Decentralizing and Optimizing Nation-Wide Employee Allocation while Simultaneously Maximizing Employee Satisfaction

A thesis
Submitted in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Submitted by

Ashna Nawar Ahmed	**160204024**
Aniqua Tabassum	**160204085**
Subhe Tasnia Alam	**160204098**
Noushin Tabassum	**160204114**

Supervised by

Dr. Md. Shahriar Mahbub

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dhaka, Bangladesh

July 2021

CANDIDATES' DECLARATION

We, hereby, declare that the thesis presented in this report is the outcome of the investigation performed by us under the supervision of Dr. Md. Shahriar Mahbub, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Ashna Nawar Ahmed
160204024

Aniqua Tabassum
160204085

Subhe Tasnia Alam
160204098

Noushin Tabassum
160204114

CERTIFICATION

This thesis titled, **"Decentralizing and Optimizing Nation-Wide Employee Allocation while Simultaneously Maximizing Employee Satisfaction"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in July 2021.

Group Members:

Ashna Nawar Ahmed	**160204024**
Aniqua Tabassum	**160204085**
Subhe Tasnia Alam	**160204098**
Noushin Tabassum	**160204114**

Dr. Md. Shahriar Mahbub
Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Professor Dr. Mohammad Shafiul Alam
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our research supervisor **Dr. Md. Shahriar Mahbub**, Professor, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, for his patient guidance, enthusiastic encouragement and useful critiques of this research work. The degree of help, wisdom and support he gave us worked as one of the most important driving forces behind us working relentlessly throughout the research. Besides giving us guidance, he has always encouraged us to explore and experiment with new ideas, which has led us to explore new ways of thinking and solving problems.

We would like to thank **Dr. Mohammad Shafiul Alam**, Professor & Head of the Department of Computer Science and Engineering of Ahsanullah University of Science and Technology, and all out teachers, lab assistants, and friends for their moral support and helpful discussions. We would especially want to thank Mr. Ahmad Subaktagin Jabir (ID: 160204061), for his co-operation with our illustration work.

Finally, we wish to thank our family members for their support and encouragement throughout our study.

Dhaka
July 2021

Ashna Nawar Ahmed

Aniqua Tabassum

Subhe Tasnia Alam

Noushin Tabassum

ABSTRACT

The optimal allocation of workers is a critical problem for both employers and employees. For a company, it is vital that employees who are allocated to a particular designation in a particular region do not feel the need to move away from that region after remaining for a short duration. Thus it is crucial that workers be allocated at a position while fulfilling their satisfaction to the highest extent. Therefore, the goal is to allocate professionals in the area where they are most likely to be satisfied rather than allocating them arbitrarily, in order to improve the probability of long term settlement of the workers in their respective regions. Besides this, it is also essential that workers are dispersed to the highest possible extent so that all workers are not assigned to the same few developed regions while the underdeveloped regions remain empty. The two objectives of maximizing individual satisfaction while also maximizing overall dispersion can be conflicting at times. This is why Multi-Objective Optimization(MOO) concepts have been used in this thesis in order to solve this problem. Neural Networks(NN) have also been used in order to predict satisfaction, which is the first phase of solving this problem. A mathematical equation has also been proposed that measures how well dispersed the employees are. NSGA-II algorithm has been used as part of an optimization framework in order to allocate employees in an optimized manner that has maximized both of our contradicting objectives: increasing satisfaction and dispersion. We have optimally allocated 36 doctors to 7 cities while maximizing their satisfaction. The accuracy of the satisfaction model is 51.30%.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Problems

Non-optimal worker allocation in different cities of a country is a crucial problem to be solved. We have approached this problem with multi-objective optimization. Our primary goal was to allocate professionals in the area where they were most likely to be satisfied rather than allocating them arbitrarily, to improve the probability of long-term settlement of the workers in their respective regions. The key problem with allocation solely based on the satisfaction of the professional is that the majority of the professionals would be more interested in working in the more developed areas, in hopes of getting better accommodations and benefits. As such, the density of worker allocation would be skewed in favor of the more developed regions, which will result in a vacuum of professionals in the underdeveloped regions. Consequently, our secondary objective was introduced: maximizing dispersion. We needed to maximize the dispersion of workers to ensure no vacuum occurs, while also trying to maximize the satisfaction of each individual. Thus our problem became a multi-objective problem.

1.2 Motivation

Worker allocation in different positions in an optimal manner is a complicated predicament for every profession. Non-optimal worker-allocation leads to dissatisfaction in workers, which lowers productivity and causes workplace dysfunctions and loss of time, effort, as well as monetary loss in working places. Additionally, it leads to the discontentment of the workers who are designated to work in the rural areas as they feel as though they are receiving fewer benefits compared to the workers in the more developed areas. These problems are more specifically observed in the context of essential workers such as doctors [1], who are often forced to work in rural and often remote areas. Owing to extreme dissatisfaction, a higher rate of turnover is seen at these places as most people designated to such areas show acute urgency to move away posthaste. Due to such a high rate of turnover, an unstable work environment is born. Such an environment is undesirable to both the workers as well as the employers. As such, we have been motivated to find an optimum worker allocation method, that takes into account various factors that influence the satisfaction of workers and tries to maximize those factors. The method will also have to ensure the maximization of worker dispersion, so that not all the workers will be concentrated in a few developed regions while the underdeveloped regions' positions remain vacant. As such, this is a multi-objective optimization problem.

The importance of designing such a model is to make sure that rural areas are receiving stable and quality service from essential workers such as doctors [1]. The solution to this problem is threefold, the first one is designing the neural network model to predict satisfaction, the second one is to apply a dispersion function that will make sure that allocations are made in under-developed areas as per demand, the third one is to allocate professionals in every area while simultaneously maximizing satisfaction and dispersion. This model is expected to take us one step further to stabilize the workers in remote areas.

Thus, our problem was formulated by defining the two subgoals in the form of two functions. For a visual representation of our objectives, we can take a look at the following maps in figures 1.1, 1.2 and 1.3.

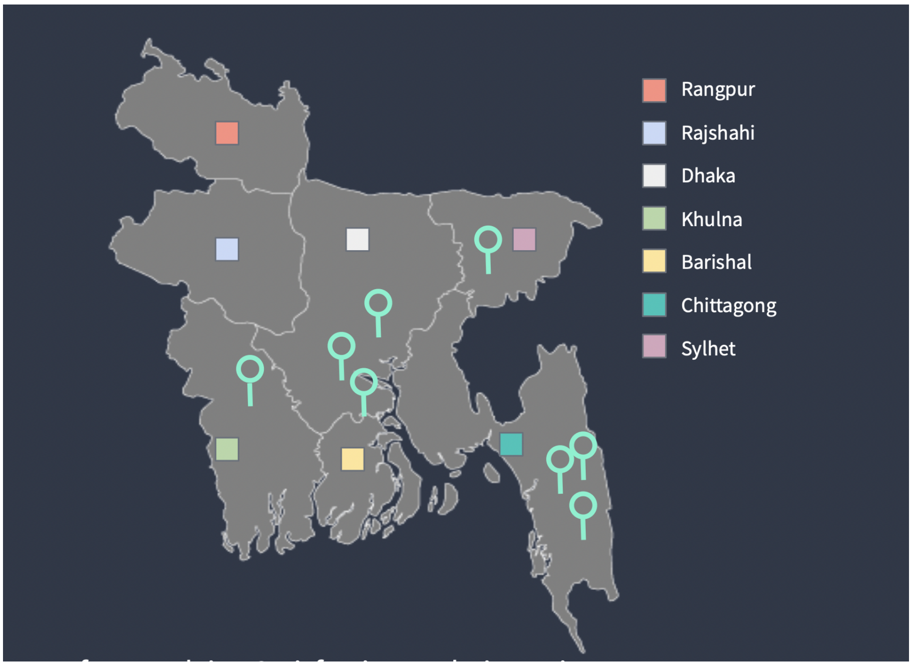


Figure 1.1: Allocation Before Applying the Model

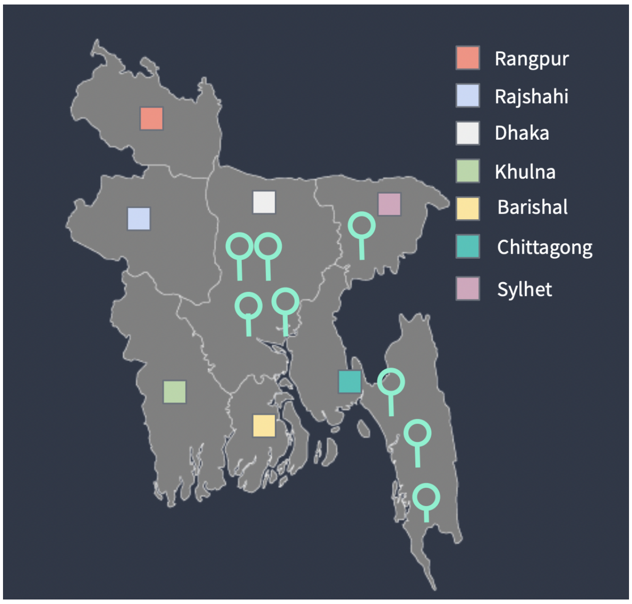


Figure 1.2: Allocation After Applying the Model

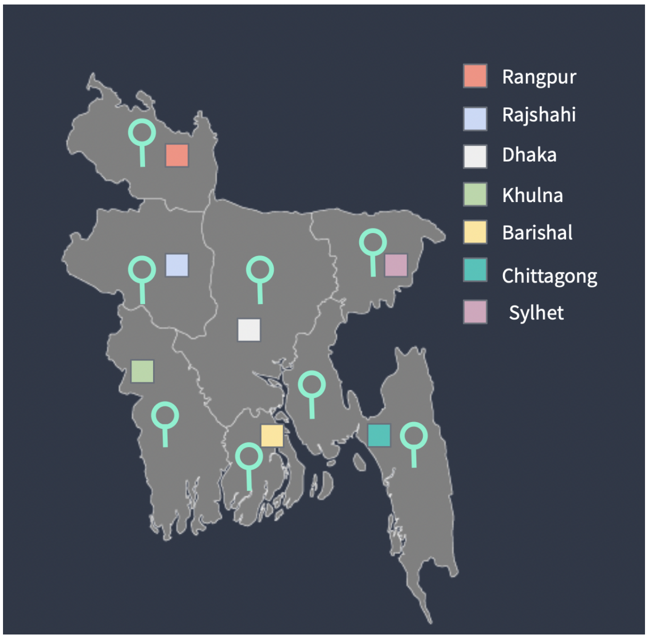


Figure 1.3: Allocation After Optimizing Dispersion (Out of Our Scope)

In figure 1.1, we can see that the placement of the workers (represented by the green figures) is slightly dispersed, yet a lot of regions remain empty. This is due to the lack of an appropriate dispersion method. Also, clustering can be seen in the more developed and desirable regions such as- Dhaka and Chittagong. In figure 1.2, we can see that due to the application of only the satisfaction function, all the workers are solely placed in the prime developed regions such as Dhaka and Chittagong [2]. So dispersion occurs to a minimal degree.

In figure 1.3, we can see that due to the application of the dispersion function on top of the satisfaction function, workers are now properly dispersed and no region remains completely worker-less.

Thus it can be seen how the two functions will have an influence upon the worker distribution from the above pictorial representations.

1.3 Research Challenges

To order to formulate our satisfaction function, we needed to take a machine learning-based approach, as taking a mathematical approach was difficult. The difficulty arose from the fact that we could not predict the weights for the function arbitrarily. As a machine learning approach was taken, data collection became a necessity, as preexisting data was unavailable. We collected data from real people by distributing a well-constructed questionnaire, which we will describe in detail in Chapter 4. We utilized the collected data to train a neural network model based on our problem in order to generate weights for the satisfaction function. After the satisfaction function was formulated with acceptable weights, we moved on to the dispersion function. A few challenges needing to be overcome to solve the mentioned problems. They are mentioned as follows:

- **Collecting dataset:** There is no available dataset that we can use to solve this problem, so we had to create a dataset on our own. This has been even more challenging during the COVID-19 pandemic, as ideally, along with online surveys, we would be going to offices and hospitals in person to conduct our surveys. But on account of the pandemic restrictions, we had to solely depend on online circulation of our survey.

- **Improving accuracy:** The satisfaction model is highly dependent on human psychology. There is no one way of predicting what will make a person content, as happiness is yet to be formulated through an equation, and there are more to play than the factors we have chosen to improve satisfaction. Each person has different thoughts, and it is especially challenging to quantify everyone's satisfaction with one model, that too created by such novice learners as us. So, the accuracy of the model is not too high.

- **Formulating Well Dispersed Solution:** We are aiming at creating stability in workplaces by assigning workers based on their satisfaction level in an area. However, relying alone on employee satisfaction may lead us to a result that is highly concentrated in well-developed cities, whereas underdeveloped or rural areas may remain sparsely allocated. Although there are people who prefer suburban lives to city lives, we cannot solely depend on the possibility of some people naturally being satisfied with being allocated outside of cities. So, in addition to our satisfaction model, we have also formulated a dispersion function that makes sure that the employees are well dispersed all around the country.

1.4 Summary of the Chapter

In this chapter, the problem we are trying to solve was introduced. Our motivation behind trying to solve this problem as well as the challenges we faced while trying to solve our problem were briefly discussed. Moving forward, we will further elaborate on the various aspects of the mentioned problem and the proposed solutions.

Chapter 2

State of the Art

2.1 Overview

In this chapter, we will shortly discuss some of the previous works related to the subject matter at hand. The previous works primarily focused on the study of statistics related to worker allocation and changes required in government policies. However, our focus will be on individual satisfaction of people, and how to disperse them optimally based on that. A list of the state of the art literature we have found which are related to our work is provided as follows:

- Improving health workforce recruitment and retention in rural and remote regions of Nigeria: Niyi Awofeso [3].
- Suggestions to ameliorate the inequity in urban/rural allocation of healthcare resources in China: Yiyi Chen, Zhou Yin & Qiong Xie [4].
- Urbanization and physician maldistribution: a longitudinal study in Japan: Shinichi Tanihara, Yasuki Kobayashi, Hiroshi Une & Ichiro Kawachi [5].
- A MATHEMATICAL MODEL TO MEASURE CUSTOMER SATISFACTION: Alexander C. Pereira [6].
- Modeling Employee Satisfaction in Relation to CSR Practices and Attraction and Retention of Top Talent: Simona Vinerean, Iuliana Cetina, Luigi Dumitrescu [7].

2.2 Descriptions of Papers

The state of the art allocation problems we have found which focused on worker distribution are elaborated on as follows:

- In paper [3], the author focused on the factors that hinder recruitment and retention of the healthcare workforce mostly in the rural areas and suggested approaches that would improve the current unsatisfactory condition of the workforce situation in one of Africa's most populated nations, Nigeria. The factors that were pointed out as hindrances towards recruitment and retaining in rural areas included issues such as lack of proper infrastructure, spartan living standards, inadequate number of properly trained staff leading to burnout, inadequate remuneration and sub-optimal distribution of healthcare workers. The suggestions for improvement included proposals such as improved incentives, improved governance, developing better strategies for recruitment and retaining and improving training methods.
- In paper [4], the authors focused on the inequity in the distribution of healthcare resources between urban and rural areas of China. The report pointed out the ethical flaws in various Government policies that have led to the inequality in resource distribution. The authors finally proposed countermeasures that would lead to the optimization of the allocation of resources, such as formulation of policies that would stimulate the flow of resources to rural areas, strengthen the responsibilities of both governmental and public financial investments, improve the utilization of resources, and strengthening the responsibilities of both governmental and public financial investments.
- In paper [5], the authors focused on the maldistribution of physicians in urban and rural areas of Japan with respect to the changes in the population growth rate in those regions. The paper further examined trends in the geographic disparities in population and physician distribution. It mostly focused on how the changing population growth rate was impacting the physician-to-doctor ratio in urban vs rural areas.
- In paper [6], to measure customer satisfaction, a Satisfaction Function (SF) has been proposed which is based on the customers' attitude regarding their attitude regarding any of the products the company is offering. The function is defined by a ratio of the customers' demands and the demands that have been met.
- In paper [7], one of the topics discussed was emphasizing more on employee satisfaction as it is vital for attracting and retaining skilled employees to companies. They have proposed to make the work environment suitable enough so that the employees can have more decision making power which will give them a sense of fulfilment,

to increase accountability, to treat jobs as products: meaning, to bring in shape the jobs such that the employees will value those more and increase sustainable practices. Their study is based on surveys from 10 multinational companies.

2.3 Research Opportunities/Gap

The term “Research Opportunities” or “Research Gap” refers to the issues that have not been addressed in currently existing works or the further work that can be done on the subject matter in question. It refers to the scope of work that can still be done on the given subject matter. For our given subject matter of distribution of workers, we have only found 3 existing works, and in addition to that, 2 papers based on employee satisfaction were found. The existing works have quite a few research gaps which we aim to solve.

Our thesis is inspired by the work of Awofeso [3] on the distribution inequity problem of healthcare workers among the rural and urban areas of Nigeria. The gaps we noticed in this literature and explored are as follows:

- “Worker Satisfaction” has not been taken into account in the paper by Awofeso, whereas ensuring worker satisfaction is one of the primary goals of our thesis.
- The work by Awofeso proposes that the government should change their policy in order to solve this problem, which creates a lot of vagueness and is an intangible goal. We, on the other hand, are posing this problem as an optimization problem and have formulated a mathematical model to solve it.
- There was no available dataset to work on when we started our study. Thus we have curated a real-world dataset to work on, which can be used by any other group who would like to work further on this topic.

