

TRANSPORTATION SERVICE SYSTEM FOR GROCERY KART

*A project report submitted in partial fulfilment of the
Requirements for the award of the degree of*

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE

By

KRITI AHLUWALIA (6BE/4013/09)

PARUL JOSHI (6BE/4063/09)

PIYUSH CHAUHAN (6BE/4019/09)

SAMICHEEN KHARIWAL (6BE/4028/09)



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BIRLA INSTITUTE OF TECHNOLOGY

MESRA-835215, RANCHI

Extension Centre Jaipur,

2013

DECLARATION CERTIFICATE

This is to certify that the project entitled “**Transportation Service System For Grocery Kart** ”submitted by Kriti Ahluwalia(6BE/4013/09), ParulJoshi(6BE/4063/09),PiyushChauhan(6BE/4019/09),SamicheenKha riwal(6BE/4028/09) in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering In Computer Science of Birla Institute Of Technology Mesra, Ranchi is an authentic work carried out under my supervision and guidance.

To the best of my knowledge, the content of this project report does not form a basis for the award of any previous degree to anyone else.

Date :

Dr. Shripal Vijayvargiya

(Associate Professor)

Birla Institute Of Technology

Mesra, Ranchi

Extension centre Jaipur

CERTIFICATE OF APPROVAL

The foregoing thesis entitled “**Transportation service for Grocery Kart**” , is hereby approved as a creditable study of research topic and has been presented in satisfactory manner to warrant its acceptance as prerequisite to the degree for which it has been submitted.

It is understood by this approval, the undersigned do not necessarily endorse any conclusion drawn or opinion expressed therein, but approve the thesis for the purpose for which it is submitted.

(Internal Examiner)

(External Examiner)

Name:

Name:

Date:

Date:

ACKNOWLEDGEMENT

We would like to gratefully acknowledge the kind guidance and expert supervision of **Mr. Santosh Sharma** and **Dr. Shripal Vijayvargiya** throughout this work. Their consistent support has always been an immense source of motivation and encouragement.

We sincerely express our thankfulness to **Dr. Madhvi Sinha**, Head of Department, Computer Science Engineering, for allotting us this project and also for resources that were made available to us whenever we needed them.

Lastly, we would likely to thank all the people who have contributed to the successful completion of the project.

Kriti Ahluwalia

Parul Joshi

Piyush Chauhan

Samicheen Khariwal

ABSTRACT

The problem we have chosen is related to on going research in vehicle routing problem with time windows. As per this day there is no deterministic approach invented which will give best result and solves the problem. Many heuristic approaches have been invented to approximate the solution and improve the solution previously achieved by different researchers. We have accepted the challenge of MuSigma named Muphoria which tends to address the delivery of grocery to various stores from a particular warehouse and determine the best paths and determine optimum number of trucks required. Using the multi objective genetic algorithm, we addressed this problem and were able to achieve approximate solutions and optimum trucks and paths. A comparison between two crossover operators is shown and our new crossover operator, the Best Cost Route Operator achieves good results with less population though it takes some running time.

TABLE OF CONTENT

1. INTRODUCTION	9
1.1. PURPOSE	9
1.2. SCOPE	9
1.3. PROBLEM STATEMENT	9
2. SOFTWARE REQUIREMENT SPECIFICATION.....	11
2.1. OVERALL DESCRIPTION	11
2.2. SYSTEM EVENTS AND DATA FLOW.....	12
2.3. SPECIFIC REQUIREMENTS	13
3. SOFTWARE DESIGN SPECIFICATION.....	14
3.1. DESCRIPTION OF PROBLEM	14
3.2. SYSTEM ARCHITECTURE	15
3.3. SYSTEM OPERATION	17
3.4. SYSTEM ARCHITECTURAL DESIGN	18
4. IMPLEMENTATION.....	20
4.1. AUTHENTICATE	20
4.2. SYSTEM DATA	20
4.3. INPUT MODULE	21
4.4. PROCESSING MODULE	24
4.5. OUTPUT MODULE	25
4.6. REPORT GENERATION	28
5. TESTING	29
6. CONCLUSIONS	33
6.1. RESULT ANALYSIS	34
7. FUTURE SCOPE.....	35
8. ADDITIONAL MATERIAL	37
8.1. DEFINITIONS	37
9. REFERENCES.....	41

1. INTRODUCTION

1.1 Purpose

The purpose of this project is to reduce the Time complexity and optimize the resources involved in routing techniques consisting of large data sets using Genetic Algorithm.

1.2 Scope

Genetic Algorithms are powerful and broadly applicable stochastic search and optimization techniques that really work for many problems that are difficult to solve by conventional techniques. Applications of GA are:

- Multi objective Optimization
- Scheduling
- Routing
- Distribution
- Pattern Matching
- Pattern Search within a large search space
- Alternative Solutions
- Network Design

Since multiple objective problems rarely have points that maximize all of the objectives simultaneously, we are typically in a situation of trying to maximize each objective to the “greatest extent possible”. We have an interest in the multiple objective transportation problems.

1.3 Problem Statement

A Grocery kart is responsible for supplying fresh vegetables to all the retail stores in Delhi. A fleet of old matador vans are used to haul the goods from the central warehouse in Rajouri Garden.

- There are about 600 stores in their network.
- Constraints on each van :
 1. All the deliveries are to be made between 5AM to 8AM
 2. Maximum number of stops are 20

3. Travelling distance not more than 65 km
4. Maximum load capacity is 1000 kg
5. At the end of the trip, each truck returns back to the central warehouse
6. Each stop at the store takes 5 min to download
7. Every km travelled takes 90 sec on an average

A new fleet of Mahindra vans are bought replacing the old matador vans in an effort to reduce costs and minimize green house emissions.

- Maintaining each van costs Rs. 20000/month
- The variable cost/km of travel is around Rs.30 for the first 3 years.

2. System Requirement Specification

The content and qualities of a good Systems Requirements Specification (SRS) are described and several sample SRS outlines are presented. This recommended practice is aimed at specifying requirements of application to be developed.

2.1 Overall Description

a) Project Abstract

Project Name : Transportation Service System For Grocery Kart

Project Members : Kriti Ahluwalia , Parul Joshi, Piyush Chauhan, Samicheen

Khariwal

Date : 7-April-2013

b) System Constraints

- **Hardware Constraints**

CLIENT: Manufacturing companies, wall mart, food manufacturers, website users.

MACHINES : Processor- Pentium 4, RAM 1GB.

- **Software Constraints**

CLIENT : The client should have at least, JVM.

DEVELOPER : JDK 1.7

- **Design Standard Compliance**

The system shall be implemented in Java.

- **Software Environment**

Type	Name
1. OS	Windows, Linux
2. Language	Java
3. Library	iText PDF

2.2 System events and Data flow

a) System Activities

System will take data sets of the distance and the requirement of goods from each store. It will then compute the routing map on the basis of various factors :-

- a) Time
- b) Distance to be travelled
- c) Stores to be visited
- d) Load capacity
- Profit, Cost and Efficiency analysis will be done to get optimal solutions.
- Generation of reports through comparative study of various cities and user data.
- Input Data Processing i.e. storing the data and processing it to provide alternative and optimized results.
- Comparative study of best solutions from future generations.

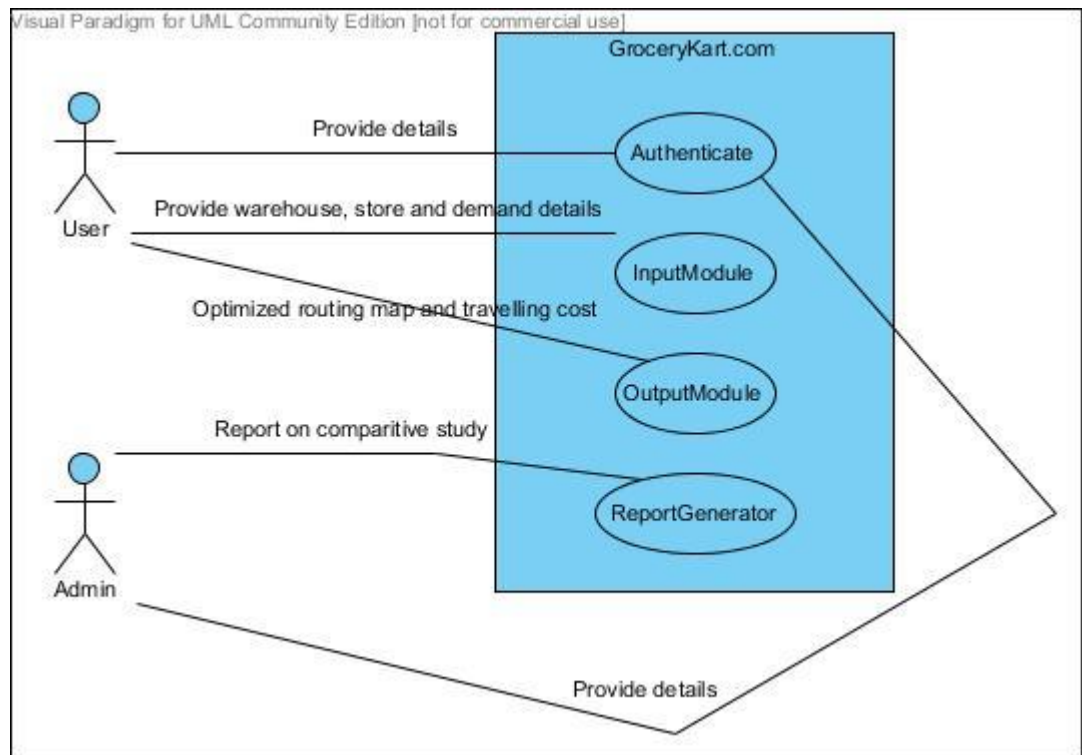
b) User Characteristics

- User should be technologically sound.
- User should be able to analyze the results.

