CHAPTER 1
INTRODUCTION

1.1 MOTIVATION

Human resources have proved to be critical to organizational success. The people in an organization are considered a crucial factor contributing to competitive advantages of organizations [Birecree and Konzelmann 1997]. Human Resources Management (HRM) function includes a variety of activities, and keys among them are deciding what staffing needs one has and whether to use independent contractors or hire employees to fill these needs, recruiting and training the best employees, ensuring they are high performers, dealing with performance issues, and ensuring your personnel and management practices conform to various regulations. Activities also include managing your approach to employee benefits and compensation, employee records and personnel policies. It should be noted that the principal functions involved with HRM are recruiting, training, ensuring or maintaining worker performances. The contribution from human resources to organization success can be categorized into productivity, service, and quality. The productivity can be measured through the amount of output per employee. The productivity of the human resources in an organization is affected significantly by management efforts, programs, and systems. The quality of products and services lead to a long term affects against the success of organizations. An expanding company can maintain and further develop its business starting with the right staffing. As demands for the product or services increase, additional manpower is needed to comply with them. The current manpower should be checked but not simply if they can comply with the demands, but it has to ensure that it can still keep its quality and standards. To maximize the contribution from human resources, HRM is indispensable and thoroughly treated. One of the HRM activities in HRM is HR planning and analysis. Executives need to estimate forces that will influence the future supply of and demand for employees. HR encompasses a broad scope in management. Modern HRM requires a perspective that differs considerably from the compliance mind-set. HRM is expected to leverage human talent within their organizations for the purpose of achieving competitive advantage [Dubois et al. 2004].

The efficiency and productivity of an organization essentially depends on knowledge workers [Drucker 1954]. The ability to integrate knowledge both across the boundaries of the firm and across disciplines and product areas within the firm is an important source of strategic advantage [Ye et al.2006]. In the age of knowledge economy, knowledge has become the most important intelligence assets for enterprises. Knowledge workers are the key strategic resources in modern organizations. HRM for knowledge workers becomes very important for the organizations in the competitive environment. In views of management of knowledge workers, training and performance maintaining can be carried out under the framework of knowledge management. Knowledge-based firms such as professional service firms (PSFs), employ individuals who are highly skilled. These firms produce, distribute and use knowledge and information as their source of competitive advantage [OECD 1996; Soo et al. 2002]. PSFs provide intangible solutions to the problems of their customer by using the knowledge of their employees [Ditillo 2004]. Studies have demonstrated that specific HR practices, which include skills, knowledge and abilities can enhance human capital [Youndt and Snell 2004], especially in the management of knowledge workers in the health service sector [Rodwell and Teo 2004]. Loss of accumulated knowledge resulting from turnover poses a potential threat to the ability of work groups. [Bordoloi and Matsuo 2001] address the human resource planning in these knowledge-intensive operations in which the management of knowledge mix, i.e., the mix of workers in different knowledge levels is the focus. The assignment of knowledge workers is a very important phase of the decision-making process, especially with respect to Research and Development (R&D) projects where performance strongly depends on human resources capabilities [Certa et al. 2009]. Yet another business is the consulting firms. Consulting is a knowledge-intensive industry in which HR is considered to be the most valuable asset [Domsch 2006].

While the conventional WKM focuses primarily on the study of strategies in attaining the goals of organization through WKM, it is difficult to predict and in particular to compare the consequences of the proposed HRM strategies. The obstacles arise due to the lack of consequence quantification. If it is possible to formulate mathematically WKM, the consequence quantification can then be realized. Consequently, the efficiency and effectiveness of HRM strategies become comparable.

The research herein introduces the notion of treating WKM in a mathematical formulation. WKM is considered in terms of optimization problems. The management goals are represented by the so-called objective functions. The management limitations and scopes are transformed into optimization constraints. The time-dependent nature is also included in the formulation. Accompanying the introduced mathematical framework of WKM is the quest for the solution tool. The selected solution tool is expected to be flexible for different problems in HRM which are generally complicate in practice. As an initiation of quantitative WKM, only single-goal management will be considered in this research. Such a scope is not however limiting the proposed notion and thus readily extendable to multi-goal management. The solution methodology as employed herein, however, may not be applicable to the latter case and require future research.

The structure of this dissertation is as follows. The next sections provide the literature relevant to the present study. The mathematical formulation and solution methodology is then described in the subsequent chapters. To demonstrate the proposed notion and methodology, a number of numerical examples are employed. Finally, key conclusions and future works will be drawn.

1.2 MODELS FOR HUMAN RESOURCE MANAGEMENT

Modern portfolio theory, originated in a paper by [Markowitz 1952] and [Elton and Gruber 1997], states that an investor should not select assets due to only characteristics that are particular to the assets but she/he need to consider how each asset co-moved with all other assets. There is also a research [Elton and Gruber 1997] focusing on the long term optimization of asset replacement in energy infrastructure.

