Modeling and Optimization of Meat Supply Chain Transportation Network

Walaa Awad Ali

A Thesis

in

The Department

of

Mechanical, Industrial and Aerospace Engineering (MIAE)

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Industrial Engineering) at

Concordia University

Montreal, QC, Canada

November 2021

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Walaa Awad Ali**

Entitled: **Modeling and Optimization of Meat Supply Chain Transportation Network**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Industrial Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

________________________________ Chair
Dr. Gerard J. Gouw

________________________________ Examiner
Dr. Daria Terekhov

________________________________ Examiner
Dr. Gerard J. Gouw

________________________________ Supervisor
Dr. Mingyuan Chen

Approved by ________________________________
Martin D. Pugh, Chair
Department of Mechanical, Industrial and Aerospace Engineering (MIAE)

____________ 2021

____
Mourad Debbabi, Dean
Concordia University

Abstract

Modeling and Optimization of Meat Supply Chain Transportation Network

Walaa Awad Ali

In this thesis, a comprehensive meat supply chain (MSC) planning problem is studied. We consider live animals, chilled and frozen meat products as it is in the real case in most meat supply chain industries. A literature review is presented, and optimization-based method has been proposed in food supply industries. Two mathematical models are proposed considering comprehensive meat supply chain problems. The considered meat supply chain problem was decomposed into two optimization sub-problems. The approach is to use two separate mathematical models for solving each sub-problem. The developed mathematical models are examined on a practical system in a meat supply chain company based in United Arab Emirates. The heuristic solution approach is introduced to solve the comprehensive meat supply chain optimization model developed. It can generate a good solution for all tested example problems with much reduced computational time.

Acknowledgments

In the name of Allah, Most Gracious, Most Merciful. First, I would like to express my sincere and earnest gratitude to my supervisors, Prof. Dr. Mingyuan Chen for his guidance, and full support during my graduate studies, this thesis would not have been achievable without his encouragement, inspiration, deep insights, and kind supervision. Working with him was a real pleasure. Special thank to my examination committee, Professor Gerard Gouw and Professor Daria Terekhov for their time in reviewing my thesis. I am over helmed in all humbleness and gratefulness to acknowledge my depth to all those who supported and helped me to complete my thesis. Special thanks to Eng. Waleed Khalil – Managing Director and General Manager of Amecath Medical Technologies for all his support and guidance. I am also would like to take this opportunity to express a deep sense of gratitude to Emirates Future group and the best team I have ever worked with, Mr. Hussein Ghousheh, Mr. Kamil Jasim, Mrs. Rawan Shamlouli, Mr. Yaser Ali, Mr. Jalal Suliman, Mr. Mohammed Baabbad, Mr. Mohammed Fathy, Dr. Ahmed Ali, Mr. Ahmad Talat, Mrs. Heba Mourad, Ms. Raqel Gumboc, Mr. Abdul Saleem, Mr. Salman Ali. My sincere thanks to Dr. Samir Eldessouky, Dr. Noha Hussein, Dr. Abdelaziz Soufyane, Dr. Mahmoud Awad, Dr. Mama Chacha, Dr. Tawfiq Jaber, Dr. Tarik Aouam, Dr. Samia Loucif, Dr. Muhamad Hasab Alnabii, for their support and encouragement during my bachelor's degree. To all my colleague and friends, Dr. Jair Ferrari and his family, Dr. Bai Qinggue, Dr. Zhifeng Zhao, Dr. Chaoqun Dong, Dr. Qiong Wei, Dr. Abdullahi Gujba , Ladan Zamirian, Marjan Sadeghi, Ruo Liang, Yasser Ghamary, Tiansheng Zhang, Zenan Zhang, Lisa Zhao, Zhen Yan, Shirin Arabshahi, Nooshin Salehabadi, and Alexander Bryce thank you very much

for all your support and I am lucky to know all of you. As well many thanks to all administration staff at the department of Mechanical, Industrial and Aerospace Engineering (MIAE) at Concordia University. To all my relatives and friends, thank you very much for your support and prayers. To my beloved family who supported and believed in me, my parents, Awad Ibrahim, Wafaa Ali, my sisters, Dr. Doaa Awad and Dr.Dyana Elsadek, my brothers, Dr. Muhammed Awad, Sayed Awad, and Abdelrahamn Awad, thank you for everything you have done for me. Finally, to my caring, loving, and supportive husband Dr. Omar Awni Abuobidalla, Assistant Professor at the department of Mechanical, Industrial and Aerospace Engineering at Concordia University, my deepest gratitude for your usual support, patience, and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the meat supply chain (MSC). The challenges of this problem in real life are presented. A summary of the main contributions of this thesis is discussed in the next section.

1.1 Background

Food industry is one of the main important industries in many countries and regions. In recent decades, rapid population growth has led to a significant increase in food demand. It was estimated that by 2050 the world must feed about 9 billion people. Such increase in the population density will increase the food consumption, not less than 60% than it is today. Moreover, the world supply chain has been changed and affected all sectors such as agriculture, health, and industry due to COVID pandemic. Nowadays most of countries is considering having a food security with a sustainable supply chain especially for the main products to avoid any shortage.

1.2 Meat supply chain

While the supply chain processes for specific meat and poultry commodities vary greatly from country to country, the primary activities of meat supply chain process are similar. It usually begins with the birth of the animal, followed by growth, slaughtering, butchering, processing, distributing, and distribution to store retailer. The meat and poultry supply chain usually includes:

- Farms
- Abattoirs
- Reproduction plants
- Dead stock collection points
- Border ports
- Quarantine stations
- Warehouses
- Distribution centers
- Cold storage facilities
- Retail grocery stores
- Food service operator restaurants

The meat supply chain is complex supply chain activities, involving crop farms, livestock farms, feedlots, transporters, abattoirs, retailers and consumers. As shown in the Figure 1.1 of stream map process, the information flows from the downstream process (retailers) to upstream process (sources of livestock), while the meat and meat products move downstream from sources of livestock to retailers. The production of livestock is typically central integrated industry where most supply chain activities are integrated and owned by one wholesaler.

Tactical meat supply chain planning problem considered in this thesis is one of the comprehensive models that considers all supply chain activities. It includes such decisions as raw material selections from various sources of origins, transportation, production planning, inventory management decisions and deliveries to end customers to satisfy the demand of each planning horizon. The periods in the model are typically of intermediate duration such as few weeks, month in a year. Meat product process is either joining or splitting process.

In the former processes, products are defined through their bill of materials, while the latter processes extracted the final products through decomposition of the main commodity. The model maximizes the total profit minus the total cost from different supply chain activities such as contractual charges, replenishment , and meat waste costs. The customer demands might not be satisfied due to complex nature of problem studied in this thesis and will be consider as lost sale through back-orders penalty. The inventory cost at retailer is explicitly considered, assuming no inventory is being hold through abattoir centres. We also consider the capacity and lead time requirements for the upstream process. The price of the meat and meat products is considered as function of production time. The tactical meat supply chain planning problem is treated as master planning including transportation and scheduling decisions. The model is solved to optimality using a mixed-integer programming (MIP) solver. Even through real case instances may be largely due to the process of large number of time slots, the solution times using CPLEX solver are typically modest. Optimizing mathematically the meat supply chain planning process will be more beneficial when the tactical plan problem considers routing and scheduling decision of the livestock from sources of origin. In such cases, the meat wholesaler also needs to decide (a) sea vessel services and trip plan of freights, (b) airline services, (c) quantity of livestock to be shipped, and (d) schedule of each freight in order to reduce the meat waste while satisfying operational constraints. Following a well-defined freight operation plan may not only helpful in reducing the operation costs but also may reduce the meat waste from deterioration.

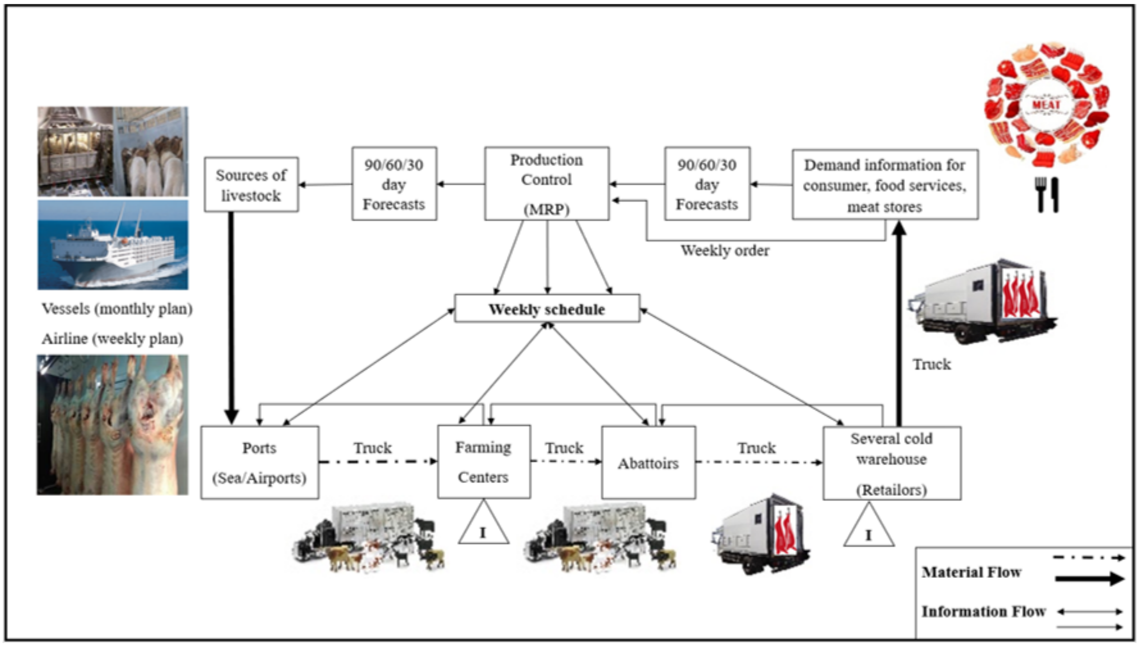

Figure 1.1: Stream map process

1.3 Challenges and motivation

Food supply chains including meat supply are crucial for every society's development, survival, and security. The demand rate of meat continues to grow in markets. The meat supply chain process is of a complex nature since the logistic of meat and its products' have special requirements such as limited shelf-life cycle, product deterioration, demand fluctuations, environmental impacts compared to other supply chain activities, and vulnerability to inclement weather (Tendall et al. (2015)). The health pandemic due to the COVID-19 virus showed the vulnerability of the meat supply chain. Most of large meat companies learned from this experience and started to plan for food security, supply chain, risk assessment and business continuity management system (ISO 22301) to secure their businesses. The prices of meat and its products grew sharply mainly due to less production and a sharp increase in demand rate from panic behaviour of consumer buying. In early 2020, meat processing factories started to shut down due to the increase in the COVID-19 cases among associates and COVID restriction measures, such as the percentage of allowed employee in each department and new distances measures. Furthermore, meat supply chain partners including farms and abattoirs faced difficulty in harvesting and shipping products due to lock-down situations, decreases in the labor force, restrictions in movement of animals within and across the country, and changes in the legislation and health protocols of local and international meat export market. These conditions negatively affected the meat industry, and it is suggested that the integration among all the meat industry partners is quite crucial for the sustainability of the meat supply chain processes.

In a summary, the rapidly growing nature of the COVID-19 virus creates many problems for the meat supply chain industry. The scarcity of animals' exportation, logistics limitations, the closing of abattoirs, accelerate the growth of meat prices which significantly affected all activities of the meat supply chain. Farmers were unable to ship or sell their live animals to the market. The abattoir capacity was reduced due to the closure of the processing plants. The shortage of livestock and meat products for the retailers due to manufacturing facility closures causes price fluctuations of meat and its products.

Despite the acknowledgement of the importance of continuity of the meat supply chain process, the integration of planning of meat supply chain practices remains mostly limited and under explored as indicated in the literature (Gholami-Zanjani, Jabalameli, and Pishvaee (2021), among others). Integrated planning of the meat supply chain can be defined as managing the logistics of all activities starting from source of livestock to the retailer. The ability to manage all activities of the meat process will lead to absorbing instabilities and protecting basic functionality against disruptions.

The meat supply chain generally consists of main-three activities: breeding centres/farms, abattoirs/slaughterhouses, and retailers used and customer points. At abattoirs, several products can be obtained such as feathers, bones, and fats. At the tactical level, the meat supply chain planning gives the allocation decisions, inventory management, and replenishment decisions. In this thesis, we assume a central decision to manage the whole logistics. The allocation decisions studies in this thesis are considered as contractual agreements with upstream providers.

The meat supply chain problem discussed in this thesis focuses on finding the optimal allocation decisions, replenishment of retailer, and routing and scheduling decisions in a four-echelon network of sources of livestock, farming centres, abattoirs, and retailers. Both sources of livestock and farming are considered sources of raw materials and livestock in the meat supply chain. However, the main difference is that the sources of livestock have abundant sources of livestock, i.e., Australia, Brazil, Canada. Abattoirs are usually in the same area of farming centres, in which various meat parts will be extracted from the livestock. Abattoirs typically do not hold any inventory and are treated as conversion points in the meat supply chain. The final layer in the meat supply chain is the retailers, which consider as sale points. The retailers are considered as cold warehouses that deals with hundreds or thousands of consumers, i.e., Walmart, Costco. The consumer includes supermarkets, meat stores, or restaurants. We assume central decision-maker manages the whole supply chain actives including transportation of the livestock from sources of origin using their own sea vessels. In addition, the wholesaler typically arranges airline services to ship urgent meat products.

Following the literature in the food supply chain, we explicitly consider the time-dependent properties of the price of meat products by adopting an exponential function as in Gholami-Zanjani et al. (2021), Juneja and Marks (2006) and Mochizuki and Hattori (1987). The price of meat products declines as a function of time reflecting the age/freshness of the meat product. In meat industries, the retailers usually adopt the first expired, first-out declines. In cold chain logistics, the retailers dispose of the product which lifetime has expired. It is also common that retailers decide to convert the meat into other meat products to increase its lifetime. For simplicity, the last approach will not be explored. The main objective of the proposed model is deciding the allocation, inventory management, routing, and scheduling decisions. It is also not obligatory to satisfy all meat demands due to the nature of the problem. The model proposed in a multi-period planning time horizon with discrete time, integrating network design, inventory management, and routing and scheduling decisions.

1.4 Contribution

This research proposes two mathematical models for comprehensive meat supply chain, we integrated routing and scheduling of meat supply chain; inventory management; and network design of meat supply chain decisions. In this study, we considered a central decision maker, wholesaler distributor manages integrated meat supply chain including transportation and inventory decisions at retailer and upstream process. We also assume that there is no competition between supply chain entities. The decision maker aims at finding the optimal allocation decisions and replenishment of retailers and product flows in a four-echelon network of farms, abattoirs, and retailers. As well as the shipment itinerary from sources supply as source of materials to ports, in which live animals are shipped to farms. For the chilled and frozen meat it is sent directly to retailers or intermediate warehouse. In this thesis, the interdependencies between the meat supply chain stages are considered and optimized. We assume that the abattoirs are intermediate facility in the meat supply chain network and do not hold inventory. Customers or retailer could be supermarket, meat stores or restaurants. Following the previous studies, we consider the price of meat product as function of freshness. It is not obligatory to satisfy all demands due to the difficulty of the problem and the unmet demands are consider as lost. The main contributions of this thesis.

(1) A comprehensive mixed integer programming model is developed for determining simultaneously the allocation, inventory management, and freight transportation decisions. The existing research in this area is limited as shown in literature.

(2) Live animals, chilled and frozen meat products are explicitly considered in developing the optimization models.

(3) The developed models are solved to find optimal or near optimal solutions of the model with reduced computation for adequate and effective applications in practice.

(4) Realistic problem instances based on real problem data and information were solved and tested to demonstrate the validity of the developed model and efficiency of the

heuristic solution approach.

1.5 Thesis outline

In the next chapter, the relevant literature's on routing and scheduling of agricultural products, inventory management and network design problem of food supply chain problem are reviewed. In chapter 3, we present real case study for meat supply chain (MSC) and then problem statement is introduced and two mathematical models are developed with two phases. The following chapter we presented the result for both models of the case study, A sensitivity analysis is conducted, and a heuristic approach is introduced to explain the methodology of solving this problem. Finally, chapter 5 presents the conclusion and the future research work of the thesis.

Chapter 2

Literature Review

In this chapter, we review the related studied on meat and poultry supply chain optimization.

2.1 Related literature

As the proposed investigation related to multiple sub-areas within meat supply chain optimization domain, we organize the literature review into three major groups: (a) routing and scheduling; (b) inventory management; and (c) network design of food supply chain problem to better situate this research within the existing literature. Comprehensive literature reviews on environmental supply chain network design and logistics related aspects are outlined in Soto-Silva, Nadal-Roig, González-Araya, and Pla-Aragones (2016), Ahumada and Villalobos (2009), Resat and Turkay (2019), Brandenburg and Rebs (2015), Dekker, Bloemhof, and Mallidis (2012) ,Demir, Bektaş, and Laporte (2014) and Eskandarpour, Dejax, Miemczyk, and Péton (2015).

2.1.1 Routing and scheduling of agricultural products

The supply chain of agricultural products such as meat and poultry products has received a great attention lately due to issues related to COVID-19. In the routing and scheduling of agricultural products, the work focuses on transportation decisions from farmer to the abattoirs and from the abattoirs to the retailer. The main objective of their models is to determine the assignment of meat commodities to a set of vehicles assuming the centroid of customer clusters (set of retailer stores), and the location the farmer and the abattoirs in such a way that total logistics costs are minimized. One of the key aspects of transportation of agricultural products is their limited shelf life, their demand and price variability, which makes the underlying supply chain more complex and harder to manage compared to other supply chains. Agri-food supply chains have primary two types; the first one is the supply chain of fresh agri-foods, and the second one is the supply chain for non-perishable agri-foods. The former supply chain includes highly perishable products with very short life cycle such as milk, fresh meat, fruits and vegetables whose useful life can be evaluated in days or weeks; non-perishable products are those that can be stored for longer periods of time such as grains. Several research works investigate different aspect of the agri-food supply chain optimization. García-Flores, Higgins, Prestwidge, and McFallan (2014) investigated the meat supply in Northern Australia and proposed a mathematical model to obtain the optimal quantity of products to be transported in multiple echelons of the network. Mohammed, Wang, and Li (2015) developed a multi-objective mathematical model with the aim of maximizing the integrity of meat, minimizing the whole investment spending, and maximizing the return of investments. Mohammed and Wang (n.d.) extended their work to deal with four objectives simultaneously: minimizing total cost, and maximizing profits, freshness, and customers satisfaction.

2.1.2 Inventory management

The inventory planning determines how to stock the commodity and when to replenish. There are several inventory planning strategies, such as (s, S), (R, Q), (s, Q) and (R, S), in which s, S, R and Q are defined as reorder point, up-to-level point, inventory review periods, and order quantity, respectively. One of the most common inventory policies for fast-moving consumer goods (FMCG) is the (s, S) replenishment policy. In the (s, S) replenishment policy, if the amount of inventory falls below the reorder point s, an order is placed to increase the inventory position level up to point S after lead time period. Martel (2003) studied rolling planning horizon policies to manage material flows in multi-echelon supply distribution networks with stochastic demand processes and procurement, transportation, inventory, and shortage cost structures. A two-product inventory model with limited storage capacity and constant demand rates was explored by Beemsterboer, Teunter, and Riezebos (2016). Mogale, Kumar, Kumar, and Tiwari (2018) proposed integrated multi-objective, multi-modal and multi-period mathematical model for grain silo location-allocation problem with dwell time. Duan and Liao (2013) showed that centralized control over the whole supply chain supports in reducing the expected outdated rate for highly perishable product.

Vertical cooperation between all entities in the meat supply chain can reduce meat deterioration by reducing production/transportation lead times or optimizing production. In the Canadian's food industry, some meat industries choose to control downstream sellers. This type of integration is called known as forward integration. After procuring meat products from upstream sources suppliers, they hold the products in their refrigerated warehouses after process at abattoirs and deliver the food products to its own sales stores. Such approach helps to the wholesaler to control the price in the market and to response quickly to change in demand rates. It also helps the wholesaler to response quickly under supply disruption. Alternative approach known as "backward integration" strategy where the collaboration between upstream producers or raw material suppliers and vendors. The central meat industries signs agreements with the downstream retailers. The latter approach has better control the production process and quality of the meat products compared to the former approach. Huang, He, and Li (2018) proposed a stackelberg gaming model for a three-level food supply chain consisting of one retailer, one vendor and one supplier with production disruption.

2.1.3 Network design of food supply chain problem

In the network design of food supply chain problem aims at designing and configuring a multi-period, multi-product, multi-echelon green meat supply chain network. The network design planning model determines the optimal number of products to be transported in each echelon of the network, and the optimal number and allocation of farms, abattoirs, and retailers. For instance, Mohebalizadehgashti, Zolfagharinia, and Amin (2020) developed a multi-objective mixed-integer linear programming formulation to optimize three objectives: (1) minimization of the total cost, (2) minimization of the total CO2 emissions released from transportation, and (3) maximization of the total capacity utilization of facilities. The model was applied to a green meat supply chain network for Southern Ontario, Canada. Jain, Groenevelt, and Rudi (2011) developed a conceptual model to estimate the parameters of the (s, S) policy to optimize the performance of the network. Under a periodic review inventory management, they proposed a heuristic method to find bound solution to estimate the (s, S) parameters. Cunha, Raupp, and Oliveira (2017) proposed a mixed integer programming model to optimize replenishment policy for a single-product single echelon network with variable order quantities. A multi-sourcing strategy to deal with non-stationary demand was introduced by Amiri-Aref, Klibi, and Babai (2018). Academic works on inventory management in a multi-echelon meat supply chain network with multiple facilities in each echelon are limited. Some studies simultaneously incorporate resilience and greenness perspectives. Tendall et al. (2015) developed concept framework for food system resilience and considered how this could be implemented through stakeholder participation. The avian flu pandemic of 2006 interrupted the entire poultry supply chain in several countries, such as in France Vo and Thiel (2011). Although meat supply chain management has been widely investigated, the focus was on a single meat industry. Gholami-Zanjani et al. (2021) developed bi-objective stochastic model to cover specific characteristics of meat inventory planning in a three-echelon network. Two resiliency strategies are adapted in their model to protect against the disruptions. Sample average approximation and Lexicographic Weighted Tchebycheff methods were proposed. They concluded that resilient solutions could recover the network performance by 6%. Workie, Mackolil, Nyika, and Ramadas (2020) conducted a comprehensive assessment of the effect of COVID-19 on food security and culture suggested coping and mitigation mechanisms that can be implemented to maintain livelihoods. Food supply chain resiliency can be defined as the capability to handle instabilities and protect against any disruptions in the supply chain. Rahbari, Hajiagha, Mahdiraji, Dorcheh, and Garza-Reyes (2021) developed two-stage model consisting of a single-echelon and a three-echelon many-to-many network with static and deterministic demand. Banasik, Kanellopoulos, Claassen, Bloemhof-Ruwaard, and van der Vorst (2017) considered alternative production options in food supply chain to deal with food waste. The objective of their model is to develop a mathematical model that can be used for such a quantitative assessment of alternative production deal with waste in food supply chains, i.e., recycling, and disposal of food waste. Golini, Moretto, Caniato, Caridi, and Kalchschmidt (2017) studied of the Italian meat industry considers the SC as a whole, considering critical points for each echelon in terms of economic, environmental and social sustainability. The necessity of multiple time periods, integration of the whole supply chain including allocating, transportation and storage decisions, environmental considerations and their conflicting nature are defined in the research works. As can be seen from those studies on food supply chain, the routing and scheduling of food supply chain; inventory management; and network design of meat supply chain decisions are highly interrelated. Limited studies have considered the integrated meat supply chain problem from their sources supply to retailers. Due to the complex nature of food supply chains, few studies on meat supply chain network explore multi-echelon supply chain. The planning of integrated optimization models typically generates more profit than the planning from decentralize planning decisions, i.e., O. Abuobidalla, Chen, and Chauhan (2019); O. A. Abuobidalla, Chen, and Chauhan (2021). In this thesis, we attempt to integrate routing and scheduling of food supply chain; inventory management; and network design of meat supply chain decisions. In this study, we assume a central decision maker, Wholesale meat distributor, manages the transportation and inventory at retailer and upstream process of meat products. We also assume that there is no competition between supply chain entities. The decision maker aims at finding the optimal allocation decisions and replenishment of retailers and product flows in a three-echelon network of breeding centres, abattoirs, and retailers as well as the shipment itinerary from sources supply as source of materials to ports in which live animals are shipped to farms and chilled and frozen meat products sent directly to retailers or intermediate warehouses. In this thesis, the interdependencies between the meat supply chain stages are considered and optimized. We assume that the abattoirs are intermediate facility in the meat supply chain network and do not hold inventory. Customers or retailer could be supermarket, meat stores or restaurants. Following the previous studies, we consider the price of meat product as function of freshness. It is not obligatory to satisfy all demands due to the difficulty of the problem and the unmet demands are consider as lost.

In next chapter, two mathematical models are developed to decide simultaneously allocation, inventory management, and freight transportation decisions in the presence of live animals (sheep and cattle), chilled, and frozen meat freights. The considered proposed problem captures most of features of the operational planning level of real-world meat supply chain problems. The proposed thesis has the following contributions. First, a comprehensive mixed integer programming models is proposed for determining simultaneously the allocation, inventory management, and freight transportation decisions. Second, the live animal, chilled and frozen meat products were explicitly planned together. Finally, we tackled several realistic-size problem instances by decomposing the problem into two optimization problems progressively solved.

Chapter 3

Problem Statement and Mathematical Formulation

3.1 Introduction

In this chapter, a real case study of integrated planning for meat supply chain (MSC) is studied; considering all activities of the MSC starting from the origin source of the meat livestock till it reaches to the consumers. Two mathematical models are developed, objective function of the first model is to minimize the transportation costs from serving each commodity using a combination of three modes, while the second model aims to get the optimal allocation decision from farms to abattoirs and from abattoirs to retailers, and replenishment of inventory in retailers and amount of meat products shipped in the three-echelon network. In the second model of the meat chain optimization, a planning and inventory management approach is adapted similar to that in Gholami-Zanjani et al. (2021). However, the main differences between the integrated meat supply chain (MSC), the planning and inventory management problem studied in Gholami-Zanjani et al. (2021) are the following aspects:

- Solving the multi-time period meat supply chain planning problem is to consider freight transportation to deliver different types of commodities to sets of farms and retailers using variety transportation modes via truck, airplane, and vessel. The approach of this model is to minimize the total cost in addition to the costs of transporting the commodities, as well booking cost associated with fulfilling all demands. The result of the model is to determine for each demand the itinerary that must follow from it is original sources to destination ports such as (airport or seaport).

