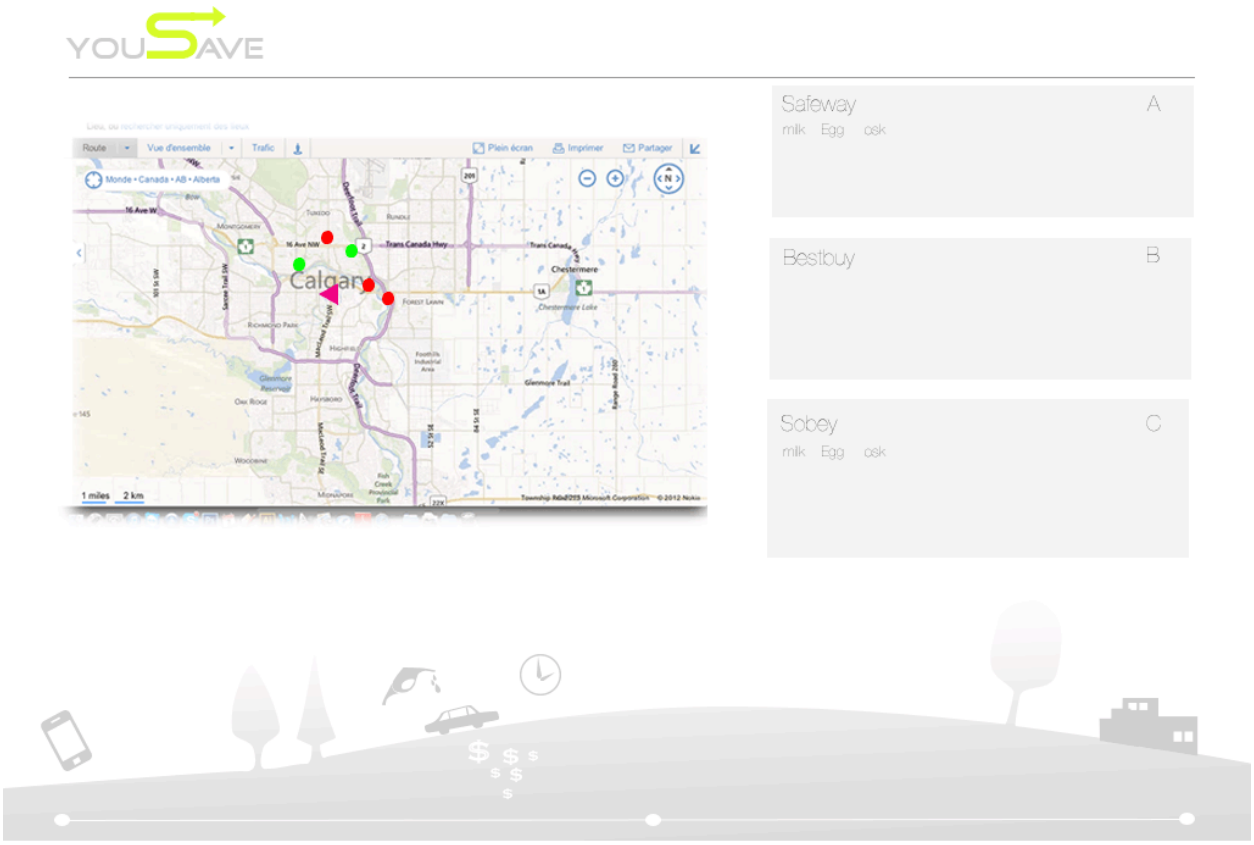


Figure 3.22 YouSave Pareto Sets from MOHKGA

Chapter Four: DISCUSSION

In general, genetic algorithms can find multiple Pareto-optimal solutions in one single simulation run if properly designed and implemented, because Genetic algorithms use population of solutions, and can be extended to maintain a diverse set of solutions toward to the true Pareto-optimal region. The common issues and challenges of genetic algorithms are performance, global optimality and result visualization, especially when the search space is complex. Many real-world problems involve a simultaneous optimization of multiple, yet competing objectives in multiple modal and high dimensional search space. In comparison with non-genetic classical algorithms, genetic programming in general has advantage on solving non-linear multi-objective optimization problems, however, the selection of algorithms and approach should be based on nature of the specific problem domain and purpose of the research. A good understanding of the problems to be solved, and the advantages and disadvantages of both genetic algorithms and classical methodologies is essential for specific problem solving. A possible hybrid approach with combination of both approaches can be an interesting research area to address the varying degrees of search-space complexity.

In this research, an innovative framework, Multi-Objective Hybrid K-Mean Genetic Algorithm (MOHKGA), was developed to solve the complex multi-objective optimization problems. The common issues from genetic algorithms are addressed with the unique operators. The uniqueness of the MOHKGA are described below.

4.1 Performance and Global Optimality

Real-world decision making on multi-objective optimization problems often has to deal with vast amounts of data, a number of alternatives and different decision situations. At the same time, the rapid diversification in industries is adding to the complexity in the search space. Because MOGAs are computational expensive, especially in multi-dimensions and multi-modal search space, MOGAs can take days or weeks to converge or complete, depending the size of datasets, the number of objectives in the objective space and hardware resources. The long execution time may not be suitable for solving real-world problems. Therefore, performance in genetic programming is a critical issue, especially when used to solve the real-world optimization problems where stringent criteria on system response time and system reliability are required, as demonstrated in the two case studies. The performance can be improved by either adding more hardware resources, like CPU, memory or solid-state drives, or by improvement of the algorithm design and implementation, or the combination of both.

The number of generations that genetic algorithms need to run and reach the true Pareto optimal front, or to converge completely, decides the performance of the algorithm because of the stochastic nature of the population-based optimization algorithms. The use of K-mean operator and parallel asynchronous process in MOHKGA is unique and proved to be very effective in improving the performance. As discussed in Chapter 3, the introduction of K-Mean operators and parallel asynchronous genetic operators can help the genetic algorithm to converge and reach the stable condition faster.

The K-mean operator utilizes the better populations and uses them to recreate (selection, mutation and crossover) locally for better populations until the process converge and meet

termination criterion. The K-mean operator during the selection steps also alleviates the local optimal problem as it increases the diversity of populations by re-grouping and selecting the representative candidates from cluster centroids as shown in Chapter 3. The framework can do a global search by maintaining a diverse population to reach global optimality and discovering potentially good regions of interest. The expert module can enhance the converge process by manual intervention and through interactive inputs in the expert module, it can achieve a more efficient search in the more specialized local search algorithm.

