GENETIC AND HEURISTIC ALGORITHMS FOR FIRE STATION LOCATION PROBLEM

THESIS

Submitted in partial fulfillment of the requirement

for the award of degree of

M. Tech. (CSA)

Submitted By
Shourabh Sholliya
Roll No.- 601103023

Supervised By

Dr. Mahesh Kumar Sharma
Associate Professor

Mr. Singara Singh
Assistant Professor

School of Mathematics and Computer Applications
THAPAR UNIVERSITY
PATIALA – 147004

JULY 2013

CERTIFICATE

I hereby certify that the work which is presented in the thesis entitled, "**Genetic and Heuristic Algorithms for Fire Station Locaton problem**", in partial fulfillment of the requirements for the award of degree of the Master of Technology in Computer Science and Applications, submitted in School of mathematics and Computer Applications of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Mahesh Kumar Sharma** and **Mr. Singara Singh** and refers others researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other university.

Shourabh Sholliya
601103023.

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

15.7.13

Dr Mahesh Kumar Sharma
Associate Professor

Mr. Singara Singh
Assistant Professor

Countersigned by

Dr. Rajesh Kumar
Head
School of Mathematics and Computer Applications
Thapar University
Patiala

Dr.S.K. Mohapatra
Dean Academic Affairs
Thapar University
Patiala

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life.

This work would not have been possible without the encouragement and able guidance of my supervisors Dr. M.K. Sharma and Mr. Singara Singh. I thank my supervisors for their valuable time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable.

I am equally grateful to Dr. Rajesh Kumar, Associate Professor and Head, School of Mathematics and Computer Applications, for motivation and inspiration that triggered me for the thesis work.

I will be failing in my duty if I don't express my gratitude to Dr. S. K. Mohapatra, Senior Professor and Dean of Academic Affairs Thapar University, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of School of Mathematics and Computer Applications Department for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable.

ABSTRACT

The simple fire station location problem is the problem of operations research, in which we have allocate the fixed numbers of fire station among given numbers of potential sites assigning areas to them in order to minimize transportation time and cost and with several constraints taking in to consideration. Given Potential sites are those sites where situation and circumstances are feasible to allocate the fire stations. The Fire Station Location Problem is more complex problem considered in this discipline.

The thesis consists of 3 chapters. The 1 chapter is introductory and contains a brief review of literature.

In chapter 2 a genetic algorithm has been proposed for the problem of selecting upto a fixed number of sites among the given number of potential fire station site for assigning the given number of areas to them considered by Singh [25].

In chapter 3 the problem considered in chapter 2 is modified in which the capacity of each of the selected potential location is fixed that is replaced with the constraint selected fire station has the sufficient service and it will satisfy the demand of the particular area assigned to it. A small modification has been made in the algorithm proposed by Singh [25] to solve this problem.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION AND LITERATURE SURVEY

1.1 Introduction

Facility location also known as location analysis, is a branch of operations research concerning with mathematical modeling an solution of problems of facilities so as to minimize transportation costs avoid placing hazardous material near residential areas, outperform competitor's facilities, etc.

A number of case studies describing the application of facility location models to the strategic design of real life have been discussed in past, showing the growing awareness and importance that practitioners are devoting to this area, the rapid evolution of computer and communications technology has made possible the optimization of facility location in real-world production –distribution systems. In some cases, however the problem size and complexity along with the management's wish to obtain "good" solutions in reasonable time have driven researchers to develop heuristic procedures. This chapter concerns with real-life applications of the location of emergency facilities or services that are related to the public sector.

Emergency services include police patrol, ambulances, fire protection, towing and emergency repair of gas, electricity and water. Fire services is conceived of as an organized public service having the primary objective of preventing fires from occurring and reducing the loss of life and property due to fires. Fire protection services and fire service relate activities are conducted in relation to both the characteristics of the spatial environment as well as to the socio-economic and demographic characteristics of the population. Individually or in combination, spatial, social and economic factors, contribute to the incidence of fire and to the response capability of a fire protection service. These factors, to a large extent, determine the nature of fire hazards and influence of fire stations and the effective delivery of services.

Spatial considerations are primary to planning fire services, in order to determine the distribution of fire stations and their specific locations given social and economic needs and the existing communication and transportation networks. Non-spatial aspects, ranging from the nature of the fire to that of the fire service itself, its resource inputs, organizational set-up, and resource allocations and deployment policies for the fire service delivery, are also involved. While most features of the spatial environment are relatively static, example the urban layout, non-spatial aspects are continuously changing. Furthermore, the spatial and non-spatial environment which presents significant constraints on, and potential solutions for, effective and efficient urban fire protection.

1.2 Multi-Objective Optimization

As the name indicate when any problem has more than one objective function that have to be minimized or maximized with several constraints known as multi-objective problem. The terminology and concepts that are necessary to describe and understand the process and the steps employed to formulate the model are the important aspects in the multi-objective problem.

Multi-objective optimization is defined *"a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. Hence, the term "optimizes" means finding such a solution which would give the values of all the objective functions acceptable to the designer/decision maker."*

This model can be formulated as following.

Optimize $f(x) = (f_1(x), f_2(x), f_3(x), f_4(x) \ldots\ldots\ldots, f_k(x))$

Subject to $g_j(x) \leq, =, \geq b_j$, $j = 1,2,3\ldots\ldots,m$

$X = (x_1, x_2, x_3, x_4, \ldots\ldots, x_n)^T$, $X \geq 0$

Where $f(x)$ is the objective function to optimize $f_1(x), f_2(x), f_3(x), f_4(x)\ldots\ldots\ldots$, $f_k(x)$ are k number of distinct objective functions subject to m constraints. X is a vector consists of decision variables $x_1, x_2, x_3, x_4, \ldots\ldots, x_n$.

One basis to determine the performance of allocated fire station at a particular site is the response time or the speed. The response time is the time duration at which the fire station performed there action at the site and when a call it received from the assigned area. It is very important to allocate a fire station from where the shorter response time can be achieved because shorter the response time results in less property damage for the client. One of the standards used by insurance service offices in their rating of cities according to their fire protection capabilities is the distance between customers and fire station.

One of the principle focuses to allocation of fire station response time in to consideration. To get shorter response time imposes greater resources and equipments requirements on the fire station, which increase operating cost .Thus, it is reasonable to use cost and time standards in the design of fire station allocation problem. Researcher believed that a multi-objective programming model is the best solution to solve the fire station allocation problem and it is a powerful technique that has found wide range of applications in many disciplines [1].

The multi –objective model have more than one objective with several constraints. Some of the early scientist proposed that first make the list all the objectives functions of the problem and then treat the problem as single-objective model by picking one of these objectives at a time. Now solve the resultant problem and repeat these steps with second objective as the single objective in the problem and make the list of all output solution at the last select the solution from all solutions obtained because that is “best” solution because it satisfy all the objective functions. This approach have at least two disadvantages, first, it converts the problem into a large number of linear programming problems (i.e., one for each objective) which have to be solved that is increase the time and overhead and second and most critical is that the resultant solution as achieved at the end may very well not represent that solution that best satisfies all the objectives (the large the problem, the more likely that this is so). Thus, we required another approach that is more flexible, efficient and ease of use and easy to implement.

1.3 Non-Dominated or Efficient Solution

As we studied the various literature given by researchers there does not exist a single solution that concurrently optimizes each and every objective for a given multi-objective optimization problem with several constraints. In such cases there exists infinite number of solutions that are known as Pareto optimal solutions and the objective functions are said to be conflicting. A solution is called non-dominated, if none of the objective functions can be improved in value without impairment in some of the other objective values. Generally it has seen without the additional preference information, all Pareto optimal solutions can be considered mathematically equally good.

Non-Dominated solution can be define as a set of solutions is said to be efficient if there exists no solution that is superior to it with respect to at least one objective function but is not inferior to it with respect to any of the objective functions.

Lets us assume x_1 and x_2 are two solutions for any given multi-objective problem, now these can have any of two possibilities- the first one is that one dominates the other and second possibilities is that one non-dominates the other. Without the loss of generality, a solution x_1 dominates x_2 in the minimization problem if and only if the following two conditions are satisfied [22].

$$\forall i \in \{1,2,.........., N_{obj}\} : f_i(x_1) \leq f_i(x_2)$$
$$\exists j \in \{1,2,..........., N_{obj}\} : f_j(x_1) \leq f_j(x_2)$$

Where, $f(x_1)$ and $f(x_2)$ are the two objective functions.

Case 1: If any of the above conditions is violated, the solution x_1 does not dominate the solution x_2.

Case 2: If x_1 dominates the solution x_2, x_1 is called the non-dominated solution within the set $\{x_1, x_2\}$.

1.4 Heuristic Approach

Heuristic is a rule of thumb that probably leads to a solution. This search technique basically an AI-Search technique that play a major role in search strategies because of exponential nature of the most problems. The number of alternatives can be reducing from an exponential number to a polynomial number by using the Heuristics algorithm. A **heuristic search** has a general and a more specialized technical meaning in Artificial Intelligence. In a general sense, the term heuristic is used for any advice that is often effective, but is not guaranteed to work in every case [13].

“Heuristic” the word basically originated from the Greek root that has the meaning “to discover”. In case of optimization of, the problems are encountered because of the highly complex nature. The real life optimizations problems are very complex problems so regular algorithms are mostly ineffective on these problems but there are other approach that may be used to find a solution to such problems. In many such cases, heuristic approach has proven capable of providing *acceptable* solutions. Thus, in such cases the employment of heuristic procedures or heuristic programming prefer than algorithm based approaches.

Heuristics are rules of thumb that are developed through intuition, experience and judgment. A heuristic is a procedure that may lack a proof but widely used in artificial intelligence and providing good results. It is used when the inter-relationships of the decision variables are not explicitly clear but there is some confidence in understanding the output for certain input. The heuristic search is also known as informed search that cannot guarantee the optimal solution. Some time a heuristic algorithm may help to find solutions which are good, but perhaps not the best they can be. The heuristic approaches widely used to solve the real world multi-objective problems [13, 22].

1.5 Introduction to Genetic Algorithm

Genetic Algorithm (GA) is an optimization technique and widely used in computational models inspired by evolution. GA first encode potential solution of a specific problem in to binary form to on a simple chromosome-like data structure and apply different operators to these structures to preserve critical and important information. GAs are well known as function optimizer and the range of problems to which GAs have been applied are quite large. The working of GA starts with an initial population of chromosomes and then it calculates these structures and gives more chance to reproduce or to generate the next generation if it provides a better solution to target problem.

The chromosomes or individuals in the population which will produce poorer solutions give less chance to reproduce the next generation. Nature inspired to human being to developed new concepts and invention and the genetic algorithm and neural networks are example of it. GA simulates the nature and evolution process. It starts there working from initial set of population or hypotheses and generates next "generations" of solutions. Idea behind the GA has taken from the nature in which healthy individuals of generation will survive and successively grow their generation and the idea is to generate a better species based on Darwinian paradigm.

It is a robust search and optimization mechanism that is inspired by natural evolution. One point has to be noted higher fitness of a solution is better solution and based on their fitness, parents are selected to reproduce a new generation. It is a class of probabilistic optimization algorithms originally developed by Holland in 1975 [15].