In addition to this, the gaps that were present in the other works are as follows:

- The literature by Yiyi Chen et al. [4] suggested the revision of policies at the government level, strengthening the responsibilities of both governmental and public financial investments and optimal allocation of resources. However, no concrete suggestions were made on how to go about performing the allocations optimally.
- The literature by Shinichi Tanihara et al. [5] focused on how the changing population growth rate was impacting the physician-to-doctor ratio in urban vs rural areas. It was primarily a study of the statistical differences between the changing eras. We are analysing how to make tangible changes, not just theoretical studies.

- The literature by Alexander C. Pereira [6] does not address employee satisfaction, rather they are more concerned with customer satisfaction. Inversely, we are working the other way around.
- The Satisfaction Function (SF) proposed by [6] is, at a higher level, a ratio between demands and the number of met demands. Whereas, we are planning on taking different features of a person and their work environment and training a neural network to predict employee satisfaction.
- The literature by Simona Vinerean et al. [7] does not propose any mathematical or statistical model to increase employee satisfaction. They have taken a different approach, based on their surveys, which is to change the office environment and culture. Conversely, by using our model, allocation to areas or offices will be based on the features of the designated areas and what can be more desirable to the employees.

2.4 Summary of the Chapter

In this chapter, we discussed the existing works by various authors and what their works have failed to discuss which we are attempting to solve in our own work. After studying these works we noticed that most of the works focused more on the bureaucratic and government aspects rather than what can be done practically, at a lower level. Thus, moving forward, we will be focusing on this area more.

Chapter 3

Background Study

3.1 Overview

In this chapter, we discuss some concepts which are necessary before working on the Worker's Allocation optimization problem. As our aim is to treat our problem as a Multi-Objective Optimization problem and to solve it using a variant of the Genetic Algorithm known as NSGA-II, it is a prerequisite to provide an explanation for these terms.

3.2 Multi-Objective Optimization

In section 3.2, we are going to conduct a discussion about Multi-Objective Optimization (MOO). We are going to explore the concept of Multi-Objective Optimization, as well as its necessity in solving our problem. Finally, we will discuss why we chose Genetic Algorithm amongst the various choices that were available to us for solving Multi-Objective Optimization problems.

3.2.1 Concept of Multi-Objective Optimization

Multi-objective optimization, which is also known as multi-criteria optimization, multi-attribute optimization or Pareto optimization, is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously and is an area of multiple criteria decision making [8]. The solution comes in the form of a set of solutions that define the optimal trade-off between contradictory objectives.

In mathematical terms, a multi-objective optimization problem can be formulated as [8]:

$$min(f_1(\vec{x}),\ f_2(\vec{x}), \dots,\ f_k(\vec{x}));\ s.t.:\ \vec{x} \in X \tag{3.1}$$

where the integer $k >= 2$ is the number of objectives and the set X is the feasible set of decision vectors. Some constraint functions typically define the feasible set. In multi-objective optimization, all objectives are not simultaneously minimized by a single feasible solution. In order to decide upon an ideal solution, Pareto optimal solutions are considered. Pareto optimal solutions are such solutions wherein the objectives are fulfilled to such an extent that the solution cannot be improved upon by enhancing one of the objectives without degrading some other objective simultaneously. Pareto optimal outcomes together form sets called **Pareto front** [8].

In multi-objective optimization problems, the integrity of a solution is determined by dominance. If there exist two solutions x_1 and x_2, x_1 is said to dominate x_2, if two conditions are fulfilled [9]:

- Solution x_1 is no worse than x_2 conversely, x_2 is not better than x_1 in all objectives.
- Solution x_1 is exclusively better than x_2 in at least one objective.

3.2.2 Necessity of Multi-Objective Optimization

For multi-objective optimization problems, there usually exist multiple Pareto optimal solutions. Thus solving such a problem is not as straightforward as it is for a conventional single-objective optimization problem. There are multiple methods for solving such problems, one of which is to convert the multiple objectives into a single objective and then treating it as a single objective problem. This method is known as **scalarization** [8].

Our problem focuses on two objectives:

- Maximizing Satisfaction of workers
- Maximizing Dispersion of workers

Theoretically, maximizing the satisfaction of workers would ensure the workers would all be more interested in staying in more urban regions, thereby leaving positions in comparatively rural areas to be vacant. As our two objectives are inversely related to each other, multi-objective optimization is a necessity for solving this problem. It cannot simply be solved as a single objective problem.

Subsequently, the MOO problems can again be solved in two different ways:

- **A priori methods**: These methods require that sufficient preference information is expressed before the solution process [10].
- **A posteriori methods**: These methods are designed to produce all the Pareto optimal solutions or a subset of the Pareto optimal solutions which represents the entire solution [8]. A posteriori methods are divided into the following two classes [8]:
  - **Mathematical programming-based a posteriori methods:** In this method, an algorithm is repeated and one Pareto optimal solution is produced in each run of the algorithm. Examples include- utility function method, lexicographic method, and goal programming [8].
  - **Evolutionary algorithms:** In this method, a set of Pareto optimal solutions is produced in one run of the algorithm. Examples include- NSGA-II, Successive Pareto Optimization (SPO), Multi-objective particle swarm optimization etc. [8].

As our goal is to obtain a set of solutions rather than a single solution, moving forward we will be discussing Genetic Algorithms, which belong to the a posteriori class of Multi-Objective Optimization solutions, as a method for solving our problem.

3.3 Genetic Algorithm

In section 3.3, we are going to conduct a discussion about Genetic Algortihm (GA). We are going to discuss the basic concept of Genetic Algorithm, its phases, its advantages as well as our motivation behind choosing this algorithm. Finally, we will discuss the limitations of Genetic Algorithm.

3.3.1 Concept of Genetic Algorithm

Genetics is a branch of biology concerned with the study of genes, genetic variation, and heredity in organisms [11]. In computer science and operations research, a genetic algorithm (GA) is a meta-heuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). The common usages of Genetic algorithms are to generate solutions to optimization and search problems of superior quality by implementing concepts inspired by evolution and biological sciences, such as mutation, crossover and selection [12].
The basic concept of genetic algorithm is derived from concepts of Genetics and Natural Selection. Natural selection is such a process that starts initially with a group of objects or individuals called the “population”. The fittest members of the population produce offspring

that inherit the characteristics of the parents. This selection process repeats again for the generation of the offspring as well, each time selecting the fittest individuals, so that the ultimate generation becomes comprised solely of the fittest individuals and unfit individuals are completely or nearly eliminated [13].

The genetic algorithm consists of 5 phases. They are as follows:

- **Initial population:** The set of individuals with which the process starts is called the **initial population.** In optimization or search problems, each individual represents a solution to the problem. The set of characteristics or parameters that identify an individual is called its **Genes**. When genes are connected together in a string form, it is known as a **Chromosome**. The concept can be seen in figure 3.1.

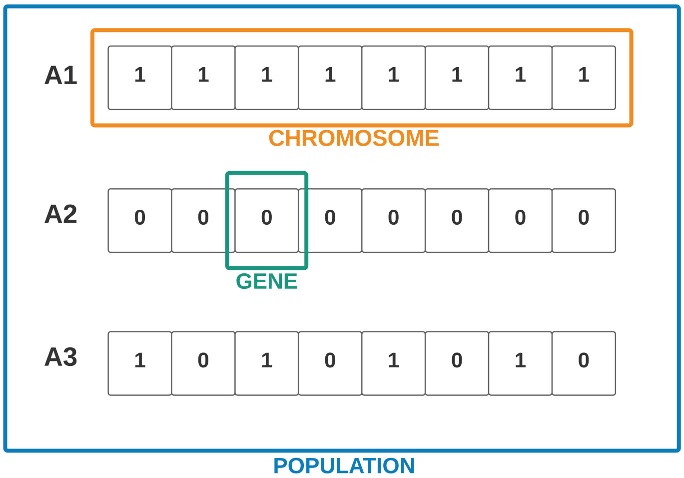


Figure 3.1: Population, Chromosomes and Genes

- **Fitness function:** The function that determines the ability of an individual to compete with the other individuals of the current generation, is known as the **fitness function**. The function gives a fitness score to each of the individuals in each iteration; depending on this score the probability of an individual being chosen for reproduction is determined.

- **Selection:** In this phase, the fittest individuals are selected in order to pass on their genes to the next generation. In this phase, the solutions that are the most ideal within the current generation are selected to be propagated.

- **Crossover:** In this phase, genetic variety gets introduced to the solutions. A **crossover point** is chosen at a random point in the genes for each pair of parents from whom the offspring is meant to be generated. Subsequently, genes are exchanged within the parents until the crossover point is reached. Thus offspring are created with characteristics of both parents. Finally, the new offspring also get added to the population. This phenomenon can be observed in figure 3.2, 3.3, and 3.4.

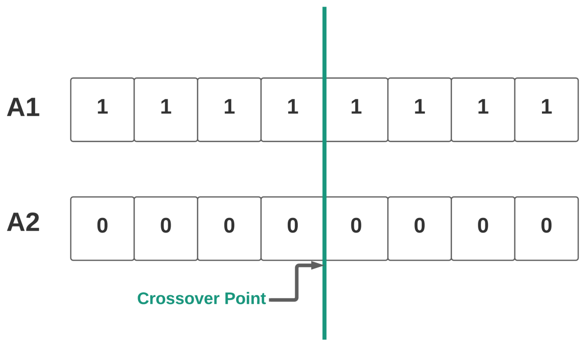


Figure 3.2: Crossover point

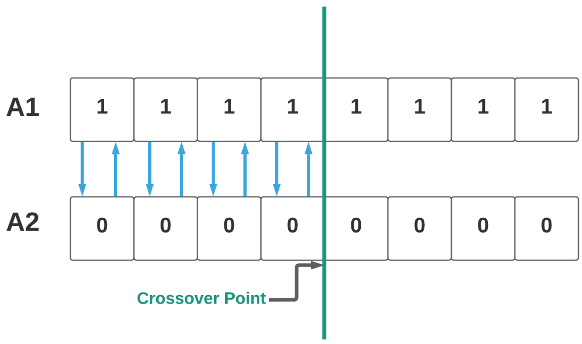


Figure 3.3: Exchanging genes among parents

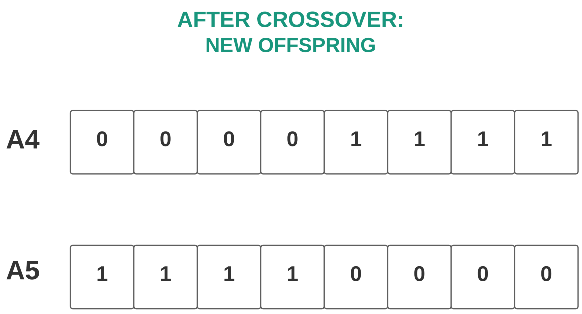


Figure 3.4: New offspring

- **Mutation:** In real life genetic science, the reproductions do not always strictly match with half of each of the parent's genes. Occasionally, genes get mutated and changed. The probability of mutation is usually determined by various factors and is not static. The purpose of mutation is the introduction of diversity within the population as well as prevention of premature convergence of solutions. This phenomenon can be observed in figure 3.5.

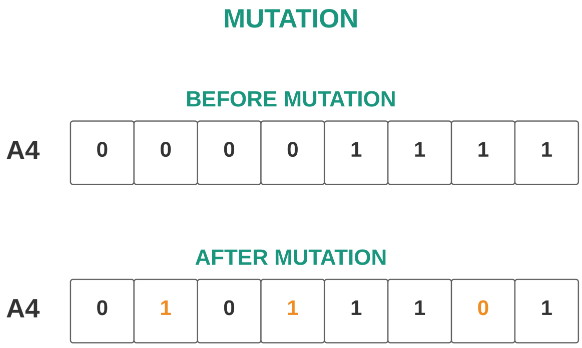


Figure 3.5: Mutation: Before and After

Genetic algorithm iterates until it fulfils some **terminating conditions.** Some of the conditions [14] are mentioned as follows:

- An ideal solution is found that fulfils all the required criteria.
- Minimum criteria for being classified as a potential solution is fulfilled.
- The number of permissible generations is specified initially and that number is reached.
- The solutions reached a plateau, i.e. they are no longer producing better results.
- The highest possible value for fitness function is reached.
- Allocated computational budget (time or memory) is reached or exceeded.

3.3.2 Necessity of Genetic Algorithm

Genetic algorithms belong to a subset of a much larger branch of computation known as Evolutionary Computation [15]. Genetic Algorithm has some advantages, which prompted us to choose this algorithm.

3.3.2.1 Advantages of Genetic Algorithm

Genetic Algorithm has some advantages [15], which are as follows:

- This algorithm can optimize single as well as multi-objective problems.
- This algorithm has parallel processing capabilities.

- This algorithm optimizes both continuous and discrete functions.
- This algorithm always provides a solution to the problem at hand. The solution might not be perfectly optimized, but it gets better over time.
- This algorithm is useful for a large search space and each individual has a large number of parameters.
- This algorithm is faster and more efficient when compared to the traditional methods.
- Instead of providing a single solution, this algorithm provides a list of solutions.
- Derivative information is not required for this algorithm, which is important for real-world problems as many of them do not have this information. As our problem is a real-world problem, this was an important consideration behind choosing this algorithm.

3.3.2.2 Motivation behind choosing Genetic Algorithm

Our motivation behind choosing Genetic algorithm can be itemized as follows [15]:

- Genetic algorithms are a useful tool for solving NP-hard problems, i.e. problems that, when solved in greedy methods, can take huge amounts of time even using powerful computation powers. The algorithm can provide an optimal or near-optimal solution in a feasible amount of time, even in such difficult cases.
- Genetic algorithms work better than traditional methods based on calculus, as calculus-based methods usually begin with a random point and work towards a local optimum by moving along the gradient. This approach is efficient for problems that have a singular solution, i.e. only one local optimum which is the ultimate global optimum. However, for complex problems with multiple parameters and objectives, there might be multiple local optima. In such cases, traditional methods might fail. As genetic algorithms process multiple solutions parallelly, it works well for multi-objective problems.

3.3.3 Limitations of Genetic Algorithm

The advantages of genetic algorithms played a crucial role in our decision to choose this algorithm in order to optimize our problem. However, we also had to keep in mind the limitations of the algorithm in order to determine the efficacy of our solutions. Some of the limitations are as follows [15]:

- Repetitive calculation of fitness value using fitness function can be computationally expensive over time.
- As the algorithm is stochastic, optimal solution is not guaranteed.
- The algorithm may get stuck on a plateau and never reach an optimal solution despite repeated iterations.

3.4 Multi-Objective Optimization Algorithm

In section 3.4, we are going to discuss Multi-Objective Optimization Algorithms (MOOA). We will discuss some criticisms for the older algorithms as well as some terms necessary for understanding the concepts of Multi-Objective Optimization Algorithms, which we will see further in the subsequent sections 3.5 and 3.6.

3.4.1 Necessity of Multi Objective Optimization Algorithm

When a problem consists of multiple objectives, the resultant solutions are constituted of a set of optimal solutions (known as Pareto-optimal solutions) instead of a single optimal solution. Without obtaining further information, one of the Pareto-optimal solutions cannot be said to be better than the other. According to classical optimization methods (including the multi-criterion decision-making methods), the multi-objective optimization problem must be converted into a single-objective optimization problem by emphasizing one particular Pareto-optimal solution at a time. Following this procedure is computationally expensive and time-consuming as it requires the application of the method repeatedly [16]. The purpose of the introduction of various adaptations of the Genetic Algorithm was to overcome this issue.
Non-dominated Sorting Genetic Algorithm (NSGA) is one of the first variants of the basic Genetic Algorithm that was introduced [17] in 1995. However, there have been several criticisms of this variant over the years. Some of the **criticisms** [16] are as follows:

1. **High computational complexity of non-dominated sorting**: NSGA algorithm has a computation complexity of $O(MN^3)$ (where M is the number of objectives and N is the population size). So NSGA becomes extremely expensive to compute for large population sizes. Non-dominated sorting procedure in every generation gives rise to this complexity in computation.
2. **Lack of elitism:** The general process of allowing the best organism(s) from the current generation to the next generation without any mutation or alteration is known

as elitist selection and guarantees that the quality of solution obtained by the Genetic Algorithm will not decrease from one generation to the next. [18] suggests that elitism can speed up the performance of the Genetic Algorithms significantly, which also helps in the prevention of the loss of good solutions due to mutation. However, NSGA does not support elitism, so each solution has a small probability of being mutated even if the solution is highly effective.

3. σ_{share} is a sharing parameter that is used to ensure the diversity in populations through sharing. This parameter is needed to ensure a wide variety of equivalent solutions can be obtained. However, it is difficult to specify the value of this parameter for optimal solutions, as it cannot be the same for all problems. So a parameter-less diversity-preservation mechanism is desirable. NSGA does not provide this parameter-less mechanism, and it relies on σ_{share}, which is not desirable in optimal methods.