- In this thesis, we assume a central decision maker manages the whole meat supply chain planning problem including transportation, allocation, and inventory decisions regrading the amount to be shipped to each activity at different periods.
- The whole supply chain optimization problem was divided into two phases optimization planning models. In the first phase, the model aims to find the optimal decisions regarding allocation farm to abattoir and finally to retailer and inventory management decisions for the retailers. The second phase of the model is to find the optimal planning decisions to transport the commodities from the source of origins to the port of destinations. A summary of the MSC problem is given in Figure 3.1.

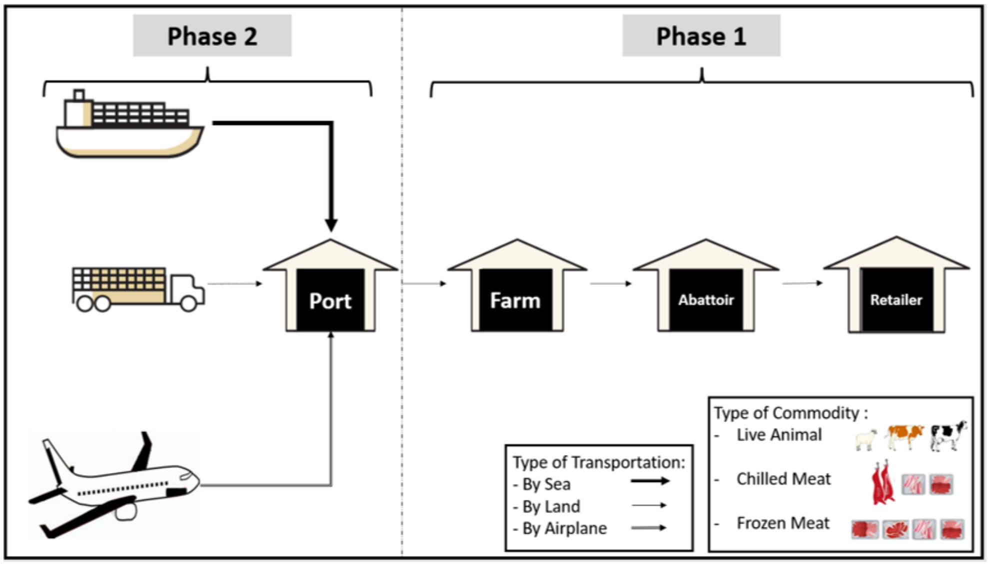

Figure 3.1: Integrated planning for meat supply chain (MSC)

3.2 Objective of models

The objective of these two models is to solve a comprehensive meat supply chain (MSC) planning problems from farms to retailers as a real-world situation.

3.3 Definitions

3.3.1 Allocation cost

The common practice in meat supply chain companies is to handle a list of farms and abattoirs in the territory of the service covered if it was not owned or managed by their end. By signing a legal agreement, the whole seller will be charged annual fees able to use any abattoir services. This fee is the allocation cost and it varies from each abattoir to another, but it is usually fixed amount. Moreover, these agreements are very helpful in case of worldwide livestock trade and to ensure food security and traceability in case of any crises.

3.3.2 Handling cost

All abattoirs have a fee for handling the livestock and this fee depends on each livestock type i.e., sheep or cattle as well the weight of each animal, slaughtering cattle weighted less than 250 kg is cheaper than slaughtering cattle above 250 kg. This handling cost is the cost of butcher one head of livestock (sheep or cattle) and may include other services such as special cuts, cleaning and packing final product as per the client and quality requirements. In the field of livestock business, there is a cost for feeding each animal per day. This cost is very important factor to reflect the handling/feeding cost in each farm to feed the livestock until it has been processed in the abattoir.

3.3.3 Holding cost

Most of fast-moving consumer goods (FMCG) companies store products to accommodate customers demand which fluctuates over time. Therefore, there is a cost to hold the item at the retailers reflecting the cost of storing under the required specifications for each product to keep it fresh, counting and dispatching etc.

3.3.4 Waste cost

The waste of product is considered as main cost in meat trading. The input product is live animal but during the different activities in the supply chain, the output is changed to meet with different types of meat cuts. In abattoir after slaughtering the animals, for example if cattle are weighted 450 kg; the total amount of meat ready for sale will be around 285 kg, with an average of 63% in cattle. Figure 3.2 below depicts the location of the four primary cuts (chuck, rib, loin, and round) on the beef carcass, as well as the remaining sub-primal cuts (shank, brisket, plate, and flank). The four beef primal cuts make-up greater than 75% of the entire weight of the carcass.

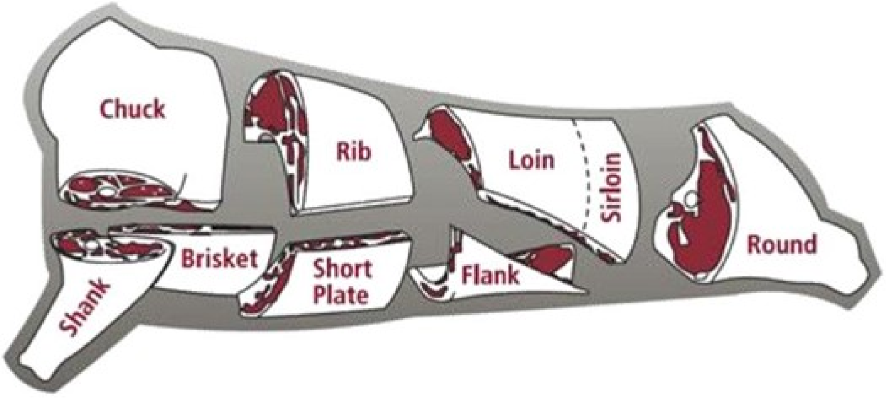

Figure 3.2: Beef carcass primal and sub-primal cuts

(source from http://www.omafra.gov.on.ca/english/livestock/beef/news/vbn0712a4.htm)

3.3.5 Inventory (s, S) policy

The inventory (s, S) policy is one of the common methods used to prevent shortage of inventory. As shown in Figure 3.3 s is the reorder point and the S is the order-up to level. The difference between the two levels is defined by the fixed cost associated with the transportation, order and the process. When the meat is running out of inventory level and it drops below the reorder point s we place an order to increase the inventory up to level S.

The reorder point is a function of the lead time , average and variability of demand. There is another common inventory strategy in MSC is to ship a constant amount of livestock with yearly contract.

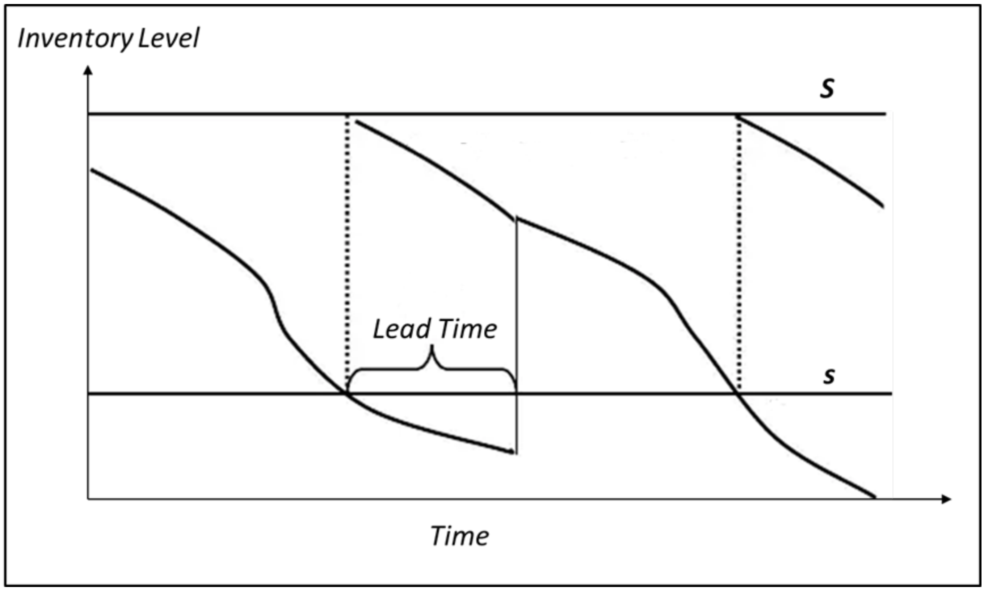

Figure 3.3: Inventory (s, S) policy

3.4 Assumptions

In this thesis, we made the following assumptions:

- The demand is deterministic and static.
- Waste management cost is explicitly considered.
- The capacity for inventory is limited.
- We assume a central decision maker for deciding the transportation allocation and inventory at the retailer's decision for the whole supply chain.
- We assume that there is no central warehouse between abattoirs and retailers. The common practice is to rent a shelf or butcher section at retailers, direct sale point to customers with semi annual or yearly contract.

- We assume no competitive between different supply chain entities. Such assumption is realistic since usually the FMCG is highly cooperative environment where all supply chain players share their logistic planning information. The meat market shares for any company for specific period from livestock is typically known and each company should have margin of it.
- We assume no flow between commodities in the same level of supply chain for example no exchange of commodities between farms or products between retailers.

A summary of the features of the considered meat supply chain problem is given in Table **??**. The information of the demand rates stems from retailers and flows to the upstream activities, i.e., abattoirs to sources of livestock. The solutions of model 2 will be fed into model 1 with information of meat and its products to be delivered to farms and/or retailers within the time windows requirements.

Table 3.1: A summary of the considered meat supply chain problem

Main features	**Routing and scheduling**	**Three echelon meat networks**
-	**Model 1**	**Model 2**
Tactical decisions		
Allocation from farms to abattoirs		X
Allocation from abattoirs to retailors		X
Order decisions		X
Operation decisions		
Inventory replenishment		X
Itinerary planning	X	
Livestock procurement		X
Holding costs		X
Objective		
Maximizing sale		X
Shipping cost	X	X
Ordering cost/fixed cost	X	X
Meat deterioration cost		X
Lost sale		X
Capacity of MSC activities		X
Capacity of Vehicles (vessels, air)	X	X
Routing and scheduling decisions	X	X
Production planning		X
Upstream of Information flow		X
Upstream of Material flow	X	
Multi-mode	X	X

3.5 Mathematical model

The two mathematical models are presented in this section, decision variables and objective function are described, and all constraints are being explained in details.

3.5.1 Model 1

Sets:

N -Set of vertices represent sources or ports (airport or seaport); indexed by $n \in N$.

O -Set of source vertices represent supply of meat products; indexed by $o \in O$ where $O \subset N$.

P -Set of port vertices; indexed by $p \in P$ where $P \subset N$.

C -Set of commodities, live animal, chilled meat , frozen Meat; indexed by $c \in C$.

S -Set of transportation schedule services either via land, sea, or air; indexed by $s \in S$.

S_n^+ -Set of transportation schedule services emanating from source n.

S_n^- -Set of transportation schedule services terminating at port n.

D -Set of demands; indexed by $d \in D$.

Parameters:

O_d - Origin of demand D (sources), where $O_d \in O$

d_d - Destination of demand D (ports), where $d_d \in P$

O_s - Origin of service s (sources), where $O_s \in O$

d_s - Destination of service s (ports), where $d_s \in P$

C_{sc} - Capacity of service s per meat product c in kilogram

W_c - Average weight of commodity c in kilogram

τ_s^{CT} - Cut-off time for service s from O_s

τ_s^{AR} - Schedule arrival for service s at d_s

τ_d^{AL} - Ready time of demand D

τ_{dc}^{DU} - Due date for demand D of meat product c

F_s - Fixed booking cost to reserve space on service s

d_{dc} - Number of meat product c for demand D

T_{cs} - Cost to ship meat product c on service s in \$ per kilogram

$$\lambda_{cs} = \begin{cases} 1\, if\; commodity\; \mathrm{c}\; can\; be\; ship\; via\; service\; \mathrm{s}, \\ 0\; otherwise \end{cases}$$

Decision variables :

$a_{d,c,n}^{AR}$ - Arrival time of demand D for commodity c at node n.

$$X_{dcs} = \begin{cases} 1\, if\; demand\; \mathrm{D}\; of\; commodity\; \mathrm{c}\; ship\; via\; service\; \mathrm{s}, \\ 0\; otherwise \end{cases}$$

$$Y_s = \begin{cases} 1\, if\; service\; \mathrm{s}\; is\; booked, \\ 0\; otherwise \end{cases}$$

Mathematical Model 1

The mathematical model 1 is presented in this section, showing the objective function and all the constrains:

$$(P_1) \qquad Min\ Z_1 = \overbrace{\sum_{s \in S} F_s \times Y_s}^{\text{Fixed service}} + \overbrace{\sum_{d \in D} \sum_{c \in C} \sum_{s \in S} T_{cs} \times X_{dcs} \times d_{dc}}^{\text{Shipping costs}} \tag{1}$$

$s.t.$

$$\sum_{s \in S_n^+} \lambda_{cs} \times d_{dc} \times X_{dcs} - \sum_{s \in S_n^-} \lambda_{cs} \times d_{dc} \times X_{dcs} = \begin{cases} d_{dc} & \forall d, c, n, : n \in O_d \\ -d_{dc} & \forall d, c, n, : n \in d_d \\ 0 & \forall d, c, n, : n \notin O_d \ or \ d_d \end{cases} \tag{2}$$

$$\sum_{d \in D} \sum_{c \in C} W_c \times d_{dc} \times X_{dcs} \leq C_{sc} \quad \forall s \in S \tag{3}$$

$$\sum_{s \in S} \lambda_{cs} \times \tau_s^{CT} \times X_{dcs} \leq \tau_D^{AL} \quad \forall d, c : d \in D, c \in C \tag{4}$$

$$\sum_{s \in S} \lambda_{cs} \times \tau_s^{AR} \times X_{dcs} \leq \tau_D^{DU} \quad \forall d, c : d \in D, c \in C \tag{5}$$

$$a_{dcn}^{AR} - \tau_s^{CT} + M(X_{dcs} - 1) \leq 0 \quad \forall d, c, s, n : n \in O_s \tag{6}$$

$$a_{dcn}^{AR} - \tau_s^{AR} + M(X_{dcs} - 1) \leq 0 \quad \forall d, c, s, n : n \in d_s \tag{7}$$

$$X_{dcs}; Y_s \in 0, 1 \quad \forall d, c, s : d \in D, c \in C, s \in S \tag{8}$$

$$a_{dcn} \geq 0 \quad \forall d \in D, c \in C, n \in N \tag{9}$$

The objective function, as in Z_1 is to minimize the total costs including fixed cost of booking services and transportation cost of shipping the commodities. Constraints in Eq (2) guarantee the commodities are shipped to their destination ports/farms based on the solutions of model 2. Constraints in Eq (3) are capacities constraints of the services. Constraints in Eq (4-7) ensure the commodities to be ready before cut-off time of the services and also guarantee that the commodities is delivered before the due date based on the result obtained from model 2. Finally, constraints in Eq (8-9) are binary requirements and continuous variable definitions.

3.5.2 Model 2

Sets:

I - Set of farms; indexed by $i \in I$

J - Set of abattoirs; indexed by $j \in J$

K - Set of retailers; indexed by $k \in K$

T - Set of time periods; indexed by t, r, and $e \in T$

C - Set of commodities; indexed by $c \in C$

Parameters:

F_{ij} -Fixed allocation cost of farm i to abattoir j

G_{jk} -Fixed allocation cost of retailer k to abattoir j

TL_{ij} -Unit truck transportation cost from farm i to abattoir j

TC_{jk} -Unit truck transportation cost from abattoir j to retailer k

UC_{ci} -Unit feeding cost per commodity c at farm i

HL_j -Unit handling cost for abattoir j at each time period

H_{ck} -Unit holding cost of commodity c for retailer k at each time period

L -Unit lost-sale cost for unmet demands

P_{crt} -Price of commodity c in period t, for commodity produced in period r

E -Unit waste cost of commodities

B_{crt} -Fraction of commodity c, produced at time period r that is wasted at the end of period t

M -A very big number

s -A very small number

FO -Fixed ordering cost

LT -Lead time between order placement by retailers and the time order is received

Y_c -Expected commodity c extraction ratio from a slaughtered livestock in abattoir

d_{ckt} -Demand of commodity c in retailer k at time period t

C_{it} -Maximum throughput of farm i at time period t

Q_{jt} -Maximum throughput capacity of abattoir j at time period t

Decision variables :

SL_{ck} - Reorder point for commodity c in the retailer k

SU_{ck} -Order-up-to level for commodity c in the retailer k

dl_{ckrt} -Amount of commodity c delivered to retailer k at time period t, which is produced in time period r

W_{ijct} -Number of commodities c transported to abattoir j from farm i at time period t

X_{cjkt} -Amount of commodity c transported to retailer k from abattoir j at time period t

Ib_{ckrt} -Retailer k's on-hand inventory of commodity c at the beginning of time period t, produced in time period r

Ie_{ckrt} -Retailer k's on-hand inventory of commodity c at the end of time period t, produced in time period r

lk_{ckt} -Lost-sale quantity of commodity c for demand in time period t and retailer k.

$$O_{ij} = \begin{cases} 1\, if\ abattoir\ \mathrm{j}\ is\ allocated\ to\ farm\ \mathrm{i}, \\ 0\ \ otherwise \end{cases}$$

$$Z_{jk} = \begin{cases} 1\, if\ retailer\ \mathrm{k}\ is\ allocated\ to\ abattoir\ \mathrm{j}, \\ 0\ \ otherwise \end{cases}$$

$$Y_{ckt} = \begin{cases} 1\, if\ retailer\ \mathrm{k}\ makes\ an\ order\ for\ commodity\ \mathrm{c}\ at\ the\ beginning\ of\ time\ period\ \mathrm{t}, \\ 0\ \ otherwise \end{cases}$$

$$u_{ckt} = \begin{cases} 1\, when\ the\ onhand\ inventory\ in\ retailer\ \mathrm{k}'s\ site\ for\ \mathrm{c}\ is\ above\ zero\ at\ end\ ofperiod\ \mathrm{t}, \\ 0\ \ otherwise \end{cases}$$

In this model, four binary variables are used O_{ij} , Z_{jk} , Y_{ckt} and u_{ckt}, The first two binary variables are designing the network of the meat supply chain (MSC) by allocating the farm i to abattoir j and from abattoir j to retailer k in the three-echelon network.

In the real-life business of meat and livestock, any disruption may affect the full supply chain such as change in health protocols between cities, diseases in animals in farms or closure of abattoirs at certain times. For those reasons, the retailers typically use multiple approved abattoirs and deal with different farmers to avoid any shortage and to satisfy the demand of the consumer. These two logical binary variables Y_{ckt} and u_{ckt} are introduced to reflect the commodity c orders made by retailer k and the on-hand inventory in retailer k at the end of the period t. The continuous variables Ib_{ckrt} and Ie_{ckrt} are introduced to measure the inventory at the beginning and the ending inventory at each period. SL_{ck} and SU_{ck} variables help to maintain inventory stock level of commodities with respect to demand for each retailer. Three decision variables dl_{ckrt} , W_{ijrt} ,and X_{cjkt} are introduced to determined amount and number of commodity c transported through the three-echelon network to clients in retailer k at each time period t , and the last decision lk_{ckt} reflect the lost in sale for each commodity c at retailer k in period time t respectively. The details of the model is given below.

Mathematical Model 2

The mathematical model 2 is presented in this section, showing the objective function and all the constrains:

$$
\begin{aligned}
(P_2) \qquad & Max\ Z_2 = \overbrace{(\sum_{c\in C}\sum_{K\in K}\sum_{r\in T:r\leq t}\sum_{t\in T} P_{crt} \times dl_{ckrt})}^{\text{Total income}} \\
& -\overbrace{(\sum_{i\in I}\sum_{j\in J} F_{ij} \times O_{ij} + \sum_{j\in J}\sum_{K\in K} G_{jk} \times Z_{jk})}^{\text{Allocation costs}} \\
& -\overbrace{(\sum_{i\in I}\sum_{j\in J}\sum_{c\in C}\sum_{r\in T} UC_{ci} \times W_{ijcr})}^{\text{Feeding costs}} - \overbrace{(\sum_{i\in I}\sum_{j\in J}\sum_{c\in C}\sum_{r\in T} TL_{ij} \times W_{ijcr})}^{\text{Transportation costs}} \\
& -\overbrace{(\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{r\in T} TC_{jk} \times X_{cjkr})}^{\text{Transportation costs}} - \overbrace{(\sum_{c\in C}\sum_{j\in J}\sum_{k\in K}\sum_{r\in T} HL_{j} \times X_{cjkr})}^{\text{Handling costs}} \\
& -\overbrace{(\sum_{c\in C}\sum_{K\in K}\sum_{r\in T:r\leq t}\sum_{t\in T} H_{ck} \times (1 - B_{crt}) \times Ie_{c,k,r,t})}^{\text{Holding costs}} \\
& -\overbrace{(\sum_{c\in C}\sum_{K\in K}\sum_{r\in T:r\leq t}\sum_{t\in T} E \times B_{crt} \times Ie_{ckrt})}^{\text{Waste costs}} \\
& -\overbrace{(\sum_{c\in C}\sum_{K\in K}\sum_{t\in T} L \times lk_{ckt})}^{\text{Lost-sale costs}} - \overbrace{(\sum_{c\in C}\sum_{K\in K}\sum_{t\in T} FO \times Y_{ckt})}^{\text{Fixed ordering costs}} \qquad (10)
\end{aligned}
$$

$s.t.$

$$\sum_{i\in I} O_{ij} = 1 \quad \forall\ j \in J \qquad (11)$$

$$\sum_{j\in J} Z_{jk} = 1 \quad \forall\ k \in K \qquad (12)$$

$$Ib_{ckrt} = 0 \quad \forall c \in C, k \in K, r\&t \in T \mid r = t \qquad (13)$$

$$Ie_{ckrt} = \sum_{j\in J} X_{cjkr} - dl_{ckrt} \quad \forall\ c \in C, k \in K, r\&t \in T \mid r = t \qquad (14)$$

$$Ie_{ckrt} = \sum_{j\in J} X_{cjkr} - \sum_{e=r\in t} dl_{ckre} - \sum_{e=r\in t-1} B_{cre} \times Ie_{ckre} \forall c \in C, k \in K, r\&t \in T \mid r < t \qquad (15)$$

$$Ib_{ckrt} = \sum_{j\in J} X_{cjkr} - \sum_{e=r\in t-1} dl_{ckre} - \sum_{e=r\in t-1} B_{cre} \times Ie_{ckre} \forall c \in C, k \in K, r\&t \in T \mid r < t \qquad (16)$$

$$\sum_{r\in T|r\leq t} dl_{ckrt} + lk_{ckt} = d_{ckt} \forall c \in C, k \in K, t \in T \tag{17}$$

$$\sum_{r\in T|r\leq t} Ib_{ckrt} + \sum_{j\in J} \sum_{e=t\in t+LT-1} X_{cjke} + M \times Y_{ckt} \geq SL_{ck} - s \forall c \in C, k \in K, t \in T \tag{18}$$

$$\sum_{r\in T|r\leq t} Ib_{ckrt} + \sum_{j\in J} \sum_{e=t\in t+LT-1} X_{cjke} - M \times (1 - Y_{ckt}) \leq SL_{ck} - s \forall c \in C, k \in K, t \in T \tag{19}$$

$$X_{c,j,k,(t+LT)} \leq M \times Y_{ckt} \; \forall c \in C, j \in J, k \in K, t \in T \tag{20}$$

$$\sum_{j\in J} X_{cjk(t+LT)} + M(1 - Y_{ckt}) \geq SU_{ck} - (\sum_{r\in T|r\leq t} Ib_{ckrt} + \sum_{j\in J} \sum_{e=t\in t+LT-1} X_{cjke}) \forall c, k, t \tag{21}$$

$$\sum_{j\in J} X_{cjk(t+LT)} \leq SU_{ck} - (\sum_{r\in T|r\leq t} Ib_{ckrt} + \sum_{j\in J} \sum_{e=t\in t+LT-1} X_{cjke}) \forall c \in C, k \in K, t \in T \tag{22}$$

$$\sum_{c\in C} \sum_{k\in K} X_{cjkt} \leq Q_{jt} \forall j \in J, t \in T \tag{23}$$

$$X_{cjkt} \leq M \times Z_{jk} \forall c \in C, j \in J, K \in k, t \in T \tag{24}$$

$$\sum_{c\in C} W_{ijct} \leq M \times O_{ij} \forall i \in I, j \in J, t \in T \tag{25}$$

$$s \times u_{ckt} \leq \sum_{r\leq T} Ie_{ckrt} \leq M \times u_{ckt} \forall i \in I, j \in J, t \in T \tag{26}$$

$$lk_{ckt} \leq M \times (1 - u_{ckt}) \forall c \in C, K \in k, t \in T \tag{27}$$

$$\sum_{i\in I} W_{ijct} \leq \sum_{k\in K} Y_c \times X_{cjkt} \forall c \in C, j \in J, t \in T \tag{28}$$

$$\sum_{j\in J} \sum_{c\in C} W_{ijct} \leq C_{it} \forall i \in I, t \in T \tag{29}$$

$$O_{ij}; Z_{jk}; Y_{ckt}; u_{ckt}; \in 0, 1 \quad \forall i \in I, j \in J, c \in C, K \in k, t \in T \tag{30}$$

$$SL_{ck}; SU_{ck}; dl_{ckrt}; W_{ijct}; X_{cjkt}; Ib_{ckrt}; Ie_{ckrt}; lk_{ckt} \geq 0 \forall i \in I, j \in J, c \in C, k \in K, r, e \& t \in T \tag{31}$$

Objective function Z_2 maximizes the total profit as the total revenue minus the total cost. The objective function includes the total income, allocation cost, feeding cost, transportation cost, handling cost, holding cost, waste cost, lost-sale cost and finally the fixed ordering cost. Constraints in Eq (11) and Eq (12) are used to determine the allocation decisions for the upstream supply chain processes. Constraints in Eq (13), Eq (14), Eq(15) and Eq (16) reflect the inventory balance for each time period. First constraints in Eq (13) enforce the on-hand inventory at the beginning of time period t of the commodity which is produced in period r to equal zero. Eq (14) shows the on-hand inventory at the end of the first period when the meat is produced at $r = t$ is equal to the total amount of commodities received from all abattoirs minus the total amount of meat sold to the customers. Third inventory constraints for Eq (15) guarantee that the amount of the ending inventory $(r < t)$ is equivalent to all the amounts delivered to the abattoirs minus the demand requested by time t minus the amount of waste from the ending product up to $t - 1$. Similarly, Eq (16) ensures that the amount of the commodity at the beginning of inventory $(r < t)$ is equivalent to all the amounts delivered from the abattoirs minus the demand requested up to time $t - 1$ minus the amount of waste from the ending product up to $t - 1$. Constraints in Eq (17) are the demand satisfaction constraints. The constraints in Eq (18) and Eq (19) guarantee that an order is being made when the inventory level reaches the reorder point, which means that $Y_{c,k,t} = 1$ otherwise no order should be placed. Constraint Eq (20) makes sure the order to delivered at time $(t+LT)$ which has been ordered at time t. Constraints in Eq (21) and Eq (22) guarantee the amount of the product to be delivered at each retailer to be less than the order-up-to level $SU_{c,k}$. Constraints in Eq (23) reflect the capacity of the abattoirs. Constraints in Eq (24) and Eq (25) are the relationship between the flow and the allocation variables. Constraints shown in Eq (26) and Eq (27) ensure that the customer is being served when there is on-hand inventory. Constraints in Eq (28) make sure that the outflow of product at abattoirs is equal to the inflow of the commodity delivered from the farm. Constraints in Eq (29) guarantee the amount of product delivered from the farms are less than its capacity. Finally, constraints in Eq (30) and Eq (31) are binary requirements and continuous variable definitions.