The GA is a probabilistic search algorithm that iteratively transforms a set (called a *population*) of mathematical objects (typically fixed-length binary character strings), each with an associated fitness value, into a new population using the Darwinian principle of natural selection and using operations that are patterned after naturally occurring genetic operations, such as crossover (sexual recombination) and mutation.

1.6 The Basic Principle of Genetic Algorithm

Basic working principle of GA is illustrated in Figure 1.1. The first major step involved generation of a population of solutions those are encoded in binary representation and then finding the objective function, fitness function and the application of genetic operators. These aspects are described brief below by the basic algorithm of GA. As the Figure 1.1 depicts the basic working of GA operation. One generation is broken down into a selection phase and recombination phase in which each population is represented by string.

/*Basic Genetic Algorithm */

Begin

formulate initial population.

encode them.

randomly initialize population.

repeat

evaluate objective function.

selection.

apply genetic operators.

crossover.

mutation.

until stopping criteria.

End.

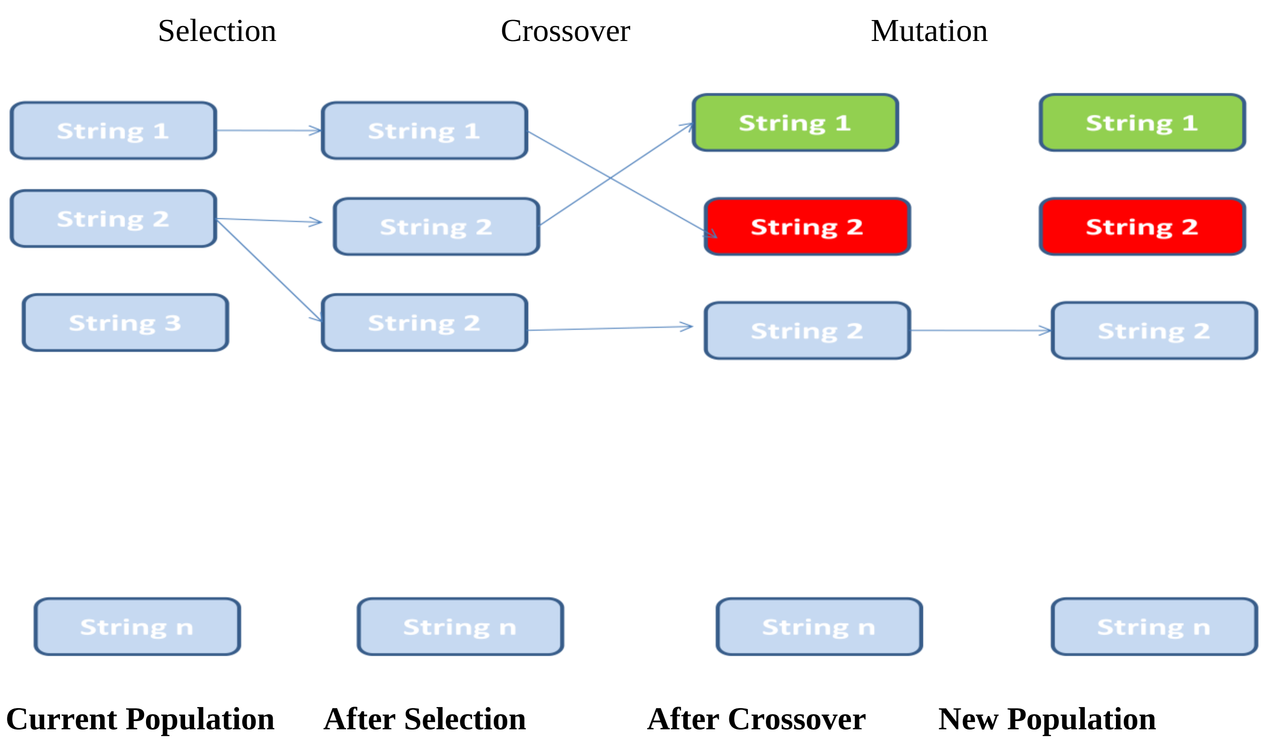

Figure 1.1: The working principle of a simple genetic algorithm

Strings are assigned into adjacent slots during selection. One of the most important characteristic of GA is the encoding of variables that describes the problem. The most common coding method is to transform the variables to a binary string or vector. GA performs best when solution vectors are binary generally. If the problem has more than one variable then a multi-variable coding is constructed by concatenating as many single variables coding as the number of variables in the problem. In the first step, initial population have to make or generated by random generators .This initial population have to encoded in binary form and the process is called encoding. This is a representation of solution vector in a solution space and is called initial solution. This ensures the search to be robust and unbiased, as it starts from wide range of points in the solution space.

Then GA has to calculate the objective function on individual members of the population and according to that the resulted value the next generation will select. In next step, the genetic operator crossover and mutation are applied on the selected population until the optimal solution has found or stopping criteria is met. The new member of the population is followed by the application of GA operator.

1.7 The Basic Operators of Genetic Algorithm

The basic operation of GAs starts with a population representing decision or design variables. The GA has the numbers of operator that may apply on population. The main operators of the GA are reproduction, crossover and mutation. These operators are used to create new population in generation. GA tries to maximize the fitness function for that it evaluates different solution vectors. Main reason to use these operators is to create new solution vectors by applying selection, crossover, mutation or alteration of the current solution vectors that have shown to be promising to provide the temporary solutions. Resultant new population is further calculated and tested till termination or until the optimal solution is met. If the optimal solution or the termination criterion is not met then the population has to be iteratively operated by the above three operators and evaluated again. A generation is the one cycle of these operations in GAs terminology. All the basic operators of genetic algorithm are discussed below.

1.7.1 Selection

The first operator that select best parents to produce the next generation in genetic algorithm and the best parents will be decide on the basis of your objective. It is the operator that makes more copies of better strings in a new population. Selection is usually the first operator applied on a population and it selects good strings in a population and forms a mating pool. That is why the reproduction operation to be sometimes known as the selection operator. The reproduction or selection of the individuals in the current population is necessary to maintain the generation of a new population. For better individuals in the generation it should select the fittest individuals from the generation. There are number of selection operators in genetic operators. The basic idea in all of them is that the above average individuals from the population are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner.

Selective pressure and diversity are two main factors that have to be considered but there is the tradeoff between selective pressure and diversity as they are strongly related to each other, since an increase in the selective pressure decreases the population diversity and if we increase the diversity decreases the selective pressure. Thus GA has to take a decision by taking both of these in consideration.

As the name suggests in selection *tournaments* are played between two solutions and the better solution is chosen and placed in the *mating pool*. Two other solutions are picked again and another slot in the *mating pool* is filled up with the better solution.

1.7.2 Encoding

The genetic algorithm starts with the initial population that can be defined as the first set of individual of the generation, those will generate the first new generation is called *Initial Population*. From the programming or implementation point of view represent each individual in the generation in the form of binary string, called *encoding* in GA and by using the binary representation of population it will be easier to implement it in to the program. Due to encoding it is easy to apply the genetic operator on them representation so that the next generation can came in to the picture. The process of encoding the solution in the form of string conveys the necessary and useful information. It provides the design alternative that is easy to transform into the program.

1.7.3 Crossover

A crossover operator is the one of the main operator used by the GA to recombine two individuals to get a better candidate for the next iteration. Due to the crossover operation a different individuals in the successive generations have found. The crossover operation is performed on two or more individuals of the previous generation and gives the new one. In reproduction a good or healthy strings in a population are probabilistically assigned a larger number of copies and a mating pool is formed in reproduction process. It is important point to note that no new strings are formed in the reproduction phase. New strings are created by exchanging information or alteration of bits among strings in the mating pool when is applied. The two strings participating in the crossover process are known as parent and the resulting strings are known as children of the next generation. The point has to take into consideration that good sub-strings from parent strings can be combined to form a better child string. The children strings produced may or may not have a combination of good sub-strings from parent strings, depending on whether or not the crossing site falls in the appropriate place with a random site. This is not a matter of serious concern because if good strings are produced by crossover and there will be more copies of them in the next mating pool that are generated by crossover process. The effects of crossover may be detrimental or beneficial that is clear from above discussion. So it can be concluded in order to preserve some of the good parents that are already present in the mating pool and all strings in the mating pool are not used in crossover.

Searching the new strings is the main responsibility of the crossover operator. Genetic operator used many crossover operators. One site and two site crossover are the most common ones adopted. Usually most crossover operators, two strings are picked from the mating pool at random and some portion of the strings are exchanged between the strings and the portion of exchange the information is also random. Crossover operation is applied at string level in which randomly selecting two strings for crossover operations. The Figures 1.2 and 1.3 illustrate the crossover operation.

A1:	**0 1 1**	1 1 1 0 0 1
A2:	**1 0 1**	0 1 1 0 0 1

Figure 1.2: Bits before crossover operation

The Figure 1.2 represents the string representation of individuals in the generation before applying the crossover operator in which the cut is randomly generated. The Figure 1.3 is complete structure of string after the crossover operation has been applied.

B1:	**1 0 1**	1 1 1 0 0 1
B2:	**0 1 1**	0 1 1 0 0 1

Figure 1.3: Bits after crossover operation

The above example is off two site crossover operation in which, a crossover site is selected randomly (shown as vertical lines). The portion left of the selected site of these two strings is exchanged to form a new pair of strings. The new strings are thus a combination of the old strings. Two site crossovers is a variation of the one site crossover, except that two crossover sites are chosen and the bits between the sites are exchanged as shown in above. One point has to be noted that one site crossover is more suitable when string length is small or the population range is small while two site crossovers is suitable for large strings of population. Hence the present work adopts a one site crossover. The underlying objective of crossover is to exchange information between strings to get a new string or new individual in the generation that is possibly better than the parents. There are following type of crossover.

1. Single point crossover

2. Two point crossover

3. Uniform crossover

1.7.4 Mutation

It is the process by which a string is deliberately changed so as to maintain diversity in the population set. Mutation is the process that randomly perturbs a candidate solution from the population.

Mutation adds new individuals in the population and it is a random way to that ultimately helps to avoid getting trapped at local optima. Mutation is an operator that provides diversity in the population whenever the population tends to become homogeneous or repeat again and again due to repeated use of reproduction and crossover operators. The chromosomes of individuals are different from those of their parent individuals this may be because of mutation process.

Mutation and crossover both operate at the bit level. When the bits are being copied from the current population to the new population of string, there is probability that each bit may become mutated. This probability is usually a quite small value, called as mutation `probability.

In the mutation process the bit zero becomes one and one becomes zero that is also called the bit inverted process. The process of mutation helps in introducing a bit of diversity to the population by scattering the occasional points. This random scattering of bits would result in a better optimal solution, or even modify a part of genetic code that will be beneficial in next operations. There might be other possibility that it might produce a weak individual that will never be selected for next operation.

The requirement for mutation is to create a point in the neighborhood of the current point in the population thereby achieving a local search around the current point in the population. The mutation is also used to maintain diversity in the population so that the homogenous population will change. Consider the given example in Figure 1.4 the following population having three four bits strings used to represent individual in the generation. In the below example mutation is performed on 3rd and 8 th bit that change the bit from 0 to 1 at 3rd position and from 1 to 0 at the 8 th position.