2.3 Specific Requirements

a) Functional Objectives

1. System Use Case :



2. Report Generation :

- Profit Analysis : Cost incurred by Best Solution-Cost incurred by Past solutions.
- Cost Analysis:
 1. No. Of Trucks
 2. No. Of Km travelled Per day
 3. Maintenance charges * (No. Of Trucks) *(years)
 4. No. Ok Km travelled Per day * (cost per KM)
 5. Efficiency of algorithm: Derived Solution cost / previous solution.

3. Software Design Specification

The purpose of this document is to outline the technical design of the Transportation Service System for Grocery Kart and provide an overview for the implementation of the model.

Its main purpose is to –

- Provide the link between the Functional Specification and the detailed Technical Design documents
- Detail the functionality which will be provided by each component or group of components and show how the various components interact in the design
- Provide a basis for the Grocery_Kart system's detailed design and development

As is true with any high level design, this document will be updated and refined based on changing requirements.

3.1 Description of Problem

GroceryKart.com is responsible for supplying fresh vegetables and produce to all mom-and-pop retail stores in Mumbai and adjoining suburbs. There are about 600 stores in their network. They have a fleet of old Matador vans that are being used to haul the goods from their central warehouse in Andheri. Given the perishable nature of the goods, all vans leave at 5:00 AM and all deliveries have to happen before 8:00 AM. Each van can make multiple stops but the union contract prevents any truck from making more than 20 stops or travelling more than 65 KM in any trip.

In an effort to reduce costs and minimize greenhouse emissions, GroceryKart.com is planning to retire all the old Matador vans and purchase a new fleet of Mahindra vans. They believe the cost of purchasing the Mahindra vans will more or less be covered by the sale of Matador vans and the rest of the money will come from the existing savings fund. However, maintaining each truck costs Rs. 20,000 a month (including driver, etc.). Also, the variable cost/km of travel is around Rs. 30 for the first three years of the truck usage. Each Mahindra truck can carry up to 1000 kg at full capacity and at the end of the trip, all trucks return back to the warehouse. Each stop at the store takes 5 minutes to download (irrespective of the weight) and every KM travelled takes 90 seconds on an average.

The distance between each store and the warehouse, the daily demand for each store is provided.

We have to determine the number of Mahindra vans required to be purchased and the total KM that will be travelled to service the daily demand of all the stores so as to minimize the total cost over the first three years?

The output of the code should clearly mention:

- The number of trucks needed
- Routing for each truck and
- The total miles travelled each day

3.2 System Architecture

The system will be constructed from multiple distinct components:

- **Input Module**

Initially, **user_authentication** takes place where he needs to provide the user_id and password. Its function includes:

1. authenticate()

Then he will have to provide main inputs:

- a) Data Centre store matrix
- b) Store Demand Table
- c) Population size
- d) Crossover probability
- e) Mutation probability
- f) Selection percent
- g) No. of generations

These inputs will then be passed on to the second module where the entire processing will take place with the help of classes: grocerykartform, Grocerykart, candidate, adjust factor, store.

- **Processing Module**

Initially with the help of existing data, a population is generated with the help of **population** class.

The functions in this class will be:

- 1) GeneratePopulation()
- 2) ParetoRanking()
- 3) Tournament Selection()
- 4) Crossover()
- 5) Mutation()

The system will then process the user's requirements through the application of a suitable algorithm. The algorithm will be based on genetic approach. Based on these parameters a fitness function will be designed. Through this function we evaluate the **fitness** of each individual by **candidate** class.

The functions in this module will be:

1. calculateFitness()

These functions will be called recursively to create Generations until the stopping criterion is satisfied.

- **Output Module**

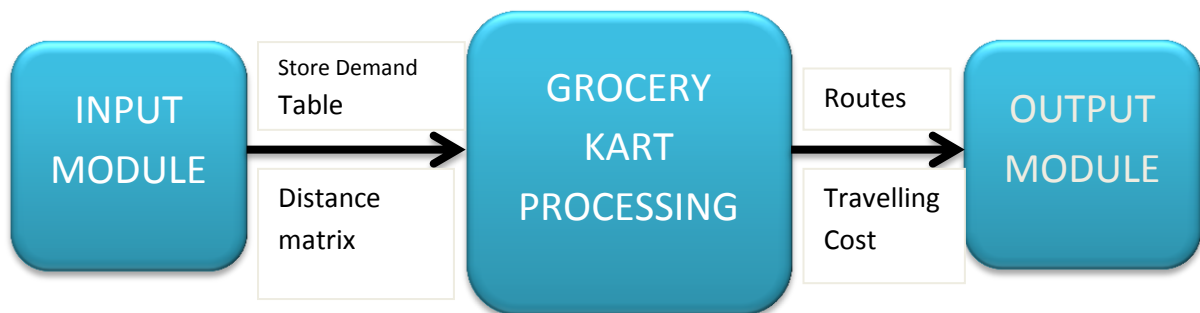
Once an optimal solution is obtained, a report will be generated through **ReportGenerator** class. Its function includes:

1. Reportgeneration()

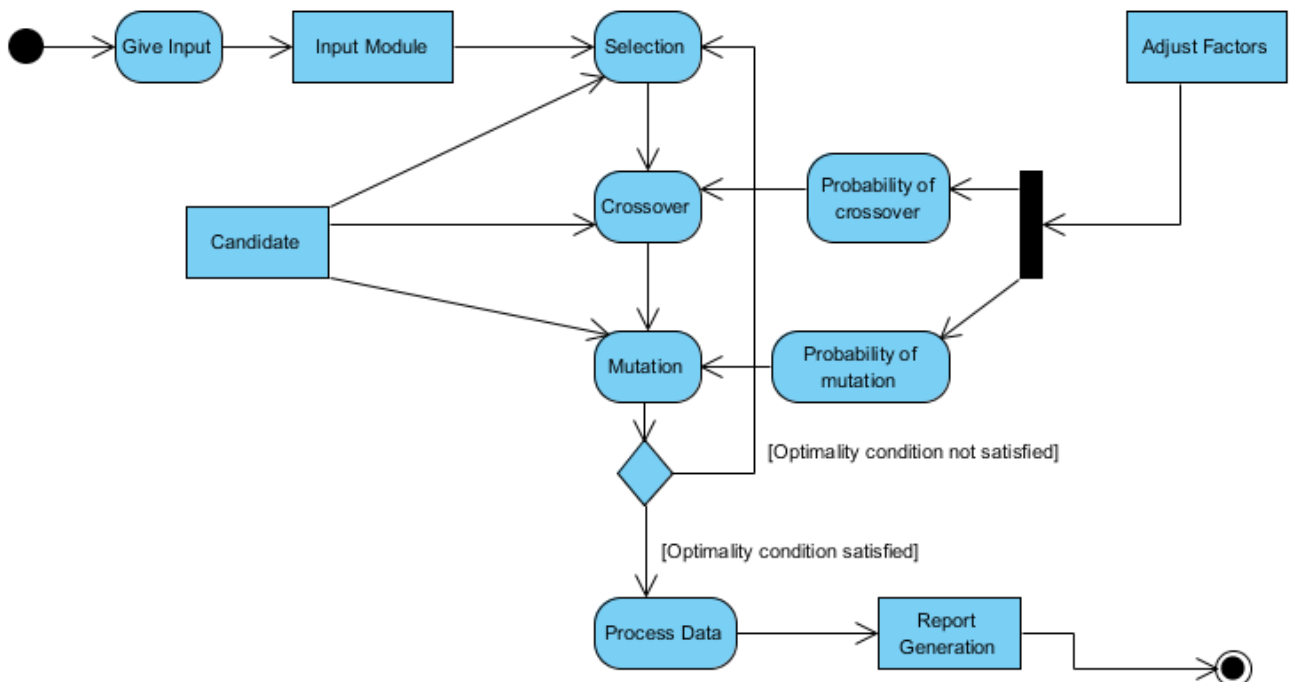
3.3 System Operation

- **Information Flow**

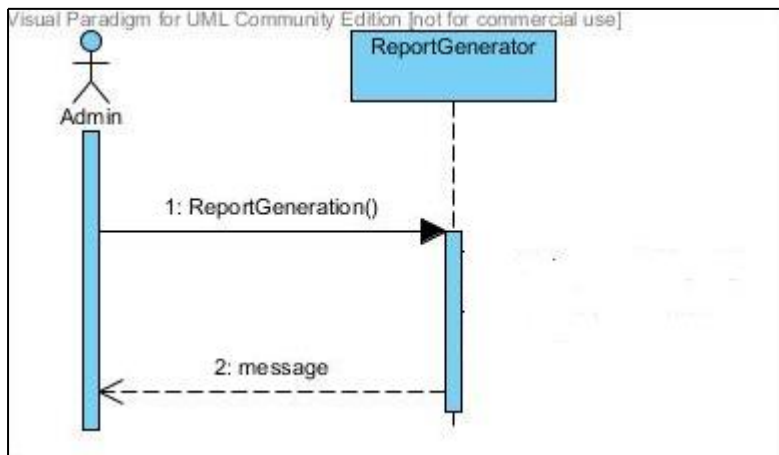
The overall information flow in the system can be shown by the following diagram :



- **Activity Diagram**



- **Sequence Diagram**



3.4 System Architectural Design

3.4.1 Chosen System Architecture

We have chosen Object Oriented Designing for the Transportation Service System For Grocery Kart. **Object-oriented design** is the process of planning a system of interacting objects for the purpose of solving a software problem. It is one approach to software design. An object contains encapsulated data and procedures grouped together to represent an entity.

