Solving Multiple Objective Optimization Problem using Multi-Agent Systems: A case in Logistics Management

Venkata Sai Teja Pennada
This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
Venkata Sai Teja Pennada
E-mail: peve17@student.bth.se

University advisor:
Lawrence Edward Henesey
Department of Computer Science

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden
Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
Abstract

Background: Multiple Objective Optimization problems (MOOPs) are common and evident in every field. Container port terminals are one of the fields in which MOOP occurs. In this research, we have taken a case in logistics management and modelled Multi-agent systems to solve the MOOP using Non-dominated Sorting Genetic Algorithm-II (NSGA-II).

Objectives: The purpose of this study is to build AI-based models for solving a Multiple Objective Optimization Problem occurred in port terminals. At first, we develop a port agent with an objective function of maximizing throughput and a customer agent with an objective function of maximizing business profit. Then, we solve the problem using the single-objective optimization model and multi-objective optimization model. We then compare the results of both models to assess their performance.

Methods: A literature review is conducted to choose the best algorithm among the existing algorithms, which were used previously in solving other Multiple Objective Optimization problems. An experiment is conducted to know how well the models performed to solve the problem so that all the participants are benefited simultaneously.

Results: The results show that all three participants that are port, customer one and customer two have gained profits by solving the problem in multi-objective optimization model. Whereas in a single-objective optimization model, a single participant has achieved earnings at a time, leaving the rest of the participants either in loss or with minimal profits.

Conclusion: We can conclude that multi-objective optimization model has performed better than the single-objective optimization model because of the impartial results among the participants.

Keywords: Multiple Objective Optimization Problem, Non-dominated Sorting Genetic Algorithm-II, Multi-agent systems, Multi-objective optimization model, Single-objective optimization model.
Acknowledgments

Foremost, I wish to express my deepest gratitude to my supervisor, Professor Lawrence Edward Henesey, who has the substance of a genius. It is whole-heartedly appreciated that his great advice for my study proved monumental towards the success of this study. His support, encouragement and patience make him a great mentor and his helpfulness makes him a great human being.

I am extremely grateful to my parents for their love, support and sacrifices for educating and preparing me for my future. I also thank all friends, colleagues working with Prof. Lawrence Henesey and my dear classmates.
Contents

Abstract 1
Acknowledgments 2

1 Introduction 9
1.1 Aim and Objectives: 10
1.2 Research Questions: 11
1.3 Problem Description: 11
1.4 Outline: 12

2 Background 13
2.1 Application of Computer Science Knowledge in Logistics 13
2.2 Types of Optimization 13
2.3 Linear Integer Programming (LIP) 14
2.4 Artificial Intelligence 14
2.4.1 Agents and Multi-Agent Systems (MAS): 14
2.5 Genetic Algorithms 16
2.6 Architecture of NSGA-II 18

3 Literature Review 22
3.1 Genetic evolutionary algorithms for solving MOOPs: 23
3.1.1 Strength Pareto Evolutionary Algorithm (SPEA): 23
3.1.2 Pareto Archived Evolutionary Strategy (PAES): 23
3.1.3 Non-dominated Sorting Genetic Algorithm (NSGA): 23
3.1.4 NSGA-II 24
3.2 Performance of NSGA-II over other algorithms: 25

4 Experiment 26
4.1 Environment 26
4.2 Data Collection 26
4.3 Data Labelling 27
4.4 Implementation 27
4.5 Performance Measures: 31

5 Results 32
6 Analysis and Discussion 36
   6.1 Analysis of Literature review ........................................... 36
   6.2 Analysis of Experiment .................................................. 36
   6.3 Discussion ........................................................................ 36
   6.4 Validity Threats ................................................................ 37
       6.4.1 Internal Validity ....................................................... 37
       6.4.2 External Validity ....................................................... 37
       6.4.3 Conclusion Validity .................................................... 38

7 Conclusion and Future Work 39
   7.1 Future Work ...................................................................... 39
List of Figures

1.1 Operational areas inside a port .......................... 10
2.1 An agent interacting with environment ..................... 15
2.2 Representation of Crossover and Mutation .................. 17
2.3 Labelling of individual units in Genetic Algorithm ........... 17
2.4 Flow chart of genetic algorithms ............................. 18
2.5 NSGA-II procedure .............................................. 19
2.6 Differentiation of fronts based on calculations ............... 20
2.7 Crowding Distance Sorting .................................... 21
3.1 The different front ranks in NSGA .......................... 24
3.2 Difference between NSGA-II and PAES in spread of solutions .... 25
4.1 Container arrangement in ports ............................... 27
4.2 Sequence of steps involved in the experiment ................ 30
5.1 Case(i): Optimization of Carrefour Profits .................. 32
5.2 Case(ii): Optimization of Metro Profits ..................... 33
5.3 Case(iii): Optimization of sum of Contract Price ............. 34
5.4 Case(iv): Multi-Objective Optimization ..................... 35
List of Tables