These three operators are simple, straightforward and easy to implement. The selection operator selects parents from population and the crossover operator recombines most promising parents together with the hope to create a better child for new population. The mutation operator alters a string locally and expecting a better child. None of these claims are guaranteed and/or tested while creating a string so it may be possible that if the bad or worst strings are created they will be removed by the reproduction operator in that will apply in the next generation and if good strings are created then they will be more chance to reproduce. These operators can be implemented in different ways and some mathematical foundations of genetic algorithms can be obtained from GA. By applying these operators on the current population creates a new population and this new population is used to generate next populations and so on and find the solutions that are closer to the optimum solution.

Offspring	**1 1 0 1 1 0 0 1 0 0 1 1**
Mutated Offspring	**1 1 1 1 1 0 0 0 0 0 1 1**

Figure 1.4: Simple mutation operation

1.8 Applications of GA

Everyone can gain benefits from Genetic Algorithms. The GA is simple, elegance as robust search algorithms as well as from their power to discover global solutions rapidly for difficult high-dimensional problems. GA is useful and efficient when

1. The search space is complicated, large, complex and poorly understood

2. When domain knowledge is less or expert knowledge is difficult to encode to narrow the search space.

3. When no mathematical analysis is available.

4. Generally traditional search methods fail

5. It can handle arbitrary kinds of constraints and objectives

There are numbers of application areas those are explained below.

Optimization GAs have been used as very good optimizer and a wide variety of optimization tasks, including numerical optimization problems such as traveling salesman problem (TSP), circuit design, job shop scheduling and video & sound quality optimization.

Machine and robot learning GA has been used for many machine learning applications that are the branch of NLP, including classification and prediction, and protein structure prediction. It has also been used in neural networks, to evolve rules for learning classifier systems, design and control robots.

Ecological models GA has been used to model many ecological phenomena such as biological arms races, host-parasite co-evolutions.

1.9 Literature Review

According to ReVelle and Schweitzer [12, 23] the numbers decision models have evolved for emergency facilities allocation problems over the past 20 years. To determine of optimal base locations for allocating fixed emergency facilities has a long history in management science/operations research literature. It is assumed that, once allocated a fire station, the emergency service would almost always be available when a request is made by client. The recognition that congestion can frequently make servers unavailable led to the development of redundant coverage models.

A new set of models has been developed and tested recently. In these models placed some explicit constraints on the availability of service within the time standards. The location set-covering problem, to fire-station locations is the simplest model and most widespread application. The fire station allocation problem is to determine the "best" base locations for allocating them so that numbers of service level objectives (cost and time) are optimized. Each allocated fire station (or fire fighting system) waits at its base location until called has arrived from its accident site, It is assumed. After providing service to its called area, the fire service returns to base location and wait for another call. Instances of successful model applications are noted in many studies, including Brandeau and Chaiken [5, 6].

There are two major categories of analytical models have been developed to analyze the following problems: queuing analysis and mathematical programming. Queuing approaches, such as the hypercube model and its approximation was utilized by Larson[17, 18]. The proposed models were used to evaluate a large variety of output measures like as vehicle utilization and average travel time. Thus the use of these models as subroutines in optimization heuristics has been suggested in the congested median location model by Berman and Larson [4].Each iteration of these optimization heuristics requires evaluating either the hypercube model or an approximation. The Jarvis' mean service calibration method played an important role for spatially distributed queuing systems in which a call service time depends on call location. Since this method evaluates the hypercube model (or its approximation) iteratively until a consistent set of average service times are determined, the method can be computationally expensive [16].

Other mathematical programming approaches like as the maximal covering model Church et al. [7], the maximum expected covering model by Daskin [8], the extended version of the maximal expected covering model by Batta et al. [2], they have been used to determine the base location for allocating a facility so that either the average travel distance is minimized or the demand covered is maximized. ReVelle et al. [23] presented an optimization model that allows unequal vehicle utilization; various call types, service times that depend on call location and the stochastic travel times recently. After applying the proposed model they validated their model using data from the Tucson Emergency Medical Services. While those studies have been more general, real and practice in nature that deals with mainly emergency vehicle basis and there are other studies which directly deals with the allocating the fire stations.

Doeksen and Oehrtman [10] presented the paper in 1976 in which he used the general transportation model to derive optimum locations sites for allocating the fire station under alternative objectives. He solved the problem many times by using different objectives.

They considered the three objectives that were used to minimize the response time to reach a fire when they received the call, minimizing total mileage for a fire station, and save as many property they can by maximizing protection per dollar's worth of burnable property. As a result, the optimum base locations were obtained to allocate the fire station over there.

Another concept that has been used in many fire location studies is the maximum covering distance concept. Plane and Hendrick [21] used a hierarchical objective function for the set-covering problem of the fire-station location problem. They simultaneously minimize the number of fire stations and maximization of the number of fire stations within the minimum total number of stations so that reducing the annual costs for allocating the fire stations for the fire companies in Denver.

Sanli and Al- Tamimi [24] have reviewed the principles that affect the performance of the service of the fire station with regard to communications patterns within cities, spatial relationships and in particular related to location sites and allocation of fire services over there. Badri et al.[1] have presented a paper in 1998 in which a multi-objective criteria modeling approach has been used, that uses the integer goal programming to solve the fire station location problem that involves conflicting objectives incorporating both the travel times and travel distances from allocated fire stations to the assigned area.

Liu et al. [19] presented an approach in 2006 to suitably situating new fire stations by considering multiple objectives of maximizing the coverage of routes for the fire station, achieving a reasonable distance between fire stations and maximizing the areas that can be served by allocated fire station within 6 minutes by using GIS and ANT algorithm. Beraldi and Bruni [3] in 2009 have formulated and solved a probabilistic model of determining the optimal location of facilities in congested emergency systems.

The *genetic algorithm* usually abbreviated nowadays to GA that was first time used by John Holland [15]. Holland writes a book on *Adaptation in Natural and Artificial Systems* in 1975 that become base for new research in of field of GA and application. The term *evolutionary computing* or *evolutionary algorithms* (EAs) used by many people in order to cover the developments of the last 10 years. Originally the genetic algorithm is heuristic algorithm that is inspired by the nature that has the ability to fulfill the requirement of wider range of peoples. Some other scientists with different backgrounds were also involved in developing similar ideas. Fogel in the USA developed their idea and they called it *evolutionary programming.* They used the concept of mutation and selection in their idea that was based on Darwinian Theory of evolution and some promising results were obtained by them.

Holland's work is on adaptive systems in which he has given optimization fairly small place and the majority of research on GAs tends to assume this is their purpose. DeJong [9] was the first person who initiated his interest in optimization, has cautioned that this emphasis may be misplaced in a paper in which he contends that GAs are not really function optimizers, and that this is in some ways incidental to the main theme of adaptation. Nevertheless, using GAs for optimization is very popular, and frequently successful in real applications.

Dwivedi et al. [11] have developed a flexible approach for solving the travelling salesman problem that is also known as TSP by using genetic algorithm. Travelling salesman is one of the example of NP-complete and it has been found to solve it polynomial time. There are many approaches had been suggested to solve this problem using classical methods such as integer programming and by very well known study of graph theory algorithms each of them have their different sucess. They have found a solution which includes a genetic algorithm implementation to give a maximal approximation to the problem as can with the minimum cost the sales man can visit the different location. They have used the genetic algorithm with crossover as a main basic operator to solve the problem of TSP. After lots of attempt to discover an appropriate and efficient crossover operator they have been presented a strategy (new crossover operator) that find the nearly optimized solution to these NP-complete type of problems, using the proposed crossover technique for genetic algorithm that generates most appropriate solution to the TSP. They conclude their research by comparing the efficiency of their proposed crossover operator with some existing crossover operators. The morel of the story in their proposed work intends to compare the efficiency of the new crossover operator with some existing ones and presented the results.

1.10 Present work

In chapter 2 an algorithm has been proposed that is the combination of genetic and heuristic approach to solve the multi-objective fire station location problem. It will allocate the given numbers of fire station and assigned areas to them in such a way that each allocated fire station will satisfy each and every constraint.

In chapter 3 the multi-objective fire station location problem proposed by Singh [25] have been modified and a heuristic algorithm has been proposed to solve it. In which N fire station have been allocated and assign areas to them with considering supply and demand and other given constraints.

CHAPTER-2

A GENETIC APPROCH FOR FIRE STATION LOCATION PROBLEM

In this chapter, the problem with the objectives to minimize the cost and duration of covering the areas from the locations identified for allocating the fire stations considered by Singh [25] reviewed and a GA is proposed to find the set of efficient solutions.

2.1 Formulation of the Problem

There are M areas, N potential fire station locations, K is the number of fire station sites where the fire station are to be allocated. The M areas are to be assigned to K unique fire station sites which are for task. Each selected potential location may cater upto of L areas.

Where the notations used explained as following.

M: The numbers of areas those are assigned to allocated fire stations.

N: The potential sites or locations where the fire stations can be allocated.

K: Numbers of fire stations that have to be allocated.

c_{ij} and t_{ij} $(i=1, 2, \ldots,N ; j=1, 2, \ldots,M)$: Represent the units of cost and time of covering the area j from location i.

C_i: Represents the setup cost of fire station at site i. The setup cost is fixed cost that contains building construction, fire station installation, Labor charge, Material cost etc. These are one time investment.

x_{ij}: The binary variable assuming value 0 or 1.If area j is assigned to fire station i then x_{ij} will be 1 otherwise it is 0.

y_i: is the binary variable assuming value 0 or 1 accordingly. If potential site i is selected then it will be 1 otherwise 0.

T: Represents the total time duration of meeting requirements of all the areas from their assigned fire stations.

C: Represents the total cost of meeting requirements of all the areas from their assigned fire stations.

L: Represents the maximum numbers of area that can cater by a allocated fire station.

The mathematical formulation of this problem is as given below. The two objective functions of cost and time that have to minimize are represented by equation 2.1 and 2.2 respectively.

$$C = \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij} x_{ij} \quad (2.1)$$

$$T = \max\{t_{ij} x_{ij} : i = 1,2,..., N;\ j = 1,2,..., M\} \quad (2.2)$$

There are numbers of constraints that are also be satisfied for this problem and are given below.

$$\sum_{i=1}^{N} Y_i = K \quad (2.3)$$

$$\sum_{j=1}^{M} x_{ij} y_i \leq L(i = 1,2,........, N) \quad (2.4)$$

$$\sum_{i=1}^{N} x_{ij} = 1(j = 1,2,........., M) \quad (2.5)$$

$$x_{ij} - y_i \leq 0(i = 1,2,...N;\ j = 1,2,....., M) \quad (2.6)$$

$$x_{ij} y_i = 0 or 1(i = 1,2,.., N;\ j = 1,2,..., N) \quad (2.7)$$

Note that the objective function of cost and time given by eqs.(2.1) and (2.2) are not according to any priorities. The constraint (2.3) ensures that *K* sites are assigned for locating fire stations at them, whereas constraint (2.4) ensures that the capacity of each of the selected potential locations is at the most *L* areas. The constraint (2.5) and (2.6) every area is assigned to unique fire station which is selected for allocating a fire station. After the applying the solution procedure a set of efficient solutions are obtained for the problem given by (2.1) and (2.2). It will make the list of all efficient solutions. A solution $(\overset{-(1)}{X}, \overset{-(1)}{Y})$ will be called the 1st efficient solution if it is the optimal solution of the problem with the minimization of C and T as the first and second prioritized objective respectively. Similarly a solution $(\overset{-(2)}{X}, \overset{-(2)}{Y})$ will be called the 2nd efficient solution if no efficient solution $(\bar{X}, \bar{Y})$ of the given problem exists satisfying the conditions.