Data structure, software design and programming skills in the implementation of genetic algorithm also have important impacts on performance. The asynchronous parallel processes on selection, crossover and mutations can help the algorithm to find the true Pareto-optimal front and converge to the global optimality quickly. In the case study of mobile shopping applications, the importance of algorithm performance was demonstrated as the response time from the framework is critical to users.

4.2 Pareto-optimal Sets Visualization

Visual representation of multi-dimension Pareto-optimal front is one of the key functions that MOHKGA provides. Visualization of the generates Pareto-optimal fronts is important in decision-making process. It enables decision makers to quickly examine large amounts of data, understand the Pareto-optimal results, expose trends and issues efficiently, exchange ideas with key players, and make an informed decision.

The proposed MOHKGA framework has a built-in expert module with visualization feature. The two case studies of MOHKGA show the importance of the visualization of Pareto-optimal results for decision makers. There are few online visualization modules like the one

described in this research existing in the previous MOGAs, at the time of this writing. Like most research that is used for solving practical real-world problems, MOHKGA has this salient feature of providing the Pareto-optimal set in a user-friendly format to decision makers in a timely and efficient manner, instead of using separate tools to interpret and visualize the results. In the case study of blood bank inventory management, the importance of Pareto-optimal sets visualization is demonstrated. Decision makers can understand and visualize the impacts under various scenarios through the visualization of Pareto-optimal sets different, and hence take the trade-offs among the optional solutions

4.3 Expert Module

Decision makers are subject matter experts. In order to solve real-world multi-objective optimization problems, the integration of the decision makers' knowledge, expertise and preferences in the framework can the help setup the objectives to be optimized accurately with initial and boundary conditions. With the expert module, decision maker can choose suitable operators and problem-specific information with preferences, including the termination conditions, as demonstrated in the two case studies. The decision maker has to deal with vast data, number of alternatives and different decision situations before taking any decision. At the same time, the rapid diversification in industries is also adding to the complexity to the search space. Therefore, expert module can take critical information such as boundary constrains and preferences into consideration in order to determine the best course of action.

4.4 Interactivity

Genetic algorithms have inherent randomness during the evolutionary process because probability model is used in every genetic operator. Use of probability model in selection,

crossover and mutation is necessary to generate even spread and distribution of populations, and maintain the diversity. The randomness in some extreme case can cause slow convergence of the algorithm, and delayed output of the Pareto-optimal results for real-time decision making. The interactivity in the expert module can address the issue with users' manual intervention by the web-based interactive tool with user-friendly graphic user interface in MOHKGA. The interactive feature of MOHKGA frameworks also provides a practical tool for decision makers to use their domain subject expertise on the specific problems for the initial framework setup, stop the program execution without waiting for complete convergence, examine the alternative solutions from MOHKGA based on different scenarios, and take trade-offs among the solutions, in accordance with their domain subject matter expertise.

The interactivity also enables decision makers to include problem specific information in in creating the initial population to speed up the initialization process as a customized initialization can get to a faster convergence.

4.5 MOHKGA Design and Applications

MOHKGA is a complete framework that was designed with innovation methods and operator, developed and implemented with n-tier architecture design pattern. Besides the innovative design of algorithms, the implementation and development that make the design into a complete working framework are challenging because of the high complexity of algorithm and inherent parallel nature of the search processes. Data structures, design patterns, threads management and programming skills are necessary to complete the framework.

At the time of this writing, the use of the mobile device is becoming a part of people's everyday lives as discussed in case study two. The development of mobile application to receive

and interpret the Pareto-optimal sets from MOHKGA is challenging because of the heterogeneous major mobile operating systems, iOS, Android and Windows UWP.

MOHKGA is less susceptible to the shape or continuity of the search space. It can deal with non-linear, discontinuous or concave Pareto fronts, which are problematic and real concerns for classical approaches to deal with. MOHKGA has some advantages over other MOGAs in the literature. It has the aforementioned innovative design and implementation. In clustering application, MOHKGA doesn't require the estimated number of clusters as input parameters, and it can find the optimal number of clustering in the Pareto-optimal sets. This is very important in the areas of data mining and knowledge discovery as no prior information are often not known about the datasets to be studies. As demonstrated in the experimental datasets tests, the optimal number of clusters are generated as shown in the Pareto-optimal set. The results are consistent with other well-known MOGAs.

4.6 Future works

With the recent advancement of computing technology, large data sets (also known as big data) processing are made much easier and faster with the help of cloud computing [206, 207, 208]. Genetic algorithm is computationally expensive. Cloud computing offers the scalable, dynamic and elastic resources management. The power and scalability that cloud computing offers can improve the performance of genetic programming, especially when response time is a critical factor in certain real-world multi-objective optimization problems. Therefore, the use of cloud computing for MOGAs can be an interesting new direction of research. Moreover, cloud-based computer also provides the ease of access and better cost-effective management. Figure. 3.21 shows the architectural design of using MOHKGA in cloud computing.

An outlier identification module can also be added in MOHKGA in the selection operator. An outlier is a piece of data which falls far outside the expected variation, i.e., an observation that appears to deviate significantly from others. However, outliers can reveal hidden knowledge or useful intelligence in the datasets, especially the extreme conditions, for example; events such as epidemics that might have an effect on red cells supply in case study one, the unexpected events can lead to the extreme conditions of the blood supply and consumption and extremely unbalanced inventories. How to predict and manage the extreme cases is an important part of decision making scope.

In MOHKGA, the dominated individuals are discarded and removed from the rest of the generations as they are usually viewed as nuisance. The dominated individuals which are discarded can be added to a separate storage for future research as they are extreme occurrences. One suggestion is that outliers can be collected and assigned into separate populations sets during the evolutionary processes, the extremely inferior dominated solution sets can be reserved for future outlier analyses after the Pareto-optimal solutions set are obtained.

Chapter Five: CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this research, an innovative hybrid framework MOHKGA was developed to find the Pareto-optimal sets on multi-objective optimization search space. The framework presented and analyzed in this research is unique. It has the combination of local K-mean search algorithm with K-mean operator, and expert module with input to integrated user or decision makers' preference, and visualization of the Pareto-optimal fronts in user friends format.