The western energy distribution system, distribution companies expect a large number of assets to fail because they reach their end of life in 10 to 30 years from now. Those failed assets will have to be replaced, which would potentially double the total workload in this 20 year period. Unfortunately, half of the work force is expected to retire in the next 10 years. Therefore the system dynamics model was being used which is based on real data from a distribution company. We determined that an increase in the number of failures is to be expected in the coming 10-20 years, even under a cost-optimal replacement strategy. However, because of employee shortages, this cost optimal strategy cannot be executed with the current Human Resource policies, thus worsening the problem. Hence, optimal strategies would not only have to define asset replacement programs but also to determine a matching human resource policy. [Jappelli 1999] defines another study which is about the age-wealth profile and the life-cycle hypothesis. The report states that at the individual level the life-cycle hypothesis predicts that wealth increases up to retirement, and declines smoothly thereafter. Life-span uncertainty and health hazards reduce the optimal rate of wealth dissimulation during retirement but do not change the basic insight of the model. The other assumption of the life-cycle model is that growth takes place across generations but not over the lifetime of a single individual, so that any increase in growth shifts the earnings profile upwards, without affecting its shape. This implies that an increase in productivity growth redistributes resources from older to younger generations, inducing an increase in the aggregate saving rate.

The intangible assets are divided in 3 major parts which are human assets, organizational assets and codified assets (see [Flamholtz 2001]). The human assets consist of knowledge, abilities (skills), workability, motivation, job satisfaction and commitment .The pioneers of “human capital” research by [Becker 1975], saw health-promoting activities(“…invest in the health of employees through medical examinations, lunches or avoidance of activities with high accident and dead rates .”) as an important investment in human capital ,not less important than investment into general and specific training. The organizational assets contain constitution, policy and mission, structure, strategy, processes and culture. Organizational structures, its management and the health of employees are interrelated. Health-promoting structures have to be enlarged by an employee-orientated management style and a culture of trust in an organization the management is one the most important factors regarding health in an organization. It also focuses on workload and the chances of development for their employees (see [Bockerl 2000]). Moreover, intellectual property and intangible asset issues abound throughout the business world, touching nearly all aspects of a company, from product development to human capital, and staff functions such as legal, accounting, finance to line operations such as R&D, marketing and general management. This wide diversity of intellectual property applications and stakeholders is a leading contributor to the complexity of managing intellectual property, as each field has its own legal, regulatory and practitioner history. The value of an intangible asset is ultimately captured through commercial exploitation (either directly or indirectly through infringement damages). Exploitation, in turn, is determined by the strategic and business environment that enables or hinders commercialization [Flignor and Orozco 2004].

[Jenkins 2006] provides tangible asset model and asset-based methods which typically involve restating both assets and liabilities to their current values to arrive at a net asset value. The restatement can be done on an individual component level (discrete valuation) or collectively (collective valuation). Given the relative difficulty of individually valuing a variety of assets, such as real estate, machinery and equipment, and inventory, it is often necessary to employ valuation specialists. Collective valuation requires a single analysis, which identifies the collective value of the assets and liabilities over its. Even with asset-based models, value remains a function of expected benefits to the owners. The value of assets is generally derived from either future income-generating potential or liquidation value, depending on the circumstances at a given time**.** The integrated asset management model is a combination between tangible model and intangible model. The purpose is to identify the gaps or overlap causes and effects on both modules. The best solution and the major critical success factors which can be expressed in the integrated model are determined.

[Propoi 1978] shows that many optimization problems for educational and manpower planning models can be written in a standard dynamic linear programming form. The benefits of advanced decision support technologies in the manpower planning function are explored in [Khoong 1996]. [O'Brien-Pallas et al. 2001] review of the approaches published between 1996 and 1999 that have been used to forecast human resource requirements for nursing. The methods of analysis employed for forecasting range from descriptive to predictive and are borrowed from demography, epidemiology, economics, and industrial engineering. It has been pointed out that simulation models offer the most promise for the future. [Jaffry and Capon 2005] compare qualitative and quantitative methods to incorporate risk in manpower planning, using the case study of UK Naval Services. The methods provide complementary insights in forecasting risk, and thus, offer an improvement over the use of a single method. [Wang 2005] reviews the applications of operations research (OR) in workforce planning (WP) to search for possible methodologies applicable in constructing a Training Force Sustainment Model (TSFM). TFSM is envisaged as a tool to help Training Command-Army (TC-A) in identifying the critical resource and planning issues to meet the training demand effectively and efficiently. [Golec and Kahya 2007] presents a comprehensive hierarchical structure for selecting and evaluating a right employee. The structure can systematically build the goals of employee selection to carry out the business goals and strategies of an organization, identify the suitable factor and measure indicators, and set up a consistent evaluation standard for facilitating a decision process. The process of matching an employee with a certain job is performed through a competency-based fuzzy mode. Resource Capacity Planning (RCP) Optimizer has been studied in [Gresh 2007]. RCP applies supply chain management techniques to the problem of planning the needs of IBM for skilled labor in order to satisfy service engagements, such as consulting, application development, or customer support. [Saaty et al. 2007] address the determination of the alternatives from existing measurements such as the range for the number of employees needed and the salaries required for various jobs. They show that the combined Analytic Hierarchy Process (AHP) and Linear Programming (LP) model is capable of solving hiring problems involving synergy, such as when two persons with different complementary skills work as a team. [Martin and Puterman 2009] describes a linear programming hierarchical planning model that determines the optimal number of nurses to train, promote to management and recruit over a 20 year planning horizon to achieve specified workforce levels. Age dynamics and attrition rates of the nursing workforce are key model components. The model was developed to help policy makers plan a sustainable nursing workforce for British Columbia, Canada. [Shan et al. 2010] propose an optimal model of human resource allocation at the planning stage of software projects. The model realized the goal of assigning different team members to different tasks at the least cost under the conditions of satisfying the resource demand of tasks and the constraint of effort of different stages, which is based on accurate estimation by the information available at the planning stage. [Weng et al. 2010] propose a multiproject human resource allocation method which designs a group of input and output indicators which can reflect the performance of human resource allocation. The method mainly uses data envelopment analysis model to evaluate the project's schedule of the parallel multiple projects, then dynamically adjusts human resource allocation and improves the performance of multiproject to improve the resources utilization and the success rate of multi-project. [Zhang and Vue 2010] investigate the situation of organizational commitment and work performance of universities teachers in Hebei province by sampling survey. The university teachers' competency model is based on the theory of organizational commitment. The research results provide reference for the universities in China to enhance the performance management level further.