4.1 Configuration of hardware and software. .............................. 26

5.1 Port’s income and customers profit when containers are picked in favour of Carrefour. .................................................. 33
5.2 Port’s income and customers profit when containers are picked in favour of Metro. ..................................................... 33
5.3 Port’s income and customers profit when containers are picked in favour of Port. ......................................................... 34
5.4 Port’s income and customer’s profit while using Multiple objective optimization model. .................................................. 35
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>DAI</td>
<td>Distributed Artificial Intelligence</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GLPK</td>
<td>GNU Linear Programming Kit</td>
</tr>
<tr>
<td>IP</td>
<td>Integer Programming</td>
</tr>
<tr>
<td>LIP</td>
<td>Linear Integer Programming</td>
</tr>
<tr>
<td>MAS</td>
<td>Multi-Agent System</td>
</tr>
<tr>
<td>MOEA</td>
<td>Multiple Objective Evolutionary Algorithm</td>
</tr>
<tr>
<td>MOGA</td>
<td>Multiple Objective Genetic Algorithm</td>
</tr>
<tr>
<td>MOOP</td>
<td>Multiple Objective Optimization Problem</td>
</tr>
<tr>
<td>NSGA</td>
<td>Non-dominated Sorting Genetic Algorithm</td>
</tr>
<tr>
<td>PAES</td>
<td>Pareto Archived Evolutionary Strategy</td>
</tr>
<tr>
<td>SPEA</td>
<td>Strength Pareto Evolutionary Algorithm</td>
</tr>
</tbody>
</table>
Glossary

**Chromosome** A set of parameters which define a proposed solution to the problem. 16

**Crossover** The process of generating a new offspring by combining genetic information of two parents. 16

**Gene** A unit of heredity which holds some characteristics of an offspring. 16

**Offspring** A result of something. 16

**Optimization** The action of making effective use of a resource. 9

**Terminal** A place where loading and unloading of goods take place. 9

**Throughput** The amount of items passing through a process. 9
Terminals are essential in global trade. In the current decade, there is a considerable increase in container transportation worldwide. High investments, as well as increased operating costs of ships and port equipment, forces improvement in terminal operations[1]. To minimize their operating costs, container terminal operators need to transport the containers to and from the trucks efficiently. High productivity and container Throughput with low prices are essential aspects for a terminal operator to compete with other terminals. An increase in container transportation resulted in terminals handling more containers in a short time at low cost. Therefore, existing infrastructure and equipment need to be efficiently used to gain profits. Optimizing container management is a crucial challenge in the domain of supply chain logistics[2]. Decisions taken by the container terminal management need to consider customer service requirements, such as Performance, Reliability, Cost, Quality and Adaptability[3]. To achieve them, port managers have to maximize their throughput by making the least number of container moves possible. In this context, a move represents moving a container from the stack or delivering it to the customer. Usually, a port can deliver limited number of containers per day, and efficiently use of the moves will gain profits. Customers have their objective of preparing a container pick-list that can yield them higher profits. Here we are dealing with two objectives simultaneously to produce an optimal solution. These kinds of problems are called Multiple Objective Optimization Problems (MOOPs).

In real-world, companies in every sector deals with multiple objectives simultaneously[4]. Problems in which numerous objectives are dealt with simultaneously to achieve an optimal solution are called Multiple Objective Optimization Problems (MOOPs)[5]. An acceptable explanation to multiple objective problems is to derive a set of solutions, where each solution satisfies the objectives at an adequate level without getting dominated by any other solution. The objectives are generally conflicting, such as maximize performance, minimize cost, and a single extreme solution will not satisfy both the objective functions. A set of optimal solutions are required to eliminate trade-offs between different objectives[6]. This set of non-dominated solutions is called a Pareto front. Figure 1.1 explains typical operational areas present in a port.
1.1 Aim and Objectives:

This research aims to solve a Multiple Objective Optimization Problem (MOOP) in the field of logistics management by developing AI-based models and compare their performance.

Objectives:

- To develop a port agent with an objective function of maximizing the total contract prices of containers delivered in a day which means increasing throughput.
- To develop a customer agent with an objective function of maximizing the total business profit.
- To develop a single-objective optimization model and derive results for the MOOP.
- To develop a multi-objective optimization model by applying a chosen algorithm from the literature review conducted and derive results for the MOOP.

An experiment is conducted to achieve the objectives. A model-based reflex agent is developed to fulfil some of the objectives and used an algorithm called Non-dominated sorting genetic algorithm-II to find the Pareto front.
1.2 Research Questions:

**RQ1.** What could be the suitable algorithm for solving Multiple Objective Optimization Problem (MOOP)?

**Motivation:** The motivation of this research question is to choose a suitable and successful algorithm among existing MOOP solving algorithms.

**RQ2.** How will the participants get affected when the problem is solved using a single-objective optimization model rather than multiple objective optimization model?

**Motivation:** The motivation of this research question is to find a better performing model between them in solving the MOOP.

1.3 Problem Description:

Most simple decision processes often deal with a single objective such as minimizing cost or maximizing profit. The chances of a company dealing with a single objective are limited only to the theory. In practical, decision-makers have to make a decision in an environment where more than one objective are dealt with simultaneously, constrain the problem and the relative value of each of these objectives is different. Such problems where more than one objective is optimized simultaneously are called Multiple Objective Optimization Problems (MOOP). Optimization deals with the finding of the best solution(s) over a set of possible choices. Multiple Objective Optimization problems arise in various fields such as industrial production, capital budgeting and logistics management. In this paper, we will discuss the Multiple Objective Optimization Problem in logistics management.