(i) $C(\overset{-(1)}{X}) < C(\bar{X}) < C(\overset{-(2)}{X})$ and (ii) $T(\overset{-(1)}{Y}) > T(\bar{Y}) > T(\overset{-(2)}{Y})$. Similarly the 3rd and its subsequent efficient solution are defined in the same way as done for the 2nd efficient solution.

2.2 Solution Procedure

Multi-objective fire station allocation problem has been solved by applying the combination of genetic and heuristic approach. The problem has N potential sites available where the circumstances and situation are feasible to allocate the fire stations, allocate K numbers of fire stations and then assign M areas to them. The given numbers of fire station have to be allocated in such way that total transportation time and cost of fulfilling the requirements of different areas assigned to them can be minimized.

Step 1: There are N potential sites at which K numbers of fire stations can be allocated by the combination theory the total numbers of such combinations are $^{N}C_{k}$. At first, take a random initial population containing K individuals and those will be the initial population for the genetic algorithm. Each individual in the population represent a set of possible potential fire station sites and each individual in the population represented K sites on which fire station can be allocated. There are many other ways for generating initial random population.

Step 2: After taking the initial population, encode them in to binary form so that the basic operator of genetic algorithm can be applied easily.

Step 3: In the next step calculate the total cost and time to fulfilling the requirements of the all areas assigned to them for each individual in the population. To calculate cost and time of the individual in the population following heuristic approach is used.

An individual from the population is taken into the consideration. Take the first individual from the population pool of the potential fire station sites calculates the cost penalty for each of the selected potential fire station site. The cost penalty is the minimum penalty that might be incurred in case the potential location is not assigned with an area which corresponds to the least cost for the said location, that is the cost penalty CP_i for the site i is the positive difference between the least and second least cost associated with i^{th} site, that is $CP_i = c_{2i}-c_{1i}$, where c_{1i} and c_{2i} are the least and second least cost associated with i^{th} site. If the least cost occurs more than the once in a column i, then the corresponding penalty $CP_i = 0$.

After calculating the penalty select the i^{th} destination with largest cost penalty CP_i and make an assignment at the least cost cell of the selected column or fire station potential site i. In case of tie of the largest penalty between columns, select the column which has the least cost lower than the least of the other. Repeat the above steps till $M-1$ areas have been assigned to the selected potential locations keeping in view their capacities of areas. But the last area is to be allocated following the least cost rule and not on the basis of cost penalty concept. The area will be allocated to the least cost cell of the available fire station that is the potential location with available capacity. One point has to take into consideration after assigning the area, drops or remove it from further considerations.

When all the areas are assigned to their concerned fire stations site now calculate the sum of these unit costs known as total cost at the same time calculate the maximum time within which all the requirements of the assigned area from their allocated fire station can be fulfilled. To find the maximum time and total cost, first store all unit time, t_{ij} and corresponding to the minimum unit cost, c_{ij}, then retrieve the maximum unit time and sum of the costs from them. For all the individuals in the initial population repeat the above process.

Step 4: In this step generate the new population from the initial population and for that find the two best parents on which the genetic operators will be applied to generate new individuals in generation. To decide two best parents the fitness has to be measure and efficient solutions are those solutions that are found the best solution obtained so far. To calculate the best solution among them, the total cost C and time T of the individuals in the population taking the cost as the prioritized function over time function have to measure. Solutions those give the minimum cost and times are considered to be best. Take that best solutions as the efficient solutions, if a current solution is better than incumbent solution obtained so far then new one become the incumbent solution.

Step 5: All individuals are encoded in binary form separately. To decide the length in order to represent each individuals take the maximum numbers of bits required to represents individuals binary digits required to represent each individuals. For example if value that a site can take goes upto five, this number can be represented in the three binary digits. Therefore, all the potential sites for allocation can be represented within three binary digits. Hence each site in a individual is represented by using 3 digits, separately in its binary form. For example if an individual contain a sites set of 3, 4 and 5 then for each site 3 is separately represented as 011 and 4 is represented as 100 and at last 5 is represented as 101. The complete combination is represented as 011100101and repeat the above discussed procedure for all the individuals in the population. Represent each set of sites or individuals within the maximum numbers of binary digits required.

Step 6: Now apply the basic genetic operators for that selected the two parents. For that first we have to apply the selection operator, this operator select the two best parents depends upon the value of the fitness function. As our problem is of minimization thus two individuals in the generation which giving the minimum value of the fitness function are taken as parents. The fitness function is calculated by calculating addition of each c_i and then dividing total cost to the sum of total cost. Now we have to compare the value of fitness function of each individual in the new population and select the individuals which have the minimum value as the first parent. This process is repeated again to find the second parent. After getting the two best parents go to the step 7.

Step 7: After getting the two best parents apply the crossover on both parents. It is easy to apply the crossover to the binary representation of the potential fire stations sites. We performed the single bit cross over in our example that is discussed further because we have only 10 possible of potential sites that is very small so we used single point crossover is performed on the parents. There other types of crossover operators available. In crossover simply bits are exchanged between the two parent sites. The crossover cut or crossover site is generated randomly which tells the position on parents from where exchange of bits is performed that is starts from left and goes upto the entire length of the array.

Consider the example in which two best parents be 0010**10011** and 0101**11101**. Crossover cut is generated randomly, let it be 4. Therefore all the bits after or at 4^{th} position will be exchanged in between the two parents considering the array start from zero and new generation will be 0010**11101**1 and 1100**10011**. One point has to be noted crossover will be performed if only if it does not produce any undesirable results for that a check is performed on the crossover operator and if it produces any invalid code then crossover will not performed on the two parents.

Step 8: After crossover operation use the mutation operator. There different theory related to mutation operation but we used the mutation operation in each iteration and independent to crossover. As studied that the probability of mutation should keep to be minimum and in it one bit is reversed in both the parents so that produce diversity in the generations. Similar to crossover mutation cut or mutation site is randomly required. Mutation site is the position in the binary strings where the mutation has to be performed. Example of mutation already discussed in previous section. Similar to crossover operation before performing mutation on the two best individuals in the population a check is performed. This check ensures that if after mutation, any undesirable population is generated then mutation has not been performed.

All the steps have to be applied to each new generation method and make the list of all incumbent solution. If the incumbent solution is better than current solution there is no change in incumbent solution list else change the list. When all the non dominated solutions are generated stop the procedure.

2.3 Algorithm

Begin

1. Set counter ≔ 0.

2. Encoded and evaluate the initial population, P[counter] in random order.

3. While (new next_generations < maximum_generation) do

    3.1. Set I :=0.

    3.2. Repeat I<number_of_Individuals

        3.2.1. Call the **objective**(C[counter][I], T[counter][I]). /* for population, P[k] */

    3.2. Store the best solution obtained so far in efficient solution.

    3.3. Retrieve two best individuals from the C[counter] as P1 and P2.

    3.4. Apply genetic operators to strings P1 and P2.

        Crossover: Perform the crossover on P1 and P2 and crossover site is generated randomly, after that performed mutation.

        Mutation: Perform mutation on P1 and P2 at mutation_site bit is inverted in both and mutation_site is generated randomly. Only one bit change at a time.

    3.5 Set counter: = counter +1.

    3.6 Repeat the step 3.

End

2.4 Case Study

We will illustrate the case study in which the algorithm explained above by applied to a numerical problem to obtain the set of efficient solutions of the problem that is used the Table 2.1 as input while considering $M=7$, $N=5$ and $K=3$. In Table 2.1 areas are represented by rows 1-7 and potential fire station locations by columns 1-5.The units of costs c_{ij} and t_{ij} are represented by upper and lower entries of a cell (i,j) respectively of covering the area j from the fire station location i. All the steps followed by the proposed algorithm have shown below in the given case study.

Table 2.1: Representation of input data for the algorithm [25]

Areas	Fire Station Potential Sites				
	1	2	3	4	5
1	40	70	80	60	90
	2	6	8	11	10
2	30	80	20	10	100
	8	3	9	6	13
3	50	130	200	70	20
	11	9	3	13	8
4	120	130	70	80	60
	6	7	8	4	12
5	180	170	140	30	40
	10	8	11	9	2
6	170	140	130	160	50
	9	10	12	8	6
7	150	20	60	210	220
	7	11	10	4	14
SetUpCost©	**100000**	**800000**	**700000**	**300000**	**400000**

One point has to be noted the given scenario is depended on the numbers of iterations, initial population, crossover_site and mutation_site. The mutation_site and crossover_site are randomly generated hence can change the given scenario so this scenario may change. The given case study represents all the steps of the given algorithm presented by Tables 2.2 and 2.3. The case study is only give the information how the iteration performed their operation that may vary at each time of execution but final result will remain same.

Table 2.2: Representation of iterations from 1 to 3

Population	**Fitness**	**Selected Parent**	**Crossover (Random)**	**Mutation (Random)**	**New Generations**	**Incumbent Solution**
Step 1: **(1,2,3)** **(1,2,4)** **(1,4,5)**	(510,11) (400,11) (380,9)	(1,4,5) ***(380,9)*** (1,2,4) ***(400,11)***	cross_site=6 fc= 1 001 100 **101** 001 010 **100** After cross 001 100 **100** **(1,4,4)** 001 010 **101** **(1,2,5)**	mut_site=3 flag=0 No mut.	(1,4,4) (1,2,5)	**(380,9)**
Step 2: **(1,2,3)** **(1,2,5)** **(1,4,4)**	(510,11) (310,12) (540,11)	(1,2,5) ***(310,12)*** (1,2,3) ***(510,11)***	cross_site=5 fc= 1 001 01**0 101** 001 01**0 011** After cross 001 01**0 011** **(1,2,3)** 001 01**0 101** **(1,2,5)**	mut_site=3 flag=0 No mut.	(1,4,4) (1,2,5)	**(380,9)** **(310,12)**
Step 3: **(1,2,3)** **(1,2,5)** **(1,4,4)**	(510,11) (310,12)	(1,2,5) ***(310,12)*** (1,2,3) ***(510,11)***	cross_site=6 fc= 1 001 01**0 101** 001 01**0 011** After cross 001 01**0 011** **(1,2,3)** 001 01**0 101** **(1,2,5)**	mut_site=4 flag=0 No mut.	(1,2,5) (1,2,3)	**(380,9)** **(310,12)**

Table 2.3: Representation of iterations from 3 to N

Population	**Fitness**	**Selected Parent**	**Crossover (Random)**	**Mutation (Random)**	**New Generations**	**Incumbent Solution**
Step 4: **(1,2,3)** **(1,2,5)** **(1,4,4)**	(510,11) (310,12) (540,11)	(1,2,5) ***(310,12)*** (1,2,3) ***(510,11)***	cross_site=1 fc= 1 0**01 010 101** 0**01 010 011** After cross 0**01 010 011** **(1,2,3)** 0**01 010 101** **(1,2,5)**	mut_site=6 flag=0 No mut.	(1,2,3) (1,2,5)	**(380,9)** **(310,12)**
Step N:						**(380,9)** **(350,10)** **(280,11)**

In the crossover column the crossover_site is randomly generated and according to that crossover operation performed in the selected parents from the population. The fifth column represents mutation process in which mutation_site is also randomly generated. The sixth column represents the new generation after the crossover and mutation process. The seventh column, in which all list of incumbent solution, has to be maintained. The Table 2.2 represents the iteration from 1 to 3 and the Table 2.3 represents the iteration from 4 to N. At last the set of final solution is represented.