1.3 CONVENTIONAL OPTIMIZATION AND SEARCH TECHNIQUES

The basic principle of optimization is the efficient allocation of scarce resources. Optimization can be applied to any scientific or engineering discipline. The aim of optimization is to find an algorithm, which solves a given class of problems. There exist no specific method, which solves all optimization problems. The various conventional optimization and search techniques available are discussed as follows:

1.3.1 Gradient-Based Local Optimization Method

When the objective function is smooth and one need efficient local optimization, it is better to use gradient based or Hessian based optimization methods. The performance and reliability of the different gradient methods varies considerably. These gradient methods search for minimum and not maximum. Several different methods are obtained based on the details of the algorithm. The steepest descent method provides poor performance. As a result, conjugate gradient method can be used. If the second derivatives are easy to compute, then Newton's method may provide best results. The secant methods are faster than conjugate gradient methods, but there occurs memory problems. Thus these local optimization methods can be combined with other methods to get a good link between performance and reliability.

1.3.2 Random Search

Random search is an extremely basic method. It only explores the search space by randomly selecting solutions and evaluates their fitness. This is quite an unintelligent strategy, and is rarely used by itself. Nevertheless, this method sometimes is worth being tested. It doesn't take much effort to implement it, and an important number of evaluations can be done fairly quickly. For new unresolved problems, it can be useful to compare the results of a more advanced algorithm to those obtained just with a random search for the same number of evaluations. Nasty surprises might well appear when comparing for example, genetic algorithms to random search. It's good to remember that the efficiency of GA is extremely dependant on consistent coding and relevant reproduction operators. Building a genetic algorithm, which performs no more than a random search, happens more often than we can expect. If the reproduction operators are just producing new random solutions without any concrete links to the ones selected from the last generation, the genetic algorithm is just doing nothing else that a random search.

Random search does have a few interesting qualities. However good the obtained solution may be, if it's not optimal one, it can be always improved by continuing the run of the random search algorithm for long enough. A random search never gets stuck in any point such as a local optimum. Furthermore, theoretically, if the search space is finite, random search is guaranteed to reach the optimal solution. Unfortunately, this result is completely useless. For most of problems we are interested in, exploring the whole search space takes far too long an amount of time.

1.3.3 Stochastic Hill Climbing

Efficient methods exist for problems with well-behaved continuous fitness functions. These methods use a kind of gradient to guide the direction of search. Stochastic Hill Climbing is the simplest method of these kinds. Each iteration consists in choosing randomly a solution in the neighborhood of the current solution and retains this new solution only if it improves the fitness function. Stochastic Hill Climbing converges towards the optimal solution if the fitness function of the problem is continuous and has only one peak (unimodal function).

On functions with many peaks (multimodal functions), the algorithm is likely to stop on the first peak it finds even if it is not the highest one. Once a peak is reached, hill climbing cannot progress anymore, and that is problematic when this point is a local optimum. Stochastic hill climbing usually starts from a random select point. A simple idea to avoid getting stuck on the first local optimal consists in repeating several hill climbs each time starting from different randomly chosen points. This method is sometimes known as iterated hill climbing. By discovering different local optimal points, it gives more chance to reach the global optimum. It works well if there is not too many local optima in the search space. But if the fitness function is very “noisy” with many small peaks, stochastic hill climbing is definitely not a good method to use. Nevertheless such methods have the great advantage to be really easy to implement and to give fairly good solutions very quickly.

1.3.4 Simulated Annealing

Simulated Annealing was originally inspired by formation of crystal in solids during cooling i.e., the physical cooling phenomenon. As discovered a long time ago by iron age blacksmiths, the slower the cooling, the more perfect is the crystal formed. By cooling, complex physical systems naturally converge towards a state of minimal energy. The system moves randomly, but the probability to stay in a particular configuration depends directly on the energy of the system and on its temperature. In the mid 70s, Kirlpatrick by analogy of these physical phenomena laid out the first description of simulated annealing.

As in the stochastic hill climbing, the iteration of the simulated annealing consists of randomly choosing a new solution in the neighborhood of the actual solution. If the fitness function of the new solution is better than the fitness function of the current one, the new solution is accepted as the new current solution. If the fitness function is not improved, the new solution is retained with a probability.