The MOOP considered in this thesis is container management in ports, a case in logistics management. A stack consisting of 400 containers arranged in an area, where the number of containers should not exceed 10, 10, 5 concerning x, y, z axes respectively. Every port can deliver a limited number of containers per day, and container movements need to be efficient. Here, a single move is defined as either moving a container within the stack to another coordiate or removing a container from the stack and delivering it to the customers. In a typical day, the crane operating on this stack can move 100 containers. Both the customers and port have their objectives. The port’s objective is to maximize its throughput by making the least moves possible, and the customer’s objective is to pick the containers that can maximize their business profit. As two objectives are optimized simultaneously to find a set of optimal solutions, we consider this problem as Multiple Objective Optimization Problem (MOOP).
1.4 Outline:

The structure of the thesis work is discussed in this section.

**Chapter 1:** In this chapter, the Introduction and motivation of the thesis are explained along with the aim and objectives, problem statement and research questions.

**Chapter 2:** This chapter explains the concepts related to the thesis.

**Chapter 3:** This chapter provides the outcomes of the literature review.

**Chapter 4:** In this chapter, dataset and performance metrics used in this thesis are explained.

**Chapter 5:** In this chapter, results from the experiment are presented.

**Chapter 6:** In this chapter, analysis and discussion is made regarding the obtained results.

**Chapter 7:** In this chapter, conclusion and future work is explained.
A detailed description of the terms and concepts related to the thesis is provided in this section.

The problem considered in this research is a case in logistics management, and it is an optimization problem with multiple objectives that need to be dealt with simultaneously. The problem is modelled as a Linear Integer Programming (LIP) problem because all the variables involved are integers. Individual model-based reflex agents are developed for the port as well as customers and to find the Pareto front genetic algorithm called Non-dominated sorting genetic algorithm-II is used.

2.1 Application of Computer Science Knowledge in Logistics

The knowledge of computer science has been widely used to improve logistics and transportation. There were many studies in which agent-based approaches were used to solve the problems in transport logistics[7]. We can divide the logistics problems into three domains, namely transport, traffic and terminal. Problems such as route planning and scheduling fall into the transport domain[8], problems such as railway slot management, air traffic management and railway traffic management fall into the traffic domain[9]. Terminal domain handles the problems such as resource allocation, scheduling of cranes and forklifts[10]. Agent-based systems were developed for solving the scheduling problem in truck delivery, coordinating and controlling of air traffic, plan and coordinate the processes within the terminal by mapping the objects and resources within the terminal.

2.2 Types of Optimization

An optimization problem is defined as, "Finding the best solution from a set of solutions where every solution in the set satisfies problem constraints." Typically, there are two types of optimization, (i) Single-Objective optimization (ii) Multi-Objective optimization [5][4].

(i) Single-Objective Optimization: The model in which only one single objective is handled at once and providing a solution concerning that single objective is called Single-objective optimization.[4][11].
(ii) **Multi-Objective Optimization**: The model in which more than one objective is handled at once and providing a solution set in which every solution accepts every objective at a considerable level is called Multi-objective optimization[11][4].

This type of optimization is also called Pareto Optimization, Vector Optimization and Multi-criteria optimization. The word Pareto optimization is titled after a famous economist named Vilfredo Pareto, who made a significant contribution in solving MOOP[12].

### 2.3 Linear Integer Programming (LIP)

An Integer Program (IP) is a mathematical optimization program in which some part or all the variables belong to integers[13]. If plotting a graph for objective functions and constraints results in a straight line residing in a plane, then it is called as Linear Integer Programming (LIP)[14]. We represent a Linear Integer Programming Optimization Problem as,

Maximize $cx$,

Subject to: $Ax \leq b$,

$x \in \mathbb{Z}^n$.

Where solution $x \in \mathbb{Z}^n$ is a vector of n integer variables, "A" is analytical data stored in matrix $m \times n$, "c" is matrix $1 \times n$, "b" is matrix $m \times 1$. "m" and "n" represents rows and columns in a matrix.

### 2.4 Artificial Intelligence

Artificial Intelligence (AI) is a branch of knowledge of science, with tremendous achievements accomplished in decades. Formally AI is initiated in the year 1956 during a conference in Dartmouth, New Hampshire. The primary goal of AI is to replace humans with software or machines[15]. From the mid-late 1970s Distributed Artificial Intelligence (DAI) has started evolving, which lead to the introduction of agent based technology in 1990s for solving problems in a distributed and autonomous manner[16].

#### 2.4.1 Agents and Multi-Agent Systems (MAS):

The term "Agent" has many definitions which were accepted by the community of Artificial Intelligence (AI)[17]. Simply an agent can be defined as "A physical or Virtual entity that can act, perceive its environments and communicate with others, is autonomous and has skills to achieve goals and objectives"[15]. According to Wooldridge and Jennings, defined an agent as a computational system interacting with the environment that comprises of features like Independence, Social ability, Re-activeness and pro-activeness. These features are explained below: Figure 2.1 explains different inputs and outputs of an agent while interacting with an environment.
- **Independence**
  Independence is defined as the ability to act and react by itself without any external support provided by a human being or other devices[18].

- **Social ability**
  Social ability is defined as the ability to communicate with other agents/devices by using defined protocols in-order to fulfill the objectives[18].

- **Re-activeness**
  Re-activeness is defined as the feedback given by an agent to the signals received from environment[18].

- **Pro-activeness**
  Pro-activeness is defined as an initiative taken by the agent in order drive in the path where it can accomplish its assigned goals[18].