In order to overcome the above-mentioned issues, NSGA-II algorithm was proposed by Kalyanmoy Deb et. al. [16]. NSGA-II is an improved version of the NSGA algorithm, and this is the one we will be using for solving our problem as well.

3.4.2 Necessary Terms of Multi Objective Optimization Algorithm

3.4.2.1 Crowding Distance:

The average distance of its two neighbouring solutions is known as the crowding distance value of a particular solution [19]. An infinite crowding distance value is given to the boundary solutions which have the lowest and highest objective function values so that they are always selected [19].

3.4.2.2 Pareto Front:

Pareto efficiency is a concept of efficiency in exchange where an individual or preference criterion cannot be made better off without doing so at the expense of another or multiple other individuals or preference criteria. Thus the other individual or individuals would be worse off in this exchange. When an initial situation is given, pareto improvement is an updated situation where some individuals will gain, and no individuals will lose their advantage. If a situation has a possibility of pareto improvement, it is called pareto dominated. Pareto front or pareto set is the set of all non-pareto-dominated solutions in a multi-objective optimisation function, for a given search space [20].

3.5 NSGA-II Algorithm

In section 3.5, we will discuss NSGA-II algorithm. We will discuss its basic concept, its features, its phases as well as its limitations. It will help in the understanding of why we chose this algorithm in order to solve our problem.

3.5.1 Concept of NSGA-II Algorithm

NSGA-II is a variant of Genetic Algorithm which is widely used in many real-world applications. NSGA-II is considered to be a solid benchmark to test against, albeit a slightly outdated method. A specific type of crossover and mutation is utilised by NSGA-II to generate offspring and subsequently, the next generation is selected according to nondominated-sorting and crowding distance comparison [21]. The concept is shown in figure 3.6 [16]. This figure was adapted from the work of Kalyanmoy Deb et. al. [16].

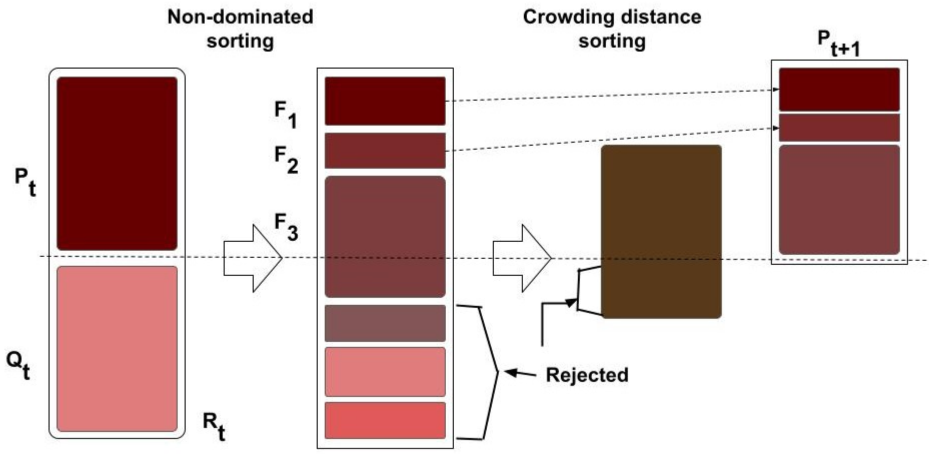


Figure 3.6: NSGA-II procedure

NSGA-II is an evolutionary algorithm. The purpose of the development of evolutionary algorithms was because of the problems of the classical direct and gradient-based techniques when leading with non-linearities and complex interactions. The problems [22] are as follows:

- The chosen initial solution determines the convergence to an optimal solution.
- The probability of getting stuck on a sub-optimal solution is quite high

3.5.2 Features of NSGA-II Algorithm

NSGA-II was developed in order to introduce the following three features [22], which were missing in the NSGA variant:

1. It utilises the elitist selection principle, i.e. the individuals with higher fitness scores (the elites) of a population are given preference so that there is a higher chance that they will be carried to the next generation.
2. It utilises an explicit diversity preserving mechanism (Crowding distance) rather than having to rely on sharing parameters such as σ_{share} which was used in NSGA.
3. It puts emphasis on the non-dominated solutions.

3.5.3 Phases of NSGA-II Algorithm

The NSGA-II consists of the following 4 phases [23]:

1. **Phase 1 - Population initialization:** The population is initialized based on the problem range and constraints.
2. **Phase 2 - Initial Evaluation and Ranking:** After initialisation, objective function evaluation takes place. After this process is accomplished, sorting takes place based on the non-domination criteria of the population.
3. **Phase 3 - Genetic Operations:** In this phase, the basic Genetic operations such as-selection, crossover and mutation take place. A binary tournament selection might be conducted to execute the selection of individuals. A crowded-comparison operator is used for this purpose. Simulated binary crossover and polynomial mutation might be utilised as genetic operators.
4. **Phase 4 - Recombination and Selection:** Recombination occurs by combining the population of the offspring and the current generation i.e. the parent generation. After non-dominated sorting is accomplished, the crowding distance value is assigned according to Pareto-front. Depending on rank and crowding distance, N individuals in the population are selected. Selection takes place to set the individuals of the next generation. Each front fills the new generation subsequently until the current population size is exceeded by the new generation's population size. When the stopping criteria is met, simulation stops. Elitism is maintained in this phase, which is one of the defining characteristics of the NSGA-II algorithm.

A flowchart showing the phases of NSGA-II is given in figure 3.7.

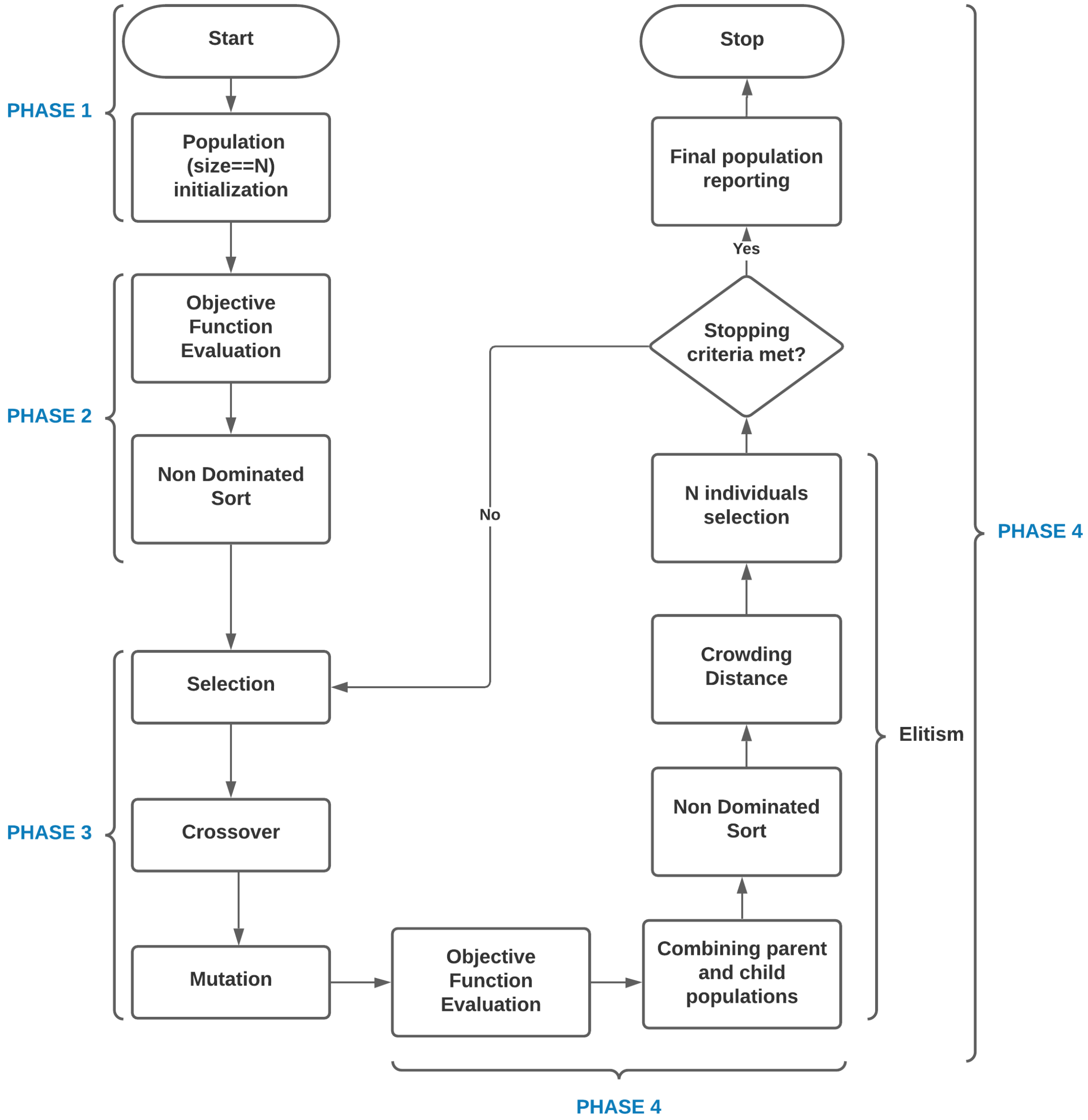


Figure 3.7: NSGA-II phases

3.5.4 Limitations of NSGA-II Algorithm

The performance of NSGA-II algorithm in noisy environments was found to be lacking. In a head-to-head comparison between NSGA-II AND SPEA-II, the former was inferior to the latter in the early generations, while the opposite was true for the later generations [24]. For this reason, we could not solely rely on NSGA-II for solving our problem. This is why we also used SPEA-II in order to determine which one gave us the best result, and ultimately use the algorithm which gave us the best performance.

3.6 SPEA-II Algorithm

In section 3.6, we will discuss SPEA-II algorithm. We will discuss its basic concept and the steps of the algorithm. Finally, we will conduct a comparison between the various features of the two algorithms: NSGA-II and SPEA-II. It will help in the understanding of why we chose these two algorithms in order to solve our problem.

3.6.1 Concept of SPEA-II Algorithm

Modified Strength Pareto Evolutionary Algorithm (SPEA-II) is similar to the NSGA-II algorithm, in that it is one of the most important multi-objective evolutionary algorithms that use elitism approach [25]. The SPEA-II is such an Evolutionary Algorithm which is multi-objective, Pareto-based that drives solutions towards the Pareto optimal front by employing a dominance fitness measure which is based on count and rank. An elitist strategy is employed by this algorithm such that an external archive of solutions is maintained and new solutions are exclusively produced from the members of the archive. This algorithm also differentiates between solutions and maintains spread of solution (also known as 'diversity') within the archive by utilizing a nearest neighbor density estimation technique [26].

Count and strength dominance measures are utilised by SPEA-II to evaluate the fitness of solutions. A **strength (S) score** reflects the count, which indicates the number of solutions dominated by a particular solution. The total number of solutions which dominate a particular solution and the sum of their strength scores is known as **Rank**. In order to fine tune fitness scores and truncate the excess non-dominated solutions, a nearest neighbor density estimate is utilised by this algorithm [26].

3.6.2 Steps of SPEA-II Algorithm

The steps of SPEA-II algorithm can be shown as follows [27]:

1. Generation of initial population P_0 and empty archive (external set) A_0. $t = 0$ is set.
2. Calculation of fitness values of individuals in P_t and A_t.
3. Nondominated individuals in P_t and $A_t = A_{t+1}$. If size of $A_{t+1} > N$ then A_{t+1} is reduced, else if size of $A_{t+1} < N$ then A_{t+1} is filled with dominated individuals in P_t and A_t.
4. If $t > T$ then the non-dominated set of A_{t+1} is output. Procedure is stopped.
5. Mating pool is filled by binary tournament selection with replacement on A_{t+1}.

6. Recombination and mutation operators are applied to the mating pool and P_{t+1} is set to the resulting population. $t = t + 1$ is set and go to Step 2.

A flowchart showing the steps of SPEA-II Algorithm is given in figure 3.8.

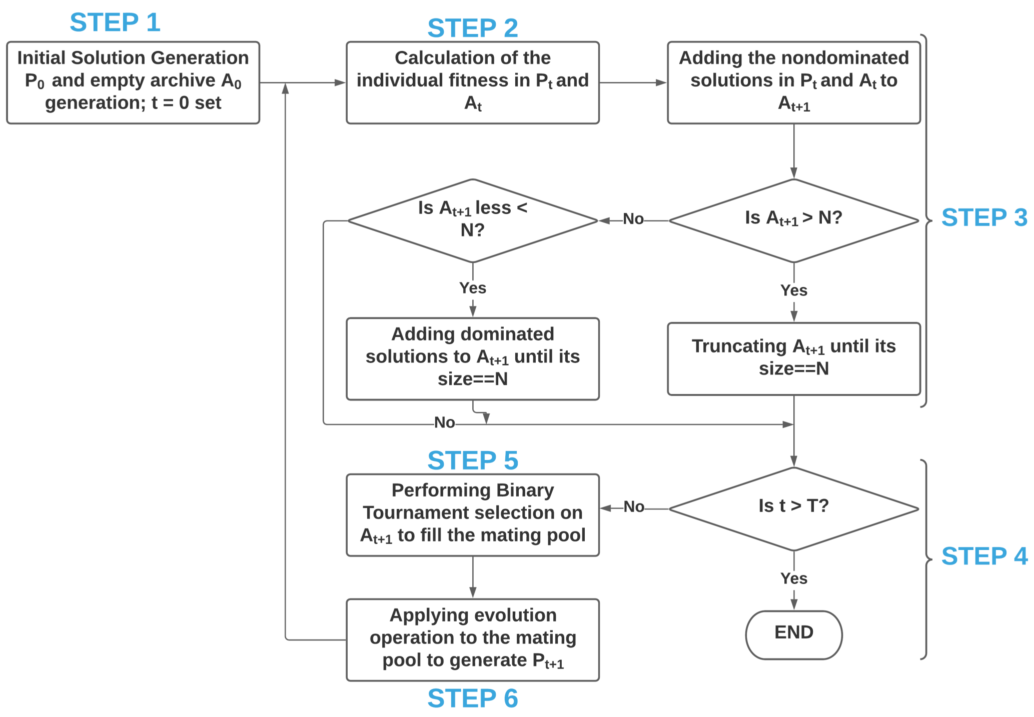


Figure 3.8: SPEA-II procedure

3.6.3 Comparison between SPEA-II and NSGA-II Algorithm

Upon performing a comparison between SPEA-II and NSGA-II [28], it was found that they shared some similarities as well as some dissimilarities.
Some of the similarities between NSGA and SPEA as shown in the work of D. Kunkle [28] are as follows:

- SPEA stores the Pareto-optimal solutions it finds through archiving i.e. externally, which is what NSGA does as well
- SPEA as well as NSGA uses Pareto dominance in order to assign fitness values to individuals
- SPEA and NSGA both perform clustering in order to reduce nondominated solutions without causing the destruction of characteristics of Pareto-optimal fronts

As NSGA-II and SPEA-II are modified forms of NSGA and SPEA respectively, so the similarities hold true for these algorithms as well.

However, SPEA also had some unique aspects of its own, which are mentioned as follows [28]:

- A method of niching is used, which does not require any fitness sharing parameters.
- Domination of members of the population is irrelevant, as fitness is determined for an individual from the archive of non-dominated solutions.
- In the selection step, all the solutions in the archive participate.

These unique aspects of SPEA are also present in SPEA-II. Other than these, a few other performance comparisons can be made for SPEA-II and NSGA-II, which are as follows [28]:

- SPEA-II seems to outperform NSGA-II in high-dimensional objective spaces
- SPEA-II had less clustering than NSGA-II, i.e. "better distribution".
- NSGA-II found solutions which were closer to the outlying edges of the Pareto-optimal front, i.e. it had a "broader range" of solutions.

From these comparisons we can observe that while NSGA-II is outperformed by SPEA-II in some scenarios, the opposite can also be true in some cases. So we cannot conclusively say which algorithm works better without obtaining concrete evidence through experimentation with our own data. This is why we could not discount either of the algorithms for solving our tasks, which is why we used both of the algorithms in order to reach our own conclusion.

3.7 Neural Network

Neural Network is analogous to the human brain [29]. Human brain consists of neurons which are interconnected by synapses. The process of thinking can be mimicked by a computer using neural networks. Neural Network is formed by input and output layers. These two layers are connected by several other layers of neurons and weighted synapses. "Learning" in the context of neural network means to update the weights of these synapses, until the network can predict the correct outputs.

3.7.1 Necessity of Neural Network

To maximize our first objective, which is to maximize the satisfaction of employees in their designated areas, first we have built a model that is capable of predicting employee satisfactions in such areas. The first step towards doing that is to select the features that most likely impact employee satisfaction. More about feature selection has been described in the following subsection, but first we will explain why we have chosen Neural Network to predict employee satisfaction.

The answer to this question is a two-fold one. The first being why do we need machine learning in the first place, and the second being why did we choose neural network to solve this problem.

To answer the first question is that: our satisfaction model needs to be capable of predicting satisfaction of employees in situations that has not been presented to the model before. This is because, in real life applications, we may have to assign employees to new cities where a company probably did not have any branches before, and all we have are some of the characteristics of the city, which we can use as the features of our model. This is where machine learning comes into play. We trained our machine learning model with our own gold-standard dataset that we have curated, so that it can predict employee satisfaction in a completely new area.