The simulated annealing behaves like a hill climbing method but with the possibility of going downhill to avoid being trapped at local optima. When the temperature is high, the probability of deteriorate the solution is quite important, and then a lot of large moves are possible to explore the search space. The more the temperature decreases, the more difficult it is to go downhill, the algorithm tries to climb up from the current solution to reach a maximum. When temperature is lower, there is an exploitation of the current solution. If the temperature is too low, number deterioration is accepted, and the algorithm behaves just like a stochastic hill climbing method. Usually, the simulated annealing starts from a high temperature, which decreases exponentially. The slower the cooling, the better it is for finding good solutions. It even has been demonstrated that with an infinitely slow cooling, the algorithm is almost certain to find the global optimum. The only point is that infinitely slow consists in finding the appropriate temperature decrease rate to obtain a good behavior of the algorithm.

Simulated Annealing by mixing exploration features such as the random search and exploitation features like hill climbing usually gives quite good results. Simulated Annealing is a serious competitor to Genetic Algorithms (GAs). It is worth trying to compare the results obtained by each. Both are derived from analogy with natural system evolution and both deal with the same kind of optimization problem. GAs differs by two main features, which should make them more efficient. First GAs uses a population-based selection whereas SA only deals with one individual at each iteration. Hence GAs are expected to cover a much larger landscape of the search space at each iteration, but on the other hand SA iterations are much more simple, and so, often much faster. The great advantage of GA is its exceptional ability to be parallelized, whereas SA does not gain much of this. It is mainly due to the population scheme use by GA. Secondly, GAs uses recombination operators, able to mix good characteristics from different solutions. The exploitation made by recombination operators is supposedly considered helpful to find optimal solutions of the problem.

On the other hand, simulated annealing are still very simple to implement and they give good results. They have proved their efficiency over a large spectrum of difficult problems, like the optimal layout of printed circuit board, or the famous traveling salesman problem. Genetic annealing is developing in the recent years, which is an attempt to combine genetic algorithms and simulated annealing.

1.3.5 Symbolic Artificial Intelligence (AI)

Most symbolic AI systems are very static. Most of them can usually only solve one given specific problem, since their architecture was designed for whatever that specific problem was in the first place. Thus, if the given problem were somehow to be changed, these systems could have a hard time adapting to them, since the algorithm that would originally arrive to the solution may be either incorrect or less efficient. Genetic Algorithms (or GA) were created to combat these problems. They are basically algorithms based on natural biological evolution. The architecture of systems that implement GAs is more able to adapt to a wide range of problems.

1.4 A SIMPLE GENETIC ALGORITHMS (GAs)

1.4.1 General

Genetic Algorithms (GAs) [Michalewicz 1996] were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s. In contrast with evolution strategies and evolutionary programming, Holland's original goal was not to design algorithms to solve specific problems, but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems. Holland's 1975 book *Adaptation in Natural and Artificial Systems* presented the genetic algorithm as an abstraction of biological evolution and gave a theoretical framework for adaptation under the GA. Holland's GA is a method for moving from one population of "chromosomes" (e.g., strings of ones and zeros, or "bits") to a new population by using a kind of "natural selection" together with the genetics−inspired operators of crossover, mutation, and inversion. Each chromosome consists of "genes" (e.g., bits), each gene being an instance of a particular "allele" (e.g., 0 or 1). The selection operator chooses those chromosomes in the population that will be allowed to reproduce, and on average the fitter chromosomes produce more offspring than the less fit ones. Crossover exchanges subparts of two chromosomes, roughly mimicking biological recombination between two single−chromosome ("haploid") organisms; mutation randomly changes the allele values of some locations in the chromosome; and inversion reverses the order of a contiguous section of the chromosome, thus rearranging the order in which genes are arrayed. (Here, as in most of the GA literature, "crossover" and "recombination" will mean the same thing.) Holland's introduction of a population−based algorithm with crossover, inversion, and mutation was a major innovation. (Rechenberg's evolution strategies started with a "population" of two individuals, one parent and one offspring, the offspring being a mutated version of the parent; many−individual populations and crossover were not incorporated until later. Fogel, Owens, and Walsh's evolutionary programming likewise used only mutation to provide variation.) Moreover, Holland was the first to attempt to put computational evolution on a firm theoretical footing. Until recently this theoretical foundation, based on the notion of "schemas," was the basis of almost all subsequent theoretical work on genetic algorithms. In the last several years there has been widespread interaction among researchers studying various evolutionary computation methods, and the boundaries between GAs, evolution strategies, evolutionary programming, and other evolutionary approaches have broken down to some extent. Today, researchers often use the term "genetic algorithm" to describe something very far from Holland's original conception.

"GAs" have at least the following elements in common: populations of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring. The chromosomes in a GA population typically take the form of bit strings. Each locus in the chromosome has two possible alleles: 0 and 1. Each chromosome can be thought of as a point in the search space of candidate solutions. The GA processes populations of chromosomes, successively replacing one such population with another. The GA most often requires a fitness function that assigns a score (fitness) to each chromosome in the current population. The fitness of a chromosome depends on how well that chromosome solves the problem at hand. The advantages of genetic algorithm includes,

1. Parallelism.
2. Liability.
3. Solution space is wider.
4. The fitness landscape is complex.
5. Easy to discover global optimum.
6. The problem has multi-objective function.
7. Only uses function evaluations.
8. Easily modified for different problems.
9. Handles noisy functions well.
10. Handles large, poorly understood search spaces easily.
11. Good for multi-modal problems Returns a suite of solutions.
12. Very robust to difficulties in the evaluation of the objective function.
13. They require no knowledge or gradient information about the response surface.
14. Discontinuities present on the response surface have little effect on overall optimization performance.
15. They are resistant to becoming trapped in local optima.
16. They perform very well for large-scale optimization problems.
17. Can be employed for a wide variety of optimization problems.