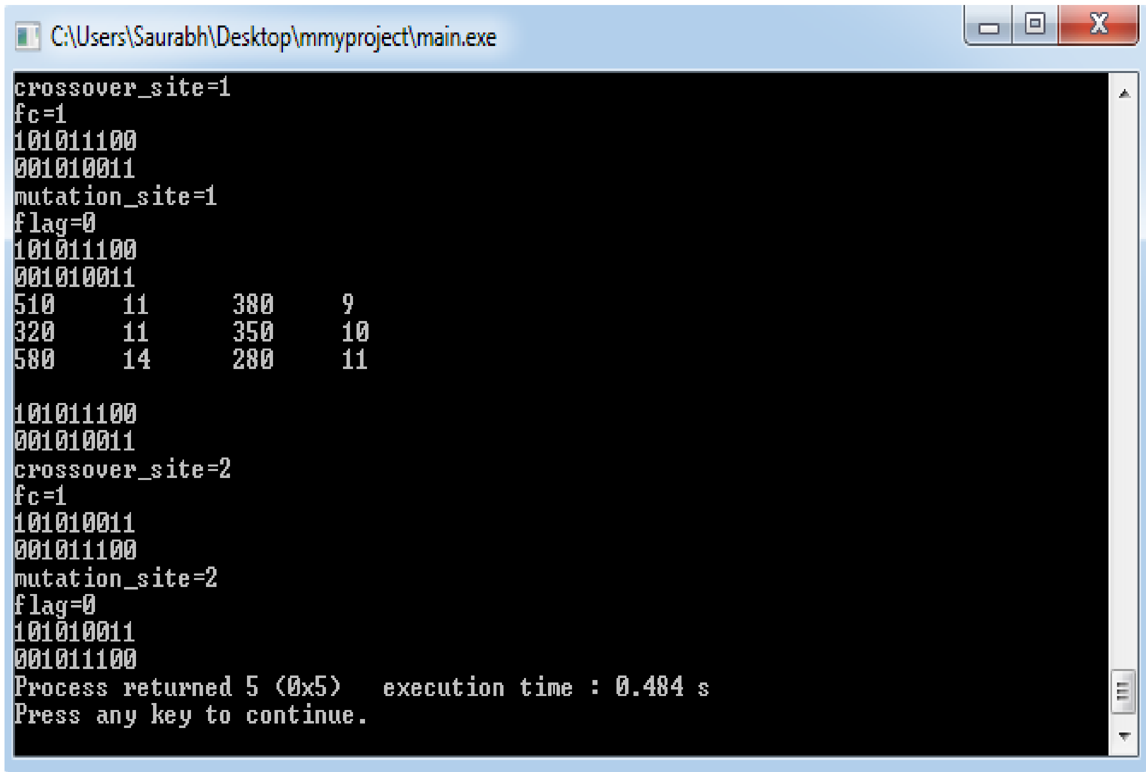

Figure 2.1: Final output

2.5 Conclusion

A GA has been proposed to obtain the efficient solution of the problem of fire station location. In which the given numbers of fire station allocated among the given number of potential location site for assigning area to them with several constraint proposed by Singh [25]. The proposed algorithm produces three efficient solutions. On comparison with the solution obtained by Singh [25]. It has been concluded that the two efficient solutions obtained by GA agreed and one efficient solution is better than produced by heuristic algorithm. The choice of the final solution depends on the priorities of the decision maker i.e. from the client perspective the solution that gives the minimum time may be the best solution but for fire station company perspective the solution that gives the minimum cost may be the best solution. The choice of solution may differ depending upon their respective budget, situation and time. The final result is illustrated by the Figure 2.1.

CHAPTER-3

A HEURSTIC APPROCH FOR FIRE STATION LOCATION PROBLEM

3.1 Introduction

The problem formulated in this chapter is the modification of the problem given by Singh [25]. The problem of selecting fixed numbers of sites, from amongst a given number of potential fire station sites for assigning the areas to them subject to several constraints and two objectives are considered. The two objective functions of cost and time have to be minimized by taking several constraints into consideration. In this problem it is assumed that the selected fire station must have sufficient services and it will satisfy the demand of the particular area assigned to it, this is known as supply and demand constraint, which is not considered by Singh [25]. Also, it is assumed that at most K sites are selected for allocating fire station to them. The two objectives functions are to be minimized by including total operating cost and the duration to cover the areas from the selected fire station sites. The first constraint ensures that allocated station has sufficient service to fulfill demand of assigned area and it is known as supply and demand constraint. The second constraint ensures that at most K sites are selected for allocating fire station to them. The constraint third and forth ensures that every area is assigned to unique fire station which is selected for allocating a fire station.

Each station has a service associated with that for a particular area. Service is maximum number of visits that a fire station can provide to particular area, that is calculated value by each fire station according to equipment, fire person, vehicles and petrol they have for a particular area. The demands are prediction of fire accidents that may happen in future for each area that can be calculated by forecasting methods. The problem is to allocate the given numbers of fire station, assigning areas to them taking supply and demand and other constraint.

A heuristic algorithm proposed by Singh [25] is modified accordingly to the new problem and set of efficient solutions of the problem are obtained. The choice of the final solution depends on priorities of the decision maker.

3.2 Formulation of the Problem

There are N potential fire station sites, M areas, K is the number of fire stations which have to be allocated. The M areas are to be assigned to a unique fire station which is selected for the task and full fill the demand of assigned area from given supply of the selected fire station. The demand is the future prediction of accidents which has been calculated with the help of forecasting methods and supply is the actual service based on equipment and recourses currently have by the particular fire station to provide the service to the assigned area.

Where the notations used explained as following.

S_{ij}: represents the maximum supply or service against accident that a fire station i can provide to area j.

D_j: Accident service or demand required by the area j.

c_{ij} and t_{ij} $(i=1, 2, \ldots, N; j=1, 2, \ldots, M)$: Represent the units of cost and time of providing service the area j from the fire station i respectively.

C_i: Setup cost of fire station at site i. The setup cost is fixed cost that contains building construction, fire station installation, labor charge, material cost etc. These are one time investment.

x_{ij}: is the binary variable assuming value 0 or 1.If area j is assigned to fire station i then x_{ij} will be 1 otherwise it is 0.

y_i: is the binary variable assuming value 0 or 1 accordingly. If potential site i is selected then it will be 1 it is 0.

T: Represents the total time duration of meeting requirements of all the areas from their assigned fire stations.

C: The total cost of meeting requirements of all the areas from their assigned fire stations.

The mathematical formulation of this problem is as given below. The two objective functions of cost and time which are to be minimize are represented by equation 3.1 and 3.2 respectively

$$C = \sum_{i=1}^{N}\sum_{j=1}^{M} c_{ij}x_{ij} \qquad (3.1)$$

$$T = \max\{t_{ij}x_{ij} : i = 1,2,...,N;\ j = 1,2,...,M\} \qquad (3.2)$$

Subject to the following constraints

$$\sum_{i=1}^{N} S_{ij}x_{ij} \geq D_j \ (j = 1,2,....,M) \qquad (3.3)$$

$$\sum_{\mathrm{i}=1}^{N} \mathrm{y_i} \geq K \qquad (3.4)$$

$$\sum_{i=1}^{N} x_{ij} = 1(j = 1,2,.....,M) \qquad (3.5)$$

$$x_{ij} - y_i \leq 0(i = 1,2,...N;\ j = 1,2,.....,M) \qquad (3.6)$$

$$x_{ij}y_i = 0 or 1(i = 1,2,....,N;\ j = 1,2,...,M) \qquad (3.7)$$

Note that the objective function of cost and time given by eqs.(3.1) and (3.2) are not according to any priorities. The new constraint (3.3) is added to the problem in chapter 2 which ensures that selected fire station has the sufficient services and will satisfy the demand of particular area assigned to it. The constraint (3.4) ensures that at most K sites are selected for allocating fire station to them. The constraint (3.5) and (3.6) each area is assigned to unique fire station which is selected for allocating a fire station. It is required to obtain the set of efficient solutions of the problem given will obtained by (3.1) and (3.7).

1^st efficient solution, a solution $(\overset{-(1)}{X}, \overset{-(1)}{Y})$ will be called the 1^{st} efficient solution if it is the optimal solution of the problem with the minimization of *C* and *T* as the first and second prioritized objective respectively. Similarly a solution $(\overset{-(2)}{X}, \overset{-(2)}{Y})$ will be called the *2^nd efficient solution* if no efficient solution $(\bar{X}, \bar{Y})$ of the given problem exists satisfying the conditions given below.

(i) $C(\overset{-(1)}{X} < C(\bar{X}) < C(\overset{-(2)}{X})$ and (ii) $T(\overset{-(1)}{X}) > T(\bar{X}) > T(\overset{-(2)}{X})$. The 3^{rd} and its subsequent efficient solution are defined in the same way as done for the 2^{nd} efficient solution.

3.2.1 Forecasting

The prediction of future is known as forecasting and no one can predict the future exactly but the experience of past and present situations can always lead him to make certain predictions about of the future of his business. The success of the businessman depends mainly on future planning based on past and present experiences. Forecasting is the establishment of future expectations by analysing the past data, or the formation of opinions.

3.2.2 Forecasting Methods

Forecasting is the technique that is widely used in those applications in which we have to predict the future and by using this prediction one make decision. Forecasting is the establishment of future expectations by analysing the past data, or the formation of opinions. Forecasting can be performed in following ways.

1. Qualitative Methods

2. Quantitative Methods

In qualitative methods use the expert opinion and collective experience to unlock the secrets of the future. Forecast is developed by a panel of experts who anonymously answer a series of questions, responses are fed back to panel members who then may change their original responses but it is very time consuming and expensive process to adopt.

In quantitative methods we generally used the *Time series forecasting*. Time series forecasting methods are based on analysis of historical data (time series: a set of observations measured at successive times or over successive periods). They make the assumption that past patterns in data can be used to forecast future data points. It can be further divide in to two categories.

Moving averages: forecast is based on arithmetic average of a given number of past data points.

Exponential Smoothing: A type of weighted moving average that allows inclusion of trends. It can be single exponential smoothing or double exponential smoothing.

a * observed value +$(1-a)$ * old forecast value

Where a is between 0 and 1

3.2.3 Formulation of Demands Using Moving Average

The demands for each area represented in Table 3.3 are generated using the moving average of forecasting. In which the past four year data (random) of demand is used to predict the next year (upcoming year) demands. The demands for each area are the maximum fire accident service required from the allocated fire station. Generation of these demands by taking of four years moving average for each area is given below. Each predicted demand is represented by D_i $(i = 1,2,..7)$.

$D_1 = (9 + 12+11+8)/4 = 10$, $D_2= (11+11+10+12)/4 = 11$

$D_3= (12+17+16+15)/4 = 15$, $D_4= (10+12+11+15)/4 = 12$

$D_5= (7+8+9+8)/4 = 8$, $D_6= (18+17+18+19)/4 = 18$

$D_7= (12+11+9+8)/4 = 10$

The above detail can be presented by Table 3.1and these calculated demands used in Table 3.3.