Now, to answer the second question is that: our satisfaction cannot be represented as a linear combination of its features, because satisfaction itself being a complicated psychological issue that can largely vary from person to person, even within the same demographic situation. We could have used linear regression instead of neural networks to find the coefficients of the features had it been possible to represent our objective as a linear combination. However, the problem being a real life problem, it is a non-linear problem, and there may be further correlation among the input parameters that we cannot predetermine, but which can be identified by neural networks. For this reason, we have used neural network to create our satisfaction model.

Neural networks can be used to solve both classification and regression problems.

3.7.2 Classification using Neural Network

Classification problems can be modeled by neural networks. If a problem has n classes, then the output layer of the neural network will have n number of neurons. Softmax function [30] is applied at the output layer to predict which class does the sample belong to. Here, softmax function can be defined as,

$$\sigma(z)_i = \frac{exp(z_i)}{\sum_{j=1}^{K} exp(z_j)} \tag{3.2}$$

Here,
σ = softmax function [30]
z = input vector
$exp(z_i)$ = standard exponential function for input vector [31]
K = number of classes in the multi-class classifier
$exp(z_j)$ = standard exponential function for output vector [31]

A diagram demonstrating classification using neural network is given in figure 3.9.

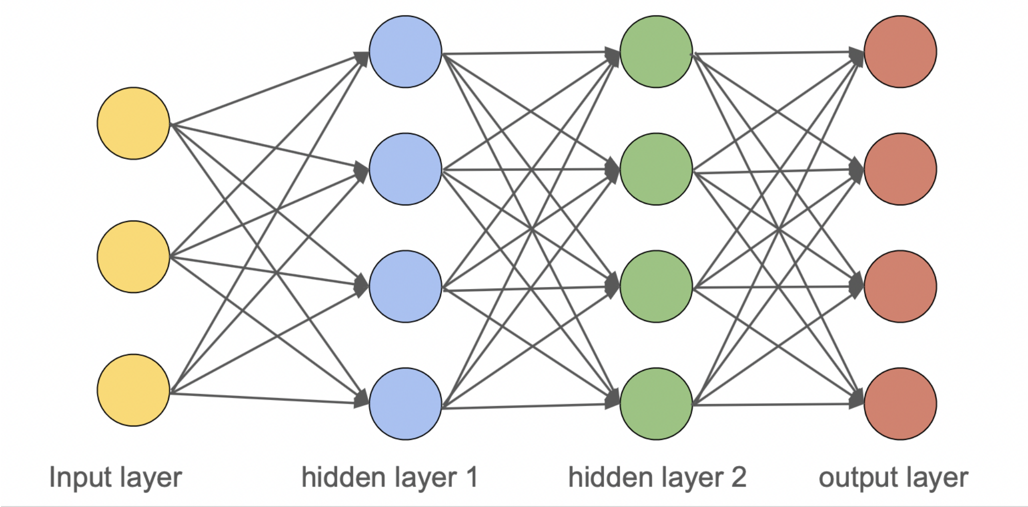


Figure 3.9: Classification using Neural Network

3.7.3 Regression using Neural Network

Neural Networks can also be used to model regression problems. Since the output of such a problem is one continuous value, so there is only one neuron at the output layer. Unlike classification problems, here, softmax is not used in the output layer.

A diagram demonstrating regression using neural network is given in figure 3.10.

3.7.4 Necessary Terms for Neural Network

In this section, we will define some of the necessary terms of neural network.

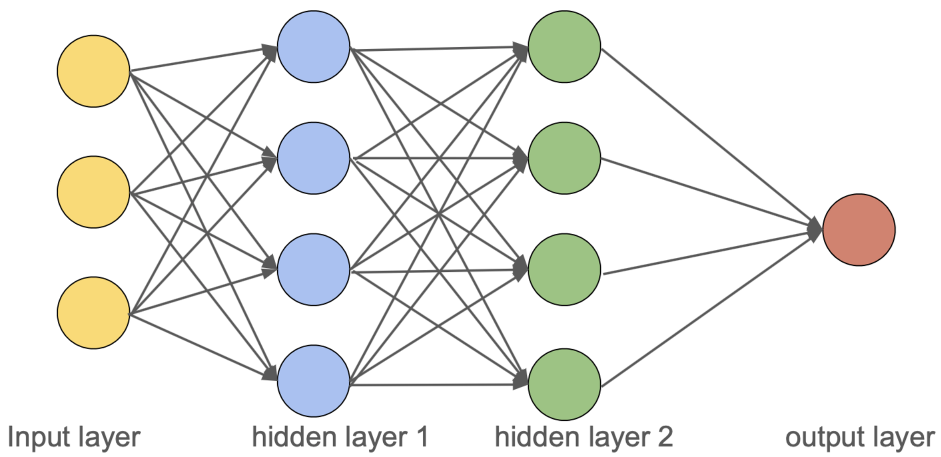


Figure 3.10: Regression using Neural Network

3.7.4.1 Minibatch

Minibatch, or "batch" refers to equally sized subsets of the dataset over which the neural network computes gradient and updates weight.

3.7.4.2 Iterations

Iteration is the number of times a batch of data has passed through the algorithm. Simply put, it is the number of passes the algorithm has done on the dataset.

3.7.4.3 Epoch

Epoch is the number of times an algorithm sees the whole dataset. One forward and one backward pass of all the training examples are known as one epoch.

3.7.4.4 Forward Propagation

Forward pass is a uni-directional process where the data is fed to the neurons in the forward direction. Input layer passes its data to the first layer of neural network. The neurons present in this layer processes the data and updates its weights. Finally, the data is passed on to the next layer of neurons for calculation and weight updating [32].

3.7.4.5 Backward Propagation

Backward propagation is the means of adjusting the values of weights and biases in such a manner that the cost function is minimized. The adjustment is done by calculating the gradient of cost function with respect to the weights and biases [33].

3.7.4.6 Activation Function

Activation function calculates the weighted sum of the inputs and adds bias to it. It then decides whether or not a neuron should be fired. Activation functions can be linear or non-linear [34].

3.7.4.7 Optimizer

Algorithms that work towards optimizing different attributes of neural networks, such as weights, are called optimizers [35]. The goal of these algorithms is to optimize weights in such a way that the cost function is minimized. Some of the optimizers are Adam [36], Adargrad [37], SGD [38]

3.8 Summary of the Chapter

In this chapter, we discussed the background knowledge necessary to understand why we used the concepts, algorithms and frameworks that we did. It helped us to gain a basic understanding of Multi-Objective Optimization, Genetic Algorithms, Multi-Objective Evolutionary Algorithms (such as NSGA-II and SPEA-II) and finally Neural Network. Moving forward, it will help one to understand the fundamental details of the work we did to reach our objective.

Chapter 4

Methodologies

4.1 Overview

As previously stated in chapter 1, our goal is to solve two problems: first, allocating workers by maximizing their satisfaction and second, trying to maximize the dispersion of workers throughout a region or country. Due to the twofold nature of our problem, where each objective is inversely related to the other, we needed multi-objective optimization, which we have also explained in chapter 2, and we also needed Neural Networks. We have used Neural Network to create our satisfaction prediction model, which can be considered as the first part of our thesis. The second and the most important part of our thesis, which is to find the optimal allocation, is done by Multi Objective Optimization. The details of our methodology have been presented in this chapter.

4.2 Problem Formulation

Simply put, the problem we have solved is that, if we have p number of people whom we want to allocate to n number of areas, then we have to find an optimal solution where all p people are satisfied, while all n cities' demand of employees have been fulfilled. To solve this, we have to bring the following two factors under consideration to solve our problem:

- **Cities or areas where we want to allocate our employees:** We must take into account the number of vacancy each of these cities have for a given profession.
- **The people we want to allocate to different cities:** We must take into account the individual satisfaction of each of these people.

To represent the first point mentioned here, we considered an array where the indexes represent the cities and the values in the indexes represent the maximum capacity or requirement the corresponding city has for a specific profession. We call this our "**Capacity Array**". A figure representing our Capacity Array is shown in 4.1.

Index (City)	0	1	2	3	4	5
Value (Capacity)	6	7	3	5	1	2

Figure 4.1: Capacity Array

Here, in the sample array, we have 6 areas that can be denoted as $a_0 - a_5$. The indexes of this array represent the 6 cities we have taken under consideration for this illustration. Each of the values in the array holds the number of vacancy available for a profession. So, city number 0, or a_0 has 6 vacancies, a_1 has 7 vacancies and so on. This array is one of the inputs that we must give to our optimization framework.

Coming to the second point mentioned previously, another input of our optimization framework is a list of employees and their details. If we have p number of employees, then the size of this list will be p. A sample example is shown in figure 4.2.

e_1	e_2	e_2	e_3	....	e_p

Figure 4.2: List of Employees

Here, each of the entries $e_1 - e_p$ represents employees we want to allocate to our n number of cities.

The output of our algorithm will be an integer array, where the indexes will represent the employees and each of the values will be a number representing a city. An illustration is shown below in figure 4.3.

Here, the indexes of the output array, denoted by $e_1 - e_p$ are the employees whom we are taking into account. The values are the cities to which the corresponding employee is being allocated.

Index(Employees)	e_1	e_2	e_2	e_3	....	e_p
Value (Cities)	0	5	5	1		3

Figure 4.3: Output of Optimization Framework

This is how we will be allocating p number of employees to c number of cities. For finding the optimum allocation, we have used NSGA-II algorithm. A flowchart is shown previously in figure 3.7. Below, in figure 4.4 we have modified the previous flowchart to fit our particular experiment.

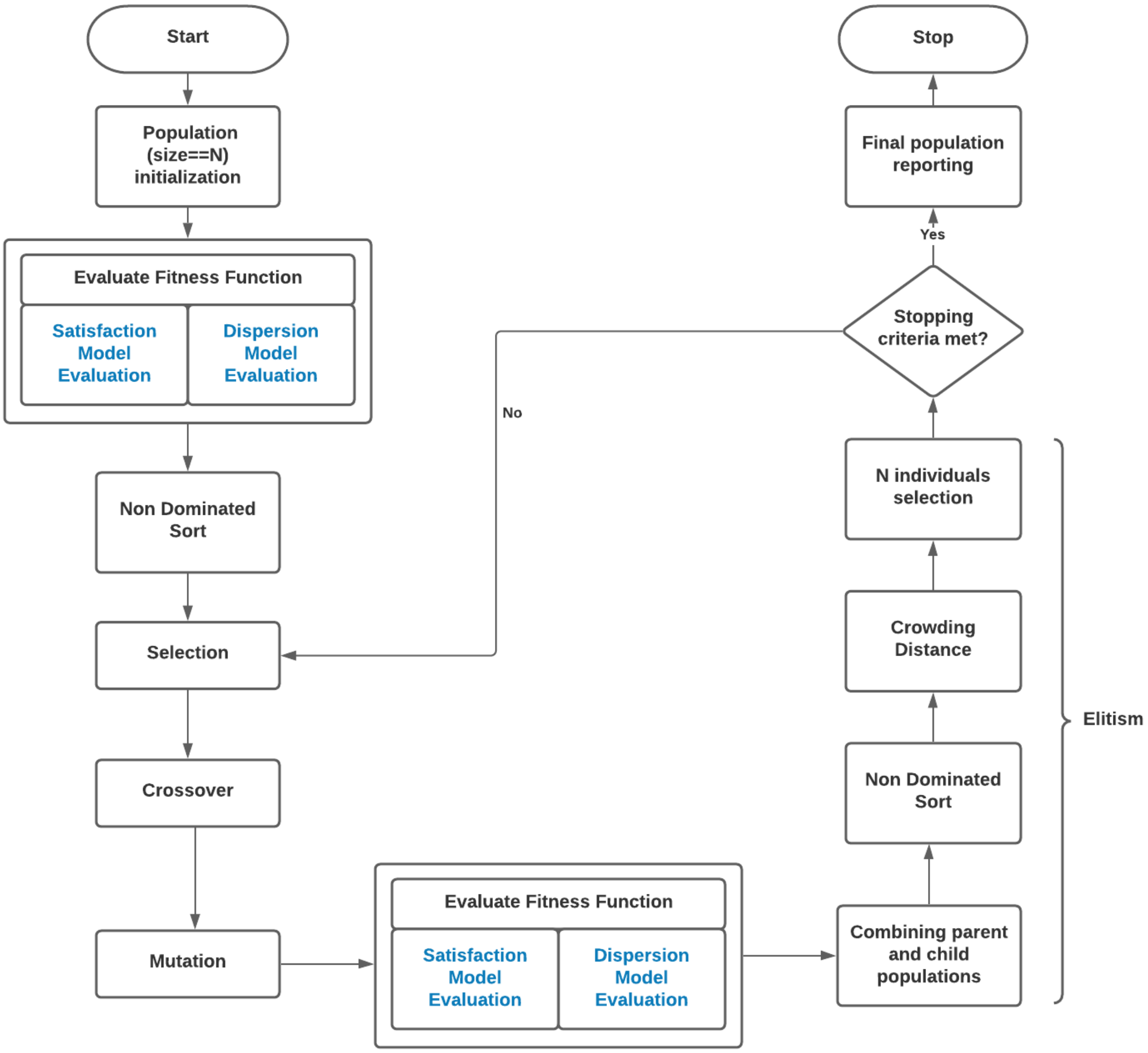


Figure 4.4: Modified NSGA-II flowchart: modified for our experiment

Here, the algorithm starts with our capacity array and employee array. It initializes a population of size N where $N = p$. After doing a non-dominated sort, it will evaluate our two objectives: satisfaction and dispersion, and will perform a population ranking based on the obtained scores. Then, it will select the parents and will perform crossover and mutation respectively. The algorithm will again measure the satisfaction and dispersion scores for the new solutions and will rank the entire population consisting of the parents and children. From this solution pool, the algorithm will choose N number of solutions. This process will keep continuing until a stopping criteria is met.

4.3 Objective Formulation

- **Satisfaction (Using Neural Network):**] One of the two objectives that we want to maximize is the satisfaction of the employees. It refers to a number that denotes how much satisfaction, on a scale of 1 to 5, an employee has achieved within a certain allocation. If we consider that there are n employees, then for an allocation, the maximum satisfaction we can get is

$$\sum_{i=1}^{n} 5 = 5n \tag{4.1}$$

- **Dispersion:** Our second objective is to increase the dispersion of our solution, which means, to make our allocation as well dispersed as possible. This objective opposes our previous objective, which is to increase the satisfaction of our employees. So the two objectives are inversely related to each other.

4.4 Phase 01: Satisfaction

In section 4.4, we will discuss the first phase of our methodology, i.e. satisfaction. We will discuss the necessity of neural networks, how our features were selected as well as the tasks we needed to perform to reach our goal in the first phase. We discussed each sub-step that was needed to reach our goal of forming a Satisfaction model.

4.4.1 Feature Selection:

In order to identify which features impact employee satisfaction, we performed a social experiment. Each of the four members of our group had selected 5 people each. For the sake of avoiding any kind of bias, none of these people were given prior knowledge that we would be extracting information from the conversation we were about to have with them and would use it as our input parameters.
We had casual conversations with them with a very simple underlying topic: if they ever had to relocate to another city for their jobs, what would their concerns be about the city where they would be reallocated to, and what factors of the city would make them interested in relocating. The **most common answers** that we found are presented below.

- All 20 people of our population set were concerned about the security of the new city. They would not move to any city that has high crime rates.

- 17 of them mentioned that they would be happier if they could stay closer to their family.
- 5 of these people were married, and 3 of them do not want to move without their spouse, whereas 2 of them did not have such a problem.
- 4 out of the 5 married people also mentioned that having good schools in their area would be important to them.

Apart from these, there were some other factors that were mentioned throughout the conversation, but was not put much emphasis on. Such as, the availability of developed roads and constructions, internet speed, recreational opportunities, restaurants, cafes and diners of the new city.

However, after we introduced the point that house rent of the city may also be an issue, as some cities do tend to have higher house rent than the others, all of them agreed that this is also a major point that they will consider.

We have also noticed that the answers were varying based on the gender, age and occupation of the person.

So, from this experiment, we chose to include the following factors in our data collection process, which would later on be used as input features of our model.

- Gender
- Age
- Occupation
- Security
- House Rent
- Distance from Hometown
- Schooling
- Marital Status
- Spouse's willingness to move with their partners

4.4.2 Tasks

We have divided our task of creating the model into these following sub-tasks. A flowchart of the tasks followed by the description of each of the tasks is provided below in 4.5.

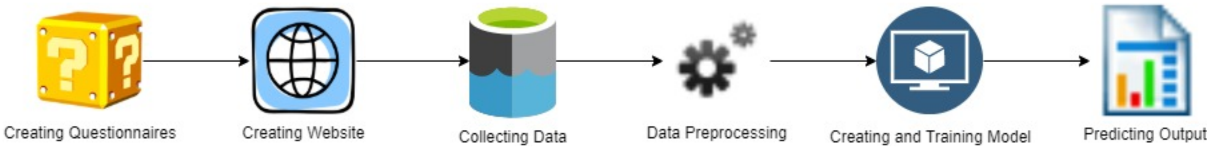


Figure 4.5: Flowchart of Tasks

4.4.3 Necessity of Data Acquisition

In order to train our Neural Network that will predict satisfaction of a worker, we must feed it with enough data. Unfortunately, there is no curated dataset for this purpose. So, we had to create our own dataset by circulating questionnaires. The details have been presented below.