GAs have been used for difficult problems (such as NP-hard problems), for machine learning and also for evolving simple programs. They have been also used for some art, for evolving pictures and music. A few applications of GA are as follows:

1. Nonlinear dynamical systems–predicting, data analysis
2. Robot trajectory planning

3. Evolving LISP programs (genetic programming)
4. Strategy planning
5. Finding shape of protein molecules
6. TSP and sequence scheduling
7. Functions for creating images
8. Control–gas pipeline, pole balancing, missile evasion, pursuit
9. Design–semiconductor layout, aircraft design, keyboard configuration, communication
10. Networks
11. Scheduling–manufacturing, facility scheduling, resource allocation
12. Machine Learning–Designing neural networks, both architecture and weights,
13. Improving classification algorithms, classifier systems
14. Signal Processing–filter design
15. Combinatorial optimization–set covering, traveling salesman (TSP), sequence scheduling, routing, bin packing, graph coloring and partitioning

GAs procedure starts with an initial set of randomly selected trial solutions, namely population. Each individual in the population is encrypted and referred to as a chromosome which represents a possible solution to the optimization problem. The chromosomes evolve through successive iterations, called generations. In each generation, the fitness of each chromosome is evaluated. The fitness of each chromosome reflects the potential to be the optimal solution. Each chromosome is reproduced according to its fitness value. Fitter chromosomes have higher probabilities to be selected for reproduction whereas weaker chromosomes tend to die off. The chromosome selection and reproduction are carried out in a reproduction process. The chromosomes resulting from the reproduction process form a mating pool and are collectively referred to as offspring. The offspring are later undergone genetic operations. The exploration of search space is carried out through the genetic operations where genetic operators are applied to existing chromosomes and transform them into new chromosomes. The genetic operators-derived chromosomes represent new trial solutions in the search space. The resulting chromosomes then form the new generation of population. It should be noted that GAs work in two spaces alternatively. The selection process is performed in the space of original variables while the genetic operations are done in the space of coded variables. Both spaces are referred to as the solution and coding space, respectively. The GAs search is terminated when a prescribed number of generations have elapsed. The procedure of GAs is summarized in Figure 1.1.

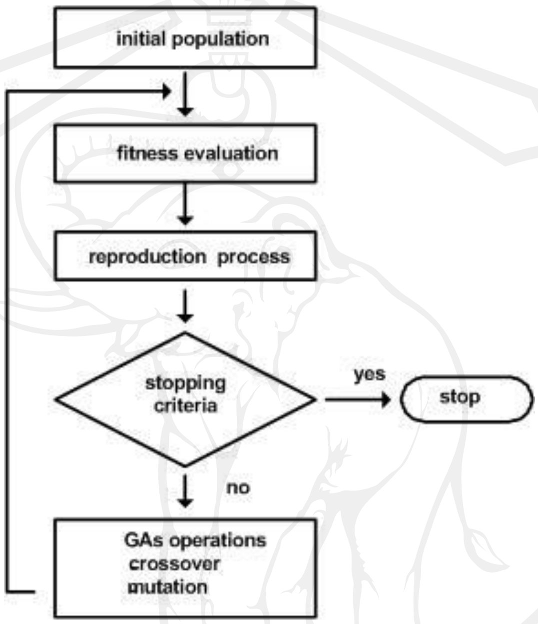

Figure 1.1. GAs search procedure.

The termination or convergence criterion finally brings the search to a halt. The following are the few methods of termination techniques:

- Maximum generations–The genetic algorithm stops when the specified number of generations has evolved.
- Elapsed time–The genetic process will end when a specified time has elapsed. If the maximum number of generation has been reached before the specified time has elapsed, the process will end.
- No change in fitness–The genetic process will end if there is no change to the population's best fitness for a specified number of generations. If the maximum number of generation has been reached before the specified number of generation with no changes has been reached, the process will end.

- Stall generations–The algorithm stops if there is no improvement in the objective function for a sequence of consecutive generations of length Stall generations.
- Stall time limit–The algorithm stops if there is no improvement in the objective function during an interval of time in seconds equal to Stall time limit.

1.4.2 Genetic Representation

The binary coding for real values will be briefly explained here. More details can be found in [Michalewicz 1996]. In context of GAs, each variable value is x_j. According to the binary coding for real values, the length of the binary strings depends on the required precision. When the domain of variable x_j is bounded by lower boundary lb_j and the upper boundary ub_j, and the required precision needs ζ_j places after the decimal point, the range of the domain of each variable should be divided into at least $(ub_j - lb_j) \times 10^{\varsigma_j}$ size ranges. The required bits l_j for the variable is then obtained from

$$2^{l_j - 1} < (ub_j - lb_j) \times 10^{\varsigma_j} \le 2^{l_j} \tag{1.1}$$

The encoding, i.e. from a real number to a binary string, and the decoding follow the relation

$$x_j = lb_j + decimal(substring_j) \times \frac{ub_j - lb_j}{2^{l_j} - 1} \tag{1.2}$$

, where *decimal* (*substring*$_j$) represents the decimal value of *substring*$_j$ for variable x_j in the solution space. The decimal value is also referred to as the decimal number. The obtained decimal number is then transformed into the binary number.

