Multi-Objective Vehicle Loading and Routing Problem for Fresh Fruit and Vegetable Transportation

Miss Supisara Krairiksh

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Industrial Engineering
Department of Industrial Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2022

ปัญหาการบรรทุกสินค้าและกำหนดเส้นทางของยานพาหนะสำหรับการขนส่งผักและผลไม้สดแบบหลายวัตถุประสงค์

น.ส.ศุภิสรา ไกรฤกษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2565

Thesis Title	Multi-Objective Vehicle Loading and Routing Problem for Fresh Fruit and Vegetable Transportation
By	Miss Supisara Krairiksh
Field of Study	Industrial Engineering
Thesis Advisor	Associate Professor PISIT JARUMANEEROJ, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Engineering

-- Dean of the FACULTY OF ENGINEERING

(Professor SUPOT TEACHAVORASINSKUN, D.Eng.)

THESIS COMMITTEE

-- Chairman

(Associate Professor Daricha Sutivong, Ph.D.)

-- Thesis Advisor

(Associate Professor PISIT JARUMANEEROJ, Ph.D.)

-- Examiner

(PUWADOL DUSADEERUNGSIKUL, Ph.D.)

-- External Examiner

(Associate Professor Chuvej Chansa-ngavej, Ph.D.)

ศุภิสรา ไกรฤกษ์ : ปัญหาการบรรทุกสินค้าและกำหนดเส้นทางของยานพาหนะสำหรับการขนส่งผักและผลไม้สดแบบหลายวัตถุประสงค์. (Multi-Objective Vehicle Loading and Routing Problem for Fresh Fruit and Vegetable Transportation) อ.ที่ปรึกษาหลัก : รศ. ดร.พิศิษฏ์ จารุมณีโรจน์

ปัจจุบัน รถห้องเย็นแบบหลายช่อง (Multi-Compartment Vehicle) ได้ถูกนำมาใช้ในอุตสาหกรรมการขนส่งแบบควบคุมอุณหภูมิอย่างแพร่หลาย เนื่องจากความสามารถในการปรับเปลี่ยนความจุที่อุณหภูมิแตกต่างกันในแต่ละช่องได้ เพื่อศึกษาความสัมพันธ์ระหว่างต้นทุนการขนส่งและประโยชน์จากการใช้รถห้องเย็นแบบหลายช่อง (Multi-Compartment Vehicle) ในการขนส่งผักและผลไม้สด ผู้วิจัยจึงได้ทำการออกแบบสร้างจำลองปัญหาการบรรทุกสินค้าและการกำหนดเส้นทางของยานพาหนะสำหรับการขนส่งผักและผลไม้สด (MCVRLP) โดยมีวัตถุประสงค์หลักสามประการ ได้แก่ (1) ลดต้นทุนการขนส่งโดยรวมให้น้อยที่สุด (2) ลดการปล่อยก๊าซคาร์บอนไดออกไซด์ให้น้อยที่สุด และ (3) ลดการสูญเสียน้ำหนักของผักและผลไม้สดให้ได้น้อยที่สุด ผู้วิจัยยังได้ออกแบบแก้ปัญหาดังกล่าวผ่านแบบจำลองทางคณิตศาสตร์ และวิธีวิวัฒนาการเชิงพันธุกรรม (Genetic-based Evolutionary Algorithm) จากผลการทดลอง ผู้วิจัยพบว่า CPLEX ไม่สามารถคำนวนหาคำตอบที่ดีที่สุดสำหรับปัญหาขนาดใหญ่ที่มีความซับซ้อนได้ เนื่องจากประสบปัญหาความจำไม่เพียงพอ ตรงกันข้าม ฮิวริสติกส์ที่ถูกพัฒนาขึ้นกลับมีประสิทธิภาพเป็นที่น่าพึงพอใจในทุกขนาดปัญหา กล่าวคือ วิธีการฮิวริสติกส์สามารถค้นหาคำตอบที่มีคุณภาพดีเทียบเท่ากับคำตอบที่เหมาะสมที่สุดจาก CPLEX สำหรับปัญหาขนาดเล็ก และคำตอบที่มีคุณภาพดีกว่า CPLEX สำหรับปัญหาขนาดใหญ่ นอกจากนี้ ผู้วิจัยยังพบว่า การเลือกเส้นทางการขนส่งและการจัดสรรผักและผลไม้สดในแต่ละคันรถเป็นปัจจัยสำคัญที่ส่งผลต่อคุณภาพของคำตอบ โดยคำตอบที่ใช้จำนวนยานพาหนะน้อย อาจมีต้นทุนที่สูงกว่าคำตอบที่ใช้จำนวนยานพาหนะที่มากกว่า เนื่องจากต้นทุนส่วนหนึ่งเป็นผลมาจากการจัดสรรผักและผลไม้ในแต่ละคันรถ

สาขาวิชา	วิศวกรรมอุตสาหการ	ลายมือชื่อนิสิต ....
ปีการศึกษา	2565	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

6370288421 : MAJOR INDUSTRIAL ENGINEERING
KEYWORD: multi-compartment vehicle, Many-objective optimization, NSGA-III, bin packing problem, hybrid-heuristic

Supisara Krairiksh : Multi-Objective Vehicle Loading and Routing Problem for Fresh Fruit and Vegetable Transportation. Advisor: Assoc. Prof. PISIT JARUMANEEROJ, Ph.D.

Multi-compartment refrigerated vehicles are recently utilized in the cold chain industry, due largely to their flexibility in storage capacities with different temperature settings. To better comprehend the costs and benefits of this vehicle type in fresh fruits and vegetable transportation, a multi-compartment vehicle routing and loading problem (MCVRLP) with three different objectives – namely (i) minimizing total transportation cost, (ii) minimizing CO2 emissions, and (iii) minimizing weight loss of fresh fruits and vegetable – is herein explored and solved by mathematical formulation and genetic-based evolutionary algorithm approaches. Based on our computational results, we find that large and complicated MCVRLP instances are less likely to be solved to optimality by CPLEX solver within a reasonable computational time period, due to its complexity. However, the proposed genetic-based evolutionary algorithm seems to work well under all MCVRLP settings, as it could provide solutions that match the optimal solutions to small MCVRLP instances and those that outperform CPLEX solutions in larger ones. We also find that, with the same input information, slight differences in loading and routing may lead to solutions with totally different quality; and, interestingly, solutions with fewer vehicles might be worse off in terms of cost under the same routing, due to different temperature settings, which are results from loading decisions.

Field of Study:	Industrial Engineering	Student's Signature
		....
Academic Year:	2022	Advisor's Signature
		....

ACKNOWLEDGEMENTS

This thesis would not have been possible without the assistance of several people. I would like to express my gratitude to my advisor, Assoc. Prof. Pisit Jarumaneeroj, for his guidance throughout this long process of my graduate program. Also, many thanks to members of my examiner, Associate Professor Daricha Sutivong, Dr. Puwadol Dusadeerungsikul, and Associate Professor Chuvej Chansa-ngavej, who provided advice and assistance. Finally, I would like to thank my parents and numerous friends who supported me throughout this difficult time.

Supisara Krairiksh

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER1

INTRODUCTION

1.1 Introduction

With the increased popularity of global trade, transportation has become a vital component in modern supply chain management for most businesses. It is the process of coordinating and moving products from one location to the desired destination in a timely, safe, and effective way. Considering the importance of transportation planning, well-planned transportation improves the supply chain by reducing product and time waste and generating cost-effective delivery routes (Li, 2014). Transportation accounts for approximately one-third of the company's total production costs. The method of transportation is a significant consideration when planning the delivery under a specific time and product quality concerns. Furthermore, road transport is the most popular means of transportation due to its ability to distribute products across the country, particularly trucks, which can reach any region worldwide (Sarder, 2021). Hence, transportation cost has become a primary concern for decision-making in supply chain management as the effectiveness of transportation often leads to positive business results.

Regarding the continuous improvement in people living standards and an instant increase in fresh produce demand, cold chain logistics are widely used for preservation and transportation to slow down microbial growth by maintaining the optimum temperature inside the container (Maiorino et al., 2021). The temperature-controlled transportation is known as refrigerated vehicles. Refrigerated vehicles are commonly used in perishable commodities distribution to control their internal temperature using cooling equipment. However, refrigerated vehicles have a cost concern due to their high operating costs (Song and Ko, 2016). Nonetheless, despite its expensive cost, with the ability to control temperature, it is possible to maintain the spoilage rate of perishable products during long transportation (Lawton, 2016). Due to technological advancement, refrigerated-type vehicles are categorized into single-compartment vehicles (SCV) and multi-compartment vehicles (MCV). The multi-compartment vehicles were originally used in gasoline supply chain networks and waste collection but are now utilized for perishable product distribution and grocery delivery. Because each compartment can be set to a specific temperature zone, distributors can consolidate the shipment with multiple types of products within one vehicle. In addition, the compartment size can be efficiently adjusted so that the goods can be loaded with no loss of capacity. However, there are trade-offs between single-compartment vehicles (SCV) and multi-compartment vehicles (MCV). First, multi-compartment vehicles (MCV) cost more in procurement costs. Second, the number of stops at customer locations and vehicles used is fewer when using multi-compartment vehicles (MCV). Lastly, the overall transportation time is lesser with multi-compartment vehicles (MCV) (Ostermeier and Wäscher, 2021). As a result, effective temperature-controlled transportation planning is required for businesses to thrive in the modern economy.

Over the years, global warming has become a primary interest worldwide as it increases a threat to human beings and other creatures on earth (Chen et al., 2019). Considering the rapid growth in refrigerated vehicle usage, a lack of appropriate transportation management has become a significant concern in the cold chain logistic, resulting in higher energy consumption and carbon emissions. The refrigeration system relies on the diesel engine to operate the cooling system, which emits up to 40% of the vehicle's carbon dioxide emissions. Even though a refrigeration system is necessary for maintaining the safety and quality of many perishable commodities, such as fresh fruits and vegetables, about 15% of the world's fossil fuels are used in refrigerated transportation (Stellingwerf et al., 2018). Consequently, adopting energy efficiency has become an essential strategy for reducing energy consumption, particularly in the transportation sector, due to its heavy reliance on fossil fuels.

Regarding the healthy dietary lifestyle trend, the demand for fresh fruits and vegetables has increased dramatically and is expected to grow continuously (Qin et al., 2019). The freshness of the fresh fruits and vegetables delivered at the destination is a primary concern as it directly impacts customer satisfaction. Fresh fruits and vegetables typically have a short life cycle and deteriorate instantly (Wang and Zhan, 2016). The common challenge for fresh fruits and vegetable transportation is to balance the quality standard and transportation cost (Nakandala et al., 2016). The mismanagement of decision-making at each supply chain stage results in a significant quality loss of fresh fruits and vegetables, negatively impacting the saleable quantity, commercial value, and customer satisfaction (Surucu-Balci and Tuna, 2021). For example, freshness loss (known as water loss) is the cause of revenue reduction as the commodities are sold by weight (Damrongpol and Pradorn, 2016). Fresh fruits and vegetables are considered highly perishable products, so their freshness is a crucial indicator of their value in the market (Chen and Shi, 2019). In addition, fresh commodities have relatively thin margins; efficient management and decision-making are necessary for maintaining product market value, consumer acceptability, food safety, and long-term business reputation (Soto-Silva et al., 2016).

Despite the fact that Thai fruits are popular in the market and have comparative advantages, Thailand faces some challenges in exporting fresh fruits. The nature of fresh fruit production in Thailand, for example, is heavily influenced by nature and climate. Because fresh fruit quality and output are difficult to control, harvesting and exporting fresh fruits that meet market standards is challenging. The reliance on nature affects the production and prices of fruits as the seasons change. Furthermore, Thailand's fruits have encountered difficulties with international trade barriers. The tax rate of global trade barriers, in particular, has caused Thailand's fruit prices to be higher in the international market than its competitors. The activities such as production and distribution have become keys to competitiveness in the global market trade. Therefore, careful transportation planning is potentially an important factor in the success of Thai exporters (Pongpanich and Phitya-Isarakul, 2008). Due to varied shipping amounts and temperature requirements, utilizing multi-compartment vehicles is expected to be beneficial for consolidating different fruit types throughout the offseason within the same vehicle. In addition, consumers are willing to pay higher prices for exotic and off-season fruits.

In summary, the research on the cold chain considers cost, global warming issues, and fresh produce freshness at the same time is still scarce. This thesis aims to propose a decision-making and problem-solving strategy to obtain a highly cost-effective solution for fresh fruits and vegetable transportation with consideration of transportation cost, the freshness of fresh fruits and vegetables, and carbon emissions in cold chain transportation.

1.2 Objectives

The objective of this study is to investigate the impact of multi-compartment vehicle layout and loading decisions on the multi-compartment vehicle routing and loading problem (MCVRLP) in relation to three distinct objectives: (1) minimizing transportation expenses, (2) reduction of overall carbon emissions, and (3) preservation of product quality. To attain this goal, mathematical models and genetic-based algorithms are employed to identify Pareto optimal solutions.

1.3 Scope of Research

- This study only includes the situation where all vehicles are homogeneous fleets with the same capacity with a limited number of compartments.
- This study focuses on the case of Thai fresh fruit exportation, which starts from the distribution center to other countries' wholesale markets as the destinations.
- This study only considers transporting fresh fruits at the optimal temperature and relative humidity, without taking temperature fluctuations into account. As a result, the decision model is limited to temperature-controlled road transportation at a constant temperature.
- The study investigates the situation in which fresh fruits are sold at the destination by weight (saleable weight). And the remaining weight was used to quantify the quality of fresh fruits.

- The data collection is not carried out in this study; instead, all required numerical data was gathered from pieces of literature as instances to validate the model.

1.4 Expected Outcome

- The mathematical model and the effective algorithm for solving multi-compartment vehicle loading and routing problem with multiple objectives are achieved.
- Decision-making and problem-solving strategy to obtain a highly-cost effective solution which included (1) the number of vehicles used each day after consolidating different fresh commodities within a single vehicle, (2) the type of refrigerated multi-compartment vehicle, and (3) the quantity of fresh fruit assigned to the compartments and vehicles (4) the amount of diesel fuel consumed to reduce carbon emissions

1.5 Benefits of the Thesis

- To help Thailand further strengthen the competitive status of Thai fruits in terms of transportation cost.
- Able to transport fresh fruits to their destination before their end shelf life while minimizing the cost.
- Lessen carbon emission emits from operating refrigerated vehicles based on the developed model.
- The multi-compartment vehicles are effectively utilized in cold chain transportation.

CHAPTER2

LITERATURE REVIEW

This chapter summarizes the research in the field of distribution network design for perishable products. Food loss occurs in every stage of the food supply chain, such as production, postharvest, and processing, caused by a small investment in food production technologies and insufficient knowledge. Many papers have investigated the fresh produce supply chain from several perspectives, such as production and distribution planning, logistic model optimization, and refrigerated vehicle routing problem for perishable goods (Paam et al., 2016). A vast amount of ongoing work is being carried out to develop efficient models that integrate vehicle routing problems with the deterioration rate of perishable products to minimize food loss and total production costs. Furthermore, a significant amount of ongoing work focused on reducing energy consumption and carbon emission. Carbon emissions have become a critical issue worldwide and have increased sharply in the last 50 years. According to global carbon emissions data, the transportation industry accounts for 14% of total carbon emissions, with road transport accounting for more than 70% of total transportation emissions (Qin et al., 2019). Hence, the related literature about fresh fruits and vegetable quality loss, multi-compartment refrigerated vehicles and its attribute, carbon emission from refrigerated trucks, and optimization models are represented in this section.

2.1 Quality Loss in Fresh fruits and Vegetables

In food supply chain management, predicting fresh produce quality is a challenging task. The primary concern in decision-making is the product's freshness, which frequently represents the remaining shelf life. Customers typically choose the freshest items on the shelf first; in other words, freshness has a significant impact on customer decisions (Wang and Li, 2012). In the fresh fruit supply chain, quality loss occurs between the orchards and the customer. The major challenge of fresh fruit and vegetable exportation is to deal with the quality changes depending on surrounding environments and the nature of fresh fruits and vegetables.

Aside from the remaining shelf-life prediction, the quality loss can be determined by the water loss from fruit transpiration. In the case of products sold by weight, water loss due to transpiration during transportation has an economic impact on the saleable weight at the destination (Damrongpol and Pradorn, 2016). A pallet of durian weighing 500 kg, for example, accounts for a 15 kg loss during transportation. If durians are worth $10 per kilogram, the weight loss results in a $150 loss per pallet at retail.

As fresh fruits and vegetables' quality can be viewed from various perspectives, scholars have proposed numerous methods for predicting the remaining quality under temperature variation conditions, such as the Arrhenius, Davey, square root, linear, and exponential (Fu and Labuza, 1993). The Arrhenius equation, on the other hand, has been widely used in mathematical models for predicting shelf life or spoiling rate for transportation optimization purposes (Damrongpol and Pradorn, 2016). The Arrhenius equation is a broad model of temperature's effects on chemical reactions in foods or a kinetic model; thus, for shelf-life estimation, a time-temperature dependence is integrated into the Arrhenius equation. With constant humidity, the remaining shelf life is inversely proportional to the exponential rate of deterioration (Peleg et al., 2012). In most cases, the quality change for a time interval (t), a chemical reaction (n), and a constant rate of reaction (k) can be expressed as follows (Wang and Li, 2012) :

$$\frac{dq}{dt} = -kq^n \tag{2-1}$$

The temperature is a crucial factor in calculating the rate of a chemical reaction (k). K_a is a constant reaction rate, E_a is the activation energy that controls quality loss, $T(t)$ is an absolute temperature at some reference temperature (T_{ref}) and R_{gas} is the ideal gas constant. The chemical reaction rate equation is shown as follows (Wang and Li, 2012):

$$k = K_a e^{-[\frac{E_a}{R_{gas} * T(t)}]} \tag{2-2}$$

The Arrhenius equation with a zero-order reaction is used in mathematical models to predict shelf life (Damrongpol and Pradorn, 2016). According to equation (2-3), the quality changes for the time interval (t) at temperature ($T(t_i)$) can be calculated based on the initial quality (q_0) during periods {$i = 1, \dots, m$} (Wang and Li, 2012):

$$q(t) = q_0 - \sum_{i=1}^{m} k_A t_i e^{-[\frac{E_A}{R_{gas} * T(t_i)}]} \qquad (2\text{-}3)$$

The water loss from transpiration during transportation is an alternative factor for the quality of fresh fruits and vegetables at the desired destination of products sold by weight. Hence, the remaining weight is used to quantify the quality of fresh fruits and vegetables at the destination. The rate of water loss (WL) can be calculated from the chemical reaction rate (k) and Vapour-pressure deficit at temperature (T) with humidity (h), $VPD_{T,h}$. The $VPD_{T,h}$ can be obtained from the psychrometric chart, while the chemical reaction rate (k) can be obtained from Eq.(2-2) and (2-3) (Damrongpol and Pradorn,2016).

$$WL = k \cdot VPD_{T,h} \qquad (2\text{-}4)$$

However, how quality affects the cost is difficult to predict accurately. A framework to reduce the total cost of intermodal transport mode selection associated with the perishable product deterioration rate. The remaining quality of perishable products at the destination of shipment 'P', Q_P^T, can be predicted by using the following equation (Dulebenets and B. Ulak ,2016):

$$Q_P^T = Q_P^0 e^{-\varphi_p T_p} \qquad (2\text{-}5)$$

The decay function is approximated by utilizing a piecewise linear approximation for the non-linear decay function where φ_p is the decay rate of a perishable product of shipment 'p' and T_p is the total transportation time of shipment 'p'. The typical decay rate for meat and fresh vegetables are 0.0067 hour^{-1} and 0.0216 hour^{-1}, respectively. Nevertheless, the decay rate of perishable that transported in a refrigerated container is assumed to be φ_p = 0.0012. This model has been tested on importing seafood products to the United States. As a result, an increased product decay significantly impacts the change in intermodal network design (Dulebenets and B. Ulak, 2016).

A simple linear model was applied to estimate the decreasing rate of fresh vegetable quality for distributing fresh vegetables. The initial quality of fresh vegetables is assumed to be 100% initially. The quality loss time function is divided into the undamaged condition stage and quality condition-stage changes. In Figure 1, the stable period for a fresh vegetable starts from the harvested period (0) to time 'A'. The changes begin from the period 'A' onwards; however, only in-between periods 'A' and' B' is acceptable. Beyond period 'B,' the products are rejected as unqualified. For instance, the acceptable quality loss percentage is between 8% and 23% of initial quality in the Slovenian market (Osvald and Stirn, 2008).

Table 1 shows the shelf life and water loss rates for different fresh fruit types during various storage conditions calculated using the above equation (Lufu et al.,2020). However, information on Thai fresh fruits and vegetables is still scarce.

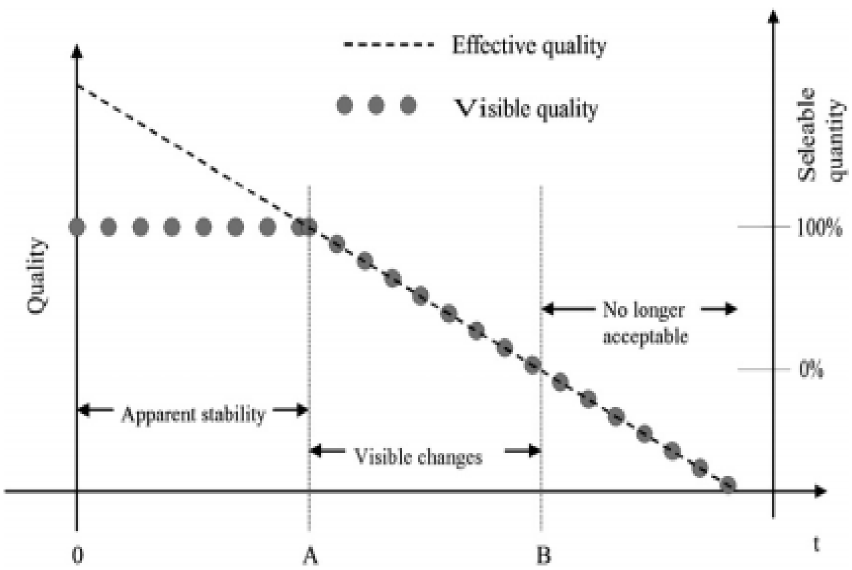


Figure 1: The simple linear model in quality loss

(Osvald and Stirn, 2008)

Fruit Types	Storage conditions, ℃	Shelf – life (days)	Related-Humidity (RH,%)	Water Loss (% per day)
Tangarine citrus (cv. Siam Banjar)	25℃	35 days	85-90 %	1.39
Strawberries	5℃	5 days	70%	1.65
Pomegranate	20℃	30 days	95%	0.19
Apples	20℃	18 days	95%	0.06
Plum	2℃	35 days	90%	0.17-0.31
Peach	2℃	28 days	90%	0.77
Pear	0℃	8 days	80%	0.18
Banana	15℃	11 days	89%	0.10
Cantaloupe	9℃	18 days	85-90%	0.25
Mangoes	5℃	15-20 days	90-95%	0.18
Oranges (cv. Valencia)	4℃	56 days	80%	0.19
Melon	2℃	18 days	85-90%	0.13
Table grapes	1.2℃	72 days	89%	0.10

Table 1: Example of Fruit shelf life and its water loss rate

(Lufu et al., 2020 2020)

2.2 Refrigerated Multi-Compartment Vehicles

The challenge of loading and unloading goods from multi-compartment vehicles is acknowledged. Horizontally loaded multi-compartment containers are commonly used in fuel distribution and waste collection. In the transportation of perishable commodities with multiple temperature zones, the compartment walls of a horizon layout may interfere with the product loading and unloading procedure (Fig. 2). To make loading and unloading perishable goods easier, a vertical layout, as shown in Fig 3, is recommended. To generate transportation routing adjustments, a minor loading layout modification is suggested (Ostermeier and Hübner ,2018).

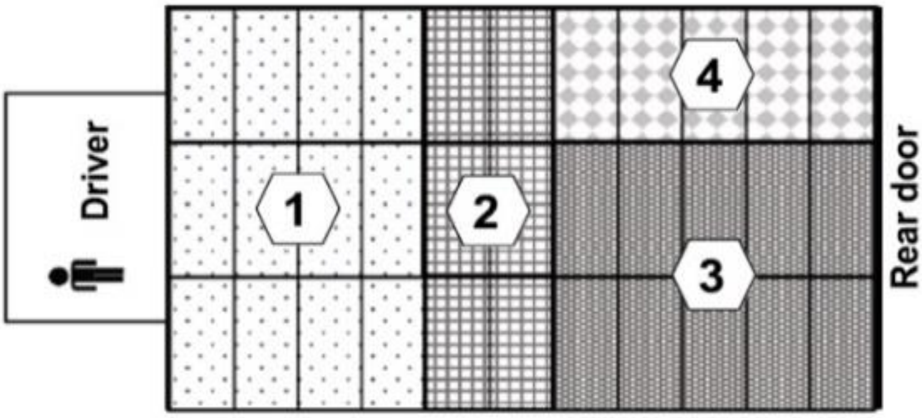


Figure 2: Loading layout of one MCV with four product segments, horizontally (illustrative example)

(Ostermeier and Hübner, 2018)

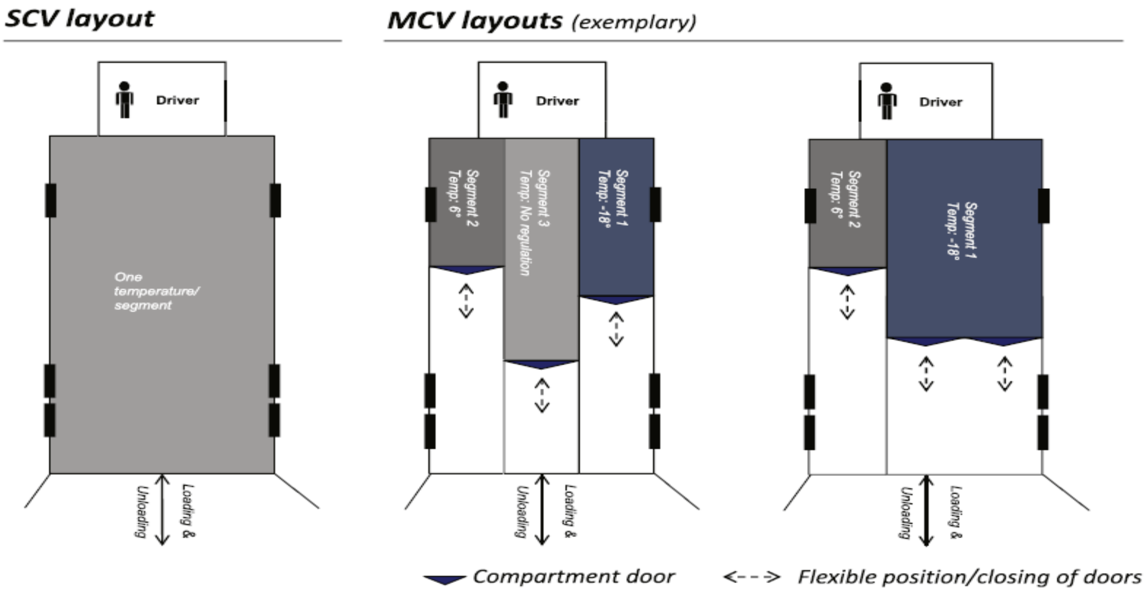


Figure 3: Examples of the layout of MCVs and an SCV for the temperature zone

(Ostermeier and Hübner, 2018)

With compartment-related aspects, there are three characteristics of vertical compartment layout. First, the compartment size can be fixed or variable for vertical layout. The number of compartments in each vehicle is based on selecting fixed or flexible compartment sizes. The compartment capacity is predetermined when using a fixed number of compartments. On the other hand, the capacity can flexibly split up to the vehicle's maximum capacity if adjustable compartment sizes are utilized. Second, product type assignments can be fixed (a particular product is assigned to a specific compartment) or flexible (all product types are assigned to any compartment based on problem criteria). Finally, the compartment can contain one product type and serve multiple customers or a single customer with various product types (Ostermeier and Wäscher, 2021). Furthermore, each distribution tour can change the partition structure based on the client and product segment delivery sequence (Ostermeier and Hübner, 2018). Each compartment's capacity limitation is a constraint for each delivery tour from the depot to several customers (Yahyaoui and Dekdouk, 2018).

2.3 Carbon Emission from Refrigeration Transportation

Recently, environmental issues have gained increasing attention worldwide, and numerous studies on the cold chain, energy consumption, carbon emissions, and cost have emerged and been conducted by many researchers. The carbon emission calculation and analysis of the cold chain are mainly divided into three phases which are 1.) the production process 2.) the pre-cooling process 3.) refrigerated vehicle transportation (Bin et al., 2022). However, refrigerated transport tends to be the critical phase of the cold chain due to its most negative impact on energy consumption and greenhouse gas emissions. In the refrigerated transportation phase, the vapor compression refrigeration (VCR) units are the most used systems in refrigerated transport. VCR systems consume approximately 15% of the world's electrical energy and contribute approximately 10% of greenhouse gas emissions. The VCR system operates on the vapor compression refrigeration cycle (Fig.4) and consists of the evaporator, compressor, condenser, and expansion.

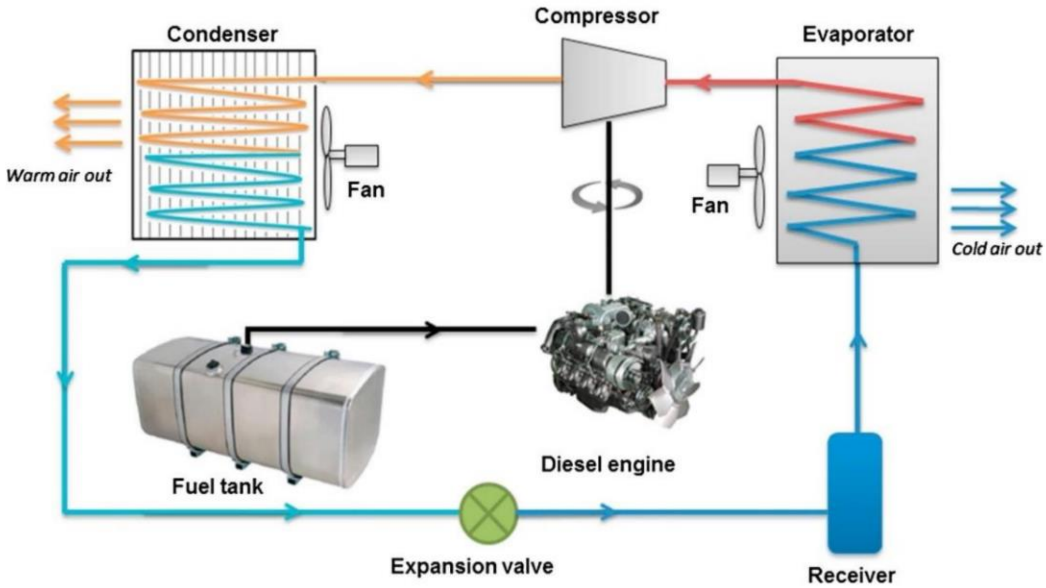


Figure 4: Vapor compression transport refrigeration unit driven by a diesel engine diagram

(Maiorino et al., 2021 2021)

There are several standard options for the VCR systems regarding the compressor power supply: direct belt drive, auxiliary diesel unit, auxiliary alternator unit, and vehicle alternator unit. The additional diesel unit is utilized for most heavy refrigerated vehicles. In other words, the compressor is powered by an auxiliary diesel engine (Maiorino et al., 2021). In diesel-driven vapor compression, diesel chemical energy is converted to electrical energy. The electrical energy is then utilized to power the heat transport from the truck's interior to the exterior. The Coefficient of Performance (COP) illustrates the heat conversion performance. It expresses the ratio of the heat removed as a function of energy supplied (Stellingwerf et al., 2018). The refrigerated transport systems are dependent on a wide range of operating conditions such as weather, the orientation of insulated walls, and loading/unloading operations frequency. As a result, the COP is generally lower than in static systems. The COP of the refrigerated vehicle with vapor compression refrigeration systems is generally between 0.5 and 1.5. In addition, the total heat load is also composed of thermal energy sources inside the compartment, such as transmission load depending on vehicle body size, infiltration load, and respiration heat load from fruits and vegetables (Maiorino et al., 2021). In addition, the different weight of products shipped affects the total heat load through its respiratory heat rate, which, as a result, requires more energy to maintain the low-temperature environment (Stellingwerf et al., 2018).

The carbon emissions calculation model is proposed as related to the loading of vehicles and driving distance. The model for carbon emission calculation from driving and refrigerant unit are shown as follows (Bin et al., 2022):

Carbon emissions from the refrigerated truck:

$$C_h = \frac{F_h C_f}{T_h} \tag{2-6}$$

C_h is carbon emission produced by the heavy truck transporting one kilogram of fresh fruits and vegetables for one kilometer, kgCO_2/(kg/km). F_h is the fuel consumption of a heavy truck driving one liter per kilometer (L/Km), while the diesel consumption for heavy refrigerated trucks is 0.2854 L/km. C_f is the carbon dioxide produced by burning 1L of fuel (kg/L). In this case, 1 L of burning diesel emits 2.630 kilograms of carbon dioxide (2.630 Kg/L). T_h is the capacity of the container.

Carbon emissions from refrigerant unit:

The refrigerated unit consists of an independent refrigeration unit that relies on its diesel engine to operate the cooling chamber. The carbon emission from the refrigerant unit is calculated from fuel consumption and cooling capacity depending on the thermal efficiency of the diesel engine and cooling system. The heat load from the refrigerated vehicle is analyzed to maintain the low-temperature environment for fresh fruits and vegetable transportation. The heat load of the refrigerated vehicles (H) is defined by the containers' body size and can be calculated by Eq. (2-7).

$$H = \frac{S}{V} \tag{2-7}$$

where

H : heat gain per unit volume of refrigerated vehicle, m^{-1}

S: surface area of the refrigerated vehicle,m^2

V: the volume of refrigerated vehicles, m^3

With the assumption that the temperature of fresh fruits and vegetables remains constant during transportation, the energy balance equation is applied, and the fuel consumption can be calculated by Eq. (2-9).

$$m \cdot q \cdot \eta \cdot COP = [H \cdot (T_0 - T) \cdot \frac{\lambda}{\delta} + m_v \cdot q_0]d\tau \quad \text{(2-8)}$$

where

H : heat gain per unit volume of refrigerated vehicle (m^{-1})

T_0: ambient temperature (℃)

T: fruits and vegetable temperature (℃)

m_v : fruits and vegetable mass (stack density) per unit volume (kg/m^3)

λ : thermal conductivity of thermal insulation materials ($W/m \cdot K$)

δ: the thickness of the insulation layer of a refrigerated vehicle (m)

q_0: respiratory heat of fruit and vegetables per unit mass ($J/kg \cdot s$)

$d\tau$: transportation time (s)

m: Fuel consumption, kg

η: The engine's thermal efficiency

q: Calorific value of fuel, KJ/kg

COP: Coefficient of performance of the refrigerated vehicle

The engine thermal efficiency of the diesel engine is generally 40%, and the unit of calorific value of diesel is 43.2 KJ/Kg. The calorific value (q) is the total energy released as heat when a substance entirely burns with oxygen under normal conditions.

2.4 Shortest Path Problem

This section discusses some basic definitions of the shortest path problem. Generally, the route selection approaches are primarily based on shortest-path algorithms, which intend to lower the cost of travel from one point to another. Transportation is often seen as the shortest path problem from origin to destination in freight logistic management. The shortest path problem has been applied to many applications based on decision criteria such as time, cost, and distance. Dijkstra's method is a well-known algorithm to find the optimal path. The algorithm enables calculating all shortest paths from a single point to all other points in a network where each edge requires a non-negative weight (Kien Hua and Abdullah, 2018). Dijkstra's algorithm begins with searching the nearest node from the source node that is connecting to the source node during the first iteration. In the second iteration, it finds the next node that is closest to the current node. The node must be a neighbor of the current node or the nearest node discovered for the next iteration. The iteration will end when the nth finds the first 'n' nodes closest to the source node (Rosita et al., 2019). For better demonstration, Fig 5 shows a simplified network diagram where edges connect nodes with non-negative weight. The line segments constrain different attribute values such as cost and time, determining the different costs of feasible paths. The shortest path function connects the source and destination nodes. The outcome is the optimal route out of all possible paths, which displays the total weight required by the Dijkstra algorithm for an optimal route (Kien Hua and Abdullah, 2018).

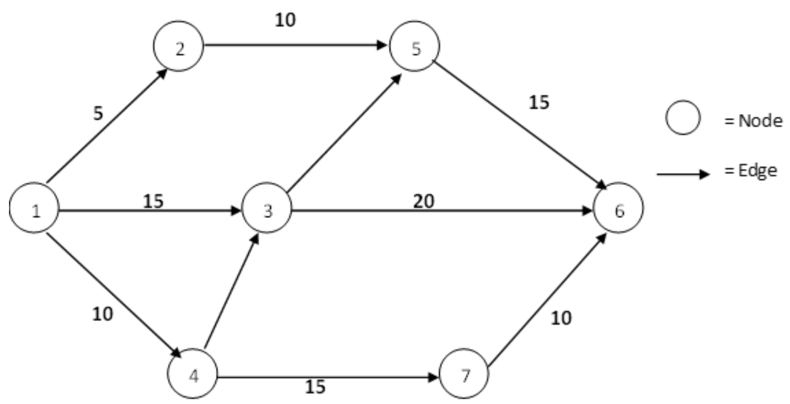


Figure 5: The illustration of the shortest path

However, a single-goal problem with a single unique optimal solution is impractical in real life. Dijkstra's algorithm was generalized to the Multi-Objective Shortest Path (MOSP) Problem. There are different sorts of multi-objective problems, which can be found in many fields such as finance, economics, robotics, manufacturing, logistics, and many others (Gunantara, 2018). In practice, multiple objective functions are considered while determining the best solution to satisfy the stakeholders. The multi-objective shortest path (MOSP) problem generally involves two objective functions known as bi-objective functions (Paixão and Santos, 2012).