4.4.3.1 Creating Questionnaires

Our questionnaires are comprised of questions regarding the following:

- First we ask some demographic questions, i.e. their age, gender, occupation, field of study, marital status, occupation of spouse and willingness of their spouse's moving in case the respondent has to move to a different city for their jobs.
- We ask questions about the 5 factors we think may have an impact on employee allocation, which are: marital status, security, schooling, house rent and distance between the designated area and the hometown of the respondent.
- Among the five satisfaction factors, only "marital status" has been kept as a binary demographic question, while the rest 4 have been used to form different virtual scenarios for the respondents. These 4 factors have 3 levels each: low, medium and high. The concept explained in the previous paragraph can be further clarified by figure 4.6.
- So there are a total of 3^4 or 81 possible scenarios. It was not practical to ask each respondent about all of these 81 virtual scenarios, so we had to carefully partition the set of 81 scenarios to smaller sets. So, we made 27 disjoint sets of questionnaires and it was absolutely vital that these 27 sets are distributed evenly among the respondents.

4.4.3.2 Creating Survey Collection Website

Since it was important for us to make sure that each of the 27 sets are being circulated evenly, we had to create a website of our own. So, we made a website using the MVC framework of

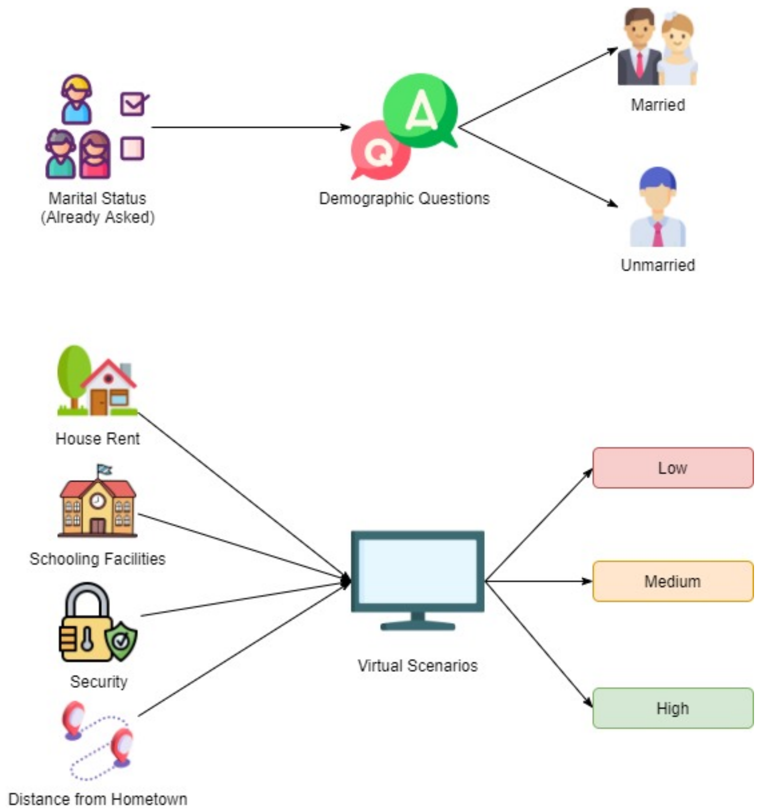


Figure 4.6: Formation of Virtual Scenarios

ASP.NET and used an algorithm that would ensure that all of the sets are being distributed evenly. Some important pages of the website have been shown below:

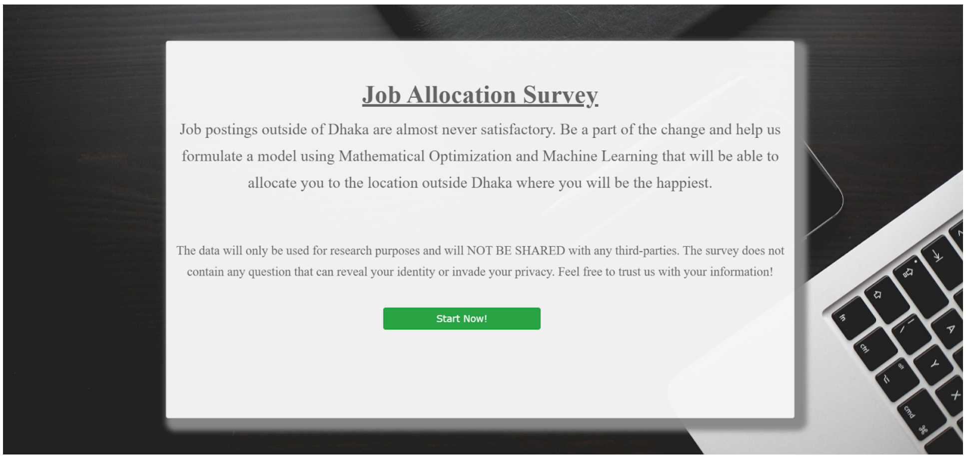


Figure 4.7: Home Page

Next we asked some demographic questions.

Finally, we gave the respondent 3 virtual scenarios, and asked them to rate their satisfaction on a scale of 1 to 5 under each of the three circumstances. A demo scenario is shown in this figure. Here we are telling the respondent that they will be allocated to an area that has the following features,

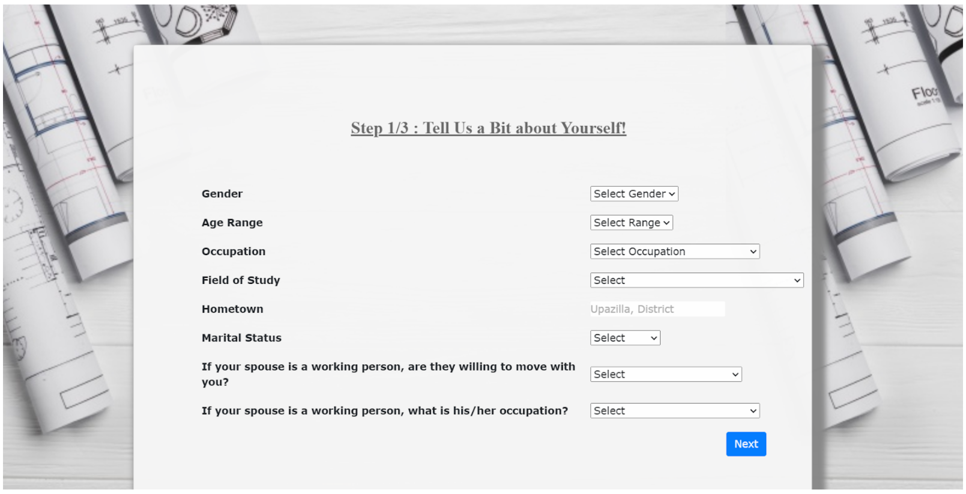


Figure 4.8: Demographic Questions

1. Security is Medium;
2. Schooling, Average House Rent and Distance from Hometown is Low.

We then asked them to rate their level of satisfaction if they had to live in the said area, on a scale of 1 to 5. Each respondent is presented with three such scenarios, which were chosen for them pseudo-randomly.

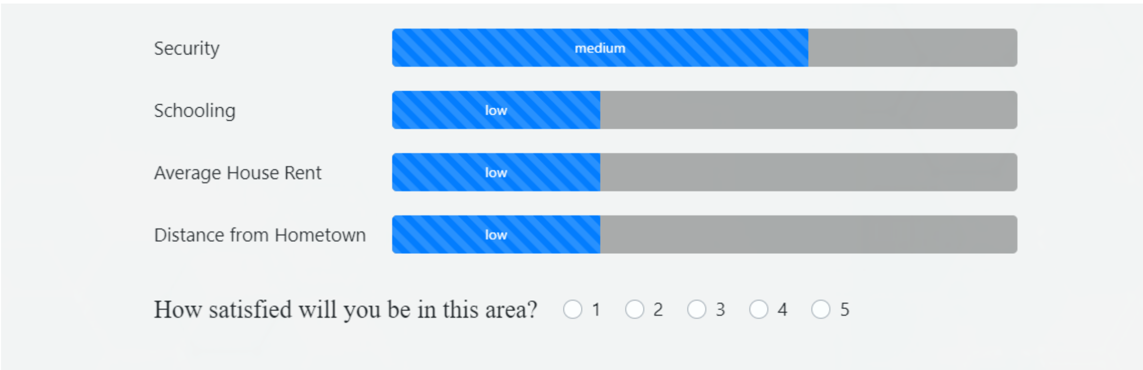


Figure 4.9: Sample Virtual Scenario

4.4.3.3 Collecting Data

The next step is to collect data. We have circulated the website for about a month, and have collected 855 data points in total.

4.4.3.4 Data Preprocessing

Following data collection, we had to apply some pre-processing on our data. We did standardization and normalization on our dataset prior to training.

4.4.3.5 Creating and Training Model

Next step is to create and train neural network models using some trial and errors by tuning the hyperparameters. Here, we varied each of the hyperparameters in turn, while keeping the other ones unchanged, to observe the difference created due to changing each. Ultimately, the hyperparameter values which gave the best results were finalized.
We have modeled our problem in two ways to see which one performs better. First we modeled it as a 5-class classification problem, where the 5 classes correspond to the satisfaction levels from 1-5.

Secondly, we modeled it as a Multi Layer Perceptron Regressor model, where the output is a continuous value of employee satisfaction.

4.4.3.6 Predicting Outputs

Finally, we can predict the outputs using the most accurate model we have made.

4.5 Illustration of Sample Input and Output

In the table 4.13 below, we have prepared some sample inputs and outputs for our model, which can help better explain how our model works.

Our Input parameters are of two types, as follows:

- **Person Specific:** Gender, Age Range, Occupation, Field of Education, Marital Status, Willingness of Their Spouses to Move with Them for a Job Posting, Spouse's Field of Education, Spouse's Occupation.
- **Designated Area Specific:** Schooling, House Rent, Security and Distance from Hometown

Our output is the satisfaction of the employee in the said area, on a scale of 1 to 5.

We have first numerically encoded the input parameters. The mapping among actual inputs and their corresponding numerical values are presented in the following tables [4.1-4.11]:

Here, we have mapped our features to alphabets for ease of representation in the sample input-output table. The mapping table is shown in table 4.12.

A number of sample inputs and outputs are presented in table 4.13. Output will be the satisfaction of the employee of the said area on a scale of 1-5.

Table 4.1: Mapping of Age Range

Age Range	Numeric Value
20 - 25	20
26 - 30	26
31 - 35	31
36 - 40	36
41 or above	41

Table 4.2: Mapping of Gender

Gender	Numeric Value
Male	1
Female	2
Prefer to not disclose	3

Table 4.3: Mapping of Occupation

Occupation	Numeric Value
Medical Field	1
Engineering and IT	2
Business Field	3
Academia	4
Student	5
Unemployed	6
Others	7

Table 4.4: Mapping of Field of Education

Field of Education	Numeric Value
Medical, Biological or Chemical studies	1
Engineering and IT	2
Business Field	3
Social Studies	4
Other	5

Table 4.5: Mapping of Marital Status

Marital Status	Numeric Value
Married	1
Unmarried	2

Table 4.6: Mapping of Willingness of the Respondent's Spouse's Moving with Them

Spouse Willing	Numeric Value
Yes	1
No	2
My Spouse Does Not Work	3
I am not Married	4

Table 4.7: Mapping of Spouse Occupation

Spouse Occupation	Numeric Value
Medical Field (Doctor, Nurse, Nutritionist, Pharmacists and other Health Care workers etc)	1
Engineering and IT	2
Business Field (Management, HR, Banking, Marketing etc)	3
Academia (Teacher, Lecturer, Assistant/ Associate Professor, Professor)	4
Student	5
Unemployed	6
Other	7
I am not Married	8

Table 4.8: Mapping of Security of designated area

Security	Numeric Value
Low	1
Medium	2
High	3

Table 4.9: Mapping of Schooling Facilities of designated area

School	Numeric Value
Low	1
Medium	2
High	3

Table 4.10: Mapping of Rent of designated area

Rent	Numeric Value
Low	1
Medium	2
High	3

Table 4.11: Mapping of Distance from Hometown from designated area

Distance	**Numeric Value**
Low	1
Medium	2
High	3

Table 4.12: Mapping of Features with alphabet

Name of Feature	**Mapped Alphabet**
Age range	a
Gender	b
Occupation	c
Marital Status	d
Spouse Willing	e
Security	f
School	g
Rent	h
Distance	i

Table 4.13: Sample Inputs and Outputs

Input									**Output**
a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	
20	1	5	2	2	4	2	2	3	1
36	2	4	3	1	1	3	1	1	2
26	1	5	5	2	4	1	2	3	2
20	1	2	2	2	4	2	1	1	1
26	1	3	2	1	3	3	2	3	2

In tables 4.14 we have specified our input variables and their mapped numbers for the neural network model we utilised, as seen in figure 4.10. In 4.15, we can observe the output variables of the classification problem are mapped to numbers, which will subsequently be mapped to the neural network model we utilised, as seen in figure 4.10.

Table 4.14: Input Variables

Input node	Input
x_1	Gender of person
x_2	Age of person
x_3	Marital status of person
x_4	If the spouse of this person will move with him/ her
x_5	Occupation of person
x_6	Security of designated area
x_7	Schooling of designated area
x_8	House rent of designated area
x_9	Distance from hometown from the designated area

Table 4.15: Output Variables

Output Node	Output
y_1	1
y_2	2
y_3	3
y_4	4
y_5	5

Description and diagram of the model we are using are explained in section 4.5.1 and figure 4.10 respectively.

4.5.1 Diagram of the Neural Network

4.5.1.1 Classification Problem

In figure 4.10, the inputs are denoted by x_1 to x_9. There can be n number of neurons per hidden layer, where n can be any positive integer. However, since the dataset we are using is small, so the number of neurons per hidden layers should also be small in order to avoid risk of overfitting. Here, AF_n refers to activation functions, which can be ReLU, SELU, GELU etc. For our experiment, we have experimented with ReLU, SELU, GELU and ReLU6. OL refers to output layers. There are 5 OLs since we have 5 outputs. y_n refers to predictions of the neural network. Cross entropy loss provides a probability between 0 and 1 that indicates how close the predicted value is to the actual value. Increase of this loss value, also known

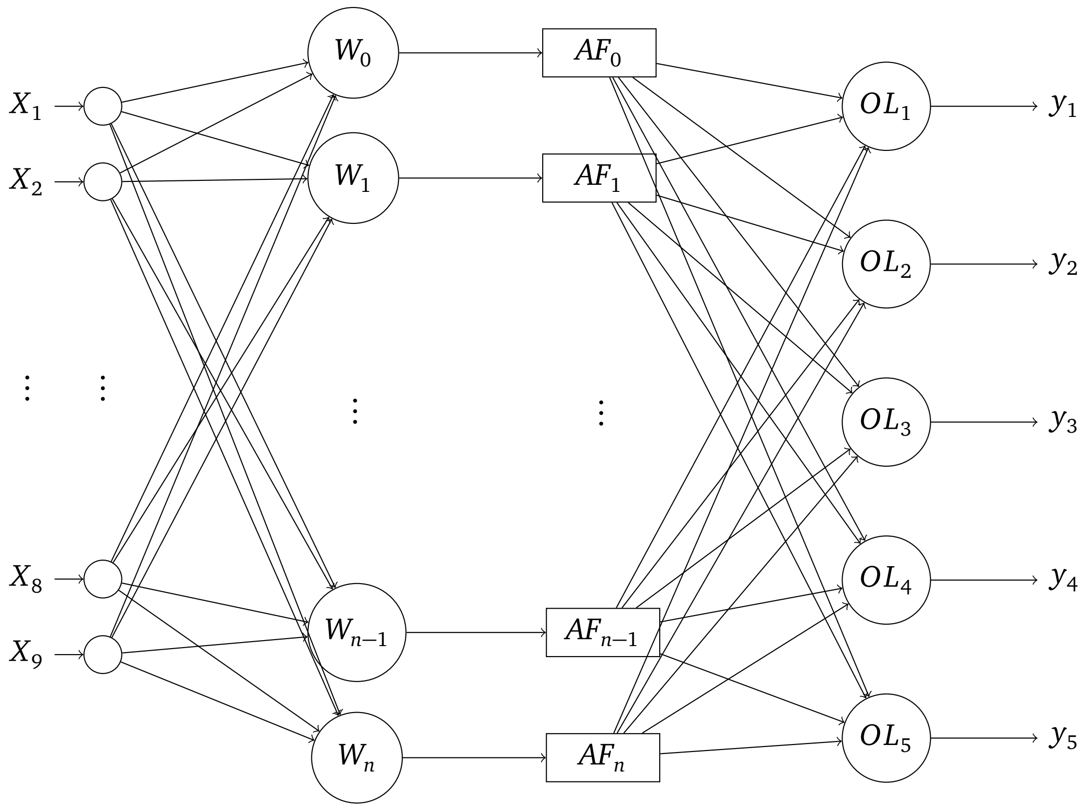


Figure 4.10: Sample Neural Network (Classification)

as log loss, is proportionate to the divergence between actual label and predicted label. It can be denoted as following:

$$L(\hat{y}^{(i)}, y^{(i)}) = -(y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})) \tag{4.2}$$

For the loss function, two situations might arise for extreme values of $y^{(i)}$, which are shown as follows:

1. If $y^{(i)} = 1 : L(\hat{y}^{(i)}, y^{(i)}) = -\log(\hat{y}^{(i)})$ where $\log(\hat{y}^{(i)})$ and $\hat{y}^{(i)}$ should be close to 1.
2. If $y^{(i)} = 0 : L(\hat{y}^{(i)}, y^{(i)}) = -\log(1 - \hat{y}^{(i)})$ where $\log(1 - \hat{y}^{(i)})$ and $\hat{y}^{(i)}$ should be close to 0.