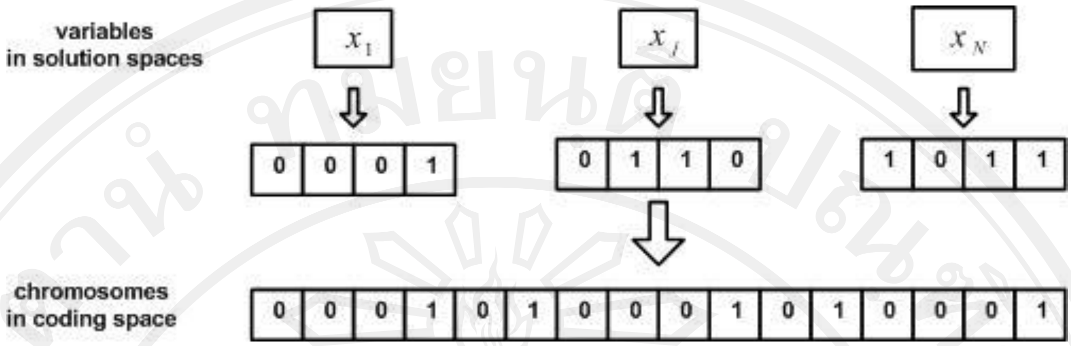

Figure 1.2. Chromosome representation using binary coding for real values [Michalewicz 1996].

As an example, the design variables are x_1 and x_2, both of which have the same domain boundaries [-1,1]. Suppose that the desired precision is three decimal places for each variable. Therefore, the required number of bits for each variable is 11 and the total length of the binary string is thus 22 bits. The decoding of a binary-coded chromosome according to this example is illustrated in Table 1.1 and Figure 1.3, respectively. η_k is a binary string representing the *k*th chromosome.

Table 1.1. Binary numbers and their corresponding decimal numbers.

Variable	Binary Number	Decimal Number
x_1	11110001001	1929
x_2	01001101110	622

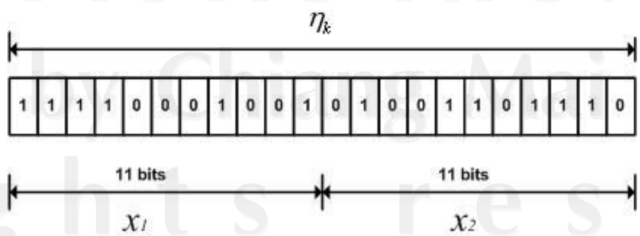

Figure 1.3. A binary-coded chromosome for real values.

1.4.3 Genetic Operators

Crossover is the process of taking two parent solutions and producing from them a child. After the selection (reproduction) process, the population is enriched with better individuals. Reproduction makes clones of good strings but does not create new ones. Crossover operator is applied to the mating pool with the hope that it creates a better offspring. Crossover is a recombination operator that proceeds in three steps:

1. The reproduction operator selects at random a pair of two individual strings for the mating.
2. A cross site is selected at random along the string length.
3. Finally, the position values are swapped between the two strings following the cross site.

The traditional genetic algorithm uses single point crossover, where the two mating chromosomes are cut once at corresponding points and the sections after the cuts exchanged. Here, a cross-site or crossover point is selected randomly along the length of the mated strings and bits next to the cross-sites are exchanged. If appropriate site is chosen, better children can be obtained by combining good parents else it severely hampers string quality.

Apart from single point crossover, many different crossover algorithms have been devised, often involving more than one cut point. It should be noted that adding further crossover points reduces the performance of the GA. The problem with adding additional crossover points is that building blocks are more likely to be disrupted.

However, an advantage of having more crossover points is that the problem space may be searched more thoroughly. In two-point crossover, two crossover points are chosen and the contents between these points are exchanged between two mated parents.

Originally, GAs were using one-point crossover which cuts two chromosomes in one point and splices the two halves to create new ones. But with this one-point crossover, the head and the tail of one chromosome cannot be passed together to the offspring. If both the head and the tail of a chromosome contain good genetic information, none of the offsprings obtained directly with one-point crossover will share the two good features. Using a 2-point crossover avoids this drawback, andthen, is generally considered better than 1-point crossover. In fact this problem can be generalized to each gene position in a chromosome. Genes that are close on a chromosome have more chance to be passed together to the offspring obtained through a *N*-points crossover. It leads to an unwanted correlation between genes next to each other. Consequently, the efficiency of a N-point crossover will depend on the position of the genes within the chromosome. In a genetic representation, genes that encode dependant characteristics of the solution should be close together. To avoid all the problem of genes locus, a good thing is to use a uniform crossover as recombination operator.

Uniform crossover is quite different from the N-point crossover. Each gene in the offspring is created by copying the corresponding gene from one or the other parent chosen according to a random generated binary crossover mask of the same length as the chromosomes. Where there is a 1 in the crossover mask, the gene is copied from the first parent, and where there is a 0 in the mask the gene is copied from the second parent. A new crossover mask is randomly generated for each pair of parents. Offspring therefore contain a mixture of genes from each parent. The number of effective crossing point is not fixed, but will average L/2 (where L is the chromosome length).