In solving the meat supply chain planning problems, we first solved the mathematical model 2 and then the solution of the second model will be input for the first model. The connection between the two models are giving below:-

$$\sum_{j \in J} W_{ijct} = d_{cti} \forall c \in C, , t \in T, i \in T \tag{32}$$

3.6 Heuristic Solution Approach

In this section, we introduce a heuristic solution approach to solve the comprehensive meat supply chain optimization model developed in the previous subsection. The supply chain optimization model studied was decomposed into two independent optimization problems. The former optimization model aims at finding (a) the allocation decisions between farms to abattoirs and from abattoirs to retailers, (b) the quantity of commodities to be shipped between farms to abattoirs and from abattoirs to retailers and (c) inventory decisions at retailers including the reorder level, order up to level for each meat product. The latter optimization model aims at finding the itinerary for each commodity from source origin to the ports (port, airport, and seaport) by trucks, airplanes, and vessels, given the allocation and inventory decisions of the second model. Solution of such problem can be found in less computational time if the two problems are solved together as the size of the decompose problems has fewer number of decisions variables. Details of the proposed solution method are given below in the Figure 3.4.

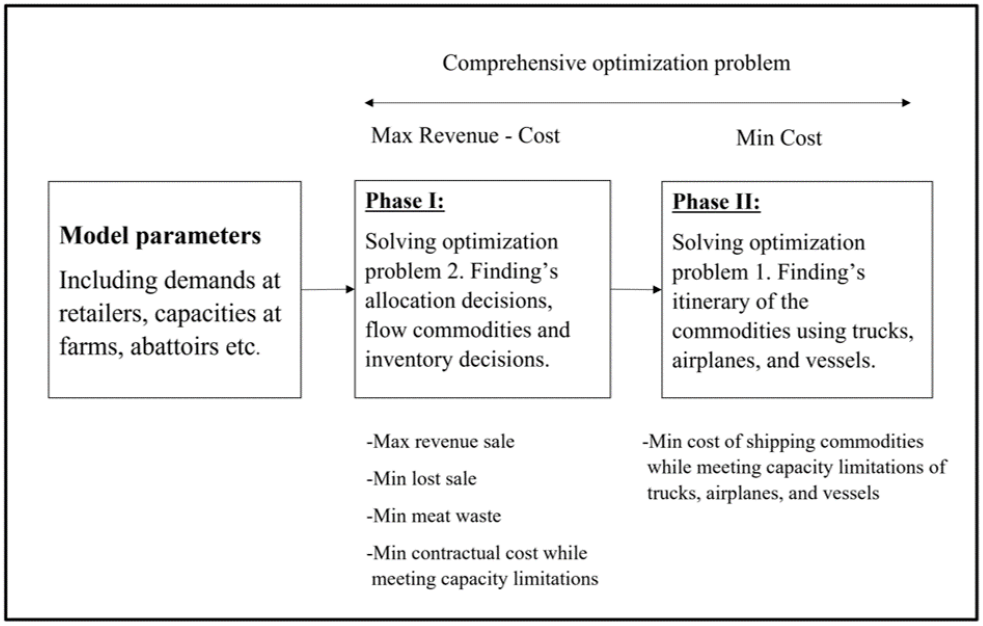
Comprehensive optimization problem
Max Revenue - Cost
Min Cost
Model parameters
Including demands at retailers, capacities at farms, abattoirs etc.
Phase I:
Solving optimization problem 2. Finding's allocation decisions, flow commodities and inventory decisions.
Phase II:
Solving optimization problem 1. Finding's itinerary of the commodities using trucks, airplanes, and vessels.
-Max revenue sale
-Min lost sale
-Min meat waste
-Min contractual cost while meeting capacity limitations
-Min cost of shipping commodities while meeting capacity limitations of trucks, airplanes, and vessels

Figure 3.4: Flow chart of heuristic solution approach

3.7 Summary

In this chapter, a mixed integer programming models are proposed to solve the considered meat supply chain problem. The supply chain optimization model studied was decomposed into two independent optimization problems. The heuristic solution approach can solve the problem with less computational time. In the next chapter, we will present a real case study and several numerical example problems to demonstrate the two developed models and solution method. Sensitivity analysis has been conducted on some parameters for each model.

Chapter 4

Computational Results

4.1 Introduction

In this section a real case study of meat supply chain located in United Arab Emirates is investigated using the proposed mathematical models developed for meat supply chain as depicted in Figure 4.1. In the first Phase the solution consists of allocation of livestock from farms to abattoirs and the quantity to be shipped to retailers, while in the second Phase we used the results of the first Phase to generate the routing and scheduling decision from source of origin of the meat to the required destinations. Generally, the computational complexity of a model depends on the size of the problem. Most of the instances are solved in few minutes and able to find optimum solution. Although we solved real case study in United Arab Emirates, the proposed method can be adopted to solve meat supply chain problems in North America such as Canada or United States of America.

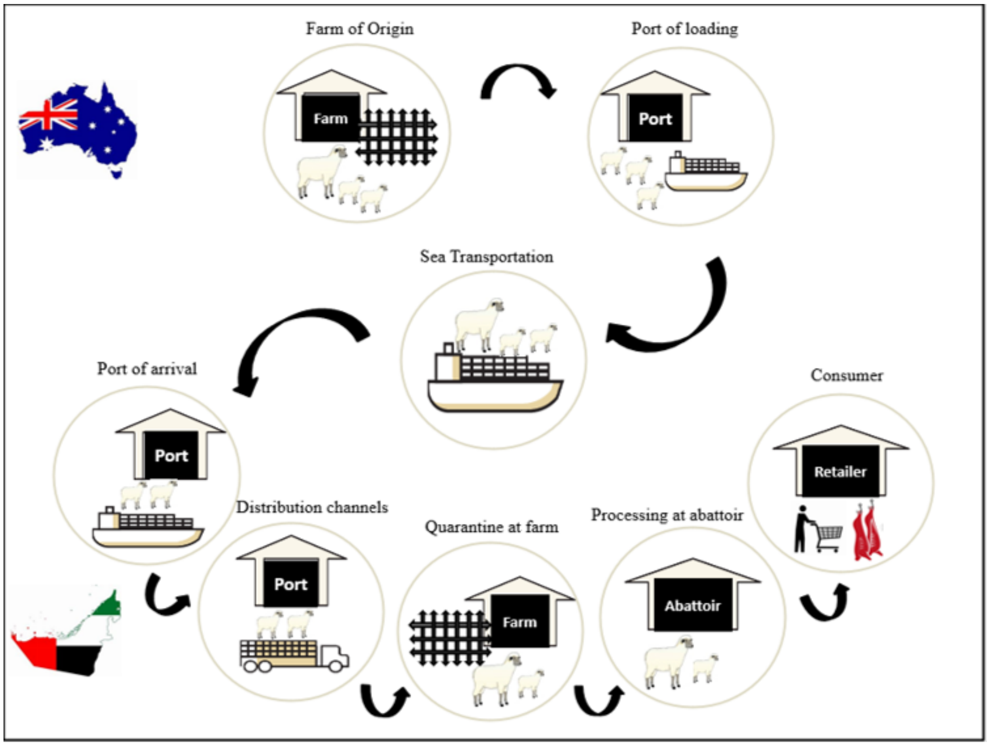

Figure 4.1: Planning horizon of the problem

4.2 Case Study

As shown in Figure 4.2 and in Figure 4.3 , we considered for Phase 1 only two farms, three abattoirs and seven retailers in United Arab Emirates. The time horizon is 7 time periods (7 days).The mode of transportation for Phase 1 is land transportation using set of vehicles. Two types of the vehicles are considered in our model. Non-refrigerator vehicles are used to transport live sheep from farm to abattoir, while refrigerator vehicles are being used to transport processed meat from abattoir to retailer to maintain the quality and have the maximum shelf life. The temperature of the refrigerator vehicles is setup between 0-5 °C for chilled/fresh meat and up to -18 °C for frozen meat.

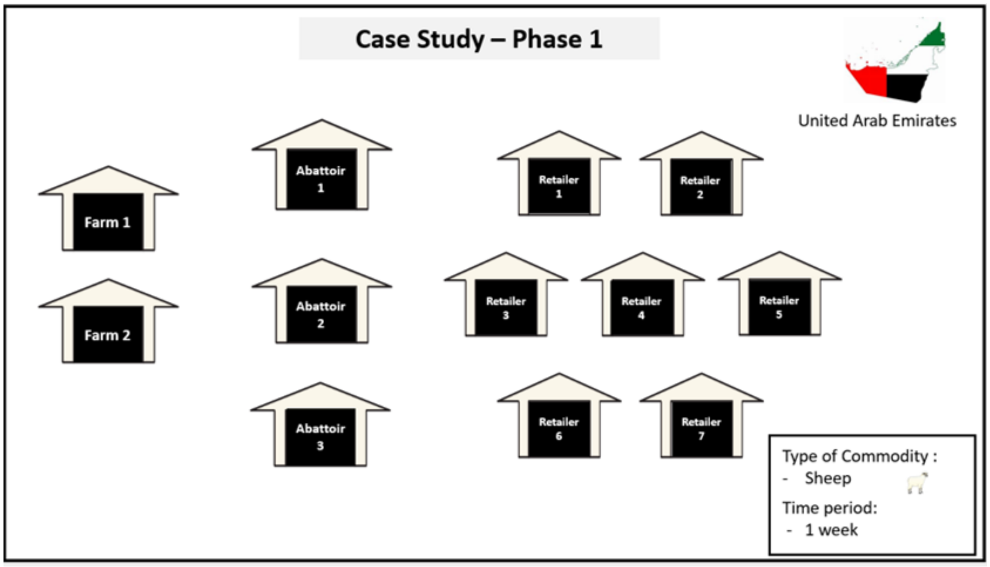

Figure 4.2: Instance of the case study – Phase 1

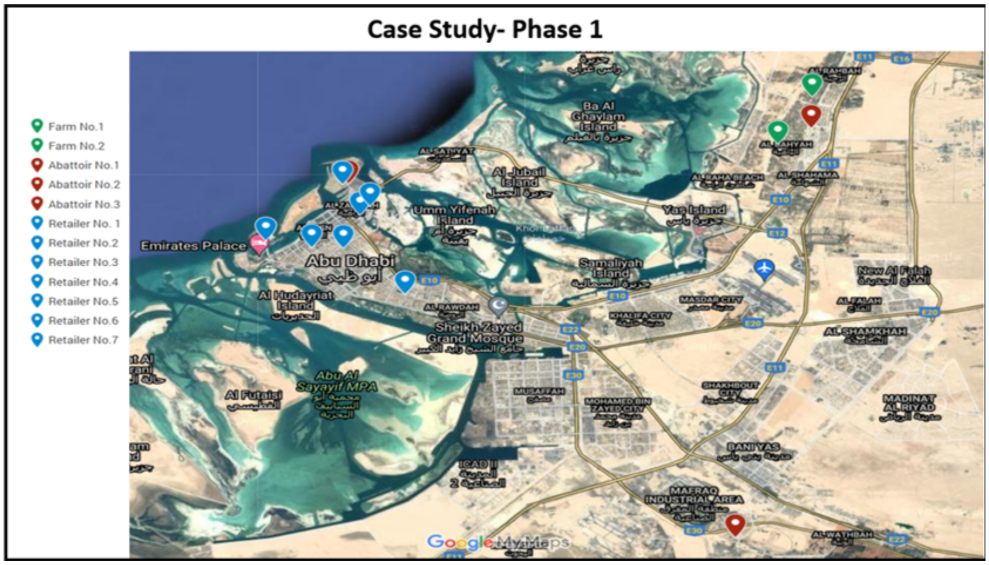

Figure 4.3: Coordination of farms, abattoirs and retailers for case study – Phase 1

For the second Phase we considered four demand regions in Australia, United Arab Emirates, Kuwait, and Qatar. Australia is the main source of live, chilled or frozen meat, to supply the three other countries with six types of commodities: live sheep, chilled lamb or mutton, frozen lamb or mutton, live cattle, chilled beef, and frozen beef. We designed nine schedule services: three schedule vessels services for sea transportation and six shipments via airplane to satisfy total of five orders as illustrated in Figure 4.4 and Figure 4.5. The plan period of this phase is 23 days considering transportation and quarantine period of live animal before transporting them to abattoirs after shipping via sea service.

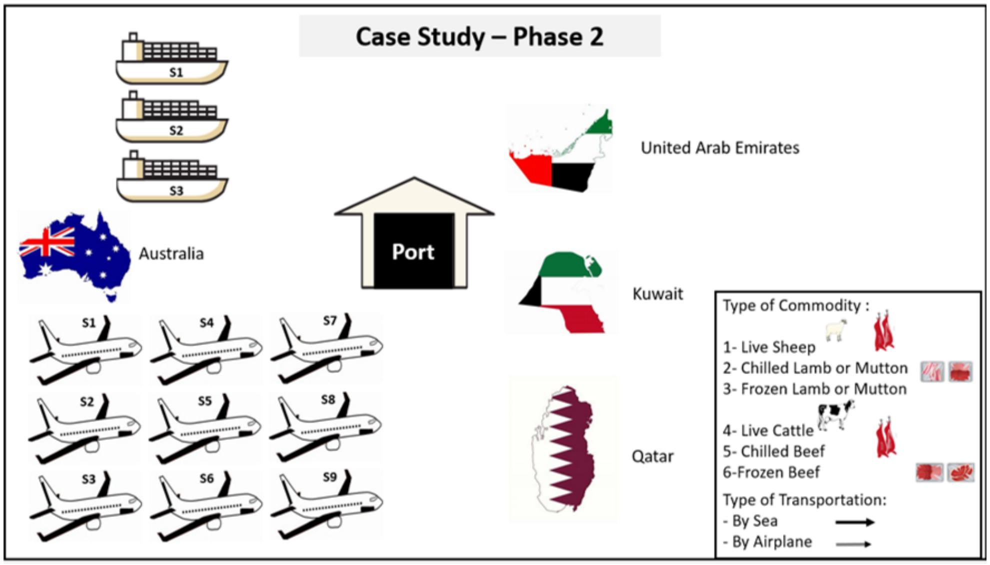

Figure 4.4: Instance of the case study – Phase 2

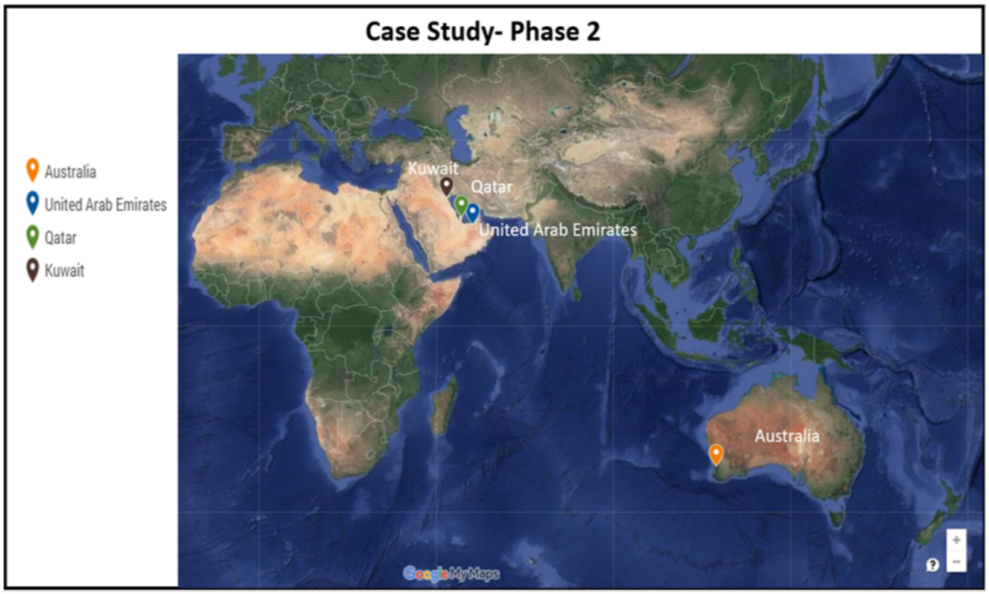

Figure 4.5: Coordination of countries for case study – Phase 2

To implement the second Phase, we need the total amount of commodity delivered to retailer at time period t as input for the first model to know how much quantity to ship from country of origin and when to ship to satisfy the total demand needed.

4.3 Experimental results

In this section, the result of Phase 1 and Phase 2 are presented for the case study. A sensitivity analysis is presented with two different scenarios for some of the parameters to compare the objective function of the two models results.

For phase 1

- The time horizon for the Phase 1 model is 7 days
- The objective function of the Phase 1 model is to maximizes the total profit as the total revenue minus the total cost
- The competition time for the results of most of instants are few seconds to several seconds depend on the data.

The result of the case study for retailer 1 is presented in the Table 4.1. Each time is

Table 4.1: The result of retailer 1

Retailer 1							
t-Days	1	2	3	4	5	6	7
Daily sale/demand	120	100	80	100	80	120	80
Stock at day 0	0						
Amount delivered	120	160	120	0	180	100	0
Inventory at beginning	0	0	60	100	0	100	80
Inventory at end	0	60	100	0	101	80	0
Reorder point	180	180	180	180	180	180	180
Order up to level	280	280	280	280	280	280	280

considered as one day and the lead-time between retailers' ordering time and the meat's delivery time is assumed equal to one time slot. For presentation purposes, one retailer is chosen to discuss the result. Figures 4.6-4.12 present the obtained results for retailers 1-7. In Fig. 4.6 with Table 4.1 , the inventory on hand, amount of delivered, the reorder point and order up to level are shown. A total of 680 live sheep have been delivered to each retailer. In this case, the reorder and order-up-to-level points are 180 and 280 units, respectively, and five orders are placed one day before delivery at periods 0,1,2,4 and 5, respectively.