4.5.1.2 Regression Problem

In figure 4.11, The inputs and internal architecture of the neural network is the same as used it figure 4.10. The difference here is that, since it is a regression problem, there is only one output layer and hence, one prediction, which is a continuous satisfaction value. We used squared error loss function here as our loss function. It can be denoted as following:

$$L(\hat{y}^{(i)}, y^{(i)}) = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y})^2 \tag{4.3}$$

Where,
L = Squared error loss function
$\hat{y}$ = Predicted output
$y^{(i)}$ = Actual output
n = Number of samples

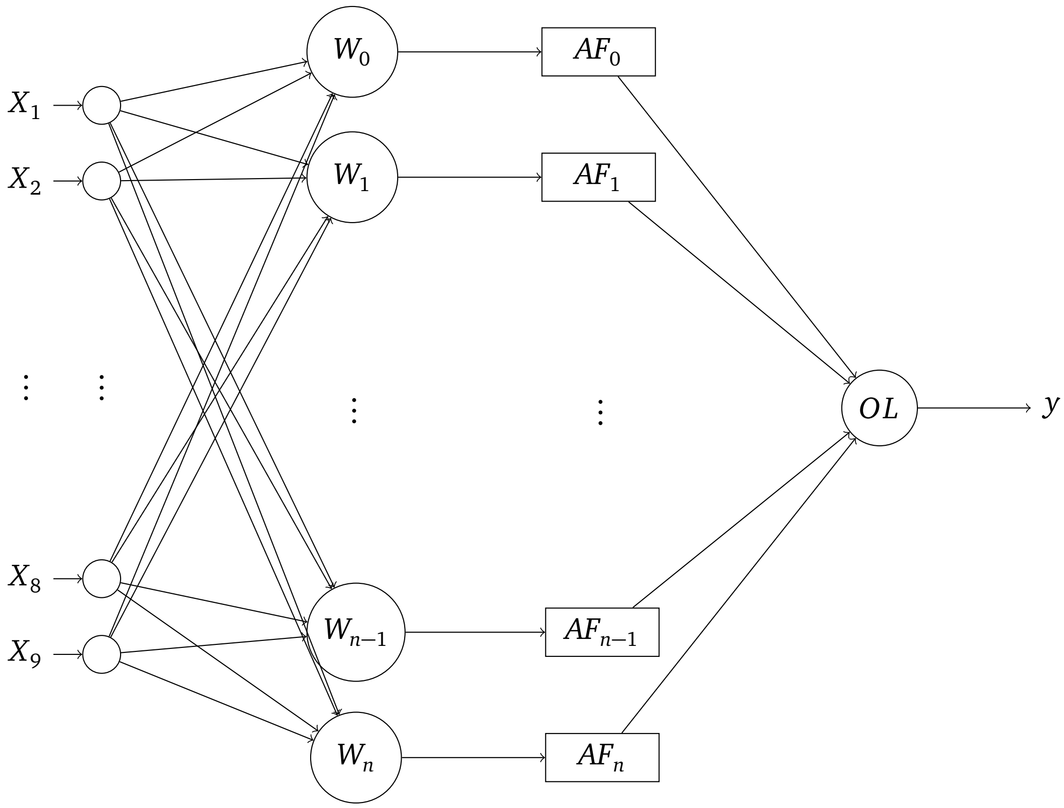


Figure 4.11: Sample Neural Network (Regression)

4.6 Phase 02: Dispersion

The task of the dispersion function is to designate the workers to respective areas in such a way that maximum dispersion or distribution is achieved. As highest satisfaction for workers will ensure that in most cases the majority of workers will be concentrated in a few of the areas i.e. the developed areas, while the less developed areas remain less desirable, so it is the task of the dispersion function to ensure that none of the areas being brought into consideration remain completely worker-less. So if the dispersion of a profession p is denoted by D(p), then the function can be represented as follows:

$$D(p) = |R^{(p)}| + \sum_{i=1}^{n} d_i^p \tag{4.4}$$

Here,

$$R^{(p)} = \{\, i \mid 0 \le i \ \&\ al_i^p \ge minreq_i^p \} \tag{4.5}$$

$$minreq_i^p = \left\lceil \frac{c_a^p * \frac{c_i^p}{\sum_{j=1}^{n} c_j^p} * 100}{100} \right\rceil \tag{4.6}$$

$$= \left\lceil c_a^p * \frac{c_i^p}{\sum_{j=1}^{n} c_j^p} \right\rceil \tag{4.7}$$

$$d_i^p = \begin{cases} \frac{al_i^p}{c_i^p}, & \text{if} \quad al_i^p \le c_i^p \\ \frac{1-(al_i^p - c_i^p)}{c_i^p}, & \text{otherwise} \end{cases} \tag{4.8}$$

Where,
$D(p)$ = Dispersion value of profession p
n = Cardinality of the set of areas brought into consideration
$R^{(p)}$ = Set of regions/areas where minimum requirement of allocation of profession p have been met

$\sum_{i=1}^{n} d_i^p$ = Sum of ratio of allocation and total demand of profession p in all n areas
al_i^p = Allocation of profession p in area i
c_i^p = Maximum capacity of profession p in area i
c_a^p = Available number of employees of profession p
$minreq_i^p$ = Minimum number of employees of profession p that must be allocated in area i

In real world scenarios, we have a shortage of skilled employees, so a company can rarely hire the number of people it actually needs. Let us suppose that a company has 5 branches and needs x_d number of employees in each of these branches, where $d = 0...4$, but the total number of available employee is, let us say, y such that $y \le \sum x_d$. Under such circumstances, it is not desirable to employ most of the employees in one of the five branches while employing few of them in the other branches, because that would effectively leave those less allocated branches at a disadvantage.

To solve this, firstly, we have to make sure that, in each branch, we are allocating at least the minimum number of required people they require. For that, we have introduced a new term named “minimum requirement”, denoted by $minreq$. Suppose we have a vacancy for x number of employees and y number of people have applied and been selected for the positions. We first calculate what percentage of our vacancy, x, can be fulfilled by y. After this, we will distribute the y number of employees to each of the branches as per their demand. So, if branch 1 needs 20% of the newly hired employees, then we assign 20% of y to branch 1. this "20% of y" will yield the number of $minreq$ for branch 1.

Secondly, we thought of the following case. There may be some solutions that allocate more employees in an area than its demand, which also has a negative impact on the rest of the areas, because there already exists a shortage in the overall supply of employees. Allocating more employees in an area than its demand in such a scenario means that we are potentially taking employees away from other areas and allocating them in the area that already has met its employee demands.

So, to tackle this problem, we decided that, d_i^p should be such that: $d_i^p = \frac{al_i^p}{c_i^p}$, if $al_i^p \leq c_i^p$. Otherwise $d_i^p = \frac{1-(al_i^p - c_i^p)}{c_i^p}$. This ensures that the value of d_i^p is is never more that the maximum number of people a branch will need.

4.6.1 Illustration of Sample Inputs and Outputs

In table 4.16, we have presented 7 cities, indexed from 0-6. We have mentioned their required number of employees, and have calculated their minimum requirements.

Table 4.16: Calculating Minimum Requirement

City Number	Capacity	% of Total Capacity	Minimum Requirement
0	6	12.24	4
1	10	20.408	7
2	8	16.326	6
3	10	20.408	7
4	3	6.1224	2
5	4	8.163	3
6	8	16.326	3

In table 4.17, we will calculate the necessary values to calculate our dispersion score.

So, our $R^{(p)} = \{0, 1, 2, 3, 4, 5, 6\}$ and $|R^{(p)}| = 7$
So, dispersion, $D(p) = |R^{(p)}| + \sum_{i=1}^{n} d_i^p = 7 + 5.316 = 12.316$

The representation of data has been further elaborated upon in Appendix A.

Table 4.17: Calculating d_i^p and $A^{(p)}$

City Number	c_i^p	al_i^p	d_i^p	Demand Met
0	6	4	0.666	1
1	10	7	0.7	1
2	8	6	0.75	1
3	10	7	0.7	1
4	3	2	1	1
5	4	3	0.75	1
6	8	3	0.75	1

4.7 Summary of the Chapter

In this chapter, we discussed how we formulated our problem in mathematical terms as well as how we took advantage of the benefits of Deep Learning in order to solve our problems. It gave an overview of the tasks we performed in order to actually solve our problem in a practical manner, moving away from strictly theoretical concepts discussed in the previous chapters. The methodologies we discussed in this chapter will be further used in the upcoming chapter to conduct real experiments in order to observe results. It is basically a transition chapter between the strictly theoretical concepts discussed in Chapters 1, 2 and 3 and the strictly practical work shown in Chapter 5.

Chapter 5

Experiments

5.1 Phase 01: Satisfaction Prediction with NN

In section 5.1, we will discuss the experiments that were conducted in the first phase, i.e. satisfaction prediction. We will discuss the dataset, samples, splitting technique, evaluation metrics used and the results that were obtained from the Neural Network model we utilised to reach our goal of satisfaction prediction.

5.1.1 Dataset

We have a total of 855 data points, which consists of 5 classes in total, which are satisfaction rated from 1-5. The statistics are shown below:

Table 5.1: Statistics of Classes

Class	**Number of Samples**
y_1	187
y_2	159
y_3	220
y_4	157
y_5	132

Bar Chart of the Statistics of Classes is provided below in figure 5.1.

Pie Charts of the demographics and Bar Chart based on their professions are provided below in figures 5.2 and 5.3.

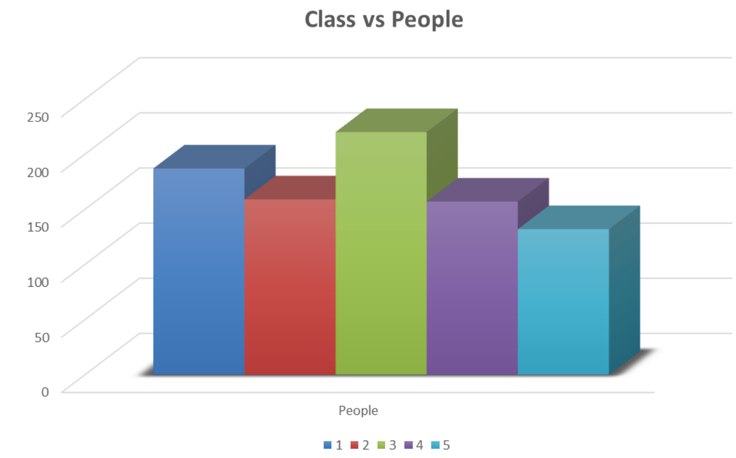
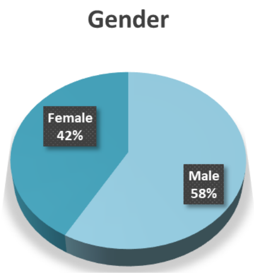
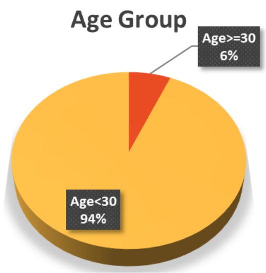


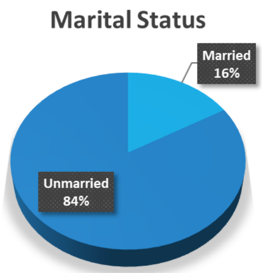
Figure 5.1: Bar Chart of the Statistics of Classes



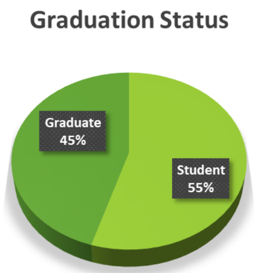
(a) Gender



(b) Age Group



(c) Marital Status



(d) Graduation Status

Figure 5.2: Pie Chart of Demographics

Table 5.2: Samples from the Dataset

Input											Output
a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	
20	2	5	2	2	4	8	3	2	1	3	5
26	1	2	2	2	4	8	3	3	3	3	3
31	2	4	4	1	3	6	1	2	1	3	3
20	1	5	2	2	4	8	3	1	3	1	4
41	2	6	3	1	2	3	1	3	1	1	1

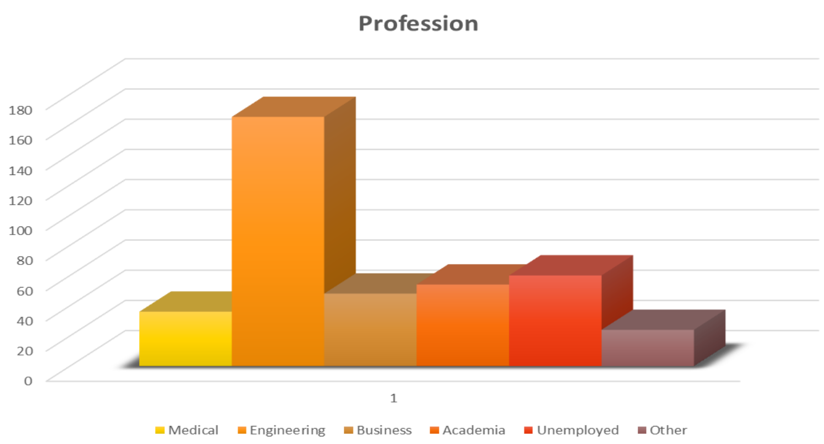


Figure 5.3: Bar Chart Based on Profession

5.1.2 Samples from Dataset

5.1.3 Train, Cross Valid, Test Split

Ratio of training data : Cross Validation Data : Testing Data for our model is 60:20:20.

5.1.4 Evaluation Metric

We will be using accuracy and loss to measure how well our model is performing.

- **List of evaluation metrics:**
  - **Loss:** For calculating loss, we are using cross entropy loss. Cross entropy loss provides a probability between 0 and 1 that indicates how close the predicted value is to the actual value. Loss function has been described previously in 4.5.1.
  - **Accuracy:** Accuracy of the model will be predicted as following:

$$\text{Accuracy} = \frac{\text{Correctly Classified Samples}}{\text{Total Samples}} \times 100 \tag{5.1}$$

5.1.5 Results

In table 5.3, we show the predicted output and actual output for different scenarios. Here, scenarios refer to different input combinations.

Here, the neural network that we are using is a 5-layer ReLU, that has 8 neurons per hidden layer. We took a batch size of 16 and performed 10,000 iterations with 0.0001 learning rate. Since, as mentioned earlier, this problem is trying to capture human psychology, there

Table 5.3: Predicted and Actual Outputs

Scenario	Predicted Output	Actual Output
1	1	2
2	3	5
3	1	1
4	3	1
5	1	3

is a lot of noise in the dataset itself, which has resulted in lower accuracy values. Accuracy on cross validation set for this setting is 32.16%, and for test set it is 35.08%. The accuracy values for cross validation dataset in different hyperparameter settings are shown in Figure 5.4. The setting we have chosen for our result belongs to Setting B of Figure 5.4.

Tables 5.4, 5.5, 5.6, 5.7 demonstrates the experiments we have done and their results for training and cross validation of classification and for training and cross validation of regression respectively.