Agents are classified into five categories based on their interaction with the environment and working process. They are:

(i) **Simple-reflex agents**
An agent is said to be Simple-reflex agent if it does not consider the percept history and only behaves based on the current percept. These agents work on condition action rule. Condition-action rule is a set of protocols given to the agent, and it acts if the condition is true and stays ideal if it is false. These agents tend to have limited intelligence and changing in the rules is mandatory if at all modifications are made in the environment.
(ii) Model-based reflex agents
Model-based agents have partial interaction with the environment, unlike Simple-reflex agents. These agents maintain an internal state to keep track of movements of the environment it is exposed. The internal state depends on the history of its observations, so it stores unobserved aspects of the current state. We provide information about two things to update the state. They are:
(a) Information about how the world evolves on its own.
(b) Affects reflected on the world due to the agent’s action.

(iii) Goal-based agents
These agents are driven based on the goal. A description of desirable situations is called a goal. These agents drive towards the goal, and it intends to reach the goal in the shortest path. These agents are flexible because of its adaptability to the change in knowledge that supports their decisions.

(iv) Utility-based agents
We choose these agents when we have to decide between multiple possibilities. We act on each state based on the preference. It is not only essential to reach the goal safer, and in the cheaper path, but we should also consider the agent’s happiness. The term "Utility" describes how happy the agent is. These agents choose the actions which maximize their utility.

(v) Learning agents
As the name of this agent is self-explanatory, these agents can learn from their previous experiences. Even though the agent starts with essential knowledge, these agents gain knowledge by progressing forward and learn in its course.

Multi-Agent System (MAS) is defined as "a set of agents which comprises of two or more agents which interact with each other through the definition of appropriate rules in order to fulfil a task in a given environment"[15]. Task-oriented coordination can be both cooperative and competitive. These systems may be homogeneous or heterogeneous which work on the same or different set of goals[17].

2.5 Genetic Algorithms
Genetic algorithm (GA) is a meta-heuristic search technique based on the Darwinian concept[4]. Darwin’s concept states that the species which can fit in the environment will be survived and proceed to reproduce, eliminating the rest. There are two crucial steps involved in GA, namely, Crossover and Mutation. Crossover is also called as recombination. It is a process of generating a new Offspring by combining genetic information of two parents[19]. It is a way of randomly generating new solutions from the existing population. The process of altering 22 Gene values in the Chromosomes from the initial set in order to maintain genetic diversity from one generation of the population to the next generation is called mutation [20][14]. Figure 2.2 is a pictorial representation of crossover and mutation steps in a genetic algorithm.
Figure 2.2: Representation of Crossover and Mutation

Figure 2.3: Labelling of individual units in Genetic Algorithm
2.6 Architecture of NSGA-II

NSGA-II uses Darwin’s survival of the fittest theory. The performance of the output of the system is called as fitness value, and the input parameters are called genes[21]. The initial population (Pt) contains several individuals which differ in genes or parameters, and we calculate the target parameter for each of them. An offspring population (Qt) is created by mixing the genes of better performing individuals and cause mutations. The new population (Rt) is twice the population of the initial population (Pt). Every individual is ranked based on the performance indicators. Better the performance, better an individual, is ranked. Individuals are divided into different fronts and smaller the number of front, better the performance. A final population (Pt+1) is created by picking individuals from best fronts until we reach the size of the original population (Pt). If the last front does not fit entirely into the final population (Pt+1), crowding distance sorting is used to pick the best individuals until the final population (Pt+1) reaches the size of the original population (Pt)[22]. This process repeats again and again, according to the demand, until we reach the target limit. The working of NSGA-II consists of two crucial steps, (i) Non-dominated sorting (ii) Crowding distance sorting[23].
Figure 2.5 is a pictorial representation of the process involved in NSGA-II

**Figure 2.5: NSGA-II procedure**

where $f_1$, $f_2$, $f_3$ are front one, front two, front three.

**i) Non-dominated sorting**
This is a technique used to find which individual belongs to which front. If one individual is said to dominate other, it means the value of the dominating individual is better or equal to the value of other individuals, and for one of those target indicators, the value of the individual is better than the value of the other individual[24].

In representation,

$$A(x_1|y_1)\text{dominates}B(x_2|y_2)\text{when} (x_1 \leq x_2 \text{ and } y_1 \leq y_2) \text{ and } (x_1 < x_2 \text{ or } y_1 < y_2)$$

(2.1)

where,

$x$ and $y$ are points on X,Y coordinates respectively.

We compare every individual in the new population (Rt) with the other individuals, and it is pairwise checked if the other individual dominates the individual or it dominates the other[21]. In parallel, it also calculates the number of individuals dominates each individual and creates a list which contains all individuals that are dominated by that specific individual[25].

All the individuals who have the domination count of "0" belongs to front one ($f_1$), which means any of the individuals does not dominate these points. Iterating through all the remaining individuals, being in one of these (front one individuals) lists we make subtraction of -1 for each dominating count of these individuals[24]. This way, we can find the front two ($f_2$) and front three ($f_3$).
Figure 2.6 is a representation of division of points into different fronts after performing the above calculations.

(ii) Crowding distance sorting
This is a technique used to decide for individuals residing in the same front. To calculate crowding distance for an individual, we take distance of smaller and next more significant values from that point and calculate the delta of those and divide them by the delta of the maximum value and minimum value for the target indicator[21]. These points should belong to the same front, where the point is located. We add this calculation to the calculated distance of the other objectives[25].