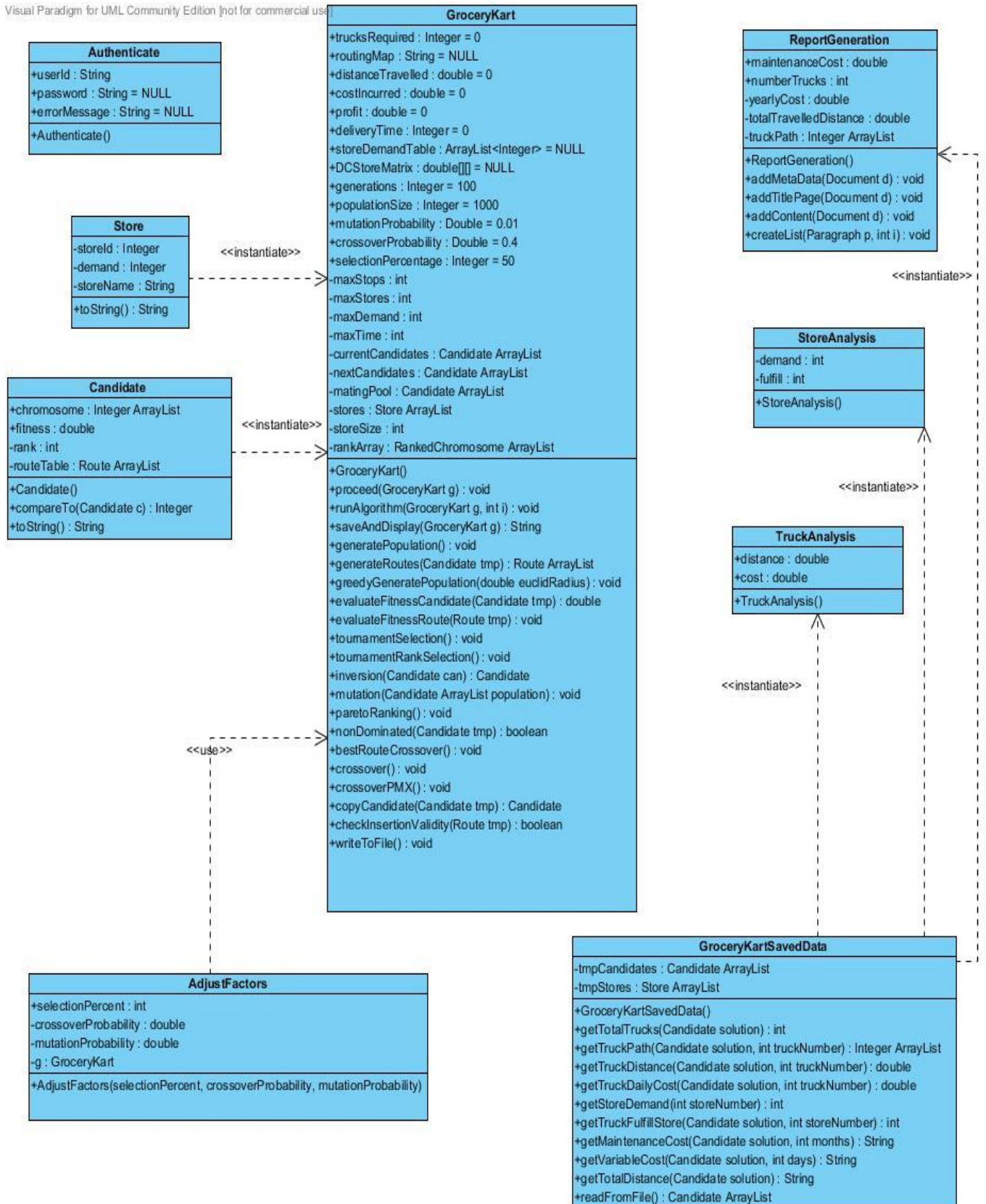
The 'object interface', how the object can be interacted with, is also defined. An object-oriented program is described by the interaction of these objects. Object-oriented design is the discipline of defining the objects and their interactions to solve a problem that was identified and documented during object-oriented analysis.

We have chosen this architecture because of the following reasons:

- Easier maintenance.
- Objects may be understood as stand-alone entities.
- Objects are potentially reusable components.
- For some systems, there may be an obvious mapping from real world entities to system objects

3.4.2 Class Diagram

Visual Paradigm for UML Community Edition [not for commercial use]



4. IMPLEMENTATION

4.1 AUTHENTICATE

Authentication involves the process of identifying an individual, which is usually based on a username and password.

Authenticate(): Default Constructor of Authenticate class which initializes a new User taking User Id and Password as input .The **message** is returned as output by this function.

4.2 SYSTEM DATA

System Data Contains Global Variables, with their Prototypes and default values already set. The Members Of this class are also inherited by Other Subclasses.

Globally declared variables are:

- No of trucks
- Routing maps
- distanceTravelled
- CostIncurred
- Profit
- Efficiency
- deliveryTime
- userId
- storeDemandTable
- DCStore Matrix
- generations
- populationsize
- crossoverProbability
- mutationProbability
- SelectionPercentage

Grocerykartform (): Default Constructor of **Super Class** Grocerykartform is used to initialize all the global variables mentioned above.

4.3 INPUT MODULE

This class is responsible for retrieving input from the user which mainly consists of Store Demand Table and DC Store Matrix, while all other factors are already containing default values, which can be altered using AdjustFactors class described in section . This class is a Subclass of System data.

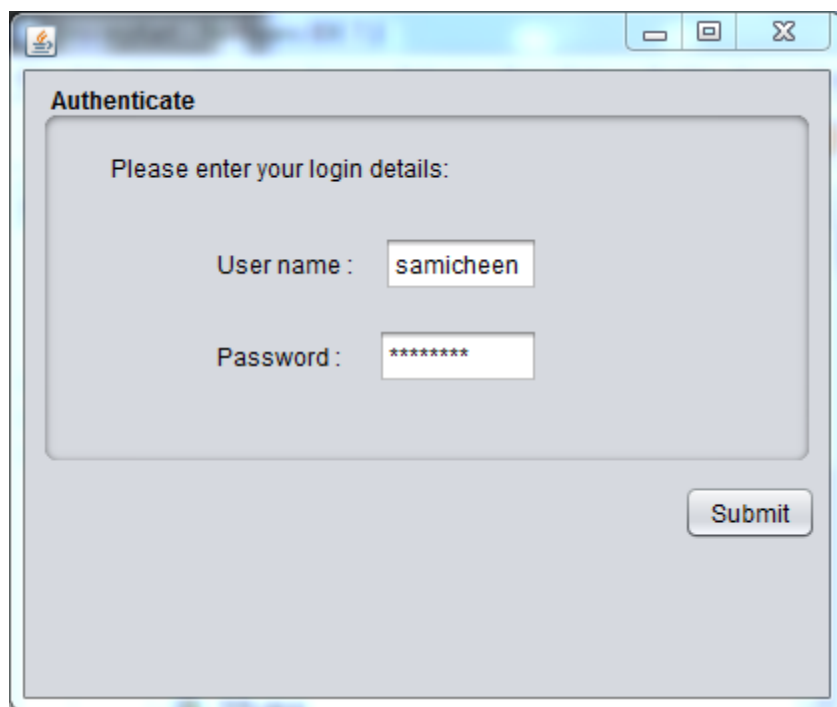
Variables used in this module :

- No of stores : Total number of stores.
- Maxstops : maximum number of stops at which truck an stop
- MaxDemand : total demand of all stores
- MaxDistance : Total distance covered by all trucks in one day
- MaxDeliveryTime : maximum Delivery time within which demand must be fulfilled.