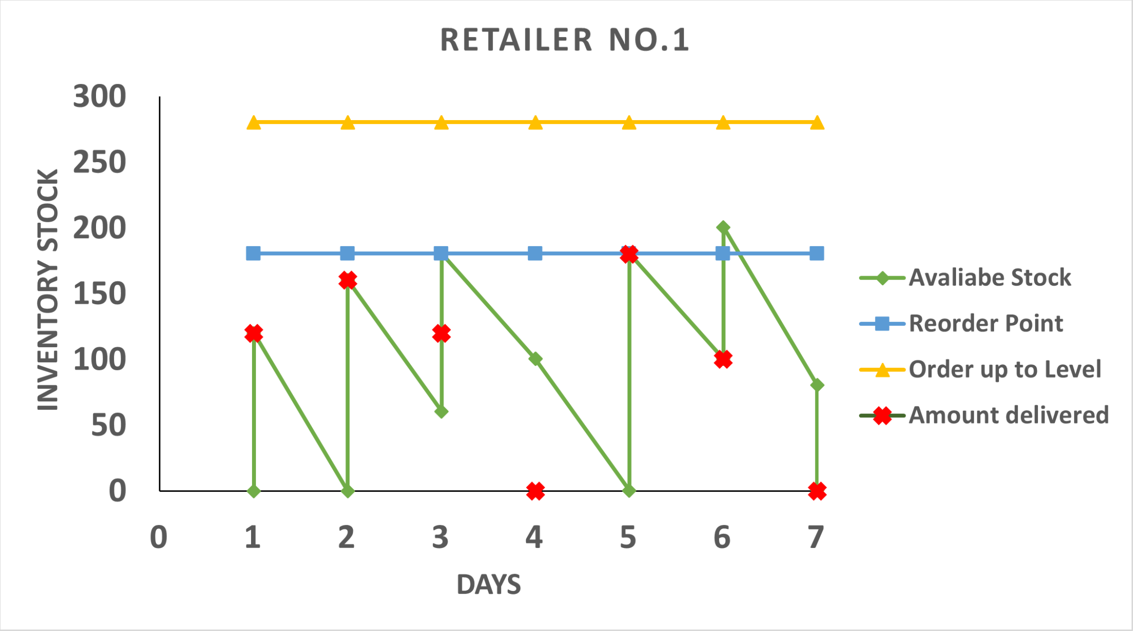

Figure 4.6: Result of retailer 1

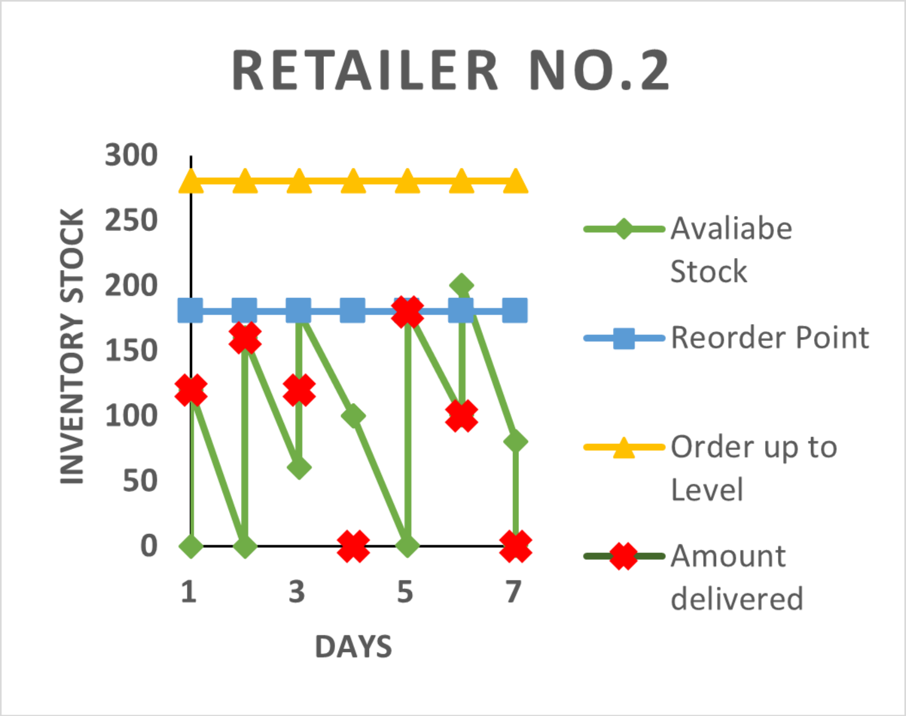

Figure 4.7: Result of retailer 2

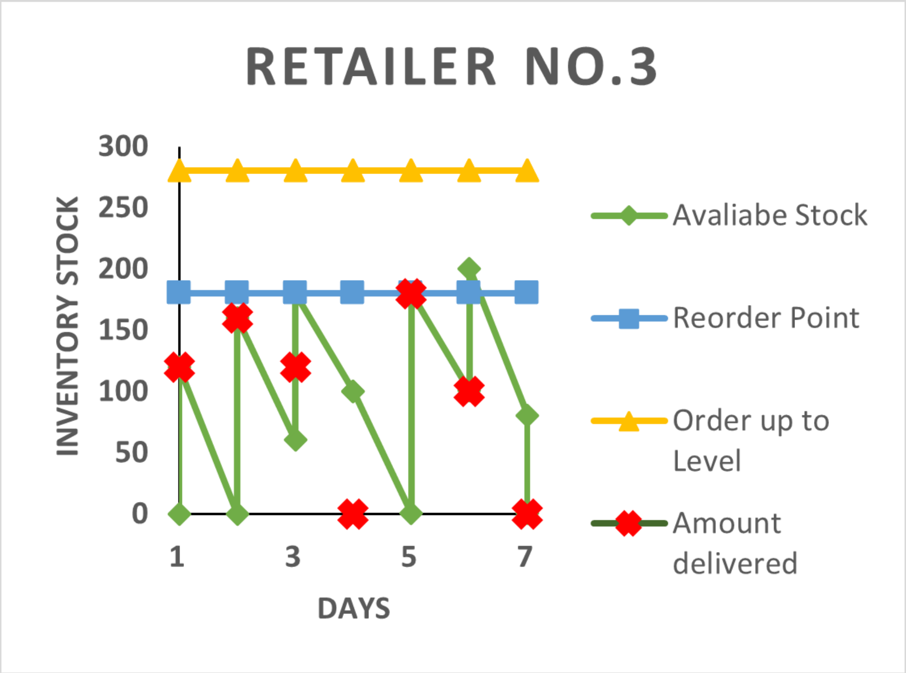

Figure 4.8: Result of retailer 3

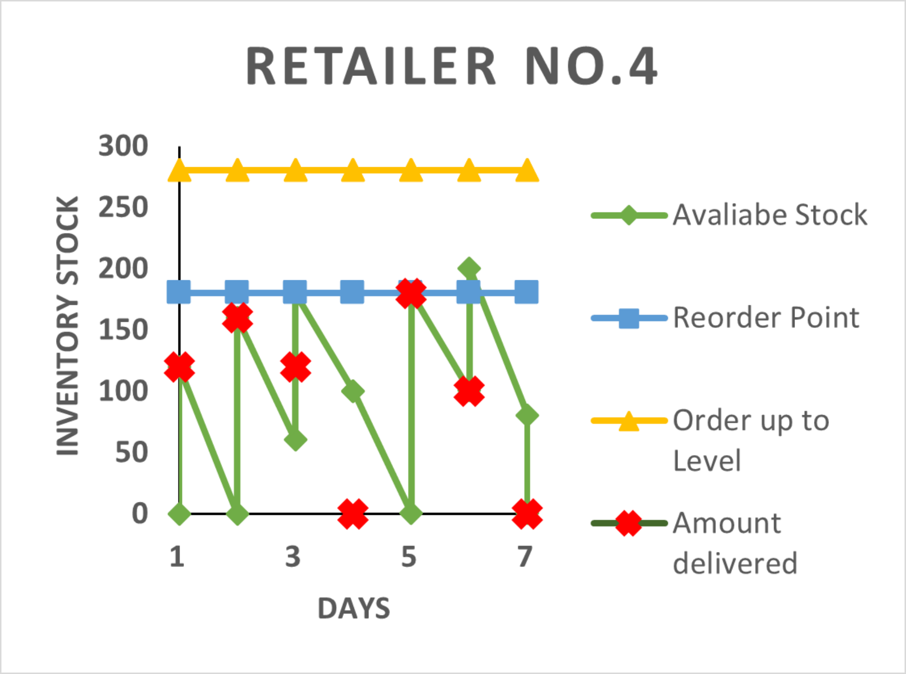

Figure 4.9: Result of retailer 4

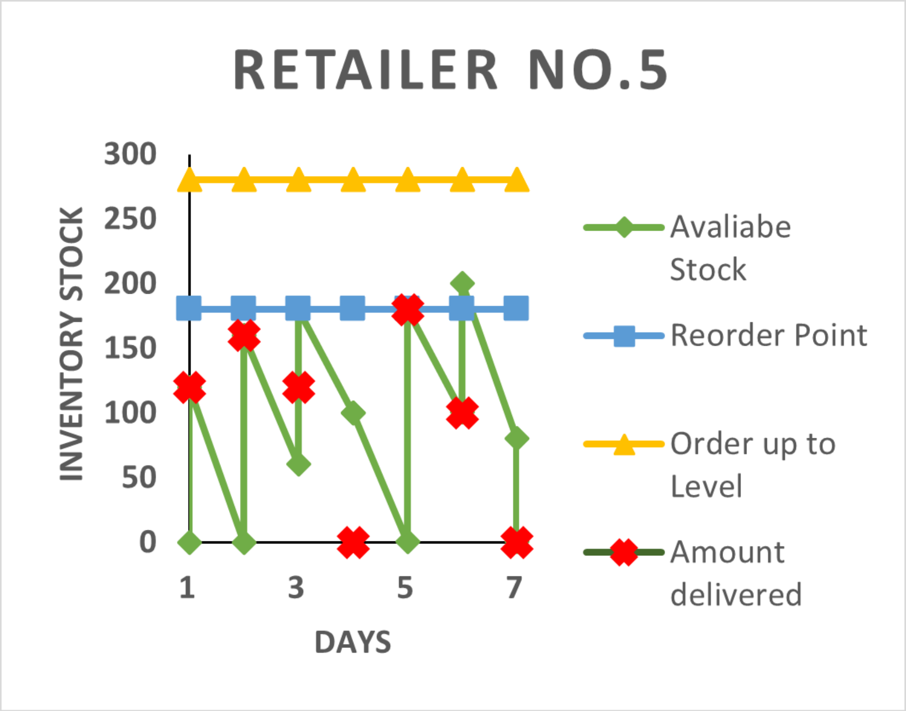

Figure 4.10: Result of retailer 5

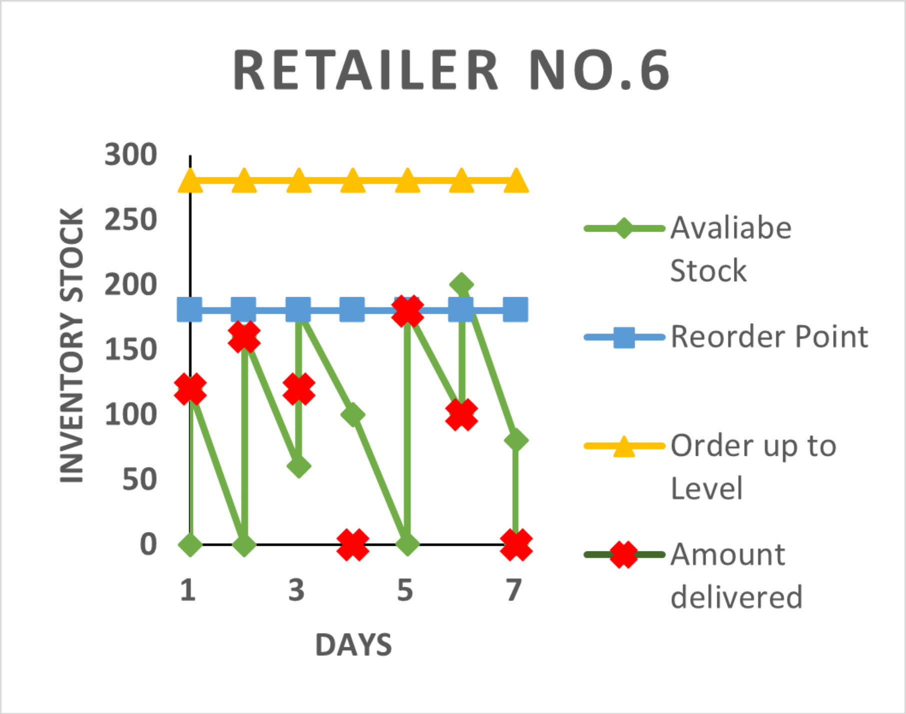

Figure 4.11: Result of retailer 6

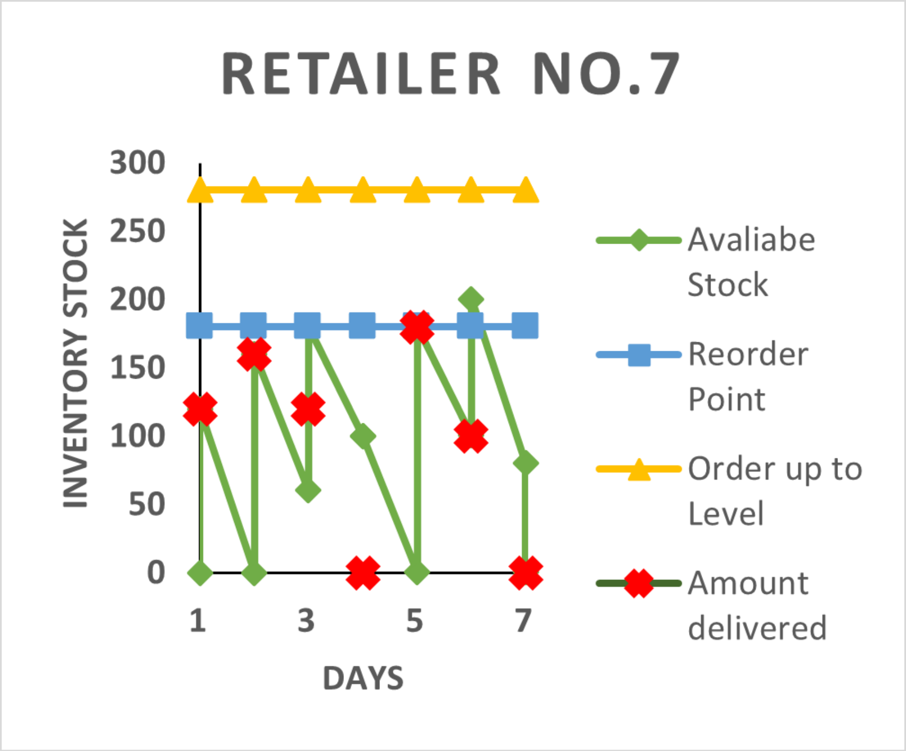

Figure 4.12: Result of retailer 7

The result of allocation decisions is illustrated in Figure 4.13. We imposed that each downstream process deals with only one upstream process to control the quality of meat process as inspired by the current practice of meat supply chain.

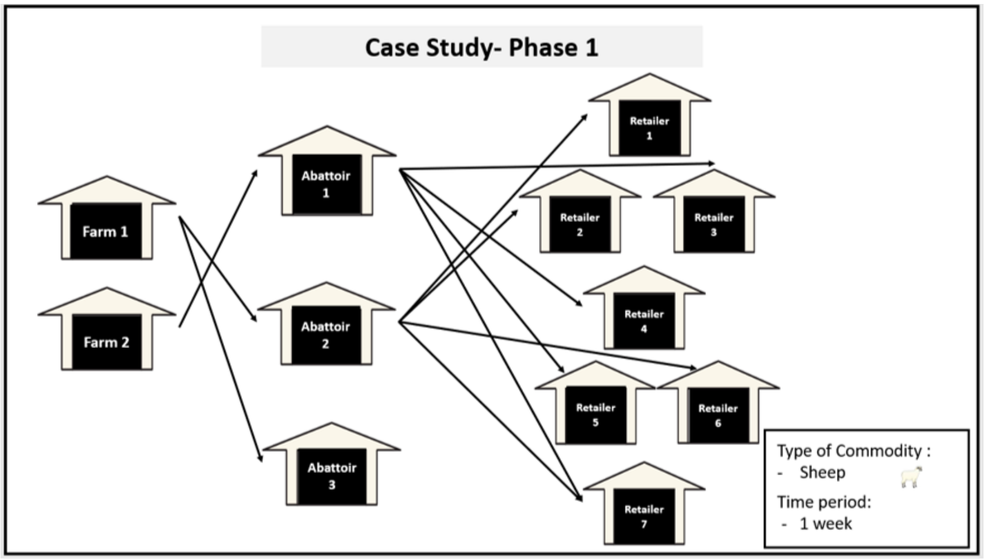

Figure 4.13: Result of the allocation decision of case study – Phase 1

In this part we will present the results of the sensitivity analysis for the phase 1 model by changing one parameter at a time: (1) unit handling cost for abattoir; (2) lost-sale cost; (3) price of commodity; (4) fixed ordering Cost; (5) lead Time; (6) demand of commodity; (7) maximum throughput of farm; and (8) - maximum throughput capacity of abattoir. Table 4.2 with Figures 4.14 - 4.17 summarizes the result of the sensitivity analysis. As shown below, the most significant parameters are demand of commodity, price of commodity, fixed ordering cost, and maximum throughput of farm. The computational time increase significantly when the capacity of farms increases by 20%. An examination of all the instances suggests that, as high-level throughput capacity of farms increases the total profit increases by 7% as the meat supply chain able to process more meats and its products to satisfy more demands. It means that the model tends to minimize the lost sale cost and increase demand satisfaction as the capacity of the farms increase.

Table 4.2: Sensitivity analysis of Phase 1

Sensitivity Analysis for Phase 1						
Parameters	**Objective Function**			**Time in Second**		
	Base	**Scenario No.1**	**Scenario No.2**	**Base**	**Scenario No.1**	**Scenario No.2**
(1) HL_j	391742	389928	393102	3707	8017	2902
(2) L	391742	390720	392201	3707	3570	3364
(3) P_{crt}	391742	325448	81982	3707	1800	783
(4) FO	391742	356311	455123	3707	547	804
(5) LT	391742	369791	369791	3707	2252	2259
(6) d_{ckt}	391742	469282	311775	3707	4351	400
(7) C_{it}	391742	422421	216732	3707	14	88774
(8) Q_{jt}	391742	391742	390764	3707	3963	5915

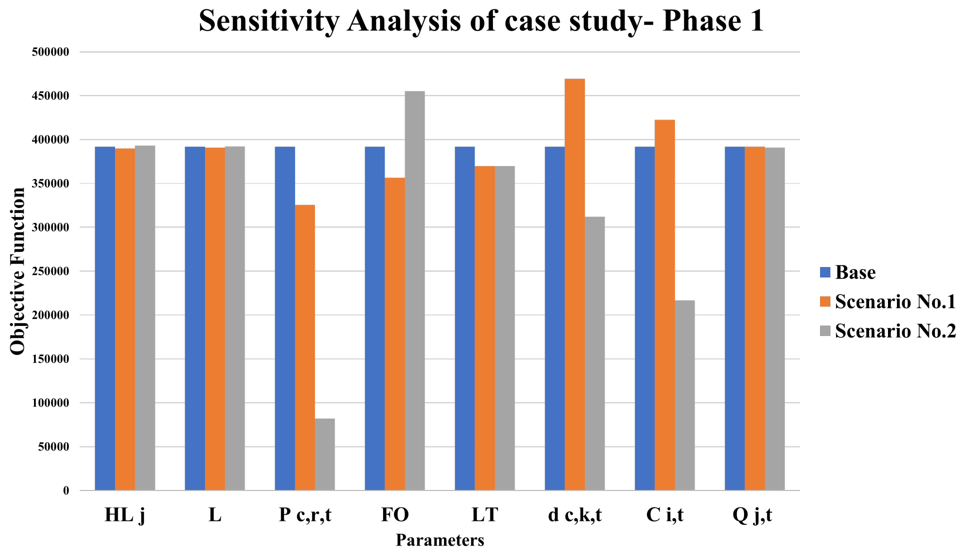

Figure 4.14: Result of sensitivity analysis of case study – Phase 1

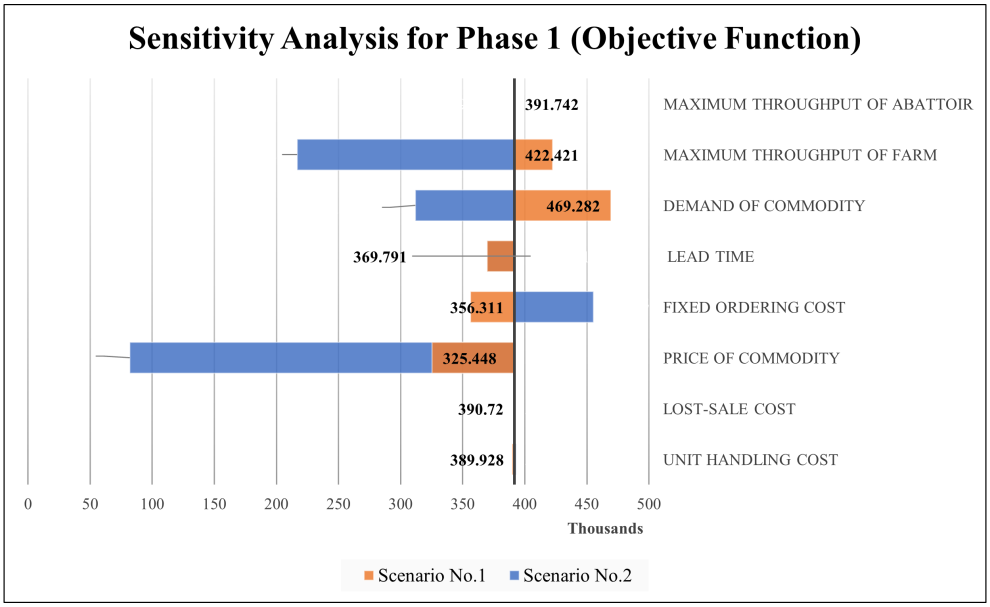

Figure 4.15: Objective function values – Phase 1

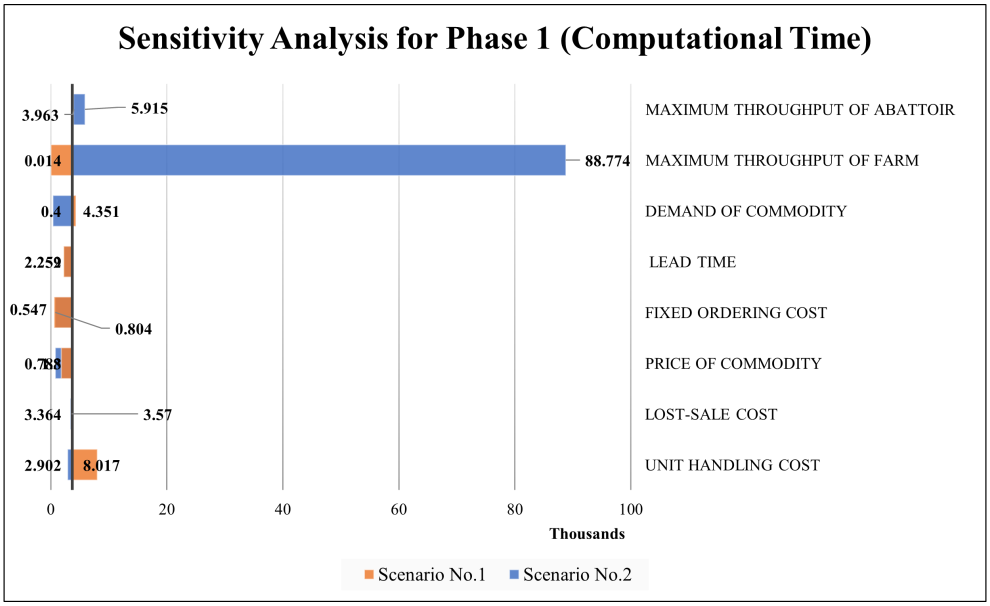

Figure 4.16: Computational time values – Phase 1

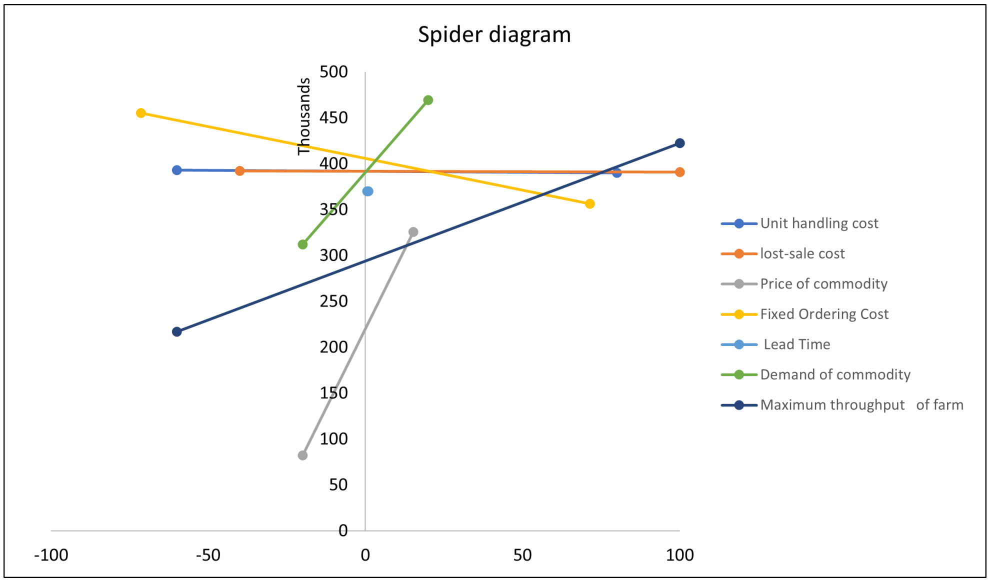

Figure 4.17: Spider diagram of sensitivity analysis – Phase 1

For Phase 2

In this section, we aim at finding the shipment planning by solving the multi-commodity service network design problems. We fed the set of possible sea vessels and airline services into the second model. The decision maker owns a set of sea vessels, where routing and scheduling plan is decided at strategic level and they are not subject of decision. The airline services can be used to ship urgent demands but at higher shipping rates.

- The time horizon for the Phase 2 is 23 days
- The objective function of the Phase 2 is to minimize the total costs including fixed cost of booking services and transportation cost
- The computational time for the results of instants is few seconds.

Table 4.3 and Figure 4.18 present the result of phase 2 including the itinerary of each shipment. For example, the live sheep demands required for UAE have been shipped via service 1 of sea vessel, while the second commodity (chilled lamb) for UAE was shipped by air using service no. 6.

Table 4.3: Result of transportation decision of case study – Phase 2

Case Study - Phase 2 results					
Demand	**Commodity**	**Service**	**Origin**	**Destination**	**Type of transportation**
1	1	1	AUS	UAE	Vessel-by Sea
2	1	1	AUS	UAE	Vessel-by Sea
3	1	1	AUS	UAE	Vessel-by Sea
3	1	2	UAE	KUW	Vessel-by Sea
4	1	1	AUS	UAE	Vessel-by Sea
4	1	2	UAE	KUW	Vessel-by Sea
5	1	1	AUS	UAE	Vessel-by Sea
5	1	2	UAE	KUW	Vessel-by Sea
5	1	3	KUW	QAT	Vessel-by Sea
1	2	6	AUS	UAE	Airplane - by air
2	2	6	AUS	UAE	Airplane - by air
1	3	1	AUS	UAE	Vessel-by Sea
2	3	1	AUS	UAE	Vessel-by Sea
3	3	1	AUS	UAE	Vessel-by Sea
3	3	2	UAE	KUW	Vessel-by Sea
4	3	1	AUS	UAE	Vessel-by Sea
4	3	2	UAE	KUW	Vessel-by Sea
5	3	1	AUS	UAE	Vessel-by Sea
5	3	2	UAE	KUW	Vessel-by Sea
5	3	3	KUW	QAT	Vessel-by Sea
1	4	1	AUS	UAE	Vessel-by Sea
2	4	1	AUS	UAE	Vessel-by Sea
3	4	1	AUS	UAE	Vessel-by Sea
3	4	2	UAE	KUW	Vessel-by Sea
4	4	1	AUS	UAE	Vessel-by Sea
4	4	2	UAE	KUW	Vessel-by Sea
5	4	1	AUS	UAE	Vessel-by Sea
5	4	2	UAE	KUW	Vessel-by Sea
5	4	3	KUW	QAT	Vessel-by Sea
1	5	6	AUS	UAE	Airplane - by air
2	5	6	AUS	UAE	Airplane - by air
1	6	1	AUS	UAE	Vessel-by Sea
2	6	1	AUS	UAE	Vessel-by Sea
3	6	1	AUS	UAE	Vessel-by Sea
3	6	2	UAE	KUW	Vessel-by Sea
4	6	1	AUS	UAE	Vessel-by Sea
4	6	2	UAE	KUW	Vessel-by Sea
5	6	1	AUS	UAE	Vessel-by Sea
5	6	2	UAE	KUW	Vessel-by Sea
5	6	3	KUW	QAT	Vessel-by Sea

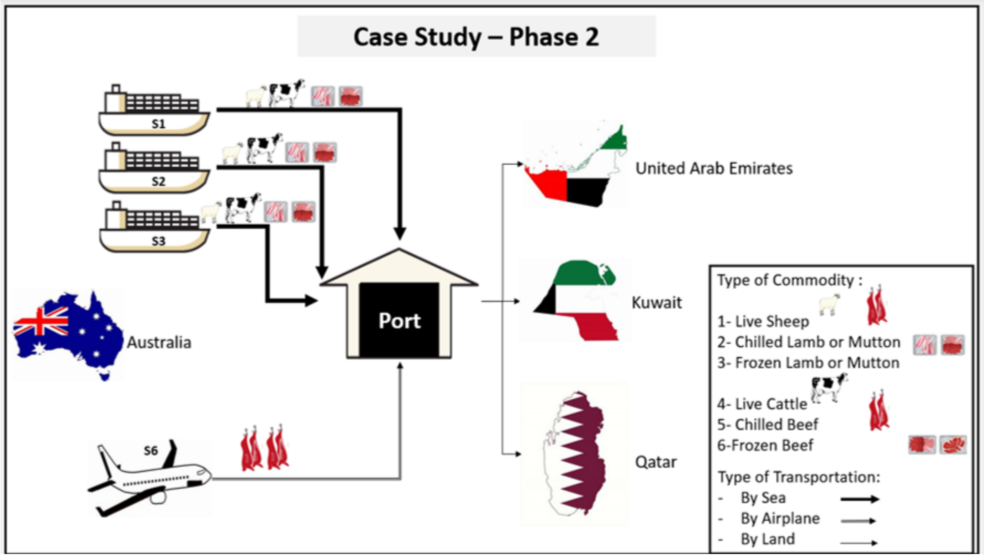

Figure 4.18: Transportation decision of case study – Phase 2

The time horizon of meat supply chain process is given in Figure 4.19. It is very obvious that the planning of livestock should plan at least 30 days; 15 days for sea transportation, 8 days inspection at port, distribution and quarantine of livestock and finally 7 days for meat activities processing including distribution to retailers. We fed the amount of livestock to be delivered for each farm using the result of Phase 1. Other meat and its products like frozen meat were also integrated in the planning model.

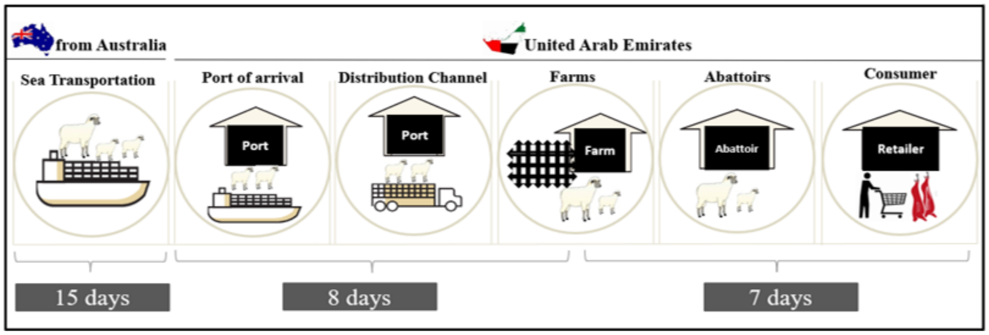

Figure 4.19: Time horizon of Phase 1 and 2

Table 4.4: Sensitivity analysis of Phase 2

Sensitivity Analysis for Phase 2 Model						
Parameters	**Objective Function**			**Time in Seconds**		
	Base	**Scenario No.1**	**Scenario No.2**	**Base**	**Scenario No.1**	**Scenario No.2**
(1) $C_{s,c}$	271100	271100	271100	0.2	0.05	0.03
(2) F_s	271100	381100	206100	0.2	0.05	0.16
(3) $d_{d,c}$	271100	371143	471189	0.2	0.03	0.05

In this part we will present the results of the sensitivity analysis for Phase 2 by changing one parameter at a time: (1) capacity of the service; (2) fixed booking cost; and (3) demand of commodity. Table 4.4 with Figure 4.20 - 4.23 summarize the result of the sensitivity analysis. As shown below, the most significant parameters are demand of commodity and fixed booking cost. The computational time decreases significantly when the capacity of transportation service decreases by 76%. An examination of all the instances suggests that, the fixed booking cost of services and demand of commodities are the most significant factors in determining the routing and scheduling plan of shipments.