In representation,

\[
distance(i) = distance(i) + \frac{o(i + 1) - o(i - 1)}{o(max) - o(min)}
\]  

(2.2)

After calculating the crowding distance for every point, individuals with high crowding distance are preferred first.
Figure 2.7 is an example representing the plotting of points and names given to different points used for calculating Crowded Distance Sorting.
A literature review is conducted to investigate the answer for RQ1. What could be a suitable algorithm for solving Multiple Objective Optimization Problem (MOOP)? This review aims to find a suitable and successful algorithm among other algorithms.

Since the past decade, genetic algorithms (GA) proved to be successful in solving Multiple Objective Optimization Problems in finance subject to different constraints. GAs caught the attention of researchers in solving MOOP because of their adaptability and robustness[19]. There may be a change in the sector, but the application is common for MOOP in the logistics sector or other resource allocation problems[5]. Multiple Objective Genetic Algorithms (MOGA) deal with a set of possible solutions. These set of possible solutions are typically called a population in genetic terminology. The advantage of using GA is, we will find many Pareto optimal solutions in a single run unlike other classes of algorithms in which we make a series of separate runs. An additional advantage of MOGA is that they give good results even for discontinuous and non-convex Pareto fronts[26][27]. These attractive features of genetic algorithms made researches use MOGA for solving complicated Multiple Objective Optimization Problems. There are several GA capable of solving MOOP, and as the time progresses, these GA getting better and better. Some of the successful GAs in solving MOOP are discussed in the following section.
3.1 Genetic evolutionary algorithms for solving MOOPs:

From time to time, many genetic evolutionary algorithms are coming into existence, and every algorithm has its strength when compared to others. This section explains some of the widely used genetic algorithms for solving MOOPs.

3.1.1 Strength Pareto Evolutionary Algorithm (SPEA):

SPEA is a GA proposed by Zitzler and Thiele for solving Multi-objective optimization problems. This algorithm works on the concept of non-domination. This algorithm keeps track of all the non-dominated solutions after every generation by maintaining an external population[28]. This external population involves in all genetic operations. We combined the external population with the current population at every generation. We assign a fitness value to all the non-dominated solutions in the combined population. Based on the number of solutions they dominate, the fitness value is assigned. All the dominated solutions will be assigned less fitness than the non-dominated solutions with the worst fitness[21]. By doing so, we can ensure that the final set contains only non-dominated solutions.

3.1.2 Pareto Archived Evolutionary Strategy (PAES):

PAES is a simple genetic algorithm proposed by Knowles and Corne. This GA uses a single-parent, single-offspring strategy, which is similar to (1+1)-evolution strategy. Bit-wise mutations are carried out by using binary strings to create offsprings[28]. We make a comparison between the parent and offspring, and if the offspring dominates parent, then it will be carried forward and becomes the parent in the next iteration. This process continues until we find a set of best possible solutions. If the parent dominates the offspring, then the offspring is said to be rejected, and a new offspring is found. There could be a situation where neither offspring nor parent dominates each other. In such cases, either parent or the offspring is chosen by comparing with the set of solutions found so far. A comparison is made between the offspring and the set of solutions. If it dominates any of the solutions in the solution set, it is accepted as the next parent. If the offspring failed to dominate the solutions in the solution set, a calculation for the closeness with the solutions is made for both parent and offspring. If the offspring is surrounded by less number of solutions, it is accepted as the next parent[29].

3.1.3 Non-dominated Sorting Genetic Algorithm (NSGA):

NSGA is a successful genetic algorithm proposed by Srinivas and Deb. This algorithm uses a sorting and fitness assignment procedure. Goldberg’s Pareto ranking method is used to rank the population[28]. We require an initial solution population in order to run a genetic algorithm. This initial solution population is plotted as a Pareto front. Large dummy fitness values are assigned to all non-dominated individuals. The individuals are shared with their dummy fitness values; to achieve diversity[30]. A front is formed if no other solutions dominate the points residing in that front.
algorithm now ignores the points which lie in front one, and a new front is composed in a similar manner. We continue this process until every individual in the population belongs to any of the fronts. The algorithm favours the front, which lies closer to the Pareto front. A new generation is produced by using stochastic remainder proportion. Figure 3.1 represents how points are distributed among different fronts.

![Figure 3.1: The different front ranks in NSGA](image)

### 3.1.4 NSGA-II

This algorithm is an extension to NSGA proposed by K. Deb. NSGA-II follows the same process of dividing the entire population into different fronts[28]. A crowded comparison approach is used in NSGA-II instead of sharing function. A promotion of solution is made based on the rank, and if there is a tie, crowding distance factor is used[29]. This approach is useful when there is a situation of eliminating some solutions from the front in order to fit the final population. This is done because the final population should always be the same size as the initial population. In order to apply crowding-distance sorting to the population, the population need to be sorted in ascending order of magnitude according to each objective function value[21]. The smallest and largest function values, i.e. the extreme values are assigned an infinite distance value. We assign a value equal to the absolute normalized difference in the function values of two adjacent solutions to all the intermediate solutions. We continue the same process with all the objective functions. The final crowding-distance value is defined as the sum of individual distance values concerning each objective. Before calculating the crowding distance, every objective function is normalized[25].
3.2 Performance of NSGA-II over other algorithms:

Based on the results obtained by earlier researchers, NSGA-II outperformed other algorithms in most of the instances. As far as convergence and divergence metrics are considered, NSGA-II leads in terms of performance leaving SPEA and PAES behind[28]. Convergence is defined as the closeness of a solution to the known set of Pareto optimal solutions, whereas divergence is defined as the area covered among the obtained solutions. The distribution of non-dominated individuals can be measured using a uniform distribution metric. [31].