Read data(): default constructor of class Read data. It invokes mainly two functions:

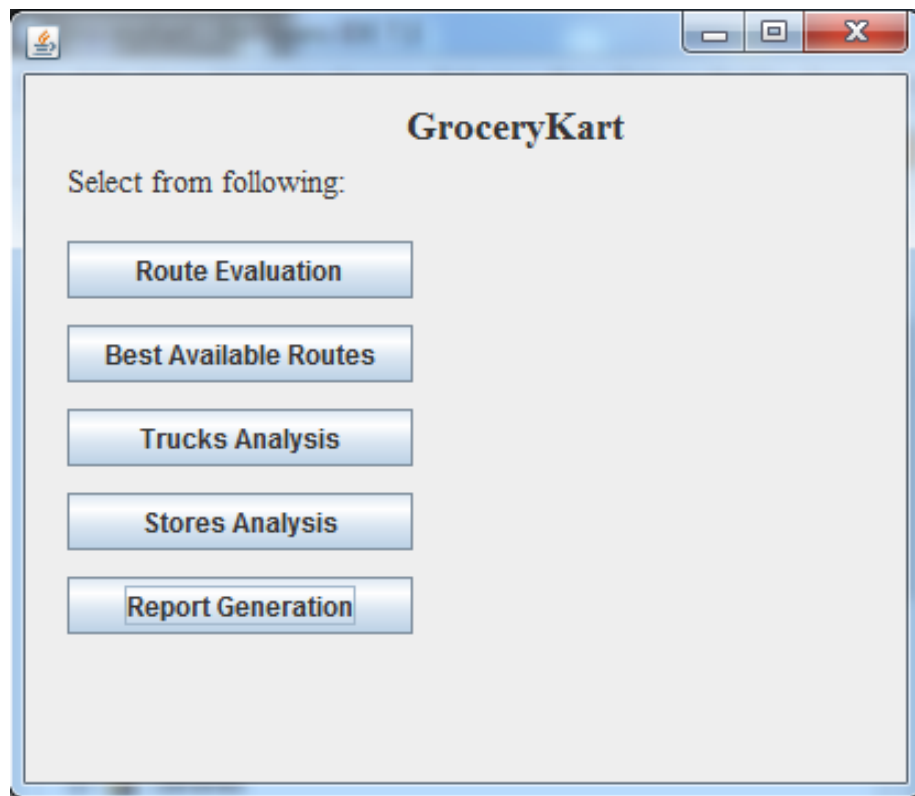
- ReadDemandTable(): it provides the demand of each store.
- ReadDistanceMatrix(): it provides the distance of each store from the warehouse and the distance among the stores.

It is the authentication window which requires the username and password as the input from the user to start the working of the grocery kart.

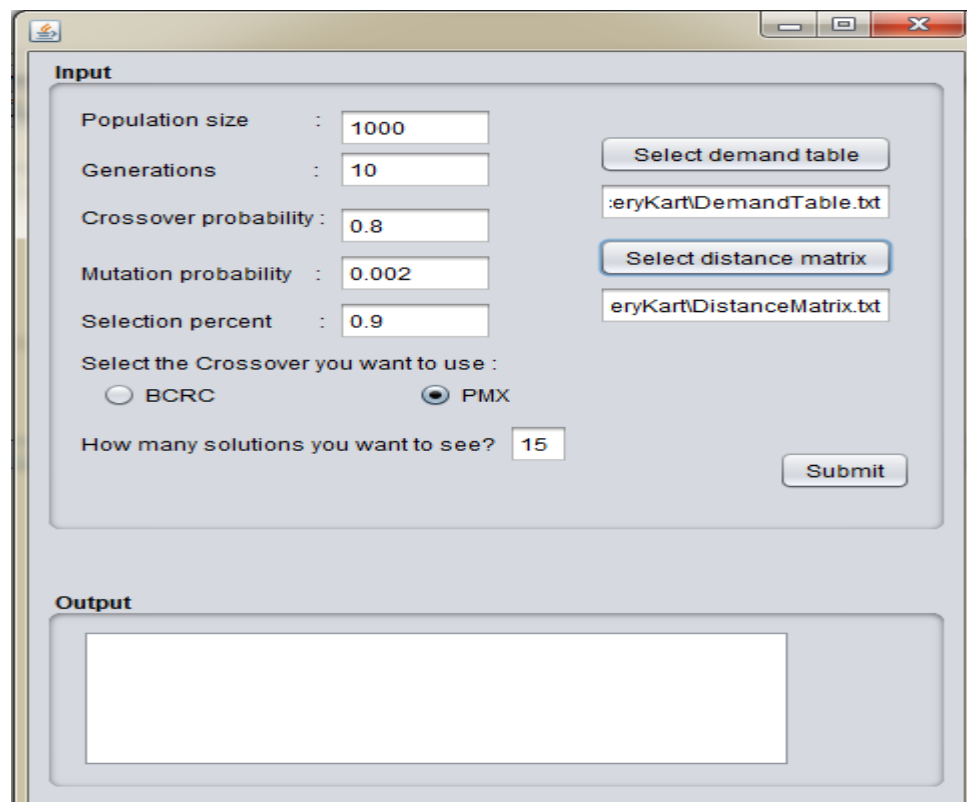


The image shows a screenshot of a software window titled "Authenticate". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area has a light gray background. At the top, the word "Authenticate" is displayed in bold. Below it, the text "Please enter your login details:" is shown. There are two input fields: "User name :" followed by a text box containing "samicheen", and "Password :" followed by a text box containing seven asterisks "*****". A "Submit" button is located at the bottom right of the form area.

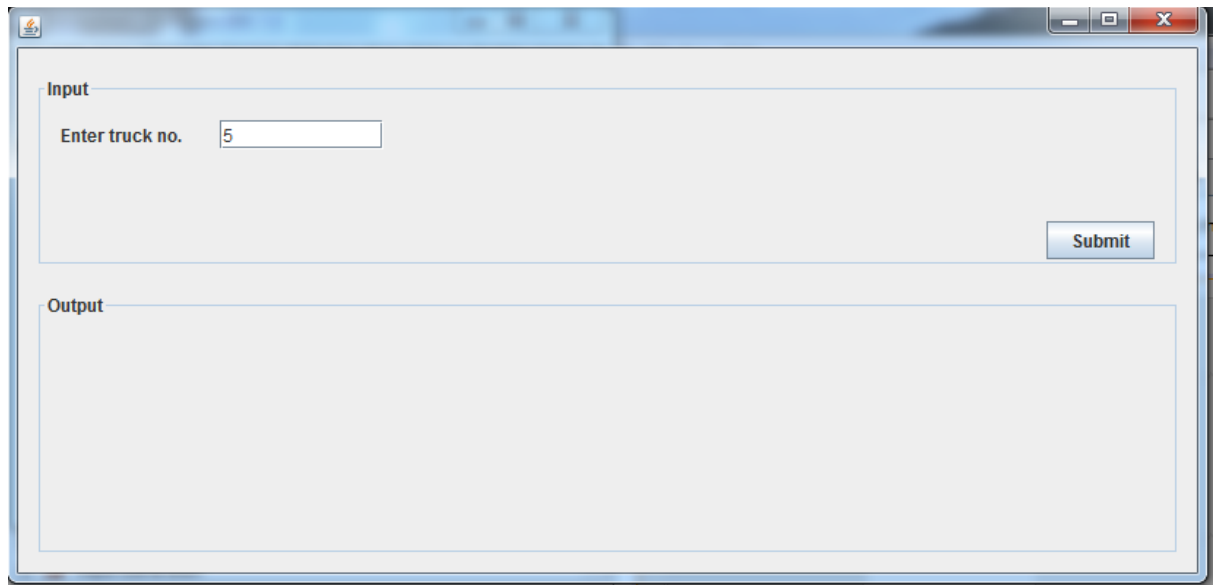
This window is the main welcome window in which the user chooses its options.



This is the route evaluation window. The user inputs the various fields in order to calculate the various optimum results produced providing the trucks required and the routes and distance travelled by these trucks.

The screenshot shows a window titled "Input" for route evaluation. It contains several input fields and buttons. On the left, there are labels and input boxes for "Population size" (1000), "Generations" (10), "Crossover probability" (0.8), "Mutation probability" (0.002), and "Selection percent" (0.9). To the right of these are two buttons: "Select demand table" and "Select distance matrix". Below these buttons are two text boxes containing the file paths "GroceryKart\\DemandTable.txt" and "GroceryKart\\DistanceMatrix.txt". Further down, there is a label "Select the Crossover you want to use :" followed by two radio buttons: "BCRC" and "PMX" (which is selected). At the bottom left, there is a label "How many solutions you want to see?" followed by an input box containing the number "15". A "Submit" button is located at the bottom right. Below the input section, there is an "Output" section with a large empty text box for results.

This is the Truck Analysis window. It takes the truck number as the input value from the user to provide various analysis or result on that particular truck bought by the Grocery Kart.