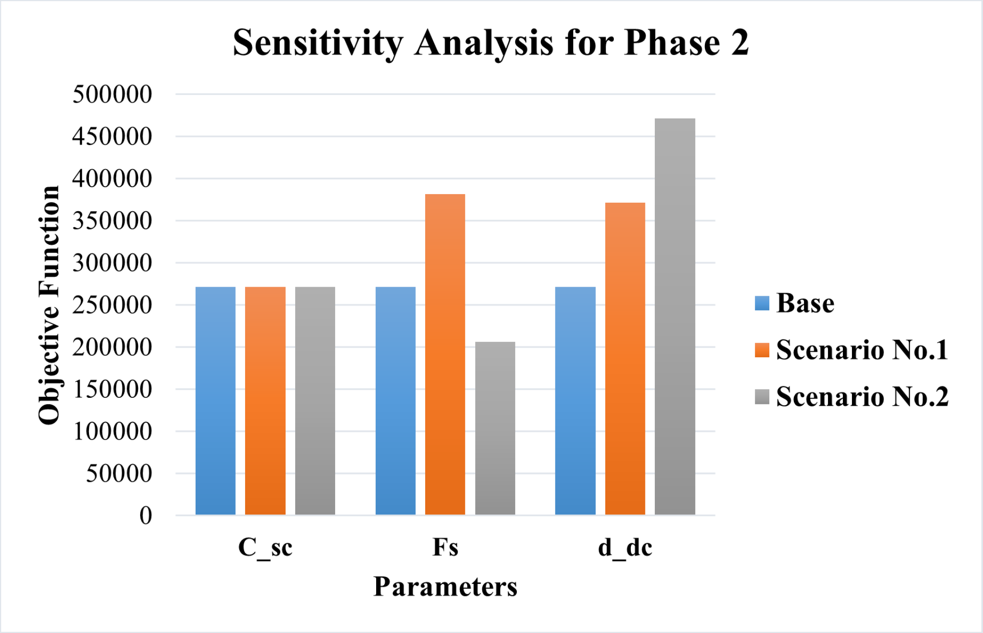

Figure 4.20: Sensitivity analysis of case study – Phase 2

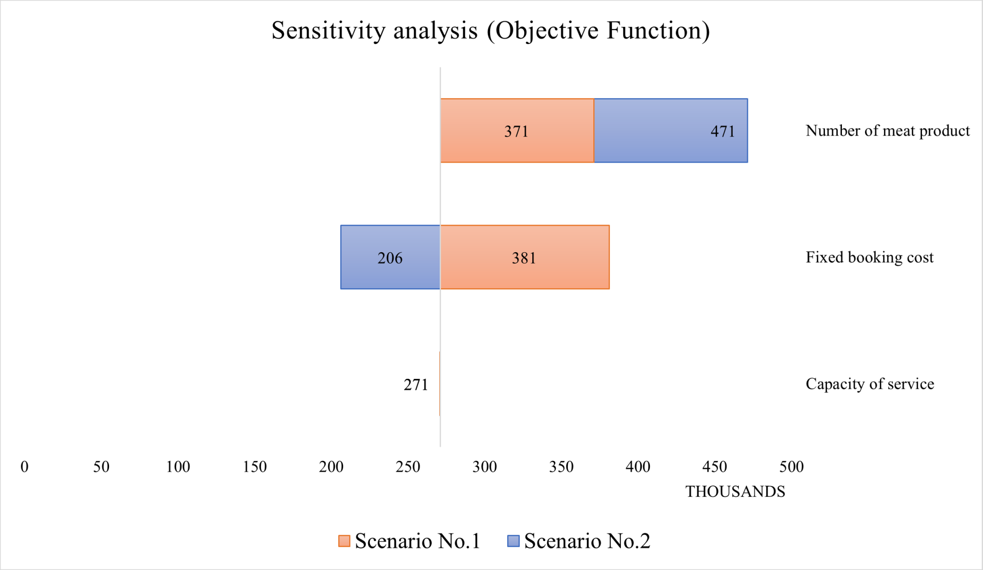

Figure 4.21: Objective function values – Phase 2

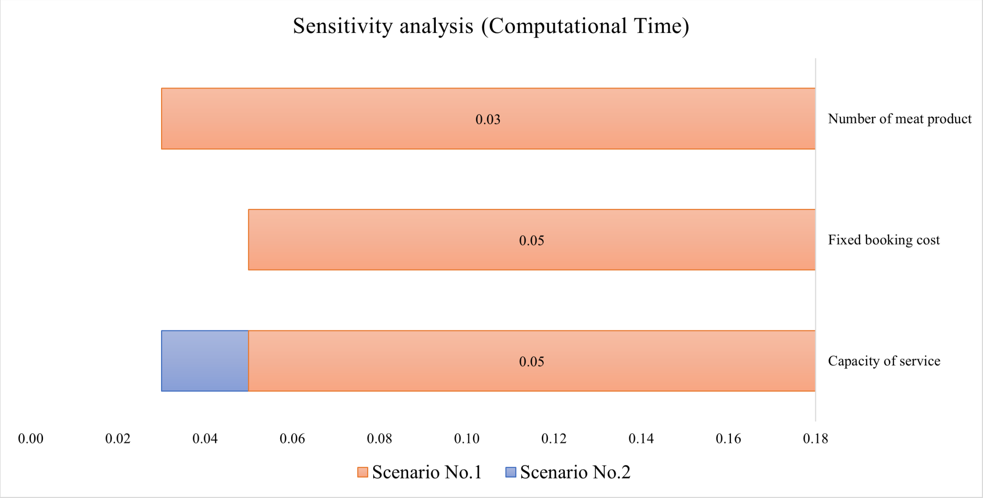

Figure 4.22: Computational time values – Phase 2

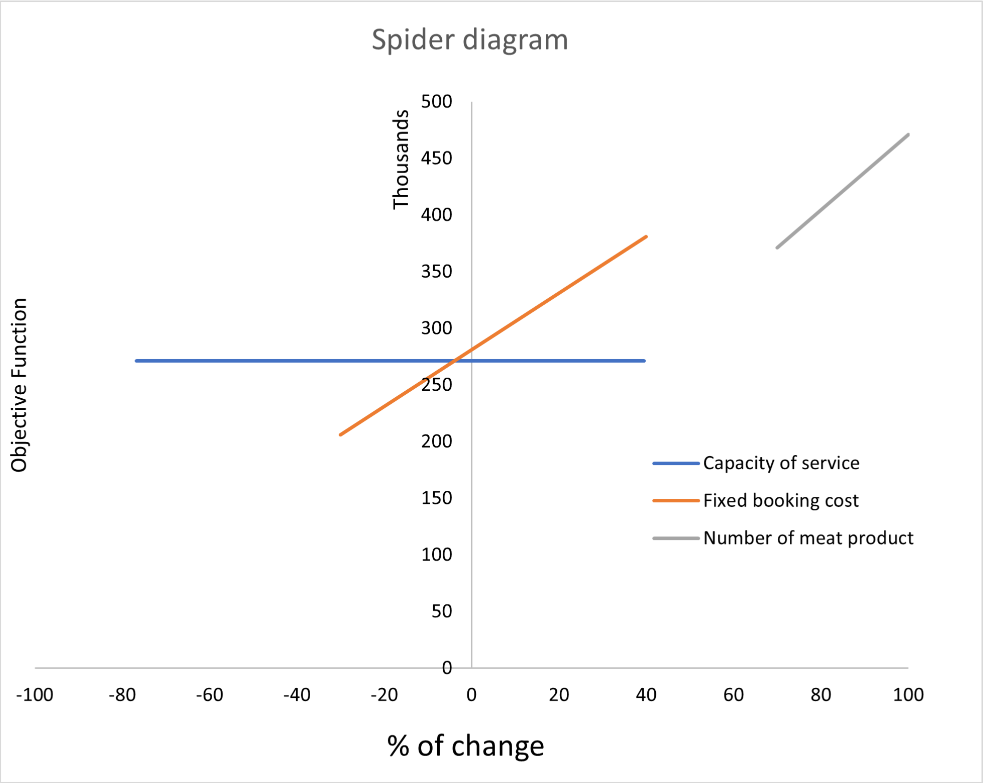

Figure 4.23: Spider diagram of sensitivity analysis – Phase 2

4.4 Summary

In this chapter, we investigated a real case study of meat supply chain using the developed mathematical models for a company located in United Arab Emirates. The proposed mathematical models include two Phases. In the first Phase, we got the optimal allocation of commodity from farms to abattoirs and the quantity required for each retailer. In the second Phase we used the result of the first Phase to develop the routing decisions from the country of origin to the port of destination for each commodity. The computational complexity of the models depends on the size of the problem. All the instances were able to find optimal solution. The computational time for the results of most of the instants are few seconds to several seconds. Finally, sensitivity analysis is performed to investigate the impact of change of some parameters to the objective function and the computational time.

Chapter 5

Conclusion and Future Work

In this study, a comprehensive meat supply chain problem was investigated. Two mathematical models were introduced in Chapter 3. Some considerations were investigated to the planning problem to make it more practical and realistic, i.e., live animal, chilled and frozen meat product were planned together. (s, S) inventory policy was considered and implemented at retailers as most of meat supply chain industries adopt it. In Chapter 3 we introduced a heuristic approach to solve the problem. Using off-shelf optimization for solving the model takes a long time for realistic instances and it cannot be applied on large and practical cases. The heuristic approach uses two phases to reach a sub-optimal solution with minimum computational time. The approach uses two mathematical models. The first Phase uses mixed integer programming model to find (a) the allocation decisions between farms to abattoir and from abattoirs to retailers, (b) the quantity of commodities to be shipped between farms to abattoirs and from abattoirs to retailers and (c) inventory decisions at retailers including the reorder level, order up to level for each meat product. The latter optimization model aims at finding the itinerary for each commodity from source origin to the ports (airport and seaport) by airplanes and ships. The proposed model was used to test a realistic number of example instances, and the results are analyzed. The two phases are solved with Concert Technology in CPLEX. In the end, the proposed heuristic approach is applied to a real case problem based on a meat supply chain problem for a company located in United Arab Emirates. Although the developed heuristic requires much less computing effort, the two optimization problems solved are still NP-Hard. A new method to tackle this problem can be developed in the future. We plan to improve the developed heuristic method in solving large size practical problems for meat supply chain with better performance. Another interesting investigation would be considering one/several cold warehouses as direct upstream activity to the retailer between abattoirs and retailers.

Appendix A

A.1 Phase 1 data: -

Table A.1: Phase 1 - instance size

Case study instance size		
Farm	i	2
Abattoir	j	3
Retailers	k	7
Time period in days	t	7
Commodity	c	1

Table A.2: Fixed allocation cost from i to j

F_{ij}	**Abattoir**		
Farm	$j1$	$j2$	$j3$
$i1$	500	400	550
$i2$	600	500	600

Table A.3: Fixed allocation cost from j to k

G_{jk}	***Retailer***						
Abattoir	$k1$	$k2$	$k3$	$k4$	$k5$	$k6$	$k7$
$j1$	500	500	500	500	500	500	500
$j2$	500	500	500	500	500	500	500
$j3$	500	500	500	500	500	500	500

Table A.4: Unit truck transportation cost from farm to abattoir

TL_{ij}	***Abattoir***		
Farm	$j1$	$j2$	$j3$
$i1$	0.6915	0.054	0.72
$i2$	0.609	0.057	0.705

Table A.5: Unit truck transportation cost from abattoir to retailer

TC_{jk}	***Retailer***						
Abattoir	*k1*	*k2*	*k3*	*k4*	*k5*	*k6*	*k7*
j1	0.186	0.1635	0.0705	0.009	0.0705	0.1095	0.1155
j2	0.591	0.825	0.6825	0.657	0.666	0.7275	0.678
j3	0.6315	0.795	0.747	0.81	0.75	0.78	0.72

Table A.6: Unit feeding cost per commodity

UC_{ci}	***Farm***	
Commodity	*i1*	*i2*
c1	35	35

Table A.7: Unit handling cost

HL_j	
Abattoir	***Handling Cost***
j1	0.5
j2	0.5
j3	0.5

Table A.8: Unit holding cost of commodity c for retailer k at each time period

H_{ck}	***Retailer***						
Commodity	*k1*	*k2*	*k3*	*k4*	*k5*	*k6*	*k7*
c1	0.75	0.75	0.75	0.75	0.75	0.75	0.75

L	Unit lost-sale cost	5
E	Unit deterioration cost of commodity	4
M	Very large number	100000
s	Very small number	0.005
FO	Fixed order cost	3500
LT	Lead time	1
Y_c	Expected commodity extraction ratio	0.9

Table A.9: Price of commodity in period t ,which is produced in period r

c	r	t	***Price***
c1	*r1*	*t1*	250
c1	*r1*	*t2*	240
c1	*r1*	*t3*	240
c1	*r1*	*t4*	215
c1	*r1*	*t5*	200
c1	*r1*	*t6*	180
c1	*r1*	*t7*	150
c1	*r2*	*t2*	260
c1	*r2*	*t3*	250
c1	*r2*	*t4*	250
c1	*r2*	*t5*	210
c1	*r2*	*t6*	200
c1	*r2*	*t7*	180
c1	*r3*	*t3*	240
c1	*r3*	*t4*	240
c1	*r3*	*t5*	220
c1	*r3*	*t6*	200
c1	*r3*	*t7*	170
c1	*r4*	*t4*	230
c1	*r4*	*t5*	230
c1	*r4*	*t6*	215
c1	*r4*	*t7*	200
c1	*r5*	*t5*	260
c1	*r5*	*t6*	260
c1	*r5*	*t7*	250
c1	*r6*	*t6*	250
c1	*r6*	*t7*	240
c1	*r7*	*t7*	240

Table A.10: Fraction of commodity produced at period t ,that is deteriorated at the end of period r

c	r	t	***Fraction***
c1	*r1*	*t1*	0.01
c1	*r1*	*t2*	0.01
c1	*r1*	*t3*	0.1
c1	*r1*	*t4*	0.15
c1	*r1*	*t5*	0.2
c1	*r1*	*t6*	0.25
c1	*r1*	*t7*	0.5
c1	*r2*	*t2*	0.01
c1	*r2*	*t3*	0.1
c1	*r2*	*t4*	0.15
c1	*r2*	*t5*	0.2
c1	*r2*	*t6*	0.25
c1	*r2*	*t7*	0.5
c1	*r3*	*t3*	0.1
c1	*r3*	*t4*	0.15
c1	*r3*	*t5*	0.2
c1	*r3*	*t6*	0.25
c1	*r3*	*t7*	0.5
c1	*r4*	*t4*	0.15
c1	*r4*	*t5*	0.2
c1	*r4*	*t6*	0.25
c1	*r4*	*t7*	0.5
c1	*r5*	*t5*	0.2
c1	*r5*	*t6*	0.25
c1	*r5*	*t7*	0.5
c1	*r6*	*t6*	0.25
c1	*r6*	*t7*	0.5
c1	*r7*	*t7*	0.5

Table A.11: Demand of commodity c in retailer k at time t

d_{ckt}		***Time***						
Commodity	***Retailer***	*t1*	*t2*	*t3*	*t4*	*t5*	*t6*	*t7*
c1	*k1*	120	100	80	100	80	120	80
c1	*k2*	120	100	80	100	80	120	80
c1	*k3*	120	100	80	100	80	120	80
c1	*k4*	120	100	80	100	80	120	80
c1	*k5*	120	100	80	100	80	120	80
c1	*k6*	120	100	80	100	80	120	80
c1	*k7*	120	100	80	100	80	120	80
	Total Demand	840	700	560	700	560	840	560

Table A.12: Maximum throughput of farm i at time t

C_{it}	***Time***						
Farm	*t1*	*t2*	*t3*	*t4*	*t5*	*t6*	*t7*
i1	500	500	500	500	500	500	500
i2	500	500	500	500	500	500	500

Table A.13: Maximum throughput capacity of abattoir j at time t

Q_{jt}	***Time***						
Abattoir	*t1*	*t2*	*t3*	*t4*	*t5*	*t6*	*t7*
j1	2000	2000	2000	2000	2000	2000	2000
j2	2000	2000	2000	2000	2000	2000	2000
j3	2000	2000	2000	2000	2000	2000	2000

A.2 Phase 2 data: -

Table A.14: Phase 2- Instance size

Instance size		
Node	N	4
Commodity	C	6
Transportation Service	S	9
Demand	D	5

Table A.15: Origin and destination of demand

Sr.No	O_d	d_d
1	1	2
2	1	2
3	1	3
4	1	3
5	1	4

Table A.16: Capacity of each service s

C_{sc}
4300000
4300000
4300000
27000
27000
27000
27000
27000
27000

Table A.17: if commodity c can be ship via service s

λ_{cs}	***Service***								
Commodity	*S* ***1***	*S* ***2***	*S* ***3***	*S* ***4***	*S* ***5***	*S* ***6***	*S* ***7***	*S* ***8***	*S* ***9***
C ***1***	1	1	1	0	0	0	0	0	0
C ***2***	0	0	0	1	1	1	1	1	1
C ***3***	1	1	1	1	1	1	1	1	1
C ***4***	1	1	1	0	0	0	0	0	0
C ***5***	0	0	0	1	1	1	1	1	1
C ***6***	1	1	1	1	1	1	1	1	1

Table A.18: Origin and destination of service

O_s	d_s
1	2
2	3
3	4
1	2
1	2
1	2
1	2
1	2
1	2

Table A.19: Cut-off time and schedule arrival for service s

Tau_s	Tar_s
1	15
17	19
20	23
0	0.7
7.0	7.7
14	14.7
21	21.7
28	28.7
35	35.7

Table A.20: Due date for demand D of meat product c

Tdu_{dc}	***Commodity***					
Demand	*C**1***	*C**2***	*C**3***	*C**4***	*C**5***	**C6**
*d**1***	16	16	16	16	16	16
*d**2***	19	19	19	19	19	2
*d**3***	20	20	20	20	20	20
*d**4***	23	23	23	23	23	23
*d**5***	25	25	25	25	25	25

Table A.21: Average weight of commodity c in kilogram

Commodity	W_c
*C **1***	30 kg per head
*C **2***	15 kg per head
*C **3***	20 kg per carton
*C **4***	250 kg per head
*C **5***	150 kg per head
*C **6***	25 kg per carton

Table A.22: Fixed booking cost

F_s
50000
50000
50000
100000
100000
100000
100000
100000
100000

Table A.23: Cost to ship meat product c on service s in dollar per kilogram

T_{cs}	***Service***								
Commodity	*S **1***	*S **2***	*S **3***	*S **4***	*S **5***	*S **6***	*S **7***	*S **8***	*S **9***
*C **1***	154	161.7	169.79	231	323.4	200.2	184.8	184.8	161.7
*C **2***	154	161.7	169.79	231	323.4	200.2	184.8	184.8	161.7
*C **3***	154	161.7	169.79	231	323.4	200.2	184.8	184.8	161.7
*C **4***	870	913.5	959.18	1305	1218	1131	1044	957	913.5
*C **5***	870	913.5	959.18	1305	1218	1131	1044	957	913.5
*C **6***	870	913.5	959.18	1305	1218	1131	1044	957	913.5

Table A.24: Number of meat product c for demand d

d_{dc}	***Commodity***					
Demand	*C **1***	*C **2***	*C **3***	*C **4***	*C **5***	*C **6***
*d**1***	1000	50	50	50	50	50
*d**2***	1000	50	50	50	50	50
*d**3***	1000	0	50	50	0	50
*d**4***	880	0	50	50	0	50
*d**5***	880	0	50	50	0	50

Appendix B

B.1 Sensitivity analysis data - Phase 1

Table B.1: Handling cost at abaittor for scenario 1 & 2

HL_j - ***handling cost at abaittor*** j			
Abaittor j	**Base**	**Scenario No.1**	**Scenario No.2**
$j1$	0.5	0.9	0.2
$j2$	0.5	0.9	0.2
$j3$	0.5	0.9	0.2

Table B.2: Unit lost-sale cost for unmet demands for scenario 1 & 2

L - ***Unit lost-sale cost for unmet demands***		
Base	**Scenario No.1**	**Scenario No.2**
5	10	3

Table B.3: Price of commodity for scenario 1 & 2

P_{crt} - ***Price of commodity*** c ***in period*** t***, for commodity produced in period*** r					
c	***r***	***t***	**Base**	**Scenario No.1**	**Scenario No.2**
c1	r1	t1	250	288	200
c1	r1	t2	240	276	192
c1	r1	t3	240	276	192
c1	r1	t4	215	247	172
c1	r1	t5	200	230	160
c1	r1	t6	180	207	144
c1	r1	t7	150	173	120
c1	r2	t2	260	299	208
c1	r2	t3	250	288	200
c1	r2	t4	250	288	200
c1	r2	t5	210	242	168
c1	r2	t6	200	230	160
c1	r2	t7	180	207	144
c1	r3	t3	240	276	192
c1	r3	t4	240	276	192
c1	r3	t5	220	253	176
c1	r3	t6	200	230	160
c1	r3	t7	170	196	136
c1	r4	t4	230	265	184
c1	r4	t5	230	265	184
c1	r4	t6	215	247	172
c1	r4	t7	200	230	160
c1	r5	t5	260	299	208
c1	r5	t6	260	299	208
c1	r5	t7	250	288	200
c1	r6	t6	250	288	200
c1	r6	t7	240	276	192
c1	r7	t7	240	276	192

Table B.4: Fixed ordering cost for scenario 1 & 2

FO - ***Fixed ordering cost***		
Base	**Scenario No.1**	**Scenario No.2**
3500	6000	1000

Table B.5: Lead Time for for scenario 1 & 2

LT - ***Lead Time***		
Base	**Scenario No.1**	**Scenario No.2**
1.0	0.9	0.5

Table B.6: Demand of commodity c in retailer k at time period t for scenario 1 & 2

d_{ckt} - ***Demand of commodity*** c ***in retailer*** k ***at time period*** t								
	Retailer/Period	*t1*	*t2*	*t3*	*t4*	*t5*	*t6*	*t7*
Base	*k1*	120	100	80	100	80	120	80
	k2	120	100	80	100	80	120	80
	k3	120	100	80	100	80	120	80
	k4	120	100	80	100	80	120	80
	k5	120	100	80	100	80	120	80
	k6	120	100	80	100	80	120	80
	k7	120	100	80	100	80	120	80
Scenario No.2	*k1*	144	120	96	120	96	144	96
	k2	144	120	96	120	96	144	96
	k3	144	120	96	120	96	144	96
	k4	144	120	96	120	96	144	96
	k5	144	120	96	120	96	144	96
	k6	144	120	96	120	96	144	96
	k7	144	120	96	120	96	144	96
Scenario No.3	*k1*	96	80	64	80	64	96	64
	k2	96	80	64	80	64	96	64
	k3	96	80	64	80	64	96	64
	k4	96	80	64	80	64	96	64
	k5	96	80	64	80	64	96	64
	k6	96	80	64	80	64	96	64
	k7	96	80	64	80	64	96	64

Table B.7: Maximum throughput of farm for scenario 1 & 2

C_{it}- ***Maximum throughput of farm i at time period t***								
	Farm/Time	*t1*	*t2*	*t3*	*t4*	*t5*	*t6*	*t7*
Base	*i1*	500	500	500	500	500	500	500
	i2	500	500	500	500	500	500	500
Scenario No.1	*i1*	1000	1000	1000	1000	1000	1000	1000
	i2	1000	1000	1000	1000	1000	1000	1000
Scenario No.2	*i1*	200	200	200	200	200	200	200
	i2	200	200	200	200	200	200	200

Table B.8: Maximum throughput capacity of abattoir for scenario 1 & 2

Q_{jt} - ***Maximum throughput capacity of abattoir*** j ***at time period*** t								
	Abattoir/Time	*t1*	*t2*	*t3*	*t4*	*t5*	*t6*	*t7*
Base	*j1*	2000	2000	2000	2000	2000	2000	2000
	j2	2000	2000	2000	2000	2000	2000	2000
	j3	2000	2000	2000	2000	2000	2000	2000
Scenario No.1	*j1*	5000	5000	5000	5000	5000	5000	5000
	j2	5000	5000	5000	5000	5000	5000	5000
	j3	5000	5000	5000	5000	5000	5000	5000
Scenario No.2	*j1*	500	500	500	500	500	500	500
	j2	500	500	500	500	500	500	500
	j3	500	500	500	500	500	500	500

B.2 Sensitivity analysis data - Phase 2

Table B.9: Capacity of service per meat product in kilogram for scenario 1 & 2

C_{sc} - ***Capacity of service*** s ***per meat product*** c ***in kilogram***			
Service S	**Base**	**Scenario No.1**	**Scenario No.2**
S1	4300000	1000000	6000000
S2	4300000	1000000	6000000
S3	4300000	1000000	6000000
S4	27000	25000	60000
S5	27000	25000	60000
S6	27000	25000	60000
S7	27000	25000	60000
S8	27000	25000	60000
S9	27000	25000	60000

Table B.10: Fixed booking cost to reserve space on service for scenario 1 & 2

Fs - ***Fixed booking cost to reserve space on service*** s			
Service S	**Base**	**Scenario No.1**	**Scenario No.2**
S1	50000	70000	35000
S2	50000	70000	35000
S3	50000	70000	35000
S4	100000	150000	80000
S5	100000	150000	80000
S6	100000	150000	80000
S7	100000	150000	80000
S8	100000	150000	80000
S9	100000	150000	80000

Table B.11: Number of meat product for demand for scenario 1 & 2

d_{dc} - ***Number of meat product*** c ***for demand*** d							
Base	*Demand/Commodity*	*C1*	*C2*	*C3*	*C4*	*C5*	*C6*
	d1	1000	50	50	50	50	50
	d2	1000	50	50	50	50	50
	d3	1000	0	50	50	0	50
	d4	880	0	50	50	0	50
	d5	880	0	50	50	0	50
Scenario No.1	*d1*	2000	100	100	100	100	100
	d2	2000	100	100	100	100	100
	d3	2000	0	100	100	0	100
	d4	1760	0	100	100	0	100
	d5	1760	0	100	100	0	100
Scenario No.2	*d1*	1700	170	170	170	170	170
	d2	1700	170	170	170	170	170
	d3	1700	0	170	170	0	170
	d4	1190	0	170	170	0	170
	d5	1105	0	170	170	0	170

Appendix C

C.1 Phase 1 model code in Concert Technology in CPLEX:

```
    #include < ilcplex/ilocplex.h >
ILOSTLBEGIN
#include < vector >
#include < fstream >
#include < iostream >
#include < iostream >
#include < fstream >

    int * create_int_vector(intdim);
longdouble * create
double
vector(intdim);

    template < typenameT >
T * * * AllocateDynamic3DArrayCSTYLE(intnRows, intnCols, intnDep);

    template < typenameT >
voidFreeDynamic3DArrayCSTYLE(T * * * dArray, intnRows, intnCols, intnDep);
```

$template < typenameT >$
$T * AllocateDynamic1DArrayCSTYLE(intnRows);$

$template < typenameT >$
$voidFreeDynamic1DArrayCSTYLE(T * dArray, intnRows);$

$template < typenameT >$
$T * *AllocateDynamic2DArrayCSTYLE(intnRows, intnCols);$

$template < typenameT >$
$voidFreeDynamic2DArrayCSTYLE(T * *dArray, intnRows, intnCols);$

$//WriteaMIPModelfortheinformationgivenbelow$

$//SetsandIndicesB > 2{-}Farms, A > 2{-}Abattoirs, I > 2{-}Retailers, T > TimePeriod(1week), P > 2 - Products(Sheep\&Cattle)$

$int * B = newint(2);$
$int * A = newint(2);$
$int * I = newint(2);$
$int * T = newint(3);$
$int * P = newint(2);$

$//Parameters$

$double * *F_ab; //Fixedallocationcostofabattoirsatofarmscenterb$
$double * *G_ai; //Fixedallocationcostofretaileritoabattoira$
$double**TL_ba; //Unittrucktransportationcostoflivestockassociatedwithgoingfromfarmcenterbtoabatto$
$double**TC_ai; //Unittrucktransportationcostassociatedwithgoingfromabattoiratoretaileri$

$double * *UC_pb; //Unitlivestockbreeding/feedingcostperproductpatfarmcenterb$

$double * HL_a; //Unitlivestockhandlingcostforabattoiraateachtimeperiod$

$double * *H_pi; //Unitholdingcostofproductpforretailer/warehouseiateachtimeperiod$

$double * L; //Unitlost - salecostforunmetdemands$

$double * * * W_prt; //Priceofproductpinperiodt, whichisproducedinperiodr$

$double * E; //Unitdeteriorationcostofproducts$

$double***B_prt; //Fractionofproductp, producedattimeperiodrthatisdeterioratedattheendofperiodt$