This section introduces the terminology and the basic theory of bi-objective shortest path problems. Let $G = (N, A)$ be a connected network with a set of nodes $N = \{1, \ldots, n\}$ and a set of arcs $A = \{(i,j) \mid i,j \in N, i \neq j\}$. Two positive cost functions are denoted as 'c_{ij}' associated with each arc (i,j). The cost functions can represent time, distance, cost, or criteria for traversing arc (i,j), respectively. X_{ij} is a binary decision variable if there is a flow from node 'i' to node 'j '. The bi-objective shortest path (BSP) problem with origin node $\{o\} \in N$ and destination node $\{d\} \in N$ can be formulated as a network flow problem as follow (Raith and Ehrgott, 2009) :

Objective: Min/Max
$$\begin{cases} Z_1(x) = \sum_{(i,j)\in A} C_{ij}^1 x_{ij} \\ Z_2(x) = \sum_{(i,j)\in A} C_{ij}^2 x_{ij} \end{cases} \tag{2-9}$$

$$subject\ to$$

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ij} = \begin{cases} 1 & \text{if } i = \{o\} \\ 0 & \text{if } i \neq \{o,d\} \\ -1 & \text{if } i = \{d\} \end{cases} \tag{2-10}$$

$$X_{ij} \in \{0,1\} \quad \forall (i,j) \in A \tag{2-11}$$

The objective functions (2-9) are maximum or minimum cost functions. Constraint (2-10) represents the flow balance at the different nodes, and it provides that just one edge is chosen from the origin '*o*' to the intermediate node. The constraint also ensures that the total inflow and outflow are equal at any node and enforces that just one edge links the intermediate node to a destination node. Due to its complications, several algorithms are proposed to solve multi-objective shortest path problems.

Martin’s algorithm is initially employed to solve the multi-objective shortest path problem utilizing Dijkstra’s label setting approach (Paixão and Santos, 2012). The algorithm searches for a Pareto front from node to node in a network (Gräbener et al., 2010). The primary goal is to reduce the total cost and time spent traveling between the origin and destination. The main concept of martin’s algorithm is straightforward. There are two different sets of labels, known as temporary labels and permanent labels. A temporary label is considered a dominated path, whereas a permanent label is marked as a non-dominated path (Domuta et al., 2012). Labeling and evolutionary algorithms are commonly employed to solve multi-objective shortest-path problems and arrive at a Pareto optimal set (Mofaddel and Hamed, 2018).

Several academic papers have used Martin’s technique and metaheuristics such as NSGA, ant colony algorithms, and swarm particle algorithms to solve the bi-objective shortest path issue in recent years. A non-dominated sorting genetic algorithm (NSGA) is more often employed and better suited for addressing multi-objective problems with more than two objective functions and uncertain decision-makers' preferences. In practice, the multi-objective problem seeks to produce a set of solutions that satisfy an acceptable degree of quality while not being dominated by other answers (Konak et al., 2006).

For example, a non-dominated sorting genetic algorithm (NSGA) is employed to solve the bi-objective shortest path problem while considering the overall cost and delay time. As a result, in a single simulation run, NSGA effectively solves the Pareto optimal solution set (Chitra and Subbaraj, 2012). A combination of Martin’s algorithm and metaheuristic (NSGA II) is utilized to solve time-dependent shortest-path problems. However, due to too many non-dominated solutions, a lack of diversity, a problematic representation of the trade-off surface, and other factors, NSGA-II has difficulty handling many objectives (Ayed et al., 2011). To alleviate these difficulties, NSGA-III is proposed as an improved algorithm of NSGA II (Cui et al., 2019). NSGA-III preserves the diversity of non-dominant solutions and has performed well in testing and other problems with 3–15 objectives (Shao et al., 2022). Hence, NSGA III is a viable alternative for solving an efficient Pareto optimal solution set.

2.5 Bin Packing Problem

The bin packing problem is one of the most well-known problems in combinatorial optimization (Lodi et al., 2013). It has been applied to several applications in various fields. The bin packing problem (BPP) involves packing items into the bin where the objective is to minimize the number of bins that will hold all items. The number of bins is generally assumed to be large enough to contain all the items with various weights, noting that all bins are identical with the same capacity (Borges and Xavier, 2020). The bin packing problem (BPP) concept is integrated into the multi-compartment vehicle assignment and loading problem. In this section, terminology and the basic theory of BPP are introduced. Let 'i' is a set of items where $i = \{1, \dots, n\}$ with weight (w_i) and an unlimited number of identical bins ' j' with identical capacity 'c'. Let 'u' be any upper bound on the minimum number of bins needed where $j = \{1, \dots, u\}$. The integer linear program (ILP) for bin packing is formulated as follows (Lodi et al., 2013):

Decision Variables

y_i is a binary variable: 1 if bin 'j' is used; 0 otherwise

x_{ij} is a binary variable: 1 if item 'i' is assigned to bin 'j'; 0 otherwise

Objective: Min $\sum_{j=1}^{u} y_j$ (2-12)

$$subject\ to$$

$$\sum_{i=1}^{n} w_j x_{ij} \leq c y_i \quad \forall j \in \{1, \dots, u\} \tag{2-13}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \in \{1, \dots, n\} \tag{2-14}$$

$$y_i, x_{ij} \in \{0,1\} \quad \forall i \in \{1, \dots, n\},\ \forall j \in \{1, \dots, u\} \tag{2-15}$$

The objective function (2-12) minimizes the number of bins used. Constraint (2-13) ensures that the weight of items 'i' assigned to bin 'j' does not exceed the bin capacity. And constraint (2-14) provides that only one item 'i' is assigned to a different bin. However, several algorithms are adopted to solve the bin packing problem with a different logical sequence. There are two common bin packing problem (BPP) algorithms: online and offline. The online algorithm is the method that sequentially assigns items to bins without knowing the amount in advance, often known as Next-Fit (NF), First-Fit (FF), and Best-Fit (BF) algorithm. In contrast, offline algorithms are applied when all items are known and available for sorting in decreasing order. This algorithm is commonly known as Next-Fit Decreasing (NFD), First-Fit Decreasing (FFD), and Best-Fit Decreasing (BFD).

Even though there are several algorithms related to the bin packing problem, only *First-fit (FF)* will be discussed in this section. The first-fit algorithm is the simplest technique to allocate items into a bin. The algorithm considers the items according to the order from the input set, then assigned them to the lowest indexed initialized bin into which it fits. The new bin is initialized only when the current item cannot fit into any initialized bin (Junkermeier, 2015).

However, to increase the efficiency of exploring more in the search space, the minimum bin slack (MBS) heuristic is introduced by Gupta and Ho (1999). MBS heuristic is defined as a bin-oriented heuristic. It generated an optimal solution only when the sum of requirements of items is less than or equal to twice the bin capacity (Gupta and Ho, 1999). For each iteration, the algorithm attempts to assign the item to the bin with minimum slack. The assigned item to the new bin is removed from the list of items, all unassigned items are then considered for the next selection that will yield the minimum slack in the current bin (Dokeroglu and Cosar, 2014).

2.6 Transformation and Linearization Techniques

In this part, the transformation and linearization approaches for non-linear optimization models are discussed. In real-world optimization problems, in most cases, the problems have been formulated in the non-linear programming form such as the non-linear objective functions and non-linear constraints (Bazaraa et al., 2013). Nonlinearities are influenced by a fair number of equations such as the multiplication of decision variables (i.e., binary variables and continuous variables), maximum/minimum operators, absolute function, floor and ceiling function, and multiple breakpoint function. The exact transformation and linearization techniques are crucial to reducing the model complexity and computational time of the original non-linear optimization model The exact transformation and linearization techniques are essential to reducing the original non-linear optimization model's model complexity and computational time (Asghari et al., 2022). However, only the linearization of continuous variable multiplication is presented in this section.

Multiplication of Two Continuous Variables

The linearization of the multiplication of continuous is complex and extremely difficult to solve. Transforming the product of two continuous variables into an LP model requires specific transformations and substitutions in the original non-linear model. First, the two new continuous variables are defined with a valid equalities constraint in order to transform the product of two continuous variables into a separable function. For instance, consider x_1 and x_2 to be decision variables from a given feasible region. To convert $x_1 \cdot x_2$ into separable functions, $y_1$and y_2 are defined as two new continuous variables and two equalities constraints are introduced as follows (Asghari et al., 2022):

$$y_1 = \frac{1}{2}(x_1 + x_2) \tag{2-16}$$

$$y_2 = \frac{1}{2}(x_1 - x_2) \tag{2-17}$$

$$y_1^2 - y_2^2 := x_1 \cdot x_2 \tag{2-18}$$

As a result, the product of $x_1 \cdot x_2$ can be replaced in a separable function shown below which then can be linearized by using the piecewise approximation as stated in the next section.

Piecewise Linear Approximation

Piecewise approximation approaches have been widely used in recent decades to convert non-linear linear programming models into linear forms to obtain an approximate global optimal value. In general, the piecewise linear approximation techniques are applied after the transformation process in which the non-linear function is viewed as $f(x)$ of a given value x. The non-linear function $f(x)$ is divided into n separate segments where $a_i = \{a_0, \dots, a_n\}$ is the breakpoints of $f(x)$, $a_0 < a_1 < \cdots < a_n$ which shown in Figure 7 (Asghari et al., 2022)

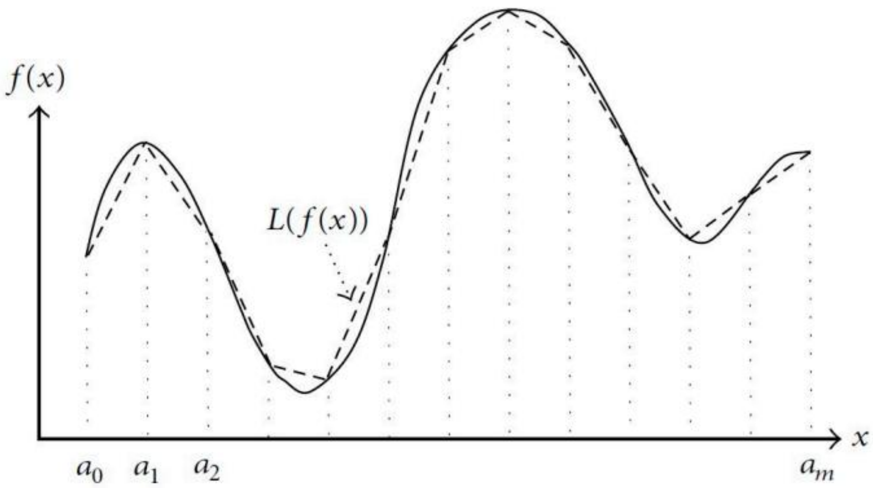


Figure 6: Piecewise Linearization of f(x)

(Asghari et al., 2022)

The extra binary variables $t_0, t_1, \dots, t_{n-1}$ are defined to force the given value x to be associated with the proper pair of consecutive breakpoints. The slope of each line between a_i and a_{i+1}, denoted as S_i. In addition, the approximation will be more accurate with more linear segments, but at the cost of higher computational complexity. Concerning all the components stated above, the piecewise approximating function is formulated as follows (Asghari et al., 2022):

$$L(f(x)) = \sum_{i=0}^{n} f(a_i)$$

$$subject\ to$$

$$x \geq a_i - (a_n - a_0)(1 - t_i) \quad \forall i \in \{0,1,..,n-1\} \tag{2-19}$$

$$x \leq a_{i+1} + (a_n - a_0)(1 - t_i) \quad \forall i \in \{0,1,..,n-1\} \tag{2-20}$$

$$f(x) \geq f(a_i) + S_i(x - a_i) - m(1 - t_i) \forall i \in \{0,1,..,n-1\} \tag{2-21}$$

$$f(x) \leq f(a_i) + S_i(x - a_i) + m(1 - t_i) \forall i \in \{0,1,..,n-1\} \tag{2-22}$$

$$S_i = \frac{f(a_{i+1} - a_i)}{a_{i+1} - a_i} \quad \forall i \in \{0,1,..,n-1\} \tag{2-23}$$

$$\sum_{i=0}^{n-1} t_i = 1 \quad t_i \in \{0,1\}, \quad \forall i \in \{0,1,..,n-1\} \tag{2-24}$$

2.7 Multi-Objective Optimization

In this section, multi-objective optimization and its relevant methods are discussed. The goal of solving multi-objective optimization problems is to find all possible trade-offs among conflicting objective functions (Murata and Ishibuchi, 1995). There are two general approaches for solving the multi-objective optimization problem. The first approach is to combine the multi-objective function into a single composite function using several methods such as the weight sum approach or moving the objectives to the constraint and leaving one function as an objective function. The second approach is to determine the Pareto front. A set of Pareto optimal solutions is a set of non-dominated solutions with respect to each other. As it is difficult to find one optimal solution, hence, a set of Pareto optimal solutions is known as a general approach to represent the solution to decision-makers depending on their preferences (Konak et al., 2006). According to numerous studies on multi-optimization problems, there are two common characteristics of the solutions. First, some decision variables have the same values in all Pareto-optimal solutions. Hence, this property of the decision variables indicates that the solution is optimal. Second, other decision variables take different values, resulting in a trade-off in the objective values of the solutions.

Furthermore, determining the Pareto front has two distinct goals: convergence to the Pareto-optimal solution and maintaining the diversity of the solution set (Deb, 2013).

The general form of multi-objective optimization (vector optimization) is presented as follows (Marler and Arora, 2004):

$$Minimize\ \ F(x) = \ [F_1(x), F_2(x), \dots, F_k(x)]^T \tag{2-25}$$

$$subject\ to$$

$$g_j(x) \ \leq \ \ 0, \quad j = 1,2, \dots m \tag{2-26}$$

$$h_l(x) = 0,\ \ l = 1,2, \dots, e \tag{2-27}$$

where

k is a number of objective functions.

m in the number of inequality constraints

x is a vector of decision variables

$F(x)$ is a vector of the objective function

Concept of dominance:

Most multi-objective optimization algorithms use the concept of dominance. Two solutions in objective space are compared to see if one outperforms the other. A dominance check has three possible outcomes to define the dominance relationship between two solutions: (1) solution 1 outperforms solution 2, (2) solution 1 is outperformed by solution 2, and (3) solutions 1 and 2 do not outperform each other (Deb, 2013).

Pareto Optimality:

When it is impossible to conclude that one solution dominates the other, the two solutions are said to be non-dominated. None of the solutions could be said that one is better than another. The non-dominated set is expected as a result of the dominance relationship process. Suppose there is a set of solution P, the non-dominated set of solution 'P'' is the solution that is not dominated by any member of the set P. Hence, the non-dominated set is known as the 'Pareto-optimal set' (Deb, 2013).

Numerous evolutionary algorithms (EAs) have been proposed in the last decade to solve multi-objective problems. The genetic algorithm (GA) is a well-known meta-heuristic, and its framework is commonly used as the main body for multi-objective design and optimization problems due to its population-based approach. The concept of genetic algorithm (GA) is based on natural evolution theory, which states that stronger species are more likely to pass on their genes to succeeding generations through reproduction. As a result, the species with the correct gene combination become dominant in their population, and the unsuccessful combination is thus eliminated by natural selection (Konak et al., 2006).

The genetic algorithm (GA) requires three essential components. The '*Population*' is a collection of solutions or individuals. Each solution is composed of a '*Chromosome'*, which is mainly comprised of a series of elements. Each element is known as a '*Gene*,' and it contains an individual's inherited traits. During the algorithm implementation, encoding the problem must be taken into account. The genetic algorithm encodes the problem by mapping the mechanism between the solution space and the chromosomes, where the chromosomes correspond to the unique solution in general. Selection, crossover, and mutation are the three main genetic operations in the genetic algorithm. The standard procedures are illustrated below (Konak et al., 2006):

Step1. <u>Population:</u> The initial population is generated at random

Step2. <u>Fitness Function</u>: Evaluate how good the solution to the problem.

Step3. <u>Selection:</u> Parental selection for reproduction and replacement

Step4. <u>Crossover:</u> Combining two pieces of information from two individuals (or parents)

Step5. <u>Mutation:</u> A small random chromosome change to obtain a new solution

Step6. <u>Repeat until stop criteria are reached</u>

The initial population is generated at random. Then, the fitness value is evaluated. The fitness function is a function that characterizes how well the solution is to the problem (i.e. how good the solution is concerning the problem in consideration). The selection operator, a parental selection, selects the fitter individuals for reproduction and replacement in the next generation after determining the fitness value. The crossover operator then combines the information from two individuals to diversify the population and expand the search space. A mutation operation is a minor random change in the chromosome that produces a new solution. It is used to keep and introduce genetic diversity into populations, ensuring that the algorithm does not become trapped in the local minima. The process is repeated throughout the generations until the stop criteria are met (Konak et al., 2006).

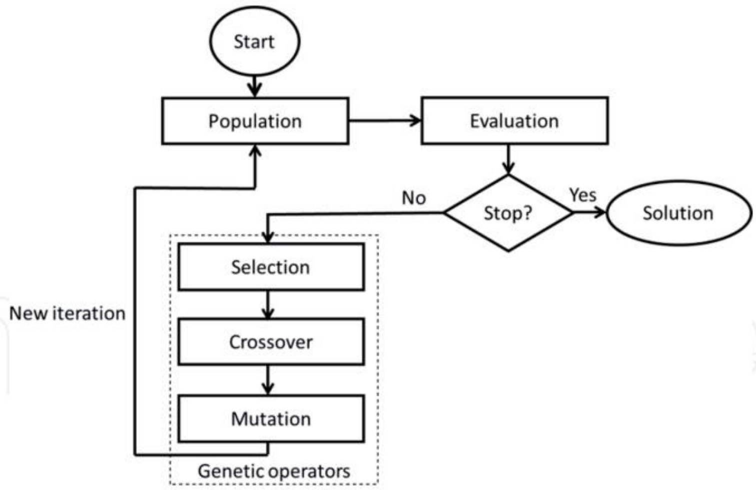


Figure 7: Flowchart of the optimization procedure based on a genetic algorithm. (Gharsalli, 2022)

However, because the multi-compartment assignment problem in this study is related to bin-packing, the Exon shuffling crossover approach is adapted for the crossover operator and is discussed as follows:

Exon Shuffling Approach:

Because of the varying lengths of the chromosomes, an exon shuffling crossover is commonly used in genetic algorithms for bin-packing problems. It is a molecular genetics technique that involves the manipulation of genetic information within and between chromosomes using recombinant techniques. Exon shuffling, as opposed to traditional crossover, uses a two-phase crossover to generate offspring. During the crossover process, two parents are chosen, sorted, and combined to produce offspring (Dokeroglu and Cosar, 2014). To simply explain, each parent chromosome represents the bin with the two sub-string of remaining space and the weight of the assigned item to each bin (Figure 8). To begin the operation, say, with a bin size of 100, two parents are chosen from the population. The chromosomes of two parents are then combined, and the chromosome is sorted in increment order based on the remaining space. In phase 1, each gene in the combined chromosome contains sub-strings containing the assigned item and an unassigned item, which are referred to as the remaining items (Figure 9). In phase 2, the current gene on the chromosome is evaluated and chosen based on the fitness value (the least amount of remaining space), and the bin is then added to the offspring, with the next gene chosen only if all items in its sub-string have not been assigned to any bin in the offspring. The process is repeated until the last gene of the combined chromosome is reached. The new bin is activated for the remaining item, and the items are assigned to the bin using any bin packing algorithm, such as First-fit (FF) and Best-fit (BF), before adding the bin to the offspring. This procedure is repeated until all of the items have been assigned (Rohlfshagen and Bullinaria, 2007).

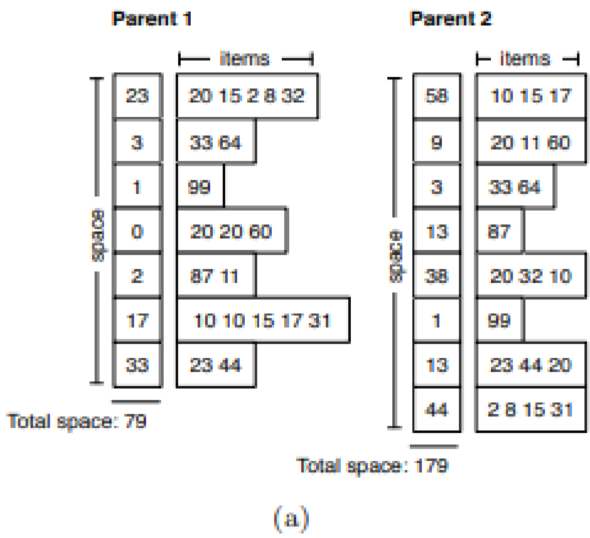


Figure 8: Example of 2 parent Chromosome in Bin packing (Rohlfshagen and Bullinaria, 2007).

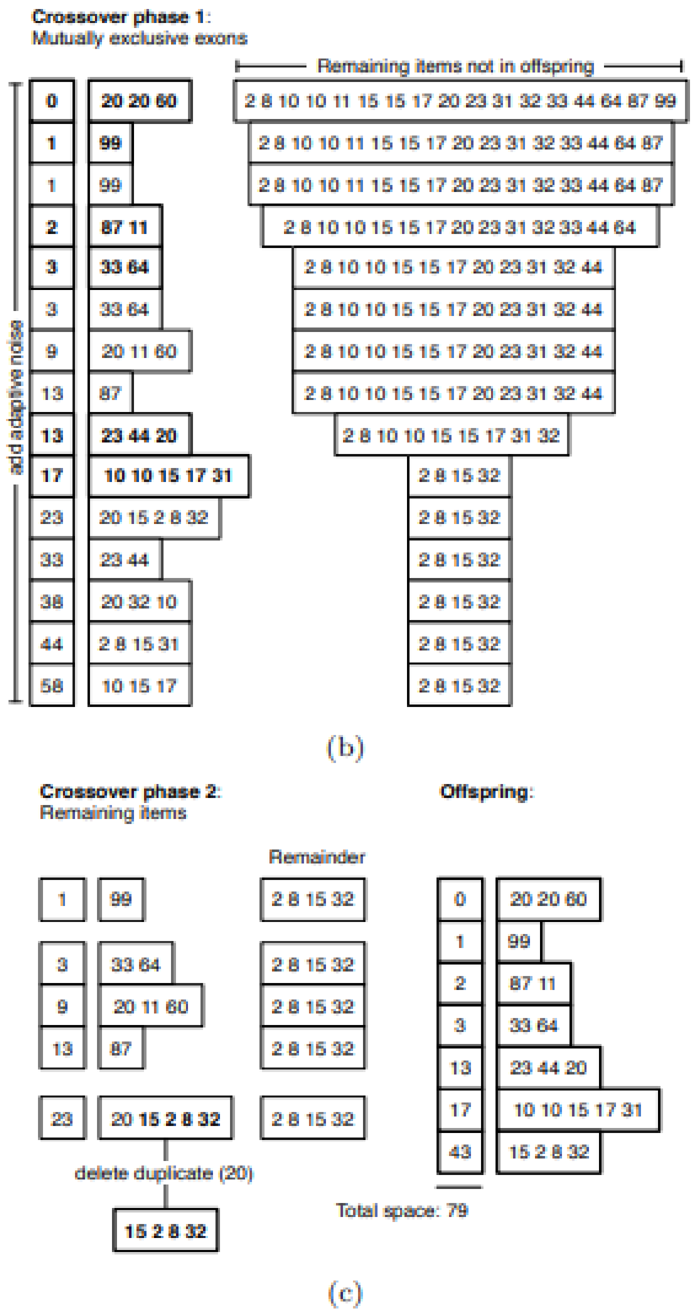


Figure 9: Example – Two-Phase Exon Shuffling crossover (Rohlfshagen and Bullinaria, 2007).

Accordingly, numerous multi-objective evolutionary algorithms have been developed based on genetic algorithms such as MOGA proposed by Fonseca and Fleming (1993), NPGA by Horn et al. (1994), and NSGA suggested by Srinivas and Deb (1995). These three algorithms shared two common features which are the selection process and diversity preservation among the solutions of the same Pareto front. In the selection process, the fitness value is assigned to the population member based on a non-dominated sorting operation (Deb et al., 2002).

The non-dominated sorting operation is primarily used to sort solutions in the population using the Pareto dominance concept, which is essential for selection operations and reflects the population's solution quality. To simply explain, in non-dominated sorting, solution A is said to dominate solution B in population only if there is no objective value of solution A is worse than solution B and there is at least one objective of solution A is better than objective B. Suppose there are solutions of population. P, and is assigned to K Pareto fronts F_i where $I = \{1 \ldots K\}$. All non-dominated solutions are first selected from population P and assigns to F_1 before being temporarily excluded from the population. The non-dominate sorting process then selects a second set of non-dominated solutions from the remaining solution and assigns them to F_2. The process repeats until all individuals are assigned to each front set (F_i) (Bao et al., 2017).

The non-dominated sorting genetic algorithm II (NSGA II) is proposed by Deb et al. (2002) to alleviate the difficulty that occurs in the prior study such as the complexity of computational time, non-elite selection approach, and maintaining the spread of solution in the search space. The original NSGA employed a sharing function approach to maintaining the diversity of the solution in the Pareto front. The sharing parameter is preselected by the user which directly impacts the performance of the algorithm in terms of preserving the diversity of the solution (Deb et al., 2002). In NSGA II, three major operations make NSGA II outperform other MOGA. First, the non-dominated sorting approach is improved by using the ranking approach, called a fast non-dominated sorting approach. The rank is assigned to each group of non-dominated individuals, and the process is repeated until the entire population has been assigned. Following that, the fitness value is assigned. A fast non-dominated sorting approach results in reducing the sorting computation complexity. Following that, a crowded-comparison operation is proposed to preserve the spread of solutions on the Pareto front. When the two solutions from different non-dominated ranks are compared, the solution from the lower rank is selected. On the other hand, if both solutions come from the same rank, the solution from a less crowded region is simply preferred. Lastly, the Elitism selection procedure is proposed to generate the next population from the given population and its offspring. The combination (R_t) of the current parent population (P_t) and its offspring (Q_t) are sorted based on the non-dominated sorting approach. With $2N$ population size, the set of F_1 is a non-dominated set containing all non-dominated individuals. All individuals from F_1 are selected if the size of F_1 less than or equal to N. Else, the individuals from other sets of sorted non-dominated ranks (i.e. $F_1, F_2 \ldots\ F_l$) are chosen using the crowded-comparison operator in descending order. This procedure continues until the solution fills the empty slot in the new population set and the size of the new population equals N as shown in Figure 10 (Benyoucef and Xie, 2011).

<u>Fast Nondominated Sorting Approach:</u>

Fast non-dominated sorting requires the calculation of two entities for each solution, which are described below (Deb et al., 2002) and illustrated in Figure 11:

1. Domination count (n_p): the number of solutions that dominate solution p.
2. Set of solutions that p dominates (S_p)

Suppose population P has k Pareto fronts ($f_1, f_2, \ldots, f_k$), the following steps are used to identify Pareto fronts from 1 to k :

<u>Step 1</u>: n_p= 0 for all solutions p in the first nondominated front

<u>Step 2</u>: Visiting each member (q) of its set $S_p$and reduce its dominant count by one (i.e. $n_p \coloneqq n_p - 1$ for any $q \in S_p$)

<u>Step 3</u>: Repeat steps 1 and 2 until all solutions in P are assigned.

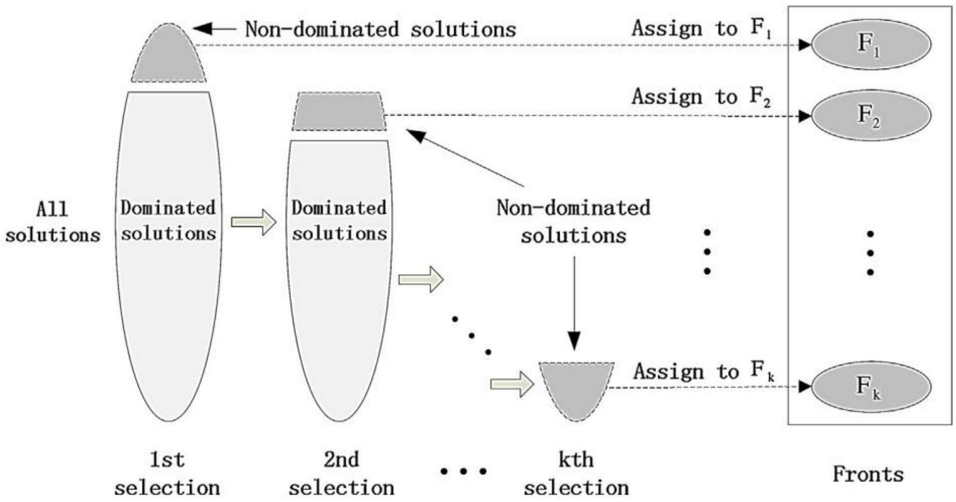


Figure 10 : Non-dominated sorting process of solutions in a population (Bao et al., 2017)

$\underline{\texttt{fast-non-dominated-sort}(P)}$	
for each $p \in P$	
$S_p = \emptyset$	
$n_p = 0$	
for each $q \in P$	
if $(p \prec q)$ then	If p dominates q
$S_p = S_p \cup \{q\}$	Add q to the set of solutions dominated by p
else if $(q \prec p)$ then	
$n_p = n_p + 1$	Increment the domination counter of p
if $n_p = 0$ then	p belongs to the first front
$p_{\text{rank}} = 1$	
$\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$	
$i = 1$	Initialize the front counter
while $\mathcal{F}_i \neq \emptyset$	
$Q = \emptyset$	Used to store the members of the next front
for each $p \in \mathcal{F}_i$	
for each $q \in S_p$	
$n_q = n_q - 1$	
if $n_q = 0$ then	q belongs to the next front
$q_{\text{rank}} = i + 1$	
$Q = Q \cup \{q\}$	
$i = i + 1$	
$\mathcal{F}_i = Q$	

Figure 11 : Fast Nondominated sorting Algorithm (Deb et al., 2002)

Later, Deb and Jain proposed NSGAIII, an improved non-dominated sorting genetic algorithm II (NSGA II), in which the solution diversity is preserved through the use of a reference points-based approach, with one individual associated with one reference point (Cui et al., 2019). Accordingly, NSGA III is chosen and used in this study, which will be discussed in greater detail in the following section.

2.8 The non-dominated sorting Genetic Algorithm III (NSGA-III)

Non-dominated sorting Genetic Algorithm (NSGA-III) is an evolutionary multi-objective optimization (EMO) methodology using reference-based points to find a set of well-converged and well-diversified non-dominated solutions. The algorithm is adapted from the basic framework of NSGA-II with a notable change. In contrast to NSGA-II, maintaining diversity among population members in NSGA-III is facilitated by providing and iteratively updating a number of widely distributed reference points. In contrast to NSGA-II, maintaining diversity among population members in NSGA-III is facilitated by providing and iteratively updating a number of widely distributed reference points. To encapsulate the NSGA-III general framework (Figure 12), the set of K reference points is first generated, and the initial population is generated at random. Consider the population of the 't' generation denoted as P_t with size N and the offspring (Q_r) are obtained by two parent selections through crossover and mutation operations. The combination of the current population and off-spring denote as ($P_t \cup Q_r$, with size 2N). To choose the best N population, the parents and off-spring are combined, then sorted according to different non-dominated levels (i.e., $F_1, F_2 \ldots\ F_l$). Each non-domination rank is then selected one at a time to construct a new population. Instead of selecting from crowding distance, predefined the set of reference points is used to ensure diversity in obtained solutions in NSGA-III and the selected new population is denoted as S_t (Cui et al., 2019).

Algorithm 1 Generation t of NSGA-III procedure

Input: H structured reference points Z^s or supplied aspiration points Z^a, parent population P_t

Output: P_{t+1}

$S_t = \emptyset$, $i = 1$
$Q_t =$ Recombination+Mutation(P_t)
$R_t = P_t \cup Q_t$
$(F_1, F_2, \ldots) =$ Non-dominated-sort(R_t)
repeat
    $S_t = S_t \cup F_i$ and $i = i + 1$
until $|S_t| \geq N$
Last front to be included: $F_l = F_i$
if $|S_t| = N$ **then**
    $P_{t+1} = S_t$, break
else
    $P_{t+1} = \cup_{j=1}^{l-1} F_j$
    Points to be chosen from F_l: $K = N - |P_{t+1}|$
    Normalize objectives and create reference set Z^r: `Normalize`$(\mathbf{f}^n, S_t, Z^r, Z^s, Z^a)$
    Associate each member $\mathbf{s}$ of S_t with a reference point: $[\pi(\mathbf{s}), d(\mathbf{s})] =$`Associate`$(S_t, Z^r)$ % $\pi(\mathbf{s})$: closest reference point, d: distance between $\mathbf{s}$ and $\pi(\mathbf{s})$
    Compute niche count of reference point $j \in Z^r$: $\rho_j = \sum_{\mathbf{s} \in S_t/F_l} ((\pi(\mathbf{s}) = j)\ ?\ 1 : 0)$
    Choose K members one at a time from F_l to construct P_{t+1}: `Niching`$(K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})$
end if

Figure 12: General framework for NSGA-III (Deb and Jain, 2013)

The detailed procedure of NSGA III is discussed as follows (Deb and Jain, 2013):

a. *Population Classification to Non-dominated level*

Classified the combination of population and offspring into Non-Dominated Levels. After a non-dominated sorting procedure, if all new population members $(|S_t| \leq N)$, there are no further operations required. On the other hand, only members from $(F_1, F_2 \ldots\ F_{l-1})$ is added to the new population if $|S_t| \geq N$, some members from F_l are then selected to fulfill the remaining slot $(K = N - S_t)$.

b. *Reference Point Generation*

The set of reference points on a hyperplane is predetermined to ensure the diversity of the obtained solutions. Das and Dennis's systematic approach is employed to predefine the reference point for the normalized hyperplane. A reference direction is composed of a vector that starts at the origin and connects to each of them. The total number of reference points (H) for M – objectives problem is calculated by:

$$H = \binom{M+p-1}{p} \tag{2-28}$$

where

p is the divisions considered along each objective.

M is the number of objectives

For example, in figure 13, in a three-objective problem where $M = 3$, the triangle is created on the hyperplane with x-,y-,z- coordinates of (1,0,0),(0,1,0) and (0,0,1) as the apex. The number of divisions is recommended to be more than the number of objective problems. In this case, with four divisions ($p = 4$) chosen, $H = \binom{3+4-1}{4} = 15$ or 15 reference points. For further clarification, the supplied set of reference points is purposely generated to find the near Pareto-optimal solutions corresponding to the reference points ensuring the diversification of the solutions.

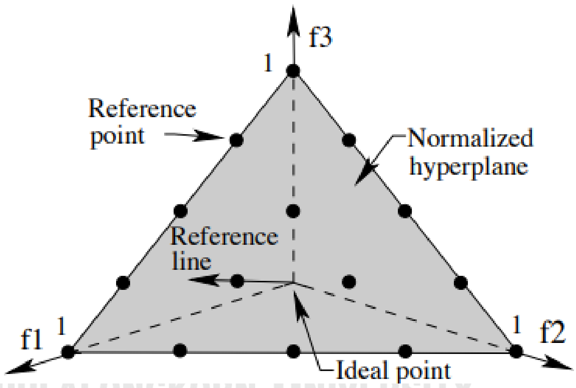


Figure 13: 15-reference points on a normalized reference plane for three objective problems with p =4

(Deb and Jain, 2013)

c. *Adaptive Normalization*

The normalization procedure is used to scalarize the fitness value of each objective (Bi and Wang, 2017). To normalize the values, the ideal point of the population (S_t) is first determined by identifying the smallest value of each objective function $i = \{1, \dots, M\}$ which results in a set of ideal points for each population, denoted as $\overline{Z} = (Z_1^{min}, Z_2^{min}, \dots, Z_M^{min})$. Next, the translated objective ($f_i'(\mathrm{x})$) is introduced. The translated objective is calculated by subtracting each objective value $f_i(x)$ by the minimum value of each objective (Z_i^{min}).

$$f_i'(\mathrm{x}) = f_i(x) - Z_i^{min} \tag{2-29}$$

The extreme M points on M objectives axis are then obtained from the achievement scalarization function (ASF) with minimum weight vector (w) where the value is a small number of 10^{-6} and the equation is determined as followed. In addition, the weight vector w denotes the axis direction.

$$ASF(x, w) = \max_{i=1}^{M} \left(\frac{f_i(x) - z_i}{w_i} \right) \tag{2-30}$$

The normalized objective can be obtained from the following equation:

$$f_i^n(x) = \frac{f_i'(\mathrm{x})}{a_i - Z_i^{min}} \tag{2-31}$$

where a_i denote as the intercept of the hyperplane with axis directions and $(a_1, a_2, \dots, a_i)$ is simply set to $(Z_1^{max}, Z_2^{max}, \dots, Z_i^{max})$.

d. The Association Operation

In association operation, with normalized objective value, each point in the objective space is connected to the reference point through the reference line. The reference line is constructed by connecting each reference point to the origin of the hyperplane. The perpendicular distance of each population member in objective space to each reference line is then calculated. This operation is illustrated in Figure 14.

After perpendicular distances are calculated, the reference point with the closest reference line to the population member is associated with the population member. In the other words, the minimum distance between reference points and population member point are determined in normalized objective space.

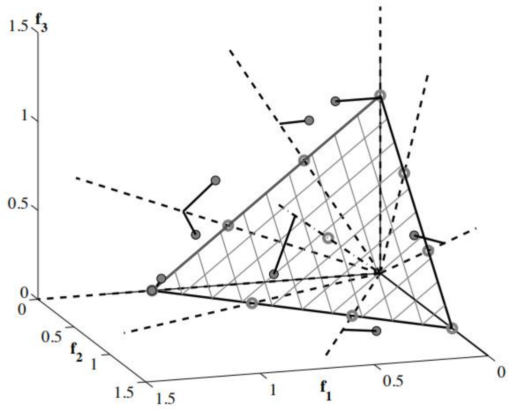


Figure 14: Illustration of population members associated with the reference points

(Deb and Jain, 2013)

The projection distance on the reference line ($d_{j,1}$) and the perpendicular distance between the reference points and the perpendicular point ($d_{j,2}$) are calculated as follows (Bi and Wang, 2017):

$$d_{j,1}(x) = \frac{\|f^n(x)^T w^j\|}{\|w^j\|} \tag{2-32}$$

$$d_{j,2}(x) = \left\| (f^n(x) - dj,1(x)) \frac{w^j}{\|w^j\|} \right\| \tag{2-33}$$

where

$d_{j,1}(x)$: The projection distance on the reference line from the origin point to the intersection point of the reference line that is associated with the objective value in the objective space.