MOHKGA can be theoretically applied in any knowledge domain of multiple objectives optimization that is subject to resource constraints and decision problems. We have demonstrated its ability to maintain a diverse population and converge to the true Pareto optimal front, applicability and effectiveness by extensive testing and analysis. We have used datasets from a variety of domains ranging from very general to very specific like gene expression data. We have also shown how the developed framework may benefit number of real-life domains. Two vital applications have been described. Blood management and utilization is a very important and challenging application with direct social impact. It helps in better serving different kinds of patients in need for blood transfusion. Shop and save is the other interesting application described in this thesis where we showed how we could help people make informed decision in terms of shopping based on their preferences through the suggested matching plan. MOHKGA provide a tool for faster decision making through the high performance and parallel computing, identification of trends and understanding of impacts of different decision actions. The built-in data visualization module presents the Pareto-optimal solutions in the form of charts and graphs in an easy to understand and summarized way.

MOHKGA can help decision maker to incorporate environmental, organizational, and managerial consideration into the model through objectives, preferences and priorities, and taking trade-offs of the conflicting objectives from the Pareto-optimal solutions, because a gain in one objective form one point happens only by sacrificing in the other objective. The trade-off property between the Pareto-optimal solutions provides recommendations to decision makers, who can pick the more applicable or optimal solution by taking considerations of all related factors and the knowledge of expertise.

MOHKGA is a complete tool that can be applied directly to solve complex multi-objective optimization problems. However, multi-objective optimization problems can be solved from different viewpoints and goals, thus, there exist different solutions and algorithms, which may be more suitable for other multi-objective optimization problems. For example, the goal may be to find a representative set of Pareto optimal solutions, and/or quantify the trade-offs in satisfying the different objectives, and/or finding a single solution that satisfies the subjective preferences of decision makers.

We are currently investigating other applications of the proposed approach in domains like resource management in countries who are suffering economic crisis and seeking for the better allocation of resources. Helping such countries in better managing their resources will be a great benefit to them, their allies and partners. We are also investigating whether other objectives would be important to consider in general or should we concentrate on domain specific objectives for more focused outcome. The latter may require tuning the approach to fit better each domain after deciding on and integrating its domain specific objectives.

References

[1] Ching-Lai Hwang; Abu Syed Md Masud (1979). Multiple objective decision making, methods and applications: a state-of-the-art survey. Springer-Verlag. ISBN 978-0-387-09111-2. Retrieved 29 May 2012

[2] Peng, Xiufeng Peter; Addam, Omer; Elzohbi, Mohamad; Özyer, Sibel T; Elhajj, Ahmad

Reporting and analyzing alternative clustering solutions by employing multi-objective genetic algorithm and conducting experiments on cancer data, Knowledge-Based Systems 56:108–122 · January 2014

[3] M. Neef, D. Thieens, & H. Arciszewski, A Case Study of a Multi-objective Elitist Recombinative Genetic Algorithm with Coevolutionary Sharing. In Angeline, P. (Ed.), Proc. of the International Congress on Evolutionary Computation, pp.796-803. Priscatawy. 1999.

[4] Silvano Martello and Paolo Toth. Knapsack Problems: Algorithms and Computer Implementations. Wiley, Chichester, 1990.

[5] Eckart Zitzler and Lothar Thiele. Multi-objective optimization using evolutionary algorithms — A comparative case study. International Conference on Parallel Problem Solving from Nature PPSN 1998: Parallel Problem Solving from Nature — PPSN V pp 292-301

[6] S. Venkata Lakshmi; Valli Kumari Vatsavayi, 2016 International Conference on Computer Communication and Informatics (ICCCI) IEEE Conference Publications) Pages: 1 - 8, DOI: 10.1109/ICCCI.2016.7479934, 2016

[7] Estivill-Castro, Vladimir. “Why so many clustering algorithms — A Position Paper". ACM SIGKDD Explorations Newsletter. 4 (1): 65–75. doi:10.1145/568574.568575., 2002

[8] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996

[9] A. Konak et al. Multi-objective optimization using genetic algorithms: A tutorial. 91 (2006) 992–1007

[10] T. Kohonen, Self-organizing Maps: Berlin/Heidelberg: Springer-Verlag, 1997.

[11] Microarray Data Analysis: Direct Gene Sample Correlations, Gene Network Science, Inc. (c). 2001

[12] W. Shannon, R. Culverhouse J. Duncan. Analyzing microarray data using cluster analysis, Pharmacogenomics, Vol.4, No.1, pp.41-52, 2003.

[13] R. M. Cole, Clustering with genetic algorithms, http://www.cs.uwa.edu.au/pub/robvis/theses/ Rowena Cole, 1998.

[14] Rokach, Lior, and Oded Maimon. "Clustering methods." Data mining and knowledge discovery handbook. . 321-352. Springer US, 2005

[15] D. Jiang, C. Tang, and A. Zhang. Cluster Analysis for Gene Expression Data: A Survey, IEEE Transactions on Knowledge and Data Engineering, pp.1-5, 2003.

[16] Shah, J. Harendra: A Review of DNA Microarray Data Analysis, Biochemistry 218/ Medical Information Sciences, 231, 2002.

[17] A.K. Jain and R.C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, N.J.: Prentice Hall, 1988.

[18] L. Kaufman and P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley & Sons, 990.,

[19] R.T. Ng and J. Han, [a]Efficient and Effective Clustering Methods for Spatial Data Mining, Proc. 20th Int'l Conf. Very Large Databases, pp. 144-155, Sept. 1994,

[20] T. Kohonen, Self-Organization and Associative Memory, third ed. New York: Springer-Verlag, 1989.

[21] M. Ester, H. Kriegel, and X. Xu, [a]A Database Interface for Clustering in Large Spatial Databases, Proc. First Int'l Conf. Knowledge Discovery and Data Mining (KDD-95), pp. 94-99, 1995.

[22] T. Zhang, R. Ramakrishnan, and M. Livny, BIRCH: A New Data Clustering Algorithm and Its Applications, Data Mining and Knowledge Discovery, vol. 1, no. 2, pp. 141-182, 1997.

[23] P.S. Bradley, U. Fayyad, and C. Reina, Scaling Clustering Algorithms to Large Databases, Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining, pp. 9-15, 1998.

[24] L. Kaufman and P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley & Sons,1990.

[25] V. Capoyleas, G. Rote, and G. Woeginger, Geometric Clusterings, J. Algorithms, vol. 12, pp. 341-356, 1991.

[26] A.K. Jain and R.C. Dubes. Algorithms for Clustering Data. Englewood Cliffs, N.J.: Prentice Hall, 1988

[27] A.K. Jain, M.N. Murty, and P.J. Flynn. Data Clustering: A Review, ACM Computing Surveys, vol. 31, no. 3, pp. 264-323, 1999.