Table 3.1: Demand calculation for the problem

Areas	1	2	3	4	5	6	7
Demands in Year 2009	9	11	12	10	7	18	12
Demands in Year 2010	12	11	17	12	8	17	11
Demands in Year 2011	11	10	16	11	9	18	9
Demands in Year 2012	8	12	15	15	8	19	8
Predicted Demand in 2013	**10**	**11**	**15**	**12**	**8**	**18**	**10**

3.3 Solution Procedure

The algorithm presented in this chapter is the algorithm proposed by Singh [25] with small modification. The above formulated problem is a binary integer nonlinear because the objective function T given by eqs. (3.2) is nonlinear and the variables x_{ij} and y_i are binary integers. The list of all efficient solution has obtained by applying the proposed heuristic algorithm. All the steps to obtain the 1st and 2nd efficient solution are explained below.

3.3.1 Procedure to Obtain 1st Efficient Solution

The 1st efficient solution $(\overset{-(1)}{X}, \overset{-(1)}{Y})$ of the problem given by equations (3.1)-(3.7) is the optimal solution of the problem wherein the total cost C and the duration T of providing service from their assigned fire stations, are minimized with the first and second priorities respectively, subject to the constraints (3.3)-(3.7). The procedure to obtained 1st and 2nd efficient solution are explained below.

Step 1: Calculate all the possible combinations ${}^{N}C_{K}$.

Step 2: Now checks for each combination weather it is a valid or not. A combination *C=(a,b,c)* is a valid combination if there exists *at least* one station out of three stations *(a,b,c)* that can satisfy the demand for each area than station *C* is valid else not a valid combination.

Step 3: Calculate the cost penalties of the selected sites.

Now calculate the cost penalties of the selected sites. The cost penalty is the positive difference between the first least cost to the second least cost. If the first least cost and the second least cost are same for the column *i*, then the corresponding penalty $CP_i = 0$.

Step 4: Select the i^{th} fire station with largest cost penalty CP_i. Make an allocation at the least cost cell of the selected fire station *i* if its supply is equals or less than its demand otherwise consider the next largest penalty and again checks weather its supply is equals or less than its demand then make an allocation at the least cost cell of the selected fire station. If its supply is greater than to its demand then select the last penalty and fire station *i* associated with it.

In case of tie of the largest penalty between fire stations, select the fire station which has the least. If there is tie on the least cost as well, select the fire station which has least amount of time. If both the least cost and associated time are same for the i^{th} fire station, then it is an arbitrary choice for the decision maker and after allocating the area, drop it from further considerations.

Step 5: Repeat step 3 to 4 till *M-1* areas have been assigned to the selected fire station. The last remaining area is to be assigned by the least cost rule and not on the basis of cost penalty. The first efficient solution $(\overset{-(1)}{X}, \overset{-(1)}{Y})$ has been found when all the areas have been assigned to unique selected fire stations.

3.3.2 Procedure to Obtain 2nd Efficient Solution

For the 2nd efficient solution $(\overset{-(2)}{X}, \overset{-(2)}{Y})$ of the problem is obtained by solving the problem obtained from the 1st problem after dropping all those cells *(i, j)* corresponding to $t_{ij} \leq T(\overset{-(1)}{X})$. The 2nd efficient solution can be obtained by adopting the method for solving the 1st efficient solution. The 3rd and the subsequent efficient solutions are obtained in the same way as done to obtain the 2nd efficient solution. The process of obtaining the efficient solution will be terminated after encountering the problem where it is impossible to allocate at least one area to one of the selected fire station because of the unavailability of the least cost cells satisfying the time constraint or supply demand constraint failed.

All the above steps of the algorithm have to be implemented for all the ${}^{N}C_{K}$ combinations of the fire station locations to obtain different sets of efficient solutions. Finally all the solutions should be compared to calculate a final set of efficient solution. The final solution can be selected by decision maker one depending on his/her priority.

3.4 Algorithm

Notations used in Algorithms

supply1: Array that represents the supply of sation1.
supply2: Array that represents the supply of sation2.
supply3: Array that represents the supply of sation3.
supply4: Array that represents the supply of sation4.
supply5: Array that represents the supply of sation5.
demand: Array that represents the demands of all areas.
pan1: Variable that represents first penalty.
pan2: Variable that represents second penalty.
pan3: Variable that represents third penalty.
flag: Variable that represents either 1 or 0.
station1: Matrix that represents cost and time for station1.
station2: Matrix that represents cost and time for station2.
station3: Matrix that represents cost and time for station3.
station4: Matrix that represents cost and time for station4.
station5: Matrix that represents cost and time for station5.
m1: Variable that represents least cost for station1.
m2: Variable that represents least cost for station2.
m3: Variable that represents least cost for station3.
ind1: Variable that represents index of least cost for station1.

ind2: Variable that represents index of least cost for station2.

ind3: Variable that represents index of least cost for station3.

st: Variable that represents the selected station.

ind: Variable that represents index of least cost.

a: Variable that represents the selected area.

c: Variable that represents the selected area cost.

t: Variable that represents the selected area time.

<u>Algorithm validComb</u>

Inputs**:** supply1, supply2, supply3, demand.
Output: flag.

Begin

1. Initialize flag with zero.
2. For each I ∈ area.
   2.1. For each supply check
      2.1.1. If supply < demand then combination is invalid.
      Else
      2.1.2. Combination is valid and update flag with value one.

End

<u>Algorithm penalty</u>

Inputs: temp1, temp2, temp3.
Outputs: pan1, pan2, pan3, m1, m2, m3, ind1, ind2, ind3.

Begin

1. Initialize the variables
   m1 ← 999(large value), m2 ← 999, len ← 14
2. For each I ∈ len do.
   2.1. Find the first least cost and do.
      m1 ← mincost1, ind1 ← mincostind1
3. For each I ∈ len do.
   3.1.Find the second least cost and do.
      m2 ← mincost2, ind2 ← mincostind2
4. Calculate the penalty
5. Repeat the steps 1 to 4 for each station.

End

Algorithm allocat

Inputs: pan1, pan2, pan3, st1, st2, st3, ind1, ind2, ind3, suply1, suply2, suply3, suply4, suply5, demand, temp1, temp2, temp3, temp4, temp5.
Outputs: st, ind, supply, a, c, t.

Begin
1. pan =max(pan1, pan2, pan3) ,find the maximum penalty.
2. Find the station & supply corresponding to the selected penalty.
3. Call the chekDemand(ind, suply, demand)
4. If chekDemand returns true
   4.1. Allocation is possible. Find the corresponding area, cost and time.
5. If chekDemand returns false
   5.1. Find the second maximum penalty.
   5.2. Find the station & supply corresponding to the selected penalty.
   5.3. Call the checkDemand(ind, supply, demand)
   5.4. If checkDemand returns true
        then Allocation is possible. Find the corresponding area, cost & time.
        else
   5.5. Allocation is possible on the basis of last penalty. Find the corresponding area, cost & time.

End

3.5 Numerical Example

This example will illustrate the algorithm explained in section 3.3 by applying it to a numerical problem considered by Singh [25] with the inclusion the supply and demand constraint. Considering *M*=7, *N*=5 and *K*=3. In Table 3.2 areas are represented by rows 1-7 and potential fire station locations by columns 1-5.

The units of costs c_{ij} and t_{ij} are represented by upper and lower entries of a cell *(i,j)* respectively of covering the area *j* from the fire station location site *i*. Table 3.3 depict the demand and supply for the above numerical problem in which supply and demands is represented. Demands (forecasted accidents or estimated numbers of accidents that may happen in future) are represented by cell *Dj* where *j*=1, 2…, 7 and Supply (the maximum services that a fire station *i* can provide the area *j*) is depict by entries of a cell *(i, j)* of Table 3.3.

The supply and demand table is given below in which demands have been calculated using the previous discussed forecasting method. All the steps of the algorithm are given below.

Table 3.2: Input data representation for the problem [25]

Areas	**Fire Station Potential Sites**				
	1	2	3	4	5
1	40	70	80	60	90
	2	6	8	11	10
2	30	80	20	10	100
	8	3	9	6	13
3	50	130	200	70	20
	11	9	3	13	8
4	120	130	70	80	60
	6	7	8	4	12
5	180	170	140	30	40
	10	8	11	9	2
6	170	140	130	160	50
	9	10	12	8	6
7	150	20	60	210	220
	7	11	10	4	14
SetUpCost©	**100000**	**800000**	**700000**	**300000**	**400000**

Step 1: Identify all the ${}^{N}C_{k}$ combinations of potential locations.

Since at most three fire stations are to be allocated, so, all combination of the locations will be ${}^{5}C_{3}$ =10 in number.

Step 2: For each combination check valid or not.

There are total 10 combinations like (1,2,5), (1,4,5) ,(1,3,4) ,(1,3,5), (1,2,4), (1,2,3), (2,4,5),(2,3,4),(2,3,5), (3,4,5). A combination *C*=(a,b,c) is a valid combination if there exists *at least* one station out of three stations (a,b,c) that can satisfy the demand for each area than station *C* is valid else not a valid combination.(refer the supply and demand Table 3.3.)

Example 1: *C*= (1,2,5) demand of area1 *d1*=10 and supply of station1 *supply11*=9 that means cannot satisfy the demand now check for the second station2 *supply12*=10 and station5 *suply15*=16 that can fulfill the demand similarly check for the area2 having demand *d2*=11 *station1,station2, station5* having supply *supply21*=10, *supply22*=25, *supply52*=9 respectively so here also at least one station exist that can satisfy the demand *d2*.Similarly checking for each area there exist at least one station that can satisfy the demand so combination *C*= (1, 2, 5) is a valid combination.

Table 3.3: Demand and supply table for the problem

		Potential Fire Stations Sites				
Areas	Dj	1	2	3	4	5
1	10	9	10	12	15	16
2	11	10	25	15	17	9
3	15	20	18	4	21	15
4	12	9	21	11	19	20
5	8	17	13	8	8	12
6	18	15	10	11	20	18
7	10	17	18	9	7	19

Example 2: *C*= (1,2,3) demand of *area6* d_6=15 and supply of *station1, station2, station3 supply61*=15, *supply62*=10 and *supply63*=11 so in this case there no station that can satisfy the demand *d6*=18 so this combination is not a valid combination while combination (1,2,3) is valid for area1 to area5.Thus all combinations expect (1,2,3) are valid.

Step 3: Calculate the cost penalties of the selected sites.

Now calculate the cost penalties of the selected sites, i.e., the positive difference of the least and the second least costs of the selected fire station.

For combination *C* = (1, 4, 5)

Iterration1:

CP_1=40-30	CP_4=30-10 (lowest cost)	CP_5=40-20
=10	=20	=20

Since there is a tie in the largest penalty for location 4 and 5 both having penalty 20 but location 4 is given priority since least cost for the location 4 is lower than that for location 5 that is 10 is lower than 20 respectively. After selecting largest penalty before allocation we have to check supply and demand constraint that is weather this allocation can satisfy the demand or not . If this allocation can satisfy the demand than allocation is made and if it cannot satisfy the demand then we look for second largest penalty and again check for the supply demand constraint if this allocation can satisfy the demand then made a allocation otherwise check for the third largest penalty and made the allocation.