$d_{j,2}(x)$: The perpendicular distance between the objective value in the objective space and the reference line

$f^n(x)$: The normalized objective vector for individual x

$f^n(x)^T$: The transposition of a normalized objective vector

w^j : Reference points

$\| \quad \|$: The size of a vector

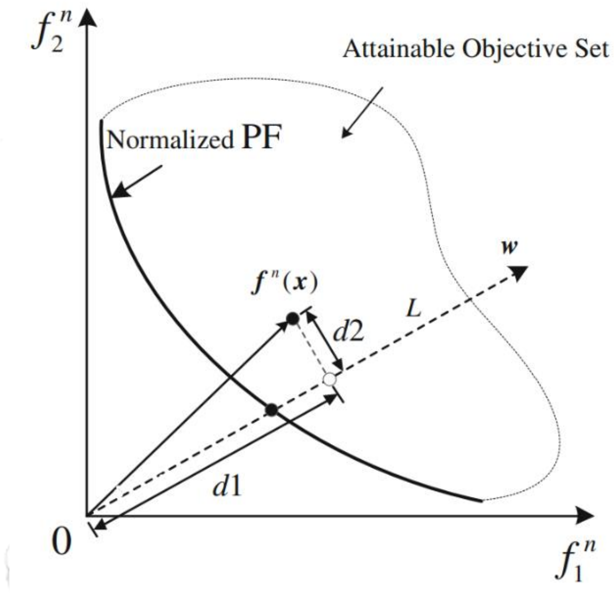


Figure 15: Illustration of distances d1 and distance d2 in 2-dimensional space (Bi and Wang, 2017)

e. Niche Preservation Operation

The niche preservation operation counts the reference points that are associated with the population member is called *'Niche Operation'*. All members from the $(F_1, F_2 \ldots F_{l-1})$ are automatically added to the new population (P_{t+1}).

- Let niche count be denoted as ρ_j for the *j-th* sub-region. Niche count is firstly counted from current (P_{t+1}) for each reference point. The remaining slots are filled by selecting members from F_l
- Next, the set of the reference point with minimum niche count (ρ_j) are sorted and listed. If there are multiple reference points with the same niche count, one reference point is randomly selected.

- If $\rho_j = 0$, there can be two scenarios.
  - Scenario 1: There are members from F_l associate with the selected reference point. Hence, the nearest perpendicular distance is added to P_{t+1} .
  - Scenario 2: There is no member from F_l associate with the selected reference point. Hence, the member from F_l is chosen at random and added to P_{t+1}
- If $\rho_j \geq 1$, this means that the reference points already associate with population members. In this case, the member from F_l is chosen at random and added to P_{t+1}

- Niche counts are updated and repeated for a total of K times to fill the remaining slot.
- P_{t+1} is completely updated for a generation 't' th

2.9 Performance Metrics of Multi-Objective Optimization

One prominent issue in algorithms development for multi-objective optimization is the algorithm's quality assessment. In multi-objective optimization problems, numerous evolutionary algorithms (EAs) are proposed to search for a solution set that is an approximation to the Pareto-optimal front. Accordingly, the techniques that give the best approximation value for a given problem are concerned. To identify the most promising techniques, performance metrics are taken into account. The conception of performances is divided into two perspectives which are the quality of the outcome and the computational time in generating the outcome (Zitzler et al., 2003).

In terms of computational time, tracking either the total run time or the number of fitness evaluations is a general procedure on a particular computer. As an outcome, there is no difference in computational time between single and multi-objective optimization. However, there is a distinction between single-objective optimization and multi-objective optimization in terms of quality. In a single objective problem, the solution quality is simply generated by comparing the value whether it is larger or smaller regards the objective function (Zitzler et al., 2003).

In multi-objective problems, even though Pareto-optimality may be sufficient to conclude that one approximation front is superior to another in some cases, the most common case is that fronts are mutually incomparable. As a result, a quality assessment is required to quantitatively evaluate the outcome. It is, however, difficult to define the appropriate quality measurement for the Pareto optimal set. Quality measurement is introduced to quantitatively compare the outcomes of multi-objective optimization. Quality metrics are commonly used to describe quality measurement. Convergence, spread, and distribution are three common features that indicate the algorithm's quality. Over the last few decades, many quality metrics have been proposed in the literature, and multiple quality metrics are frequently used concurrently because each metric assigns different importance to these three features. And each feature is described in detail below (Bezerra et al., 2017):

1. **Convergence:** refers to the optimality of the solutions. To be said, the solutions in the Pareto front are concluded to be converged.
2. **Spread:** refers to the distance between the optimal point in the Pareto front
3. **Distribution:** refers to the evenness of the Pareto front. Describe the uniformity of the distance between pairs of solutions.

There are several common quality metrics such as generational distance (GD), inverted generational distance (IGD), hypervolume (HV), Spread (Δ), Averaged Hausdorff distance (Δp), and R2-indicator (Santos and Xavier, 2018). Inverted Generational Distance (IGD) will be discussed further in this section because it is widely recognized as a reliable performance for concurrently quantifying the convergence and diversity of many-objective evolutionary algorithms (EMOs) (Sun et al., 2019).

Inverted Generational Distance (IGD):

The Inverted Generational Distance (IGD) is viewed as an approximate distance between the Pareto front and the solution set in objective space (Santos and Xavier, 2018). The IGD is an improvement of Generational Distance (GD) which had been proposed two decades ago. The GD is the average Euclidean distance between each solution point of individual a in a given approximation Pareto front A (or non-dominated solutions generated by MOEAs) and the reference point r in a reference front R and is calculated as follows (Bezerra et al., 2017):

$$GD(A,R) = \frac{1}{|A|}\left(\sum_{a\in A} min_{r\in R}\, d(a,r)^p\right)^{\frac{1}{p}} \quad (2\text{-}34)$$

$$d(a,r) = \sqrt{\sum_{k=1}^{M}(a_k - r_k)^2} \quad (2\text{-}35)$$

where

A : a set of approximation Pareto front (solution set)

a : an individual in the set of approximation Pareto front

R : a set of reference point

r : an individual in the set of reference points

$d(a,r)$: Euclidean distance between each solution point 'a' to each reference point 'r'

It should be noted that p = 2 was used in the original proposal, but this was later replaced by p = 1 for ease of interpretation and computation. However, the GD is claimed to be sensitive to the size of obtained solution (or Pareto Front). Hence, the IGD was proposed and is discussed as follows (Bezerra et al., 2017):

$$IGD(R,A) = \frac{1}{|R|}\left(\sum_{a\in A} min_{r\in R}\, d(r,a)^p\right)^{\frac{1}{p}} \quad (2\text{-}36)$$

$$d(r,a) = \sqrt{\sum_{k=1}^{M}(r_k - a_k)^2} \quad (2\text{-}37)$$

where

A : a set of non-dominated solutions generated by the algorithm.

a : an individual in the set of approximation Pareto front

R : a set of reference point

r : an individual in the set of reference points

$d(r,a)$: Euclidean distance between each reference point 'r' to each solution point 'a'.

In comparison to the GD, the IGD is not sensitive to the size of the approximation Pareto Optimal Front and It provides a ranking that subjectively best fits the desired convergence, spread, and distribution more accurately. The IGD then becomes the common metric for assessing the quality of multi-objective EMO algorithms (Bezerra et al., 2017). To quantify the quality of algorithms, a smaller value of IGD implies how better the approximation of the Pareto Front convergence toward the Pareto-optimal front (reference points) is and indicates how uniform of solution distribution is (Zhang et al., 2008). It is not, however, necessary to imply that lower IGD is preferable. The IGD value will be high if no solution associated with a reference point is found (Deb and Jain, 2013).

Apart from the three common quality measurements, counting the number of solutions is known as a direct performance measurement of a non-dominated sorting algorithm (Fang et al., 2008). The variation in population size has a significant impact on the algorithm's performance. Small populations have poor convergence, whereas large populations have high computational complexity in finding high-quality solutions. Selection of an appropriate population size for the algorithm to obtain a good performance is therefore challenging (Rosenthal and Borschbach, 2014). To set a proper initial population size, Reed et al. (2003) proposed using the empirical rule of thumb to estimate the population size given in equations 2-38.

$$N_0 \approx 2R_{ND} \tag{2-38}$$

where

N_0 : initial population size

R_{ND}: Approximated number of nondominated solution

The approximated number of non-dominated solutions (R_{ND}) is normally equal to 30, which is approximately half of the theoretical maximum number of nondominated solutions that could exist. The first population size normally starts with N_0 and the population size is then doubled for the next run (Reed et al., 2003). According to Reed et al. (2003), in the experiment, the population size started at 60 and then doubled to 120, 240, and 480 respectively. As a result, the number of solutions improved at a decrement rate as the population size increased. For example, at the population size of 480, the number of non-dominated solutions is reported to be increased by one solution which represents only a 3% change. While the number of the non-dominated solution increased by 26% when doubled the population size from 60 to 120.

2.10 Related Literature

Various models and policies for exporting perishables commodities from Thailand are studied. For example, the time-cost model and qualitative decisions were employed to examine six alternative international routes from Thailand to China, collecting data from surveys and in-depth interviews and all information-related routes from port/border to port/border (Krebs and Panichakarn, 2019). As a result, the optimal path is determined by stakeholders' decision criteria and conditions. In general, sea routes are much less expensive than road transportation. However, in addition to the shipping time, numerous other issues must be considered, such as long customs processes and the possibility of product quality degradation.

Furthermore, the beverage transportation model from Thailand to Cambodia has been studied to minimize cost, time, and risk. The time-cost model was applied to determine the duration and expense of each route. In addition, the qualitative technique is utilized for risk assessment by interviewing specialists or logistic service providers. The in-depth systemic analysis and prioritization of decision criteria have been conducted by identifying risk variables and giving a weight scale. A zero-one-goal programming strategy is applied to solve the optimal path selection problem (Kaewfak et al., 2021).

In addition, various researchers have developed mathematical models and algorithms to solve logistic problems in an agricultural setting. For example, a multi-objective model has been developed for the Thai sugar industry (Jarumaneeroj et al., 2021). The model utilizes multi-objective mixed integer programming to optimize stakeholders' objectives (e.g., economic objectives and environmental impact). In addition, Particle Swarm Optimization (PSO) is applied to find an optimal solution. Results show that maximizing sugar production volume may not optimize the entire sugar supply chain objectives. On the other hand, scarifying some sugar production volume may improve the profitability for all stakeholders.

Many quantitative model approaches have been developed concerning fresh fruit degradation, such as the optimization model for freshness loss for managing fresh fruit exportation (Damrongpol and Pradorn, 2016). For instance, a linear programming method is proposed to enhance agricultural-produced transportation efficiency with customer satisfaction (Ali and VanOudheusden, 2009). The LP model with a single objective considering total revenue, logistic cost, and freshness losses cost is constructed to solve the optimal route for fruit exportation under different circumstances, such as temperature and transportation time. And the freshness loss is determined in terms of weight loss cost. As a result, the approach enables the fruit exporters to plan and operate the fruit exportation in production and distribution planning. However, this method can only apply to single-product distribution (Nakandala et al., 2016).

The multi-product food transportation cost function is generated to examine the situation in which a truck collected a specific product from multiple farm locations. The goal is to investigate the effects of different initial quality levels of various food product types with varying rates of quality deterioration stored in a single trip. The approaches, including a genetic algorithm (GA), fuzzy genetic algorithm (FGA), and improved simulated annealing (SA) are adopted, and all performances are evaluated. As a result, the fuzzy genetic algorithm (FGA) is more likely to produce better results than the other two approaches (Nakandala et al., 2016).

In addition, several works have applied the vehicle routing problem (VRP) to perishable product distribution. For instance, The spoilage rate of perishable commodities for cold chain distribution is studied, and a vehicle routing optimization model considering spoilage time was proposed (Verbic, 2006). The refrigerated and general-type vehicles are explored and compared in a vehicle routing problem for perishable food product delivery. The non-linear mathematical model and heuristic approaches are designed to generate efficient vehicle routings to increase customer satisfaction based on food's freshness (Song and Ko, 2016). The bi-objective with mixed-integer linear programming (MILP) is constructed to solve vehicle routing problems (VRP) for perishable products under uncertainty. The study aims to analyze the effects of transportation time variability and shelf-life on optimality and feasibility. The model's objectives are to minimize the total cost and maximize the total weighted freshness of the delivered products. The findings indicate that ignoring the perishability of the products results in an unfavorable situation regarding the freshness of delivered products (Rahbari et al., 2019).

The model for vehicle routing problems with time windows and time-dependent travel times (VRPTW-TD) considering the impact of perishability is developed. The model aims to minimize the total distribution cost where the quality loss is included as a part of the objective function. The loss of quality is time-dependent and represented in quantity loads (kgs). The heuristic approach based on tabu search is applied to solve the problem. The Slovenian vegetable market distribution network is used for algorithm performance validation. Compared to the traditional model, where the loss of quality is not considered, the proposed model results in a 27.9% reduction in quality loss (Osvald and Stirn, 2008).

The freshness of products has also been viewed as customer satisfaction. The multi-objective vehicle routing problem with time windows dealing with the time-sensitivity spoilage rate of perishable products is proposed. The goals are to minimize total costs, including fixed, transportation, penalty, and damaged costs, and to maximize the average freshness of the products' remaining shelf life. The problem is solved using a two-phase heuristic algorithm based on the Pareto variable neighborhood search genetic algorithm with temporal-spatial distance (STVNS-GA). Computation results demonstrate the algorithm's effectiveness and feasibility; it significantly improves solution quality and accelerates algorithm convergence compared to a genetic algorithm (Wang and Zhan, 2016).

However, work related to multi-compartment vehicles in perishable distribution is quite scarce. A flexible multi-compartment truck with vertical loading is recommended for grocery delivery. The problem is defined as two-dimensional loading and multi-compartment vehicle routing (2L-MCVRP). The objectives are to find an effective multi-compartment loading with appropriate distribution pathways. A branch-and-cut (B&C) method and a large neighborhood search (LNS) heuristic are employed to produce optimal routing for a multi-compartment truck with multiple product segments (Ostermeier and Hübner, 2018).

In addition, vehicle selection in multi-compartment vehicle routing problems (MCVRP) for a refrigerated vehicle is studied in grocery distribution. Transportation, loading, and unloading costs are vehicle-dependent based on vehicle type. The objective is to reduce the total cost by selecting the tour, selecting a vehicle for each trip, and assigning products to each compartment. The saving algorithm is adopted to construct an initial solution. The large neighborhood search, Shaw removal, and regret-k insertion approaches are employed as improvement methods. They conclude that operating a mixed fleet is preferable to perform a single-compartment or a multi-compartment to deliver goods. As a result, transportation costs can be lowered by 30%, maximum, by using a diverse fleet (Ostermeier and Hübner, 2018).

With the advantages of multi-compartment vehicles, numerous studies have focused on grocery delivery tours using a multi-compartment refrigerated truck under the fixed or flexible area size for each product segment. The multi-compartment vehicle problem (MCVRP) is either generally viewed as the integration of capacitated vehicle routing (CVRP) (Yahyaoui and Dekdouk, 2018). MCVRP is renowned as a multi-compartment vehicle problem with a time window (MCVRP-TW). However, there is no general algorithm for the optimal global solution in multi-compartment vehicle problems with a time window. Hence, several heuristics approaches such as Lagrangian relaxation, memetic algorithm, colony algorithm, tabu search, large neighborhood search (LNS), 2-opt, and Clarke wright have been employed to solve a feasible solution (Chen and Shi, 2019).

The most popular modeling approaches for the optimization of distribution in agricultural products are mixed-integer linear programming (MILP), followed by multi-objective linear programming (MOLP), and stochastic programming (SP). However, not all research has concentrated on reducing food loss due to deterioration during shipment (Paam et al., 2016). The literature is divided into single-objective and multi-objective functions shown in Table 2.

Author	Objective function		Product Type
	Single Objective	Multi-objective	
Lu et al. (2006)	Maximize profit for perishable		Perishable
Cholette (2017)	Maximize the weighted volume of all wine		Wine/USA
Osvald and Stirn (2008)		Minimize distance, time traveled, delay cost, and perishability cost	Vegetable/ Slovenia
Arnaout and Maatouk (2010)	Minimize the operational costs and the loss of quality		Grape / Lebanon
Ahumada and Villalobos(2011)	Maximize the revenues of the grower		Tomato/USA
Paksay et al. (2012)		Minimize transportation costs between suppliers and soils and minimize transportation costs between suppliers and warehouses	Vegetable oil / turkey
Zanoni and Zavanella(2012)	Minimize total cost		Frozen fried potato
Ampatzidis et al. (2014)	Maximize production output and fruit quality		Grape, Cherry/ Greece, USA

Table 2: Literature survey in agricultural product distribution (Paam et al., 2016)

Author	Objective function		Product Type
	Single Objective	Multi-objective	
Govindan et al. (2014)		Minimize logistic cost and minimize environmental impacts of CO2 emissions	Perishable
Amorim and Almada-Lobo(2014)		Minimize the distribution costs and maximize the freshness state if the delivered products	perishable
Soysal et al.(2015)	Minimize total cost		Tomato/turkey
Botoloni et al. (2016)		Minimize the operating cost, minimize the carbon foot print and minimize delivery time	Fruits and vegetable / Italy

Table 2: Literature survey in agricultural product distribution (Paam et al., 2016) (Cont.)

In several studies, carbon emissions have become a primary concern regarding global warming. For example, the optimization model is constructed to solve refrigerated trucks' load-dependent vehicle routing problem. The fuel consumption and carbon emission related to vehicle load are considered for model development (Stellingwerf et al., 2018). A hybrid genetic algorithm with heuristics algorithms to solve fresh food distribution problems is developed considering a low-carbon point of view. The algorithm effectively provides an environmentally friendly route for cold chain distribution (Wang et al., 2018). The cyclic evolutionary genetic algorithm (CEGA) is employed to solve cold chain vehicle routing problems to minimize customer satisfaction and total operation cost, including carbon emission costs. The results obtained from numerical experiments are cost-effective. The experimental results show that as carbon prices rise, total costs follow two opposing trends, depending on whether a carbon quota is sufficient. Increasing the price of carbon within a specific range can effectively reduce carbon emissions, but it will also reduce average customer satisfaction to some extent (Qin et al., 2019).

Additionally, the non-dominated sorting genetic algorithm III (NSGA-III) has become a popular multi-objective optimization algorithm to solve complex optimization problems in various fields such as engineering design, and transportation. For example, it has been used to solve several problems in the airforce industry such as solving the many-objective cockpit crew pairing problem of the low-cost airline (Krisanaphan, 2019) and the multi-objective airport gate assignment problem (AGAP) for Suvarnabhumi airport (Khaowong, 2021). Furthermore, the non-dominated sorting genetic algorithm III (NSGA-III) was also applied to solve the problem related to workers in production line allocation such as the worker allocation problem on multiple cellular u-shaped under many objectives (Chimrakhang,2019).

Hence, this chapter summarizes the previous works and theories related to distribution network design, perishable products' decay rate, and carbon emission from refrigerated trucks. This thesis applies the shortest path, bin packing, and vehicle loading problems to construct the mathematical model and heuristic approach to solve the multi-compartment vehicle loading with route selection problem.

CHAPTER 3

METHODOLOGY

3.1 Problem Description

A multi-compartment vehicle loading, and transportation routing problem (MCVRLP) is an integration of the shortest path and bin packing problem. This study considers the transportation of fresh fruits and vegetables as a case regarding their varied harvesting period and diverse characteristics. The various characteristics necessitate different requirements to extend their shelf life and slow the rate of spoilage. Because of the large volume of fresh fruit shipped across the continent, fresh fruit is frequently placed in a refrigerated container with a single temperature for shipping (Chen et al., 2019 2018). As fresh fruits and vegetables are considered seasonal products, shipment consolidation with different temperatures is expected to be more cost-effective than splitting the delivery into several vehicles during the low season.

In multi-compartment vehicle loading, and transportation routing problems, suppose there are a set of identical vehicles $(v \in V)$ with up to three vertical compartments (c) and set of the types of fresh fruits and vegetables$(f \in F)$, the layout type of vehicle (t) is selected for fresh fruits and vegetable assignment. The weight of fresh fruits and vegetables (y_{vtcf}), in kgs, are determined and assigned to the selected compartment on the activated vehicle with layout type (a_{vt}). In addition, the assigned weight of fresh fruits and vegetables on each activated vehicle (y_{vtcf}) cannot exceed the vehicle compartment capacity (cap_{ct}). While one type of fresh fruit and vegetables can be assigned to more than one vehicle, no two or more types of fresh fruits and vegetables can be mixed in the same compartment. The selection of compartment layout and the allocation of fresh fruits and vegetables in each vehicle is related to decision-making because it affects the route selection of each vehicle. Finally, the allowable transportation time is determined by sorting the shelf-life of fresh fruits and vegetables (SL_f) in each vehicle.

With a given $G = (N, E)$ where N is the set of transit cities and E is the set of edges connecting different cities associated with transportation time (tt_{ij}), handling time (ht_j), transportation cost (tc_{ij}), handling cost (hc_j) and distance (d_{ij}), the transportation route is selected under the restriction of quality loss. In this study, the shipment quality is assumed to be 100% at the beginning of the distribution and the quality starts to decay as the distribution begins. The quality loss percentage of the shipment is set not to exceed 23%, where the maximum allowable time is derived from equation (2-5) with a constant decay rate (φ_p) of 0.0012 hour. Hence, the set of maximum allowable times, $Tmax_K$ where $K = \{1,2, \dots k\}$ is an index number for the quality and time set. The set of permissible transportation times is calculated and shown in table 3. However, the desire for overall shipment quality selection is restricted by the actual fresh fruits and vegetable shelf-life of each vehicle (SL_f). Consequently, the maximum allowable time does not exceed the product's shelf life within the vehicle.

K	Quality (%)	Maximum allowable total time (hours)($\boldsymbol{Tmax_k}$)
1	92	69.5
2	90	87.8
3	88	106.5
4	86	125.7
5	84	145.3
6	82	165.4
7	80	186.0
8	78	207.1

Table 3: Set of maximum allowable time

The heavy refrigerated vehicle is used for transportation in this study, and diesel fuel is used. Only carbon emissions from vehicle operation and refrigerator cooling units are considered when calculating carbon emissions. According to Bin et al. (Bin et al.), the fuel consumption and carbon emissions factors are shown in Table 4.

Mode	Type	Data	Data Type
Transportation	Heavy truck	0.2857 L/Km	The driving fuel consumption
Carbon emission factor	Diesel	3060g/kg (per kg diesel fuel emits 3060g)	CO2 emission
		2630g/L (per L diesel fuel emits 2630g)	CO2 emission

Table 4: Fuel consumption and carbon emission data
(Bin et al., 2022)

Due to the nonlinearity of the multi-compartment vehicle loading and transportation routing problem, the transformation and piecewise linear approximation approaches are adopted in this study to approximate the optimal value of the mathematical model. The main decisions to be made are the allocation of fresh fruits and vegetables and the selection of transportation routes, intending to optimize the three objectives listed below:

1) Minimize the total transportation cost
2) Minimize the total carbon emission from driving and thermal energy
3) Minimize the total weight loss of fresh fruits and vegetables

Hence, this thesis aims to solve the non-linear optimization problem of fresh fruits and vegetable transportation by effectively finding the optimal distribution route under three conflicting objectives.

3.2 Problem Scope and Assumptions

The problem is formulated under assumptions to obtain the optimal route selection in relation to the proposed mathematical model, and all notation and elements are determined in this section.

For the simplification of the model, several assumptions are made as follows:

1) The number of vehicles available for transportation is assumed to be sufficiently large and identical (same size and capacity).
2) The total vehicle capacity is predefined, and the type of layout is limited to three different types.
3) The partition that separates each chamber is adjustable.
4) The quality of fresh fruits and vegetables is assumed to be in an acceptable condition in the warehouse.
5) The quantity shipped is sold by weight (kgs), so the quality of fresh fruits and vegetables is mixed.
6) The total quantity of each fresh fruit can be split and shipped in a different vehicle.
7) Different types of fruit cannot be mixed in the same compartment due to their specific requirement but can be assigned to any compartment for transportation.
8) Fresh fruits in each shipment are assumed to be transported at specific optimal temperature points.
9) The weight loss rate represents the actual quality loss of each fresh fruit and vegetable in the shipment.
10) The respiration heat rate from fresh fruits and vegetables is constant.
11) The temperature of the air surrounding the vehicle (temperature ambient) during the transportation is constant.
12) The vehicle capacity, fresh fruits, and vegetables are preselected, and the total shipping quantity is randomly generated at random for the mathematical model and heuristic approach validation.

3.3 Mathematical Model Formulation For Multi-compartment Vehicle Loading, and Transportation Routing Problem (MCVLRP)

The proposed mathematical model is non-linear programming that mainly employed the concept of shortest path and bin packing in solving multi-compartment vehicle loading and transportation routing problem (MCVRLP). In this section, the discussion is divided into two main parts which are the mathematical model, and the linearized mathematical model as follows:

3.3.1 Mathematical Model Formulation

3.3.1a Sets and Parameters

Network Elements

- Let $G = (N, E)$ be a graph of the transportation route.
- Let $N = \{0,1,2,\ldots,n\}$ be a set of nodes; nodes represent the transit cities while $\{o\}$ is the origin city and $\{d\}$ is the destination city.
- Let $E = \{(i,j) | i,j \in N, i \neq j\}$ be the set of edges connecting different cities is associated with transportation time (tt_{ij}), handling time (ht_j), transportation cost (tc_{ij}), handling cost (hc_j) and distance (d_{ij}).

Vehicle Types and Their Compartment

- Let $V = \{1,2,\ldots,v\}$ be the set of identical vehicles with a total capacity 'P'.
- Let $T = \{1,\ldots,t\}$ be the set of vehicle layout types (compartments utilization).
- Let $C = \{1,\ldots,|T|\}$ be the set of available compartments.
- A_{ct} is a binary parameter indicating whether compartment 'c' exists in layout 't' (e.g., $A_{12} = 0$)
- cap_{ct} represents the capacity of compartment 'c' in layout type 't'
  - Type I ($T = 1$): A single compartment is utilized for one product type. ($cap_{11} = P$).
  - Type II ($T = 2$): Two compartments are used when shipping two different types of fresh fruit in one shipment. ($cap_{21} = \frac{P}{3}$ and $cap_{22} = \frac{2P}{3}$).
  - Type III ($T = 3$): If the container is utilized in up to three compartments, the capacity of the container is split equally. ($cap_{31} = cap_{32} = cap_{33} = \frac{P}{3}$)

- Type IV ($T = 4$): An inactivated type of vehicle where all capacity is zero.

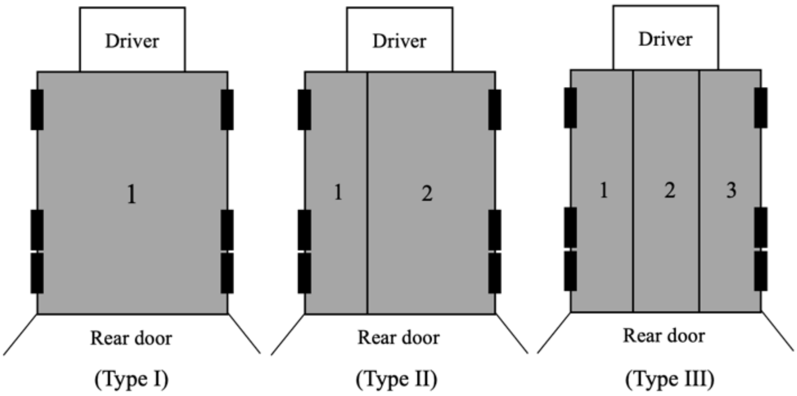


Figure 16: Vehicle Compartment Layout (2D)

- Cost-related to utilizing multi-compartment refrigerated vehicles consist of partition adjustment costs (ADJ_{vt}) and fixed cost of operating a vehicle 'v' (FIX_v)

<u>*Carbon Emissions from Refrigerated Vehicles*</u>

The total carbon emissions consist of carbon emissions from driving and the refrigeration cooling unit.

1. Carbon emissions from driving are precalculated as follows:

The carbon emission from the driving factor is calculated from fuel consumption and the carbon emission from burning 1 L of diesel.

$$CD = FR * C$$

where

CD: the carbon emissions emit from driving one kilometer (kgCo2/km)

FR: Fuel consumption of refrigerated truck (L/km)

C: Carbon emission emits from burning 1L of diesel (kg/L)

2. Carbon emissions from the refrigerated cooling unit

The total carbon emissions of the refrigerated cooling unit come from the calculation of fuel consumption and cooling capacity depending on the thermal efficiency (η) of the diesel engine, the calorific value of fuel (q), and the COP of a refrigeration system. Contrasting the carbon emission from driving, the carbon emission from the refrigerated cooling unit is a constant rate and is denoted as CR. The value is obtained from Table 4 which is per kilogram of diesel consumption emits 3.06 kilograms of carbon dioxide (3.06 CO2kg/kg)

The fuel consumption (or fuel mass, kg per hour) is calculated from the thermal energy generated by the temperature ambient and the respiration heat from fresh fruits and vegetables, and the variables are defined below:

- Fuel mass from thermal energy generated by respiration heat of fresh fruits and vegetables. The thermal energy is influenced by the load mass of fresh fruits and the respiration rate of the fresh fruits where rf is the respiratory heat of fruit and vegetables per unit mass (J/kg•s)
- Fuel Mass from the thermal energy generated by the temperature ambient and the parameters are defined as follows:
  - λ is the thermal conductivity of thermal insulation materials (J/m· ℃)
  - δ is the thickness of the insulation layer of a refrigerated vehicle (m)
  - T_0 is an ambient temperature, ℃ (Temperature outside the truck).
  - T_f is the optimal temperature based on fresh fruits and vegetables inside the truck, ℃

- HG_{tc} is a heat gain per unit of compartment c in layout type t calculated as follows:

$$HG_{tc} = \frac{volume_{tc}}{surface_{tc}}$$

where

$volume_{tc}$ is the volume of compartment c in layout type t (m^3)

$surface_{tc}$ is denoted as the surface of compartment c in layout type t (m^2).

L is the length of the refrigerated container (m)

H is the height of the refrigerated container (m)

W_{tc} is the width of existing compartment c in layout type t (m)

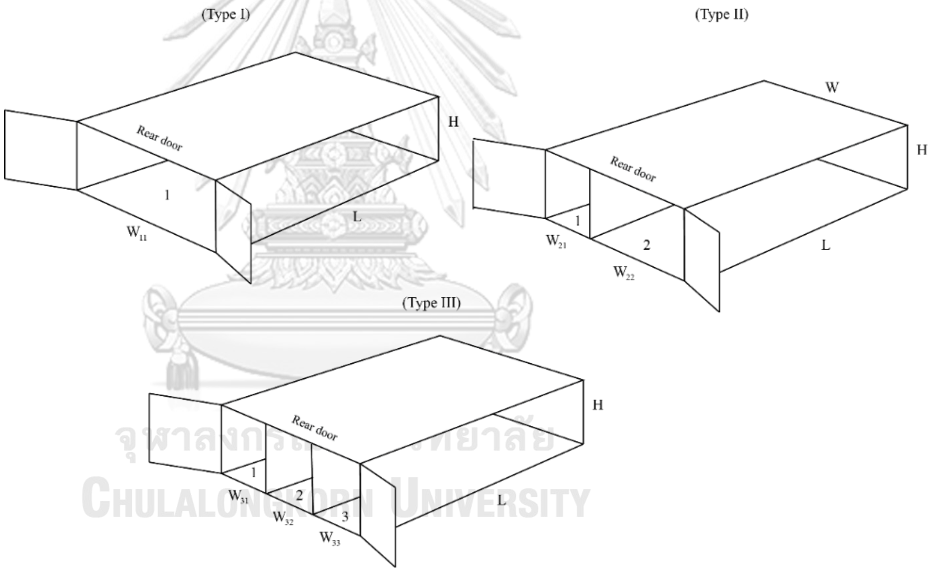


Figure 17: Vehicle Compartment Layout (3D)

<u>*Fresh Fruits, Shelf life, and Weight Loss*</u>

- Let $F = \{1,2, \ldots f\}$ be a set of fresh fruits and vegetable types.
- W_f is denoted as the total weight of fresh fruits required to be shipped on a given day.
- SL_f is the shelf life of fresh fruit f stored at optimal temperature after harvested.

- WL_f is a rate of water loss or weight loss (% per hour) of fresh fruit f.
- Let $K = \{1,2, \dots k\}$ be the index set of allowable maximum transportation time based on the acceptance range of overall shipment quality.
- Q_K is the set of acceptable quality range
- $Tmax_K$ is the set of maximum allowable transportation time at a set of specific quality levels (Q_K).
- M is the sufficiently large constant.

3.3.1b Decision Variables

- *Decision variable on activating vehicles*

a_{vt} is a binary variable.

$$a_{vt} = \begin{cases} 1, & \text{If layout type } t \text{ is selected on vehicle } v \text{ .} \\ 0, & \text{otherwise} \end{cases}$$

- Decision variable on fruits assignment on each vehicle

x_{vtcf} is a binary variable.

$$x_{vtcf} = \begin{cases} 1, & \text{If fresh fruit and vegetable } f \text{ is assigned to compartment } c \\ & \text{on vehicle } v \text{ with type } t \\ 0, & \text{otherwise} \end{cases}$$

- Decision variable on the amount of fruit to be assigned on each vehicle.

y_{vtcf} is a variable indicating the quantity (in kgs) of fresh fruit f is assigned to compartment c on vehicle v with type t

- Decision variable on path selection

z_{vij} is a binary variable.

$$z_{vij} = \begin{cases} 1, & \text{If vehicle `}v\text{' is traveling from node `}i\text{' to `}j\text{.} \\ 0, & \text{otherwise} \end{cases}$$

- Decision variable on selecting the maximum allowable time to set the transportation time constraint from the transportation time set in table 5 for each vehicle.

ts_{vk} is a binary variable.

$$ts_{vk} = \begin{cases} 1, & \text{If } 'tmax_k' \text{ is selected for vehicle } 'v'. \\ 0, & \text{otherwise} \end{cases}$$

3.3.1c Auxiliary Decision Variables

- Decision variable indicating the existence of fresh fruits and vegetables in each vehicle.

$fs_{vf} \in \{0,1\}$

$$fs_{vf} = \begin{cases} 1, & \text{if fresh fruit } `f' \text{ exist in vehicle } `v' \\ 0, & \text{otherwise} \end{cases}$$

- Decision variable on the set of available transportation time based on the shelf life of fresh fruits and vegetable.

$setSL_{vf}$ Shelf life of assigned fresh fruit and vegetables f in vehicle v.

- Decision variable on total transportation time on each vehicle.

TT_v The total transportation time on each vehicle v.

- Decision variable on diesel fuel used to maintain the optimal temperature inside the vehicle.

HW_v The diesel fuel usage for thermal energy (kg/hour) from heat entering through the truck wall

HF_v The diesel fuel usage for thermal energy (kg/hour) from fresh fruits and vegetable respiration heat load per unit volume.

- Decision variable on mass density of loaded fresh fruits and vegetables converted from the fresh fruits and vegetable amount assigned to each compartment to calculate the respiration heat per unit volume.

m_{vtcf} Fresh fruits and vegetable mass (stack density) per unit volume (kg/m^3) of fresh fruit f in compartment c on vehicle v with type t

- Decision variable on quantity loss (in kg) based on weight loss rate (%/hour)

qwl_{vf} The quantity loss (in kg) of fresh fruits and vegetables f of vehicle v

3.3.1d Objective Function Descriptions

Objective function 1:

This objective function aims to select the specific route for each vehicle to minimize the total transportation cost considering transportation cost, handling cost, the fixed cost of operating vehicles, and compartment adjustment cost.

Minimize Objective1 : $Z_1 = \sum_v \sum_i \sum_j ((tc_{ij} + hc_j) \cdot z_{vij}) + \sum_v \sum_t (FIX_v \cdot a_{vt}) + \sum_v \sum_t \sum_c (ADJ_{tc} \cdot a_{vt})$

Objective function 2:

This objective goal is to minimize the total carbon emission from the driving distance and the refrigerant cooling unit which is defined as follows:

1. Carbon emissions from driving (CD) are obtained from the precalculated carbon emission (CO2kg/km) and the total carbon emission emitted is calculated from the multiplication of carbon emissions from driving and total distance $(CD \cdot \sum_v \sum_i \sum_j (d_{ij} \cdot z_{vij}))$.

2. In contrast, the carbon emissions factor from the refrigerated cooling unit is constant (CR). The total carbon emissions are calculated from the multiplication of the carbon emissions factor and total fuel consumption in maintaining the optimal temperature inside the vehicle $(CR \cdot \sum_v (TT_v \cdot (HW_v + HF_v))$.

Hence, the total carbon emissions are calculated based on total transportation distance and time.

Minimize Objective2: $Z_2 = CD \cdot \sum_v \sum_i \sum_j (d_{ij} \cdot z_{vij}) + CR \cdot \sum_v (TT_v \cdot (HW_v + HF_v))$

Objective function 3:

This objective function aims to minimize the actual weight obtained from the multiplication of the fruit weight loss rate (kg per hour) and transportation time on each vehicle. This thesis uses the weight loss rate to approximate the remaining total weight at the destination due to a linear relationship with time.