[28] A.K. Jain, P.W. Duin, and J. Mao, [a]Statistical Pattern Recognition: A Review, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 4-37, Jan. 2000.

[29] P. Tamayo, et al, Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation, Proc. of. Nat'l Acad Sci, pp.2907-2912, 1999.

[30] B.J.T. Morgan, and A.P.G. Ray, Non-uniqueness and Inversions in Cluster Analysis, Applied Statistics, Vol.44, No.1, pp.117-134. 1995.

[31] E. Segal, D. Koller: Probabilistic hierarchical clustering for biological data. Proc. Inter. Conf. on Research in Computational Molecular Biology, Washington, DC, pp. 273--280.April 2002

[32] S. Selim and M. Ismail, "K-means-type algorithms: A generalized convergence theorem and characterization of local optimality," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, pp. 81–87, 1984.

[33] Franz Aurenhammer (1991). Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys, 23(3):345–405, 1991

[34] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara & Sung Nok Chiu. Spatial Tessellations – Concepts and Applications of Voronoi Diagrams. 2nd edition. John Wiley, 671 pages ISBN 0-471-98635-6, 2000

[35] Aloise, D.; Deshpande, A.; Hansen, P.; Popat, P. "NP-hardness of Euclidean sum-of-squares clustering". Machine Learning. 75: 245–249. doi:10.1007/s10994-009-5103-0. 2009

[36] Dasgupta, S.; Freund, Y"Random Projection Trees for Vector Quantization". Information Theory, IEEE Transactions on. 55: 3229–3242. arXiv:0805.1390 doi:10.1109/TIT.2009.2021326. 2009.

[37] Mahajan, M.; Nimbhorkar, P.; Varadarajan, K. "The Planar k-Means Problem is NP-Hard". Lecture Notes in Computer Science. 5431: 274–285. doi:10.1007/978-3-642-00202-1_24. 2009.

[38] Inaba, M.; Katoh, N.; Imai, H. Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. Proceedings of 10th ACM Symposium on Computational Geometry. pp. 332–339. doi:10.1145/177424.178042. 1994

[39] Lloyd, Stuart P., "Least squares quantization in PCM", IEEE Transactions on Information Theory, 28 (2): 129–137, doi:10.1109/TIT.1982.1056489. 1982

[40] Y. Barash & N. Friedman. Context-specific Bayesian clustering for gene expression data. Proc. of RECOMB, pp.12-21, 2001.

[41] J. C. Mar, G. J. McLachlan: Model-Based Clustering in Gene Expression Microarrays: An Application to Breast Cancer Data. APBC, pp.139-144. 2003.

[42] A. Ben-Dor, R. Shamir and Z. Yakhini, Clustering gene expression patterns, Journal of Computational Biology, 6(3-4), pp.281, 1999.

[43] K. Curtis & M. Brand, Control analysis of DNA microarray expression data. Mol Biol Rep. 29(1-2), pp.67-71. 2002.

[44] M. Ramoni, P. Sebastiani and I.S. Kohane. Cluster Analysis of Gene Expression Dynamics. Proc. Nat Acad Sci., Vol.14, pp.9121-6, 2002.

[45] Kriegel, Hans-Peter; Kröger, Peer; Sander, Jörg; Zimek, Arthur "Density-based Clustering". WIREs Data Mining and Knowledge Discovery. 1 (3): 231–240. doi:10.1002/widm.30, 2011.

[46] Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei. "A density-based algorithm for discovering clusters in large spatial databases with noise". In Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231. CiteSeerX 10.1.1.71.1980 . ISBN 1-57735-004-9. 1996

[47] Ankerst, Mihael; Breunig, Markus M.; Kriegel, Hans-Peter; Sander, Jörg. "OPTICS: Ordering Points To Identify the Clustering Structure". ACM SIGMOD international conference on Management of data. ACM Press. pp. 49–60. CiteSeerX 10.1.1.129.6542, 1999

[48] Achtert, E.; Böhm, C.; Kröger, P. "DeLi-Clu: Boosting Robustness, Completeness, Usability, and Efficiency of Hierarchical Clustering by a Closest Pair Ranking". LNCS: Advances in Knowledge Discovery and Data Mining. Lecture Notes in Computer Science. 3918: 119–128. doi:10.1007/11731139_16. ISBN 978-3-540-33206-0. 2006

[49] Guttman, A. (1984). "R-Trees: A Dynamic Index Structure for Spatial Searching". Proceedings of the 1984 ACM SIGMOD international conference on Management of data – SIGMOD '84 (PDF). p. 47. doi:10.1145/602259.602266. ISBN 0897911288. 1984

[50] Roy, S.; Bhattacharyya, D. K. "An Approach to find Embedded Clusters Using Density Based Techniques". LNCS Vol.3816. Springer Verlag. pp. 523–535. 2005

[51] Cheng, Yizong "Mean Shift, Mode Seeking, and Clustering". IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE. 17 (8): 790–799. doi:10.1109/34.400568. 1995.

[52] Comaniciu, Dorin; Peter Meer. "Mean Shift: A Robust Approach Toward Feature Space Analysis". IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE. 24 (5): 603–619. doi:10.1109/34.1000236. 2002

[53] J. Grabmeier, et al, Techniques of Cluster Algorithms in Data Mining, Kluwer Academic Publishers, Data Mining and Knowledge Discovery, Vol.6, pp.303-360, 2003.

[54] N. Bolshakova, F. Azuaje, Improving expression data mining through cluster validation, Proc. of IEEE Conference on Information Technology Applications in Biomedicine, pp.19-22, 2003.

[55] U. Scherf, et al. A Gene Expression Database for the Molecular Pharmacology of Cancer, Nat Genet, Vol.24, pp.236-244, 2000.

[56] T. R. Golub, et al, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science ,286, pp.531-537, 1999.

[57] E. Domany, Cluster Analysis of Gene Expression Data, physics,110, pp.11-17, 2002.

[58] S. Kaski, Data exploration using Self-Organizing maps. Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series, No. 82, pp 57.March 1997.

[59] D. Wang, H. Ressom, M. Musavi, C. Domnisoru; Double Self-Organizing Maps to Cluster Gene Expression Data, Proc. of ESANN, pp.45-50, 2000.

[60] U. Möller, D. Radke, F. Thies, Testing the significance of clusters found in gene expression data. Proc. of European Conference on Computational Biology, Paris, pp.26,-30, 2003.