Here CP_4=20 is selected, first allocation can be made in the cell (2, 4) that means area 2 allocated to station 4 and that represent by allocation (2, 4) if it satisfy the supply and demand constraint that means area2 having demand d_2 =11 and supply of station4 *supply(4, 2)*= 17 i.e. *supply(4, 2)> =d2* (Refer Table 2.3 supply and demand). So before each allocation we have to check supply(i, j)>= dj i.e. Supply of station *i* is greater or equal to demand of area *j*. Thus the first allocation is made *allocation(2, 4)* because fulfill the supply and demand constraint i.e. area 2 is assigned station 4 with cost $c1$=10 and time $t1$=6.

Select the ith destination with largest cost penalty CP_i. Make an allocation at least cost cell of the selected column i. If there is tie between largest penalties then select the column which has the least cost lower than the least cost of other. If there is tie on least cost as well, select the column which has least amount of time associated with the least cost. In case both least cost and time are same, then choice up to decision maker. After allocating the area, drop it from the table because no need further consideration.

Table 3.4: Representation of data after allocation(2, 4) for the problem [25]

Areas	**Fire Station Potential Sites**				
	1	2	3	4	5
1	40	70	80	60	90
	2	6	8	11	10
2	-1	80	20	-1	-1
	-1	3	9	-1	-1
3	50	130	200	70	20
	11	9	3	13	8
4	120	130	70	80	60
	6	7	8	4	12
5	180	170	140	30	40
	10	8	11	9	2
6	170	140	130	160	50
	9	10	12	8	6
7	150	20	60	210	220
	7	11	10	4	14
SetUpCost©	**100000**	**800000**	**700000**	**300000**	**400000**

Area 2 is assigned to station 4 so row 2 has to be deleted for that we use -1 indicating the row has been allocated and no need for further consideration. Similarly delete the row.

Iterration2:

CP_1=50-40=10 CP_4=60-30=30 CP_5=40-20=20

Penalty CP_4=30 is selected, selected area is equal to least cost in selected station that is 30 whose area is 4 so second allocation can be *allocation(5, 4)* . *Supply(5, 4)* = 8 and d5 = 8 therefore it satisfy the supply and demand constraint i.e. *Supply(5, 4)*>= d5.Hence second allocation is made *allocation(5, 4)* i.e. area 5 is assigned to station 4 with cost c_2=30 and time t_2=9.After allocation and deletion, the Table 3.4 looks like Table 3.5.

Table 3.5: Representation of data after allocation(5, 4) for the problem

Areas	**Fire Station Potential Sites**				
	1	2	3	4	5
1	40	70	80	60	90
	2	6	8	11	10
2	-1	80	20	-1	-1
	-1	3	9	-1	-1
3	50	130	200	70	20
	11	9	3	13	8
4	120	130	70	80	60
	6	7	8	4	12
5	-1	170	140	-1	-1
	-1	8	11	-1	-1
6	170	140	130	160	50
	9	10	12	8	6
7	150	20	60	210	220
	7	11	10	4	14
SetUpCost©	**100000**	**800000**	**700000**	**300000**	**400000**

Area 5 is assigned to station 4 so row 5 has to be deleted for that we use -1 indicating the row has been allocated and no need for further consideration.

Iterration3:

CP_1=50-40=10 CP_4=70-60 =10 CP_5=50-20 =30

Penalty CP_5=30 is selected, selected area is equal to least cost in selected station that is 20 whose area is 3 so second allocation can be *allocation(3, 5)* . *Supply(3, 5)* = 15 and d_3 = 15 therefore allocation is made *allocation(3, 5)* it satisfies the supply and demand constraint .

Table 3.6: Representation of data after allocation(3, 5) for the problem

Areas	**Fire Station Potential Sites**				
	1	2	3	4	5
1	40	70	80	60	90
	2	6	8	11	10
2	-1	80	20	-1	-1
	-1	3	9	-1	-1
3	-1	130	200	-1	-1
	-1	9	3	-1	-1
4	120	130	70	80	60
	6	7	8	4	12
5	-1	170	140	-1	-1
	-1	8	11	-1	-1
6	170	140	130	160	50
	9	10	12	8	6
7	150	20	60	210	220
	7	11	10	4	14
SetUpCost©	**100000**	**800000**	**700000**	**300000**	**400000**

Area 3 is assigned to station 5 so row 3 has to be deleted for that we use -1 indicating the row has been allocated and no need for further consideration.

Iterration4:

CP_1=120-40	CP_4=80-60	CP_5=60-50
=80	=20	=10

First CP_1 =80 is selected and least cost is 40 so area is 1 and according to that allocation will be *allocation(1, 1)* but supply and demand constraint fails. *Supply(1, 1)* = 9 while demand d_1=10 so *supply(1, 1)*>= d^j fails.

Now select second largest penalty that is CP_4=20, least cost 60 so area 1 and check the supply and demand constraint. *Supply(1, 4)*=15, d_1=10 hence *supply(1, 4)*>= d_1. Thus, *allocation(1, 4)* is a valid hence made and deleted the respected row from the previous table.

allocation(1, 4), c_4=60 , t_4=11

Iterration5: Similarly

CP_1=150-120	CP_4=160-80	CP_5=60-50
=30	=80	=10

allocation(4 ,4) {A valid allocation because satisfy *supply(4,4)*>=d_4 constraint, Accepted} c_5= 80, t_5= 4

Iterration6: Similarly

CP_1=170-150	CP_4=210-160	CP_5=220-50
=20	=50	=170

allocation(6 , 5){A valid allocation because satisfy *supply(6, 5)* >=d_6 constraint, Accepted} c_6= 50, t_6=6

Iterration7:

C_1= 150 C_4=210 C_5=220

After repeating step 3 till *M-1* areas have been assigned the last area is to be assigned by least cost rule not on the basis of cost penalty followed by supply and demand constraint. The last area will be assigned to the least cost cell of available locations if it is satisfy the supply and demand constraint. If not then looking for second least cell if it is satisfy the supply and demand constraint than allocation made otherwise allocation is made on third remaining station.

Hence *allocation(7, 1)* {A valid because satisfy *supply(7, 1)>=*d_7, Accepted} with $c_7=150$, $t_7=7$

Step 4: After step 3 the procedure of calculating the penalties and allocating the areas at the least cost cells corresponding to the largest penalties followed by supply and demand constraint obtained the following allocations represented by Table 3.7.

Table 3.7: Allocation of the areas of 1st efficient solution for 1, 4 and 5

Areas	**Allocations**	**Cost**	**Time**
1	4	60	11
2	4	10	6
3	5	20	8
4	4	80	4
5	4	30	9
6	5	50	6
7	1	150	7

Thus, the first efficient solution has been calculated corresponding cost and time to cover all the areas is represented by Table 3.7. Where the cost is the submission of $c_1, c_2..., c_7$ and time to cover all the areas is *max* $(t_1, t_2... t_7)$.

Table 3.8: 1st Efficient Solution for 1, 4 and 5

Cost to cover all the areas	400 units
Time to cover all the areas	11 units
Total Setup Cost	800000 units

Table 3.9: Table representation of data after blocking t_{ij} for the problem

Areas	**Fire Station Potential Sites**				
	1	2	3	4	5
1	40	70	80	-1	90
	2	6	8	-1	10
2	30	80	20	10	-1
	8	3	9	6	-1
3	-1	130	200	-1	20
	-1	9	3	-1	8
4	120	130	70	80	-1
	6	7	8	4	-1
5	180	170	140	30	40
	10	8	11	9	2
6	170	140	130	160	50
	9	10	12	8	6
7	150	20	60	210	-1
	7	11	10	4	-1
SetUpCost©	**100000**	**800000**	**700000**	**300000**	**400000**

Step 5: The second efficient solution (next solution) of the problem for the same combination of the selected locations i.e.1,4,5 is obtained by dropping the respected cells. *(i, j)* where in the time t_{ij} >= $T(X)$ =11 units.

Step 6: Calculate the next efficient solution

All the deleted entries are represented by -1and now calculation for the second efficient solution can be done by repeating the repeat the step 3 to 6 on the up Table 3.9 as shown given above.

Iterration1: Area 3 has no choice because for area 3 the station 1 and 4 not available due to blocking so assigned to station 5 and search such areas if exist and make feasible allocation to them. Thus area has to assigned to station 5 as it fulfill the supply and demand constraint if it will not, then next efficient solution is not possible to found.

allocation(3 ,5) {A valid allocation because satisfy *supply(3, 5) >=d3*, Accepted } c_1= 20, t_1=8

After the allocation of the area has been done delete the corresponding row form the Table 2.9 similar to previous steps.

Iterration2:

CP_1=40-30	CP_4=30-10	CP_5=50-40
=10	=20	=10

allocation(2 ,4) {A valid allocation because satisfy *supply(2, 4)* $>=d_2$ constraint, Accepted} c_2= 10, t_6=6

Iterration3:

CP_1=120-40	CP_4=80-30	CP_5=50-40
=80	=50	=10

allocation(1, 1) {Not a valid allocation because do not satisfy *supply(1, 1)* $>=d_1$ constraint, Rejected and look for next largest penalty}

allocation(5 , 4) {A valid allocation because satisfy *supply(5, 4)* $>=d_5$ constraint, Accepted}c_3= 30,t_3= 9

Iterration4:

CP_1=120-40 CP_4=160-80 CP_5=90-50

=80 =80 =40

allocation(1, 1) {Not a valid allocation because do not satisfy *supply(1, 1)* $>=d_1$ constraint, Rejected and look for next largest penalty}

allocation(4, 4) {A valid allocation because satisfy *supply(4, 4)* $>=d_4$ constraint, Accepted} c_4= 80, t_4= 4

Iterration5:

CP_1=150-40 CP_4=210-160 CP_5=90-50

=110 =50 =40

allocation(1, 1) {Not a valid allocation because do not satisfy *supply(1, 1)* $>=d_1$ constraint, Rejected and look for next largest penalty}

allocation(6, 4) {A valid allocation because satisfy *supply(6, 4)* $>=d_6$ constraint, Accepted}c_5=160, t_5= 8

Iterration6:

CP_1=150-40 C_4=210 (cost not penalty) C_5=90

=110

allocation(1, 1) {Not a valid allocation because do not satisfy *supply(1, 1)* $>=d_1$ constraint, and look for next largest penalty or smallest cost}

In this step penalty can be calculated for station 4 and station 5 hence follow the least cost rule and followed by supply and demand constraint.

allocation(1, 5){A valid allocation because satisfy *supply(1, 5)* $>=d_1$ constraint, Accepted} c_6=90, t_6= 10

Iterration7:

$C1$=150 C_4=210 C_5=nil

allocation(7, 1){A valid allocation because satisfy *supply(7, 1)* $>=d_7$ constraint, Accepted} c_7=150, t_7= 7

After the procedure of calculating the penalties and allocating the areas at the least cost cells corresponding to the largest penalties followed by supply and demand constraint we obtained the following allocations represented by Table 3.10 for 2nd efficient solution.

Table 3.10: Allocation of the areas of 2nd efficient solution for 1, 4 and 5

Areas	Allocations	Cost	Time
1	5	90	10
2	4	10	6
3	5	20	8
4	4	80	4
5	4	30	9
6	4	160	8
7	1	150	7

Thus, the second efficient solution has been calculated and corresponding cost and time to cover all the areas is represented by Table 3.10. Now further blocking for t_{ij} =10 is not possible because area 1 has only one choice that is station 1 and *supply(1,1)*>= d_1 is failed so 3rd efficient solution is not possible to calculate.