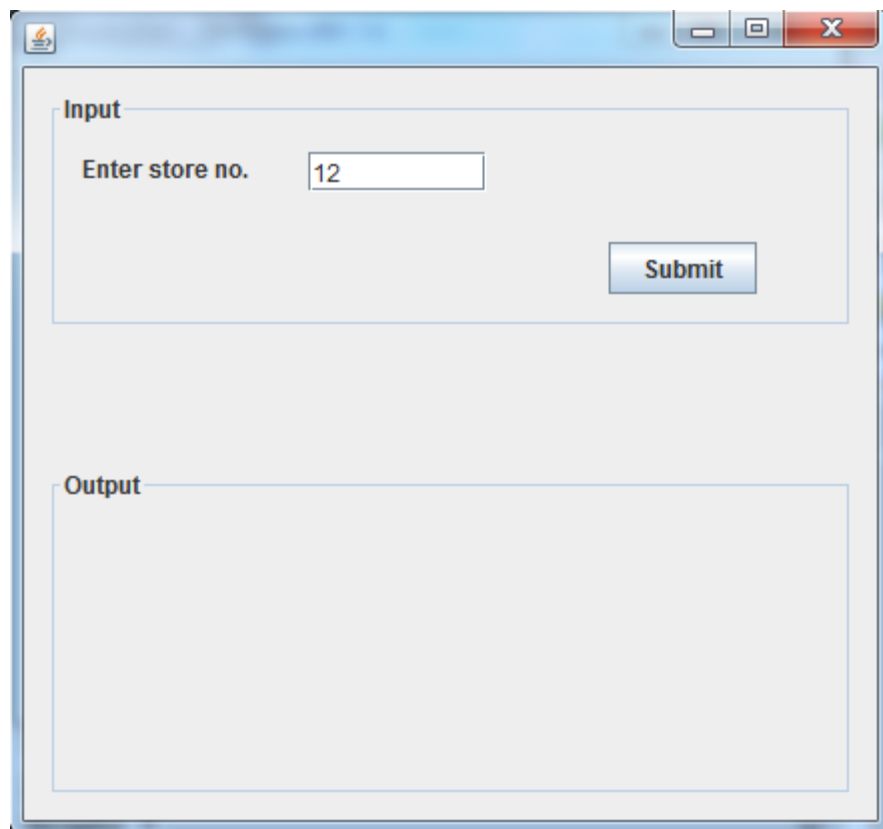
Input

Enter truck no.

Submit

Output

This is Store Analysis window. It takes the Store number as the input value from the user to provide various analysis or other informations on that particular store.



Input

Enter store no.

Submit

Output

- **AdjustFactors Class**

It is responsible for adjusting some factors according to the requirement of the user. Factors such as **mutation probability, crossover probability, and maximum number of stops** can be adjusted easily.

AdjustFactor(): default constructor reinitialises the values of factors to be adjusted.

4.4 Processing Module

The Processing module contains the working mechanism of the algorithm. It defines the flow of randomly generated possible solutions and optimising them to the best solution. It contains the following classes :

- GroceryKart
- Store
- Candidate

4.4.1 GroceryKart Class

This class is the main class in the project, the main functions involved in calculating the optimum solution such as Selection, Mutation, Crossover are invoked by this class. Functions invoked by GroceryKart class are :

- Generatepopulation() :
- ParetoRankinh() :
- TournamentSelection() :
- Crossover() :
- Mutation() :

4.4.2 Store class

This class contains all the detailing about all the stores in the city where the grocery is to be delivered. It serves as an entity and describes all the demand of a store.

The methods which are invoked in this class are :

- toString() : to show the representation of store.

4.4.3 Candidate Class

This class models the solution to be generated. It includes the best routes available and the fitness.

4.5 Output Module

Output Module provides the required results for the user.

Classes used in this module are:

- Store analysis : It includes all the demands of a store and provides which truck satisfies it.
- Truck Analysis : It provides the best path for a truck and total distance travelled by it.
- This is window provides various candidate generation with the total distance travelled and maximum number of trucks required the sorts the result according to the ranks.

The screenshot shows a software window titled "Output" with a standard Windows-style title bar. The window is divided into two main sections: "Input" and "Output".

Input Section:

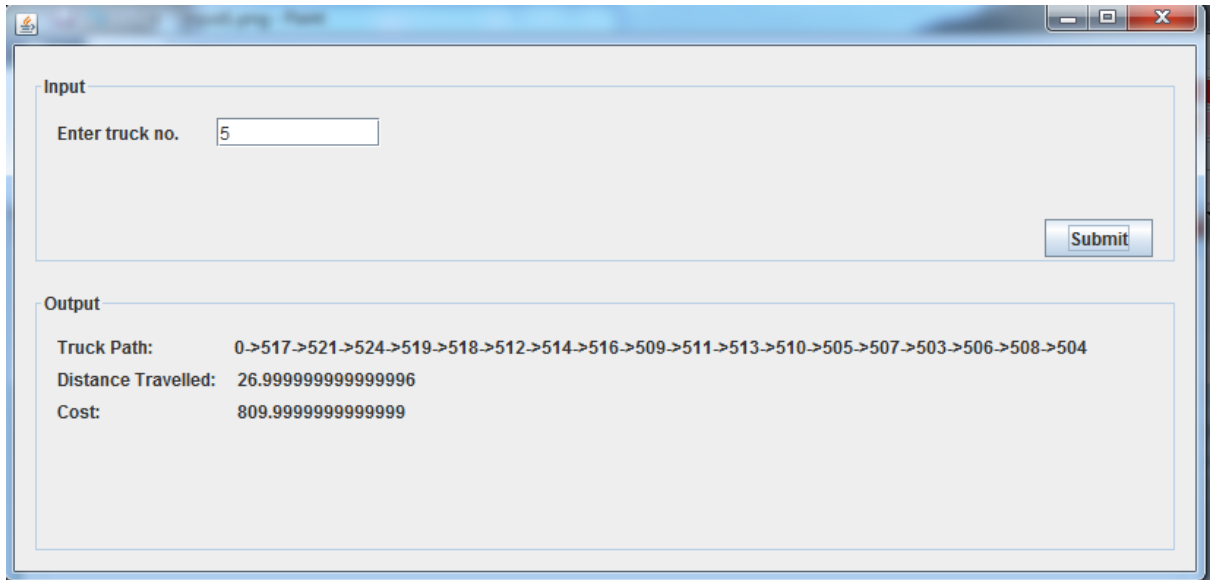
- Population size : 1000
- Generations : 10
- Crossover probability : 0.8
- Mutation probability : 0.002
- Selection percent : 0.9
- Select the Crossover you want to use :
 - ☐ BCRC
 - ☒ PMX
- How many solutions you want to see? 15
- Buttons: "Select demand table" (with file path :eryKartDemandTable.txt), "Select distance matrix" (with file path :eryKartDistanceMatrix.txt), and "Submit".

Output Section:

A text area displaying the following information:

```
Candidate: 1
Distance travelled by trucks:830.9000000000001
Number of Trucks: 30
Rank: 1
Candidate: 2
Distance travelled by trucks:830.9000000000001
```

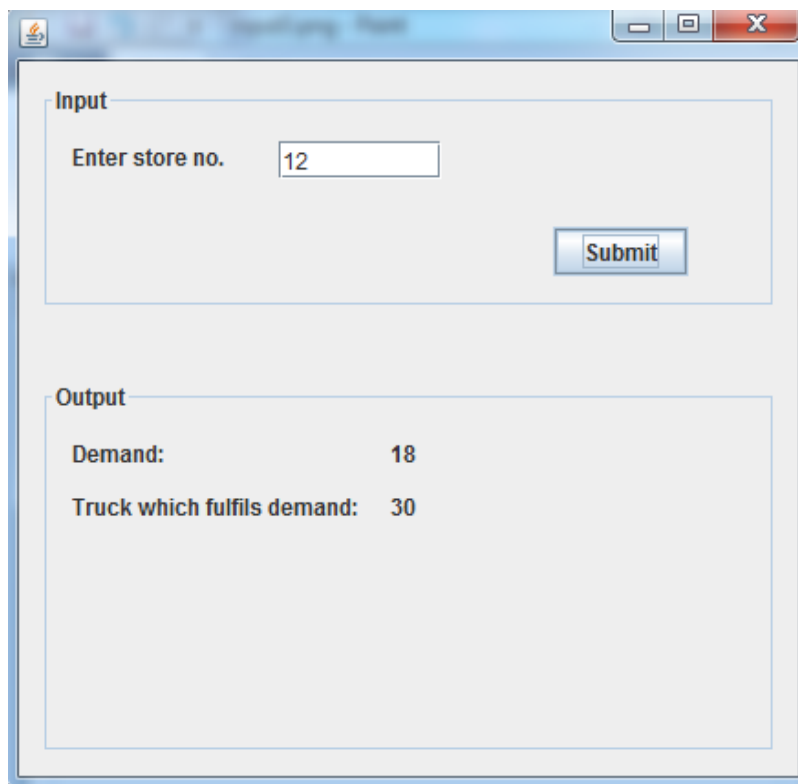
- This window shows the result for the truck number specified by the user. It provides the path of that truck (i.e. which all stores it visits) and the distance travelled by that truck and the cost incurred.