[61] E. Levine, E. Domany. Resampling Method for Unsupervised Estimation of Cluster Validity, Neural Computation, Vol.13, pp.2573-2593, 2001.

[62] V. Roth, T. Lange, M. Braun, M. Buhmann. A Resampling Approach to Cluster Validation. Computational Statistics - COMPSTAT, Physica Verlag. pp.123-128. 2002.

[63] A. Ben-Hur, A. Elisseeff and I. Guyon, A stability based method for discovering structure in clustered data, Proc. of PSB, pp.6-17, 2002.

[64] M. Kathleen Kerr and G. Churchill. Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. PNAS, 98, pp.8961-8965, 2001.

[65] Peter J. Rousseeuw. "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics. 20: 53–65. doi:10.1016/0377-0427(87)90125-7. 1987

[66] R. Ulrich, and S. Friend, Toxicogenomics and drug discovery: will new technologies help us produce better drugs? Nat. Rev. Drug Discov. 1, pp.84-88, 2002.

[67] P. McConnell, K. Johnson, and D. J. Lockhart, An introduction to DNA microarrays, Proc. of CAMDA, 2001.

[68] A. Ben-Hur and I. Guyon, Detecting Stable Clusters Using Principal Component Analysis, In Methods in Molecular Biology, M.J. Brownstein and A. Kohodursky (eds.) Humana Press, pp.159-182, 2003.

[69] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, JRSS-B, 63, pp.411-423, 2001.

[70] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 881-892,

doi: 10.1109/TPAMI.2002.1017616, 2002.

[71] Th. Bäck, D.B. Fogel, and Z. Michalewicz (Editors), Handbook of Evolutionary Computation, ISBN 0750303921, 1997

[72] Th. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation, 1(1):1–23, 1993.

[73] Holland, John H. Adaptation in Natural and Artificial Systems. University of Michigan Press. ISBN 0-262-58111-6. 1975

[74] Koza, John R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press. ISBN 0-262-11170-5. 1992

[75] G. C. Onwubolu and B V Babu, "New Optimization Techniques in Engineering". ISBN 978-3-540-39930-8, 2004.

[76] Holland, J. H., Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, 1975

[77] Deb K Optimization for engineering design: Algorithms and examples (Delhi: Prentice-Hall) 1995

[78] Deb K, Agrawal R B Simulated binary crossover for continuous search space. Complex Syst. 9: 115-148, 1995

[79] Deb K, Goldberg D E An investigation of niche and species formation in genetic function optimization Proceedings of the Third International Conference on Genetic Algorithms, pp. 42-50, 1989

[80] J. Jahn, Mathematical Vector Optimization in Partially Ordered Linear Spaces, Vol. 31, Verlag Peter Lang GmbH, 1986

[81] R.E. Steuer, Multiple Criteria Optimization: Theory, Computation, and Applications, John Wiley & Sons, Inc., 1986

[82] K. Miettinen, Nonlinear Multi-objective Optimization, Kluwer Academic Publishers, 1999.

[83] Julio B. Clempnera, Alexander S. Poznyakb. Solving the Pareto front for multi-objective Markov chains using the minimum Euclidean distance gradient-based optimization method, Mathematics and Computers in Simulation Volume 119, Pages 142–160, 2016

[84] Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the international conference on genetic algorithm and their applications, 1985.

[85] A. M. ElSheikh, T. Jarada, M. Nagi, G. Naji, R. Alhajj, Xiufeng P. Peng, T. Özyer, K. Kianmehr. Effectiveness of Feature Selection and Classification Techniques for Gene Expression Data Analysis. Conference: The 5th International Conference on Information Technology (ICIT 2011)

[86] Eiben, A. E. et al "Genetic algorithms with multi-parent recombination". PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: 78–87. ISBN 3-540-58484-6. 1994

[87] Ting, Chuan-Kang. "On the Mean Convergence Time of Multi-parent Genetic Algorithms Without Selection". Advances in Artificial Life: 403–412. ISBN 978-3-540-28848-0. 2005

[88] Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, Massachusetts: Addison-Wesley. 1989

[89] Goldberg, D. E. and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J. J. Grefenstette (Ed.), Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms, Hillsdale, NJ, pp. 41–49. 1987

[90] Mitchell, M. An Introduction to Genetic Algorithms. MIT Press. 1996

[91] Kaisa Miettinen Nonlinear Multi-objective Optimization. Springer. ISBN 978-0-7923-8278-2. 1999

[92] Cohon, J.L. Multi-objective Programming and Planning. New York: Academic Press. 1978

[93] Whitley, Darrell. "A genetic algorithm tutorial". Statistics and Computing. 4 (2): 65–85. doi:10.1007/BF00175354. 1994

[94] Shirazi, Ali; Najafi, Behzad; Aminyavari, Mehdi; Rinaldi, Fabio; Taylor, Robert A. "Thermal–economic–environmental analysis and multi-objective optimization of an ice thermal energy storage system for gas turbine cycle inlet air cooling". Energy. 69: 212–226. doi:10.1016/j.energy.2014.02.071. 2014

[95] Najafi, Behzad; Shirazi, Ali; Aminyavari, Mehdi; Rinaldi, Fabio; Taylor, Robert A. 02-03). "Exergetic, economic and environmental analyses and multi-objective optimization of an SOFC-gas turbine hybrid cycle coupled with an MSF desalination system". Desalination. 334 (1): 46–59. doi:10.1016/j.desal.2013.11.039. 2014

[96] Jürgen Branke; Kalyanmoy Deb; Kaisa Miettinen; Roman Slowinski (21 November 2008). Multi-objective Optimization: Interactive and Evolutionary Approaches. Springer. ISBN 978-3-540-88907-6. Retrieved 1 November 2012.

[97] E.S. Gelsema (Ed.), Special Issue on Genetic Algorithms, Pattern Recognition Letters, vol. 16(8), Elsevier Sciences Inc., Amsterdam, 1995.

[98] S.K. Pal, P.P. Wang (Eds.), Genetic Algorithms for Pattern Recognition, CRC Press, Boca Raton. 1996

[99] Dunn, J. "Well separated clusters and optimal fuzzy partitions". Journal of Cybernetics. 4: 95–104. doi:10.1080/01969727408546059. 1974

[100] Manning, Christopher D.; Raghavan, Prabhakar; Schütze, Hinrich. Introduction to Information Retrieval. Cambridge University Press. ISBN 978-0-521-86571-5.