Table 3.11: 2nd efficient solution for 1, 4 and 5

Cost to cover all the areas	540 units
Time to cover all the areas	10 units
Total Setup Cost	800000 units

Again using the initial Table 3.2 for a new combination (1, 3, 4) and apply the step 2 to step 6 respectively.

Step 2: The combination (1, 3, 4) is a valid combination so apply step 3.

Step 3: Calculating the penalties and allocating the areas at the least cost cells corresponding to the largest penalties followed by supply and demand constraint explained in previous steps for combination (1, 4, 5). We got the following tables. Thus, after the procedure of calculating the penalties and allocating the areas at the least cost cells corresponding to the largest penalties followed by supply and demand constraint.

Table 3.12: Allocation of the areas of 1st efficient solution for 1, 3 and 4

Areas	**Allocations**	**Cost**	**Time**
1	4	60	11
2	3	20	9
3	1	50	11
4	4	80	4
5	4	30	9
6	4	160	8
7	1	150	7

Table 3.13: 1st efficient Solution for 1, 3 and 4

Cost to cover all the areas	550 units
Time to cover all the areas	11 units
Total Setup Cost	1100000 units

In this case 2nd efficient solution is not possible to found because when we will block for t_{ij} =11 the area 3 has only one choice that is station 3 that will not fulfill the demand of area 3 i.e. $supply(3, 3) >= d_3$ fails.

Again using the initial Table 3.2 for a new combination (1, 3, 5) and apply the step 2 to step 6 respectively. We obtained the following cost & time represented by Table 3.14 for 1st efficient solution and allocations represented by Table 3.15.

Table 3.14: 1st efficient Solution for 1, 3 and 5

Cost to cover all the areas	430 units
Time to cover all the areas	12 units
Total Setup Cost	1200000 units

Table 3.15: Allocation of areas of 1st efficient solution for 1, 3 and 5

Areas	Allocations	Cost	Time
1	5	90	10
2	3	20	9
3	5	20	8
4	5	60	12
5	5	40	2
6	5	50	6
7	1	150	7

In this case 2nd efficient solution is not possible to found because when we will block for t_{ij} =12 the area 4 has only two choices left that are , station 1 and station 4 both will not fulfill the demand of area 4.

Again using the initial Table 3.2 for a new combination (1, 2, 4) and apply the step 2 to step 6 respectively. We obtained the following allocations, represented by Table 3.16 and cost & time represented by Table 3.17 for 1st efficient solution.

In this case to calculate 2nd efficient solution first we will block for t_{ij} =11 and then repeating the previous steps.

Table 3.16: 1st efficient Solution for 1, 2 and 4

Cost to cover all the areas	420 units
Time to cover all the areas	11 units
Total Setup Cost	1200000 units

Table 3.17: Allocation of the areas of 1st efficient solution for 1, 2 and 4

Areas	Allocations	Cost	Time
1	2	70	6
2	4	10	6
3	1	50	11
4	4	80	4
5	4	30	9
6	4	160	8
7	2	20	11

Table 3.18: 2nd Efficient Solution for 1, 2 and 4

Cost to cover all the areas	630 units
Time to cover all the areas	9 units
Total Setup Cost	1200000 units

Table 3.19: Allocation of the areas of 2nd efficient solution for 1, 2 and 4

Areas	Allocations	Cost	Time
1	2	70	6
2	4	10	6
3	2	130	9
4	4	80	4
5	4	30	9
6	4	160	8
7	1	150	7

2nd efficient solution and Allocation Table are represented by Tables 3.18 and 3.19 respectively.

Again using the initial Table 3.2 for a new combination (1, 2, 3) and apply the step 2 to step 6 respectively. The combination (1, 2, 3) is not a valid combination so reject it, step 2 failed. So select a new combination (1, 2, 5).

Again using the initial Table 3.2 for each remaining new combination (1,2,5) and apply the step 2 to step 6 respectively obtained the following allocations and the cost & time of each efficient solution.

Table 3.20: 1st efficient solution for 1, 2 and 5

Cost to cover all the areas	340 units
Time to cover all the areas	12 units
Total Setup Cost	1300000 units

Table 3.21: Allocation of the areas of 1st efficient solution for 1, 2 and 5

Areas	Allocations	Cost	Time
1	2	70	6
2	2	80	3
3	5	20	8
4	5	60	12
5	5	40	2
6	5	50	6
7	2	20	11

Table 3.22: 2nd efficient solution for 1, 2 and 5

Cost to cover all the areas	410 units
Time to cover all the areas	11 units
Total Setup Cost	1300000 units

Table 3.23: Allocation of the areas of 2nd efficient solution for 1, 2 and 5

Areas	Allocations	Cost	Time
1	2	70	6
2	2	80	3
3	5	20	8
4	2	130	7
5	5	40	2
6	5	50	6
7	2	20	11

Table 3.24: 3rd Efficient Solution for 1, 2 and 5

Cost to cover all the areas	540 units
Time to cover all the areas	8 units
Total Setup Cost	1300000 units

All the possible solutions for each combination obtained after following the steps of above algorithm and those shown by Table 3.25 to 3.45.

Table 3.25: Efficient solutions obtained with locations 1, 4 and 5

	Cost	Duration	Setup Cost
1st Efficient Solution	400	11	800000
2nd Efficient Solution	540	10	800000

Table 3.26: Efficient solutions obtained with locations 1, 3 and 4

	Cost	**Duration**	**Setup Cost**
1st Efficient Solution	550	11	1100000

Table 3.27: Efficient solutions obtained with locations 1, 3 and 5

	Cost	**Duration**	**Setup Cost**
1st Efficient Solution	430	12	1200000

Table 3.28: Efficient solutions obtained with locations 1, 2 and 4

	Cost	**Duration**	**Setup Cost**
1st Efficient Solution	420	11	1200000
2nd Efficient Solution	630	9	1200000

Table 3.29: Efficient solutions obtained with locations 1, 2 and 5

	Cost	**Duration**	**SetupCost**
1st Efficient Solution	340	12	1300000
2nd Efficient Solution	410	11	1300000
3rd Efficient Solution	540	8	1300000

Table 3.30: Efficient solutions obtained with locations 2, 4 and 5

	Cost	**Duration**	**Setup Cost**
1st Efficient Solution	280	11	1500000

Table 3.31: Efficient solutions obtained with locations 2, 3 and 5

	Cost	**Duration**	**Setup Cost**
1st Efficient Solution	280	12	1900000
2nd Efficient Solution	350	11	1900000

Table 3.32: Efficient solutions obtained with locations 2, 3 and 4

	Cost	**Duration**	**Setup Cost**
1st Efficient Solution	450	13	1800000
2nd Efficient Solution	510	11	1800000

Table 3.33: Efficient solutions obtained with locations 3, 4 and 5

	Cost	**Duration**	**Setup Cost**
1st Efficient Solution	480	14	1400000

After comparing all the efficient solutions shown in above Tables for all the combinations, the final solution is given by Table 3.34.

Table 3.34: Set of final efficient solutions of the numerical problem

	Cost	**Duration**	**Combination of Locations**	**SetupCost**
1st Efficient solution	280	11	2,4,5	15,00,000
2nd Efficient solution	540	8	1,2,5	13,00,000

3.6 Conclusion

The problem given by Singh [25] of allocating a fixed numbers of fire stations among the given number of potential location sites, for assigning area to them is revisited. Singh [25] considered that the capacity of each of the allocated station is fixed to at most L areas. However this restriction is replaced with that the selected fire station has the sufficient services to fulfill demands of their assigned area. The problem has been solved by applying the algorithm proposed by Singh [25] with some modification. By using this algorithm two efficient solutions are obtained. The choice of the final solution depends on the priorities of the decision maker.

REFERENCES

[1] Badri, M.A., Mortagy, A.K., Alsayed, A., 1998. A multi-objective model for locating fire stations. European Journal of operational Research 110, 243-260.

[2] Batta, R., Dolan, J., Krishnamurthy, N., 1989. The maximal expected covering location problem: Revisited. Transportation Science 23, 277-287.

[3] Beraldi, P., Bruni, M.E., 2009. A probabilistic model applied to emergency service vehicle location. European Journal of Operational Research 196, 323-331.

[4] Berman, O., Larson, R., 1982. The median problem with congestion. Computers and Operations Research 9, 119-126.

[5] Brandeau, M., Larson, R., 1986. Extending and applying the hypercube queuing model to deploy ambulances in Boston.TIMS Studies in the Management Sciences 22, 121-153.

[6] Chaiken, J., 1978. Transfer of emergency service deployment models to operation agencies. Management Science 24, 719-731.

[7] Church, R., ReVelle, C., 1974. The maximal covering locational problem. Papers of the Regional Science Association 32, 101-118.

[8] Daskin, M., 1983. A maximal expected covering location model: Formulation, properties, and heuristic solution. Transportation Science 17, 48-69.

[9] De Jong, K.A., 1993. Genetic algorithms are NOT function optimizers. In D. Whitley (ed.), Foundations of Genetic Algorithms 2. Morgan Kaufmann, San Mateo, CA, pp. 5–18.

[10] Doeksen, G., Oehrtman, R., 1976. Optimum locations for a rural fire system: A study of a major county in Oklahoma. Southern Journal of Agricultural Economics (December), 121-127.

[11] Dwivedi, V., Chauhan, T., Saxena, Sanu., Agrawal, P. Travelling Salesman Problem using genetic Algorithm.

[12] Galvano, R., ReVelle, C., 1996. A Lagrangean heuristic for the maximal cvering location problem. European Journal of Operational Research 88 (1), 114-123.

[13] Heuristic search. Internet:www.intelligence/ai-search/he-search.htm[June21, 2013].

[14] Heuristic Search. Internet: www.en.wikipedia.org/wiki/Heuristic, [June. 21, 2013].

[15] Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan; re-issued by MIT Press (1992).

[16] Jarvis, J., 1975. Optimization in stochastic systems with distinguishable servers, Operations Research Center, MIT.

[17] Larson, R., 1974. A hypercube queuing model for facility location and re-sub-areaing in urban emergency services. Computers and Operations Research 1, 67-95.

[18] Larson, R., 1975. Approximating the performance of urban emergency service systems. Operations Research 23, 845-868.

[19] Liu, N., Huang, B., Chandramouli, M., 2006. Optimal siting of fire station using GIS and ANT algorithm. Journal of Computing in Civil Engineering 20, 361-369.

[20] Multi-Objective Optimization. Internet: www//en.wikipedia.org/wiki/Multi-obj_optzation, [June. 21, 2013].

[21] Plane, D., Hendrick, T., 1977. Mathematical programming and the location of fire companies for the Denver fire department. Operations Research 25 (4), 563-578.

[22] Prakash, S, Kumar, P., Prasad, B.V.N.S., Gupta, G., 2008. Pareto optimal solutions of cost-time trade-off bulk transportation problem. European Journal of Operational Research 188, 85-100.

[23] ReVelle, C., Schweitzer, J., Snyder, S., 1996. The maximal conditional covering problem. INFOR 34 (2), 77-91.

[24] Sanli, I., Al- Tamimi, F., 1990. The spatial distribution and resource allocation of fire safety service system. Journal of King Saud University, Architecture and Planning 2, 23-41.

[25] Singh, A., Multi-Objective optimization problems with a bearing on real-life, Ph.D. dissertation, Thapar University, SMCA Dept., Patiala, 2010.