Minimize Objective 3 $\quad Z_3 = \sum_v (qwl_v \cdot TT_v)$

3.3.1e Mathematical Model and Constraints

$$\text{Minimize Objective} \quad \boldsymbol{Z_{All}}\,(Z_1, Z_3, Z_3) \tag{3-1}$$

$$subject\ to$$

$$\sum_t a_{vt} \le 1 \quad \forall v \in V \tag{3-2}$$

$$\sum_f x_{vtcf} \le A_{tc} \cdot a_{vt} \quad \forall v \in V, \forall t \in T, \forall c \in C \tag{3-3}$$

$$y_{vtcf} \le cap_{tc} \cdot x_{vtcf} \quad \forall v \in V, \forall t \in T, \forall c \in C, \forall f \in F \tag{3-4}$$

$$\sum_v \sum_t \sum_c y_{vtcf} = W_f \quad \forall f \in F \tag{3-5}$$

$$\sum_j^n z_{voj} = \sum_t a_{vt} \quad \forall v \in V, \forall t \in T \tag{3-6}$$

$$\sum_i^n z_{vik} = \sum_j^n z_{vkj} \quad \forall v \in V, k \in N \setminus \{o, d\} \tag{3-7}$$

$$\sum_i^n z_{vid} = \sum_t a_{vt} \quad \forall v \in V \tag{3-8}$$

$$\sum_k ts_{vk} \le \sum_t a_{vt} \quad \forall v \in V \tag{3-9}$$

$$\sum_i \sum_j ((tt_{ij} + ht_j) \cdot z_{vij}) \le \sum_k (ts_{vk} \cdot Tmax_k) \quad \forall v \in V \tag{3-10}$$

$$\sum_k (ts_{vk} \cdot Tmax_k) \le min_f\{setSL_{vf}\} \quad \forall v \in V \tag{3-11}$$

$$fs_{vf} = \sum_t \sum_c x_{vtcf} \quad \forall v \in V, \forall f \in F \tag{3-12}$$

$$setSL_{vf} = \begin{cases} M, \; fs_{vf} \cdot SL_f & \le 0 \\ fs_{vf} \cdot SL_f, \; fs_{vf} \cdot SL_f & \ge 0 \end{cases} \quad \forall v \in V, \forall f \in F \tag{3-13}$$

$$TT_v = \sum_i \sum_j ((tt_{ij} + ht_j) \cdot z_{vij}) \quad \forall v \in V \tag{3-14}$$

$$HW_v = \sum_t \sum_c \sum_f \frac{[x_{vtcf} \cdot HG_{tc} \cdot (T_0 - T_f) \cdot \frac{\lambda}{\delta})}{q \cdot \eta \cdot COP} \quad \forall v \in V \tag{3-15}$$

$$HF_v = \sum_t \sum_c \sum_f \frac{m_{vtcf} \cdot r_f}{q \cdot \eta \cdot COP} \quad \forall v \in V \tag{3-16}$$

$$m_{vtcf} \quad = \quad \frac{y_{vtcf}}{volume_{tc}} \qquad \forall v \in V, \forall t \in T, \forall c \in C, \forall f \in F \qquad (3\text{-}17)$$

$$qwl_{vf} \quad = \quad \sum_t \sum_c y_{vtcf} \cdot WL_f \qquad \forall v \in V, \forall f \in F \qquad (3\text{-}18)$$

$$y_{vtcf,} HW_v, HF_{v,} qwl_{vf}, TT_v \quad \geq \quad 0 \qquad \forall v \in V, \forall t \in T, \forall c \in C, \forall f \in F \quad (3\text{-}19)$$

$$a_{vt}, x_{vtcf}, z_{vij}, ts_{vk}, fs_{vf} \in \{0,1\} \qquad \forall v \in V, \forall t \in T, \forall c \in C, \forall f \in F, \forall v \in V \quad (3\text{-}20)$$

Vehicle Selection Constraint:

Constraint (3-2) represents vehicle layout decisions which indicate that only one type of compartment layout can be utilized on each vehicle 'v'; hence, vehicle 'v' is active.

Fruit Assignment Constraints:

Constraint (3-3) – (3-5) defines the assignment of fresh fruits on each vehicle. Constraint (3-3) ensures that a compartment of any layout could be used only for one fresh fruit type, and the compartment exists. Constraint (3-4) specifies that the assigned amount must not exceed the selected compartment capacity. Constraint (3-5) restricts that the total assigned amount must meet the total shipping requirement.

Route Selection Constraints:

Constraints (3-6) – (3-8) represent route selection constraints. Constraint (3-6) provides that just one edge is chosen from the origin to the transit node if the vehicle 'v' is activated. At the same time, Constraint (3-7) specifies that the total inflow and total outflow are equal at any transit city. Constraint (3-8) enforces that just one edge links to a destination node for the activated vehicle.

Transportation time Constraint:

Constraints (3-9) – (3-14) represent total transportation time constraints. Constraint (3-9) ensures that only one maximum allowable time is selected from the set of maximum allowable times for the activated vehicle. Constraint (3-10) specifies that each vehicle's transportation time must not exceed the selected maximum allowable total time. Meanwhile, Constraint (3-11) ensures that the selected maximum allowable time does not exceed the least fresh fruit shelf-life on vehicle v. Constraints (3-12) – (3-13) determine the existence of fresh fruit type in each vehicle and assign the shelf life set for each vehicle. Each shelf-life set is varied depending on the existence of fresh fruit type in each vehicle. For the non-existence of fresh fruits and vegetables, the shelf-life is equal to the sufficiently large number (M). Constraints (3-14) determine the total transportation time of each vehicle.

Equality Constraints:

Constraints (3-15) - (3-18) are the equalities constraint that defines the value of fuel consumption from maintaining the optimal temperature inside the activated vehicle and the quality loss of the assigned fresh fruits and vegetables on each activated vehicle.

Non-negativity:

Constraints (3-19) - (3-20) define the model's decision variables.

3.3.2 Linearized Mathematical Model Formulation

The objective functions mentioned above, objectives 2 and 3, are non-linear equations containing the multiplication of two continuous decision variables. It is complex to directly solve the multiplication of two continuous decision variables employing the piecewise approximation method. To reduce the complexity, in this section, the new continuous variables and constraints are introduced to transform the multiplication of two continuous decision variables into separable non-linear functions. The simplified non-linear functions are then approximated by using the piecewise linear approximation approach through the breakpoints to coordinate the selection and its slope value.

3.3.2a Set and Parameter

To linearize the non-linear function, the piecewise linear approximation is associated with the number of breakpoint segments which is denoted as nb^1, $nb^2$and nb^3 and more explanations are defined below:

Number of segments

nb^1 is the number of segments for piecewise linear approximation related to the product of diesel fuel used for thermal energy from heat entering through the truck wall and the total transportation time.

nb^2 is the number of segments for piecewise linear approximation related to the product of diesel fuel used for thermal energy from fresh fruits and respiration heat loads per unit volume and the total transportation time.

nb^3 is the number of segments for piecewise linear approximation related to the product of total fresh fruits and vegetable quantity loss and the total transportation time

After defining the number of the coordinates breakpoints segments, the range of breakpoints is then defined as follows:

Breakpoints

s The set of index numbers for nonlinear function where $s = \{1,..,S\}$

l The index value of breakpoints where l = $\{0,..,nb^1\}$

m The index value of breakpoints where m = $\{0,..,nb^2\}$

n The index value of breakpoints where n = $\{0,..,nb^3\}$

a_l The value of breakpoints within the interval $[a_0, a_{nb^1}]$

b_m The value of breakpoints within the interval $[b_0, b_{nb^2}]$

c_n The value of breakpoints within the interval $[c_0, c_{nb^3}]$

3.3.2b Decision Variables

As mentioned above, the objective function transformation into a separable equation is required to employ the piecewise linear approximation approach.

<u>*New continuous variables*</u>

$t1_v$, $t2_v$ — are two new continuous variables for transforming the product of diesel fuel used for thermal energy from heat entering through the truck wall and the total transportation time of vehicle v into a separable form.

FH_v — is an approximate value of the product of diesel fuel used for thermal energy from heat entering through the truck wall and the total transportation time of vehicle v.

$m1_v$, $m2_v$ — Two new continuous variables for transforming the product of diesel fuel used for thermal energy from fresh fruits and vegetable respiration heat loads per unit volume and the total transportation time of vehicle v into a separable form.

FF_v — An approximate value of the product of diesel fuel used for thermal energy fresh fruits and vegetable respiration heat loads per unit volume and the total transportation time of vehicle v.

$w1_v$, $w2_v$ — Two new continuous variables for transforming the product of total fresh fruits and vegetable quantity loss on vehicle and the total transportation time of vehicle v into a separable form.

QL_v — An approximate value of the product of total fresh fruits and vegetable quantity loss on vehicle and the total transportation time of vehicle v.

<u>*New Non-linear Functions*</u>

$f_s(x_v)$ where $s = \{1,2,3,4,5,6\}$ denoted as new non-linear functions and is defined as follows:

- Let $f_1(t1_v) = t1_v \cdot t1_v$ be a non-linear function of a single variable $t1_v$ of vehicle v

- Let $f_2(t2_v) = t2_v \cdot t2_v$ be a non-linear function of a single variable $t2_v$ of vehicle v
- Let $f_3(m1_v) = m1_v \cdot m1_v$ be a non-linear function of a single variable $m1_v$ of vehicle v
- Let $f_4(m2_v) = m2_v \cdot m2_v$ be a non-linear function of a single variable $m2_v$ of vehicle v
- Let $f_5(w1_v) = w1_v \cdot w1_v$ be a non-linear function of a single variable $w1_v$ of vehicle v
- Let $f_6(w2_v) = w2_v \cdot w2_v$ be a non-linear function of a single variable $w2_v$ of vehicle v

Breakpoint Decision Variables

The extra binary variables are introduced to force the new values $(t1_v, t2_v, m1_v, m2_v, w1_v, w2_v)$ to associate with the proper pair of consecutive breakpoints.

- ${t_{vl}}^s$ is a binary variable of vehicle v for $f_{s\epsilon\{1,2\}}(x_v)$

$${t_{vl}}^s = \begin{cases} 1, & If\ \ break\ point\ at\ point\ 'l'\ is\ selected \\ 0, & otherwise \end{cases}$$

- ${h_{vm}}^s$ is a binary variable of vehicle v for $f_{s\epsilon\{3,4\}}(x_v)$

$${h_{vm}}^s = \begin{cases} 1, & If\ \ break\ point\ at\ point\ 'm'\ is\ selected \\ 0, & otherwise \end{cases}$$

- ${u_{vn}}^s$ is a binary variable of vehicle v for $f_{s\epsilon\{5,6\}}(x_v)$

$${u_{vn}}^s = \begin{cases} 1, & If\ \ break\ point\ at\ point\ 'n'\ is\ selected \\ 0, & otherwise \end{cases}$$

3.3.2c Objective Function Description

Objective Function 4:

This objective aims to minimize the total carbon emission by using piecewise linear approximation, the multiplication of total transportation time and fuel consumption from the refrigerant cooling unit are substituted by new continuous variables ($CR \cdot \sum_v (FH_v + FF_v)$).

Minimize Objective4 $$Z_4 = CD \cdot \sum_v \sum_i \sum_j (d_{ij} \cdot z_{vij}) + CR \cdot \sum_v (FH_v + FF_v)$$

Objective Function 5:

The approximated optimal value of the total quality loss is approximated by substituting the multiplication of total weight loss and total time with a new continuous variable into the objective function $\sum_v(QL_v)$.

Minimize Objective 5 $\qquad Z_5 = \sum_v QL_v$

3.3.2d Mathematical Model and Constraints

$$\text{Minimize} \qquad \boldsymbol{Z_{ALL}} = (\boldsymbol{Z_1}, \boldsymbol{Z_4}, \boldsymbol{Z_5}) \qquad \text{(3-21)}$$

$$subject\ to$$

$$t1_v = \frac{1}{2}(TT_v + HW_v) \qquad \forall v \in V \qquad \text{(3-22)}$$

$$t2_v = \frac{1}{2}(TT_v - HW_v) \qquad \forall v \in V \qquad \text{(3-23)}$$

$$FH_v = {t1_v}^2 - {t2_v}^2 \qquad \forall v \in V \qquad \text{(3-24)}$$

$$m1_v = \frac{1}{2}(TT_v + HF_v) \qquad \forall v \in V \qquad \text{(3-25)}$$

$$m2_v = \frac{1}{2}(TT_v - HF_v) \qquad \forall v \in V \qquad \text{(3-26)}$$

$$FF_v = {m1_v}^2 - {m2_v}^2 \qquad \forall v \in V \qquad \text{(3-27)}$$

$$w1_v = \frac{1}{2}(TT_v + \textstyle\sum_f qwl_{vf}) \qquad \forall v \in V \qquad \text{(3-28)}$$

$$w2_v = \frac{1}{2}(TT_v - \textstyle\sum_f qwl_{vf}) \qquad \forall v \in V \qquad \text{(3-29)}$$

$$QL_v = {w1_v}^2 - {w2_v}^2 \qquad \forall v \in V \qquad \text{(3-30)}$$

$$L(f_1(t1_v)) = \textstyle\sum_l f_1(a_l) \qquad \forall v \in V, \forall l \,\epsilon\, \{0, \dots, nb^1 - 1\} \qquad \text{(3-31)}$$

$$L(f_2(t2_v)) = \textstyle\sum_l f_2(a_l) \qquad \forall v \in V, \forall l \,\epsilon\, \{0, \dots, nb^1 - 1\} \qquad \text{(3-32)}$$

$$L(f_3(m1_v)) = \textstyle\sum_m f_3(b_m) \qquad \forall v \in V, \forall m \,\epsilon\, \{0, \dots, nb^2 - 1\} \qquad \text{(3-33)}$$

$$L(f_4(m1_v)) = \textstyle\sum_m f_4(b_m) \qquad \forall v \in V, \forall m \,\epsilon\, \{0, \dots, nb^2 - 1\} \qquad \text{(3-34)}$$

$$L(f_5(w1_v)) = \textstyle\sum_n f_5(c_n) \qquad \forall v \in V, \forall n \,\epsilon\, \{0, \dots, nb^3 - 1\} \qquad \text{(3-35)}$$

$$L(f_6(w2_v)) = \textstyle\sum_n f_6(c_n) \qquad \forall v \in V, \forall n \,\epsilon\, \{0, \dots, nb^3 - 1\} \qquad \text{(3-36)}$$

$$t1_v \geq a_l - (a_{nb^1} - a_0)(1 - {t_{vl}}^s) \qquad \forall l \,\epsilon\, \{0, \dots, nb^1 - 1\}, \forall v \in V, s \in \{1\} \qquad \text{(3-37)}$$

$$t1_v \le a_{l+1} + (a_{nb^1} - a_0)(1 - t_{vl}{}^s)\ \forall l \in \{0, \dots, nb^1 - 1\}, \forall v \in V\ s \in \{1\} \quad (3\text{-}38)$$

$$t2_v \ge a_l - (a_{nb^1} - a_0)(1 - t_{vl}{}^s)\quad \forall l \in \{0, \dots, nb^1 - 1\}, \forall v \in V, s \in \{2\} \quad (3\text{-}39)$$

$$t2_v \le a_{l+1} + (a_{nb^1} - a_0)(1 - t_{vl}{}^s)\ \forall l \in \{0, \dots, nb^1 - 1\}, \forall v \in V, s \in \{2\} \quad (3\text{-}40)$$

$$m1_v \ge b_m - (b_{nb^2} - b_0)(1 - h_{vm}{}^s)\ \forall m \in \{0, \dots, nb^2 - 1\}, \forall v \in V, s \in \{3\} \quad (3\text{-}41)$$

$$m1_v \le b_{m+1} + (b_{nb^2} - b_0)(1 - h_{vm}{}^s)\ \forall m \in \{0, \dots, nb^2 - 1\}, \forall v \in V, s \in \{3\} \quad (3\text{-}42)$$

$$m2_v \ge b_m - (b_{nb^2} - b_0)(1 - h_{vm}{}^s)\quad \forall m \in \{0, \dots, nb^2 - 1\}, \forall v \in V, s \in \{4\} \quad (3\text{-}43)$$

$$m2_v \le b_{m+1} + (b_{nb^2} - b_0)(1 - h_{vm}{}^s)\ \forall m \in \{0, \dots, nb^2 - 1\}, \forall v \in V, s \in \{4\} \quad (3\text{-}44)$$

$$w1_v \ge c_n - (c_{nb^3} - c_0)(1 - u_{vn}{}^s)\quad \forall n \in \{0, \dots, nb^3 - 1\}, \forall v \in V, s \in \{5\} \quad (3\text{-}45)$$

$$w1_v \le c_{n+1} + (c_{nb^3} - c_0)(1 - u_{vn}{}^s)\ \forall n \in \{0, \dots, nb^3 - 1\}, \forall v \in V,\ s \in \{5\} \quad (3\text{-}46)$$

$$w2_v \ge c_n - (c_{nb^3} - c_0)(1 - u_{vn}{}^s)\ \forall n \in \{0, \dots, nb^3 - 1\}, \forall v \in V, s \in \{6\} \quad (3\text{-}47)$$

$$w2_v \le c_{n+1} + (c_{nb^3} - c_0)(1 - u_{vn}{}^s)\ \forall n \in \{0, \dots, nb^3 - 1\}, \forall v \in V,\ s \in \{6\} \quad (3\text{-}48)$$

$$f_1(t1_v) \ge f(a_l) + \frac{f(a_{l+1}) - f(a_l)}{a_{l+1} - a_i}(x - a_l) - M(1 - t_{vl}{}^s)$$
$$\forall l \in \{0, \dots, nb^1 - 1\}, \forall v \in V, s \in \{1\} \quad (3\text{-}49)$$

$$f_1(t1_v) \le f(a_l) + \frac{f(a_{l+1}) - f(a_l)}{a_{l+1} - a_l}(x - a_l) + M(1 - t_{vl}{}^s)$$
$$\forall l \in \{0, \dots, nb^1 - 1\}, \forall v \in V, s \in \{1\} \quad (3\text{-}50)$$

$$f_2(t2_v) \ge f(a_l) + \frac{f(a_{l+1}) - f(a_l)}{a_{l+1} - a_l}(x - a_l) - M(1 - t_{vl}{}^s)$$
$$\forall l \in \{0, \dots, nb^1 - 1\}, \forall v \in V, s \in \{2\} \quad (3\text{-}51)$$

$$f_2(t2_v) \le f(a_l) + \frac{f(a_{i+1}) - f(a_i)}{a_{i+1} - a_i}(x - a_l) + M(1 - t_{vl}{}^s)$$
$$\forall l \in \{0, \dots, nb^1 - 1\}, \forall v \in V, s \in \{2\} \quad (3\text{-}52)$$

$$f_3(m1_v) \ge f(b_m) + \frac{f(b_{m+1}) - f(b_m)}{b_{m+1} - b_m}(x - b_m) - M(1 - h_{vm}{}^s)$$
$$\forall m \in \{0, \dots, nb^2 - 1\}, \forall v \in V, s \in \{3\} \quad (3\text{-}53)$$

$$f_3(m1_v) \le f(b_m) + \frac{f(b_{m+1}) - f(b_m)}{b_{m+1} - b_m}(x - b_m) + M(1 - h_{vm}{}^s)$$
$$\forall m \in \{0, \dots, nb^2 - 1\}, \forall v \in V, s \in \{3\} \quad (3\text{-}54)$$

$$f_4(m2_v) \ge f(b_m) + \frac{f(b_{m+1}) - f(b_m)}{b_{m+1} - b_m}(x - b_m) - M(1 - h_{vm}{}^s)$$
$$\forall m \in \{0, \dots, nb^2 - 1\}, \forall v \in V, s = \{4\} \quad (3\text{-}55)$$

$$f_4(m2_v) \leq f(b_m) + \frac{f(b_{m+1}) - f(b_m)}{b_{m+1} - b_m}(x - b_m) + M(1 - h_{vm}{}^s)$$
$$\forall m \,\epsilon\, \{0, \dots, nb^2 - 1\}, \forall v \in V, s = \{4\} \quad (3\text{-}56)$$

$$f_5(w1_v) \geq f(c_n) + \frac{f(c_{n+1}) - f(c_n)}{c_{n+1} - c_n}(x - c_n) - M(1 - u_{vn}{}^s)$$
$$\forall n \,\epsilon\, \{0, \dots, nb^3 - 1\}, \forall v \in V, s = \{5\} \quad (3\text{-}57)$$

$$f_5(w1_v) \leq f(c_n) + \frac{f(c_{n+1}) - f(c_n)}{c_{n+1} - c_m}(x - c_n) + M(1 - u_{vn}{}^s)$$
$$\forall n \,\epsilon\, \{0, \dots, nb^3 - 1\}, \forall v \in V, s = \{5\} \quad (3\text{-}58)$$

$$f_6(w2_v) \geq f(c_n) + \frac{f(c_{n+1}) - f(c_n)}{c_{n+1} - c_n}(x - c_n) - M(1 - u_{vn}{}^s)$$
$$\forall n \,\epsilon\, \{0, \dots, nb^3 - 1\}, \forall v \in V, s = \{6\} \quad (3\text{-}59)$$

$$f_6(w2_v) \leq f(c_n) + \frac{f(c_{n+1}) - f(c_n)}{c_{n+1} - c_m}(x - c_n) + M(1 - u_{vn}{}^s)$$
$$\forall n \,\epsilon\, \{0, \dots, nb^3 - 1\}, \forall v \in V, s = \{6\} \quad (3\text{-}60)$$

$$\sum_{l=1}^{nb^1-1} t_{vl}{}^s = 1 \qquad \forall l \,\epsilon\, \{0, \dots, nb^1 - 1\}, \forall v \in V, \forall s \in \{1,2\} \quad (3\text{-}61)$$

$$\sum_{m=1}^{nb^2-1} h_{vm}{}^s = 1 \qquad \forall m \,\epsilon\, \{0, \dots, nb^2 - 1\}, \forall v \in V, \forall s \in \{3,4\} \quad (3\text{-}62)$$

$$\sum_{m=1}^{nb^3-1} u_{vn}{}^s = 1 \qquad \forall n \,\epsilon\, \{0, \dots, nb^3 - 1\}, \forall v \in V, \forall s \in \{5,6\} \quad (3\text{-}63)$$

Transformation Constraints:

Constraints (3-22) – (3-23) transformed the term of $TT_v \cdot HW_v$ into a separable form. In constraints (3-24), $TT_v \cdot HW_v$ is replaced with FH_v which is equal to the summation of the square of two separable terms. Meanwhile, constraints (3-25) – (3-26) transformed the term of $TT_v \cdot HF_v$ into a separable form. In constraints (3-27), $TT_v \cdot HF_v$ is replaced with FF_v which is equal to the summation of the square of two separable terms. Similarly, constraints (3-28) – (3-29) transformed the term of $TT_v \cdot \sum_f qwl_{vf}$ into a separable form. In constraints (3-30), likewise, $TT_v \cdot \sum_f qwl_{vf}$is replaced with QL_v which is equal to the summation of the square of two separable terms.

Piecewise Linear Approximation Constraints:

Constraints (3-31) – (3-36) represent an approximate linearize the non-linear function over the interval $[a_0, a_l]$, $[b_0, b_m]$and $[c_0, c_n]$ consecutively. Constraints (3-37) -(3-48) force the values to be associated with the proper pair of consecutive breakpoints. It results in a more accurate approximation with less deviation. Constraints (3-49) – (3-60) reformulate the non-linear function into the equivalent piecewise linear forms integrating with the slope of the interpolating function. Constraint (3-61)-(3-63) ensures that only one pair of consecutive breakpoints of each function is selected.

3.4 Hybrid NSGA-III for multi-compartment vehicle loading, and transportation routing problem (MCVLRP)

In this study, NSGA III is used as the main algorithm framework. And the adapted exon crossover and mutation operation are customized designed and proposed to generate offspring individuals for the multi-compartment vehicle loading and routing problem (MCVLRP). Further details are provided below.

3.4.1 The Framework of the proposed hybrid NSGA-III

The general framework of the proposed hybrid NSGA-III for the multi-compartment vehicle loading and routing problem (MCVLRP) is illustrated. The initial population (P_t) with size 'N' is first generated for each iteration. The reference points are then constructed. The pair of parents are selected from the current population to generate the offspring (Q_t) through the adapted Exon shuffling crossover operation (or hybrid Exon shuffling). The parent chromosomes from the current population are then mutated using the proposed mutation operation. After performing crossover and mutation operations, the mutated parents (P_t^*) and offspring (Q_t) with the size of $\frac{N}{2}$ are combined and represented as (R_t) and is ranked to each Pareto front level from 1 to l by non-dominated sorted operation. If the size of Pareto front 1 (F_1) equals to 'N' then the new population (P_{t+1}) equals to Pareto Front 1 (F_1) and the procedure continues for 't' generation. After 't' generation the process stops, and the Pareto optimal front is identified. On the other hand, If the size of Pareto front 1 (F_1) less than 'N', the individual from Pareto Front 1 (F_1) is first added to the new population (P_{t+1}), then the remaining value (K) is calculated ($K = N - |F_1|$). The 'K' members are selected from the next Pareto front set (F_l). The fitness values of each member from every Pareto front set are normalized. Each individual is associated with a reference point on the reference line. Niche count is performed by selecting the individual with the closest distance to the reference point on the reference line. The 'K' members are selected and added to the new population set (P_{t+1}). The process is repeated until 't' generation stops, at which point the Pareto optimal front is reported.

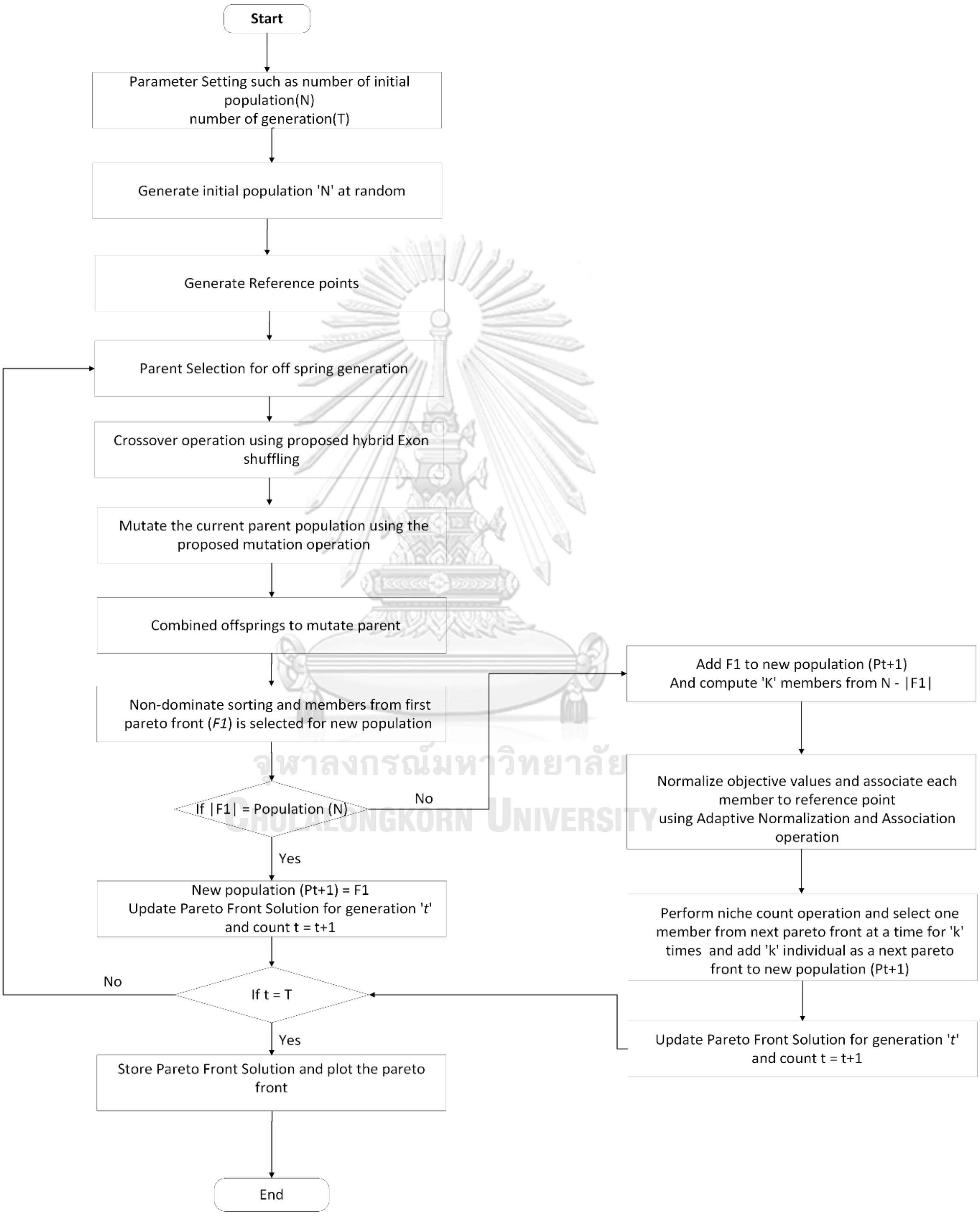


Figure 18: The general framework of the proposed algorithm

3.4.2 Parameter Setting for proposed hybrid NSGA-III

To employ the hybrid NSGA-III heuristic, several parameters need to be defined Some of the concepts in this section are from the original NSGA-III, while others are distinctive to the original NSGA-III, such as the initial population, crossover, and mutation operations.

Population Size (N):

The population size denotes the number of solutions generated by the initial population algorithm as well as the improvement algorithm. According to Reed et al. (2003), the lower bound of population size ($\boldsymbol{N_0}$) equals two times the ideal number of non-dominated solutions (R_{ND}). The number of non-dominated solutions (R_{ND}) is can be set to 30, which is roughly half of the theoretical maximum number of nondominated solutions that could exist. Hence, the lower bound of the initial population is 60 and the population size is then doubled for the next population size.

Crossover rate (Pc):

According to Goldberg (1989), the probability of crossover (Pc) is the likelihood that the pair of parents will be chosen from the population, implying that not all mating pairs must reproduce through crossover, but Pc=1.0 could be chosen. The crossover selection probability is set to one in this study (Pc=1.0). In contrast to normal crossover, only half of the population is produced as offspring because two parents are combined to produce the offspring.

Parent mutation rate (Pm):

The mutation probability is used in the solution improvement procedure to expand the solution in the search space, resulting in the avoidance of the local optimal. In contrast to the original NSGA-III, the mutation probability indicates the likelihood of the individual (or parent) being mutated. Following that, the mutation operation options are therefore chosen. In this study, the mutation rate was set to 0.25.

Gene mutation rate (Pmg):

After assigning the mutation probability to the parents in the initial population, to improve the solution in the search space, the gene mutation probability is introduced in this hybrid NSGA-III algorithm. This gene mutation probability only applies when not every bit in each chromosome is chosen to be mutated. Similarly, the gene mutation rate was set to 0.25. The method details are discussed further in the following section for clarification.

Generation (t):

The iterations of improvement algorithms used to search for better solutions in search spaces are defined by generations. The optimal solution may not be found if the number of iterations is too small. On the other hand, an excessive number of iterations will be time-consuming in real life. According to Yuan et al. (2015), the number of generations is set to 1000 at the maximum.

3.4.3 Chromosome Coding

Encoding

The decision variables chromosome is made up of four sub-strings that correspond to the vehicle layout type, the assigned amount, the fresh fruit and vegetable type, and route selection. The first sub-string represents the selected vehicle layout with a varied length of 'n' and each gene contains an integer decision variable that indicates vehicle layout type. The second substring represents the assigned amounts of fresh fruit and vegetables. Each gene is composed of a sub-string with a fixed length of three that represents the compartment of each vehicle with available capacity and contains continuous decision variables that indicate the amount of fresh fruit and vegetables assigned to each compartment. The third sub-string represents the set of fresh fruit and vegetable types. Each gene also has one three-bit of sub-string that represents an integer decision variable of vegetables and fresh fruits type in each compartment. It is relevant to note that the genes in the second and third sub-strings correspond to the layout type in the first sub-string, as the layout type limits the capacity of each compartment. Because the assigned amount and vegetable fresh fruit type is chosen at the same time, sub-strings 2 and 3 are related. Finally, the fourth sub-string with length 'n' represents the selected route, and each gene contains an integer decision variable that indicates the route selection decision. Furthermore, sub-string 4 is the same length as sub-string 1 and corresponds to sub-string 3 (Figure 19).

For example, there are three types of fresh fruit where $f = \{1,2,3\}$ with a given requirement amount (kgs) and eight transportation routes from origin to destination. As shown in Figure 20, the sub-string1 indicates that three vehicles are used with different layout types. The sub-string 2 represents the weight (kgs) of fresh fruits assigned to each compartment on each vehicle constrained by specific capacity while sub-string 3 defines the type of fresh fruits assigned to each compartment. Lastly, sub-string 4 specifies the selected route for each vehicle.

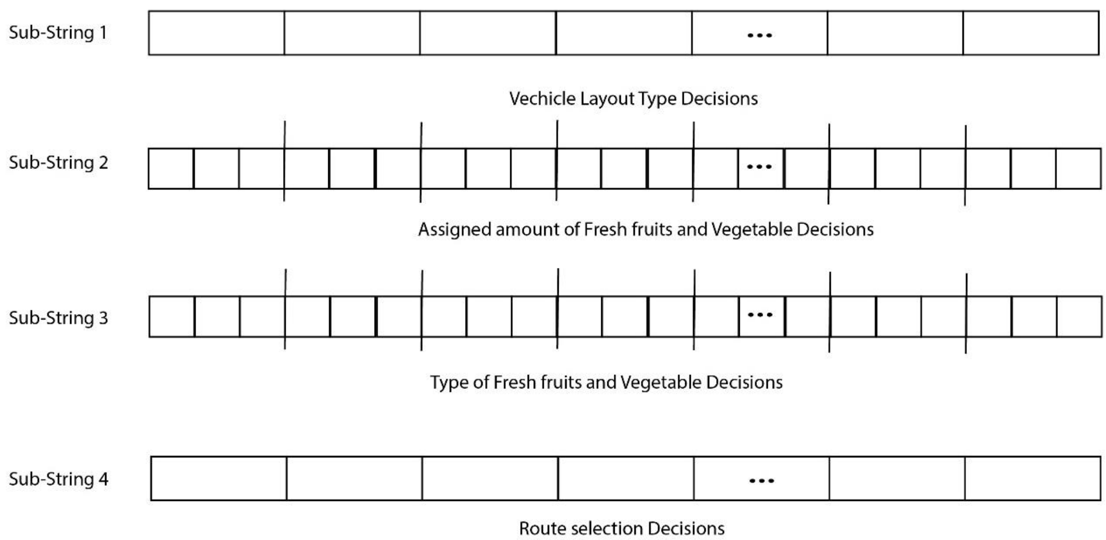


Figure 19: Decision Variables in Chromosome for the multi-compartment vehicle loading and routing problem (MCVLRP)

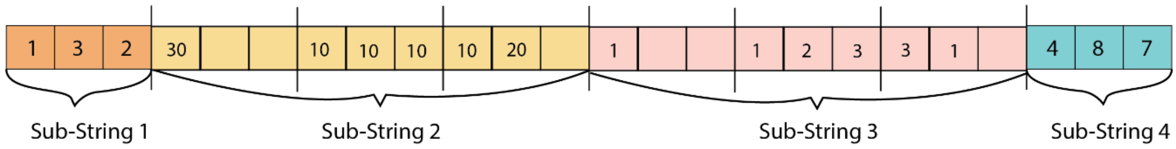


Figure 20: Description of a coding example

(Noted that the unit in substring 3 is in thousand kilograms.)

Decoding

The data decoding array structure is depicted in Figure 21. The data such as compartments' capacity of each vehicle, heat gain through wall per unit volume, and average mass of each compartment on the activated vehicle can be extracted from each gene in sub-string 1. While fresh fruit's specific requirements, such as optimal temperature, shelf-life, and water loss rate, can be decoded from a gene in sub-string 3 of fresh fruit and vegetable types. Lastly, each gene in sub-string 4 contains route-specific information such as transportation time, transportation costs, and transportation distances while the operation cost is obtained from the adjustment cost from sub-string 1 (the vehicle layout type).

Vehicle	Compartment 'C'																	
	Capacity			Average Mass			Heat Gain			Optimal Temp			Shelf life			Weight Loss Rate		
1																		
2																		
.																		
V																		

Vehicle	Operation Cost	Transportation Cost	Transportation Time	Transportation Distance
1				
2				
.				
V				

Figure 21: Decoding the chromosome for the multi-compartment vehicle loading and routing problem (MCVLRP)

3.4.4 Input Data for Multi-compartment vehicle loading, and transportation routing problem (MCVLRP)

In this study, the mathematical model and the proposed heuristic are applied to fresh fruit exportation data from Talaad-Thai (Thailand) to Nanning (China) as an example of a real-life case. China is one of the main destinations for Thai fruit exporters. The Chinese market has emerged as a popular destination for Southeast Asian fruit export markets regarding the China-ASEAN free trade zone. Road and sea freight are popular modes of transporting fresh fruits from Thailand to China. The products are exported by crossing ASEAN borders or shipping from port to port with varying distances, circumstances, and customs clearance processes. Even though sea routes usually are less expensive than land routes, the ideal route from Thailand to China is heavily influenced by other factors such as stakeholders' circumstances and types of fruit. In addition, a continuous improvement in transportation infrastructure increases route links between Thailand and China, making transportation via trucks more efficient (Krebs and Panichakarn, 2019). Accordingly, this study examines and analyzes the optimal route for transporting Thai fresh fruits to China. With a significant increase in export growth rate and an expansion of connecting routes from Thailand to China, an appropriate distribution network design is critical for market competitiveness and profitability.

3.4.4a Fresh Fruits and Vegetable Data

Tropical Thai fruits have various harvesting seasons and various transport temperature requirements. For example, the highly popular fruits, Durian, Mangosteen, and Longan, all have different harvesting seasons. According to the Thai Office of Agricultural Economics exporting statistics (Fig. 22), Durian and Mangosteen usually ship in May, whereas Logan has the highest in the fourth quarter of the year. From July through September, the export volume of three types of fruit was distributed evenly. Due to varied shipping amounts and temperature requirements, utilizing multi-compartment vehicles could be beneficial for consolidating different fruit types throughout the offseason within the same vehicle. Accordingly, Durian, Logan, and Mangosteen are chosen as small instances of this problem.

However, more fresh fruit types are chosen based on their popularity for medium and large instances to further investigate the model and heuristic performance. Five types of fresh fruits are used for medium instances while eight types of fresh fruits are applied for large instances. In a report by Bangkok Post, Phusit Ratanakul Sereroengrit, director-general of the Trade Policy and Strategy Office, said that among other important export products, the country's fruit exports will continue to grow significantly. Thailand's fresh fruit exports increased by 42.21 percent in the first half of 2021, led by durian, longan, mangoes, banana, pineapple, citrus, and lychee (Arunmas, 2021). Therefore, exporting mangoes, bananas, pineapple, citrus, and lychee are included in this study.

Figure 22 : Export quantity (kgs) of Durian, Mangosteen, and Longan in 2020

In addition, according to the Division of Agriculture and Natural Resources, University of California (n.d.) and Lufu et. Al. (2020), the optimal temperature, shelf-life, and weight loss data are presented in Table 5.

Fruit	Optimal Temperature	Shelf-life	Weight loss (%per day)
Durian	14 ± 1 °C	3 - 5 weeks	1.5
Longan	5 ± 1 °C	2 - 4 weeks	0.77
Mangosteen	13 ± 1 °C	2 - 4 weeks	1
Mangoes	5 ± 1 °C	2 - 3 weeks	0.4
Bananas	13± 1 °C	4 - 6 weeks	0.10
Pineapple	10 ± 1 °C	4 - 6 weeks	0.67
Citrus	13 ± 1 °C	3 - 4 weeks	0.19
Lychee	5 ± 1 °C	3 - 4 weeks	0.66

Table 5: Optimal Temperature and Shelf life for the Selected Fruit

To process further steps in finding the optimal solution set, the obtained data are converted to the appropriate units for the fitness calculation of each objective. The shelf-life and weight loss rate (%) from Table 5 are converted into hours and kilogram per hour (kg/hour) consecutively and are shown in Table 6.