[101] Färber, Ines; Günnemann, Stephan; Kriegel, Hans-Peter; Kröger, Peer; Müller, Emmanuel; Schubert, Erich; Seidl, Thomas; Zimek, Arthur. On Using Class-Labels in Evaluation of Clusterings. Discovering, Summarizing, and Using Multiple Clusterings. ACM SIGKDD. 2010

[102] J.R. Koza, A genetic approach to the truck backer upper problem and the inter-twined spiral problem. In Proceedings of IJCNN International Joint Conference on Neural Networks, vol. IV (IEEE Press,), pp. 310–318, 1992

[103] N. Kashtan, U. Alon, Spontaneous evolution of modularity and network motifs. In Proceedings of the National Academy of Sciences 102, 39, pp. 13773–13778, 2005

[104] Taherdangkoo, Mohammad; Paziresh, Mahsa; Yazdi, Mehran; Bagheri, Mohammad Hadi. "An efficient algorithm for function optimization: modified stem cells algorithm". Central European Journal of Engineering. 3 (1): 36–50. doi:10.2478/s13531-012-0047-8, 2012

[105] Wolpert, D.H., Macready, W.G. No Free Lunch Theorems for Optimization. Santa Fe Institute, SFI-TR-05-010, Santa Fe. 1995

[106] J. Branke, Evolutionary Optimization in Dynamic Environments (Kluwer, Dordrecht), 2001

[107] I. Dempsey, M. O'Neill, A. Brabazon, Foundations in Grammatical Evolution for Dynamic Environments, vol. 194 of Studies in Computational Intelligence (Springer, 2009, Apr)

[108] R. Morrison, Designing Evolutionary Algorithms for Dynamic Environments (Springer, Berlin, 2004)

[109] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms. Chichester, U.K.: Wiley, 2001.

[110] J. Horn, N. Nafpliotis and D. E. Goldberg, A Niched Pareto Genetic Algorithm for Multi-objective Optimization, Proc. of IEEE Conference on Evolutionary Computation, Vol.1, pp.82-87, Piscataway, NJ. 1994.

[111] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization," in Proceedings of the Fifth International Conference on Genetic Algorithms, S. Forrest, Ed. San Mateo, CA: Morgan Kauffman, pp. 416–423. 1993

[112] N. Srinivas and K. Deb, "Multi-objective function optimization using

non-dominated sorting genetic algorithms," Evol. Comput., vol. 2, no. 3, pp. 221–248, Fall 1995.

[113] E. Zitzler and L. Thiele, "Multi-objective optimization using evolutionary algorithms—A comparative case study," in Parallel Problem Solving From Nature, V, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, pp. 292–301. 1998

[114] P. Hajela, C.-y. Lin, Genetic search strategies in multi-criterion optimal design, Struct Optimization, 4 (2), pp. 99–107, 1992

[115] Blickle, T., J. Teich, and L. Thiele . System-level synthesis using evolutionary

algorithms. Design Automation for Embedded Systems 3(1), 1998

[116] Peng, Xiufeng Peter; Nagi, Mohamad; Oair, Omer; Suleiman, Iyad; Qabaja, Ala. From Alternative Clustering to Robust Clustering and Its Application to Gene Expression Data

Intelligent Data Engineering and Automated Learning - IDEAL, Volume 6936, pp. 421 – 428, 2011

[117] Nagi, Mohamad; ElSheikh, Abdallah; Sleiman, Iyad; Peng, Xiufeng Peter; Rifaie, Mohammad. Association rules mining based approach for Web usage mining IEEE International Conference on Information Reuse & Integration 2011, pp. 166 – 171, 2011

[118] Rasheed, Faraz; Peng, Xiufeng Peter; Rokne, Jon. Fourier Transform Based Spatial Outlier Mining. Intelligent Data Engineering and Automated Learning - IDEAL, Volume 5788, pp. 317 – 324, 2009

[119] D.A. Van Veldhuizen, J.B. Zydallis, G.B. Lamont. Considerations in engineering parallel multi-objective evolutionary algorithms IEEE Trans Evol Comput, 7 (2), pp. 144–173, 2003

[120] de Toro F, Ortega J, Fernandez J, Diaz A. PSFGA: a parallel genetic algorithm for multi-objective optimization. In: Proceedings of the 10th Euromicro workshop on parallel, distributed and network-based processing, Canary Islands, Spain: IEEE Computer Society. 9–11 January, 2002.

[121] S. Xiong, F. Li. Parallel strength Pareto multi-objective evolutionary algorithm

Proceedings of the fourth international conference on parallel and distributed computing, applications and technologies, 27–29 August, 2003, IEEE, Chengdu, China. 2003

[122] L.A. Wilson, M.D. Moore, J.P. Picarazzi, S.D.S. Miquel. Parallel genetic algorithm for search and constrained multi-objective optimization. Proceedings of the 18th international parallel and distributed processing symposium, 26–30 April, 2004, IEEE Computer Society, Santa Fe, NM, USA (2004)

[123] D.F. Jones, S.K. Mirrazavi, M. Tamiz. Multi-objective meta-heuristics: an overview of the current state-of-the-art. Eur J Oper Res, 137 (1), pp. 1–9, 2002

[124] B.L. Miller, M.J. Shaw. Genetic algorithms with dynamic niche sharing for multimodal function optimization. Proceedings of the 1996 IEEE international conference on evolutionary computation, ICEC'96, May 20–22, 1996, Nagoya, Japan, IEEE, Piscataway, NJ, USA (1996)

[125] H. Ratschek, J Rokne. New Computer methods for Global Optimization. 2007

[126] E. Zitzler, L. Thiele. Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput, 3 (4), pp. 257–271. 1999

[127] D.A. Van Veldhuizen, G.B. Lamont. Multi-objective evolutionary algorithms: analyzing the state-of-the-art. Evol Comput, 8 (2), pp. 125–147, 2000

[128] M.T. Jensen. Reducing the run-time complexity of multi-objective EAs: The NSGA-II and other algorithms. IEEE Trans Evol Comput, 7 (5), pp. 503–515, 2003

[129] J.E. Fieldsend, R.M. Everson, S. Singh. Using unconstrained elite archives for multi-objective optimization. IEEE Trans Evol Comput, 7 (3), pp. 305–323, 2003

[130] S. Mostaghim, J. Teich, A. Tyagi. Comparison of data structures for storing Pareto-sets in MOEAs. Proceedings of the 2002 world congress on computational intelligence—WCCI'02, 12–17 May, 2002, IEEE, Honolulu, HI, USA. 2002