The screenshot shows a software window with a title bar. Inside, there are two main sections: 'Input' and 'Output'. The 'Input' section contains a label 'Enter truck no.' followed by a text box containing the number '5'. A 'Submit' button is located to the right of the text box. The 'Output' section displays the results of the calculation:

Truck Path:	0->517->521->524->519->518->512->514->516->509->511->513->510->505->507->503->506->508->504
Distance Travelled:	26.999999999999996
Cost:	809.9999999999999

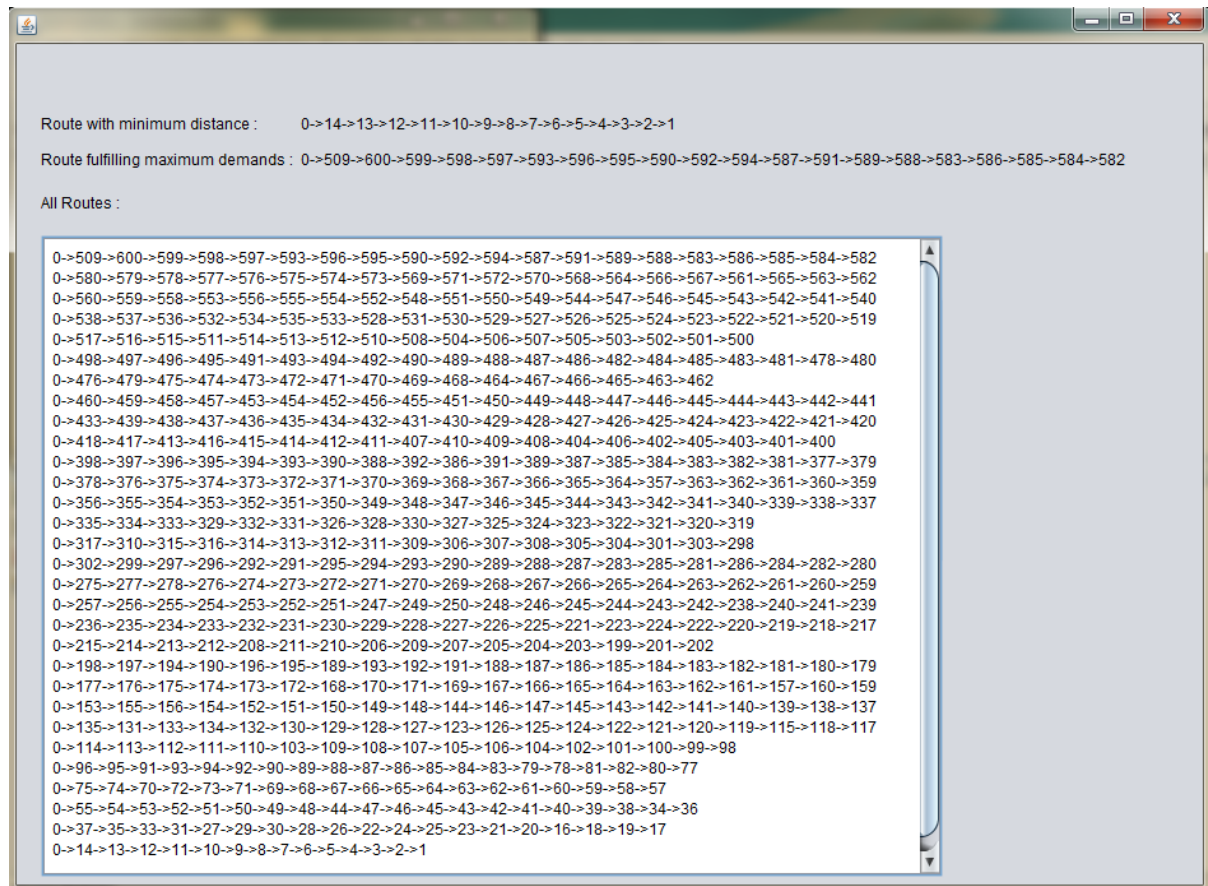
- This window shows the result for the store number specified by the user. It provides the demand of that store and which truck fulfils it



The screenshot shows a software window with a title bar. Inside, there are two main sections: 'Input' and 'Output'. The 'Input' section contains a label 'Enter store no.' followed by a text box containing the number '12'. A 'Submit' button is located to the right of the text box. The 'Output' section displays the results of the calculation:

Demand:	18
Truck which fulfils demand:	30

This window provides us the best available routes for the distribution of Grocery from the warehouse to various stores.



4.6 Report Generation

It is a segment of Output module which concludes the results. It includes the function `reportgeneration()` which generates the best 10 solutions providing –

- Maintenance cost
- Yearly cost
- No. of trucks to buy
- Total distance travelled

5. TESTING

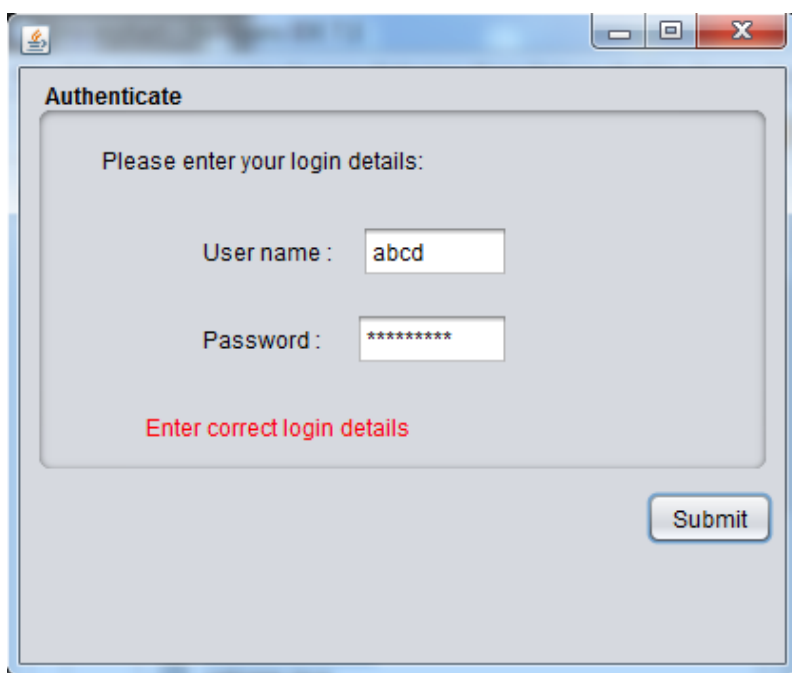
Once the source code has been developed testing is required to uncover the error before it is implemented. In order to perform software testing, a series of test cases is designed. Since testing is a complex process hence, in order to make the process simpler, testing activities are broken into similar activities. Due to this, for a project incremental testing is generally preferred. In this testing process the system is broken in set of subsystems and there subsystems are tested separately before integrating them to form the system for system testing.

Testing means the process of analysing a software item to detect the difference between existing and acquired conditions and to evaluate the feature of the software item.

BLACK BOX TESTING

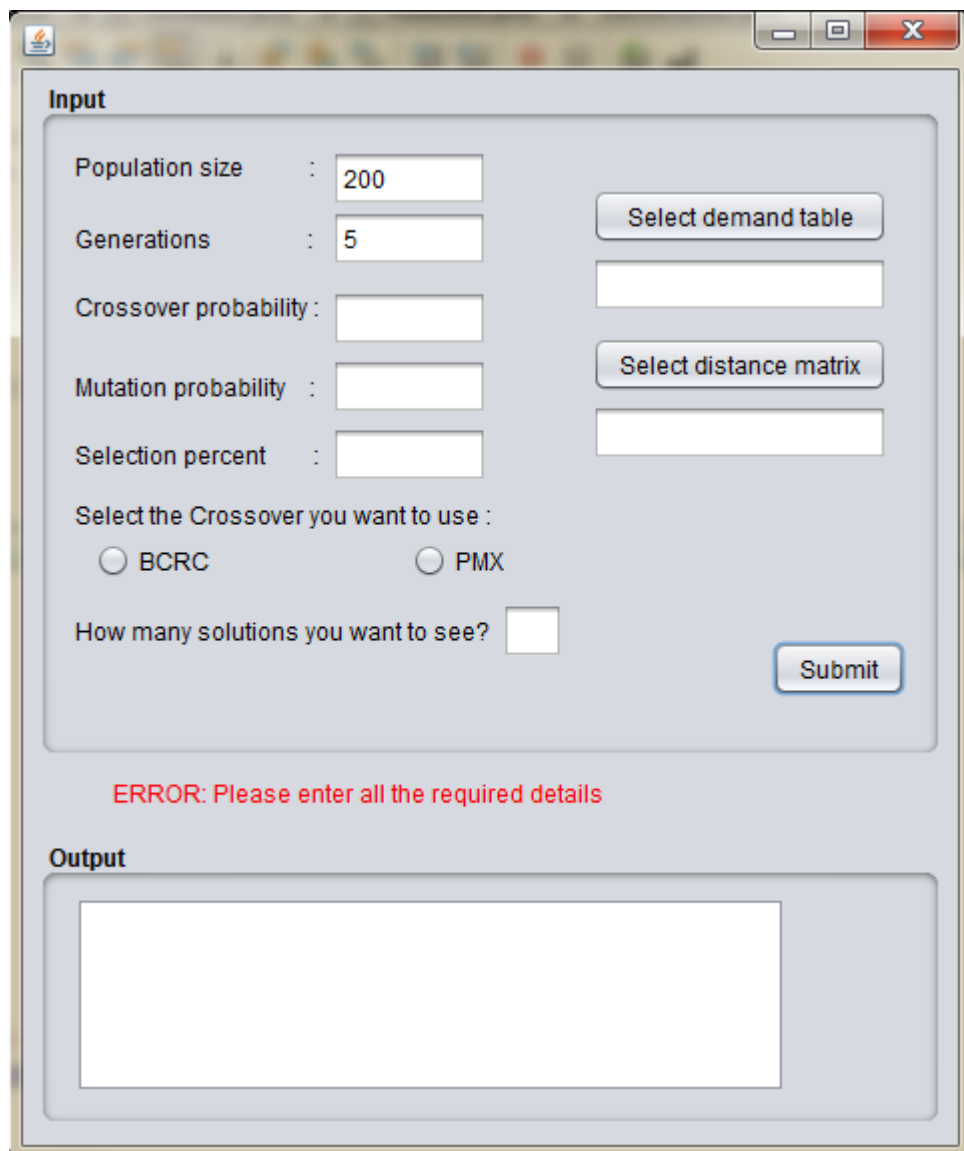
If the testing component is viewed as a “**black box**”, the inputs have to be given to observe the behaviour (output) of the program. In this case, it makes sense to give both valid and invalid inputs. The observed output is then matched with expected result (from specifications). The advantage of this approach is that the tester need not to worry about the internal structure of the program. However, it is impossible to find all errors using this approach. For instance, if one tried three equilateral triangle test cases for the triangle program, we cannot be sure whether the program will detect all equilateral triangles. This is an astronomical task – but still not exhaustive. To be sure of finding all possible errors, we not only test with valid inputs but all possible inputs.