Fruit	Optimal Temperature	Shelf-life (hours)	Weight loss rate (kg/hour)
Durian	14 ± 1 °C	336	1.5
Longan	5 ± 1 °C	672	0.77
Mangosteen	13 ± 1 °C	672	1
Mangoes	5 ± 1 °C	504	0.4
Bananas	13± 1 °C	1008	0.10
Pineapple	10 ± 1 °C	1008	0.67
Citrus	13 ± 1 °C	672	0.19
Lychee	5 ± 1 °C	672	0.66

Table 6: Converted data of Selected Fruits

3.4.4b Route Data from Thailand to China

There are 19 route segments with 14 cities connecting Talaad-Thai to Nanning. The relevant data, obtained from Krebs and Panichakarn (2019), are shown in Table 7 and Table 8.

Segment	From	To	Modes
1	Talaad Thai	Beungkan	Road
2	Talaad Thai	Mukdahan	Road
3	Talaad Thai	Nakhonpathom	Road
4	Talaad Thai	Laemchabang Port	Road
5	Beungkan	Paksan	Road
6	Mukdahan	Suvannakhet	Road
7	Nakhonpathom	Khommuan	Road
8	Laemchabang Port	Haiphong Port	Sea
9	Laemchabang Port	Qinzhou Port	Sea
10	Laemchabang Port	Guangzhou Port	Sea
11	Laemchabang Port	Fangchenggang Port	Sea
12	Paksan	Hanoi	Road
13	Suvannakhet	Hanoi	Road
14	Khommuan	Hanoi	Road
15	Haiphong Port	Hanoi	Sea
16	Hanoi	Nanning	Road
17	Qinzhou Port	Nanning	Road
18	Guangzhou Port	Nanning	Road
19	Fangchenggang Port	Nanning	Road

Table 7: The Available Cities and Segment

Segment	Cost ($)		Time (hours)	
	Transportation	Handling	Transportation	Handling
1	700	0	12	0
2	800	0	10	0
3	1,300	0	11	0
4	290	0	2.5	0
5	540	220	6	4
6	580	220	5	2.5
7	350	220	3.5	3.5
8	1,300	120	90	24
9	1,495	120	144	24
10	1,700	120	168	24
11	1,210	120	144	24
12	1,510	760	13	4
13	2,070	750	18	3
14	1,430	820	12.5	4
15	720	210	8	24
16	520	1,400	3	2.5
17	380	110	2.5	15
18	1,500	210	8.5	24
19	375	210	4	27

Table 8: Associate Cost and Time

To lessen the complexity of the route selection tested in the proposed heuristic, the route options are first rearranged into 7 routes with total distance, total time, and total cost which are shown in Table 9.

Route	Total Distance	Total Time	Total Cost
1	1797	44.5	6250
2	1959	44	10910
3	1694	40	6040
4	3400	208	4560
5	3200	194	2395
6	4000	227	3820
7	3300	201.5	2205

Table 9: Route Options with total distance, total time, and total cost

3.4.4c Vehicle Data

Suppose a 40-foot container (12.19m (L) x 2.44m (W) x 2.56 m(H)) is used and the capacity can have up to 30,000 kg. All vehicles have identical attributes with varied adjustment costs (Table 10). Based on the previous assumption, with the 30,000 kgs size of the container, the volume and surface of each compartment for each layout type are precalculated as shown in Tables 11 and 12. The heat gain through wall per unit volume (m^{-1}) for each compartment is precalculated from the division of the surface area of the refrigerated vehicle,m^2 and the volume of refrigerated vehicles, m^3 $(H = \frac{S}{V})$ which results are shown in Table 13.

Layout type	1	2	3
Fixed Cost ($)	125	125	125
Adjustment Cost ($)	0	40	60

Table 10: Associated Vehicle Cost

Volume, m^3	1	2	3
1	76.14	0	0
2	25.38	50.76	0
3	25.38	25.38	25.38

Table 11: Volume of each compartment for each layout type.

Surface, m^2	1	2	3
1	1.765169425	0	0
2	2.347952367	2.028228673	0
3	2.347952367	2.347952367	2.347952367

Table 12: Surface area of each compartment for each layout type

Heat Gain m^{-1}	1	2	3
1	1.765169425	0	0
2	2.347952367	2.028228673	0
3	2.347952367	2.347952367	2.347952367

Table 13: Heat Gain through truck wall per unit volume

3.4.5 Data Processing

The fitness value of each objective function calculation is reviewed in this section. First, the total cost is calculated from transportation cost, partition adjustment cost, and vehicle operation cost (fixed cost). Next, in this study, total carbon emissions are calculated from vehicle fuel consumption generated from a vehicle driving regards the travel distance and vehicle vapor compression cooling unit and are illustrated as follows:

Carbon Emissions from vehicle driving ($kgCO_2/km$)

$$CD = FR * C$$

where

CD: carbon emission produced by the heavy truck transporting per kilometer ($kgCO_2/km$)

FR: Fuel consumption of refrigerated truck (= 0.2854 L/Km)

C: Carbon emission emits from burning 1L of diesel (=2.630 kg/L)

Hence, the total carbon emissions generated from vehicle driving are equal to the multiplication of total distance and carbon emission produced by the heavy truck transporting per kilometer (CD). The carbon emissions from the refrigerant cooling unit are caused by keeping the temperature inside the container steady. To elaborate, the temperature fluctuation inside the chamber is caused by two major factors: heat entering from outside and respiration heat from fresh fruits and vegetables. The carbon emissions of the refrigerant cooling unit are calculated from diesel fuel consumption for thermal energy from heat entering through the truck wall and fresh fruits respiration (kg/hour).

Total fuel consumption from refrigerant unit:

$$\text{Total Fuel consumption} = \frac{[H\cdot(T_0-T)\cdot\frac{\lambda}{\delta}+m_v\cdot q_0]d\tau}{q\cdot \eta\cdot COP}$$

where

H: heat gain per unit volume of refrigerated vehicle (m^{-1}) from table13

T_0: ambient temperature ($= 30$ ℃)

T: fruits and vegetable optimal temperature (℃), from table 6

m_v : assigned fruits and vegetable mass (stack density) per unit volume (kg/m^3)

λ : thermal conductivity of thermal insulation materials ($= 0.08\ W/m\cdot K$)

δ: the thickness of the insulation layer of a refrigerated vehicle ($= 0.06m$)

q_0: respiratory heat of fruit and vegetables per unit mass ($= 324\ J/kg\cdot s$)

$d\tau$: transportation time (s)

η: The engine thermal efficiency $(= 0.40)$

q: Calorific value of fuel, $(= 43.2\ KJ/kg)$

COP: Coefficient of performance of the refrigerated vehicle $(=\ 1)$

The carbon emission from the refrigerated cooling unit is at a constant rate which one kilogram of diesel consumption emits 3.06 kilograms of carbon dioxide (3.06 CO2kg/kg). After obtaining the total fuel consumption (kg/hour), the carbon emissions rate is then multiplied by the total fuel consumption to obtain the total carbon emissions. As a result, the total emission from vehicle driving and the refrigerant unit is added together to calculate the fitness value of objective 2.

Finally, the total weight loss from fresh fruits and vegetables is calculated by multiplying the weight loss rate from Table 6 by the assigned weight of each fresh fruit and vegetable, as well as the total transportation time.

3.4.6 Proposed Heuristic Approach

This section will go over the proposed heuristic in full depth and the NSGA III algorithm framework. The problem of this study is divided into two phases, the first phase deals with the vehicle loading and layout selection problem. In phase 2, the available route is selected and assigned to each population member. Noted that the selected route must not exceed the least shelf life in each vehicle.

To illustrate the concept of the problem, in this problem, several fresh fruits and vegetables must be assigned to vehicles and transported to the destination. To avoid mixing different product types transported in the same vehicle, the vehicle layout can be divided into a limited number of compartments. The size of each compartment can be chosen at random based on the remaining requirements items within the constraints of the vehicle's layout. The route is chosen based on the least shelf-life of assigned fresh fruits and vegetables in each vehicle.

3.4.6a Initial Population

The initial population is generated at random which represent in Algorithm 1. The vehicle layout is randomly selected for the activated vehicle and the capacity of each compartment is stored in a separate sub-array. The fresh fruits and vegetables are selected at random and assigned to each available compartment and the assigned amount must not exceed the maximum capacity of each compartment. The items' attributes are then stored in a separate sub-array to select a transportation route and compute the fitness value. For transportation route selection, the route is randomly selected under the least shelf-life criteria of each vehicle. The least shelf life of each vehicle is varied based on the attribute of fresh fruits and vegetable type assigned to each vehicle. After storing the required data, the fitness value is calculated. The process repeats for 'n' replications (number of initial population).

Algorithm 1: Initial Population Generation at random for the multi-compartment vehicle loading and routing problem (MCVLRP)

Input:

V: a set of vehicle capacity

F: a set of fruit types 'i' with requirement amount (2D-array)

R: a set of the route from the selected origin to the selected destination

P: number of populations

p: population index

Output: Set of Chromosomes generated at random

Procedure:

WHILE $p < P$

    WHILE $|F| > 0$ (Length of a set of unassigned items)

        randomly selected the vehicle layout

        assigned requirement weight of the item to each compartment

        store fresh fruits and vegetable data attribute

        calculate the least shelf-life of the set of assigned fresh fruits and vegetable

        randomly select the route

        IF total transportation time > the least shelf life THEN

```
                randomly select the route
        ELSE
                assign the selected route to activate the vehicle
                store route data in sub-array
        ENDDIF
        calculate fitness value
        update remaining weight
        IF    remaining weight of each item equals 0  THEN
              remove F[ i ] from the fruits set
        IF    compartment of activated is available   THEN
              assigned requirement weight of the item to each compartment
        ELSE:
            repeat
        ENDIF
    ENDWHILE
  p= p+1
  repeat
ENDWHILE
```

3.4.6b Reference Point Generation

In this study, the set of reference points on a hyperplane is predetermined using Das and Dennis's systematic approach. A reference direction is composed of a vector that starts at the origin and connects to each of them. The total number of reference points (H) for M – objectives problem is calculated by:

$$H = \binom{M+p-1}{p} \tag{3-64}$$

where

p is the divisions considered along each objective.

M is the number of objectives

In a three-objective problem where $M = 3$, the triangle is created on the hyperplane with x-,y-,z- coordinates of (1,0,0),(0,1,0), and (0,0,1) as the apex.

The number of divisions is recommended to be more than the number of objective problems. In this case, with four divisions ($p = 4$) chosen, $H = \binom{3+4-1}{4} = 15$ or 15 reference points. For further clarification, the supplied set of reference points is purposely generated to find the near Pareto-optimal solutions corresponding to the reference points ensuring the diversification of the solutions. The set of references points are shown as follows (Table 14):

	f1	f2	f3
1	0	0	1
2	0	0.25	0.75
3	0	0.5	0.5
4	0	0.75	0.25
5	0	1	0
6	0.25	0	0.75
7	0.25	0.25	0.5
8	0.25	0.5	0.25
9	0.25	0.75	0
10	0.5	0	0.5
11	0.5	0.25	0.25
12	0.5	0.5	0
13	0.75	0	0.25
14	0.75	0.25	0
15	1	0	0

Table 14: Reference Points

3.4.6c Parent Selection and its offspring

Crossover and mutation are operations to generate new solutions for new populations (P_{t+1}). In crossover operation, the pair of chromosomes are selected to produce offspring. A crossover probability (Pc) is generated to indicate the probability of selection which is set to 1, indicating that all of the chromosomes chosen are used in reproduction (Pc=1.0). The probability of crossover selection is assigned to each parent at random from a uniform distribution (or chromosome). This study proposed an adapted crossover algorithm to generate offspring in the crossover operator, which is shown in Algorithm 2. The algorithm is based on the Exon shuffling crossover operation. The operation for proposed crossover operation is divided into two phases: combining the parent pair and reassigning the item to each activated vehicle using the First-Fit algorithm concept (FF). In phase 1, different from the traditional Exon shuffling, only the sub-string 1, vehicle layout type, the sub-string 1 of a pair of chromosomes are combined into one string as a current chromosome while other sub-strings are set to be empty. In phase 2, all fresh fruits and vegetable requirements and types are reassigned to the empty sub-string 2 and sub-string 3. If the required amount meets the existing layout capacity, under the first-fit concept, the fresh fruits and vegetables are then reassigned to the activated vehicle with the existing layout type. If the appropriate layout type is not found in the parent combination list, the available layout type (gene) is mutated to a more appropriate layout. The route is then reassigned to each gene of sub-string 4 under the shelf-life constraints. The required information data are decoded from each sub-string reassigning process, the fitness value is then calculated and stored. The procedure is repeated until all of the items have been assigned. There is also an additional string with two substrings called an unassigned string. It is used to keep the remaining item and the remaining weight. An example of data structure is shown in Figure 23. The layout type string is made up of two parents, and the assigned string is made up of three substrings: assigned weight, fruit types, and transportation route. The unassigned string is used to temporarily store the remaining item and is deleted once all items have been assigned. And the following section will go over the detailed algorithm for the reassigning process.

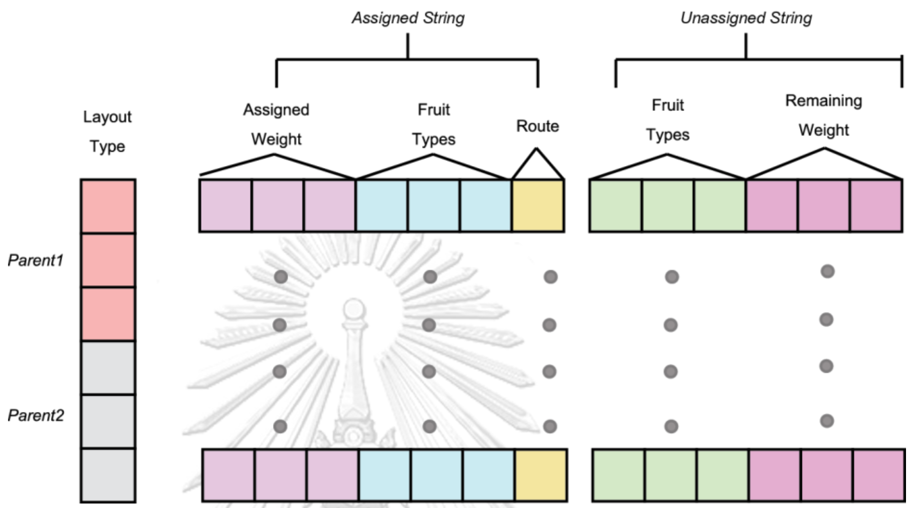


Figure 23 : Example of proposed crossover operation

Algorithm 2: Hybrid Exon Crossover for the multi-compartment vehicle loading and routing problem (MCVLRP)

Input: Population (P_t), crossover probability (Pc)

Output: Set of Offspring (Q_t)

Procedure:

1: Parent Selection

2: Assign selection probability to each chromosome $(prob)$

3: FOR each chromosome in the population

4: IF $prob \leq Pc$ THEN

5: add each chromosome to the selected parent set $(parent)$

6: ELSE

7: continue

8: ENDIF

9: ENDFOR

10: Combined pair of chromosome (vehicle sub-string)

FOR index = 0 to the size of the selected parent set $(parent)$
    IF index is an odd number
        $parent\ 1 = parent$ [index]
    IF index is even number
        $parent\ 2 = parent$ [index]
    combined parent = $parent\ 1 \cup parent\ 2$
    ENDIF
ENDFOR
FOR index = 0 to the size of the combined parent set
    Reassigned fresh fruits and vegetables to the activated vehicle (Algorithm 3 and 4)
    Generate new transportation routes at random under shelf-life constraint
    Calculate fitness
ENDFOR

3.4.6d Reassignment Operation

To explain briefly, the First-fist bin packing algorithm is adapted into this operation, instead of randomly select the vehicle layout and assigned to the vehicle, the vehicle with the current vehicle layout is chosen if its capacity structure meets the algorithm conditions. In case of none of the current vehicle, layout meets the condition, the vehicle layout is automatically changed (or mutated) to the more suitable vehicle layout regards the remaining weight of vegetable and fresh fruits requirements. The reassignment operation is divided into two phases as follows:

1. Select the most suitable vehicle layout and update the gene in sub-string 1.
2. Reassign the weight and type of fresh fruits and vegetables to each compartment of each vehicle and update the gene in sub-string 2 and sub-string 3.

Algorithm 3 gives the details of the vehicle layout type selection procedure in the reassignment operation. The vehicle layout type is selected based on the largest value of the remaining weight of vegetable and fresh fruit requirements. After each assignment, the remaining weight of vegetable and fresh fruits requirements is then updated and sorted in descending order. The problems are categorized into four cases and are defined as follows:

Case 1: The largest value is more than or equal to the compartment 1 capacity of vehicle layout 1

Case 2: The largest value is more than the compartment 2 capacity of vehicle layout 2 and less than the compartment 1 capacity of vehicle layout 1

Case 3: The largest value is more than or equal to the compartment 1 capacity of vehicle layout 2 and less than or equal to the compartment 2 capacity of vehicle layout 2

Case 4: The largest value is less than or equal to the compartment 1 capacity of vehicle layout 3

Accordingly, if the remaining weight of fresh fruits and vegetable falls in case 1 or case 2, the process is simple. If vehicle layout type 1 exists in the current combination substring 1, the gene with vehicle layout type 1 is added to the offspring and removed from the current vehicle sub-string. Otherwise, if no vehicle layout type 1 exists in the current vehicle sub-string, the current vehicle layout is automatically changed to type 1, and the current gene is removed from the current vehicle substring. The fresh fruits and vegetable weights and types are then assigned using Algorithm 4 which will be discussed further in the next section. Following that, the fresh fruit and vegetable types are simultaneously added to the gene in sub-strings 2 and 3. The remaining weight of fresh fruits and vegetables is then updated.

In Case 3, regards the current vehicle sub-string, if either vehicle layout type 1 or type 2 exists in the current combination substring 1, the gene with vehicle layout type 1 or type 2 is added to the offspring and removed from the current vehicle sub-string. On the other hand, if only a vehicle layout type 3 exists, the current vehicle layout is automatically changed to either type 1 at type 2 at random and added to the offspring. After that, the current gene is removed from the current vehicle substring. Algorithm 4 is then employed to assign the weight of fresh fruits and vegetables to each compartment. And the obtained fresh fruit and vegetable types are assigned to the gene in sub-strings 2 and 3. Then, the remaining weight of fresh fruits and vegetables is updated.

Eventually, in case 4, if vehicle layout type 3 is present in the current combination substring 1, the vehicle layout type 3 gene is added to the offspring and removed from the current vehicle substring. Otherwise, if no vehicle layout type 3 exists in the current vehicle substring, the current vehicle layout is modified to type 3, and the current gene is removed. Again, the weight assignment and item type selection are carried out by Algorithm 4 and added to the gene in sub-string2 and sub-string3.

However, there is one special condition, if the remaining weight is equal to the compartment 1 capacity of either layout type 2 or type 3, any existing layout type in the current vehicle sub-string can be added to the offspring. The process repeats until all weights are assigned to the vehicle.

Algorithm3: Vehicle Layout Type Selection

Input:

C: A combined parent (vehicle-layout substring)

V : Vehicle Type; = $\{1,2,3\}$

$comp_{ij}$: Set of layout type 'i' compartment 'j' ; $i = \{1,2,3\}$ and $j = \{1,2,3\}$

F: a set of fresh fruits and vegetable type $= \{0, \dots, f\}$

R : The set of the remaining weight of item (= $\{ r_{0f}, r_{1f}, \dots r_{kf}\}$)

k: index number $\{0,1,2, \dots, |F|\}$

W_f: Fresh fruits, and vegetable weight requirements

Output: individual off-spring Q_k

Procedure:

count = 0

WHILE count < $|F|$

    remaining set (R) is sorted in descending order

    IF $r_{0f} \geq comp_{11}$ or $comp_{22} < r_{0f} < comp_{11}$ THEN

IF layout type 1 exists in C THEN
a select current gene with layout type one is added to the vehicle substring
current gene is added to vehicle substring
current gene is removed from C
IF layout type1 is not in C THEN
select any current gene and mutated to type 1
the gene with new vehicle layout type is added to vehicle-substring
the selected current gene is removed from C
ENDIF
assign items according to Algorithm 4
add updated compartment sub-string containing weight assignment to sub-string 2
add updated compartment sub-string containing item type to sub-string 3
updated the remaining weight of fresh fruits and vegetables (R) and sorted in descending order
IF the updated remaining weight of the selected item equals 0 THEN
count = count + 1
IF $comp_{21} < r_{0f} < comp_{22}$ THEN
IF layout type 1 or type 2 exists in C THEN
select layout type between type 1 or type 2 at random
IF Type 1 is chosen THEN
a select current gene with layout type 1 is added to the vehicle substring
current gene is added to vehicle substring
current gene is removed from C
assign items according to Algorithm 4
IF Type 2 is selected THEN
a select current gene with layout type 2 is added to the vehicle substring
current gene is added to vehicle substring
current gene is removed from C
assign items according to Algorithm 4
ENDIF
IF layout type 1 and type 2 not exist in C THEN
select any current gene and mutated to type 1 or type2 at random
the mutated current gene is added to vehicle substring
the selected current gene is removed from C

IF Type 1 is chosen THEN
select current gene with layout type 1 is added to the vehicle substring
current gene is added to vehicle substring
current gene is removed from C
assign items according to Algorithm 4
IF Type 2 is selected THEN
select current gene with layout type 2 is added to the vehicle substring
current gene is added to vehicle substring
current gene is removed from C
assign items according to Algorithm 4
ENDIF
ENDIF
add updated compartment sub-string containing weight assignment to sub-string 2
add updated compartment sub-string containing item type to sub-string 3
updated remaining weight of fresh fruits and vegetables and sorted in descending order
IF Type 1 is selected and the updated remaining weight equals to 0 THEN
count = count + 1
IF Type 2 is selected and the only one updated remaining weight THEN equals to 0 THEN
count = count + 1
IF Type 2 is selected and the two updated remaining weight equals to 0 THEN
count = count + 2
ENDIF
IF $0 < r_{0f} < comp_{31}$ THEN
IF layout type 3 exists in C THEN
select current gene with layout type 3 is added to the vehicle substring
current gene is added to vehicle substring
current gene is removed from C
IF layout type 3 is not in C THEN
select any current gene and mutated to type 3
the gene with new vehicle layout type is added to vehicle-substring
the selected current gene is removed from C
ENDIF

assign items according to Algorithm 4
add updated compartment sub-string containing weight assignment to sub-string 2
add updated compartment sub-string containing item type to sub-string 3
updated remaining weight of fresh fruits and vegetables and sorted in descending order
IF only one updated remaining weight equals to 0 THEN
count = count + 1
IF two updated remaining weight equals to 0 THEN
count = count + 2
IF three updated remaining weight equals to 0 THEN
count = count + 3
ENDIF
IF $r_{0f} = comp_{31}$ or $r_{0f} = comp_{21}$ THEN
IF layout type 2 or type 3 exists in C THEN
select layout type between type 2 or type 3 at random
IF Type 2 is chosen THEN
select current gene with layout type 2 is added to the vehicle substring
current gene is added to vehicle substring
current gene is removed from C
assign items according to Algorithm 4
IF Type 3 is selected THEN
select current gene with layout type 1 is added to the vehicle substring
current gene is added to vehicle substring
current gene is removed from C
assign items according to Algorithm 4
ENDIF
IF layout type 2 and type 3 not exist in C THEN
select any current gene and mutated to type 2 or type 3 at random
the mutated current gene is added to vehicle substring
the selected current gene is removed from C
IF Type 2 is chosen THEN
select current gene with layout type 2 is added to the vehicle substring
current gene is added to vehicle substring
current gene is removed from C

```
                    assign items according to Algorithm 4
                IF   Type 3 is selected      THEN
                    select current gene with layout type 3 is added to the vehicle substring
                    current gene is added to vehicle substring
                    current gene is removed from C
                    assign items according to Algorithm 4
                ENDIF
            ENDIF
            add updated compartment sub-string containing weight assignment to sub-string 2
            add updated compartment sub-string containing item type to sub-string 3
            updated remaining weight of fresh fruits and vegetables and sorted in descending
            order
            IF      only one updated remaining weight equals to 0    THEN
                    count = count + 1
            IF      two updated remaining weight equals to 0    THEN
                    count = count + 2
            IF      three updated remaining weight equals to 0    THEN
                    count = count + 3
            ENDIF
      ENDIF
ENDWHILE
```


Algorithm 4 describes the decision-making procedure for fresh fruits and vegetable assignment in the reassignment operation. For vehicle layout type 1, the assignment pattern is straightforward; any selected item is assigned to compartment 1 since only compartment 1 is activated. There are mutual conditions between layout types 2 and 3, which are as follows:

<u>Condition 1:</u> There are at least two types of fresh fruits and vegetables remaining in the set of the remaining weight of fresh fruits and vegetable requirement.

Condition 2: Fewer than two types of fresh fruits and vegetables left in the set of the remaining weight of fresh fruits and vegetable requirement.

Condition 3: There is only one type of remaining weight of fresh fruits and vegetable requirement left.

Next, if vehicle layout type 2 is selected, the decision-making in fresh fruits and vegetable assignment is not only based on the remaining item in the list but also the fresh fruits and vegetable type selection criteria. Different from vehicle layout type 1, there are three decision-making criteria for fresh fruits and vegetable assignment which are the remaining weight, the difference in optimal temperature, and the difference in shelf life. In layout type 2, the greatest value r_{0f} is first assigned to compartment 2 of vehicle layout type 2. The procedure of fresh fruits and vegetable type selection for compartment 1 begins afterward.

For the case of more than two items contained in the list (condition 1), all mentioned criteria are applied and selected at random when more than two items are remaining on the list. The first criterion is very straightforward; the least remaining amount is selected and assigned to compartment 1. For criteria 2, the difference of optimal temperature requirement, the difference in optimal temperature is the absolute number obtained by subtracting the optimal temperature of the item type assigned to compartment 2 from the optimal temperature of each individual in the item list. The item type with the least difference in optimal temperature is selected and assigned to compartment 1. If the remaining weight is less than the capacity of compartment 1, the remaining weight is assigned. If the remaining weight is greater than or equal to the compartment 1 capacity, the weight of an assigned item is assigned full compartment 1 capacity. In the particular circumstances where the greatest value is less than or equal to compartment 1, the item is assigned to compartment 1. In Criteria 1, instead of selecting the item with the least remaining weight, the item with the second greatest remaining weight is chosen. For criteria 3, the difference in shelf life is calculated instead of optimal temperature requirements, as in the criteria 2 procedure.

Conversely, when fewer than two items are remaining on the list (Condition 2), none of the criteria are applied because the item with the highest weight remaining on the list is assigned to compartment 2 and another to compartment 1. After items are assigned to compartments 1 and 2, the weight and type of the selected item are added to sub-strings 2 and 3, and the remaining weights are sorted in descending order.

Similarly, if only one item remains (Condition 3), there are two possible ways. If the remaining weight of the last item is less than or equal to the capacity of compartment 1, the item is assigned to compartment 1, otherwise, it is assigned to compartment 2.

At last, the decision-making process of vehicle layout type three is similar to type 2. The fresh fruits and vegetable assignment for layout type 3 are differentiated into three patterns regards three different conditions. For condition 1, the largest value is assigned to compartment 1 and any two items are assigned to compartments 2 and 3 based on criteria selection, with the assigned weight not exceeding the capacity of each compartment. Different from type 2, the difference in optimal temperature and the difference in shelf life is chosen as criteria set for type 3. Again, the criterion is selected at random, the two least values are determined, and the selected items are then add assigned to compartments 2 and 3. Condition 2 assigns the largest value to compartment 1 again, and the second item in the remaining list (in descending order) to compartment 2 with a weight no greater than the compartment capacity. If only one item remains unassigned, condition 1, the unassigned item with remaining weight is assigned to compartment 1.

Algorithm4: Fruit Assignment for Each Type of Vehicle layout

Input:

V : Vehicle Type; = $\{1,2,3\}$

$comp_{ij}$: Set of layout type 'i' compartment 'j' ; $i = \{1,2,3\}$ and $j = \{1,2,3\}$

F: a set of fresh fruits and vegetable type $= \{0, \dots, f\}$

R : The set of the remaining weight of item (= $\{ r_{0f}, r_{1f}, \dots r_{kf}\}$)

k: index number $\{0,1,2, \dots, |F|\}$

W_f: Fresh fruits, and vegetable weight requirements

O_f: Fresh fruits, and vegetables optimal temperature requirements

SL_f: Fresh fruits, and vegetable weight requirements

Cr: the set of selection criteria; = $\{1,2,3\}$

Output: Items assignment

Procedure:

$w_{fmax} = r_{0f}$
IF $V = 1$ THEN
    Assign w_{fmax} to compartment 1 and a select item is $f = fmax$
IF $V = 2$ THEN
    IF more than two items remain THEN
        Assign w_{fmax} to compartment 2 and a select item is $f = fmax$
        Select selection criteria at random
        IF $Cr = 1$ THEN
            $wassgined = \min\{w_f \mid f \neq fmax\}$
            $fassigned = f$
        IF $Cr = 2$ THEN
            $diff\_T = \min\{|O_{fmax} - O_f|\}$
            $fassigned = f$
            IF $w_{fassigned}$ exceed $comp_{21}$ capacity THEN
                $wassgined$ equals to $comp_{21}$ capacity
            IF $w_{fassigned}$ less than $comp_{21}$ capacity THEN
                $wassgined$ equals to $w_{fassigned}$
            ENDIF
        IF $Cr = 3$ THEN

$diff\_SL = min\{|SL_{fmax} - SL_f|\}$

$fassigned = f$

IF $w_{fassigned}$ exceed $comp_{21}$ capacity THEN

$wassgined$ equals to $comp_{21}$ capacity

IF $w_{fassigned}$ less than $comp_{21}$ capacity THEN

$wassgined$ equals to $w_{fassigned}$

ENDIF

ENDIF

IF less than two items remain THEN

Assign w_{fmax} to compartment 2 and a select item is $f = fmax$

Assign $w_f \mid f \neq max$ to compartment 1 and a select item is f

IF only one remains THEN

Assign w_f to compartment 2 and a select item is f

ENDIF

IF $V = 3$ THEN

Assign w_{max} to compartment 1 and a select item is $f = fmax$

IF more than two items remain THEN

Select selection criteria at random

IF $Cr = 1$ THEN

FOR each fruit in fruit remaining

set $diff\_T = [|O_{fmax} - O_f|]$

ENDFOR

Sorted in increasing order and select the first two value (two least value)

$fassigned = [f_{sorted1}, f_{sorted2}]$

IF $w_{f_{sorted1}}$ exceed $comp_{32}$ capacity THEN

$wassgined1$ equals to $comp_{32}$ capacity

IF $w_{f_{sorted1}}$ less $comp_{32}$ capacity THEN

$wassgined1$ equals to $w_{f_{sorted1}}$

IF $w_{f_{sorted2}}$ exceed $comp_{33}$ capacity THEN

$wassgined2$ equals to $comp_{32}$ capacity

IF $w_{f_{sorted2}}$ less $comp_{33}$ capacity THEN

$wassgined2$ equals to $w_{f_{sorted2}}$

ENDIF
IF $Cr = 2$ THEN
FOR each fruit in fruit remaining
set $diff_{SL} = [|SL_{fmax} - SL_f|]$
ENDFOR
Sorted in increasing order and select the first two value (two least value)
$fassigned = [f_{sorted1}, f_{sorted2}]$
IF $w_{f_{sorted1}}$ exceed $comp_{32}$ capacity THEN
$wassgined1$ equals to $comp_{32}$ capacity
IF $w_{f_{sorted1}}$ less $comp_{32}$ capacity THEN
$wassgined1$ equals to $w_{f_{sorted1}}$
IF $w_{f_{sorted2}}$ exceed $comp_{33}$ capacity THEN
$wassgined2$ equals to $comp_{32}$ capacity
IF $w_{f_{sorted2}}$ less $comp_{33}$ capacity THEN
$wassgined2$ equals to $w_{f_{sorted2}}$
ENDIF
ENDIF
IF less than two items remain THEN
Assign the next two remaining items (i.e. r_1) to compartments 2 and the assigned weight not exceeding the capacity of each compartment and a select item is f
IF only one remains THEN
Assign w_f to compartment 1 and a select item is f
ENDIF
ENDIF

3.4.6e Parent Mutation Operation

To escape from local optimal, the parent mutation operation is divided into two phases (Algorithm5). In the first phase, only the parents with a chromosome length that exceeds the minimum number of vehicles used (chromosome length) are selected. The vehicle layout selection and item reassignment procedures are repeated using the aforementioned proposed reassignment operation. The route chromosome is then generated at random for each vehicle. In Phase 2, after randomly selecting a transportation route, the mutation operation on the route chromosome of the parent population (P_t) is proposed to expand the search space. A random number between and one are generated and assigned to the route chromosome for a possibility of mutation. The route chromosome is mutated if the number is lower than or equal to the specified mutation probability (e.g. 0.25). The proposed mutation operations are divided into five cases and the gene mutates based on three different factors which are total time, total distance, and total cost. The choice of mutation operation and factor selection is randomly selected. The mentioned five cases are described as follows:

1.) Reassign all routes to the selected chromosome.

The gene of the route sub-string in a particular chromosome is reassigned based on a factor. The mutation factor is chosen at random, and then the route is selected. If the route is reassigned based on transportation time, the existing route with a transportation time less than the route in the current gene is recorded and randomly selected. Each gene is then replaced by a new route. Similar to the procedures above, instead of transportation time, the current route is replaced based on either total transportation distance or cost.

2.) Mutate some genes on the selected chromosome.

Some genes in sub-string 4 of each chromosome are mutated in this case. Each gene is assigned a mutation probability, and if the probability is less than the gene permutation probability (Pmg), the gene is mutated. Again, the mutation factor is chosen at random, and the specific gene in sub-string 4 is mutated as explained above.

3.) Reassign all routes to the route with the least total time to the selected chromosome.

4.) Reassign all routes to the route with the shortest distance to the selected chromosome.

5.) Reassign all routes to the route with the lowest total cost to the selected chromosome.

Algorithm 5: Mutation Operation for the multi-compartment vehicle loading and routing problem (MCVLRP)

Input:

P_t: Population

$Route$: a set of the transportation route

$Cond$: a set of conditions $= \{1,2,3,4\}$

$factor$: a set of factors$= \{1,2,3\}$

Pm: Mutation probability

Pmg: Gene mutation probability

Ouput: Mutated parent population (P_t^*)

Procedure:

Count the number of the vehicle used in each chromosome and define the
FOR each chromosome in the population
    IF the length of the chromosome is more than the minimum vehicle used THEN
        reassign using the reassignment operator (Algorithm 3 and Algorithm 4)
        randomly generated transportation route
        updated the chromosome in P_t
    ENDIF
ENDFOR

Assign selection probability to each chromosome at random ($prob$)
FOR each chromosome in population
    IF $prob \geq Pm$ THEN
        select the condition at random ($Cond$)
        select the factors for mutation at random ($factor$)
        IF $Cond = 1$ THEN
            FOR each gene of chosen chromosome
                IF $factor = 1$
                    Randomly select the route that contains less total transportation time than the current route
                    Update route gene
                IF $factor = 2$
                    Randomly select the route that contains less total transportation distance than the current route
                    Update route gene
                IF $factor = 3$
                    Randomly select the route that contains less total transportation cost than the current route
                    Update route gene
                ENDIF
            ENDFOR
        IF $Cond = 2$ THEN
            assign probability to each gene ($prob\_g$) of chosen chromosome
            FOR each gene of chosen chromosome
                IF $prob\_g \geq Pmg$ THEN
                    select the factors for mutation at random
                    IF $factor = 1$
                        Randomly select the route that contains less total transportation time than the current route
                        Update route gene
                    IF $factor = 2$
                        Randomly select the route that contains less total transportation distance than the current route
                        Update route gene

```
                        IF  factor = 3
                            Randomly select the route that contains less total transportation
                            cost than the current route
                            Update route gene
                        ENDIF
                ENDIF
            ENDFOR
        IF  Cond = 3    THEN
            define least total transportation time from routing data set
            FOR each gene of chosen chromosome
                mutate all the transportation route to route with least total transportation time
                Update route gene
            ENDFOR
        IF  Cond = 4    THEN
            define least total transportation distance from routing data set
            FOR each gene of chosen chromosome
                mutate all the transportation route to route with least total transportation
                distance
                Update route gene
            ENDFOR
        IF  Cond = 5    THEN
            define the least total transportation cost from the routing data set
            FOR each gene of chosen chromosome
                mutate all the transportation routes to the route with the least total
                transportation cost
                Update route gene
            ENDFOR
        ENDIF
ENDFOR
Return updated population (mutated population, P_t^*)
```

3.4.6f Fast Non-dominated Sorting

The mutated population P_t^* with size, N is combined with the offspring Q_k with size $\frac{N}{2}$ as a current population size of N+$\frac{N}{2}$ solutions. Individuals are sorted into different non-dominance levels using the fast non-dominated sorting approach proposed by Deb et al. (2002) and illustrated in Algorithm 6. The set of solutions for the first non-dominated front is identified by comparing each solution in the population to determine if it is dominated. Accordingly, to obtain another set of solutions for the next non-dominated front, the solutions from the first front are temporarily excluded and the procedure described above is repeated.

For a more detailed explanation, non-dominated fronts are obtained one after the other, beginning with the one with the highest rank and ending with the one with the lowest rank. The following is the procedure for generating the Pareto front set of a population of N+$\frac{N}{2}$ solutions:

1. Combine the parent and offspring populations to form a new population P'.
2. Create an empty front F_i. Remove the current solution S_i from P' and add S_i into F_i.
3. Compare S_i with all solutions (S_j) in P' with where $i, j \in \{1, \dots, N + \frac{N}{2}\}$ and $i \neq j$
   - If S_i dominates S_j, S_j remains in P' and proceed to the next step.
   - If S_j is non-dominated by S_i, S_j is removed from P' and added into F_i.
   - If S_i is dominated by S_i, remove S_i from F_i and include into to P'.
   - If S_i and S_j are non-dominated, exclude from P' and add to F_i
4. When all the comparisons in the preceding procedure have been completed, the solutions in F_i form the non-dominated front. The remaining solutions in the population (P') are used to generate the next fronts with the same procedure until all individuals are assigned to the fronts.