[131] D.W. Corne, J.D. Knowles, M.J. Oates. The Pareto envelope-based selection algorithm for multi-objective optimization. Proceedings of sixth international conference on parallel problem solving from Nature, 18–20 September, 2000, Springer, Paris, France (2000)

[132] H. Lu, G.G. Yen. Rank-density based multi-objective genetic algorithm. Proceedings of the 2002 world congress on computational intelligence—WCCI'02, 12–17 May, 2002, IEEE, Honolulu, HI, USA (2002)

[133] H. Ishibuchi, T. Murata. Multi-objective genetic local search algorithm. Proceedings of the IEEE international conference on evolutionary computation, 20–22 May, 1996, IEEE, Nagoya, Japan (1996)

[134] G.G. Yen, H. Lu. Dynamic multi-objective evolutionary algorithm: adaptive cell-based rank and density estimation. IEEE Trans Evol Comput, 7 (3), pp. 253–274, 2003

[135] C.A.C. Coello. A survey of constraint handling techniques used with evolutionary algorithms. Laboratorio Nacional de Informtica Avanzada, Veracruz, Mexico (1999)

[136] F. Jimenez, A.F. Gomez-Skarmeta, G. Sanchez, K. Deb. An evolutionary algorithm for constrained multi-objective optimization. Proceedings of the 2002 world congress on computational intelligence—WCCI'02, 12–17 May, 2002, IEEE, Honolulu, HI, USA (2002)

[137] Amirahmadi, Ahmadreza; A. Dastfan; S.M.R. Rafiei. "Optimal Controller Design for Single-phase PWM Rectifier Using SPEA Multi-objective Optimization". Journal of Power Electronics. 12 (1). 2012

[138] Shirazi, Ali; Najafi, Behzad; Aminyavari, Mehdi; Rinaldi, Fabio; Taylor, Robert A. Thermal–economic–environmental analysis and multi-objective optimization of an ice thermal energy storage system for gas turbine cycle inlet air cooling . Energy. 69: 212–226. doi:10.1016/j.energy.2014.02.071. 2014

[139] Najafi, Behzad; Shirazi, Ali; Aminyavari, Mehdi; Rinaldi, Fabio; Taylor, Robert A. Exergetic, economic and environmental analyses and multi-objective optimization of an SOFC-gas turbine hybrid cycle coupled with an MSF desalination system. Desalination. 334 (1): 46–59. doi:10.1016/j.desal.2013.11.039. 2014

[140] R. Ng and J. Han. "Efficient and effective clustering method for spatial data mining". In: Proceedings of the 20th VLDB Conference, pages 144-155, Santiago, Chile, 1994.

[141] Koza, J. R. Genetic Programming II: Automatic Discovery of Reusable Programs. MIT 293 Press. Cambridge, MA. 1994

[142] Cunha, A. G., P. Oliviera, and J. Covas. Use of genetic algorithms in multicriteria optimization to solve industrial problems. In T. Back (Ed.), Proceedings of the Seventh International Conference on Genetic Algorithms,San Francisco, California, pp. 682–688. Morgan Kaufmann. 1997

[143] Openshaw, S. & Wymer, C. Classifying and regionalizing census data. In S. Openshaw, Census Users Handbook (pp. 239-270). Cambridge, UK: Geo Information International. 1994

[144] M. Hamdy, M. Palonen, A. Hasan. Implementation of pareto-archive NSGA-II algorithms to a nearly-zero-energy building optimisation Problem. Proceedings of the 2012 Building Simulation and Optimization Conference, Loughborough, Leicestershire, UK (2012)

[145] M. Hamdy, A. Hasan, K. Siren. Impact of adaptive thermal comfort criteria on building energy use and cooling equipment size using a multi-objective optimization scheme Energy Build., 43, pp. 2055–2067, 2011

[146] A.T. Nguyen, S. Reiter, P. Rigo. A review on simulation-based optimization methods applied to building performance analysis. Appl. Energ., 113 (2014), pp. 1043–1058

[147] M.S. Osmana, M.A. Abo-Sinnab, A.A. Mousab. An effective genetic algorithm approach to multi-objective resource allocation problems (MORAPs). Applied Mathematics and Computation. Volume 163, Issue 2, 15 April 2005, Pages 755–768.

[148] S.S. Rao. Optimization Theory and Application. Wiley Eastern Limited, New Delhi. 1991

[149] Chi-Ming Lina, Mitsuo Gena. Multi-objective resource allocation problem by multistage decision-based hybrid genetic algorithm. Applied Mathematics and Computation

Volume 187, Issue 2, 15 April 2007, Pages 574–583

[150] M.L. Hussein, M.A. Abo-Sinna. A fuzzy dynamic approach to the multi-criterion resource allocation problem. Fuzzy Sets and Systems, 69, pp. 115–124, 1995

[151] Hsieh CL. An Evolutionary-Based Optimization for a Multi-Objective Blood Banking Supply Chain Model. In: Ali M., Pan JS., Chen SM., Horng MF. (eds) Modern Advances in

Applied Intelligence. IEA/AIE 2014. Lecture Notes in Computer Science, vol 8481. Springer, Cham. 2014

[152] Sivakumar P., Ganesh K., Punnniyamoorthy M. and Lenny Koh S.C. Genetic Algorithm for Inventory Levels and Routing Structure Optimization in Two Stage Supply Chain. International Journal of Information Systems and Supply Chain Management (IJISSCM), 2013, vol. 6, issue 2, pages 33-49, 2013

[153] Aderemi Adewumi, Nigel Budlender and Micheal Olusany, Optimizing the Assignment of Blood in a Blood Banking System: Some Initial Results, WCCI 2012 IEEE World Congress on Computational Intelligence June, 10-15, 2012 - Brisbane, Australia

[154] M. Sibuya, "A random clustering process," *Annals of the Institute of Statistical Mathematics*, vol. 45, no. 3, pp. 459–465, 1993.

[155] N. Hoshino, "Random clustering based on the conditional inverse Gaussian-Poisson distribution," *Journal of Japan Statistic Society*, vol. 33, no. 1, pp. 105–117, 2003.

[156] C. A. Charalambides, "Distributions of random partitions and their applications," *Methodology and Computing in Applied Probability*, vol. 9, no. 2, pp. 163–193, 2007.

[157] S. Bandyopadhyay, A. Mukhopadhyay and U. Maulik, "An Improved Algorithm for Clustering Gene Expression Data", *Bioinformatics*, Vol. 23, No. 21, pp. 2859-2865, 2007.