For three parent crossover, three parents are randomly chosen. Each bit of the first parent is compared with the bit of the second parent. If both are the same, the bit is taken for the offspring otherwise; the bit from the third parent is taken for the offspring.

The reduced surrogate operator constrains crossover to always produce new individuals wherever possible. This is implemented by restricting the location of crossover points such that crossover points only occur where gene values differ.

Shuffle crossover is related to uniform crossover. A single crossover position (as in single-point crossover) is selected. But before the variables are exchanged, they are randomly shuffled in both parents. After recombination, the variables in the offspring are unshuffled. This removes positional bias as the variables are randomly reassigned each time crossover is performed.

The basic parameter in crossover technique is the crossover probability (*Pc*). Crossover probability is a parameter to describe how often crossover will be performed. If there is no crossover, offspring are exact copies of parents. If there is crossover, offspring are made from parts of both parent's chromosome. If crossover probability is 100%, then all offspring are made by crossover. If it is 0%, whole new generation is made from exact copies of chromosomes from old population (but this does not mean that the new generation is the same!). Crossover is made in hope that new chromosomes will contain good parts of old chromosomes and therefore the new chromosomes will be better. However, it is good to leave some part of old population survives to next generation. In accordance with the binary representation of chromosomes, a simple binary crossover is shown in Figure 1.4.

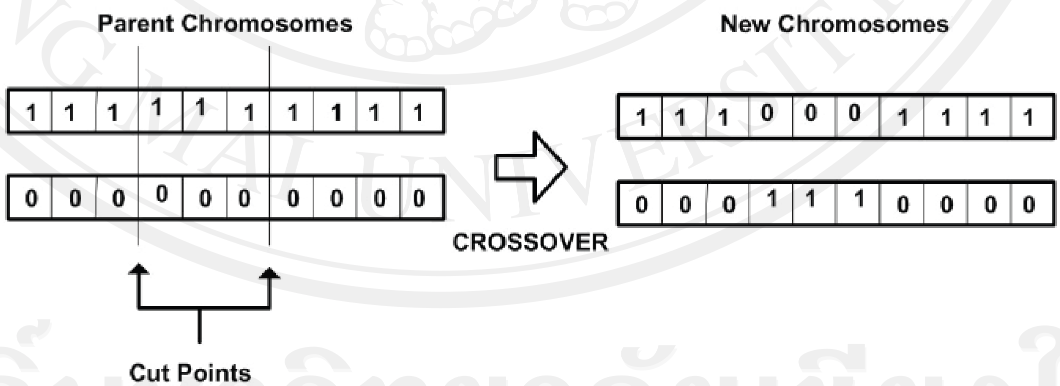

Figure 1.4. Crossover of two chromosomes.

After crossover, the strings are subjected to mutation. Mutation prevents the algorithm to be trapped in a local minimum. The mutation operation also assists the exploration for potential solutions which may be overlooked by the crossover operation. Mutation plays the role of recovering the lost genetic materials as well as for randomly disturbing genetic information. It is an insurance policy against the irreversible loss of genetic material. Mutation has traditionally considered as a simple search operator. If crossover is supposed to exploit the current solution to find better ones, mutation is supposed to help for the exploration of the whole search space. Mutation is viewed as a background operator to maintain genetic diversity in the population. It introduces new genetic structures in the population by randomly modifying some of its building blocks. Mutation helps escape from local minima's trap and maintains diversity in the population. It also keeps the gene pool well stocked, and thus ensuring ergodicity. A search space is said to be ergodic if there is a non-zero probability of generating any solution from any population state.

There are many different forms of mutation for the different kinds of representation. For binary representation, a simple mutation can consist in inverting the value of each gene with a small probability. The probability is usually taken about 1/L, where L is the length of the chromosome. It is also possible to implement kind of hill-climbing mutation operators that do mutation only if it improves the quality of the solution. Such an operator can accelerate the search. But care should be taken, because it might also reduce the diversity in the population and makes the algorithm converge toward some local optima. Mutation of a bit involves flipping a bit, changing 0 to 1 and vice-versa.

The important parameter in the mutation technique is the mutation probability (*Pm*). The mutation probability decides how often parts of chromosome will be mutated. If there is no mutation, offspring are generated immediately after crossover (or directly copied) without any change. If mutation is performed, one or more parts of a chromosome are changed. If mutation probability is 100%, whole chromosome is changed, if it is 0%, nothing is changed. Mutation generally prevents the GA from falling into local extremes. Mutation should not occur very often, because then GA will in fact change to random search. According to the chromosome representation, a binary mutation is shown in Figure 1.5.

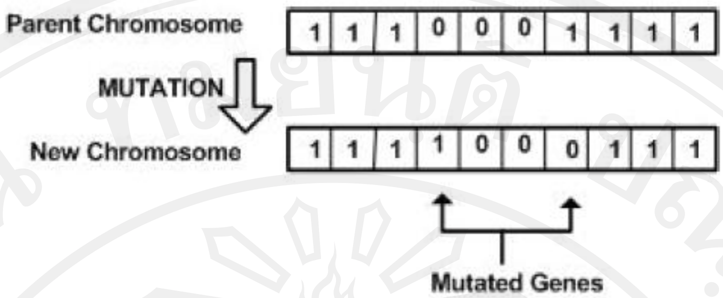

Figure 1.5. Mutation on a chromosome.