$double * M; //Averybignumber$

$double * c; //Averysmallnumber$

$double * FO; //Fixedorderingcost$

$double * LT; //Lead - timebetweenorderplacementbyretailersandthetimeorderisreceived$

$double * Y_p; //Expectedproductpextractionratiofromaslaughteredlivestockinabattoirs$

$int * * * d_pit; //Demandofproductpinretaileriattimeperiodt$

$int * *C_pt; //Maximumthroughputoffarm/breedingcenterbattimeperiodt$

$int * *Q_at; //Maximumthroughputcapacityofabattoiraattimeperiodt$

$voidreadData();$

$voidfreeData();$

$voidCall_Optimizaer();$

$ofstreamoutput(”Result.txt”);$

$intmain(intargc, char * *argv)$

{

$readData();$

$Call_Optimizaer();$

$freeData();$

$return0;$

}

$voidreadData()\{$

$int * b = newint(0);$

$int * a = newint(0);$

$int * i = newint(0);$

$int * t = newint(0);$

$int * r = newint(0);$

$int * e = newint(0);$

$int * p = newint(0);$

$F_ab = AllocateDynamic2DArrayCSTYLE < double > (*A, *B);$

$G_ai = AllocateDynamic2DArrayCSTYLE < double > (*A, *I);$

$TL_ba = AllocateDynamic2DArrayCSTYLE < double > (*B, *A);$

$TC_ai = AllocateDynamic2DArrayCSTYLE < double > (*A, *I);$

$UC_pb = AllocateDynamic2DArrayCSTYLE < double > (*P, *B);$

$HL_a = AllocateDynamic1DArrayCSTYLE < double > (*A);$

$H_pi = AllocateDynamic2DArrayCSTYLE < double > (*P, *I);$

$W_prt = AllocateDynamic3DArrayCSTYLE < double > (*P, *T, *T);$

$B_prt = AllocateDynamic3DArrayCSTYLE < double > (*P, *T, *T);$

$Y_p = AllocateDynamic1DArrayCSTYLE < double > (*P);$

$d_pit = AllocateDynamic3DArrayCSTYLE < int > (*P, *I, *T);$

$C_pt = AllocateDynamic2DArrayCSTYLE < int > (*P, *T);$

$Q_at = AllocateDynamic2DArrayCSTYLE < int > (*A, *T);$

```
    L = newdouble(0);
E = newdouble(0);
M = newdouble(0);
c = newdouble(0);
FO = newdouble(0);
LT = newdouble(0);

    / * readdata * /
ifstreamin(”DataSet1.txt”);

    in >> *B;
in >> *A;
in >> *I;
in >> *T;
in >> *P;

    cout << ”ISNTANCE : ” << ”” << *B << ”” << *A << ”” << *I << ”” <<
*T << ”” << *P << endl;

    for(*b = 0; *b < *B; *b = *b + 1){
for(*a = 0; *a < *A; *a = *a + 1){
in >> F_ab[*a][*b];
}
}

    //for(*b = 0; *b < *B; *b = *b + 1){
//for(*a = 0; *a < *A; *a = *a + 1){
//cout << F_ab[*a][*b] << ””;
//}
//cout << endl;
/*}

    ;}*/
for(*a = 0; *a < *A; *a = *a + 1){
for(*i = 0; *i < *I; *i = *i + 1){
in >> G_ai[*a][*i];
}
}

    //for(*a = 0; *a < *A; *a = *a + 1){for(*i = 0; *i < *I; *i = *i + 1){
//for(*i = 0; *i < *I; *i = *i + 1){cout << G_ai[*a][*i] << ””;
//cout << G_ai[*a][*i] << ””; }
//}cout << endl;
//cout << endl; }
//}

    for(*b = 0; *b < *B; *b = *b + 1){
for(*a = 0; *a < *A; *a = *a + 1){
in >> TL_ba[*b][*a];
}
}

    /* for(*b = 0; *b < *B; *b = *b + 1){
for(*a = 0; *a < *A; *a = *a + 1){
cout << TL_ba[*b][*a] << ””;
}
cout << endl;
}*/
```

$for(*a=0;*a<*A;*a=*a+1)\{$

$for(*i=0;*i<*I;*i=*i+1)\{$

$in>>TC_ai[*a][*i];$

$\}$

$\}$

$/*for(*a=0;*a<*A;*a=*a+1)\{$

$for(*i=0;*i<*I;*i=*i+1)\{$

$cout<<TC_ai[*a][*i]<<””;$

$\}$

$cout<<endl;$

$\}*/$

$for(*p=0;*p<*P;*p=*p+1)\{$

$for(*b=0;*b<*B;*b=*b+1)\{$

$in>>UC_pb[*p][*b];$

$\}$

$\}$

$//for(*p=0;*p<*P;*p=*p+1)\{for(*b=0;*b<*B;*b=*b+1)\{$

$//for(*b=0;*b<*B;*b=*b+1)\{cout<<UC_pb[*p][*b]<<””;$

$//cout<<UC_pb[*p][*b]<<””;\}$

$//\}cout<<endl;$

$//cout<<endl;$

$//\}$

$for(*a=0;*a<*A;*a=*a+1)\{$

```
    in >> HL_a[*a];

    }

    /*for(*a = 0; *a < *A; *a = *a + 1){

    cout << HL_a[*a] << "";
}
cout << endl;
}*/

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
in >> H_pi[*p][*i];
}
}

    //for(*p = 0; *p < *P; *p = *p + 1){for(*i = 0; *i < *I; *i = *i + 1){
//for(*i = 0; *i < *I; *i = *i + 1){cout << H_pi[*p][*i] << "";
//cout << H_pi[*p][*i] << "";}
//}cout << endl;
//cout << endl;
//}

    in >> *L;
//cout << *L << endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*r = 0; *r < *T; *r = *r + 1){
```

```
for(*t = 0; *t < *T; *t = *t + 1){
in >> W_prt[*p][*r][*t];
}
}
}

    /* for(*p = 0; *p < *P; *p = *p + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
cout << W_prt[*p][*r][*t] << ””;
}
cout << endl;
}
}
*/

    in >> *E;
//cout << *E << endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
in >> (B_prt[*p][*r][*t]);
B_prt[*p][*r][*t] = B_prt[*p][*r][*t]/100;
}
}
}

    /* for(*p = 0; *p < *P; *p = *p + 1){
```

```
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
cout << B_prt[*p][*r][*t] << ””;
}
cout << endl;
}
} * /

    in >> *M;
//cout << *M << endl;

    in >> *c;
//cout << *c << endl;

    in >> *FO;
//cout << *FO << endl;

    in >> *LT;
//cout << *LT << endl;

    for(*p = 0; *p < *P; *p = *p + 1){

    in >> Y_p[*p];
}

    //for(*p = 0; *p < *P; *p = *p + 1){
//
//cout << Y_p[*p] << ””;
//}
```

```
//cout << endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
in >> d_pit[*p][*i][*t];
}
}
}

    /* for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
cout << d_pit[*p][*i][*t] << ””;
}
cout << endl;
}
} */

    for(*p = 0; *p < *P; *p = *p + 1){
for(*t = 0; *t < *T; *t = *t + 1){


    in >> C_pt[*p][*t];


    }
}

    /* for(*p = 0; *p < *P; *p = *p + 1){
for(*t = 0; *t < *T; *t = *t + 1){
```

$cout << C_pt[*p][*t] << ""$;

}

$cout << endl;$

}

$*/$

$for(*a = 0; *a < *A; *a = *a + 1)\{$

$for(*t = 0; *t < *T; *t = *t + 1)\{$

$in >> Q_at[*a][*t];$

}

}

$//for(*a = 0; *a < *A; *a = *a + 1)\{for(*t = 0; *t < *T; *t = *t + 1)\{$

$//for(*t = 0; *t < *T; *t = *t + 1)\{$

$//cout << Q_at[*a][*t] << ""$;

$//cout << Q_at[*a][*t] << ""$; }

$//\}cout << endl;$

$//cout << endl;\}$

$//\}$

$//$

$//$

}

$voidCall_Optimizaer()\{$

$IloEnvenv;$

$/ * IloNumstartTime; */$
$try\{$

$longdouble * maxErrDefault = newlongdouble(1e - 6);$
$time_tstart, end;$

$start = clock();$

$IloModelMODEL1(env);$

$int * b = newint(0);$
$int * a = newint(0);$
$int * i = newint(0);$
$int * t = newint(0);$
$int * r = newint(0);$
$int * e = newint(0);$
$int * p = newint(0);$

$double * total_time = newdouble(0);$

$charvarName[100];$

$IloArray < IloBoolVarArray > O(env, *B);$

$for(*b = 0; *b < *B; *b = *b + 1)\{$
$O[*b] = IloBoolVarArray(env, *A);$
$\}$

$for(*b = 0; *b < *B; *b = *b + 1)\{$

```
for(*a = 0; *a < *A; *a = *a + 1){
sprintf(varName, ”Ob%da%d”, *b + 1, *a + 1);
O[*b][*a].setName(varName);
}
}

    IloArray < IloBoolVarArray > Z(env, *A);

    for(*a = 0; *a < *A; *a = *a + 1){
Z[*a] = IloBoolVarArray(env, *I);
}

    for(*a = 0; *a < *A; *a = *a + 1){
for(*i = 0; *i < *I; *i = *i + 1){
sprintf(varName, ”Za%di%d”, *a + 1, *i + 1);
Z[*a][*i].setName(varName);
}
}

    IloArray < IloArray < IloBoolVarArray >> Y(env, *P);

    for(*p = 0; *p < *P; *p = *p + 1){
Y[*p] = IloArray < IloBoolVarArray > (env, *I);
for(*i = 0; *i < *I; *i = *i + 1){
Y[*p][*i] = IloBoolVarArray(env, *T);
}
}

    for(*p = 0; *p < *P; *p = *p + 1){
```

```
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
sprintf(varName, ”Yp%di%dt%d”, *p + 1, *i + 1, *t + 1);
Y[*p][*i][*t].setName(varName);
}
}
}

    IloArray < IloArray < IloBoolVarArray >> U(env, *P);

    for(*p = 0; *p < *P; *p = *p + 1){
U[*p] = IloArray < IloBoolVarArray > (env, *I);
for(*i = 0; *i < *I; *i = *i + 1){
U[*p][*i] = IloBoolVarArray(env, *T);
}
}

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
sprintf(varName, ”Up%di%dt%d”, *p + 1, *i + 1, *t + 1);
U[*p][*i][*t].setName(varName);
}
}
}

    IloArray < IloNumVarArray > SL(env, *P);

    for(*p = 0; *p < *P; *p = *p + 1){
```

```
SL[*p] = IloNumVarArray(env, *I, 0, INFINITY, ILOFLOAT);
}

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
sprintf(varName, ”SLp%di%d”, *p + 1, *i + 1);
SL[*p][*i].setName(varName);
}
}

    IloArray < IloNumVarArray > SU(env, *P);

    for(*p = 0; *p < *P; *p = *p + 1){
SU[*p] = IloNumVarArray(env, *I, 0, INFINITY, ILOFLOAT);
}

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
sprintf(varName, ”SUp%di%d”, *p + 1, *i + 1);
SU[*p][*i].setName(varName);
}
}

    IloArray < IloArray < IloArray < IloNumVarArray >>> dl(env, *P);
for(*p = 0; *p < *P; *p = *p + 1){
dl[*p] = IloArray < IloArray < IloNumVarArray >> (env, *I);
for(*i = 0; *i < *I; *i = *i + 1){
dl[*p][*i] = IloArray < IloNumVarArray > (env, *T);
for(*r = 0; *r < *T; *r = *r + 1){
```

```
dl[*p][*i][*r] = IloNumVarArray(env, *T, 0, INFINITY, ILOFLOAT);
}
}
}
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
sprintf(varName, ”dlp%di%dr%dt%d”, *p + 1, *i + 1, *r + 1, *t + 1);
dl[*p][*i][*r][*t].setName(varName);
}
}
}
}

    IloArray < IloArray < IloArray < IloNumVarArray >>> W(env, *B);
for(*b = 0; *b < *B; *b = *b + 1){
W[*b] = IloArray < IloArray < IloNumVarArray >> (env, *A);
for(*a = 0; *a < *A; *a = *a + 1){
W[*b][*a] = IloArray < IloNumVarArray > (env, *P);
for(*p = 0; *p < *P; *p = *p + 1){
W[*b][*a][*p] = IloNumVarArray(env, *T, 0, INFINITY, ILOFLOAT);
}
}
}

    for(*b = 0; *b < *B; *b = *b + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*p = 0; *p < *P; *p = *p + 1){
```

```
for(*t = 0; *t < *T; *t = *t + 1){
sprintf(varName, ”Wb%da%dp%dt%d”, *b + 1, *a + 1, *p + 1, *t + 1);
W[*b][*a][*p][*t].setName(varName);
}
}
}
}
IloArray < IloArray < IloArray < IloNumVarArray >>> X(env, *P);
for(*p = 0; *p < *P; *p = *p + 1){
X[*p] = IloArray < IloArray < IloNumVarArray >> (env, *A);
for(*a = 0; *a < *A; *a = *a + 1){
X[*p][*a] = IloArray < IloNumVarArray > (env, *I);
for(*i = 0; *i < *I; *i = *i + 1){
X[*p][*a][*i] = IloNumVarArray(env, *T, 0, INFINITY, ILOFLOAT);
}
}
}
for(*p = 0; *p < *P; *p = *p + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
sprintf(varName, ”Xp%da%di%dt%d”, *p + 1, *a + 1, *i + 1, *t + 1);
X[*p][*a][*i][*t].setName(varName);
}
}
}
}

    IloArray < IloArray < IloArray < IloNumVarArray >>> Ib(env, *P);
```

```
for(*p = 0; *p < *P; *p = *p + 1){
Ib[*p] = IloArray < IloArray < IloNumVarArray >> (env, *I);
for(*i = 0; *i < *I; *i = *i + 1){
Ib[*p][*i] = IloArray < IloNumVarArray > (env, *T);
for(*r = 0; *r < *T; *r = *r + 1){
Ib[*p][*i][*r] = IloNumVarArray(env, *T, 0, INFINITY, ILOFLOAT);
}
}
}
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
sprintf(varName, ”Ibp%di%dr%dt%d”, *p + 1, *i + 1, *r + 1, *t + 1);
Ib[*p][*i][*r][*t].setName(varName);
}
}
}
}

    IloArray < IloArray < IloArray < IloNumVarArray >>> Ie(env, *P);
for(*p = 0; *p < *P; *p = *p + 1){
Ie[*p] = IloArray < IloArray < IloNumVarArray >> (env, *I);
for(*i = 0; *i < *I; *i = *i + 1){
Ie[*p][*i] = IloArray < IloNumVarArray > (env, *T);
for(*r = 0; *r < *T; *r = *r + 1){
Ie[*p][*i][*r] = IloNumVarArray(env, *T, 0, INFINITY, ILOFLOAT);
}
}
```

```
}
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
sprintf(varName, ”Iep%di%dr%dt%d”, *p + 1, *i + 1, *r + 1, *t + 1);
Ie[*p][*i][*r][*t].setName(varName);
}
}
}
}

    IloArray < IloArray < IloNumVarArray >> Lk(env, *P);

    for(*p = 0; *p < *P; *p = *p + 1){
Lk[*p] = IloArray < IloNumVarArray > (env, *I);
for(*i = 0; *i < *I; *i = *i + 1){
Lk[*p][*i] = IloNumVarArray(env, *T, 0, INFINITY, ILOFLOAT);

    }
}

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
sprintf(varName, ”Lkp%di%dt%d”, *p + 1, *i + 1, *t + 1);
Lk[*p][*i][*t].setName(varName);
}
}
```

```
$}$

    $IloExprOBJ(env);$

    $for(*p = 0; *p < *P; *p = *p + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$for(*r = 0; *r < *T; *r = *r + 1)\{$
$for(*t = 0; *t < *T; *t = *t + 1)\{$
$if(*r <= *t)\{$
$OBJ+ = 1000 * W_prt[*p][*r][*t] * dl[*p][*i][*r][*t];$
$}$
$}$
$}$
$}$
$}$

    $for(*b = 0; *b < *B; *b = *b + 1)\{$
$for(*a = 0; *a < *A; *a = *a + 1)\{$
$OBJ- = F_ab[*a][*b] * O[*b][*a];$
$}$
$}$

    $for(*a = 0; *a < *A; *a = *a + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$OBJ- = G_ai[*a][*i] * Z[*a][*i];$

    $}$
$}$
```

```
    for(*p = 0; *p < *P; *p = *p + 1){
for(*b = 0; *b < *B; *b = *b + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*r = 0; *r < *T; *r = *r + 1){
OBJ- = UC_pb[*p][*b] * W[*b][*a][*p][*r];
}
}
}
}
for(*p = 0; *p < *P; *p = *p + 1){
for(*b = 0; *b < *B; *b = *b + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*r = 0; *r < *T; *r = *r + 1){
OBJ- = TL_ba[*b][*a] * W[*b][*a][*p][*r];
}
}
}

    }
for(*p = 0; *p < *P; *p = *p + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
OBJ- = TC_ai[*a][*i] * X[*p][*a][*i][*r];

    }
}
}
}
```

```
$for(*p = 0; *p < *P; *p = *p + 1)\{$
$for(*a = 0; *a < *A; *a = *a + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$for(*r = 0; *r < *T; *r = *r + 1)\{$
$OBJ- = HL_a[*a] * X[*p][*a][*i][*r];$
$\}$
$\}$
$\}$
$\}$

    $for(*p = 0; *p < *P; *p = *p + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$for(*r = 0; *r < *T; *r = *r + 1)\{$
$for(*t = 0; *t < *T; *t = *t + 1)\{$
$if(*r <= *t)\{$
$OBJ- = H_pi[*p][*i] * (1 - B_prt[*p][*r][*t]) * Ie[*p][*i][*r][*t];$
$\}$
$\}$
$\}$
$\}$
$\}$

    $for(*p = 0; *p < *P; *p = *p + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$for(*r = 0; *r < *T; *r = *r + 1)\{$
$for(*t = 0; *t < *T; *t = *t + 1)\{$
$if(*r <= *t)\{$
$OBJ- = *E * B_prt[*p][*r][*t] * Ie[*p][*i][*r][*t];$
$\}$
}
}
}
}

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){

    OBJ- = *L * Lk[*p][*i][*t];
}
}
}

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){

    OBJ- = *FO * Y[*p][*i][*t];
}
}
}

    MODEL1.add(IloMaximize(env, OBJ));

    IloExprC1(env);

    for(*a = 0; *a < *A; *a = *a + 1){
```

```
    for(*b = 0; *b < *B; *b = *b + 1){
C1+ = O[*b][*a];
}
sprintf(varName, ”C1.a%d”, *a + 1);
C1.setName(varName);
IloConstraintc(C1 == 1);
c.setName(varName);
MODEL1.add(c);
C1.end();
C1 = IloExpr(env);
}

    IloExprC2(env);

    for(*i = 0; *i < *I; *i = *i + 1){

    for(*a = 0; *a < *A; *a = *a + 1){
C2+ = Z[*a][*i];
}
sprintf(varName, ”C2.i%d”, *i + 1);
C2.setName(varName);
IloConstraintc(C2 == 1);
c.setName(varName);
MODEL1.add(c);
C2.end();
C2 = IloExpr(env);
}

    IloExprC3(env);
```

```
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*r == *t){
C3 = Ib[*p][*i][*r][*t];
sprintf(varName, ”C3.p%di%dr%dt%d”, *p + 1, *i + 1, *r + 1, *t + 1);
C3.setName(varName);
IloConstraintc(C3 == 0);
c.setName(varName);
MODEL1.add(c);
C3.end();
C3 = IloExpr(env);
}
}
}
}
}
IloExprC4(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*r == *t){
for(*a = 0; *a < *A; *a = *a + 1){
//cout << ”p” << *p << ”” << ”i” << *i << ”” << ”r” << *r << ”” << ”t” <<
*t << ”” << ”a” << *a << endl;
C4+ = X[*p][*a][*i][*r];
}
```

```
C4-=dl[*p][*i][*r][*t];
sprintf(varName,"C4.p%di%dr%dt%d",*p+1,*i+1,*r+1,*t+1);
C4.setName(varName);
IloConstraintc(C4==Ie[*p][*i][*r][*t]);
c.setName(varName);
MODEL1.add(c);
C4.end();
C4=IloExpr(env);
}
}
}
}
}

    IloExprC5(env);
for(*p=0;*p<*P;*p=*p+1){
for(*i=0;*i<*I;*i=*i+1){
for(*r=0;*r<*T;*r=*r+1){
for(*t=0;*t<*T;*t=*t+1){
if(*r<*t){

    for(*a=0;*a<*A;*a=*a+1){
C5+=X[*p][*a][*i][*r];
}
for(*e=*r;*e<=*t;*e=*e+1){
C5-=dl[*p][*i][*r][*e];
}
for(*e=*r;*e<*t;*e=*e+1){
C5-=B_prt[*p][*r][*e]*Ie[*p][*i][*r][*e];
}

    sprintf(varName,"C5.p%di%dr%dt%d", *p + 1, *i + 1, *r + 1, *t + 1);
C5.setName(varName);
IloConstraintc(C5 == Ie[*p][*i][*r][*t]);
c.setName(varName);
MODEL1.add(c);
C5.end();
C5 = IloExpr(env);
}
}
}
}
}

    IloExprC6(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*r < *t){
for(*a = 0; *a < *A; *a = *a + 1){
C6+ = X[*p][*a][*i][*r];
}
for(*e = *r; *e < *t;  *e = *e + 1){
C6− = (dl[*p][*i][*r][*e] + B_prt[*p][*r][*e] * Ie[*p][*i][*r][*e]);
}
sprintf(varName,"C6.p%di%dr%dt%d", *p + 1, *i + 1, *r + 1, *t + 1);
C6.setName(varName);
```

```
IloConstraintc(C6 == Ib[*p][*i][*r][*t]);
c.setName(varName);
MODEL1.add(c);
C6.end();
C6 = IloExpr(env);

    }
}
}
}
}

    IloExprC7(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*r = 0; *r <= *t; *r = *r + 1){
C7+ = (dl[*p][*i][*r][*t]);
}
C7+ = Lk[*p][*i][*t];
sprintf(varName, ”C7.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C7.setName(varName);
IloConstraintc(C7 == d_pit[*p][*i][*t]);
c.setName(varName);
MODEL1.add(c);
C7.end();
C7 = IloExpr(env);

    }
```

```
}
}

    IloExprC8(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*r = 0; *r <= *t; *r = *r + 1){
C8+ = Ib[*p][*i][*r][*t];
}
for(*a = 0; *a < *A; *a = *a + 1){
for(*e = *t; *e <= *t + *LT − 1; *e = *e + 1){
C8+ = X[*p][*a][*i][*e];
}
}
C8− = *M * (1 − Y[*p][*i][*t]);
sprintf(varName, ”C8.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C8.setName(varName);
IloConstraintc(C8 <= (SL[*p][*i] − *c));
c.setName(varName);
MODEL1.add(c);
C8.end();
C8 = IloExpr(env);
}
}
}

    IloExprC9(env);
for(*p = 0; *p < *P; *p = *p + 1){
```

```
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*r = 0; *r <= *t; *r = *r + 1){
C9+ = Ib[*p][*i][*r][*t];
}
for(*a = 0; *a < *A; *a = *a + 1){
for(*e = *t; *e <= *t + *LT − 1; *e = *e + 1){

    C9+ = X[*p][*a][*i][*e];
}
}
C9+ = *M * Y[*p][*i][*t];
sprintf(varName, ”C9.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C9.setName(varName);
IloConstraintc(C9 >= (SL[*p][*i] − *c));
c.setName(varName);
MODEL1.add(c);
C9.end();
C9 = IloExpr(env);
}
}
}

    IloExprC10(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T − 1; *t = *t + 1){//checkwithwalaa
C10 = X[*p][*a][*i][*t + *LT];
sprintf(varName,”C10.p%da%di%dt%d”, *p + 1, *a + 1, *i + 1, *t + 1);
C10.setName(varName);
IloConstraintc(C10 <= *M * Y[*p][*i][*t]);
c.setName(varName);
MODEL1.add(c);
C10.end();
C10 = IloExpr(env);
}
}
}
}

   IloExprC11(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T − 1; *t = *t + 1){//checkwithwalaa
for(*a = 0; *a < *A; *a = *a + 1){
C11+ = X[*p][*a][*i][*t + *LT];
}
C11+ = *M * (1 − Y[*p][*i][*t]);
for(*r = 0; *r <= *t; *r = *r + 1){
C11+ = Ib[*p][*i][*r][*t];
}
for(*a = 0; *a < *A; *a = *a + 1){
for(*e = *t; *e <= *t + *LT − 1; *e = *e + 1){
C11+ = X[*p][*a][*i][*e];
}
}
sprintf(varName,”C11.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
```

```
C11.setName(varName);
IloConstraintc(C11 >= SU[*p][*i]);
c.setName(varName);
MODEL1.add(c);
C11.end();
C11 = IloExpr(env);
}
}
}

    IloExprC12(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T − 1; *t = *t + 1){
for(*a = 0; *a < *A; *a = *a + 1){
C12+ = X[*p][*a][*i][*t + *LT];
}
for(*r = 0; *r <= *t; *r = *r + 1){
C12+ = Ib[*p][*i][*r][*t];
}
for(*a = 0; *a < *A; *a = *a + 1){
for(*e = *t; *e <= *t + *LT − 1; *e = *e + 1){
C12+ = X[*p][*a][*i][*e];
}
}
sprintf(varName, ”C12.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C12.setName(varName);
IloConstraintc(C12 <= SU[*p][*i]);
c.setName(varName);
```

```
MODEL1.add(c);
C12.end();
C12 = IloExpr(env);
}
}
}

    IloExprC13(env);
for(*a = 0; *a < *A; *a = *a + 1){
for(*t = 0; *t < *T; *t = *t + 1){

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
C13+ = X[*p][*a][*i][*t];
}
}
sprintf(varName, ”C13.a%dt%d”, *a + 1, *t + 1);
C13.setName(varName);
IloConstraintc(C13 <= Q_at[*a][*t]);
c.setName(varName);
MODEL1.add(c);
C13.end();
C13 = IloExpr(env);
}
}

    IloExprC14(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*a = 0; *a < *A; *a = *a + 1){
```

```
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
C14 = X[*p][*a][*i][*t];
sprintf(varName, ”C14.p%da%di%dt%d”, *p + 1, *a + 1, *i + 1, *t + 1);
C14.setName(varName);
IloConstraintc(C14 <= *M * Z[*a][*i]);
c.setName(varName);
MODEL1.add(c);
C14.end();
C14 = IloExpr(env);
}
}
}
}

    IloExprC15(env);
for(*b = 0; *b < *B; *b = *b + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*p = 0; *p < *P; *p = *p + 1){
C15 = W[*b][*a][*p][*t];
}
sprintf(varName, ”C15.b%da%dt%d”, *b + 1, *a + 1, *t + 1);
C15.setName(varName);
IloConstraintc(C15 <= *M * O[*b][*a]);
c.setName(varName);
MODEL1.add(c);
C15.end();
C15 = IloExpr(env);
```

```
}
}
}

    IloExprC16(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*r = *t; *r <= *t; *r = *r + 1){
C16+ = Ie[*p][*i][*r][*t];
}
sprintf(varName, ”C16.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C16.setName(varName);
IloConstraintc(C16 <= *M * U[*p][*i][*t]);
c.setName(varName);
MODEL1.add(c);
C16.end();
C16 = IloExpr(env);
}
}
}

    IloExprC17(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*r = *t; *r <= *t; *r = *r + 1){
C17+ = Ie[*p][*i][*r][*t];
}
```

```
sprintf(varName,”C17.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C17.setName(varName);
IloConstraintc(C17 >= *c * U[*p][*i][*t]);
c.setName(varName);
MODEL1.add(c);
C17.end();
C17 = IloExpr(env);
}
}
}

    IloExprC18(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
C18+ = Lk[*p][*i][*t];
sprintf(varName,”C18.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C18.setName(varName);
IloConstraintc(C18 <= *M * (1 − U[*p][*i][*t]));
c.setName(varName);
MODEL1.add(c);
C18.end();
C18 = IloExpr(env);
}
}
}

    IloExprC19(env);
for(*p = 0; *p < *P; *p = *p + 1){
```

```
for(*a = 0; *a < *A; *a = *a + 1){
for(*t = 0; *t < *T; *t = *t + 1){


    for(*b = 0; *b < *B; *b = *b + 1){
C19+ = W[*b][*a][*p][*t];
}


    for(*i = 0; *i < *I; *i = *i + 1){
/ * cout << Y_p[*p] << endl; */
C19− = Y_p[*p] * X[*p][*a][*i][*t];
}
sprintf(varName, ”C19.p%da%dt%d”, *p + 1, *a + 1, *t + 1);
C19.setName(varName);
IloConstraintc(C19 == 0);
c.setName(varName);
MODEL1.add(c);
C19.end();
C19 = IloExpr(env);
}
}
}


    IloExprC20(env);
for(*b = 0; *b < *B; *b = *b + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*p = 0; *p < *P; *p = *p + 1){
C20+ = W[*b][*a][*p][*t];
}
```

```
}
C20- = C_pt[*b][*t];
sprintf(varName,"C20.b%d
t%d",*b+1,*t+1);
C20.setName(varName);
IloConstraintc(C20 <= 0);
c.setName(varName);
MODEL1.add(c);
C20.end();
C20 = IloExpr(env);
}
}

    IloExprC21(env);
for(*p = 0;*p < *P;*p = *p+1){
for(*i = 0;*i < *I;*i = *i+1){
for(*t = 0;*t < *T;*t = *t+1){
if(*t <= *LT){
for(*a = 0;*a < *A;*a = *a+1){
C21+ = X[*p][*a][*i][*t];
}
sprintf(varName,"C21.p%di%dt%d",*p+1,*i+1,*t+1);
C21.setName(varName);
IloConstraintc(C21 == Ib[*p][*i][0][*t]);
c.setName(varName);
/*MODEL1.add(c);*/
C21.end();
C21 = IloExpr(env);
```

```
    }
}
}
}

    IloCplexcplex(MODEL1);

    /*cplex.setParam(IloCplex::TiLim,60);*/

    /*cplex.setOut(env.getNullStream());*/
cplex.extract(MODEL1);
/*cplex.exportModel(”MODELL.alp”);*/
/*cplex.exportModel(”MODELL.RLP”);*/
cplex.exportModel(”MODELL.lp”);
/*cplex.exportModel(”MODELL.sav”);*/

    cplex.setParam(IloCplex::Param::Emphasis::MIP,4);
/*cplex.setParam(IloCplex::Param::MIP::Limits::Nodes,5000);*/

    cplex.solve();

    //Createafilestreamobjectandcheckthatitopenedcorrectly
ofstreamoutFile(”Result.csv”);
if(!outFile.is_open())
{
std::cerr<<”Failedtoopenoutputfile”<<std::endl;
}

    float*val=newfloat(0);
```