Table 5.4: Accuracy Table for Training Dataset with Different Settings (Classification)

Batch Size	No. of Iterations	Optimizer	Activation Function	No. of Layers	No. of Neurons per Layer	Learning Rate	Best Accuracy
16	20,000	Adam	ReLU	5	16	0.001	43.86%
16	20, 000	Adam	ReLU	5	16	0.0001	44.44%
16	20, 000	Adam	ReLU	5	16	0.00001	32.16%
16	20, 000	Adam	ReLU	5	16	0.01	42.11%
16	20, 000	Adam	ReLU	7	16	0.0001	40.94%
16	20, 000	Adam	SeLU	5	16	0.0001	42.11%
16	20, 000	Adam	GeLU	5	16	0.0001	43.86%
16	20, 000	Adam	GeLU	7	16	0.0001	42.11%
16	20, 000	Adam	ReLU6	3	16	0.0001	42.11%
16	20, 000	SGD	SeLU	5	16	0.0001	26.32%
16	20, 000	Adargrad	SeLU	5	16	0.0001	24.56%
16	20, 000	Adam	ReLU	5	8	0.0001	40.94%
16	40, 000	Adam	ReLU	7	8	0.0001	43.27%
8	20, 000	Adam	ReLU	7	8	0.0001	40.94%
16	20, 000	Adam	ReLU	7	32	0.0001	44.44%
16	15, 000	Adam	ReLU	7	8	0.0001	44.44%
16	10, 000	Adam	ReLU	5	8	0.0001	32.16%
16	10, 000	Adam	ReLU	3	8	0.0001	38.60%
16	10, 000	Adam	ReLU	3	4	0.0001	32.75%

16	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.0013	51.30%
16	20, 000	LBFGS	ReLU	3	(100, 32, 32)	0.0013	48.65%
16	20, 000	LBFGS	ReLU	3	(8, 8, 8)	0.0013	47.37%
16	20, 000	LBFGS	ReLU	5	(100, 16 16, 8, 8)	0.0013	49.53%

Table 5.5: Accuracy Table for Cross Validation Dataset with Different Settings (Classification)

Batch Size	No. of Iterations	Optimizer	Activation Function	No. of Layers	No. of Neurons per Layer	Learning Rate	Best Accuracy
16	20,000	Adam	ReLU	5	16	0.001	14.619%
16	20, 000	Adam	ReLU	5	16	0.0001	15.78%
16	20, 000	Adam	ReLU	5	16	0.00001	14.61%
16	20, 000	Adam	ReLU	5	16	0.01	29.82%
16	20, 000	Adam	ReLU	7	16	0.0001	28.07%
16	20, 000	Adam	SeLU	5	16	0.0001	28.06%
16	20, 000	Adam	GeLU	5	16	0.0001	29.82%
16	20, 000	Adam	GeLU	7	16	0.0001	28.09%
16	20, 000	Adam	ReLU6	3	16	0.0001	21.01%
16	20, 000	SGD	SeLU	5	16	0.0001	26.32%
16	20, 000	Adargrad	SeLU	5	16	0.0001	24.56%
16	20, 000	Adam	ReLU	5	8	0.0001	24.21%
16	40, 000	Adam	ReLU	7	8	0.0001	20.22%
8	20, 000	Adam	ReLU	7	8	0.0001	21.01%
16	20, 000	Adam	ReLU	7	32	0.0001	19.21%
16	15, 000	Adam	ReLU	7	8	0.0001	16.61%
16	10, 000	Adam	ReLU	5	8	0.0001	27.31%
16	10, 000	Adam	ReLU	3	8	0.0001	12.86%
16	10, 000	Adam	ReLU	3	4	0.0001	28.07%
16	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.0013	51.30%
16	20, 000	LBFGS	ReLU	3	(100, 32, 32)	0.0013	48.65%
16	20, 000	LBFGS	ReLU	3	(8, 8, 8)	0.0013	47.37%

16	20, 000	LBFGS	ReLU	5	(100, 16 16, 8, 8)	0.0013	49.53%

Table 5.6: Accuracy Table for Training Dataset with Different Settings (Regression)

Batch Size	**No. of Iterations**	**Optimizer**	**Activation Function**	**No. of Layers**	**No. of Neurons per Layer**	**Learning Rate**	**Best Accuracy**
8	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.0013	51.30%
16	20, 000	LBFGS	ReLU	3	(100, 32, 32)	0.0013	48.65%
16	20, 000	LBFGS	ReLU	3	(8, 8, 8)	0.0013	47.37%
16	20, 000	LBFGS	ReLU	5	(100, 16 16, 8, 8)	0.0013	49.53%
32	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.0013	51.30%
8	20, 000	Adam	ReLU	3	(100, 8, 8)	0.0013	40.67%
8	20, 000	Adam	ReLU	3	(100, 8, 8)	0.0013	-0.8%
32	20, 000	LBFGS	tanh	3	(100, 8, 8)	0.0013	44.25%
8	20, 000	LBFGS	tanh	3	(100, 8, 8)	0.0013	45.06%
8	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.003	49.15%
8	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.0003	50.95%
8	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.0003	48.45%

Table 5.7: Accuracy Table for Cross Validation Dataset with Different Settings (Regression)

Batch Size	No. of Iterations	Optimizer	Activation Function	No. of Layers	No. of Neurons per Layer	Learning Rate	Best Accuracy
8	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.0013	49.74%
16	20, 000	LBFGS	ReLU	3	(100, 32, 32)	0.0013	44.03%
16	20, 000	LBFGS	ReLU	3	(8, 8, 8)	0.0013	40.88%
16	20, 000	LBFGS	ReLU	5	(100, 16 16, 8, 8)	0.0013	49.53%
32	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.0013	49.71%
8	20, 000	Adam	ReLU	3	(100, 8, 8)	0.0013	37.32%
8	20, 000	Adam	ReLU	3	(100, 8, 8)	0.0013	-0.35%
32	20, 000	LBFGS	tanh	3	(100, 8, 8)	0.0013	45.27%
8	20, 000	LBFGS	tanh	3	(100, 8, 8)	0.0013	46.83%
8	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.003	48.89%
8	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.003	47.84%
8	20, 000	LBFGS	ReLU	3	(100, 8, 8)	0.03	39.84%

The observation from training our model with different hyperparameter settings is that, how easily the model was being over-fitted. The dataset being quite small, a relatively medium neural network with just 8 neurons per hidden layer and 7 hidden layers was leading us towards overfitting. We understood that the model was being overfitted when, for some of the shown settings of 5.4 and 5.5, the accuracy on cross validation set was around 44%, which is quite high given the complexity of our problem, but for the testing dataset, it went down to as low as 12%.

Another observation was that, without considering overfitting, on cross validation sets, minor changes in the hyperparameters did not effect the training accuracy much, with some exceptions. Such as, for the first row in table 5.4, we can see that the accuracy is 43.86%, but if we change the learning rate to 0.0001 as shown in the second row, then the accuracy becomes 44.44%, which is not much of a change. Of course, as shown in table 5.4 and 5.6, some hyperparameter settings have led to drastic decrease in the accuracy.

For our study, regression has outperformed classification in terms of accuracy in both training and cross validation dataset. So, for our final result, we have chosen the first hyperparameter setting in table 5.6 with accuracy 51.30% on training dataset, which performs the best on the cross validation set with the accuracy of 49.74% as shown in table 5.7.

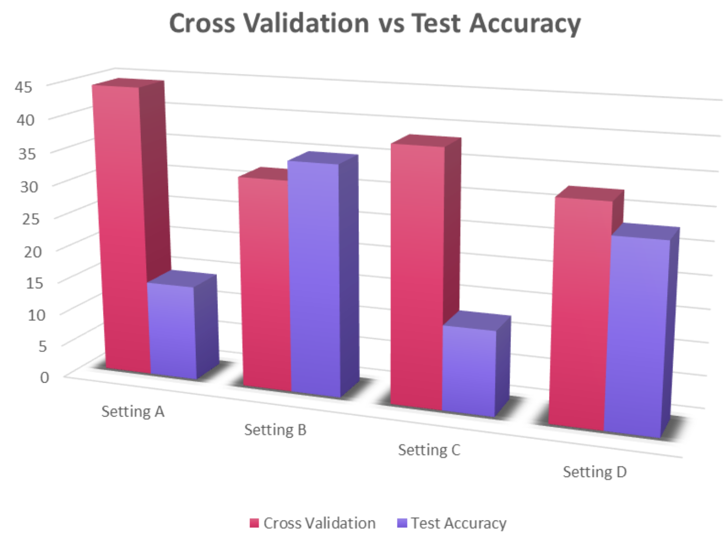


Figure 5.4: Bar Chart of Cross Validation vs Test Accuracy

5.2 Phase 02: Allocation with Optimization Framework

In section 5.2, we will discuss the second phase of experiments we performed, i.e. allocation with optimization framework. We will discuss details about the dataset that was used as well as provide samples from said dataset. The information extracted from said dataset will be shown in the form of visual representations. The evaluation metrics that were used in this phase will also be discussed. Finally, the results obtained in this phase will be discussed.

5.2.1 Dataset

For experimental purposes and in order to objectively observe data without having to consider inter-occupational differences in decision-making, we have chosen only one profession whose employees we want to allocate optimally: medical profession. So, from the 855 datapoints we had previously, we extracted the doctors and their information. In total, we had 37 doctors. We have selected 7 cities where we want to optimally allocate our 37 doctors.

5.2.2 Samples from Dataset

In table 5.8 we have shown some of the information of the doctors that we have extracted.

Table 5.8: Sample Doctors

Serial	Gender	Age Range	Hometown
1	Female	20-25	Dhaka
2	Male	26-30	Mymensingh
3	Female	36-40	Rangpur
4	Male	20-25	Rangpur
5	Female	26-30	Sylhet

A bar chart of the demographics representing their hometowns are given below in figure 5.5.

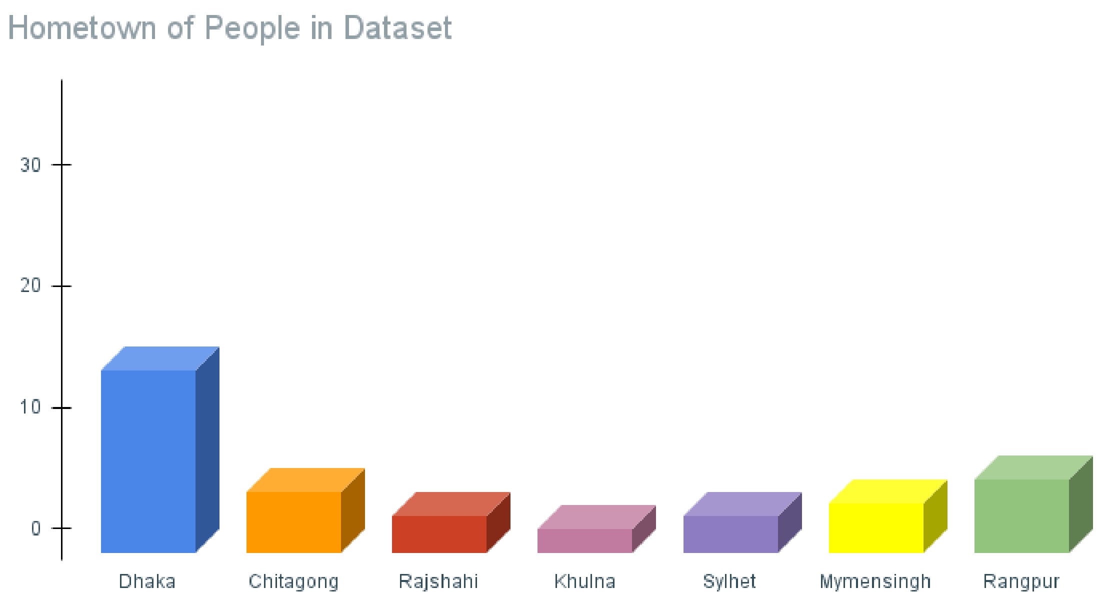


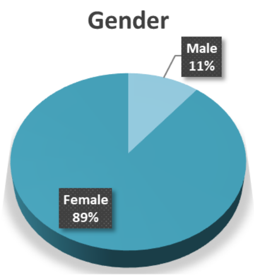
Figure 5.5: Bar Chart of Hometown in Dataset

Pie charts representing the gender and age group of the people in our dataset have been given in figure 5.6.

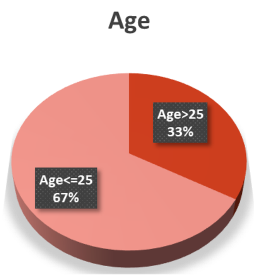
5.2.3 Evaluation Metric

Following is a list of evaluation metrics that we will be using in the second phase of experimentation.

- **Hypervolume:** Hypervolume is a measure of the total volume that is covered by a



(a) Gender



(b) Age

Figure 5.6: Pie Charts of demographics

set of solutions with respect to a pre-defined reference point. It is calculated in the objective space. Higher HV values represent better solutions.

- **Inverted Generational Distance:** The average distance from all the solutions present in the true Pareto Front to to the nearest solution of a predefined solution set is called inverted generational distance (IGD). Lower IGD values indicate better solutions.
- **Spread:** The Spread metric tells us how well a solution is distributed. Lower spread values indicate better solutions.

5.2.4 Results

First, we tried to determine which crossover works better for our experiment. We experimented with Single Point Crossover (SPC) and Multi Point Crossover (MPC). The parameter settings for this experiment is given in table 5.9. We have used NSGA-II here.

Table 5.9: Parameter Setting for Optimal Crossover

Parameter	**SPC**	**MPC**
Population Size	100	100
Crossover Probability	0.9	0.9
Mutation Type	Integer Random Mutation	Integer Random Mutation
Mutation Probability	0.0625	0.0625
Max Evaluations	30000	30000

In table 5.9, we have taken a subset of our total available doctors. The subset size was 16. Mutation Probability here is the default one, which is $\frac{1}{Number of Variables} = \frac{1}{16} = 0.0625$.

The hypervolume for this experiment was 0.66 for both of the parameter settings. A boxplot of hypervolume is shown in figure 5.7. Since both SPC and MPC perform the same, we chose SPC as our crossover operator.

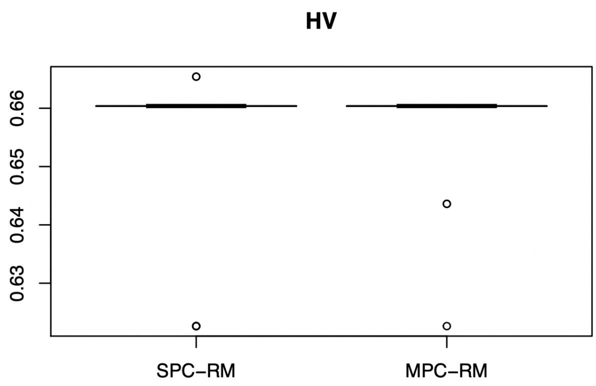


Figure 5.7: Boxplot of HV for Single and Multi Point Crossover

Next we experimented with the mutation probability to find the one that produces the best results. The parameter settings are given in table 5.10.

Table 5.10: Parameter Setting for Optimal Mutation Probability

Parameters \ Mutation Probability	0.1	0.2	0.3	0.9	Default
Crossover Type	SPC	SPC	SPC	SPC	SPC
Population	100	100	100	100	100
Crossover Probability	0.9	0.9	0.9	0.9	0.9
Max Evaluations	30000	30000	30000	30000	30000

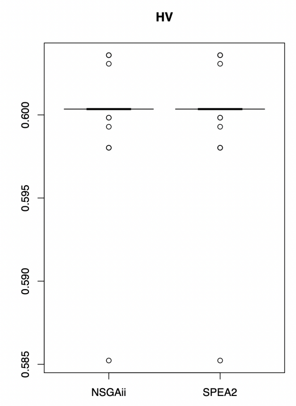
The hypervolume scores for all of these experiments were the same: 0.66. So, we picked the default mutation probability, which is 0.0625 for now. However, here we were experimenting with 16 doctors to initially find the best parameter settings. We have 37 doctors in total, so our default mutation probability will be $\frac{1}{Number of Variables} = \frac{1}{37} = 0.027$.

Next we ran experiments to find the best algorithm for our experiment. We have experimented with NSGA-II and SPEA-II algorithms. Their parameter settings are given below in table 5.11. We have used our entire doctor dataset here so our number of variable is 37 and mutation probability is 0.027.

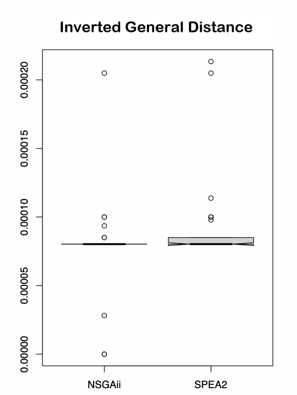
Table 5.11: Parameter Setting for Optimal Algorithm

Parameter	NSGA-II	SPEA-II
Population Size	100	100
Crossover Type	SPC	SPC
Crossover Probability	0.9	0.9
Mutation Type	Integer Random Mutation	Integer Random Mutation
Mutation Probability	0.027	0.027
Max Evaluations	30000	30000

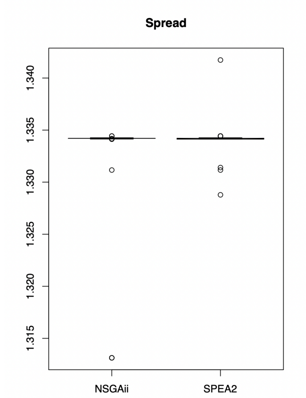
Boxplots of hypervolume, IDG and spread for this experiment has been shown in figure 5.8 where we can see that the two algorithms are producing similar results, so we have chosen NSGA-II as our optimal algorithm.



(a) HV



(b) IGD



(c) Spread

Figure 5.8: Comparison between NSGA-II and SPEA-II

We did one last experiment, where we converted our problem from a multi-objective one to a single-objective one. The task of our algorithm was to maximize the satisfaction of our solution, while keeping the theoretical maximum value of dispersion as a constraint. To determine the theoretical maximum dispersion value, we added constraints on our problem that made sure that in each of the cities, their minimum number of required employees are being allocated. The generated solution from this experiment was also generated by the experiments we did while considering this problem as a multi-objective one. This indirectly yields to the fact that, our multi-objective experiment is exploring almost all of the possible solutions for this problem, and that the single-objective experiment validates the multi-objective one.

The plot of our True Pareto Front is given in figure 5.9.