1.4.4 Reproduction Process

Reproduction in GAs is a process in which individual chromosomes are copied according to their fitness values. Copying chromosomes according to their fitness values implies that a chromosome with higher fitness value has a higher probability of contributing one or more offspring in the next generation. This operation imitates the survival of the fittest or the natural selection as used by Darwin in. Fitness in natural population is determined by the ability of a creature to survive predators, pestilence, and the other obstacles to adulthood and subsequent reproduction. Fitness in an optimization by GAs is defined by a fitness function. Based on a constrain optimization problem, the fitness function $F(\boldsymbol{x})$ of a chromosome representing a vector $\boldsymbol{x}$ of variables in the solution space is defined as

$$F(x) = \begin{cases} O_1(\boldsymbol{x}) & ; \boldsymbol{x} \text{ is feasible} \\ O_1(\boldsymbol{x}) - \sum_{j=1}^{NC} k_j v_j(\boldsymbol{x}) & ; \boldsymbol{x} \text{ is infeasible} \end{cases} \tag{1.3}$$

where $O_1(\boldsymbol{x})$ is the objective function of $\boldsymbol{x} = [x_1 \ . \ . \ . \ x_k \ . \ . \ . \ x_N]^T$, x_k is the kth designvariable, N is total number of design variables, $v_j(\boldsymbol{x})$ is the violation magnitude of the jth constrain, k_j is the penalty parameter for the jth constraint defined at each generation, and NC is total number of constraints.

An adaptive penalty scheme will be employed to handle the constraints. The improved adaptive penalty scheme shows its excellent capability in handling a very large number of constraints. This adaptive scheme is given by

$$k_j = \left|max\left(O_1^{\inf}(\boldsymbol{x})\right)\right| \frac{<v_j(\boldsymbol{x})>}{\sum_{l=1}^{NC}[<v_l(\boldsymbol{x})>]^2} \qquad (1.4)$$

, where $max(O_1^{\inf}(\boldsymbol{x}))$ is the maximum of the objective function values in the current population in the infeasible region. $<v_j(\boldsymbol{x})>$ is the average of $v_j(\boldsymbol{x})$ over the current population. . The violation magnitude is defined as

$$v_l(\boldsymbol{x}) = \begin{cases} |g_l(\boldsymbol{x})| & ; g_l(\boldsymbol{x}) > 0 \\ 0 & ; \text{otherwise} \end{cases} \qquad (1.5)$$

$g_j(\boldsymbol{x})$ is the jth constraint

The reproduction operator may be implemented in a number of ways. The easiest and well-known approach is the roulette-wheel selection. Roulette selection is one of the traditional GA selection techniques. The commonly used reproduction operator is the proportionate reproductive operator where a string is selected from the mating pool with a probability proportional to the fitness. The principle of roulette selection is a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness values. A target value is set, which is a random proportion of the sum of the fit nesses in the population. The population is stepped through until the target value is reached. This is only a moderately strong selection technique, since fit individuals are not guaranteed to be selected for, but somewhat have a greater chance. A fit individual will contribute more to the target value, but if it does not exceed it, the next chromosome in line has a chance, and it may be weak. It is essential that the population not be sorted by fitness, since this would dramatically bias the selection. The above described roulette process can also be explained as follows: The expected value of an individual is that fitness divided by the actual fitness of the population. Each individual is assigned a slice of the roulette wheel, the size of the slice being proportional to the individual's fitness. The wheel is spun N times, where N is the number of individuals in the population. On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation.

This method is implemented as follows:

1. Sum the total expected value of the individuals in the population. Let it be *T*.
2. Repeat *N* times:

   i. Choose a random integer 'r' between o and *T*.

   ii. Loop through the individuals in the population, summing the expected values,

   until the sum is greater than or equal to 'r'. The individual whose expected value

   puts the sum over this limit is the one selected.

According to the roulette-wheel scheme, the *j*th chromosome will be reproduced with the probability of

$$P_j = \frac{F_j}{\sum_{l=1}^{NPop} F_l} \tag{1.6}$$

, in which N_{Pop} is the population or sample size. The fitness value F_j is obtained from Eq.(1.3). On passing, it should be noted that GAs utilize only the numerical values of the objective function and of its associated constraints for the evaluation of the chromosome fitness.

All the advantageous features describe above make GAs readily applicable to real-world problems where the objective functions and constraints are generally implicit with respect to design variables.

1.5 CLOSURE

The previous research works reveal a number of interesting issues. Firstly, the problem of workforce planning can be cast in terms of mathematical problems. Secondly, different mathematical programming techniques have been employed as solution tools. These imply that the KWM problem can be formulated in terms of an optimization problem too. Nevertheless, due to the possible complexity in KWM, a more versatile solution tool should be utilized instead of conventional mathematical programming techniques which may not be suitable for the complicate nature. As reviewed, it has been shown that GAs is a potential tool for the computational KWM. Consequently, the adaptive penalty GAs will be selected as a solution tool in this research.

The structure of this dissertation hereinafter will be as follows. The next chapter contains the problem formulation and proposed solution methodology. A number of numerical examples are then presented to clarify and reveal the potential of the proposed study. Finally, the conclusions, research implications, and future works will be addressed at the end.