Algorithm 6: Fast Non-Dominated Sorting proposed by Deb et al. (2002)

```
fast-non-dominated-sort(P)
for each p ∈ P
  S_p = ∅
  n_p = 0
  for each q ∈ P
    if (p ≺ q) then                  If p dominates q
      S_p = S_p ∪ {q}                Add q to the set of solutions dominated by p
    else if (q ≺ p) then
      n_p = n_p + 1                  Increment the domination counter of p
  if n_p = 0 then                    p belongs to the first front
    p_rank = 1
    F_1 = F_1 ∪ {p}
i = 1                                Initialize the front counter
while F_i ≠ ∅
  Q = ∅                              Used to store the members of the next front
  for each p ∈ F_i
    for each q ∈ S_p
      n_q = n_q − 1
      if n_q = 0 then                q belongs to the next front
        q_rank = i + 1
        Q = Q ∪ {q}
  i = i + 1
  F_i = Q
```

3.4.6g Adaptive Normalization

After a set of solutions are obtained from non-dominated sorting, the fitness value from the three objectives is normalized using the adaptive normalization method. The normalization procedure is used to scalarize the fitness value of each objective. To normalize the values, the ideal point of the population (S_t) is first determined by identifying the smallest value of each objective function $i = \{1,2,3\}$ which results in a set of ideal points for each population, denoted as $\overline{Z} = (Z_1^{min}, Z_2^{min}, Z_3^{min})$. Next, the translated objective $(f_i'(\mathrm{x}))$ is introduced. The translated objective is calculated by subtracting each objective value $f_i(x)$ by the minimum value of each objective (Z_i^{min}) where $f_i'(\mathrm{x}) = f_i(x) - Z_i^{min}$.

The three extreme points on the three objectives axis are then obtained from the achievement scalarization function (ASF) with minimum weight vector (w) where the value is a small number of 10^{-6}. The normalized objective ($f_i^n(x)$) is then equal to $\frac{f_i'(x)}{a_i - Z_i^{min}}$ where a_i denote as the intercept of the hyperplane with axis directions and $(a_1, a_2, \ldots, a_i)$ is simply set to $(Z_1^{max}, Z_2^{max}, Z_3^{max})$.

Algorithm 7: Adaptive normalization

Input:

M: number of the objective function

$f_i(x)$: fitness value of the objective function

Output: $f_i^n(x)$ normalized fitness value

Procedure:

1: FOR i =1 to M

2: Compute the minimum value of each fitness value (Z_i^{min})

3: Compute the extreme point of each fitness value (Z_i^{max})

4: Normalize the fitness value of each objective $\frac{f_i(x) - Z_i^{min}}{a_i - Z_i^{min}}$

5: ENDFOR

3.4.6h Association Operation

In association operation, with normalized objective value, each point in the objective space is connected to the reference point through the reference line. The reference line is constructed by connecting each reference point to the origin of the hyperplane. The projection distance on the reference line ($d_{j,1}$) and the perpendicular distance between the reference points and the perpendicular point ($d_{j,2}$) are calculated. The reference point where the reference line is closest to the population member in the normalized objective space is considered associated with the population member.

Algorithm 8: Associate Operation

Input:

S_t: individuals in each front

t : number of the front set

$f_i^n(x)$ normalized fitness value

w^j: Reference points

j: number of reference points

Output: closet reference point to each population $\pi(s)$, minimum distance $d(s)$ where $s \in S_t$

Procedure:

FOR each s in S_t

  FOR each w in w^j

    calculate $d_{j,1}(x) = \frac{\| f^n(x)^T w^j \|}{\| w^j \|}$

    calculate $d_{j,2}(x) = \left\| (f^n(x) - dj, 1(x)) \frac{w^j}{\| w^j \|} \right\|$

  ENDFOR

  Assign $\pi(s) = w: argmin(w, s, d_{j,2}(x))$

  Assign $d(s)$ = $d_{j,2}(s, \pi(s))$

ENDFOR

3.4.6i Niche Count Operation

The niche preservation operation counts the reference points that are associated with the population member. All members from the $(F_1, F_2 \ldots\ F_{l-1})$ are automatically added to the new population (P_{t+1}). The niche count be denoted as ρ_j for the *j-th* sub-region. Niche count is firstly counted from current (P_{t+1}) for each reference point. The remaining slots are filled by selecting members from F_l. Next, the set of the reference point with minimum niche count (ρ_j) are sorted and listed. If there are multiple reference points with the same niche count, one reference point is randomly selected. If $\rho_j = 0$, there are two possible scenarios which are (1) There are members from F_l associate with the selected reference point. Hence, the nearest perpendicular distance is added to P_{t+1}. (2) There is no member from F_l associate with the selected reference point. Hence, the member from F_l is chosen at random and added to P_{t+1}. If $\rho_j \geq 1$, This means that the reference points are already associated with members of the population. In this case, the member from F_l is chosen at random and added to P_{t+1}. Next, niche counts are updated and repeated for a total of K times to fill the remaining slot. And P_{t+1} is completely updated for a generation 't' .

CHAPTER 4

RESULT AND DISCUSSION

In this section, a detailed description of the different instance data sets is introduced and used to verify the effectiveness of the proposed hybrid NSGA-III for multi-compartment vehicle loading, and transportation routing problems (MCVLRP). And the computational experiments were carried out to evaluate the performance of the proposed approach. The optimal solution obtained from single objective optimization by solving the proposed nonlinear programming (NLP) model presented in Chapter 3 is utilized as a benchmark for validating the proposed algorithm's performance. In this study, attempts are made to solve the NLP model using the software IBM-ILOG CPLEX 12.6 solvers. The proposed heuristic is implemented, replicated, and modified in Jupyter Notebook (in Python language). All experiments are computed and evaluated on PC with Intel ® Core™ i5-4460 CPU@3.20GHZ and RAM of 24 GB.

4.1 Instance Generation

The test data set is introduced and used to validate the effectiveness of the hybrid NSGA-III. The data set is named '$t - w$', where 't' represents the types of fresh fruits and vegetables and 'w' defined the required weight of each fresh fruit and vegetable type. As mentioned in Chapter 3, Table 15 shows three, five, and eight different types of fresh fruits that have been pre-selected as small, medium, and large cases for the algorithm's performance evaluation, respectively. Following that, the required weight (w) of each fresh fruit and vegetable type is randomly selected from a range of 10,000 to 50,000 kgs for small, medium, and large instances.

Fruit Types (t)	Optimal temperature (℃)	Weight loss rate (kg/hour)	Shelf life (hours)
Durian	14	0.000625	336
Longan	5	0.000321	672
Mangosteen	13	0.000417	672

(a) Three pre-selected fresh fruits for small instances

Fruit Types (t)	Optimal temperature (℃)	Weight loss rate (kg/hour)	Shelf life (hours)
Durian	14	0.000625	336
Longan	5	0.000321	672
Mangosteen	13	0.000417	672
Mangoes	10	0.000167	504
Bananas	13	0.000417	1008

(b) Five pre-selected fresh fruits for medium instances

Fruit Types (t)	Optimal temperature (℃)	Weight loss rate (kg/hour)	Shelf life (hours)
Durian	14	0.000625	336
Longan	5	0.000321	672
Mangosteen	13	0.000417	672
Mangoes	10	0.000167	504
Bananas	13	0.000417	1008
Pineapple	10	0.000279	1008
Citrus	13	0.000792	672
Lychee	5	0.000275	672

(c) Eight pre-selected fresh fruits for large instances

Table 15: Pre-selected fresh fruits type for small, medium, and large instances

4.2 Experimental Design for Algorithm's Performance Evaluation

It should be noted that the number of non-dominated solutions is used to assess the performance of the algorithm. The impact of variation in population size and number of generations on the proposed hybrid NSGA-III is studied and evaluated in this section by comparing the results obtained from 12 sample runs on the small instance with one data set to see if there is an improvement in the average number of non-dominate solutions in the Pareto front when increasing the population size and the number of evolution iterations. According to Reed et al. (2003), the parameters are set as follows: the initial population is set to 60, 120, 240, and 480, and the number of evolution iterations is set to 100, 200, 500, and 1000 for each iteration.

The results from Table 16 clearly show that a larger population size with a smaller number of generations generates a greater average number of solutions in less computational time than a smaller population size with a larger number of generations. For example, a population size of 120 with 1000 generations yields approximately 15 solutions in 1 hour and 47 minutes. While a population of 240 with 100 generations can generate approximately 17 solutions in 35 minutes. As a result, the nondominated set grew by two members, about a 13.33% increase, with less total runtime. However, at the population size of 480, doubling the population size has only a minor effect on the number of solution improvements. Regardless of the computational result, a population size of 480 with 100 generations appears to be a good combination because the computational time begins to significantly increase after 100 generations with a small improvement in the average number of non-dominated solutions. Thus, an average of 21 solutions are generated within approximately 2 hours and 25 minutes. In another word, using a population size of 480 with 100 generations increased the nondominated solution by 4 solutions, resulting in a 23.5294 percent increase.

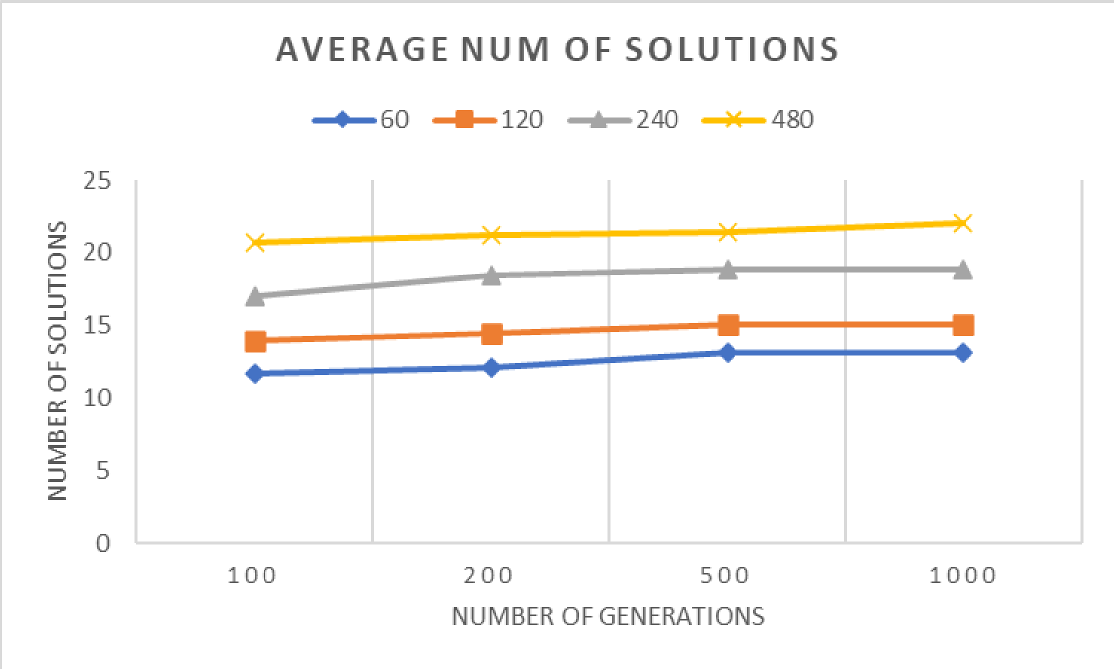
In Figure 24, the graph depicts the results in varying population sizes and generations to provide a clearer picture. The results show a small improvement in the same population size with a different number of generations, and the runtime increased dramatically as the number of generations increased, notably for population sizes of

480. With a significant increase in computational runtime, a stop criterion is suggested to reduce runtime because it is not necessary to run up to 1000 generations to generate better performance. Furthermore, the results of combining 480 population size with 100 generations, in which an average of 21 solutions are generated in approximately 2 hours and 25 minutes, are used as the benchmark for selecting the stop criterion for small instances and determining whether there is an improvement from using a stop criterion.

Population Size (N)	Generation (t)															
	100				200				500				1000			
60	11	11	12	11.66667	14	11	11	12.08333	12	13	12	13.08333	12	12	13	13.08333
	12	12	11		11	12	12		13	15	13		13	13	14	
	12	12	12		12	12	11		13	12	13		13	13	13	
	11	12	12		13	14	12		14	13	14		13	14	14	
120	18	13	14	13.91667	15	15	14	14.5	15	15	14	15.08333	15	15	15	15.08333
	12	14	15		14	15	14		15	17	15		14	16	15	
	12	14	14		14	13	14		14	15	14		15	15	14	
	14	14	13		16	15	15		17	14	16		15	17	15	
240	24	17	16	17	23	19	18	18.41667	18	19	20	18.83333	25	19	20	18.91667
	16	16	17		16	18	18		19	18	19		18	19	18	
	15	17	16		18	18	19		18	17	21		18	17	17	
	16	16	18		17	18	19		20	20	17		18	19	19	
480	20	23	20	20.75	31	21	20	21.25	25	20	21	21.41667		20	19	21.90909
	24	19	19		20	21	20		24	21	22		21	23	24	
	25	20	22		22	21	19		21	19	20		24	22	20	
	20	19	18		21	19	20		20	22	22		22	22	24	

Population Size (N)	Generation (t)			
	100	200	500	1000
60	2 min 43 s	4 mins 55s	13 mins 41 s	23 min 14 s
120	10 min 43 s	21 mins 26 s	53 mins 55 s	1 hour 47 min 32 s
240	35 mins	1 h 15 min 8s	3h 6 min 15s	6 h 15 mins 20 s
480	2 h 25m 29 s	4 h 17 min 32s	6 h 21 min 28 s	15 h

Table 16: Results from running 12 iterations on one data set with different population size and the number of generations



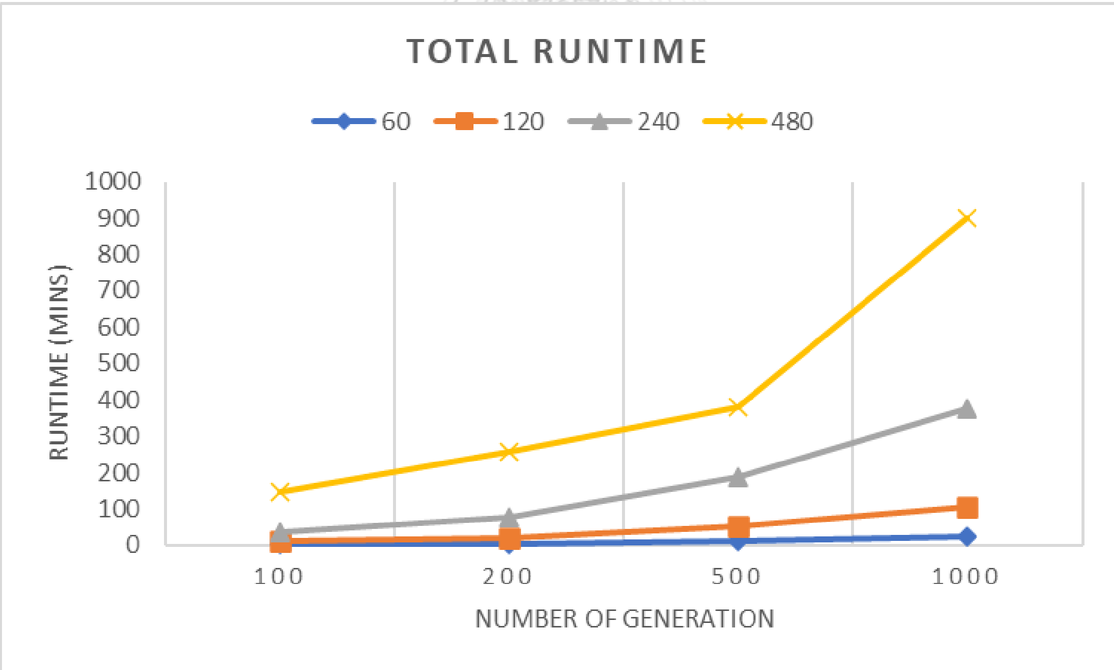


Figure 24: Average number of non-dominated solutions and runtime for 12 sample runs for small instances.

Due to the long runtime from the previous computational result, non-improvement in the number of solutions is introduced as a stop criterion in this study to reduce the computational runtime while maintaining solution quality, in terms of the number of non-dominated solutions. For a more detailed explanation, the algorithm is terminated if there is no improvement in the number of non-dominated solutions after 't' generations. For example, if the total number of solutions found over the next hundred generations does not improve, the algorithm will terminate and return the maximum number of solutions found with the non-dominated solution set.

The effect of population size variation and the stopping criterion on allocating three, five, and eight different types of fresh fruits is then thoroughly investigated and discussed. The stop criterion in this computational experiment starts at 50 non-improvement generations and gradually increases by 50 until it reaches 500. Apart from that, the population size is limited to 60, 120, 240, and 480. In addition, the experiment uses a single data set with 12 iterations to determine the average number of non-dominated solutions. The appropriate population size and stopping criterion for allocating three, five, and eight different types of fresh fruits are then suggested.

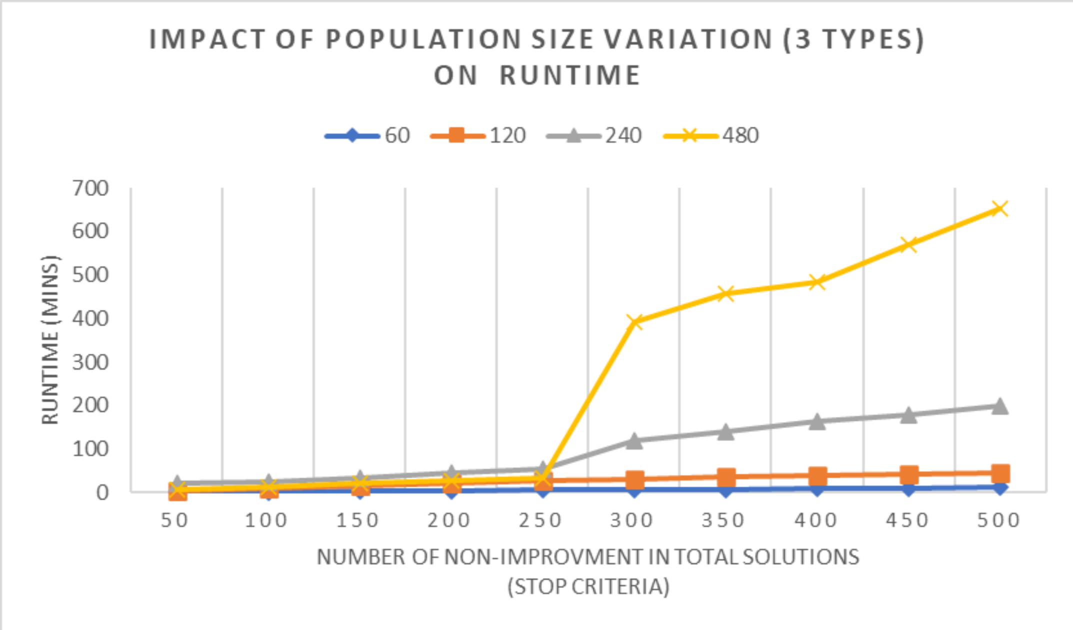
The computational experiment begins with the problem of allocating three different types of fresh fruits. The graph in Figure 25 shows that the population size of 240 outperforms other population sizes. The number of non-dominated solutions reaches a steady state after limiting the generation of non-improvement solutions to 150. However, the computational starts to have a significant increase after setting the generation of non-improvement solutions to 250. When the number of solutions does not improve after 150 generations, a population size of 240 generates approximately 21 solutions within 33 minutes after the algorithm terminates. While a population of 480 takes approximately 2 hours and 20 minutes to generate 20 solutions after running for 100 generations. As a result, a population size of 240 is recommended, with non-improvement solutions limited to 150 generations. According to the results, it is possible to conclude that introducing a stop criterion reduced computational time while maintaining algorithm performance.

Following that, the medium and large instances are tested. For medium instances, in Figure 25, the 240 population size tends to generate the same number of non-dominated solutions as the 480 population size after 50 non-improvement generations, but the 240 population size appears to outperform the 480 population size afterward. Besides, the computational time of the 480 population size increased significantly without any improvement in the number of non-dominated solutions. As a result, the population size of 240 outperformed all other population sizes. After limiting the generation of non-improvement solutions to 150 for a population of 240, the number of nondominated solutions becomes steady.

Additionally, the results from allocating three and five types of fresh fruit show that doubling the population size generate improvement in a number of non-dominated solutions and yields diminishing points at the population size of 480 (Figures 25 and 26). However, this is not the case for the allocation of the eight types of fresh fruit.

For large instances, even though 240 and 480 population sizes generate the same number of non-dominated solutions after 50 non-improvement generations in both five and eight-type allocation cases, the 480 population size outperforms the 240 population size in the eight-type allocation case (Figure 27). However, the generated number of solutions has a slight improvement after setting the generation of non-improvement solutions to 100. In terms of computational time, the runtime increased slightly and dramatically after limiting the generation of non-improvement solutions to 300, and the runtime then became constant. As a result, a 480 population size is used in this study, with 300 non-improvement generations chosen as a stop criterion. The runtime of 300 for non-improvement generation is deemed acceptable.

In conclusion, the appropriate population size for allocating three types of fresh fruits is 240, and the algorithm terminates after 150 generations if the number of solutions does not improve. While a population size of 240 is recommended for five different types of fresh fruits, the algorithm terminates after 150 generations if no better number solution is found. Finally, for eight types of fresh fruit allocation, a population size of 480 is recommended, with a stop criterion of 300 generations of non-improvement total solutions. And these rules will be applied in the following computational section of this study.



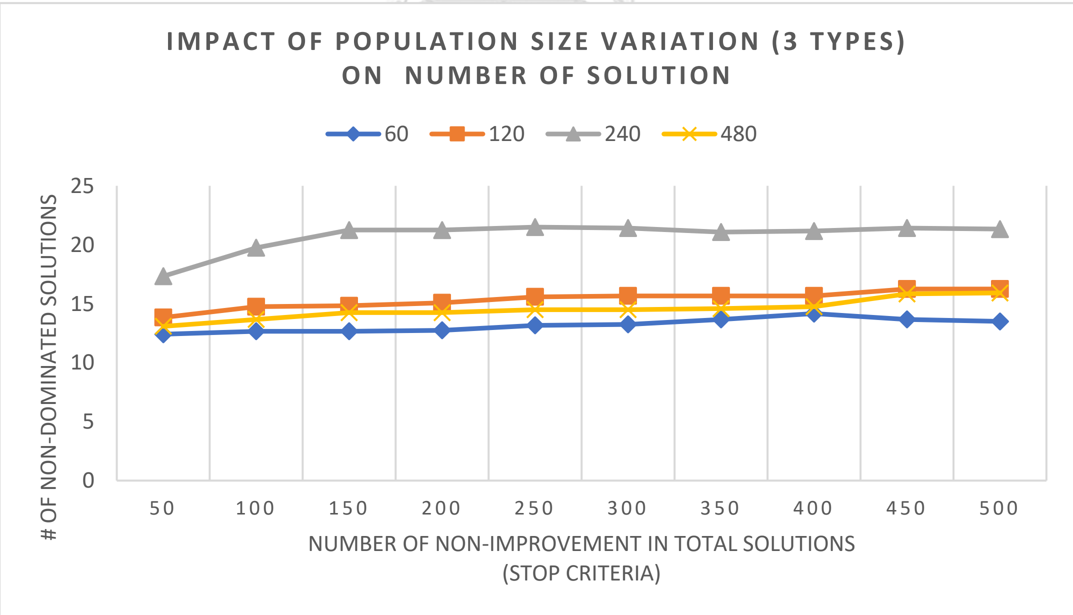


Figure 25: Impact of population size variation on the number of solutions and runtime (3 types)

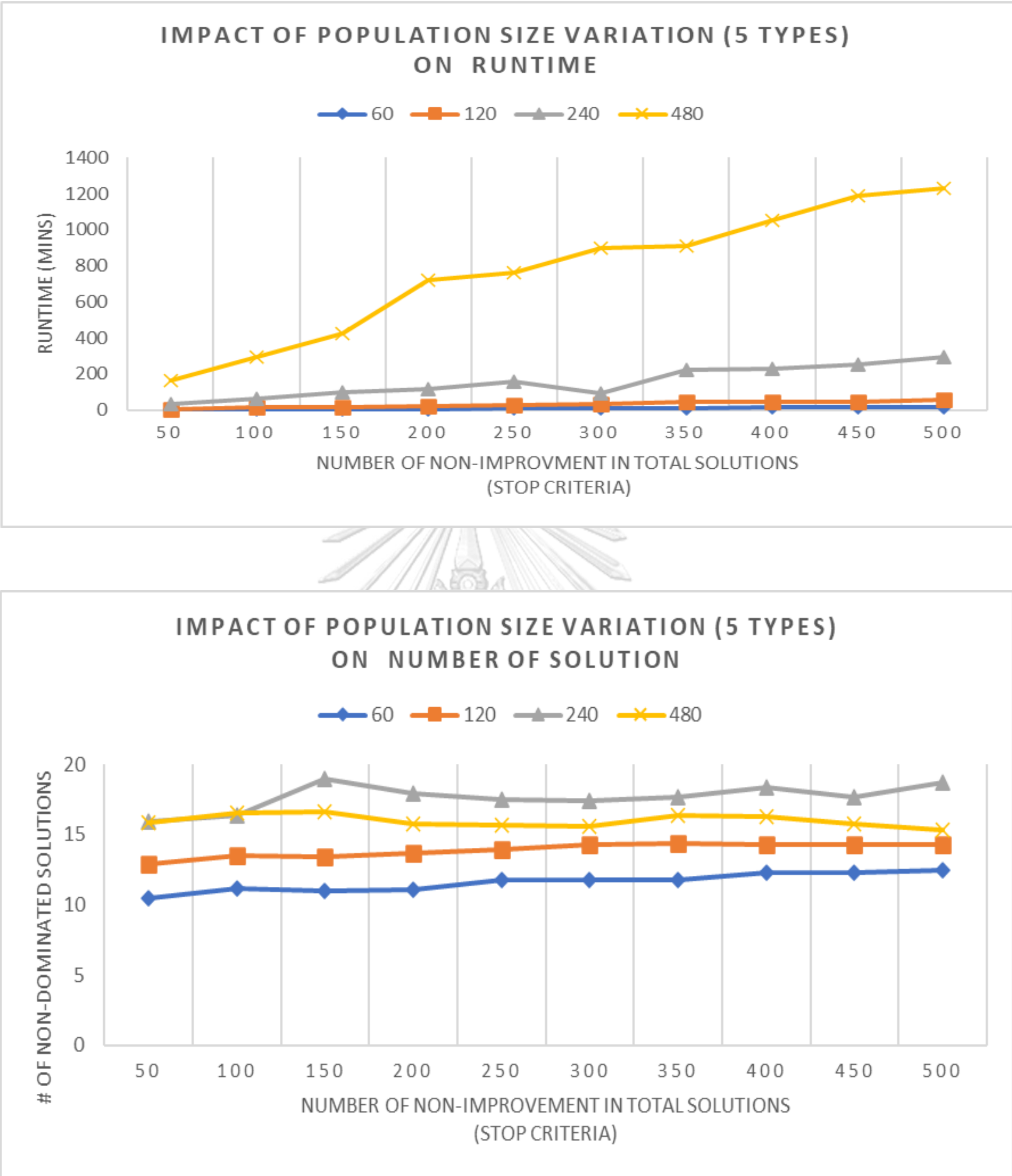


Figure 26: Impact of population size variation on the number of solutions and runtime (5 types)

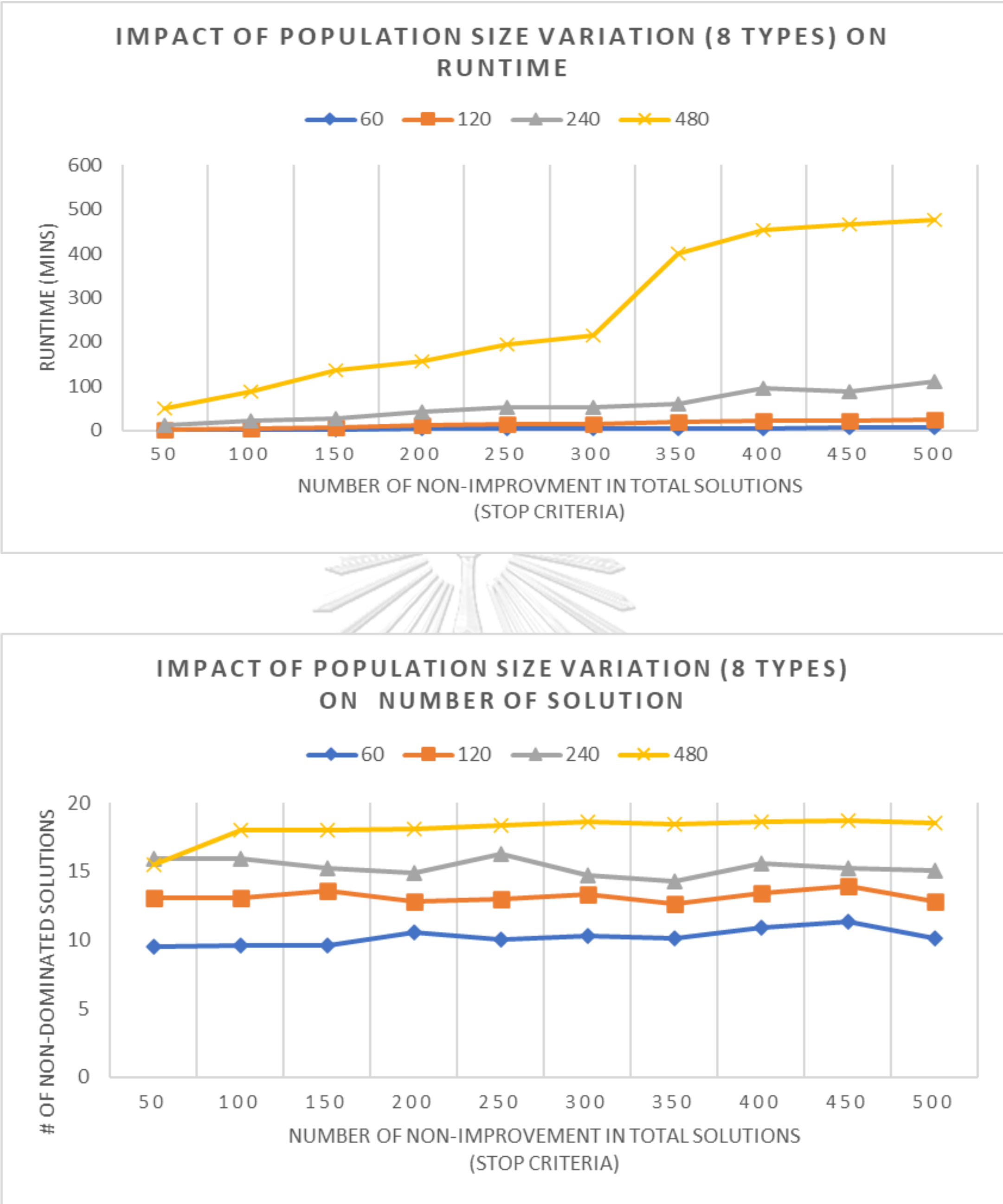


Figure 27: Impact of population size variation on the number of solutions and runtime (8 types)

4.3 Single Objective Optimization Computational Experiment

In this study, the single objective optimization computational method is used to validate the MCVLRP model, and the CPLEX results are used as a benchmark for the proposed hybrid NSGA-III effectiveness verification. As a result, the optimal value of each objective function as well as the computational time generated by the NLP model and hybrid NSGA-III are compared. This comparison computation experiment, however, can only be carried out in small instances of problems due to CPU memory limitations. The error occurs when implementing the medium and large instances which are known as out-of-memory (OOM). This indicates that there is insufficient virtual memory available and that the computer is unable to allocate additional memory (Molina and Mishra, 2013). Therefore, only small instances are investigated in this computational comparison. The twelve data sets of three types of fresh fruits were chosen at random from a range of 10,000 to 50,000 kgs and tested on a single run, with the total shipment in each case not exceeding 100,000 kilograms.

According to the previous computational experiment, a population of 240 with suggested stopping criteria is used in this experiment as well. When no improvement number is found in the solution after 150 generations, the algorithm is terminated. The optimal results from CPLEX and the best solutions based on each objective from the proposed heuristic are shown in Table 17 and the sample of the Pareto fronts obtained from the first four instances is displayed in Figure 28. The results demonstrate that the proposed heuristic successfully obtained the optimal solution regards the optimal solution from CPLEX.

As single objective optimization is employed to find an optimal solution, this means not all objective value is optimal. By comparing the result based on the objective1 optimal solution, the proposed heuristic is able to find a better solution with the same minimum cost. For example, in Instance 1, with the same minimum cost of 9,360 USD, the total carbon emissions of 19,870,000 kgCo2 are emitted from CPLEX computational result meanwhile the total carbon emissions of 17,953,572 kgCo2 are found from the proposed heuristic.

Furthermore, the results show that the vehicle type selection and loading allocation influence both total cost and total carbon emissions. For example, the total cost from the objective 2 optimal value solution is lower than the total cost from objective 3 optimal value from CPLEX which is caused by utilizing the different vehicle types. In other words, the partition adjustment cost is what causes the different total costs. The set of vehicle types in the optimal value solution for objective 2 is [1,1,1,2] whereas the set of vehicle types in the optimal value solution for objective 3 is [1,3,2,2].

Instances	Method	Result	Obj1 Value	Obj2 Value	Obj3 Value	Vehicle Used	Runtime
1	CPLEX	Objective 1	**9,360**	19,870,000	9,899	4	0.03 min
		Objective 2	24,700	**3,594,200**	1,965	4	4.15 min
		Objective 3	24,740	4,866,900	**1,965**	4	1.40 min
	Heuristic	Num of solutions	29				
		Objective 1	9,360	17,953,572	9,899	4	10 min
		Objective 2	24,700	3,594,231	1,965	4	
		Objective 3	24,700	3,594,231	1,965	4	
		Average Obj	19,083	13,289,430	5,666		
		SD Obj	4,696	7,460,052	2,986		
	Differences	Objective 1	-	1,916,428	-		
		Objective 2	-	31	-		
		Objective 3	40	1,272,669	-		
	% Differences	Objective 1	-	10	-		
		Objective 2	-	0	-		
		Objective 3	0	30	-		
2	CPLEX	Objective 1	**9,360**	19,870,000	9,479	4	0.03 min
		Objective 2	24,700	**3,594,200**	1,882	4	1.41 min
		Objective 3	24,820	6,854,200	**1,882**	4	3.36 min
	Heuristic	Num of solutions	32				
		Objective 1	9,360	17,953,557	9,479	4	9.28 min
		Objective 2	24,700	3,594,228	1,882	4	
		Objective 3	24,700	3,594,228	1,882	4	
		Average Obj	18,649	12,766,579	5,260		
		SD Obj	5,076	7,308,248	2,861		
	Differences	Objective 1	-	1,916,443	-		
		Objective 2	-	28	-		
		Objective 3	120	3,259,972	-		
	% Differences	Objective 1	-	10	-		
		Objective 2	-	0	-		
		Objective 3	0	62	-		

Table 17: Result comparison obtained from CPLEX and proposed heuristic

Instances	Method	Result	Obj1 Value	Obj2 Value	Obj3 Value	Vehicle Used	Runtime
3	CPLEX	Objective 1	**9,360**	19,739,000	9,399	4	0.04 min
		Objective 2	24,700	**3,519,100**	1,866	4	4.33 min
		Objective 3	24,800	6,239,100	**1,866**	4	9.11 min
	Heuristic	Num of solutions	29				
		Objective 1	9,360	18,713,527	9,399	4	7.28min
		Objective 2	24,700	3,519,054	1,866	4	
		Objective 3	24,700	3,519,054	1,866	4	
		Average Obj	18,666	13,378,995	5,454		
		SD Obj	5,014	9,546,683	2,894		
	Differences	Objective 1	-	1,025,473	-		
		Objective 2	-	46	-		
		Objective 3	100	2,720,046	-		
	% Differences	Objective 1	-	5	-		
		Objective 2	-	0	-		
		Objective 3	0	56	-		
4	CPLEX	Objective 1	**9,360**	19,111,000	8,769	4	0.05
		Objective 2	24,700	**3,240,200**	1,741	4	25.18 min
		Objective 3	24,760	4,951,800	**1,741**	**4**	21 min
	Heuristic	Num of solutions	32				
		Objective 1	9,360	17,954,329	8,769	4	9.57 min
		Objective 2	24,700	3,240,263	1,741	4	
		Objective 3	24,700	3,292,963	1,741	4	
		Average Obj	17,870	12,292,813	5,062		
		SD Obj	5,343	6,831,421	2,568		
	Differences	Objective 1	-	1,156,671	-		
		Objective 2	-	63	-		
		Objective 3	60	1,658,837	-		
	% Differences	Objective 1	-	6	-		
		Objective 2	-	0	-		
		Objective 3	0	40	-		
5	CPLEX	Objective 1	**9,320**	15,308,000	10,012	4	0.03min
		Objective 2	24,660	**3,066,300**	1,988	4	0.39 min
		Objective 3	24,840	6,329,800	**1,988**	4	0.49 min
	Heuristic	Num of solutions	31				
		Objective 1	9,320	15,294,281	10,012	4	7.37 min
		Objective 2	24,660	3,066,332	1,988	4	
		Objective 3	24,660	3,066,332	1,988	4	
		Average Obj	18,049	10,340,861	5,387		
		SD Obj	5,112	6,053,513	2,848		
	Differences	Objective 1	-	13,719	-		
		Objective 2	-	32	-		
		Objective 3	180	3,263,468	-		
	% Differences	Objective 1	-	0	-		
		Objective 2	-	0	-		
		Objective 3	1	69	-		

Table 17: Result comparison obtained from CPLEX and proposed heuristic (Cont.)