$outFile << cplex.getObjValue() << endl;$

$end = clock();$

$outFile << " = 1ifabattoir/slaughterhouseaisallocatedtobreeding/farmcenterb, otherwise0" << endl;$

$for(*b = 0; *b < *B; *b = *b + 1)\{$
$for(*a = 0; *a < *A; *a = *a + 1)\{$
$*val = cplex.getValue(O[*b][*a]);$
$outFile << "O" << "b" << *b + 1 << "" << "a" << *a + 1 << "," << *val << endl;$
$\}$
$\}$

$outFile << endl;$

$outFile << " = 1ifretailer/warehouseiisallocatedtoabattoir/slaughterhousea, otherwise0" << endl;$

$for(*a = 0; *a < *A; *a = *a + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$*val = cplex.getValue(Z[*a][*i]);$
$outFile << "Z" << "a" << *a + 1 << "" << "i" << *i + 1 << "," << *val << endl;$
$\}$
$\}$
$outFile << endl;$

$outFile << "Amountofproductptransportedtoretailerifromabattoiraattimeperiodt" <<$

```
endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
*val = cplex.getValue(X[*p][*a][*i][*t]);
outFile << ”X” << ”p” << *p+1 << ”” << ”a” << *a+1 << ”” << ”i” << *i+1 <<
”” << ”t” << *t + 1 << ”,” << *val << endl;
}
}
}
}

    outFile << endl;
outFile << ” = 1ifretailer/warehouseimakesanorderforproductpatthebeginningoftimeperiodt, otherw
endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
*val = cplex.getValue(Y[*p][*i][*t]);
outFile << ”Y” << ”p” << *p+1 << ”” << ”i” << *i+1 << ”” << ”t” << *t+1 <<
”,” << *val << endl;
}
}
}

    outFile << endl;
```

$outFile << " = 1whentheon-handinventoryinretailer/warehousei'ssiteforproductpisabovezeroatth$
$endl;$

$for(*p = 0; *p < *P; *p = *p + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$for(*t = 0; *t < *T; *t = *t + 1)\{$
$*val = cplex.getValue(U[*p][*i][*t]);$
$outFile << "U" << "p" << *p + 1 << "" << "i" << *i + 1 << "" << "t" << *t + 1 <<$
$"," << *val << endl;$
$\}$
$\}$
$\}$

$outFile << endl;$
$outFile << "Reorderpointforproductpintheretailer/warehousei" << endl;$

$for(*p = 0; *p < *P; *p = *p + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$*val = cplex.getValue(SL[*p][*i]);$
$outFile << "SL" << "p" << *p + 1 << "" << "i" << *i + 1 << "," << *val << endl;$
$\}$
$\}$

$outFile << endl;$
$outFile << "Order - up - tolevelforproductpintheretailer/warehousei" << endl;$

$for(*p = 0; *p < *P; *p = *p + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$

```
*val = cplex.getValue(SU[*p][*i]);
outFile << ”SU” << ”p” << *p + 1 << ”” << ”i” << *i + 1 << ”,” << *val << endl;
}
}

    outFile << endl;
outFile << ”Amountofproductpdeliveredtocustomersinretailer/warehousesiteiattimeperiodt, whichisp
endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*r <= *t){
*val = cplex.getValue(dl[*p][*i][*r][*t]);
outFile << ”dl” << ”p” << *p+1 << ”” << ”i” << *i+1 << ”” << ”r” << *r+1 <<
”” << ”t” << *t + 1 << ”,” << *val << endl;
}
}
}
}
}

    outFile << endl;
outFile << ”Numberoflivestocktransportedtoabattoirafrombreedingcenterbattimeperiodt” <<
endl;

    for(*b = 0; *b < *B; *b = *b + 1){
for(*a = 0; *a < *A; *a = *a + 1){
```

```
for(*p = 0; *p < *P; *p = *p + 1){
for(*t = 0; *t < *T; *t = *t + 1){
*val = cplex.getValue(W[*b][*a][*p][*t]);
outFile << ”W” << ”b” << *b + 1 << ”” << ”a” << *a + 1 << ”” << ”p” <<
*p + 1 << ”” << ”t” << *t + 1 << ”,” << *val << endl;
}
}
}
}

    outFile << endl;
outFile << ”Retaileri′son−handinventoryofproductpatthebeginningoftimeperiodt, producedintimeperi
endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*r <= *t){
*val = cplex.getValue(Ib[*p][*i][*r][*t]);
outFile << ”Ib” << ”p” << *p+1 << ”” << ”i” << *i+1 << ”” << ”r” << *r+1 <<
”” << ”t” << *t + 1 << ”,” << *val << endl;
}
}
}
}
}

    outFile << endl;
outFile << ”Retaileri′son−handinventoryofproductpattheendoftimeperiodt, producedintimeperiodr” <
endl;

    for(∗p = 0; ∗p < ∗P; ∗p = ∗p + 1){
for(∗i = 0; ∗i < ∗I; ∗i = ∗i + 1){
for(∗r = 0; ∗r < ∗T; ∗r = ∗r + 1){
for(∗t = 0; ∗t < ∗T; ∗t = ∗t + 1){
if(∗r <= ∗t){
∗val = cplex.getValue(Ie[∗p][∗i][∗r][∗t]);
outFile << ”Ie” << ”p” << ∗p+1 << ”” << ”i” << ∗i+1 << ”” << ”r” << ∗r+1 <<
”” << ”t” << ∗t + 1 << ”,” << ∗val << endl;
}
}
}
}
}

    outFile << endl;
outFile << ”Lost − salequantityofproductpfordemandintimeperiodtandretaileri.” <<
endl;

    for(∗p = 0; ∗p < ∗P; ∗p = ∗p + 1){
for(∗i = 0; ∗i < ∗I; ∗i = ∗i + 1){
for(∗t = 0; ∗t < ∗T; ∗t = ∗t + 1){
∗val = cplex.getValue(Lk[∗p][∗i][∗t]);
outFile << ”LK” << ”p” << ∗p + 1 << ”” << ”i” << ∗i + 1 << ”” << ”t” <<
∗t + 1 << ”,” << ∗val << endl;
}
}
}

    //closethefile
outFile.close();

    }
catch(IloException&ex){
cerr << ”Error : ” << ex << endl;
}
catch(...){
cerr << ”Error” << endl;
}

    }

    voidfreeData(){

    template < typenameT >
T * *AllocateDynamic2DArrayCSTYLE(intnRows, intnCols){

    int * m = newint(0);

    T * *dynamicArray;

    dynamicArray = (T * *)calloc(nRows, sizeof(T*));
for(*m = 0; *m < nRows; *m = *m + 1)
{
dynamicArray[*m] = (T*)calloc(nCols, sizeof(T));
}
```

$return(dynamicArray);$

$free(m);$

$\}$

$template < typenameT >$

$voidFreeDynamic2DArrayCSTYLE(T * *dArray, intnRows, intnCols)\{$

$int * m = newint(0);$

$for(*m = 0; *m < nRows; *m = *m + 1)$

$\{$

$free(dArray[*m]);$

$\}$

$free(dArray);$

$free(m);$

$\}$

$int * create_{i}nt_{v}ector(intdim)$

$\{$

$int * ptr;$

$if((ptr = (int*)calloc(dim, sizeof(int))) == NULL)\{$

$printf(\text{”}Error : Memoriainsuficiente\text{”});$

$exit(8);$

$\}$

$returnptr;$

$free(ptr);$

$\}$

$longdouble * create_double_vector(intdim)$

$\{$

$longdouble * ptr;$

$if((ptr = (longdouble*)calloc(dim, sizeof(longdouble))) == NULL)\{$

$printf(" : Memoriainsuficiente");$

$exit(8);$

$\}$

$returnptr;$

$free(ptr);$

$\}$

$template < typenameT >$

$T * * * AllocateDynamic3DArrayCSTYLE(intnRows, intnCols, intnDep)\{$

$int * m = newint(0);$

$int * i = newint(0);$

$T * * * dynamicArray;$

$dynamicArray = (T * **)calloc(nRows, sizeof(T * *));$

$for(*m = 0; *m < nRows; *m = *m + 1)$

$\{$

$dynamicArray[*m] = (T * *)calloc(nCols, sizeof(T*));$

$for(*i = 0; *i < nCols; *i = *i + 1)$

$\{$

$dynamicArray[*m][*i] = (T*)calloc(nDep, sizeof(T));$

$\}$

$\}$

$return(dynamicArray);$

$free(m);$

$free(i);$

$\}$

$template < typenameT >$

$voidFreeDynamic3DArrayCSTYLE(T * * * dArray, intnRows, intnCols, intnDep)\{$

$int * m = newint(0);$

$int * i = newint(0);$

$for(*m = 0; *m < nRows; *m = *m + 1)$

$\{$

$for(*i = 0; *i < nCols; *i = *i + 1)$

$\{$

$free(dArray[*m][*i]);$

$\}$

$free(dArray[*m]);$

$\}$

$free(dArray);$

$free(m);$

$free(i);$

$\}$

$template < typenameT >$

$T * AllocateDynamic1DArrayCSTYLE(intnRows)\{$

$T * dynamicArray;$

$dynamicArray = (T*)calloc(nRows, sizeof(T));$

$return(dynamicArray);$

$\}$

$template < typenameT >$

$voidFreeDynamic1DArrayCSTYLE(T * dArray, intnRows)\{$

$free(dArray);$

$\}$

Appendix D

D.1 Phase 2 model code in Concert Technology in CPLEX:

$\#include < ilcplex/ilocplex.h >$
$ILOSTLBEGIN$

$\#include < vector >$
$\#include < fstream >$
$\#include < iostream >$

$int * create_int_vector(intdim);$
$longdouble * create_double_vector(intdim);$

$template < typenameT >$
$T * * * AllocateDynamic3DArrayCSTYLE(intnRows, intnCols, intnDep);$
$template < typenameT >$
$voidFreeDynamic3DArrayCSTYLE(T * * * dArray, intnRows, intnCols, intnDep);$
$template < typenameT >$
$T * AllocateDynamic1DArrayCSTYLE(intnRows);$
$template < typenameT >$
$voidFreeDynamic1DArrayCSTYLE(T * dArray, intnRows);$
$template < typenameT >$

$T * *AllocateDynamic2DArrayCSTYLE(intnRows, intnCols);$

$template < typenameT >$

$voidFreeDynamic2DArrayCSTYLE(T * *dArray, intnRows, intnCols);$

$//WriteaMIPModelfortheinformationgivenbelow$

$int * N = newint(4);$

$int * C = newint(6);$

$int * S = newint(9);$

$int * K = newint(5);$

$int * O_k;$

$int * D_k;$

$int * C_s;$

$int * W_c;$

$int * *Lamda_cs;$

$int * O_s;$

$int * D_s;$

$double * Tau_s;$

$double * Tar_s;$

$double * *Tdu_kc;$

$double * F_s;$

$double * *T_cs;$

$int * *d_kc;$

$int * Big_Num = newint(100000);$

$voidreadData();$

$voidfreeData();$

$voidCall_Optimizaer();$

$ofstreamoutput(''Result.csv'');$

$intmain(intargc, char**argv)$

$\{$

$readData();$

$Call_Optimizaer();$

$freeData();$

$return0;$

$\}$

$voidCall_Optimizaer()\{$

$IloEnvenv;$

$/*IloNumstartTime;*/$

$try\{$

$longdouble*maxErrDefault = newlongdouble(1e-6);$

$time_tstart, end;$

$start = clock();$

$IloModelMODEL(env);$

$int*i = newint(0);$

$int * j = newint(0);$
$int * m = newint(0);$
$int * z = newint(0);$

$double * total_time = newdouble(0);$

$charvarName[100];$

$IloArray < IloArray < IloBoolVarArray >> X(env, *K);$

$for(*i = 0; *i < *K; *i = *i + 1)\{$
$X[*i] = IloArray < IloBoolVarArray > (env, *C);$
$for(*j = 0; *j < *C; *j = *j + 1)\{$
$X[*i][*j] = IloBoolVarArray(env, *S);$
$\}$
$\}$

$for(*i = 0; *i < *K; *i = *i + 1)\{$
$for(*j = 0; *j < *C; *j = *j + 1)\{$
$for(*m = 0; *m < *S; *m = *m + 1)\{$
$sprintf(varName, ”Xk\%dc\%ds\%d”, *i + 1, *j + 1, *m + 1);$
$X[*i][*j][*m].setName(varName);$
$\}$
$\}$
$\}$

$IloBoolVarArrayY(env, *S);$

$for(*i = 0; *i < *S; *i = *i + 1)\{$

$sprintf(varName, ”Ys\%d”, *i + 1);$

$Y[*i].setName(varName);$

$\}$

$IloExprOBJ(env);$

$for(*i = 0; *i < *K; *i = *i + 1)\{$

$for(*j = 0; *j < *C; *j = *j + 1)\{$

$for(*m = 0; *m < *S; *m = *m + 1)\{$

$OBJ+ = T_cs[*j][*m] * X[*i][*j][*m];$

$\}$

$\}$

$\}$

$for(*i = 0; *i < *S; *i = *i + 1)\{$

$OBJ+ = F_s[*i] * Y[*i];$

$\}$

$MODEL.add(IloMinimize(env, OBJ));$

$IloExprC1_1(env);$

$for(*i = 0; *i < *K; *i = *i + 1)\{$

$for(*j = 0; *j < *C; *j = *j + 1)\{$

$for(*m = 0; *m < *N; *m = *m + 1)\{$

$if(*m + 1 == O_k[*i])\{$

$for(*z = 0; *z < *S; *z = *z + 1)\{$

$if(O_s[*z] == *m + 1)\{$

$C1_1+ = d_kc[*i][*j] * Lamda_cs[*j][*z] * X[*i][*j][*z];$

$\}$

$if(D_s[*z] == *m + 1)\{$

$C1_1- = d_kc[*i][*j] * Lamda_cs[*j][*z] * X[*i][*j][*z];$

$\}$

$\}$

$sprintf(varName, "C1_1.k\%dc\%dn\%d", *i + 1, *j + 1, *m + 1);$

$C1_1.setName(varName);$

$IloConstraintc(C1_1 == 1 * d_kc[*i][*j]);$

$c.setName(varName);$

$MODEL.add(c); for(*i = 0; *i < *I; *i = *i + 1)\{$

$C1_1.end(); cout << G_ai[*a][*i] << "";$

$C1_1 = IloExpr(env); \}$

$\}\}$

$\}$

$\}$

$\}$

$IloExprC1_2(env);$

$for(*i = 0; *i < *K; *i = *i + 1)\{$

$for(*j = 0; *j < *C; *j = *j + 1)\{$

$for(*m = 0; *m < *N; *m = *m + 1)\{$

$if(*m + 1 == D_k[*i])\{$

$for(*z = 0; *z < *S; *z = *z + 1)\{$

$if(O_s[*z] == *m + 1)\{$

$C1_2+ = d_kc[*i][*j] * Lamda_cs[*j][*z] * X[*i][*j][*z];$

$\}$

$if(D_s[*z] == *m + 1)\{$

$C1_2 -= d_kc[*i][*j] * Lamda_cs[*j][*z] * X[*i][*j][*z];$

}

}

$sprintf(varName, ”C1_2.k\%dc\%dn\%d”, *i+1, *j+1, *m+1);$

$C1_2.setName(varName);$

$IloConstraintc(C1_2 == -1 * d_kc[*i][*j]);$

$c.setName(varName);$

$MODEL.add(c);$

$C1_2.end();$

$C1_2 = IloExpr(env);$

}

}

}

}

$IloExprC1_3(env);$

$for(*i = 0; *i < *K; *i = *i + 1)\{$

$for(*j = 0; *j < *C; *j = *j + 1)\{$

$for(*m = 0; *m < *N; *m = *m + 1)\{$

$if(*m + 1! = D_k[*i]\&\& * m + 1! = O_k[*i])\{$

$for(*z = 0; *z < *S; *z = *z + 1)\{cout << UC_pb[*p][*b] << ””;$

$if(O_s[*z] == *m + 1)\{\}$

$C1_3 += d_kc[*i][*j] * Lamda_cs[*j][*z] * X[*i][*j][*z]; cout << endl;$

}

$if(D_s[*z] == *m + 1)\{$

$C1_3 -= d_kc[*i][*j] * Lamda_cs[*j][*z] * X[*i][*j][*z];$

}

```
}
sprintf(varName, ”C1_3.k%dc%dn%d”, *i + 1, *j + 1, *m + 1);
C1_2.setName(varName);
IloConstraintc(C1_3 == 0);
c.setName(varName);
MODEL.add(c);
C1_2.end();
C1_2 = IloExpr(env);

    }
}
}
}

    IloExprC2(env);
for(*i = 0; *i < *S; *i = *i + 1){

    for(*j = 0; *j < *K; *j = *j + 1){
for(*m = 0; *m < *C; *m = *m + 1){
C2+ = W_c[*m] * d_kc[*j][*m] * X[*j][*m][*i];
}

    C2.setName(varName); cout << H_pi[*p][*i] << ””;
IloConstraintc(C2 <= C_s[*i] * Y[*i]); }
c.setName(varName); cout << endl;
MODEL.add(c);
C2.end();
C2 = IloExpr(env);
```

```
    }

    IloExprC3(env);
for(*i = 0; *i < *K; *i = *i + 1){
for(*j = 0; *j < *C; *j = *j + 1){
for(*m = 0; *m < *N; *m = *m + 1){
if(*m + 1 == O_k[*i]){

    for(*z = 0; *z < *S; *z = *z + 1){
if(O_s[*z] == *m + 1){
C3+ = Tau_s[*z] * Lamda_cs[*j][*z] * X[*i][*j][*z];
}
}
sprintf(varName, ”C3.k%dc%dn%d”, *i + 1, *j + 1, *m + 1);
C3.setName(varName);
IloConstraintc(C3 >= 0);
c.setName(varName);
MODEL.add(c);
C3.end();
C3 = IloExpr(env);

    }
}
}
}

    IloExprC4(env);
for(*i = 0; *i < *K; *i = *i + 1){
for(*j = 0; *j < *C; *j = *j + 1){
```

```
for(*m = 0; *m < *N; *m = *m + 1){
if(*m + 1 == D_k[*i]){

    for(*z = 0; *z < *S; *z = *z + 1){
if(D_s[*z] == *m + 1){
C3+ = Tar_s[*z] * Lamda_cs[*j][*z] * X[*i][*j][*z];
}
}
sprintf(varName, ”C4.k%dc%dn%d”, *i + 1, *j + 1, *m + 1);
C4.setName(varName);
IloConstraintc(C4 <= Tdu_kc[*i][*j]);
c.setName(varName);
MODEL.add(c);
C4.end();
C4 = IloExpr(env);

    }
}
}
}

    IloCplexcplex(MODEL);

    /* cplex.setParam(IloCplex :: TiLim, 60); */

    /* cplex.setOut(env.getNullStream()); */
cplex.extract(MODEL);
/* cplex.exportModel(”MODELL.alp”); */
/* cplex.exportModel(”MODELL.RLP”); */
```

```
cplex.exportModel(”MODELL.lp”);
/ * cplex.exportModel(”MODELL.sav”); */

    cplex.setParam(IloCplex :: Param :: Emphasis :: MIP, 4);
/ * cplex.setParam(IloCplex :: Param :: MIP :: Limits :: Nodes, 5000); */

    cplex.solve();

    output << cplex.getObjValue() << endl;

    end = clock();

    for(*i = 0; *i < *K; *i = *i + 1){
for(*j = 0; *j < *C; *j = *j + 1){
for(*m = 0; *m < *S; *m = *m + 1){
if(cplex.getValue(X[*i][*j][*m]) == 1){
output << *i + 1 << ”” << *j + 1 << ”” << *m + 1 << ”” << ”” << endl;
}
}
}
}

    output << endl;

    for(*i = 0; *i < *S; *i = *i + 1){
if(cplex.getValue(Y[*i]) == 1){
output << *i + 1 << endl;
}
}
```

```
output << endl;

    for(*i = 0; *i < *K; *i = *i + 1){
for(*j = 0; *j < *C; *j = *j + 1){
for(*m = 0; *m < *S; *m = *m + 1){
if(cplex.getValue(X[*i][*j][*m]) == 1){

    output << *i + 1 << endl;
}
}
}

    env.end();

    free(i);
free(j);
free(m);

    }

    catch(IloException&ex){
cerr << ”Error : ” << ex << endl;
}
catch(...){
cerr << ”Error” << endl;
}
}for(*a = 0; *a < *A; *a = *a + 1){

    voidreadData(){cout << Q_at[*a][*t] << ””;
```

$int * i = newint(0); \}$
$int * j = newint(0);$
$int * k = newint(0);$

$O_k = AllocateDynamic1DArrayCSTYLE < int > (*K);$
$D_k = AllocateDynamic1DArrayCSTYLE < int > (*K);$

$C_s = AllocateDynamic1DArrayCSTYLE < int > (*S);$
$W_c = AllocateDynamic1DArrayCSTYLE < int > (*C);$

$Lamda_cs = AllocateDynamic2DArrayCSTYLE < int > (*C, *S);$

$O_s = AllocateDynamic1DArrayCSTYLE < int > (*S);$
$D_s = AllocateDynamic1DArrayCSTYLE < int > (*S);$

$Tau_s = AllocateDynamic1DArrayCSTYLE < double > (*S);$
$Tar_s = AllocateDynamic1DArrayCSTYLE < double > (*S);$

$Tdu_kc = AllocateDynamic2DArrayCSTYLE < double > (*K, *C);$

$F_s = AllocateDynamic1DArrayCSTYLE < double > (*S);$

$T_cs = AllocateDynamic2DArrayCSTYLE < double > (*C, *S);$
$d_kc = AllocateDynamic2DArrayCSTYLE < int > (*K, *C);$

$/ * readdata * /$
$ifstreamin(”Data_Set.txt”);$

$in >> *N;$

$in >> *C;$

$in >> *S;$

$in >> *K;$

$cout << ''ISNTANCE : '' << '''' << *N << '''' << *C << '''' << *S << '''' << *K << endl;$