On assessing the results for the experiments conducted by earlier researchers, PAES and SPEA require less computational time than NSGA-II. Even though PAES and SPEA perform best in terms of computational time, NSGA-II dominates these algorithms in terms of convergence. When a medium-size instance is considered, NSGA-II and SPEA scores almost equal and dominance cannot be noticed over one another. However, when large size instances are considered, NSGA-II outperforms SPEA[21]. The researchers experimented with 30 instances, and NSGA-II performs better in 93% problem instances, whereas SPEA performs better in only 7% instances[21]. These results show a clear dominance of NSGA-II on large-size instances, which are encountered in practical scenarios. Even though NSGA-II and SPEA perform equally in the experiment conducted using medium-size instances, researchers suggested NSGA-II over SPEA because of the less requirement of computational resources[21]. On the other hand, researchers eliminated PAES from the competition because it exhibits poor performance in terms of the spread of solutions. We define the spread of solutions as the diversity of non-dominated solutions achieved by an algorithm.

A difference in spread of solutions between PAES and NSGA-II is represented in Figure 3.2.

![Figure 3.2: Difference between NSGA-II and PAES in spread of solutions](image)

where f1, f2 are front one and front two.
Chapter 4

Experiment

An experiment is conducted to find the answers for RQ2. RQ2: How will the participants get affected when the problem is solved using a single-objective optimization model rather than multi-objective optimization model?

The experiment involves three participants. The first participant is Port, and the other two are customers named Carrefour and Metro.

The problem is solved by developing two models to conclude a solution for RQ2, i.e. Single-objective optimization model and Multi-objective optimization model. In the first model of solving the problem, three container picklists are created by optimizing a single objective at a time. In the second model, a container picklist is created by optimizing multiple objectives simultaneously. By doing so, we can compare the results and assess the performance between these two models.

4.1 Environment

The hardware and software equipment used to perform experiment are mentioned in the table 4.1.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Linux Ubuntu</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU</td>
<td>NVIDIA GeForce GTX 1050 Ti 4GB</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel Core i7-8550U @ 1.80GHz x 8</td>
</tr>
<tr>
<td>Memory (RAM)</td>
<td>12GB DDR4</td>
</tr>
</tbody>
</table>

Table 4.1: Configuration of hardware and software.

4.2 Data Collection

The company Swarm engineering works to improve key metrics such as throughput and cost. They offer their suggestions to logistics companies in minimizing cost and maximizing value. The experiment is conducted using a dataset provided by this company. This company also had proficiency in the fields of multi-agents, machine learning and blockchain.
4.3 Data Labelling

The dataset consists of information related to the containers in the port. Every container has its properties and value. The data is stored in a CSV file, and the schema is in the following sequence:

< id ><x><y><z><Carrier><Price><Value>

Where,
ID represents the container id,
x,y,z represents the x,y,z coordinates respectively where the container is present,
Carrier represents the name of the company to which the container belong,
Price represents the contract price of a container,
Value represents the business value of a container.

4.4 Implementation

The implementation is divided into four steps,
(i) Developing a port agent
(ii) Developing a client agent
(iii) Developing single-objective optimization model
(iv) Developing multi-objective optimization model

Figure 4.1 represents the container stacking in ports and axis assigned to the stacks of containers.

Figure 4.1: Container arrangement in ports
There are four rules concerning the container stack,

- Each container resides within a specific x,y,z coordinate.
- The maximum x coordinate that containers can be placed in is 10, the maximum y coordinate is 10 and the maximum z coordinate is 5.
- In any column, the container with the highest z coordinate is the only container that can be moved.
- Containers cannot float in air, for example a container can only be placed on (5,2,5) if and only if container is already placed on (5,2,4).

Two terms, promoted and excluded, are frequently used in the upcoming sentences. I will explain these words for better understanding. Promoting a container means including a container to the container picklist and excluding means either removing a container from container picklist or not considering a container while deciding on best possible containers available for promotion.

**Port agent**

The problem of the port agent is modelled as a Linear Integer Programming (LIP) problem. A variable is assigned to each container, which can be either 0 or 1 and constrained the sum of the variables to be less than 100. The variables of the containers that have containers on top are constrained to be less than the variable of the container on its top, which in practice means that the container cannot be removed if there is a container on its top.

The objective can also be defined linearly by multiplying each variable by its contract value and adding them all.

Since the problem contains integers, it is solved by using branch and cut method, one of the LIP solving methods. GNU Linear Programming Kit (GLPK)[32] solver is used in order to achieve this. GLPK package is a widely used software package for solving large scale linear programming and mixed-integer programming problems.

**Client agent**

The best solution for client agent is found procedurally in the following way,

1. Find the best available promotion to make.
2. Make promoted container and exchanged container unavailable.
3. Repeat until the best available promotion is worse than not promoting anything.

By doing so, the problem is reduced to finding the best available promotion at each step. This can be achieved by using following algorithm.

1. For each excluded container we find the best set of available included containers to be removed in place of it.
2. We compute the profit increase in making each promotion.
3. The promotion with the higher profit increase is selected as the best.
By doing so, the problem is further reduced to finding the best set of available included containers to be removed for a given excluded container at each step. This can be achieved in the following method.