[158] T. Özyer, M. Zhang and R. Alhajj, "Integrating Multi-Objective Genetic Algorithm Based Clustering and Data Partitioning for Skyline Computation," *Applied Intelligence*, Vol.35, No.1, pp.110-122, 2011.

[159] T. Özyer and R. Alhajj, "Parallel Clustering of High Dimensional Data by Integrating Multi-Objective Genetic Algorithm with Divide and Conquer," *Applied Intelligence*, Vol.31, No.3, pp.318-331, December, 2009.

[160] M. Kaya and R. Alhajj, "Multi-Objective Genetic Algorithms Based Automated Clustering for Fuzzy Association Rules Mining," *Journal of Intelligent Information Systems*, Vol.31, No.3, pp.243-264, December 2008.

[161] T. Özyer and R. Alhajj, "Deciding on Number of Clusters by Multi-Objective Optimization and Validity Analysis," *Journal of Multiple-Valued Logic and Soft Computing*, pp.457-474, Vol.14, No.3-5, 2008.

[162] U Maulik, A. Mukhopadhyay and S. Bandyopadhyay, "Combining Pareto-Optimal Clusters using Supervised Learning for Identifying Co-expressed Genes", *BMC Bioinformatics*, Vol. 10, No. 27, 2009.

[163] S. Saha and S. Bandyopadhyay. A New Symmetry Based Multi-objective Clustering Technique for Automatic Evolution of Clusters. *Pattern Recognition*, Vol.43, No.3, pp.738-751, March 2010.

[164] F. Folino and C. Pizzuti, "A Multi-objective and Evolutionary Clustering Method for Dynamic Networks", *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, pp. 256-263, Odense, Denmark, August 2010.

[165] D. Datta, J.R Figuera, C.M. Fonseca, and F. Tavares-Pereira. Graph partitioning through a multi-objective evolutionary algorithm: *A preliminary study. Proc. of the Genetic and Evolutionary Computation Conference (GECCO'08)*, pp.625-632, 2008.

[166] Y. Chi, X. Song, D.Zhou, K.Hino, and B.L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. Proc. International Conference on Knowledge Discovery and Data Mining (KDD'07), pp.153-162, 2007.

[167] H. Li and Q. Zhang, .Multi-objective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II, IEEE Trans. *Evolutionary Computation,* Vol.12, No.2, 2008.

[168] Nobukazu Matake, Tomoyuki Hiroyasu, Mitsunori Miki, Tomoharu Senda, "Multi-objective Clustering with Automatic k-determination for Large-scale Data", *GECCO* 2007, pp. 861-868.

[169] E. R. Hruschka, Ricardo J. G. B. Campello, A. A. Freitas & A. C. P. L. F. de Carvalho, A Survey of Evolutionary Algorithms for Clustering, *IEEE Transactions On Systems, Man, and Cybernetics-Part C: Applications And Reviews*, VOL. 39, NO. 2, 2009

[170] Y. Lu, et al, FGKA: A Fast Genetic K-means Clustering Algorithm, *Proceedings of ACM Symposium on Applied Computing*, Nicosia, Cyprus, pp.162-163, 2004.

[171] J.A. Hartigan, Clustering Algorithms: New York: John Wiley and Sons, pp.353. 1975.

[172] E. Zitzler, Evolutionary algorithms for multi-objective optimization: Methods and applications, Doctoral thesis ETH NO. 13398, Zurich: *Swiss Federal Institute of Technology (ETH)*, Aachen, Germany: Shaker Verlag, pp.19-39, 1999.

[173] B. Stein, S. Meyer and F. Wissbrock. On Cluster Validity and the Information Need of Users. *Proc. of the International Conference on Artificial Intelligence and Applications*, Benalmadena, Spain, September 2003.

[174] Y. Zhang, A. M. Sieuwerts, M . McGreevy, Casey G et al. “The 76-gene signature defines high-risk patients that benefit from adjuvant tamoxifen therapy,” *Breast Cancer Research Treatment*, vol. 116, no. 2, pp. 303–309, 2009.

[175] S . Loi, B Haibe-Kains, C . Desmedt, Wirapati P et al. “Predicting prognosis using molecular profiling in estrogen receptor-positive breast cancer treated with tamoxifen,” *BMC Genomics*, vol. 9, no. 239, 2008.

[176] K. Chen, L. Liu: Validating and Refining Clusters via Visual Rendering. Gene Expression Data of the Genomic Resources, *International Conference on Data Mining (ICDM),* pp.501-504, 2003.

[177] M. Halkidi, Y. Batistakis and M. Vazirgiannis, Clustering Validity Checking Methods: Part II. SIGMOD Record, Vol.31, No.3, pp.19-27, 2002.

[178] U. Scherf, et al, A Gene Expression Database for the Molecular Pharmacology of Cancer, Nat Genet, Vol.24, pp.236-44, 2000.

[179] Y. Liu, T. Özyer, R. Alhajj and K. Barker, “Multi-objective Genetic Algorithm based Clustering Approach and Its Application to Gene Expression Data,” Proc. of the International Conference on Advances in Information Systems, Springer-Verlag, Oct. 2004.

[180] J.A. Hartigan, Clustering Algorithms: New York: John Wiley and Sons, pp.353. 1975.

[181] K.Y. Yeung, et al, Model-based clustering and data transformations for gene expression data, Bioinformatics, Vol.17, pp.977-987, 2001

[182] https://compete.imagine.microsoft.com/en-us/canada

[183] D. Sculley. Web-scale k-means clustering, *Proceedings of the 19th international conference on world wide web*, pp. 1177 – 1178, 2010

[184] R. Ng and J. Han. "Efficient and effective clustering method for spatial data mining". In: Proceedings of the 20th VLDB Conference, pages 144-155, Santiago, Chile, 1994.

[185] Coello CAC, VanVeldhuizen DA, Lamont G. Evolutionary Algorithms for Solving Multi-Objective Problems. Boston, MA: Kluwer; 2002.

[186] Osyczka A. Evolutionary algorithms for single and multicriteria design optimization. Heidelberg. Physica-Verlag; 2002.

[187] Abdullah Konaka, David W. Coitb, Alice E. Smithca, Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering and System Safety 91 (2006) 992–1007

[188] E.Segal, D. Koller. Probabilistic Abstraction Hierarchies. Advances in Neural Information Processing Systems 14 (NIPS 2001)