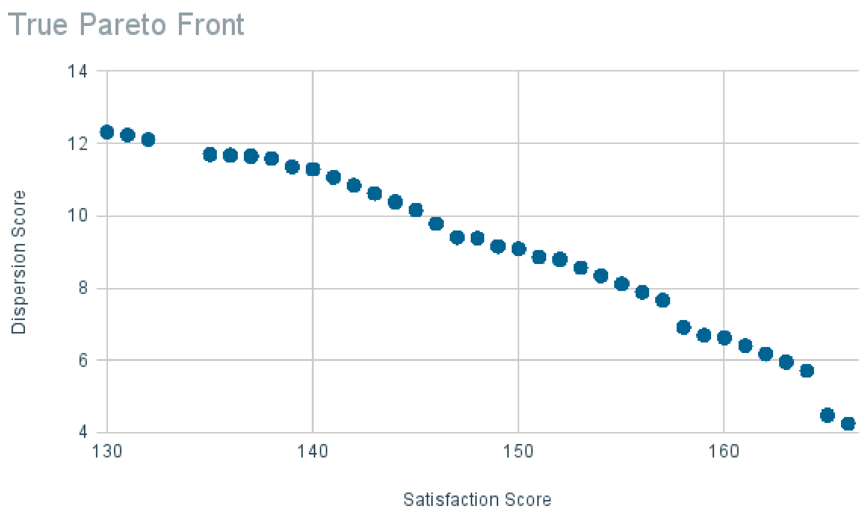


Figure 5.9: True Pareto Front

5.3 Summary of the Chapter

In this chapter, we have discussed the phases in which we have completed our experiments, as well as the results we obtained from the experiments. We discussed how we used our collected dataset as well as how we evaluated our results. It was a representation of the practical work done based on the theoretical concepts we discussed in Chapters 1, 2 and 3.

The experiments we have performed in order the find the best parameter setting for allocation yields to the decision that, the optimization framework will generate more or less the same result irrespective of the hyperparameters. The two experiments we have done to find the best mutation probability and the best algorithm for the experiment supports our previous statement.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Satisfaction is a very subjective criteria. It is not an easy criteria to measure and express in mathematical terms. However, we went a long way towards accomplishing just this, as well as trying to optimize dispersion of people based on satisfaction. We have completed our goal to the best of our capabilities with the resources we had in our hands, but being apprentices, there are lots of scopes of improvements. Perhaps in the future some of these scopes can be taken advantage of in order to further our work.

6.2 Future Work

We have envisioned some opportunities for improvement or research on our work in the future. They are mentioned as follows:

- **Enrichment of Dataset:** Deep Neural Networks perform better with more data, but due to the limitations posed upon us due to the pandemic, we could not physically collect our dataset. As a result, we had a dataset richer with doctors and engineers relative to the other professions. As a result, we conducted our experiments on the dataset of doctors. In the future, upon collection of additional data for other professions, we could perhaps conduct experiments on the other professions as well.
- **Increasing Factors:** We conducted our experiments keeping only 5 factors (Security, House rent, Distance from hometown, Schooling and Marital Status) in mind. However, if further factors can be determined in the future, which affect satisfaction and dispersion, those factors can be incorporated into our experiments as well.

- **Incrementing of Algorithms:** In the experiments we performed, we were only able to test with 2 algorithms(NSGA-II and SPEA-II). In the future, testing with further algorithms might be a possibility.

6.3 Summary of the Chapter

In this chapter, we have discussed about the possible work that can be done in the future based on the work we have done in this thesis. We hope that the work done in our thesis can come to help in the actual allocation of workers in various regions and thus come to help reduce some of the stress in the workplace.

References

[1] "Interim list of categories of essential workers mapped to standardized industry codes and titles." https://www.cdc.gov/vaccines/covid-19/categories-essential-workers.html. Accessed: 2021-06-10.

[2] "Bangladesh bureau of statistics (bbs)." http://203.112.218.65:8008/PageWebMenuContent.aspx?MenuKey=246. Accessed: 2021-06-10.

[3] N. Awofeso, "Improving health workforce recruitment and retention in rural and remote regions of nigeria," *Rural and Remote Health*, vol. 10, no. 1, p. 1319, 2010.

[4] Y. Chen, Z. Yin, and Q. Xie, "Suggestions to ameliorate the inequity in urban/rural allocation of healthcare resources in china," *International Journal for Equity in Health*, vol. 13, no. 34, 2014.

[5] S. Tanihara, Y. Kobayashi, H. Une, and I. Kawachi, "Urbanization and physician maldistribution: a longitudinal study in japan," *BMC Health Services Research*, vol. 11, no. 260, 2011.

[6] A. C. Pereira, "A mathematical model to measure customer satisfaction," *Quality Engineering*, vol. 11, no. 2, pp. 281–286, 1998.

[7] S. Vinerean, I. Cetina, and L. Dumitrescu, "Modeling employee satisfaction in relation to csr practices and attraction and retention of top talent," *Expert Journal of Business and Management*, vol. 1, no. 1, 2013.

[8] W. contributors, "Multi-objective optimization — Wikipedia, the free encyclopedia." https://en.wikipedia.org/w/index.php?title=Multi-objective_optimization&oldid=1008283743, 2021. [Online; accessed 2021-05-01].

[9] P. U. Lecture, "Lecture 9: Multi-objective optimization." https://engineering.purdue.edu/~sudhoff/ee630/Lecture09.pdf. Accessed: 2021-05-01.

[10] C.-L. Hwang and A. S. M. Masud, *Multiple objective decision making, methods and applications: a state-of-the-art survey*. Reading, Massachusetts: Springer-Verlag, 1979.

[11] A. Griffiths, J. Miller, D. Suzuki, and R. Lewontin, *An Introduction to Genetic Analysis*. New York: W.H. Freeman, 7th ed., 2000.

[12] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996.

[13] V. Mallawaarachchi, "Introduction to genetic algorithms — including example code." https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e Accessed: 2021-04-27.

[14] Wikipedia contributors, "Genetic algorithm — Wikipedia, the free encyclopedia." https://en.wikipedia.org/wiki/Genetic_algorithm#Termination, 2021. [Online; accessed 2021-04-27].

[15] TutorialsPoint, "Genetic algorithms - introduction." https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_introduction.htm#:~:text=Genetic%20Algorithm%20(GA)%20is%20a,take%20a%20lifetime%20to%20solve. Accessed: 2021-04-27.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[17] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 3, p. 221–248, 1995.

[18] S. Baluja and R. Caruana, "Removing the genetics from the standard genetic algorithm," *Twelfth International Conference on Machine Learning*, p. 38–46, 1995.

[19] C. R. Raquel and P. C. Naval, "An effective use of crowding distance in multiobjective particle swarm optimization," in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, GECCO '05, (New York, NY, USA), p. 257–264, Association for Computing Machinery, 2005.

[20] Wikipedia contributors, "Pareto efficiency — Wikipedia, the free encyclopedia." https://en.wikipedia.org/w/index.php?title=Pareto_efficiency&oldid=1022104697, 2021. [Online; accessed 2021-04-29].

[21] "Non dominated sorting genetic algorithm (NSGA-II)." https://esa.github.io/pagmo2/docs/cpp/algorithms/nsga2.html, 2021. [Online; accessed 2021-04-28].

[22] P. Calle, "NSGA-II explained!." http://oklahomaanalytics.com/data-science-techniques/nsga-ii-explained/, 2017. [Online; accessed 2021-04-28].

[23] Y. Yusoff, M. S. Ngadiman, and A. M. Zain, "Overview of NSGA-II for optimizing machining process parameters," *Procedia Engineering*, vol. 15, pp. 3978–3983, 2011. CEIS 2011.

[24] L. Bui, D. Essam, H. Abbass, and D. Green, "Performance analysis of evolutionary multi-objective optimization methods in noisy environments," *Proceedings of Asia Pacific Symposium on Intelligent And Evolutionary Systems*, vol. 11, pp. 29–39, 01 2005.

[25] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," 2001.

[26] G. Lesinski, "Application of the strength pareto evolutionary algorithm (SPEA2) approach to rail repair investments," *Department of Systems Engineering United States Military Academy, New York*, 2016.

[27] E. Zitzler, M. Laumanns, and S. Bleuler, "A tutorial on evolutionary multiobjective optimization," in *Metaheuristics for Multiobjective Optimisation* (X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, eds.), (Berlin, Heidelberg), pp. 3–37, Springer Berlin Heidelberg, 2004.

[28] D. Kunkle, "A summary and comparison of moea algorithms," 2005.

[29] "What is the difference between artificial neural network and the human brain?." https://www.verzeo.com/blog-artificial-neural-network-vs-human-brain. Accessed: 2021-06-10.

[30] "Softmax function." https://en.wikipedia.org/wiki/Softmax_function. Accessed: 2021-06-10.

[31] "Exponential function." https://en.wikipedia.org/wiki/Exponential_function?fbclid=IwAR3_Hj6B-S9bMs0Zokw1s1L6k7NiGfI-CgGtpUfD9K7Cs8vTTx0wJCDuG7A. Accessed: 2021-06-10.

[32] "Forward propagation in neural networks — simplified math and code version." https://towardsdatascience.com/forward-propagation-in-neural-networks-simplified-math-and-code- Accessed: 2021-06-10.

[33] "Understanding backpropagation algorithm." https://towardsdatascience.com/understanding-backpropagation-algorithm-7 Accessed: 2021-06-10.

[34] "Understanding activation functions in neural networks." https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884 Accessed: 2021-06-10.

[35] "Various optimization algorithms for training neural network." https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6. Accessed: 2021-06-10.

[36] "Gentle introduction to the adam optimization algorithm for deep learning." https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/. Accessed: 2021-06-10.

[37] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.

[38] "Stochastic gradient descent." https://en.wikipedia.org/wiki/Stochastic_gradient_descent. Accessed: 2021-06-10.

Appendix A

Implementation

We have total 855 data points from our generated questionnaire. For finding the optimal allocation, we extracted one database, consisting of the persons who have filled out the forms and their information, from our datapoints. The second database that we are using contains the information of the areas that we have brought into consideration. So, the two databases that we are using are:

- **Person:** From our responses, we have extracted each person's information, which includes their gender, profession, age range and hometown
- **Area:**: From the 64 districts of Bangladesh, we chose 7 cities and have gathered their information. We have collected each city's corresponding house rent, crime statistics, distance from each of the selected cities and their education qualities. As the metric of education quality, we chose the GPA5 rate of the cities for the year 2019.

Figure A.1 demonstrates the source of our databases.

The 36 entries in our Person database represent "Person" obejcts. Likewise, the 7 cities present in our Area database represents 7 "Area" objects. Their class diagrams are presented in figure A.2

We also need an array on integers, where each index will represent a city and each entry will represent their maximum capacity. We call this our "Capacity Array". A figure representing our Capacity Array is shown in A.3.

For finding the optimum allocation, we need an optimization framework. We have used the built-in implementation of JMetal framework's several genetic algorithms for our optimization. There are two objectives of our multi objective GAs: satisfaction and dispersion.

The main inputs of our optimization framework are:

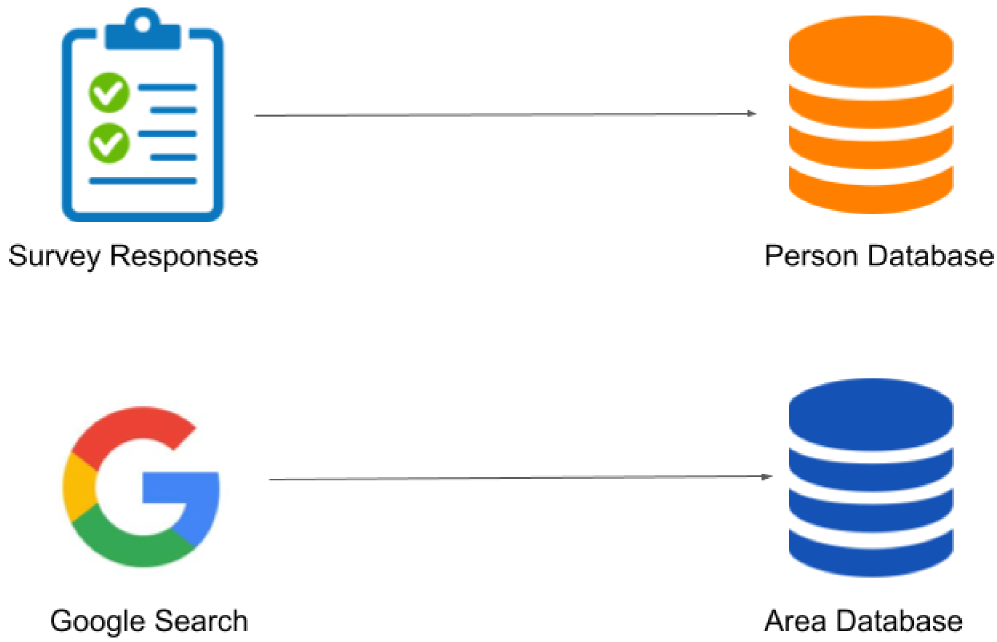


Figure A.1: Sources of Databases

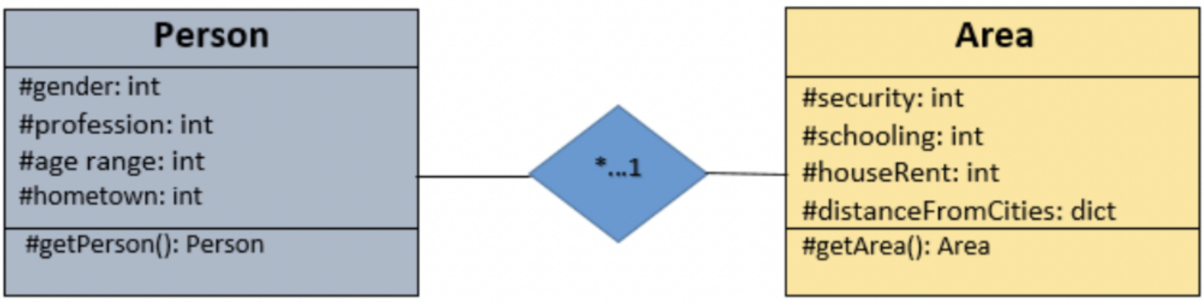


Figure A.2: Class Diagram

Index (City)	0	1	2	3	4	5
Value (Capacity)	6	7	3	5	1	2

Figure A.3: Capacity Array

- Array of Doctors
- Capacity Array:

Output of the optimization framework is an array. Indexes of these arrays represent people, while the values represent which city the corresponding person has been allocated to. So, if we choose to optimally allocate 10 persons of the same profession to 7 different cities, then our output array will be of dimension 1x10, and each of the values will be between 0-6. Figure A.4

Index (person)	0	1	2	3	4	5	6	7	9	9
Value (Allocated City)	1	1	2	2	4	3	5	5	6	6

Figure A.4: Output Array

A flowchart showing how the optimization framework works is figure A.5.

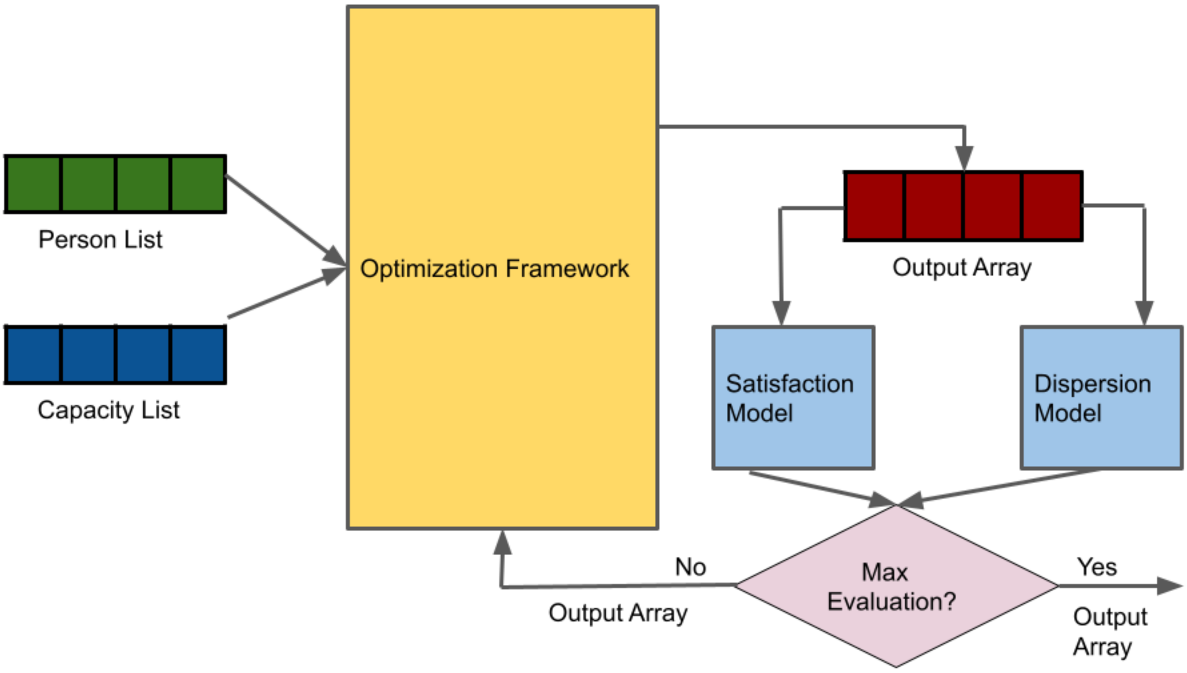


Figure A.5: Optimization Flowchart

Appendix B

Resources and Data sets

- Link to the survey collection website: Website link
- Link to Neural Network: Link to neural network
- Total dataset link: Dataset link
- Allocation Optimization as multi-objective problem: multi-objective optimization code link
- Allocation Optimization as single-objective problem: single-objective optimization code link