1. If the z-value of the container to be promoted is greater than the current selected set of the containers to remove, then find the available included container that would produce the best profit increase, by iterating over all of them.
2. If there is no container that keeps the profit increase positive then mark this container as not to be promoted.
3. Repeat until the z-value of the container to be promoted is lower or equal than the current selected set of containers to remove (or the container is marked as not to be promoted).
4. If marked as not to be promoted return 0, else return the current selected set of containers to remove.

**Single-objective optimization model**

For each of the objectives (carrefour’s profit, metro’s profit, sum of contract prices), a linear program like the port problem is optimized, but with the updated prices obtained from the clients’ agents.

As in the port case, assign a variable to each container which can be 0 or 1, constraint the sum of the variables to be less than 100 and the variables of containers that have containers on top is constrained to be less than the variable of the container on top, which in practice means that the container can’t be removed if there is a container on top still.

The objective can be defined linearly by multiplying each variable by its value and adding them all

GLPK solver is used for these three optimizations as well and this model gives us three new different solutions, each focusing on a different objective.

**Multi-objective optimization model**

The final Pareto front solutions are found by NSGA-II (genetic algorithm).

The problem is modelled as finding for each x,y coordinates how many containers to be delivered to their owner, since the order of delivery is not taken into account for this problem, except by having to deliver first the ones on top, but having decided how many to deliver from a coordinate the order of delivery is obvious (from top to bottom).
This gives us 100 variables (10x10) that we need to find. Each value is an integer and the range goes from 0 to the amount of containers that are at its respective x,y coordinate.

For a genetic algorithm to work with this system, we define a function that computes the value of each objective given a list of values for these variables. The objective values that this function returns are: The profit improvement for each of the customers, the sum of the contract prices and the sum of containers delivered. Also the constraint for this problem is that the number of moves done have to be less than 150.

Having defined the variables, the objectives and the constraint we run the genetic algorithm NSGA-II for 1000000 iterations to find the Pareto front.

Having the Pareto front defined we need to find one of those solutions for the problem. This can be found by calculating an average of all the Pareto front solutions and find the solution closest to the average.

Figure 4.2 represents steps that a CSV file passes through different agents in order to obtain final result.

![Figure 4.2: Sequence of steps involved in the experiment](image-url)
4.5 Performance Measures:

The performance of the model is based on the algorithm that was chosen to find the Pareto front. So, the performance is based on the performance exhibited by the NSGA-II algorithm. The metrics that were consider while comparing genetic algorithms are Convergence and Divergence.

Convergence
The output comes closer to a particular value on performing several iterations. This behaviour is called Convergence\[28\]. This value can be found in the following method,
(i) A certain amount of uniformly spaced solutions from best Pareto-optimal front in the objective space.
(ii) We calculate a minimum Euclidean distance to each solution obtained after applying the algorithm from the chosen points.
(iii) By calculating the average of these distances, the convergence value can be found.

Euclidean distance is straight line distance between two points in a given objective space\[21\].

Divergence
The extent of spread achieved among the obtained solutions is defined as Divergence\[28\]. The value of divergence can be found by the following method,
(i) We calculate the euclidean distance between two consecutive solutions in the Pareto optimal set.
(ii) An average is calculated for these distances.
(iii) We calculate the extreme solutions in the Pareto optimal set.

Formula for finding Divergence:

\[
Divergence = \frac{df + dl + \sum_{i=1}^{n-1} |di - D|}{df + dl + (N - 1)D} \tag{4.1}
\]

Where,
df and dl are the euclidean distances between extreme solutions and the boundary solutions of obtained Pareto optimal set,
N is the number of solutions in Pareto optimal set,
D is the average of two consecutive solutions in Pareto optimal set\[28\][21].
Chapter 5

Results

In this section, the results obtained after experimenting are presented. The profits of customers are calculated using below formula.

\[ BusinessProfit = BusinessValue - ContractPrice. \] (5.1)

The schema of the output is,

Deliver container <Container ID> from coordinate (x,y,z) with updated price <Value>.

On calculating the individual profits of the customers, the outputs in different instances are as follows.

![Optimization of Carrefour Profits](image)

Figure 5.1: Case(i): Optimization of Carrefour Profits
Table 5.1: Port’s income and customers profit when containers are picked in favour of Carrefour.

<table>
<thead>
<tr>
<th>Port’s income</th>
<th>Carrefour’s profit</th>
<th>Metro’s profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1325</td>
<td>275</td>
<td>31</td>
</tr>
</tbody>
</table>

Figure 5.2: Case(ii): Optimization of Metro Profits

Table 5.2: Port’s income and customers profit when containers are picked in favour of Metro.

<table>
<thead>
<tr>
<th>Port’s income</th>
<th>Carrefour’s profit</th>
<th>Metro’s profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1309</td>
<td>22</td>
<td>312</td>
</tr>
</tbody>
</table>
Figure 5.3: Case(iii): Optimization of sum of Contract Price

![Bar chart showing optimization of contract prices, port, Carrefour's profit, and Metro's profit.]

Table 5.3: Port’s income and customers profit when containers are picked in favour of Port.

<table>
<thead>
<tr>
<th>Port’s income</th>
<th>Carrefour’s profit</th>
<th>Metro’s profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1777</td>
<td>-71</td>
<td>-22</td>
</tr>
</tbody>
</table>
Table 5.4: Port’s income and customer’s profit while using Multiple objective optimization model.