- Generates error if the username and password field is entered wrong or kept empty.



The screenshot shows a web browser window displaying a login form. The form has a title bar with standard window controls. The main content area is titled "Authenticate" and contains the instruction "Please enter your login details:". There are two input fields: "User name" containing the text "abcd" and "Password" containing masked characters "*****". Below these fields, a red error message "Enter correct login details" is displayed. A "Submit" button is positioned at the bottom right of the form area.

- Generates error if any of the field in route evaluation window kept empty or filled wrong.



Input

Population size : 200

Generations : 5

Crossover probability :

Mutation probability :

Selection percent :

Select the Crossover you want to use :
☐ BCRC ☐ PMX

How many solutions you want to see?

Select demand table

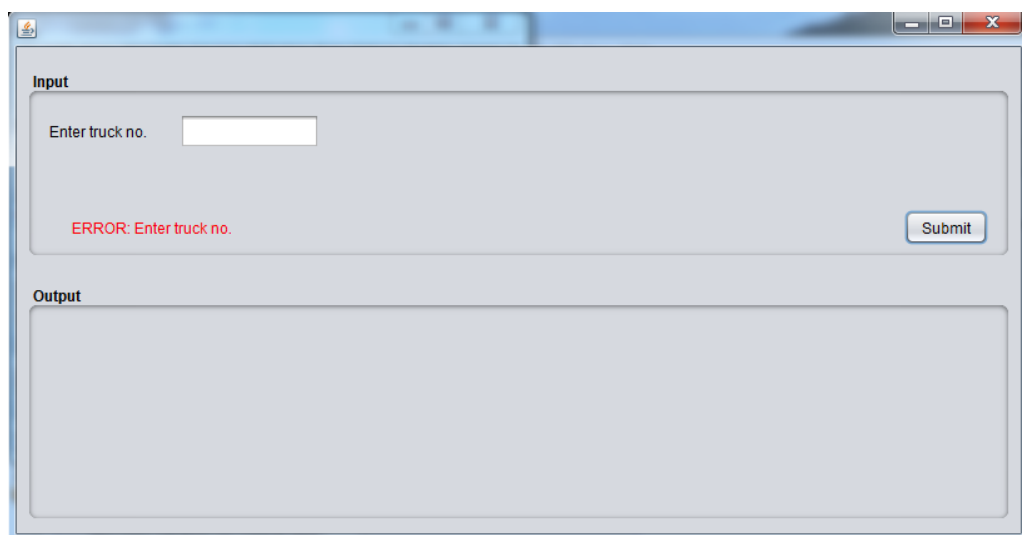
Select distance matrix

Submit

ERROR: Please enter all the required details

Output

- Generates error if truck number is entered wrong or kept empty.



Input

Enter truck no.

Submit

ERROR: Enter truck no.

Output

A screenshot of a Java Swing window titled "Input". The window has a standard Mac OS X title bar with red, yellow, and green buttons. Inside the window, there is a section labeled "Input" containing a text field with the value "350" and a "Submit" button. Below the text field, a red error message reads "ERROR: Enter the value in range". Below the "Input" section is an empty "Output" section.

- Generates error if store number is entered wrong or field is kept empty.

A screenshot of a Java Swing window titled "Input". The window has a standard Mac OS X title bar. Inside, the "Input" section contains a text field that is empty and a "Submit" button. A red error message below the text field reads "ERROR: Enter the store no.". Below the "Input" section is an empty "Output" section.

The image shows a Java Swing window with a title bar containing a Java logo, a maximize button, and a close button. The window is divided into two main sections: 'Input' and 'Output'.

Input Section:

- Label: "Enter store no."
- Text Field: Contains the value "800".
- Error Message: "ERROR: Enter the value in range" (displayed in red text).
- Submit Button: A button labeled "Submit".

Output Section:

- Label: "Output".
- Area: A large, empty rectangular box for displaying output.

6. CONCLUSIONS

Transportation Service System for Grocery kart is a multi objective optimization problem, which needed a suitable approach to obtain a global optimum solution. So, for this high complexity problem without any known sophisticated solution technique, a genetic approach was well suited. In the optimization process, due to the problem specific encoding, the genetic algorithm is able to compute and optimize the routing quite easily as compared to the other deterministic approaches. Genetic algorithm is a search heuristic that mimics the process of natural evolution. Genetic algorithms generate solutions to optimization problems using techniques inspired by natural evolution, such as selection, crossover and mutation.

The basic problem which we solved here consists of delivering goods to a depot. It consists of a set of stores with known demands which had to be fulfilled through minimum-cost vehicle routes originating and terminating at the depot. There were certain criteria that had to be taken care of, they were :-

1. All the deliveries are to be made between 5AM to 8AM.
2. Maximum number of stops are 20.
3. Travelling distance not more than 65 km.
4. Maximum load capacity is 1000 kg.
5. At the end of the trip, each truck returns back to the central warehouse.
6. Each stop at the store takes 5 min to download.
7. Every km travelled takes 90 sec on an average.

For this problem, initially we generated a Random Population to maintain diversity and a Greedy Population to get optimal solutions. Then for the **Selection** we used Tournament Rank selection which further optimised our dataset. Weighted Sum approach and Pareto Ranking (to get the alternative solutions of same nature) are the **Fitness Evaluation** procedure used.

We have used two types of **Crossover** here:-

- The BCRC (Best Cost Route Crossover), we used not only to maintain the candidates' good nature but improves it too to drive the force of obtaining a better solution by manipulating existing good candidates.
- Partially mapped crossover (PMX) aims at keeping as many positions from the parents as possible. To achieve this goal, a sub string is swapped like in two point crossover & the values are kept in all other non conflicting positions. The conflicting positions are replaced by the values which were swapped to the other offspring.

These two types of crossover were the basis of our analysis and for the comparison of our results .

• **Result Analysis**

PMX		BCRC	
Distance	Time Taken	Distance	Time Taken
• 818.1	1 min	799.8	20 min
• 819.4	40 sec	807.8	12 min
• 821.599	30 sec	812.1	2 min

From the results shown above which were compiled after the execution of the program we concluded that BCRC gives more optimum distance and takes more time for the execution and the population taken is less. Whereas for PMX gives less optimum distance and takes less time for the execution and the population taken is more.

From these conclusions we cannot judge the best out of the two but can be done in a particular aspect.

- BCRC is better than PMX in providing more optimum solution.
- PMX is better than BCRC as the execution time is less.

7. FUTURE SCOPE

Genetic Algorithms are of major significance to the development of solutions to multi objective and high complexity problems. The potential which they offer over existing techniques is enormous. They find application in biogenetics, computer science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields. And the list will continue to grow especially if Genetic Algorithms are combined with other optimization methods.

There is a great scope :

- For finding new crossover methods for more convergent steps.
- For finding new mutation methods for reducing divergence due to excess mutation.
- For finding a function which will decide suitable number of initial populations.

The method used to solve this Transportation Service System for Grocery Kart problem can be referred to solve :

- Location Allocation Problems
- Real Coded Genetic Algorithm (RCGA) for Integer Linear Programming in Production Transportation problems.
- Network Designing and routing problems.
- Planning of Packet Switched networks.

The further working on optimization can be obtained by :

- We can use roulette wheel selection and rank selection also as mentioned in (Ombuki et.al 2006) in references. The paper described that as compared to tournament selection these selection method were slow and gives nearly same answer.
- There are various other heuristic approaches mentioned in (Tan et.al 2000) LSD, Tabu search , Simulated Annealing and NSGA-II.
- We can also use PFIH approach for generating a seed for population generation (Tan et.al 2000).