Instances	Method	Result	Obj1 Value	Obj2 Value	Obj3 Value	Vehicle Used	Runtime
6	CPLEX	Objective 1	**9,360**	20,325,000	9,041	4	0.03
		Objective 2	24,700	**3,609,500**	1,795	4	59 min
		Objective 3	24,800	5,858,400	**1,795**	4	0.22min
	Heuristic	Num of solutions	31				
		Objective 1	9,360	18,030,482	9,041	4	4 min
		Objective 2	24,700	3,609,499	1,795	4	
		Objective 3	24,700	3,609,499	1,795	4	
		Average Obj	18,187	12,805,965	5,308		
		SD Obj	5,083	8,506,364	2,828		
	Differences	Objective 1	-	2,294,518	-		
		Objective 2	-	1	-		
		Objective 3	100	2,248,901	-		
	% Differences	Objective 1	-	12	-		
		Objective 2	-	0	-		
		Objective 3	0	48	-		
7	CPLEX	Objective 1	**9,320**	13,334,000	7,359	4	0.02min
		Objective 2	24,660	**2,674,600**	1,461	4	6.45 min
		Objective 3	24,780	4,792,500	**1,461**	4	29 min
	Heuristic	Num of solutions	34				
		Objective 1	9,320	13,321,123	7,359	4	10.40min
		Objective 2	24,660	2,674,638	1,461	4	
		Objective 3	24,660	2,674,638	1,461	4	
		Average Obj	18,982	10,205,002	3,972		
		SD Obj	5,162	5,979,124	2,284		
	Differences	Objective 1	-	12,877	-		
		Objective 2	-	38	-		
		Objective 3	120	2,117,862	-		
	% Differences	Objective 1	-	0	-		
		Objective 2	-	0	-		
		Objective 3	0	57	-		
8	CPLEX	Objective 1	**9,360**	16,529,000	7,636	4	0.04 min
		Objective 2	24,700	**2,840,900**	1,516	4	3.30 min
		Objective 3	24860	7243300	**1515.8**	4	15 min
	Heuristic	Num of solutions	28				
		Objective 1	9,360	14,158,452	7,636	4	11.38 min
		Objective 2	24,700	2,840,858	1,516	4	
		Objective 3	24,700	2,840,858	1,516	4	
		Average Obj	20,053	8,687,213	3,603		
		SD Obj	4,845	5,880,730	2,313		
	Differences	Objective 1	-	2,370,548	-		
		Objective 2	-	42	-		
		Objective 3	160	4,402,442	-		
	% Differences	Objective 1	-	15	-		
		Objective 2	-	0	-		
		Objective 3	1	87	**0**		

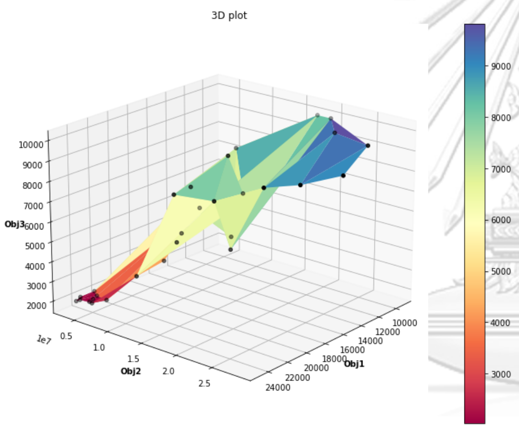
Table 17: Result comparison obtained from CPLEX and proposed heuristic (Cont.)

Instances	Method	Result	Obj1 Value	Obj2 Value	Obj3 Value	Vehicle Used	Runtime
9	CPLEX	Objective 1	**7,070**	19,846,000	6,477	3	0.01 min
		Objective 2	18,575	**3,883,200**	1,286	3	3.03 min
		Objective 3	18,615	5,466,700	**1,286**	3	1.3 min
	Heuristic	Num of solutions	23				
		Objective 1	7,070	19,447,353	6,477	3	2.24 min
		Objective 2	18,575	3,883,202	1,286	3	
		Objective 3	18,575	3,883,202	1,286	3	
		Average Obj	14,441	12,021,939	3,496		
		SD Obj	3,766	7,729,575	1,867		
	Differences	Objective 1	-	398,647	-		
		Objective 2	-	2	-		
		Objective 3	40	1,583,498	-		
	% Differences	Objective 1	-	2	-		
		Objective 2	-	0	-		
		Objective 3	0	34	-		
10	CPLEX	Objective 1	**9,320**	12,574,000	7,509	4	0.03 min
		Objective 2	24,660	2,523,800	1,491	4	11.37 min
		Objective 3	24,880	6,565,400	**1,491**	4	9.07 min
	Heuristic	Num of solutions	35				
		Objective 1	9,320	12,561,253	7,509	4	15.48 min
		Objective 2	24,660	2,523,796	1,491	4	
		Objective 3	24,660	2,523,796	1,491	4	
		Average Obj	18,544	11,288,659	4,216		
		SD Obj	4,878	6,761,159	2,198		
	Differences	Objective 1	-	12,747	-		
		Objective 2	-	4	-		
		Objective 3	220	4,041,604	-		
	% Differences	Objective 1	-	0	-		
		Objective 2	-	0	-		
		Objective 3	1	89	-		
11	CPLEX	Objective 1	**9,320**	14,245,000	8,230	4	0.03 min
		Objective 2	24,660	**2,855,400**	1,634	4	0.26 min
		Objective 3	24,800	5,690,500	**1,634**	**4**	3.14 min
	Heuristic	Num of solutions	32				
		Objective 1	9,320	14,231,488	8,230	4	8.58 min
		Objective 2	24,660	2,855,356	1,634	4	
		Objective 3	24,660	2,855,356	1,634	4	
		Average Obj	18,974	9,323,490	4,193		
		SD Obj	5,045	5,185,726	2,180		
	Differences	Objective 1	-	13,512	-		
		Objective 2	-	44	-		
		Objective 3	140	2,835,144	-		
	% Differences	Objective 1	-	0	-		
		Objective 2	-	0	-		
		Objective 3	1	66	-		

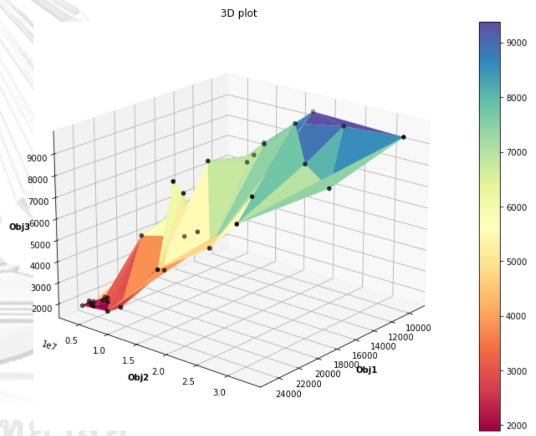
Table 17: Result comparison obtained from CPLEX and proposed heuristic (Cont.)

Instances	Method	Result	Obj1 Value	Obj2 Value	Obj3 Value	Vehicle Used	Runtime
12	CPLEX	Objective 1	**9,360**	17,592,000	8,366	4	0.04 min
		Objective 2	24,700	**3,021,700**	1,661	4	4.42 min
		Objective 3	24,860	7,063,200	**1,661**	**4**	2.26 min
	Heuristic	Num of solution	27				
		Objective 1	9,360	15,069,611	8,366	4	9.44 min
		Objective 2	24,700	3,021,733	1,661	4	
		Objective 3	24,700	3,021,733	1,661	4	
		Average Obj	18,894	10,368,965	4,494		
		SD Obj	4,660	5,851,127	2,345		
	Differences	Objective 1	-	2,522,389	-		
		Objective 2	-	33	-		
		Objective 3	160	4,041,467	-		
	% Differences	Objective 1	-	15	-		
		Objective 2	-	0	-		
		Objective 3	1	80	-		

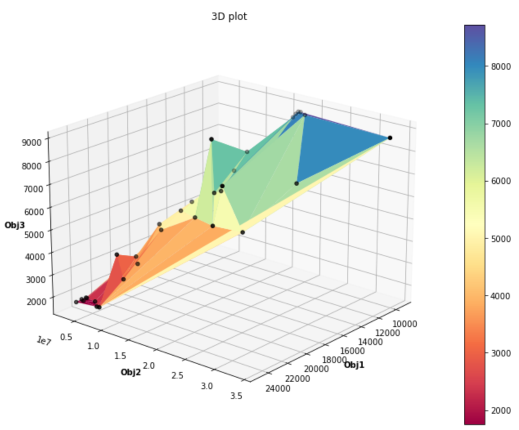
Table 17: Result comparison obtained from CPLEX and proposed heuristic (Cont.)



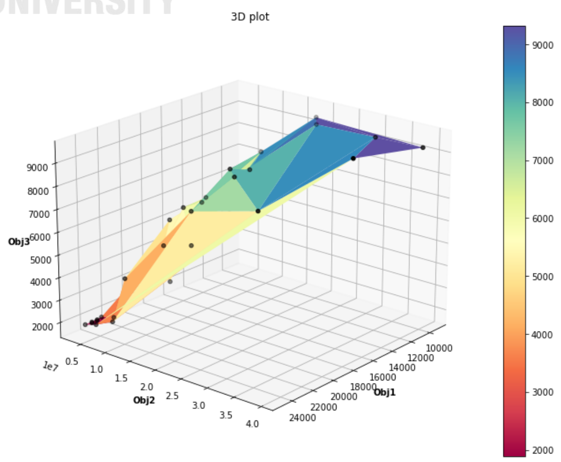
(a) Instance 1 Pareto Front



(b) Instance 2 Pareto Front



(c) Instance 3 Pareto Front



(d) Instance 4 Pareto Front

Figure 28: Pareto Fronts of first 4 instances obtained from the proposed heuristic.

In terms of loading allocation, in Table 18, the different allocation amount pattern results in lessening the carbon emissions from operating the refrigeration unit. If types 2 and 3 are utilized, selecting the type of fresh fruits with similar optimal temperature requirements is recommended as less fuel is consumed to maintain the temperature inside the container which is caused by the heat from the outside through the container wall and heat respired from the fresh fruits inside the container.

As illustrated by the Pareto front in Figure 28, the relationship between the three objectives is that lowering total cost leads to higher total carbon emissions and total weight loss at the destination. The results show that utilizing the proposed heuristic by assigning different routes to different vehicle generates more transportation options for the stakeholders. The choice is therefore dependent on the stakeholder's decision and circumstances. The solutions include various vehicle type combinations, fresh fruit types, amount allocation, and a combination of different transportation routes, as shown in Table 19.

According to Table 20, the results showed that using the proposed mathematical model's piecewise linear approximation caused a slight difference in the optimal value of objective 2, with an average percentage of differences of 0.1354 percent, while there are no percent differences in Objective 1 and Objective 2. In other words, when compared to CPLEX, the proposed heuristic successfully obtain the optimal value for objectives 2 and 3.

In addition, in terms of total runtime and the best value obtained, it is possible to conclude that the proposed heuristic is effective. As in some cases, the proposed heuristic provides the set of solutions in less time, as evidenced by the results of instances 4, 6, and 7.

					Vehicle Loading			Route	
Instances	Method	Min. Objective	Vehicle	Vehicle Type	Compartment 1	Compartment 2	Compartment 3	Total Time (hr)	Total distance (km)
1	Cplex	1	1	1	Durian,30000			806	13856
			2	1	Mangosteen,25000				
			3	1	Durian,15000				
			4	2	Mangosteen,10000	Logan,20000			
		2	1	1	Durian, 30000			160	6776
			2	1	Mangosteeen,30000				
			3	1	Longan,20000				
			4	2	Mangosteeen,5000	Durian, 15000			
		3	1	1	Durian,25600			160	6776
			2	1	Mangosteen,15000				
			3	2	Longan,10000	Durian,19400			
			4	2	Longan,10000	Mangosteen,20000			
	Heuristic	1	1	1	Durian, 30000			806	13856
			2	1	Mangosteen, 30000				
			3	1	Logan, 20000				
			4	2	Mangosteen, 5000	Durian, 15000			
		2	1	1	Durian, 30000			160	6776
			2	1	Mangosteen, 30000				
			3	1	Logan, 20000				
			4	2	Mangosteen, 5000	Durian, 15000			
		3	1	1	Durian, 30000			160	6776
			2	1	Mangosteen, 30000				
			3	1	Logan, 20000				
			4	2	Mangosteen, 5000	Durian, 15000			

Table 18: Vehicle Loading Allocation Result for instance 1 obtained from CPLEX and proposed heuristic

Solution	Obj 1 ($)	Obj2(KgCO2)	Obj3(Kgs)	Vehicle	Vehicle Type	Vehicle Loading Comp 1	Comp 2	Comp 3	Route Time(Hour)	Distance(Km)	Cost ($)
1	13,135	23,281,173	6,512	1	3	Logan, 10000	Durian, 10000	Mangosteen, 10000	44.5	1797	5650
				2	2	Logan, 10000	Mangosteen, 15000		201.5	3464	2205
				3	2	Logan, 10000	Durian, 20000		194	3200	2395
				4	2	Logan, 5000	Durian, 5000		201.5	346	2205
2	17,575	17,493,870	6,723	1	3	Logan, 10000	Durian, 10000	Mangosteen, 10000	44.5	1797	5650
				2	1	Durian, 25000			201.5	3464	2205
				3	2	Logan, 10000	Mangosteen, 15000		205	3400	4560
				4	1	Logan, 15000			208	2400	4560
3	18,275	16,958,017	5,041	1	1	Logan, 30000			44.5	1797	5650
				2	3	Mangosteen, 10000	Logan, 5000	Durian, 10000	227	4000	3820
				3	1	Mangosteen, 15000			201.5	3464	2205
				4	1	Durian, 25000			40	1694	6040
4	23,980	5,483,815	1,805	1	1	Durian, 30000			40	1694	6040
				2	3	Mangosteen, 10000	Durian, 5000	Logan, 10000	40	1694	6040
				3	2	Logan, 10000	Mangosteen, 15000		44.5	1797	5650
				4	1	Logan, 15000			44.5	1797	5650

Table 19: The sample of non-dominated solutions from instance 4 solution

Instances	% differences of the best value		
	Objective 1	Objective 2	Objective 3
1	0	0.0008691	0
2	0	0.0007851	0
3	0	0.0013063	0
4	0	0.0019450	0
5	0	0.0010515	0
6	0	0.0000356	0
7	0	0.0014323	0
8	0	0.0014947	0
9	0	0.0000427	0
10	0	0.0001696	0
11	0	0.0015408	0
12	0	0.0010841	0
Average	0	0.000979726	0

Table 20: Differences between the result from CPLEX and the proposed heuristic in each Best Objective Value

4.4 Multi-Objective Optimization Computational Experiment

In this section, only the proposed heuristic with suggested population size and stop criterion for multi-objective optimization problems is used and evaluated. In multi-objective problems, multiple sample runs are required to obtain the most effective solutions for stakeholders. In this computational experiment, 12 data sets of small, medium, and large instances with 10 replications each are tested over 1000 generations. Because the proposed heuristic generates random solutions, the best solution set is chosen from 10 replications based on the maximum number of non-dominated solutions. And the value of inverted generational distance (IGD) is then calculated to access their quality. As mentioned in the previous section, the weight requirement of each fresh fruit and vegetable type is randomly selected from a range of 10,000 to 50,000 kgs for all instances. The results of small, medium, and large instances are discussed as follows:

4.4.1 Small Instance Problem

The population size of 240 with 150 non-improvement generations as a stop criterion has been applied to small instance cases. Table 21 displays the results in terms of minimum value, maximum value, number of non-dominated solutions, and IGD value.

Instances	Value	Obj1 (USD)	Obj2 (KgCO2)	Obj3 (Kgs)	No. solutions	Runtime (mins)	IGD
1	Max	30,885	27,442,754	9,899	36	49.37	0.536
	Min	9,360	3,380,856	1,965			
2	Max	30,885	31,736,366	9,609	40	49.36	0.607
	Min	9,360	3,380,850	1,881			
3	Max	31,125	42,976,060	9,938	46	53.42	0.487
	Min	9,360	30,794,77	1,865			
4	Max	30,825	29,467,730	8,812	37	58.54	0.534
	Min	9,360	29,287,68	1,741			
5	Max	24,660	31,955,375	10,012	35	38.53	0.444
	Min	9,320	10,361,809	1,987			
6	Max	30,825	34,127,452	9,041	39	27.24	0.448
	Min	9,360	9,351,709	1,795			
7	Max	24,660	28,068,599	7,358	36	75	0.488
	Min	9,320	26,746,38	1,460			
8	Max	30,825	20,843,491	7,636	36	72	0.457
	Min	9,360	2,717,792	1,515			
9	Max	24,700	20,286,808	6,477	26	18.43	0.421
	Min	7,070	3,293,140	1,285			
10	Max	24,700	35,258,069	7,875	35	92	0.49
	Min	9,320	2,523,795	1490			
11	Max	24,700	38,616,269	8,620	37	69	0.561
	Min	9,320	2,855,356	1,633			
12	Max	30,825	35,052,104	8,839	38	66	0.431
	Min	9,360	2,868,501	1661			

Table 21: Multi-objective problem results for small instances

According to Table 12, the average CPU runtime is 55.74 minutes, with 37 solutions generated on average. The algorithm appears to generate a longer run time than the results obtained from using a 240 population size with the stopping criterion of 150 non-improvement generations in section 4.2. The computational time increases from 33 minutes to 55.7 minutes on average. Regarding the randomness of the input, the results indicate that variations in the required weight of fresh fruits necessitate a lengthy search for loading combination solutions with a CPU runtime standard deviation of 20.94 minutes. Even though the computational time increased by 68.9115 percent, approximately, the number of non-dominated solutions also improved from 20 to about 37, representing an 85 percent increase. As a result, it is possible to conclude that the proposed heuristic with the stopping criterion performs effectively on small instance problems.

Furthermore, to access the quality of the non-dominated solution set, the solutions are first normalized. The minimum Euclidean distance between the solution point and the reference point is then calculated and selected to obtain the IGD value. Hence, the average IGD value is reported to be 0.492 approximately.

4.4.2 Medium Instance Problem

The results of applying the proposed heuristic to medium instances are discussed in this section. For medium instance cases, a population size of 240 with 150 non-improvement generations as a stop criterion was used. The results are shown in Table 22 in terms of minimum value, maximum value, number of non-dominated solutions, and IGD value.

Table 22 shows that the average runtime is about 134.33 minutes. By including two more types of fresh fruits, the average runtime increased by about 155 percent. Again, the variation in fresh fruit requirement weight affects runtime variation, with a standard variation of 120.668. Furthermore, the number of solutions increased from 37 to 41 solutions which increased by 10.8%. Meanwhile, an increase in the variety of fresh fruits results in an increase in all objective values, which increases in IGD value because a higher value results in a greater distance from the reference point in the search space. As a result, the IGD value is 0.6072 which increased by approximately 23.374 percent, with a standard deviation of 0.092.

Instances	Value	Obj1 (USD)	Obj2 (KgCO2)	Obj3 (Kgs)	No. solutions	Runtime (mins)	IGD
1	Max	43,155	31,701,415	9,020	43	20	0.533
	Min	14,020	4,733,277	1,790			
2	Max	43,195	45,835,881	8,577	49	347	0.689
	Min	14,060	4,854,042	1,702			
3	Max	43,215	47,858,031	10,750	35	17	0.477
	Min	14,020	5,215,618	2,134			
4	Max	43,235	45,235,147	11,405	41	100	0.669
	Min	16,390	60,00,004	2,264			
5	Max	30,825	34,909,186	6.260	40	15	0.579
	Min	9,400	2,988,985	1,242			
6	Max	30,905	39,260,584	5,794	41	305	0.789
	Min	11,690	3,848,466	1,150			
7	Max	49,400	71,072,290	12,831	43	203	0.69
	Min	18,700	7,098,192	2,547			
8	Max	37,070	38,957,157	8,691	34	99	0.622
	Min	14,060	5,595,069	1,646			
9	Max	43,155	65,397,630	9,616	39	223	0.577
	Min	16,310	5,215,497	1,909			
10	Max	49,380	64,681,012	12,157	44	223	0.621
	Min	16,350	5,620,474	2,413			
11	Max	11,730	43,782,727	6,646	43	23	0.481
	Min	37,090	4,590,899	1,319			
12	Max	37,050	34,532,365	6,534	37	37	0.56
	Min	9440	3,424,076	1,297			

Table 22: Multi-objective problem results for medium instances

4.4.3 Large Instance Problem

A population size of 480 with 150 non-improvement generations as a stop criterion was used for large example cases. Table 23 displays the results in terms of minimum value, maximum value, number of non-dominated solutions, and IGD value. The average runtime increased from 134 minutes to 267 minutes by increasing from 5 to 8 types. The runtime variation is also increased by 94% compared to medium instances. Meanwhile, the number of discovered solutions has increased by 39 percent, from 41 to 57, representing a 28.2 percent increase when comparing the changes from small to medium instances. The average IGD value as well increased from 0.6 to 0.868 with a standard deviation of 0.11.

Instances	Value	Obj1 (USD)	Obj2 (KgCO2)	Obj3 (Kgs)	No. solutions	Runtime (mins)	IGD
1	Max	61,690	45,166,534	10,627	54	120	1.019
	Min	21,050	7,500,619	2,109			
2	Max	61,890	77,595,719	11,902	63	377	0.96
	Min	23,340	7,591,111	2,362			
3	Max	49,520	51,094,520	12,417	50	777	0.878
	Min	18,720	6,796,837	2,464			
4	Max	80,265	107,713,739	14,999	57	111	0.774
	Min	25,790	10,358,617	2,977			
5	Max	61,750	57,853,301	13,591	64	215	0.982
	Min	21,030	7,651,533	2,697			
6	Max	55,565	75,072,252	12,704	52	76	0.747
	Min	21,050	8,904,887	2,522			
7	Max	61,710	70,693,705	13,330	55	256	0.741
	Min	21,050	8,058,622	2,646			
8	Max	61,690	84,301,740	12,663	60	140	0.668
	Min	21,010	7,455,081	2,507			
9	Max	61,770	83,166,586	14,280	63	206	0.928
	Min	23,400	9,475,988	2,834			
10	Max	67,955	56,486,481	15,307	57	134	1.027
	Min	21,130	9,564,006	3,038			

Table 23: Multi-objective problem results for large instances

Instances	Value	Obj1 (USD)	Obj2 (KgCO2)	Obj3 (Kgs)	No. solutions	Runtime (mins)	IGD
11	Max	61,730	79,064,940	11846.221	48	692	0.818
	Min	21,130	8,616,279	2351.607			
12	Max	74,040	101,284,719	17188.744	66	105	0.876
	Min	23,420	8,868,073	3362.229			

Table 23: Multi-objective problem results for large instances (Cont.)

When comparing the objective value results from small, large, and medium (Figures 29, 30, and 31), the total cost values in Objective 1 consistently increase in both the average minimum and maximum value as the types of fresh fruits are increased. Similarly, when more types of fresh fruits were added, the average maximum and minimum total carbon emissions (Objective 2) increased instantly. In contrast, increasing the number of fresh fruit types has the least impact on the value of objective 3. The average of the maximum value in Objective 3 began to rise at 8 types of fresh fruits, while the minimum value increased slightly. The weight loss rate of the added fresh fruits is suspected to be the factor influencing the changes.

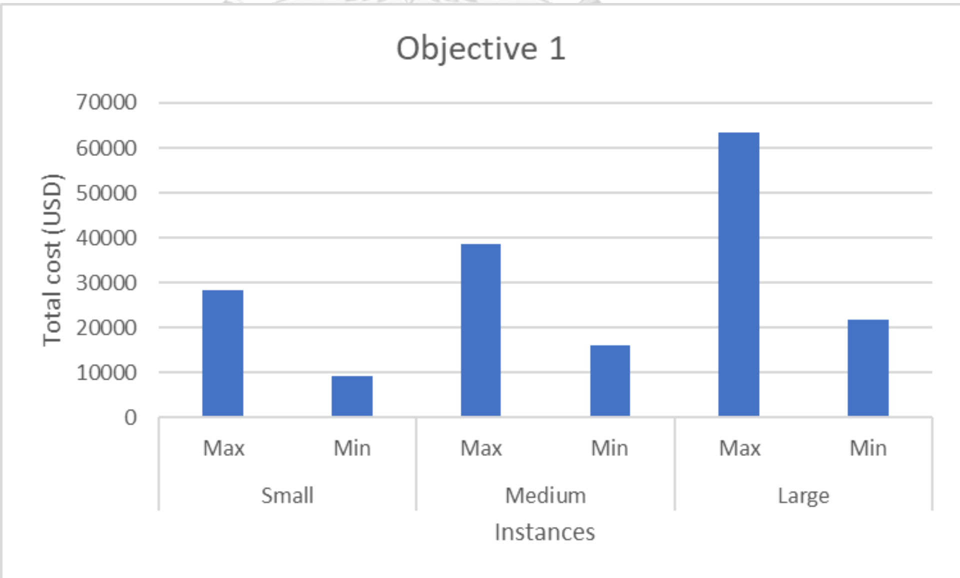


Figure 29: Objective 1 comparison of small, medium, and large comparison

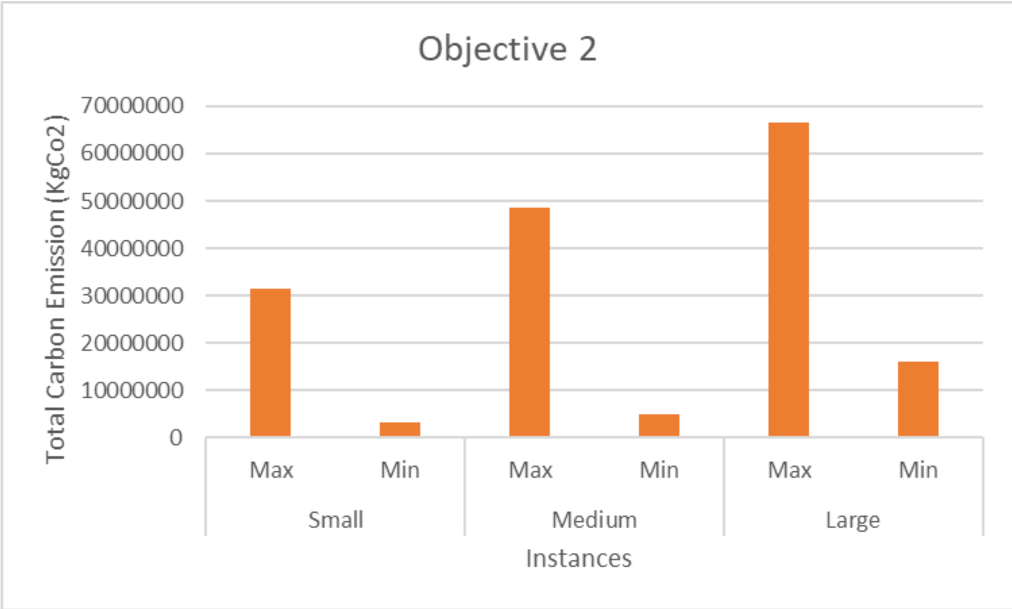


Figure 30: Objective 2 comparison of small, medium, and large comparison

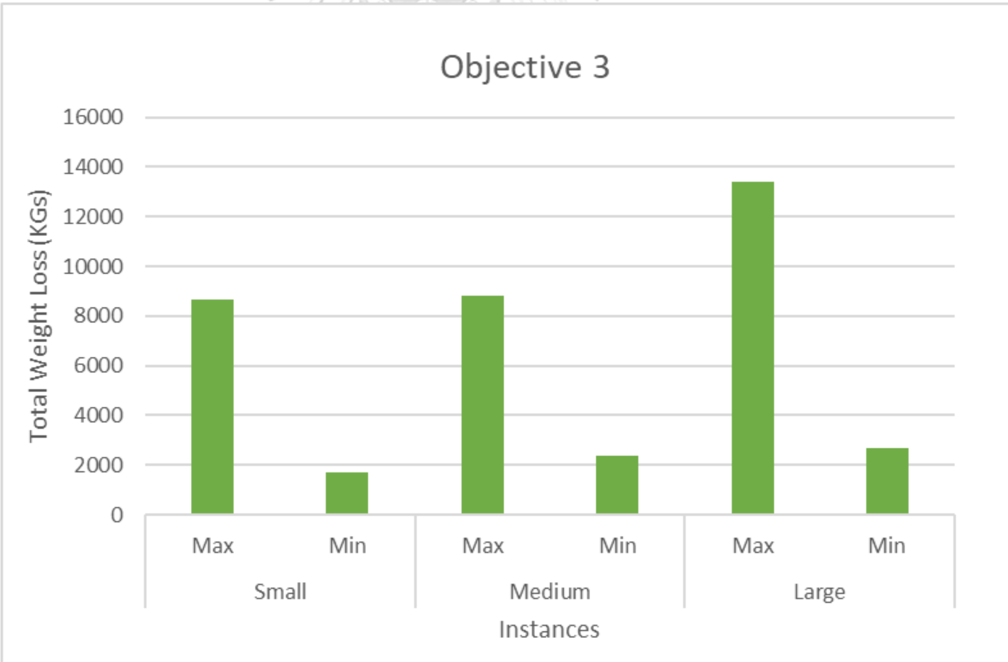


Figure 31: Objective 3 comparison of small, medium, and large comparison

To summarize, there are computational time trade-offs when increasing the type of fresh fruits as well as the IGD values. When the number of fresh fruit types is increased from 3 to 5 and 5 to 8, both computational time and IGD increase dramatically (Figures 32 and 33). The variation in the required weight of fresh fruits, on the other hand, caused a large deviation in computational time while having a minor impact on the IGD value. All objectives' values are more likely to have a direct impact on changes in IGD value. When all values are normalized, especially Objective 2, it tends to result in a greater distance from the set of selected reference points. Furthermore, as more vehicles are used, an increase in the number of fresh fruit types is more likely to have an impact on total distance and total carbon emissions. While the weight loss rate of the assigned fresh fruits is likely to influence Objective 3.

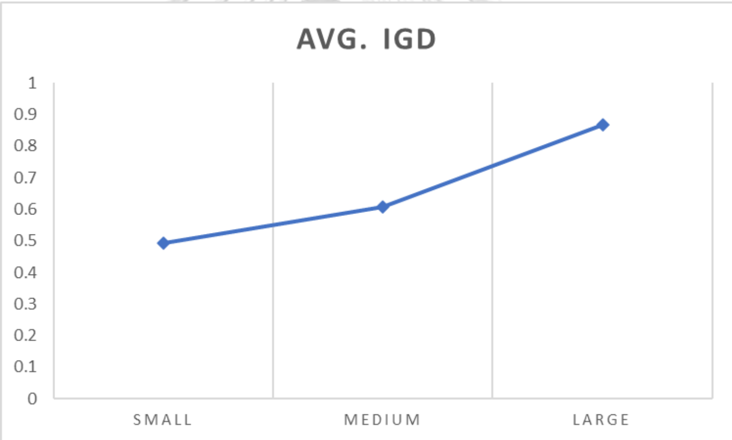


Figure 32: Average IGD value for small, medium, and large instances

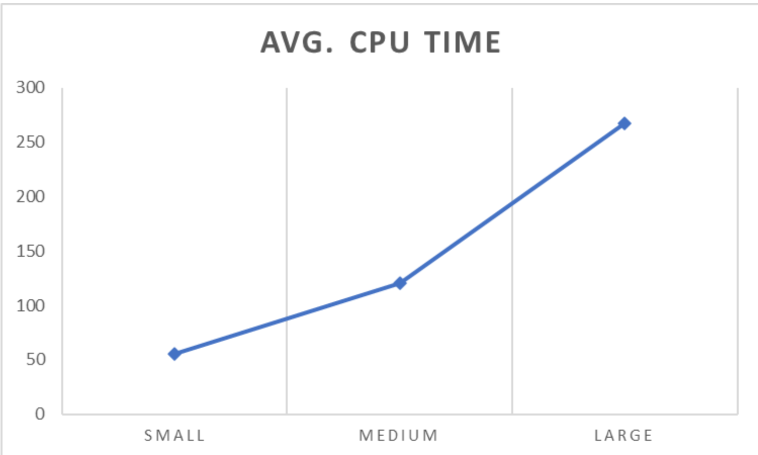


Figure 33: Average CPU time of small, medium, and large instances

4.5 Pareto Solutions Selection for Decision-makers

According to the previous section's findings, the number of non-dominated solutions increased as the number of fresh fruit types increased. Furthermore, the results demonstrated the relationship between the three objectives. Objective 1 (or total transportation cost) has a conflicting relationship with objectives 2 and 3 (or total carbon emissions and total weight loss). In other words, lowering objective 1 leads to higher values for objectives 2 and 3. Minimizing objectives 2 and 3, on the other hand, results in higher transportation costs. Therefore, prioritizing the objectives is recommended to select the solutions in the Pareto fronts in this study.

Though minimizing total transportation is the most important factor, freshness becomes another influential concern in the transportation of fresh fruits and vegetables because it affects both customer satisfaction and the market price. Therefore, prioritizing cost and freshness at the same level is suggested in this study to balance the quality standard and transportation cost, with carbon emissions coming last. As this prioritizing level is set at the same level for Objective 1 and Objective 3, and the two objectives have a conflicting relationship, the solutions between the 25th and 75th percentiles are considered in this section assessment.

For this section evaluation, instance 1 from the medium instance is selected for analysis. Figure 34 depicts the 3-D Pareto front, whereas Table 24 contains 43 solutions and their percentile for example 1 of a medium case problem. The solutions between the 25th and 75th percentiles are sorted separately based on the fitness values (or objective values), and the details information, such as the layout of each vehicle, the loading assignment of fresh fruits, and the route selection of each vehicle, are investigated further.

According to Table 24, the solutions for objectives 1 and 3 are chosen based on the selected percentile range (highlighted in Table 24). As shown in Table 25, the mutual solutions are selected from the non-dominated solutions set by sorting the number of solution points and selecting the mutual point in objectives 1 and 3. As a result, the number of selected solutions decreased from 43 to 20 solutions (Figure 35).

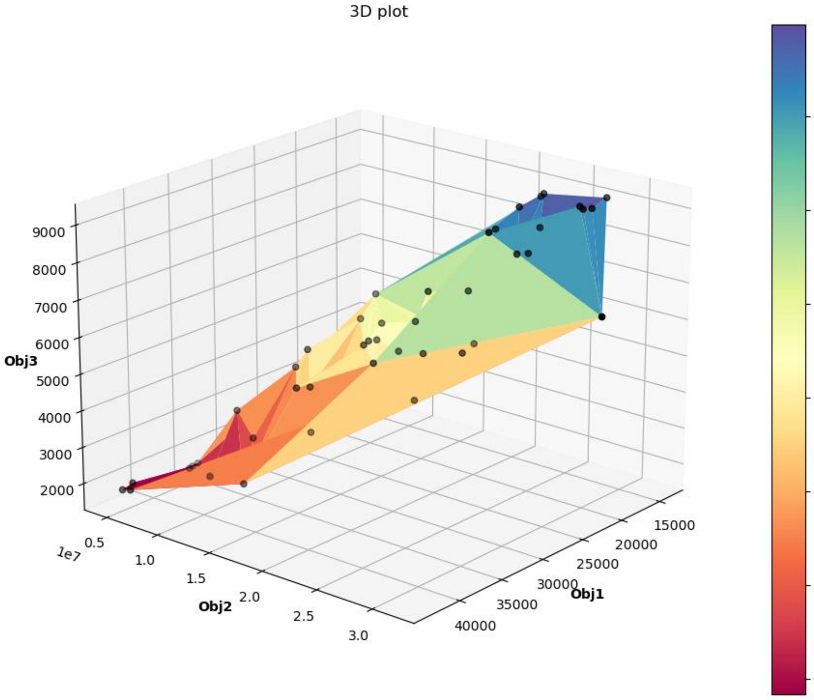


Figure 34: 3-D Pareto front for instance 1 of medium case problem

Point	Obj1	Rank	Percent	Point	Obj2	Rank	Percent	Point	Obj3	Rank	Percent
2	43155	1	100.00%	26	31701415	1	100.00%	1	9020.096	1	97.60%
38	43125	2	97.60%	23	27714009	2	97.60%	4	9020.096	1	97.60%
9	42285	3	95.20%	1	27651826	3	95.20%	15	8995.958	3	95.20%
25	36740	4	92.80%	21	27430639	4	92.80%	31	8951.346	4	92.80%
6	36640	5	90.40%	15	27277513	5	90.40%	21	8917.105	5	90.40%
33	36470	6	88.00%	29	24343816	6	88.00%	23	8897.251	6	88.00%
34	36250	7	85.70%	13	24088790	7	85.70%	19	8860.95	7	85.70%
24	35860	8	83.30%	22	23871976	8	83.30%	3	8423.773	8	83.30%
5	34720	9	80.90%	4	23577243	9	80.90%	13	8233.526	9	80.90%
17	34330	10	78.50%	31	23450871	10	78.50%	36	8211.885	10	78.50%
41	32365	11	76.10%	19	23322181	11	76.10%	22	7714.062	11	76.10%
43	31165	12	73.80%	3	22986949	12	73.80%	29	7705.148	12	73.80%
30	30985	13	71.40%	36	21748506	13	71.40%	14	6687.384	13	71.40%
11	30940	14	69.00%	28	21485489	14	69.00%	26	6664.907	14	69.00%
20	30585	15	66.60%	8	20937072	15	66.60%	37	6631.51	15	66.60%
35	28510	16	64.20%	37	20791669	16	64.20%	16	6343.36	16	64.20%
10	27930	17	61.90%	27	19150821	17	61.90%	40	5827.077	17	61.90%
7	27240	18	59.50%	14	18926714	18	59.50%	12	5706.435	18	59.50%
42	26360	19	57.10%	18	18720282	19	57.10%	39	5659.171	19	57.10%
12	26205	20	54.70%	40	18038785	20	54.70%	7	5316.038	20	54.70%
32	25905	21	52.30%	32	17353196	21	52.30%	42	5274.94	21	52.30%
18	25735	22	50.00%	35	17008186	22	50.00%	10	5273.536	22	50.00%

Table 24: The non-dominated solutions from instance 1 of the medium case problem

Point	Obj1	Rank	Percent	Point	Obj2	Rank	Percent	Point	Obj3	Rank	Percent
39	25625	23	47.60%	42	15652465	23	47.60%	8	5141.75	23	47.60%
27	25235	24	45.20%	10	15634846	24	45.20%	28	5133.087	24	45.20%
16	25125	25	42.80%	7	15557834	25	42.80%	20	5123.881	25	42.80%
40	24735	26	40.40%	39	15543179	26	40.40%	27	5083.728	26	40.40%
14	24345	27	38.00%	16	14595523	27	38.00%	32	5060.502	27	38.00%
28	23495	28	35.70%	12	13982779	28	35.70%	35	4944.565	28	35.70%
3	22360	29	33.30%	30	12983056	29	33.30%	43	4639.144	29	33.30%
37	21870	30	30.90%	11	12895711	30	30.90%	41	4261.48	30	30.90%
8	21290	31	28.50%	41	12743252	31	28.50%	11	4164.897	31	28.50%
26	20260	32	26.10%	20	12398016	32	26.10%	18	3803.534	32	26.10%
22	20040	33	23.80%	43	11707702	33	23.80%	5	3531.631	33	23.80%
36	19820	34	21.40%	33	10928665	34	21.40%	30	2919.145	34	21.40%
29	19270	35	19.00%	17	10114610	35	19.00%	17	2834.208	35	19.00%
19	19045	36	16.60%	5	8894695	36	16.60%	9	1918.736	36	16.60%
13	17465	37	14.20%	25	7836889	37	14.20%	24	1900.111	37	14.20%
15	16945	38	11.90%	24	5886341	38	11.90%	38	1852.778	38	11.90%
21	16755	39	9.50%	6	5708385	39	9.50%	33	1845.152	39	9.50%
31	16500	40	7.10%	34	5695553	40	7.10%	34	1836.528	40	7.10%
4	16310	41	4.70%	38	5493176	41	4.70%	6	1815.548	41	4.70%
23	16015	42	2.30%	9	4987547	42	2.30%	25	1795.278	42	2.30%
1	14020	43	0.00%	2	4733278	43	0.00%	2	1790.59	43	0.00%

Table 24: The non-dominated solutions from instance 1 of the medium case problem (Cont.)