<table>
<thead>
<tr>
<th>Port’s income</th>
<th>Carrefour’s profit</th>
<th>Metro’s profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1511</td>
<td>112</td>
<td>112</td>
</tr>
</tbody>
</table>
6.1 Analysis of Literature review

On conducting a literature review to answer RQ1, NSGA-II was found out to be the best existing algorithm in solving MOOP. This conclusion is based on results for the experiments conducted by researchers on various similar problems. The researchers also measured the performance of this algorithm with other algorithms and on comparing the convergence and divergence of different algorithms, NSGA-II exhibited better performance among all.

6.2 Analysis of Experiment

On analyzing the results, solving the problem by using a multi-objective optimization model has given impartial outputs among the three participants. Carrefour and Metro have equal profits, and contract prices made by the port are better in this case when compared to the case (i) and case (ii). We can observe that there are no unsatisfactory participants while using this model, which we faced in the remaining cases.

6.3 Discussion

The main research question was, How will the participants get affected when the problem is solved using a single-objective optimization model rather than multi-objective optimization model. Therefore the problem is solved in two different approaches. The first approach is to consider a single objective at once by not bothering about the rest of the participants. The second approach does consider all the other participant’s objectives and provide a solution. On comparing the results, we can know which model performed better between them.

Figure 5.1 shows the result of profits made by customers and income made by the port when containers are picked in favour of Carrefour. We can observe, Carrefour’s profit is 275 and Metro’s profit is 31. Metro’s profit is very less compared to Carrefour’s profit, and we can conclude that the unsatisfactory participant, in this case, is Metro. On adding the contract prices of delivered containers, the total is 1325,
which is the port’s income.

Figure 5.2 shows the result of profits made by customers and income made by the port when containers are picked in favour of Metro. We can observe, Metro’s profit is 312 and Carrefour’s profit is 22. Carrefour’s profit is very less compared to Metro’s profit, and we can conclude that the unsatisfactory participant, in this case, is Carrefour. On adding the contract prices of delivered containers, the total is 1309, which is the port’s income.

Figure 5.3 shows the result of losses faced by customers and income made by the port when containers are picked in favour of Port. We can observe, Carrefour is in loss of 71 and Metro’s loss is 22. In this case, the results are unsatisfactory for both the customers. On adding the contract prices of delivered containers, the total is 1777, which is the port’s income.

Figure 5.4 the result of profits made by customers and income made by the port when containers are picked by using multi-objective optimization model. In this case, Carrefour and Metro made equal profits, that is 112. On adding the contract prices of delivered containers, the total is 1511, which is the port’s income.

This research proves that by individually optimizing single objectives once at a time when more that one participant with individual objectives is involved will affect the other participant(s). We can apply the same approach used to solve the MOOP in this research to similar problems that occur in different sectors. So, this research helps in solving the problems where multiple objectives are involved simultaneously irrespective of the field.

6.4 Validity Threats

In this section, validity threats regarding this project are discussed,

6.4.1 Internal Validity

Internal validity mainly focuses on how well the research has been directed to answer the research questions. This validity is achieved by selecting an algorithm that produced good results in the experiments conducted by other researchers. The dataset used to conduct the research is filtered, and unnecessary data which does not contribute to this thesis is excluded from the dataset.

6.4.2 External Validity

External validity mainly focuses on the generalization of the results produced from the experiment. The data used to experiment is real-world data, and the set of rules considered while experimenting are real-world practices. So we can generalize the implementation and the results in this research.
6.4.3 Conclusion Validity

Conclusion validity focuses on the precision of the results produced from the experiment. This validity is achieved by ensuring proper implementation and selecting the appropriate methods to conduct the research.
Chapter 7

Conclusion and Future Work

Among all the problems faced by companies, Multiple Objective Optimization Problems (MOOPs) tend to possess difficulties in solving. These type of problems are common in every sector, and the problem considered for this thesis belongs to the logistics sector. The problem is solved using two models, i.e. the single-objective optimization model and multi-objective optimization model. A literature review is conducted to find the best suitable algorithm for solving the MOOP in the multi-objective optimization model. On comparing different algorithms, NSGA-II has given best results in earlier researches, so this algorithm was chosen to solve our MOOP. An experiment is conducted by developing AI-based agents for the port and customers. After analyzing the results, I concluded that multi-objective optimization model performed better than the single-objective optimization model because of exhibiting impartiality among the participants.

RQ1. What could be the suitable algorithm for solving Multiple Objective Optimization Problem (MOOP)?
Answer: After conducting a literature review, NSGA-II is found to possess high convergence and divergence rates. This algorithm also produced good results compared to other algorithms in the experiments conducted by earlier researchers.

RQ2. How will the participants get affected when the problem is solved using a single-objective optimization model rather than multi-objective optimization model?
Answer: After experimenting, the multi-objective optimization model produced better results than the single-objective optimization model by eliminating trade-off’s and ensuring impartial results between the participants.

7.1 Future Work

The model built in this project concentrates on optimizing container picklist concerning port as well as customers. The values considered for building this model are limited to id, x, y, z, price and value. The dataset also consists of two other values such as Refrigerated and Hazmat. These values represent if the container is refrigerated or not, and if it is dangerous or not. Introducing these values to this model changes the priority of the containers, thus resulting in a change in final profits of both port and customers. So, the existing model can be further extended by introducing these values.
References