- Insertion mutation is another mutation approach which gives the same result as inversion.
- The best population can be passed as seeds and can improve the running time of Best Cost Route Crossover.
- The best till date solutions can be saved and pass them as initiators to BCRC Crossover and it will further improve the result tremendously.
- There can be a new algorithm which can be invented and will run on updating the solutions only if any minimal changes occur in demand and distance matrix.
- Merging the two heuristic approaches like LSD and MOGA can also improve results but it will take a lot of running time.

8. ADDITIONAL MATERIAL

8.1 Definitions

- **Genetic Algorithm** : A Genetic Algorithm is a problem solving method that uses genetics as its model of problem solving. It is a search and optimization tool to find approximate solutions, which work differently from classical search method. All form of possible solutions corresponding to a given problem constitute the Search Space.

So the problem consists of finding out the solution that fits the best. But, when the search space becomes large, enumeration is no longer feasible as it will consume a large amount of time. So, a specific technique is used to find out the optimal solution. GA handles a population of possible solutions by applying a set of genetic operators directly on the population. These operators include:-

- 1) Selection
- 2) Crossover
- 3) Mutation

Selection (Reproduction) is used to compare each individual in the population. It is done by using a fitness function. The fitness corresponds to an evaluation of how good the candidate solution is. The optimal solution is the one, which maximizes or minimizes the fitness function according to the given problem. In the Fitness Evaluation step, each individual's quality is assessed. Then the Selection process helps to decide which individuals are to be used for reproduction and mutation in order to produce new search points. Various selection methods are:-

- a) Roulette Wheel Selection
- b) Random Selection
- c) Rank Selection
- d) Tournament Selection
- e) Boltzmann Selection
- f) Elitism

Tournament Selection: It is a method of selecting an individual from a population of individuals in a genetic algorithm. Tournament selection involves running several "tournaments" among a few individual chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. Selection pressure is easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected.

Crossover(Recombination) is the process of taking two parent solutions and producing from them a child (another solution). After the Selection process, the population is enriched with the better individuals. Then, this operator is applied on the mating pool(search space) to produce a better offspring. It proceeds in 3 steps:-

- i. Reproduction operator selects at random a pair of two individual strings for mating.
- ii. A cross site is selected at random along the string length.
- iii. Finally, the position values are swapped between the two strings.

The various crossover techniques are:

- a) Single point crossover
- b) Two point crossover
- c) Multi-point crossover
- d) Uniform Crossover

Mutation is used to maintain genetic diversity in the population. It introduces new genetic structures in the population by randomly modifying some of its building blocks. It plays the role of recovering the lost genetic materials as well as for randomly distributing the genetic information. Mutation can be generally done by:

- a) Flipping
- b) Interchanging
- c) Reversing

The basic Genetic Algorithm is as follows :

Step 1: [start] Selecting Genetic random population of 'n' chromosomes
(suitable solutions for the problem)

Step 2: [Fitness] Evaluate the fitness $f(x)$ of each chromosome x in the population

Step 3: [New population] Create a new population by repeating following steps until the

new population is complete

[selection] select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to get selected).

[crossover] With a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is

the

exact copy of parents.

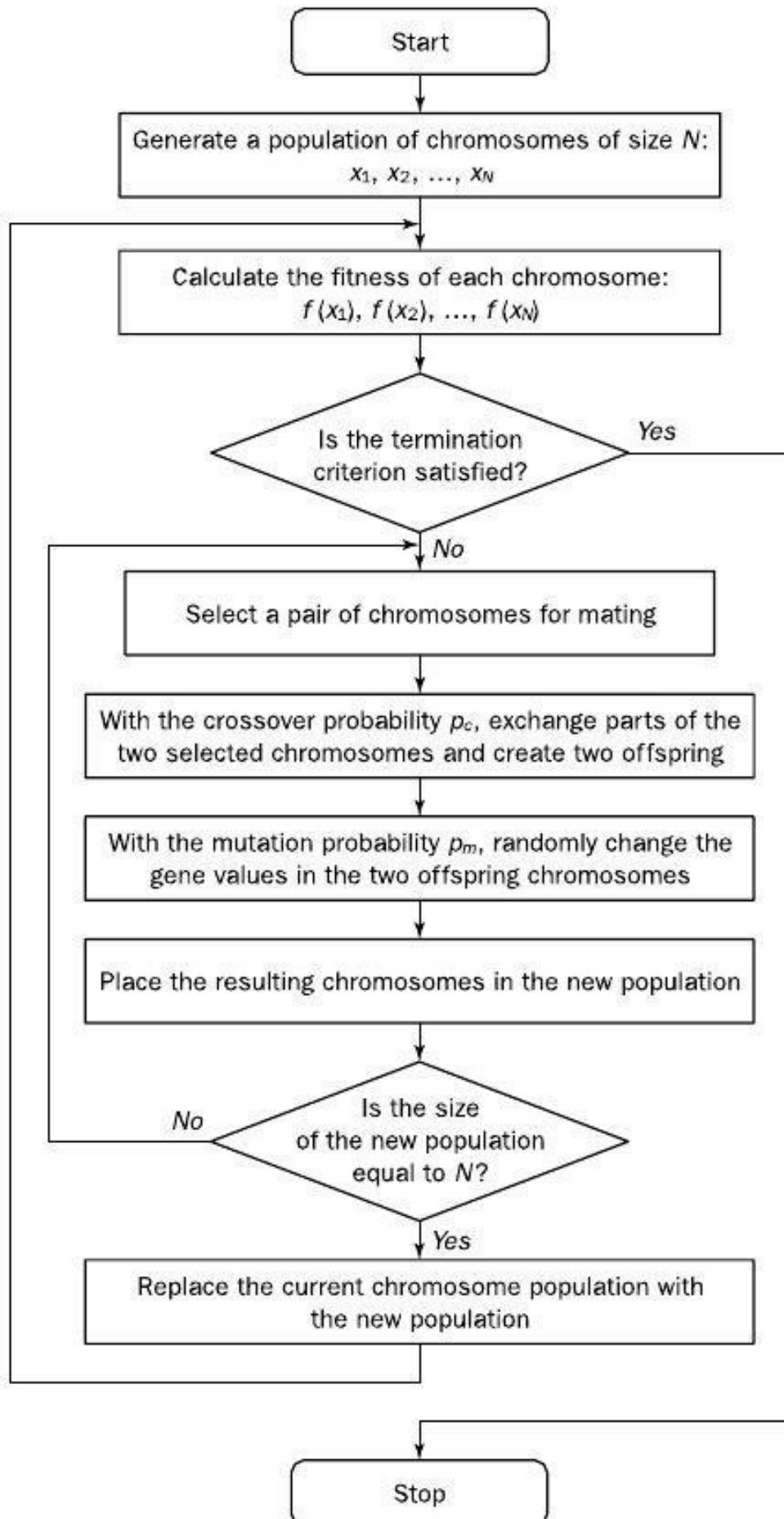
[Mutation] With a mutation probability, mutate new offspring at each locus (position in chromosome)

[Accepting] Place new offspring in the new population.

Step 4: [Replace] Use new generated population for a further sum of the algorithm.

Step 5: [Test] If the end condition is satisfied, stop, and return the best solution in Current population.

Step 6: [Loop] Go to step2 for fitness evaluation.



- **VRP** : In the general Transportation problem, the objective is to minimize the total transportation cost. But this assumption is not always valid. As in the transportation problem there can be multiple objectives such as the fulfillment of transportation schedule contracts, balancing of load among the given number of vehicles, delivering the required quantity in every plant, minimization of transportation hazards and minimizing the cost. Generally, a multiobjective linear program is written as:

$$\text{Max } z_1(x) = c^1(x)$$

$$\text{Max } z_2(x) = c^2(x)$$

$$\text{Max } z_3(x) = c^3(x)$$

•

•

•

$$\text{Max } z_q(x) = c^q(x)$$

$$\text{s.t. } x \in S$$

Where q is the number of objectives, c^i is the vector of the i^{th} objective function coefficients (gradient), S the feasible region, and $z_i(x)$ the i^{th} objective function value (criterion value). Since multiple objective problems rarely have points that maximize all of the objectives simultaneously, we try to maximize each objective to the greatest extent possible.

9. REFERENCES

- K.C.Tan, L.H.Lee, Q.L.Zhu, K.OU,2000. Heuristic methods for vehicle routing problem with time windows, Artificial Intelligence in Engineering, 1-15
- Mitsuo Gen and Runwei Cheng,2000. Genetic Algorithms and Engineering Optimization, Wiley Series, 340-400
- Randy L.Haupt ,2004. Practical Genetic Algorithms -2nd Edition , Wiley Series, 200-250
- Beatrice Ombuki,B.J.Ross and Franklin Hanshar,2006. Multi Objective Vehicle Routing Problem with Time Windows, Applied Intelligence, 1-14
- S.M.SIVANANDAM, S.N.DEEPA,2008. Introduction to Genetic Algorithms, Springer, 170-200
- Mitchell Melanie,1999. An introduction to Genetic Algorithm, MIT Press, 160-165
- <http://www.mu-sigma.com/muphoria/>
- http://en.wikipedia.org/wiki/Genetic_algorithm