Point	Obj1	Rank	Percent	Point	Obj2	Rank	Percent	Point	Obj3	Rank	Percent
7	27240	18	59.50%	7	15557834	25	42.80%	7	5316.038	20	54.70%
8	21290	31	28.50%	8	20937072	15	66.60%	8	5141.75	23	47.60%
10	27930	17	61.90%	10	15634846	24	45.20%	10	5273.536	22	50.00%
11	30940	14	69.00%	11	12895711	30	30.90%	11	4164.897	31	28.50%
12	26205	20	54.70%	12	13982779	28	35.70%	12	5706.435	18	59.50%
14	24345	27	38.00%	14	18926714	18	59.50%	14	6687.384	13	71.40%
16	25125	25	42.80%	16	14595523	27	38.00%	16	6343.36	16	64.20%
18	25735	22	50.00%	18	18720282	19	57.10%	18	3803.534	32	26.10%
20	30585	15	66.60%	20	12398016	32	26.10%	20	5123.881	25	42.80%
26	20260	32	26.10%	27	19150821	17	61.90%	26	6664.907	14	69.00%
27	25235	24	45.20%	28	21485489	14	69.00%	27	5083.728	26	40.40%
28	23495	28	35.70%	30	12983056	29	33.30%	28	5133.087	24	45.20%
30	30985	13	71.40%	32	17353196	21	52.30%	32	5060.502	27	38.00%
32	25905	21	52.30%	35	17008186	22	50.00%	35	4944.565	28	35.70%
35	28510	16	64.20%	36	21748506	13	71.40%	37	6631.51	15	66.60%
37	21870	30	30.90%	37	20791669	16	64.20%	39	5659.171	19	57.10%
39	25625	23	47.60%	39	15543179	26	40.40%	40	5827.077	17	61.90%
40	24735	26	40.40%	40	18038785	20	54.70%	41	4261.48	30	30.90%
42	26360	19	57.10%	41	12743252	31	28.50%	42	5274.94	21	52.30%
43	31165	12	73.80%	42	15652465	23	47.60%	43	4639.144	29	33.30%

Table 25: Selected solutions between 25th and 75th percentile of instance 1 from medium case problem

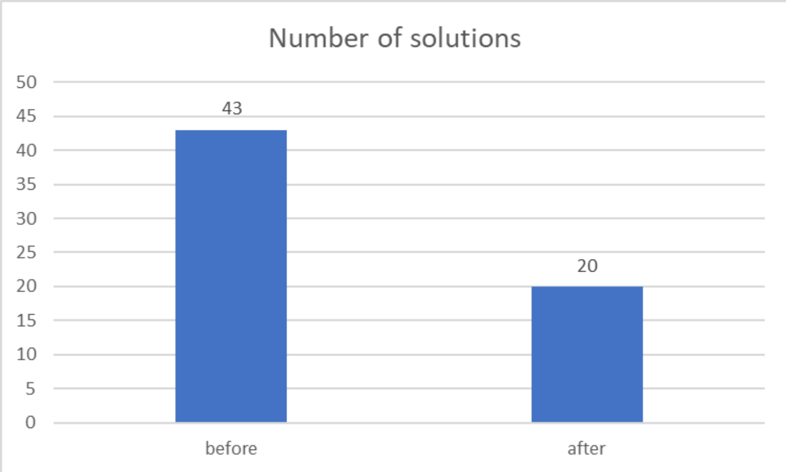


Figure 35: Reduction in the number of solutions

Furthermore, as shown in Figure 36, the gap between the minimum and maximum value of all objectives is reduced because the minimum value of the selected solution set is increased while the maximum value is decreased. The difference between the minimum and maximum values of objectives 1, 2, and 3 is reduced by 62.57 percent, 65.33 percent, and 60 percent, respectively. As a result, according to the recommended solutions, the total transportation cost falls between \$20,260 and \$31,165, while the total weight loss falls between 3,803 Kgs and 6,687 Kgs. This results in total carbon emissions ranging from 12,398,016 KgCO2 to 21,748,505 KgCO2.

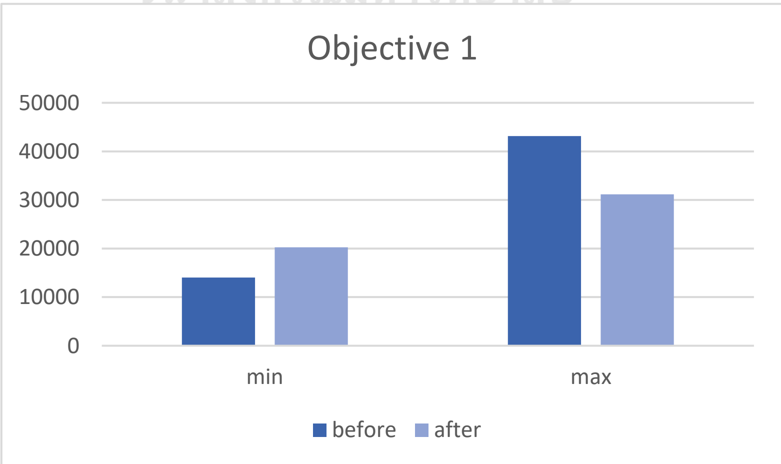
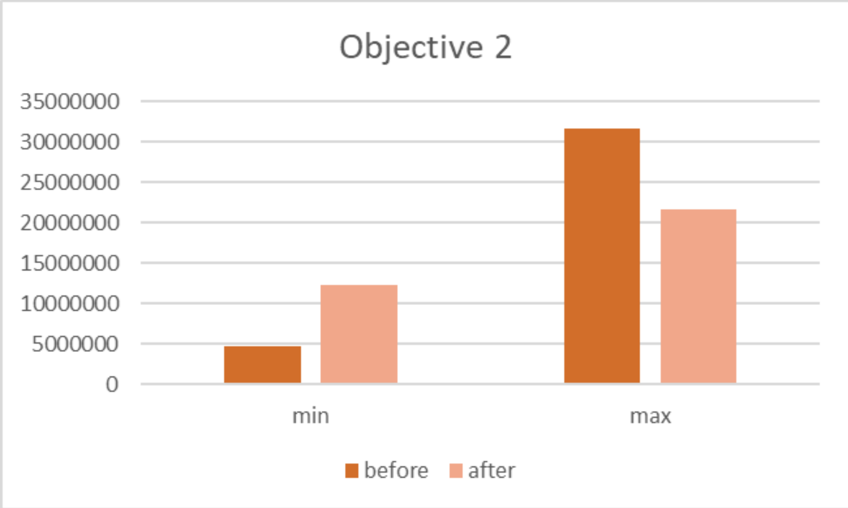


Figure 36: The changes in minimum and maximum values of all objectives



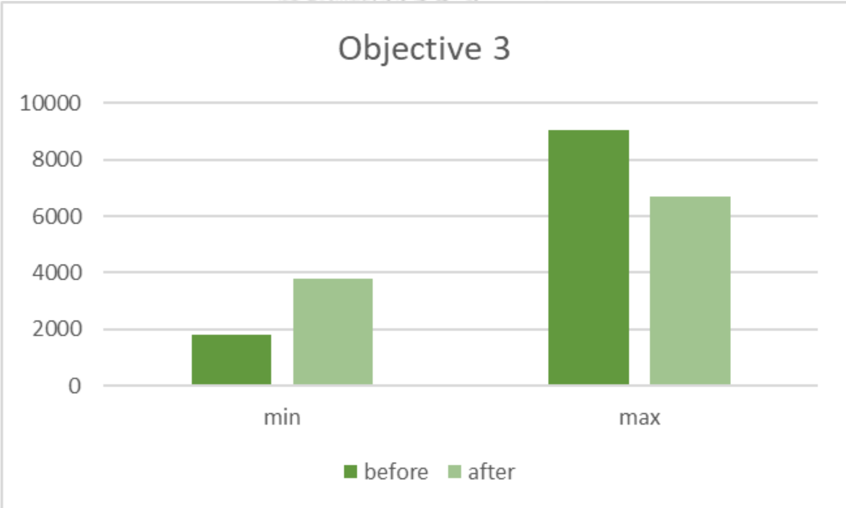


Figure 36: The changes in minimum and maximum values of all objectives (Cont.)

For an in-depth analysis, three pairs of solutions are selected as samples and compared. The first pair denoted the same vehicle layout and loading pattern but a different route selection, whereas the second pair represented a different vehicle layout, loading pattern, and route selection. And the last pair represented the number of vehicles used. In Table 26, the first pair shows that a small difference in total weight loss can result in a large difference in total carbon emissions with higher costs. The varied route selection has a great impact on total carbon emissions when it comes to the same vehicle selection and loading pattern.

For the second pair, a difference in vehicle layout selections and loading with different route selections for each vehicle indicates that the loadings assignment influences a higher carbon emission because the total heat load is made up of thermal energy sources inside the compartment, such as transmission load based on compartment body size and respiration heat from fresh fruits. Furthermore, the weight of fresh fruits influences total heat through respiratory heat rate, which requires more fuel to maintain the temperature low. Route selection, on the other hand, continues to be a major factor in determining whether total transportation costs are high or low, whereas vehicle layout has a smaller impact on adjustment costs.

Finally, the results of the last sample pair show that using fewer vehicles does not result in lower objective values. As shown in Table 26, the solution with 7 vehicles appears to produce less total carbon emissions and total weight loss at a slightly higher total cost and route selection decision. Again, vehicle loadings and route selection appear to have a significant impact on these differences.

In conclusion, fresh fruit loadings and route selection have a significant impact on all objective values and the variation of the solution set. Another finding is that having fewer vehicles does not always result in better solutions. More vehicles appear to result in lower total carbon emissions and weight loss at a slightly higher total transportation cost in this case. However, this study only suggests alternative solutions that prioritize cost and freshness at the same level. Solutions between the 25th and 75th percentiles are considered in decision-making evaluation. As a result, an appropriate solution is ultimately determined by stakeholder preferences and circumstances.

Solution	Vehicle	Comp1	Comp2	Comp3	Route	Time	Distance	Cost	Obj 1	Obj 2	Obj 3
1	1	'Durian', 23000			1	44.5	1797	5650	27240	15557834	5316.038
	1	'Mangoes', 30000			2	44	1959	6340			
	1	Logan', 30000			6	227	4000	3820			
	1	'Bananas', 25000			5	194	3200	2395			
	1	'Mangosteen', 22000			7	201.5	3464	2205			
	2	'Logan', 10000	'Mangoes', 14000		3	40	1694	6040			
2	1	'Durian', 23000			3	40	1694	6040	21290	20937072	5141.75
	1	'Mangoes', 30000			1	44.5	1797	5650			
	1	Logan', 30000			3	40	1694	6040			
	1	'Bananas', 25000			2	44	1959	6340			
	1	'Mangosteen', 22000			1	44.5	1797	5650			
	2	'Logan', 10000	'Mangoes', 14000		1	44.5	1797	5650			
3	1	'Durian', 23000			1	44.5	1797	5650	25735	18720282	3803.534
	1	'Mangoes', 30000			7	201.5	3464	2205			
	1	Logan', 30000			2	44	1959	6340			
	1	'Bananas', 25000			7	201.5	3464	2205			
	1	'Mangosteen', 22000			2	44	1959	6340			
	2	'Logan', 10000	'Mangoes', 14000		7	201.5	3464	2205			

Table 26 : Three pairs of sample solutions from the selected solutions from Instance 1 of the medium case problem

Solution	Vehicle	Comp1	Comp2	Comp3	Route	Time	Distance	Cost	Obj 1	Obj 2	Obj 3
4	1	'Bananas', 25000			4	208	3400	4560	20260	31701415	6664.907
	2	'Mangoes', 10000	'Mangosteen', 20000		5	194	3200	2395			
	3	'Mangosteen', 2000	'Mangoes', 10000	'Logan', 10000	7	201.5	3464	2205			
	1	'Durian', 23000			1	44.5	1797	5650			
	1	'Logan', 30000			7	201.5	3464	2205			
	1	'Mangoes', 24000			5	194	3200	2395			
5	1	'Durian', 23000			6	227	4000	3820	24735	18038785	5827.077
	1	'Mangoes', 30000			6	227	4000	3820			
	1	Logan', 30000			5	194	3200	2395			
	1	'Bananas', 25000			3	40	1694	6040			
	1	'Mangosteen', 22000			3	40	1694	604			
	2	'Logan', 10000	'Mangoes', 14000		5	194	3200	2395			
6	1	'Durian', 23000			7	201.5	3464	2205	26360	15652465	5274.94
	1	'Mangoes', 30000			7	201.5	3464	2205			
	1	'Logan', 30000			1	44.5	1797	5650			
	1	'Bananas', 25000			7	201.5	3464	2205			
	1	'Mangosteen', 22000			2	44	1959	6340			
	1	'Mangoes', 14000			1	44.5	1797	5650			
	3	'Logan', 10000			1	44.5	1797	5650			

Table 26 : Three pairs of sample solutions from the selected solutions from Instance 1 of the medium case problem (Cont.)

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

This chapter summarizes the study's objective and problem, as well as the methodology used to solve single-objective and multi-objective optimization problems and the performance of the proposed algorithm. Subsequently, in this section, the recommendation for this study is discussed.

5.1 Conclusions

In recent years, the common challenge of exporting fresh fruits and vegetables is balancing the quality standard and transportation costs as they typically have a short life cycle and deteriorate instantly. The refrigerated vehicles are used to prolong the shelf life of fresh fruits and vegetables by maintaining the temperature inside the chamber at an optimal level. However, the refrigeration unit's fuel consumption appears to be a cause of high carbon emissions. A multi-compartment technology was recently introduced into the market as an alternative solution for business owners, allowing them to transport multiple products with different temperature requirements in a single shipment. Therefore, this study has taken mentioned matter into account and represents the problem as a multi-compartment vehicle loading, and transportation routing problem (MCVRLP). The proposed mathematical model and heuristic are an integration of the shortest path and bin packing problem which aim to minimize the total transportation cost, total carbon emissions, and the total weight loss at the destination.

Because the multi-compartment vehicle loading and transportation routing problem (MCVRLP) is considered an NP-hard problem, the hybrid NSGA III algorithm is proposed in this study to solve the problem. The algorithm performance evaluation for the heuristic computational experiment is divided into three parts. The sample data range from 10,000 to 50,000 kilograms at random. Regarding the long runtime, stop criteria are introduced, and non-improvement generation in the number of solutions is used as a stop criterion. The impact of population size and generation number variation is assessed to select appropriate stop criteria (small, medium, and large). The population size is set to 60, 120, 240, and 480, with evolution iterations of 100, 200, 500, and 1000. In terms of the number of solutions found and computational time, the results of running all combinations on small case problems are chosen as the benchmark. Thus, using stop criteria performs better in terms of runtime while generating the same number of solutions for a small instance. Regarding the outcome of a small instance, the appropriate stop criteria are also chosen for both medium and large instances. As a result, in this study, a population size of 240 with a stop criteria of 150 non-improvement generations is used in the small and medium cases, while a population size of 480 with a stop criteria of 300 non-improvement generations is used in the large case.

Next, the optimal solution obtained from single objective optimization by solving the proposed nonlinear programming (NLP) model presented in Chapter 3 is utilized as a benchmark for validating the proposed algorithm's performance. The twelve data sets of three types of fresh fruits were chosen at random from a range of 10,000 to 50,000 kgs and tested on a single run, with the total shipment in each case not exceeding 100,000 kilograms. By comparing the results of CPLEX and the proposed heuristic, it is possible to conclude that the algorithm can generate a near-optimal solution set. For each objective, there is no difference between the optimal CPLEX solution and the minimum value from the proposed heuristic. Furthermore, the heuristic results show a relationship between the three objectives, implying that lowering total cost leads to higher total carbon emissions and total weight loss at the destination. The results also show that by assigning different routes to different vehicles, the proposed heuristic generates more transportation options for the stakeholders. As a result, the choice is determined by the stakeholder's decision and circumstances.

To further investigate, small, medium, and large case computational experiments for multi-objective optimization are performed to see the effects of changes in the number of fresh fruit types. As a result, when more types of fresh fruit are added, the values for objectives 1 and 2 increase, while the value for Objective 3 remains unchanged until the number of fresh fruit types increases to eight. Furthermore, increasing the number of fresh fruit types from 3 to 5 and 5 to 8 increases both computational time and IGD significantly. However, the goal of this study is to balance the total transportation cost and the quality of fresh fruits at the destination. It is recommended to prioritize cost and freshness equally and to select solutions between the 25th and 75th percentiles. The results also revealed that varied route selection has a significant impact on total carbon emissions when the same vehicle selection and loading pattern are used. The majority of carbon dioxide emissions are caused by operating refrigeration units and are greatly influenced by the weight of fresh fruits through their respiratory heat rate; thus, loadings assignment influences a higher carbon emission. Another discovery is that using fewer vehicles does not always result in better solutions. More vehicles appear to reduce total carbon emissions and weight loss at the expense of a slightly higher total cost and route selection decision.

To summarize, the proposed heuristic can provide stakeholders with more transportation options. The variation in generated solutions is due to different vehicle type combinations, amount allocation, and a combination of different transportation routes. However, the decision and circumstances of the stakeholder determine the option.

5.2 Recommendations

In this section, consideration of practical and strategic that arise from the research findings are discussed and introduced as the managerial implication. In other words, it's the application of insights gained from research to decision-making and problem-solving in real-world business settings. This section highlights the key takeaways from the research and their potential impact on managerial decision-making, offering suggestions for action that managers can take to improve business performance.

To apply the proposed heuristic to a situation where the number of vegetables and fresh fruit types is more than eight, the new stop criteria or the non-improvement generation is needed to be defined before solving the problem using the proposed heuristic. This has to be noted that the proposed heuristic is only applicable for the vegetable and fresh fruits that are sold by weight where the size of the packaging is not taken into account.

Moving to the insights gained from this research, as this research rate the importance of total cost and the freshness quality at the same level, the solution obtained was suggested to sort the obtained solutions and selected the solutions between percentile 25^{th} and 75^{th}. However, the solutions sorting still results in several alternative solutions. The versatility of the solution is influenced by three main factors which are the loading pattern, number of vehicles used, and route selection. The are two trade-offs that must be considered during the decision-making process to select one solution to solve a real-life problem. Most solutions generate the trade-off among the cost, total carbon emissions, and fresh fruit quality. With lessen total cost tends to generate high carbon emissions and huge total weight loss. On the other hand, lowering carbon emissions and reducing total weight loss results in higher transportation costs. If the company's goal is focusing on green policy, for example, if the company needs to select the solution between the two solutions with different numbers of vehicles used and there are trade-offs between the total cost and the total carbon emissions. The solution with more vehicle usage generates lower total carbon emissions but higher costs due to the number of vehicles used. A solution with a slight increase in the total cost is suggested as the company goal is to focus on the environmental policy and vice versa the solution with fewer vehicles used may be selected if the company goal is to lessen the total cost while maintaining the total carbon emissions and the total weight at the acceptable level.

Thus, the managerial implications section is critical because it translates academic findings into practical recommendations for managers to make informed decisions that drive business growth and success. By understanding the implications of research, managers can use evidence-based insights to develop effective strategies, implement changes, and ultimately achieve their organizational goals.

5.3 Future Works

- This study only considers the weight assigned to each compartment; however, to perform practically, a further study is suggested by considering adapting the container box allocation problem to multi-compartment vehicle loading. This can be either in 2-D or 3-D box allocation problems.
- As only the exporting case is considered in this study, the problem can be simplified to the VRP or CVRP problem by using a multi-compartment vehicle to transport the fresh produce.
- In the real world, the temperature outside and inside refrigerated vehicles fluctuate all the time. However, this issue was not considered in this study and is assumed to be constant. A temperature data logger is suggested for future work to consider both fuel consumption in maintaining internal temperature and predicting the freshness of fresh fruits and vegetables at the destination.
- To calculate the heat produced by fresh fruits and vegetables precisely, the respiration rate of each should be collected separately. However, in this study, the respiration rate is assumed to be constant, and the data is taken from other literature.
- Furthermore, the decay rate equation (i.e. Arrhenius equation) could be used to calculate the quality of fresh fruits more precisely than simply using the weight loss rate.
- Because transportation route data collection is not conducted in this study, the route information is taken from other literature, and only exporting fresh fruit from Thailand to China with limited information on transportation routes is considered. Collecting more route options and adding more cases is suggested for further investigation to see how the results vary.

REFERENCES

ALI, O. & VANOUDHEUSDEN, D. 2009. Logistics planning for agricultural vehicles. *2009 IEEE International Conference on Industrial Engineering and Engineering Management.* IEEE.

ARUNMAS, P. 2021. *Fruit exports stand out as the country's sweet spot* [Online]. Available: https://www.bangkokpost.com/business/2160043/fruit-exports-stand-out-as-the-countrys-sweet-spot [Accessed].

ASGHARI, M., FATHOLLAHI-FARD, A. M., MIRZAPOUR AL-E-HASHEM, S. M. J. & DULEBENETS, M. A. 2022. Transformation and Linearization Techniques in Optimization: A State-of-the-Art Survey. *Mathematics,* 10, 283.

AYED, H., HABBAS, Z. & KHADRAOUI, D. 2011. A parallel time-dependent multimodal shortest path algorithm based on geographical partitioning. *2011 Third World Congress on Nature and Biologically Inspired Computing.* IEEE.

BAO, C., XU, L., GOODMAN, E. D. & CAO, L. 2017. A novel non-dominated sorting algorithm for evolutionary multi-objective optimization. *Journal of Computational Science,* 23, 31-43.

BAZARAA, M. S., SHERALI, H. D. & SHETTY, C. M. 2013. *Nonlinear Programming: Theory and Algorithms,* NY,USA, Wiley-Interscience.

BENYOUCEF, L. & XIE, X. 2011. Supply Chain Design Using Simulation-Based NSGA-II Approach. *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing.* Springer London.

BEZERRA, L. C. T., LÓPEZ-IBÁÑEZ, M. & STÜTZLE, T. An Empirical Assessment of the Properties of Inverted Generational Distance on Multi- and Many-Objective Optimization. *In:* TRAUTMANN, H., RUDOLPH, G., KLAMROTH, K., SCHÜTZE, O., WIECEK, M., JIN, Y. & GRIMME, C., eds. Evolutionary Multi-Criterion Optimization, 2017 Cham. Springer International Publishing, 31-45.

BI, X. & WANG, C. 2017. A niche-elimination operation based NSGA-III algorithm for many-objective optimization. *Applied Intelligence,* 48, 118-141.

BIN, L., JIAWEI, L., AIQIANG, C., THEODORAKIS, P. E., ZONGSHENG, Z. & JINZHE, Y. 2022. Selection of the cold logistics model based on the carbon footprint of fruits and vegetables in China. *Journal of Cleaner Production,* 334, 130251.

BORGES, Y. G. F., MIYAZAWA, F. K., SCHOUERY, R. C. S. & XAVIER, E. C. 2020. Exact algorithms for class-constrained packing problems. *Computers & Industrial Engineering,* 144, 106455.

CARAMIA, M. & DELL'OLMO, P. 2020. Maritime Freight Logistics. *Multi-objective Management in Freight Logistics.* Springer International Publishing.

CHEN, J. & SHI, J. 2019. A multi-compartment vehicle routing problem with time windows for urban distribution – A comparison study on particle swarm optimization algorithms. *Computers & Industrial Engineering,* 133, 95-106.

CHEN, L., LIU, Y. & LANGEVIN, A. 2019. A multi-compartment vehicle routing problem in cold-chain distribution. *Computers & Operations Research,* 111, 58-66.

CHIMRAKHANG, J. 2019. *The workers allocation problem on multiple cellular u-shaped under many-objectives.* Master Thesis, Chulalongkorn University.

CHITRA, C. & SUBBARAJ, P. 2012. A nondominated sorting genetic algorithm solution for shortest path routing problem in computer networks. *Expert Systems with Applications,* 39, 1518-1525.

CONNELL, J. P. & WELLBORN, J. G. 1991. Competence, autonomy, and relatedness: A motivational analysis of self-system processes. *Self processes and development.* Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc.

CUI, Z., CHANG, Y., ZHANG, J., CAI, X. & ZHANG, W. 2019. Improved NSGA-III with selection-and-elimination operator. *Swarm and Evolutionary Computation,* 49, 23-33.

DAMRONGPOL, K., GILLES, N., AICHA, S. & PRADORN, S. 2016. Quantification of freshness loss and an optimization approach for managing fresh fruit exportation. *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA).* IEEE.

DEB, K. 2013. Multi-objective Optimization. *Search Methodologies.* Springer US.

DEB, K. & DEB, K. 2013. Multi-objective Optimization. *Search Methodologies.* Springer US.

DEB, K. & JAIN, H. 2013. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation,* 18, 577-601.

DEB, K., PRATAP, A., AGARWAL, S. & MEYARIVAN, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation,* 6, 182-197.

DIVISION OF AGRICULTURE AND NATURAL RESOURCES, U. O. C. *Produce Fact Sheets - UC Postharvest Technology Center* [Online]. Available: https://postharvest.ucdavis.edu/Commodity_Resources/Fact_Sheets/ [Accessed 1 March 2022].

DOKEROGLU, T. & COSAR, A. 2014. Optimization of one-dimensional Bin Packing Problem with island parallel grouping genetic algorithms. *Computers & Industrial Engineering,* 75, 176-186.

DOMUTA, C., TIPI, N. S., RADU, D., HULEA, M. & ASTILEAN, A. 2012. Finding intermodal and multi-objective time-dependent best routes with exact algorithm. *Proceedings of 2012 IEEE International Conference on Automation, Quality and Testing, Robotics.* IEEE.

DULEBENETS, M. A., OZGUVEN, E. E., MOSES, R. & B. ULAK, M. 2016. Intermodal Freight Network Design for Transport of Perishable Products. *Open Journal of Optimization,* 05, 120-139.

FANG, H., WANG, Q., TU, Y.-C. & HORSTEMEYER, M. 2008. An Efficient Non-dominated Sorting

Method for Evolutionary Algorithms. *Evolutionary computation,* 16, 355-84.

FU, B. & LABUZA, T. P. 1993. Shelf-life prediction: theory and application. *Food Control,* 4, 125-133.

GHARSALLI, L. 2022. Hybrid Genetic Algorithms. *Optimization Algorithms [Working Title].* IntechOpen.

GOLDBERG, D. E. 1989. *Genetic algorithms in search, optimization, and machine learning,* Noida, India, Dorling Kindersley under licence from Pearson.

GRÄBENER, T., BERRO, A. & DUTHEN, Y. 2010. Time dependent multiobjective best path for multimodal urban routing. *Electronic Notes in Discrete Mathematics,* 36, 487-494.

GUNANTARA, N. 2018. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering,* 5, 1502242.

GUPTA, J. N. D. & HO, J. C. 1999. A new heuristic algorithm for the one-dimensional bin-packing problem. *Production Planning & Control,* 10, 598-603.

JARUMANEEROJ, P., OAK DUSADEERUNGSIKUL, P., CHOTIVANICH, T. & AKKERMAN, R. 2021. A multi-objective modeling approach to harvesting resource scheduling: Decision support for a more sustainable Thai sugar industry. *Computers & Industrial Engineering,* 162, 107694.

JUNKERMEIER, J. A Genetic Algorithm for the Bin Packing Problem. 2015.

KAEWFAK, K., AMMARAPALA, V. & HUYNH, V.-N. 2021. Multi-objective Optimization of Freight Route Choices in Multimodal Transportation. *International Journal of Computational Intelligence Systems,* 14, 794.

KHAOWONG, W. 2021. *A Development of heuristics for the Airport Gate Assignment Problem.* Master Thesis, Chulalongkorn University.

KIEN HUA, T. & ABDULLAH, N. 2018. Weighted Sum-Dijkstra's Algorithm in Best Path Identification based on Multiple Criteria. *Journal of Computer Science & Computational Mathematics*, 107-113.

KONAK, A., COIT, D. W. & SMITH, A. E. 2006a. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety,* 91, 992-1007.

KONAK, A., COIT, D. W. & SMITH, A. E. 2006b. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety,* 91, 992-1007.

KREBS, N. & PANICHAKARN, B. 2019. Optimal Route in International Transportation of Thailand – Guangxi (China). *GATR Global Journal of Business Social Sciences Review,* 7, 33-47.

KRISANAPHAN, N. 2019. *Solving many-objective cockpit crew pairing problem of low-cost airline.* Master Thesis, Chulalongkorn University.

LAWTON, R. 2016. How Refrigerated Containers Work. *Reference Module in Food Science.* Elsevier.

LI, X. 2014. Operations Management of Logistics and Supply Chain: Issues and Directions. *Discrete Dynamics in Nature and Society,* 2014, 1-7.

LODI, A., MARTELLO, S., MONACI, M. & VIGO, D. 2013. Two-Dimensional Bin Packing Problems.

Paradigms of Combinatorial Optimization. John Wiley & Sons, Inc.

LUFU, R., AMBAW, A. & OPARA, U. L. 2020. Water loss of fresh fruit: Influencing pre-harvest, harvest and postharvest factors. *Scientia Horticulturae,* 272, 109519.

MAIORINO, A., PETRUZZIELLO, F. & APREA, C. 2021. Refrigerated Transport: State of the Art, Technical Issues, Innovations and Challenges for Sustainability. *Energies,* 14, 7237.

MARLER, R. T. & ARORA, J. S. 2004. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization,* 26, 369-395.

MOFADDEL, M. A. & HAMED, A. Y. 2018. Multi-Objective Optimization to Find The Shortest Paths Tree Problem in The Computer Networks. *Applied Mathematics & Information Sciences,* 12, 151-158.

MOLINA, J. A. N. & MISHRA, S. Addressing memory exhaustion failures in Virtual Machines in a cloud environment. 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 24-27 June 2013 2013. 1-6.

MURATA, T. & ISHIBUCHI, H. 1995. MOGA: multi-objective genetic algorithms. *Proceedings of 1995 IEEE International Conference on Evolutionary Computation.* IEEE.

MURRAY-SMITH, D. J. 2012. Experimental modelling: system identification, parameter estimation and model optimisation techniques. *Modelling and Simulation of Integrated Systems in Engineering.* Elsevier.

NAKANDALA, D., LAU, H. & ZHANG, J. 2016. Cost-optimization modelling for fresh food quality and transportation. *Industrial Management & Data Systems,* 116, 564-583.

OSTERMEIER, M., HENKE, T., HÜBNER, A. & WÄSCHER, G. 2021. Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions. *European Journal of Operational Research,* 292, 799-817.

OSTERMEIER, M. & HÜBNER, A. 2018. Vehicle selection for a multi-compartment vehicle routing problem. *European Journal of Operational Research,* 269, 682-694.

OSTERMEIER, M., MARTINS, S., AMORIM, P. & HÜBNER, A. 2018. Loading constraints for a multi-compartment vehicle routing problem. *OR Spectrum,* 40, 997-1027.

OSVALD, A. & STIRN, L. Z. 2008. A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. *Journal of Food Engineering,* 85, 285-295.

PAAM, P., BERRETTA, R., HEYDAR, M., MIDDLETON, R. H., GARCÍA-FLORES, R. & JULIANO, P. 2016. Planning Models to Optimize the Agri-Fresh Food Supply Chain for Loss Minimization: A Review. *Reference Module in Food Science.* Elsevier.

PAIXÃO, J. M. & SANTOS, J. L. 2012. Labeling Methods for the General Case of the Multi-objective Shortest Path Problem – A Computational Study. *Intelligent Systems, Control and Automation: Science and Engineering.* Springer Netherlands.

PELEG, M., NORMAND, M. D. & CORRADINI, M. G. 2012. The Arrhenius Equation Revisited. *Critical Reviews in Food Science and Nutrition,* 52, 830-851.

PONGPANICH, C. & PHITYA-ISARAKUL, P. 2008. Enhancing the Competitiveness of Thai Fruit Exports: An Empirical Study in China. *Contemporary Management Research,* 4.

QIN, G., TAO, F. & LI, L. 2019. A Vehicle Routing Optimization Problem for Cold Chain Logistics Considering Customer Satisfaction and Carbon Emissions. *International journal of environmental research and public health,* 16, 576.

RAHBARI, A., NASIRI, M. M., WERNER, F., MUSAVI, M. & JOLAI, F. 2019. The vehicle routing and scheduling problem with cross-docking for perishable products under uncertainty: Two robust bi-objective models. *Applied Mathematical Modelling,* 70, 605-625.

RAITH, A. & EHRGOTT, M. 2009. A comparison of solution strategies for biobjective shortest path problems. *Computers & Operations Research,* 36, 1299-1331.

REED, P., MINSKER, B. & GOLDBERG, D. 2003. Simplifying Multiobjective Optimization — An Automated Design Methodology for the Nondominated Sorted Genetic Algorithm-II. *Water Resources Research - WATER RESOUR RES,* 39.

ROHLFSHAGEN, P. & BULLINARIA, J. A. 2007. A genetic algorithm with exon shuffling crossover for hard bin packing problems. *Proceedings of the 9th annual conference on Genetic and evolutionary computation - GECCO '07.* ACM Press.

ROSENTHAL, S. & BORSCHBACH, M. 2014. Impact of population size, selection and multi-parent recombination within a customized NSGA-II and a landscape analysis for biochemical optimization. *International Journal on Advances in Life Sciences,* 6, 310-324.

ROSITA, Y. D., ROSYIDA, E. E. & RUDIYANTO, M. A. 2019. Implementation of Dijkstra Algorithm and Multi-Criteria Decision-Making for Optimal Route Distribution. *Procedia Computer Science,* 161, 378-385.

SANTOS, T. & XAVIER, S. 2018. A Convergence Indicator for Multi-Objective Optimisation Algorithms. *TEMA (São Carlos),* 19, 437.

SARDER, M. D. 2021. Intermodal transportation. *Logistics Transportation Systems.* Elsevier.

SHAO, C., WANG, H. & YU, M. 2022. Multi-Objective Optimization of Customer-Centered Intermodal Freight Routing Problem Based on the Combination of DRSA and NSGA-III. *Sustainability,* 14, 2985.

SONG, B. D. & KO, Y. D. 2016. A vehicle routing problem of both refrigerated- and general-type vehicles for perishable food products delivery. *Journal of Food Engineering,* 169, 61-71.

SOTO-SILVA, W. E., NADAL-ROIG, E., GONZÁLEZ-ARAYA, M. C. & PLA-ARAGONES, L. M. 2016. Operational research models applied to the fresh fruit supply chain. *European Journal of Operational Research,* 251, 345-355.

STELLINGWERF, H. M., KANELLOPOULOS, A., VAN DER VORST, J. G. A. J. & BLOEMHOF, J. M. 2018. Reducing CO 2 emissions in temperature-controlled road transportation using the LDVRP model. *Transportation Research Part D: Transport and Environment,* 58, 80-93.

SUN, Y., YEN, G. G. & YI, Z. 2019. IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation,* 23, 173-187.

SURUCU-BALCI, E. & TUNA, O. 2021. Investigating logistics-related food loss drivers: A study on fresh fruit and vegetable supply chain. *Journal of Cleaner Production,* 318, 128561.

VERBIC, M. 2006. Discussing the parameters of preservation of perishable goods in a cold logistic chain model. *Applied Economics,* 38, 137-147.

WANG, S., TAO, F. & SHI, Y. 2018. Optimization of Location-Routing Problem for Cold Chain Logistics Considering Carbon Footprint. *International journal of environmental research and public health,* 15, 86.

WANG, X. & LI, D. 2012. A dynamic product quality evaluation based pricing model for perishable food supply chains. *Omega,* 40, 906-917.

WANG, X., WANG, M., RUAN, J. & ZHAN, H. 2016. The Multi-objective Optimization for Perishable Food Distribution Route Considering Temporal-spatial Distance. *Procedia Computer Science,* 96, 1211-1220.

YAHYAOUI, H., KAABACHI, I., KRICHEN, S. & DEKDOUK, A. 2018. Two metaheuristic approaches for solving the multi-compartment vehicle routing problem. *Operational Research,* 20, 2085-2108.

YAO, Q., ZHU, S. & LI, Y. 2022. Green Vehicle-Routing Problem of Fresh Agricultural Products Considering Carbon Emission. *International Journal of Environmental Research and Public Health,* 19, 8675.

YUAN, X., TIAN, H., YUAN, Y., HUANG, Y. & IKRAM, R. M. 2015. An extended NSGA-III for solution multi-objective hydro-thermal-wind scheduling considering wind power cost. *Energy Conversion and Management,* 96, 568-578.

ZHANG, Q., ZHOU, A., ZHAO, S., SUGANTHAN, P., LIU, W. & TIWARI, S. 2008. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. *Mechanical Engineering*.

ZHOU, L., LI, Q., LI, F. & JIN, C. 2022. Research on Green Technology Path of Cold-Chain Distribution of Fresh Products Based on Sustainable Development Goals. *Sustainability,* 14, 16659.

ZITZLER, E., THIELE, L., LAUMANNS, M., FONSECA, C. M. & DA FONSECA, V. G. 2003. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation,* 7, 117-132.

VITA

NAME	Supisara Krairiksh
DATE OF BIRTH	18 July 1992
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Industrial Engineering, University of Missouri - Columbia