$for(*i = 0; *i < *K; *i = *i + 1)\{$

$in >> O_k[*i] >> D_k[*i];$

$\}$

$/ * for(*i = 0; *i < *K; *i = *i + 1)\{$

$cout << O_k[*i] << '''' << D_k[*i] << endl;$

$\} * /$

$for(*i = 0; *i < *S; *i = *i + 1)\{$

$in >> C_s[*i];$

$\}$

$/ * for(*i = 0; *i < *S; *i = *i + 1)\{$

$cout << C_s[*i] << endl;$

$\}$

$*/$

$for(*i = 0; *i < *C; *i = *i + 1)\{$

$in >> W_c[*i];$

$\}$

$/ * for(*i = 0; *i < *C; *i = *i + 1)\{$

$cout << W_c[*i] << endl;$

```
}*/

    for(*i = 0; *i < *C; *i = *i + 1){
for(*j = 0; *j < *S; *j = *j + 1){
in >> Lamda_cs[*i][*j];
}
}

    /*for(*i = 0; *i < *C; *i = *i + 1){
for(*j = 0; *j < *S; *j = *j + 1){
cout << Lamda_cs[*i][*j] << ””;
}
cout << endl; */

    for(*i = 0; *i < *S; *i = *i + 1){
in >> O_s[*i] >> D_s[*i];
}
/*for(*i = 0; *i < *S; *i = *i + 1){
cout << O_s[*i] << ”” << D_s[*i];
cout << endl;
}*/

    for(*i = 0; *i < *S; *i = *i + 1){
in >> Tau_s[*i] >> Tar_s[*i];
}
/*for(*i = 0; *i < *S; *i = *i + 1){
cout << Tau_s[*i] << ”” << Tar_s[*i];
cout << endl;
}*/
```

```
    for(*i = 0; *i < *K; *i = *i + 1){
for(*j = 0; *j < *C; *j = *j + 1){
in >> Tdu_kc[*i][*j];
}
}
/ * for(*i = 0; *i < *K; *i = *i + 1){
for(*j = 0; *j < *C; *j = *j + 1){
cout << Tdu_kc[*i][*j] << ””;
}
cout << endl;
} * /

    for(*i = 0; *i < *S; *i = *i + 1){
in >> F_s[*i];
}
/ * for(*i = 0; *i < *S; *i = *i + 1){
cout << F_s[*i] << endl;
} * /

    for(*i = 0; *i < *C; *i = *i + 1){
for(*j = 0; *j < *S; *j = *j + 1){
in >> T_cs[*i][*j];
}
}

    / * for(*i = 0; *i < *C; *i = *i + 1){
for(*j = 0; *j < *S; *j = *j + 1){
cout << T_cs[*i][*j] << ””;
```

$\}$
$cout << endl;$
$\} * /$

$for(*i = 0; *i < *K; *i = *i + 1)\{$
$for(*j = 0; *j < *C; *j = *j + 1)\{$
$in >> d_kc[*i][*j];$
$\}$
$\}$

$/ * for(*i = 0; *i < *K; *i = *i + 1)\{$
$for(*j = 0; *j < *C; *j = *j + 1)\{$
$cout << d_kc[*i][*j] << "";$
$\}$
$cout << endl;$
$\} * /$
$\}$

$voidfreeData()\{$

$FreeDynamic1DArrayCSTYLE(O_k, *K);$
$FreeDynamic1DArrayCSTYLE(D_k, *K);$
$FreeDynamic1DArrayCSTYLE(C_s, *S);$
$FreeDynamic1DArrayCSTYLE(W_c, *C);$
$FreeDynamic2DArrayCSTYLE(Lamda_cs, *C, *S);$
$FreeDynamic1DArrayCSTYLE(O_s, *S);$
$FreeDynamic1DArrayCSTYLE(D_s, *S);$
$FreeDynamic1DArrayCSTYLE(Tau_s, *S);$
$FreeDynamic1DArrayCSTYLE(Tar_s, *S);$

$FreeDynamic2DArrayCSTYLE(Tdu_kc, *K, *C);$

$FreeDynamic1DArrayCSTYLE(F_s, *S);$

$FreeDynamic2DArrayCSTYLE(T_cs, *C, *S);$

$FreeDynamic2DArrayCSTYLE(d_kc, *K, *C);$

$\}$

$template < typenameT >$

$T * *AllocateDynamic2DArrayCSTYLE(intnRows, intnCols)\{$

$int * m = newint(0);$

$T * *dynamicArray;$

$dynamicArray = (T * *)calloc(nRows, sizeof(T*));$

$for(*m = 0; *m < nRows; *m = *m + 1)$

$\{$

$dynamicArray[*m] = (T*)calloc(nCols, sizeof(T));$

$\}$

$return(dynamicArray);$

$free(m);$

$\}$

$template < typenameT >$

$voidFreeDynamic2DArrayCSTYLE(T * *dArray, intnRows, intnCols)\{$

$int * m = newint(0);$

$for(*m=0;*m<nRows;*m=*m+1)$

$\{$

$free(dArray[*m]);$

$\}$

$free(dArray);$

$free(m);$

$\}$

$int*create_int_vector(intdim)$

$\{$

$int*ptr;$

$if((ptr=(int*)calloc(dim,sizeof(int)))==NULL)\{$

$printf(":Memoriainsuficiente");$

$exit(8);$

$\}$

$returnptr;$

$free(ptr);$

$\}$

$longdouble*create_double_vector(intdim)$

$\{$

$longdouble*ptr;$

$if((ptr=(longdouble*)calloc(dim,sizeof(longdouble)))==NULL)\{$

$printf(":Memoriainsuficiente");$

$exit(8);$

$\}$

$returnptr;$

$free(ptr);$

$\}$

$template < typenameT >$

$T *** AllocateDynamic3DArrayCSTYLE(intnRows, intnCols, intnDep)\{$

$int * m = newint(0);$

$int * i = newint(0);$

$T *** dynamicArray;$

$dynamicArray = (T * **)calloc(nRows, sizeof(T * *));$

$for(*m = 0; *m < nRows; *m = *m + 1)$

$\{$

$dynamicArray[*m] = (T * *)calloc(nCols, sizeof(T*));$

$for(*i = 0; *i < nCols; *i = *i + 1)$

$\{$

$dynamicArray[*m][*i] = (T*)calloc(nDep, sizeof(T));$

$\}$

$\}$

$return(dynamicArray);$

$free(m);$

$free(i);$

$\}$

$template < typenameT >$

$voidFreeDynamic3DArrayCSTYLE(T *** dArray, intnRows, intnCols, intnDep)\{$

$int * m = newint(0);$

$int * i = newint(0);$

$for(*m = 0; *m < nRows; *m = *m + 1)$

$\{$

$for(*i = 0; *i < nCols; *i = *i + 1)$

$\{$

$free(dArray[*m][*i]);$

$\}$

$free(dArray[*m]);$

$\}$

$free(dArray);$

$free(m);$

$free(i);$

$\}$

$template < typenameT >$

$T * AllocateDynamic1DArrayCSTYLE(intnRows)\{$

$T * dynamicArray;$

$dynamicArray = (T*)calloc(nRows, sizeof(T));$

$return(dynamicArray);$

$\}$

$template < typenameT >$

$voidFreeDynamic1DArrayCSTYLE(T * dArray, intnRows)\{$

```
free(dArray);
}

    OBJ-=G_ai[*a][*i]*Z[*a][*i];

    }
}

    for(*p=0;*p<*P;*p=*p+1){
for(*b=0;*b<*B;*b=*b+1){
for(*a=0;*a<*A;*a=*a+1){
for(*r=0;*r<*T;*r=*r+1){
OBJ-=UC_pb[*p][*b]*W[*b][*a][*p][*r];
}
}
}
}
for(*p=0;*p<*P;*p=*p+1){
for(*b=0;*b<*B;*b=*b+1){
for(*a=0;*a<*A;*a=*a+1){
for(*r=0;*r<*T;*r=*r+1){
OBJ-=TL_ba[*b][*a]*W[*b][*a][*p][*r];
}
}
}

    }
for(*p=0;*p<*P;*p=*p+1){
for(*a=0;*a<*A;*a=*a+1){
```

```
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$for(*r = 0; *r < *T; *r = *r + 1)\{$
$OBJ- = TC_a i[*a][*i] * X[*p][*a][*i][*r];$

    $\}$
$\}$
$\}$
$\}$
$for(*p = 0; *p < *P; *p = *p + 1)\{$
$for(*a = 0; *a < *A; *a = *a + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$for(*r = 0; *r < *T; *r = *r + 1)\{$
$OBJ- = HL_a[*a] * X[*p][*a][*i][*r];$
$\}$
$\}$
$\}$
$\}$

    $for(*p = 0; *p < *P; *p = *p + 1)\{$
$for(*i = 0; *i < *I; *i = *i + 1)\{$
$for(*r = 0; *r < *T; *r = *r + 1)\{$
$for(*t = 0; *t < *T; *t = *t + 1)\{$
$if(*r <= *t)\{$
$OBJ- = H_p i[*p][*i] * (1 - B_p rt[*p][*r][*t]) * Ie[*p][*i][*r][*t];$
$\}$
$\}$
$\}$
$\}$
$\}$
```

```
    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*r <= *t){
OBJ- = *E * B_prt[*p][*r][*t] * Ie[*p][*i][*r][*t];
}
}
}
}
}

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){

    OBJ- = *L * Lk[*p][*i][*t];
}
}
}

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){

    OBJ- = *FO * Y[*p][*i][*t];
}
}
```

```
}

    MODEL1.add(IloMaximize(env, OBJ));

    IloExprC1(env);

    for(*a = 0; *a < *A; *a = *a + 1){

    for(*b = 0; *b < *B; *b = *b + 1){
C1+ = O[*b][*a];
}
sprintf(varName, ”C1.a%d”, *a + 1);
C1.setName(varName);
IloConstraintc(C1 == 1);
c.setName(varName);
MODEL1.add(c);
C1.end();
C1 = IloExpr(env);
}

    IloExprC2(env);

    for(*i = 0; *i < *I; *i = *i + 1){

    for(*a = 0; *a < *A; *a = *a + 1){
C2+ = Z[*a][*i];
}
sprintf(varName, ”C2.i%d”, *i + 1);
C2.setName(varName);
```

```
IloConstraintc(C2 == 1);
c.setName(varName);
MODEL1.add(c);
C2.end();
C2 = IloExpr(env);
}

    IloExprC3(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*r == *t){
C3 = Ib[*p][*i][*r][*t];
sprintf(varName,”C3.p%diC3.setName(varName);
IloConstraint c(C3 == 0);
c.setName(varName);
MODEL1.add(c);
C3.end();
C3 = IloExpr(env);
}
}
}
}
}
IloExpr C4(env);
for (*p = 0; *p ¡ *P; *p = *p + 1){
for (*i = 0; *i ¡ *I; *i = *i + 1){
for (*r = 0; *r ¡ *T; *r = *r + 1){
```

```
for (*t = 0; *t ¡ *T; *t = *t + 1){
if (*r == *t){
for (*a = 0; *a ¡ *A; *a = *a + 1){
//cout ¡¡ ”p” ¡¡ *p ¡¡ ” ” ¡¡ ”i” ¡¡ *i ¡¡ ” ” ¡¡ ”r” ¡¡ *r ¡¡ ” ” ¡¡ ”t” ¡¡ *t ¡¡ ” ” ¡¡ ” a” ¡¡ *a ¡¡ endl;
C4 += X[*p][*a][*i][*r];
}
C4 -= dl[*p][*i][*r][*t];
sprintf(varName, ”C4.pC4.setName(varName);
IloConstraintc(C4 == Ie[*p][*i][*r][*t]);
c.setName(varName);
MODEL1.add(c);
C4.end();
C4 = IloExpr(env);
}
}
}
}
}

    IloExprC5(env);
for(*p = 0;*p < *P;*p = *p + 1){
for(*i = 0;*i < *I;*i = *i + 1){
for(*r = 0;*r < *T;*r = *r + 1){
for(*t = 0;*t < *T;*t = *t + 1){
if(*r < *t){

    for(*a = 0;*a < *A;*a = *a + 1){
C5+ = X[*p][*a][*i][*r];
}
```

```
for(*e = *r; *e <= *t; *e = *e + 1){
C5- = dl[*p][*i][*r][*e];
}
for(*e = *r; *e < *t; *e = *e + 1){
C5- = B_prt[*p][*r][*e] * Ie[*p][*i][*r][*e];
}

    sprintf(varName, ”C5.p%di%dr%dt%d”, *p + 1, *i + 1, *r + 1, *t + 1);
C5.setName(varName);
IloConstraintc(C5 == Ie[*p][*i][*r][*t]);
c.setName(varName);
MODEL1.add(c);
C5.end();
C5 = IloExpr(env);
}
}
}
}
}

    IloExprC6(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*r < *t){
for(*a = 0; *a < *A; *a = *a + 1){
C6+ = X[*p][*a][*i][*r];
}
```

```
for(*e = *r; *e < *t; *e = *e + 1){
C6- = (dl[*p][*i][*r][*e] + B_prt[*p][*r][*e] * Ie[*p][*i][*r][*e]);
}
sprintf(varName, ”C6.p%di%dr%dt%d”, *p + 1, *i + 1, *r + 1, *t + 1);
C6.setName(varName);
IloConstraintc(C6 == Ib[*p][*i][*r][*t]);
c.setName(varName);
MODEL1.add(c);
C6.end();
C6 = IloExpr(env);

  }
}
}
}
}

  IloExprC7(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*r = 0; *r <= *t; *r = *r + 1){
C7+ = (dl[*p][*i][*r][*t]);
}
C7+ = Lk[*p][*i][*t];
sprintf(varName, ”C7.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C7.setName(varName);
IloConstraintc(C7 == d_pit[*p][*i][*t]);
c.setName(varName);
```

```
MODEL1.add(c);
C7.end();
C7 = IloExpr(env);

    }
}
}

    IloExprC8(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*r = 0; *r <= *t; *r = *r + 1){
C8+ = Ib[*p][*i][*r][*t];
}
for(*a = 0; *a < *A; *a = *a + 1){
for(*e = *t; *e <= *t + *LT − 1; *e = *e + 1){
C8+ = X[*p][*a][*i][*e];
}
}
C8− = *M * (1 − Y[*p][*i][*t]);
sprintf(varName, ”C8.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C8.setName(varName);
IloConstraintc(C8 <= (SL[*p][*i] − *c));
c.setName(varName);
MODEL1.add(c);
C8.end();
C8 = IloExpr(env);
}
```

```
}
}

    IloExprC9(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*r = 0; *r <= *t; *r = *r + 1){
C9+ = Ib[*p][*i][*r][*t];
}
for(*a = 0; *a < *A; *a = *a + 1){
for(*e = *t; *e <= *t + *LT − 1; *e = *e + 1){

    C9+ = X[*p][*a][*i][*e];
}
}
C9+ = *M * Y[*p][*i][*t];
sprintf(varName, ”C9.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C9.setName(varName);
IloConstraintc(C9 >= (SL[*p][*i] − *c));
c.setName(varName);
MODEL1.add(c);
C9.end();
C9 = IloExpr(env);
}
}
}

    IloExprC10(env);
```

```
for(*p = 0; *p < *P; *p = *p + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T - 1; *t = *t + 1){//checkwithwalaa
C10 = X[*p][*a][*i][*t + *LT];
sprintf(varName,”C10.p%da%di%dt%d”, *p + 1, *a + 1, *i + 1, *t + 1);
C10.setName(varName);
IloConstraintc(C10 <= *M * Y[*p][*i][*t]);
c.setName(varName);
MODEL1.add(c);
C10.end();
C10 = IloExpr(env);
}
}
}
}

    IloExprC11(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T - 1; *t = *t + 1){//checkwithwalaa
for(*a = 0; *a < *A; *a = *a + 1){
C11+ = X[*p][*a][*i][*t + *LT];
}
C11+ = *M * (1 - Y[*p][*i][*t]);
for(*r = 0; *r <= *t; *r = *r + 1){
C11+ = Ib[*p][*i][*r][*t];
}
for(*a = 0; *a < *A; *a = *a + 1){
```

```
for(*e = *t; *e <= *t + *LT − 1; *e = *e + 1){
C11+ = X[*p][*a][*i][*e];
}
}
sprintf(varName, ”C11.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C11.setName(varName);
IloConstraintc(C11 >= SU[*p][*i]);
c.setName(varName);
MODEL1.add(c);
C11.end();
C11 = IloExpr(env);
}
}
}


    IloExprC12(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T − 1; *t = *t + 1){
for(*a = 0; *a < *A; *a = *a + 1){
C12+ = X[*p][*a][*i][*t + *LT];
}
for(*r = 0; *r <= *t; *r = *r + 1){
C12+ = Ib[*p][*i][*r][*t];
}
for(*a = 0; *a < *A; *a = *a + 1){
for(*e = *t; *e <= *t + *LT − 1; *e = *e + 1){
C12+ = X[*p][*a][*i][*e];
}
```

```
}
sprintf(varName,"C12.p%di%dt%d", *p + 1, *i + 1, *t + 1);
C12.setName(varName);
IloConstraintc(C12 <= SU[*p][*i]);
c.setName(varName);
MODEL1.add(c);
C12.end();
C12 = IloExpr(env);
}
}
}

    IloExprC13(env);
for(*a = 0; *a < *A; *a = *a + 1){
for(*t = 0; *t < *T; *t = *t + 1){

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
C13+ = X[*p][*a][*i][*t];
}
}
sprintf(varName,"C13.a%dt%d", *a + 1, *t + 1);
C13.setName(varName);
IloConstraintc(C13 <= Q_at[*a][*t]);
c.setName(varName);
MODEL1.add(c);
C13.end();
C13 = IloExpr(env);
}
```

```
}

    IloExprC14(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
C14 = X[*p][*a][*i][*t];
sprintf(varName, ”C14.p%da%di%dt%d”, *p + 1, *a + 1, *i + 1, *t + 1);
C14.setName(varName);
IloConstraintc(C14 <= *M * Z[*a][*i]);
c.setName(varName);
MODEL1.add(c);
C14.end();
C14 = IloExpr(env);
}
}
}
}

    IloExprC15(env);
for(*b = 0; *b < *B; *b = *b + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*p = 0; *p < *P; *p = *p + 1){
C15 = W[*b][*a][*p][*t];
}
sprintf(varName, ”C15.b%da%dt%d”, *b + 1, *a + 1, *t + 1);
C15.setName(varName);
IloConstraintc(C15 <= *M * O[*b][*a]);
c.setName(varName);
MODEL1.add(c);
C15.end();
C15 = IloExpr(env);
}
}
}

    IloExprC16(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*r = *t; *r <= *t; *r = *r + 1){
C16+ = Ie[*p][*i][*r][*t];
}
sprintf(varName, ”C16.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C16.setName(varName);
IloConstraintc(C16 <= *M * U[*p][*i][*t]);
c.setName(varName);
MODEL1.add(c);
C16.end();
C16 = IloExpr(env);
}
}
}

    IloExprC17(env);
for(*p = 0; *p < *P; *p = *p + 1){
```

```
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
for(*r = *t; *r <= *t; *r = *r + 1){
C17+ = Ie[*p][*i][*r][*t];
}
sprintf(varName, ”C17.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C17.setName(varName);
IloConstraintc(C17 >= *c * U[*p][*i][*t]);
c.setName(varName);
MODEL1.add(c);
C17.end();
C17 = IloExpr(env);
}
}
}


    IloExprC18(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
C18+ = Lk[*p][*i][*t];
sprintf(varName, ”C18.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C18.setName(varName);
IloConstraintc(C18 <= *M * (1 − U[*p][*i][*t]));
c.setName(varName);
MODEL1.add(c);
C18.end();
C18 = IloExpr(env);
}
}
}

    IloExprC19(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*t = 0; *t < *T; *t = *t + 1){

    for(*b = 0; *b < *B; *b = *b + 1){
C19+ = W[*b][*a][*p][*t];
}

    for(*i = 0; *i < *I; *i = *i + 1){
/ * cout << Y_p[*p] << endl; */
C19− = Y_p[*p] * X[*p][*a][*i][*t];
}
sprintf(varName, ”C19.p%da%dt%d”, *p + 1, *a + 1, *t + 1);
C19.setName(varName);
IloConstraintc(C19 == 0);
c.setName(varName);
MODEL1.add(c);
C19.end();
C19 = IloExpr(env);
}
}
}

    IloExprC20(env);
for(*b = 0; *b < *B; *b = *b + 1){
```

```
for(*t = 0; *t < *T; *t = *t + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*p = 0; *p < *P; *p = *p + 1){
C20+ = W[*b][*a][*p][*t];
}
}
C20− = C_pt[*b][*t];
sprintf(varName, ”C20.b%dt%d”, *b + 1, *t + 1);
C20.setName(varName);
IloConstraintc(C20 <= 0);
c.setName(varName);
MODEL1.add(c);
C20.end();
C20 = IloExpr(env);
}
}

    IloExprC21(env);
for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*t <= *LT){
for(*a = 0; *a < *A; *a = *a + 1){
C21+ = X[*p][*a][*i][*t];
}
sprintf(varName, ”C21.p%di%dt%d”, *p + 1, *i + 1, *t + 1);
C21.setName(varName);
IloConstraintc(C21 == Ib[*p][*i][0][*t]);
c.setName(varName);
/ * MODEL1.add(c); */
C21.end();
C21 = IloExpr(env);

    }
}
}
}

    IloCplexcplex(MODEL1);

    / * cplex.setParam(IloCplex :: TiLim, 60); */

    / * cplex.setOut(env.getNullStream()); */
cplex.extract(MODEL1);
/ * cplex.exportModel(”MODELL.alp”); */
/ * cplex.exportModel(”MODELL.RLP”); */
cplex.exportModel(”MODELL.lp”);
/ * cplex.exportModel(”MODELL.sav”); */

    cplex.setParam(IloCplex :: Param :: Emphasis :: MIP, 4);
/ * cplex.setParam(IloCplex :: Param :: MIP :: Limits :: Nodes, 5000); */

    cplex.solve();

    //Createafilestreamobjectandcheckthatitopenedcorrectly
ofstreamoutFile(”Result.csv”);
if(!outFile.is_open())
{
```

```
std :: cerr << ”Failedtoopenoutputfile” << std :: endl;
}

    float ∗ val = newfloat(0);

    outFile << cplex.getObjValue() << endl;

    end = clock();

    outFile << ” = 1ifabattoir/slaughterhouseaisallocatedtobreeding/farmcenterb, otherwise0” <<
endl;

    for(∗b = 0; ∗b < ∗B; ∗b = ∗b + 1){
for(∗a = 0; ∗a < ∗A; ∗a = ∗a + 1){
∗val = cplex.getValue(O[∗b][∗a]);
outFile << ”O” << ”b” << ∗b + 1 << ”” << ”a” << ∗a + 1 << ”, ” << ∗val << endl;
}
}

    outFile << endl;

    outFile << ” = 1ifretailer/warehouseiisallocatedtoabattoir/slaughterhousea, otherwise0” <<
endl;

    for(∗a = 0; ∗a < ∗A; ∗a = ∗a + 1){
for(∗i = 0; ∗i < ∗I; ∗i = ∗i + 1){
∗val = cplex.getValue(Z[∗a][∗i]);
outFile << ”Z” << ”a” << ∗a + 1 << ”” << ”i” << ∗i + 1 << ”, ” << ∗val << endl;
}
```

```
}
outFile << endl;

    outFile << "Amountofproductptransportedtoretailerifromabattoiraattimeperiodt" <<
endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
*val = cplex.getValue(X[*p][*a][*i][*t]);
outFile << "X" << "p" << *p+1 << "" << "a" << *a+1 << "" << "i" << *i+1 <<
"" << "t" << *t + 1 << "," << *val << endl;
}
}
}
}

    outFile << endl;
outFile << " = 1ifretailer/warehouseimakesanorderforproductpatthebeginningoftimeperiodt, otherw
endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
*val = cplex.getValue(Y[*p][*i][*t]);
outFile << "Y" << "p" << *p+1 << "" << "i" << *i+1 << "" << "t" << *t+1 <<
"," << *val << endl;
}
}
}

    outFile << endl;

    outFile << ” = 1whentheon−handinventoryinretailer/warehousei′ssiteforproductpisabovezeroatth
endl;

    for(∗p = 0; ∗p < ∗P; ∗p = ∗p + 1){
for(∗i = 0; ∗i < ∗I; ∗i = ∗i + 1){
for(∗t = 0; ∗t < ∗T; ∗t = ∗t + 1){
∗val = cplex.getValue(U[∗p][∗i][∗t]);
outFile << ”U” << ”p” << ∗p + 1 << ”” << ”i” << ∗i + 1 << ”” << ”t” << ∗t + 1 <<
”,” << ∗val << endl;
}
}
}

    outFile << endl;
outFile << ”Reorderpointforproductpintheretailer/warehousei” << endl;

    for(∗p = 0; ∗p < ∗P; ∗p = ∗p + 1){
for(∗i = 0; ∗i < ∗I; ∗i = ∗i + 1){
∗val = cplex.getValue(SL[∗p][∗i]);
outFile << ”SL” << ”p” << ∗p + 1 << ”” << ”i” << ∗i + 1 << ”,” << ∗val << endl;
}
}

    outFile << endl;
outFile << ”Order − up − tolevelforproductpintheretailer/warehousei” << endl;

    for(∗p = 0; ∗p < ∗P; ∗p = ∗p + 1){
for(∗i = 0; ∗i < ∗I; ∗i = ∗i + 1){
∗val = cplex.getValue(SU[∗p][∗i]);
outFile << ”SU” << ”p” << ∗p + 1 << ”” << ”i” << ∗i + 1 << ”,” << ∗val << endl;
}
}

    outFile << endl;
outFile << ”Amountofproductpdeliveredtocustomersinretailer/warehousesiteiattimeperiodt, whichisp
endl;

    for(∗p = 0; ∗p < ∗P; ∗p = ∗p + 1){
for(∗i = 0; ∗i < ∗I; ∗i = ∗i + 1){
for(∗r = 0; ∗r < ∗T; ∗r = ∗r + 1){
for(∗t = 0; ∗t < ∗T; ∗t = ∗t + 1){
if(∗r <= ∗t){
∗val = cplex.getValue(dl[∗p][∗i][∗r][∗t]);
outFile << ”dl” << ”p” << ∗p+1 << ”” << ”i” << ∗i+1 << ”” << ”r” << ∗r+1 <<
”” << ”t” << ∗t + 1 << ”,” << ∗val << endl;
}
}
}
}
}

    outFile << endl;
outFile << ”Numberoflivestocktransportedtoabattoirafrombreedingcenterbattimeperiodt” <<
```

```
endl;

    for(*b = 0; *b < *B; *b = *b + 1){
for(*a = 0; *a < *A; *a = *a + 1){
for(*p = 0; *p < *P; *p = *p + 1){
for(*t = 0; *t < *T; *t = *t + 1){
*val = cplex.getValue(W[*b][*a][*p][*t]);
outFile << ”W” << ”b” << *b + 1 << ”” << ”a” << *a + 1 << ”” << ”p” <<
*p + 1 << ”” << ”t” << *t + 1 << ”,” << *val << endl;
}
}
}
}

    outFile << endl;
outFile << ”Retaileri'son−handinventoryofproductpatthebeginningoftimeperiodt, producedintimeperi
endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*r <= *t){
*val = cplex.getValue(Ib[*p][*i][*r][*t]);
outFile << ”Ib” << ”p” << *p+1 << ”” << ”i” << *i+1 << ”” << ”r” << *r+1 <<
”” << ”t” << *t + 1 << ”,” << *val << endl;
}
}
}
```

```
}
}

    outFile << endl;
outFile << "Retaileri'son−handinventoryofproductpattheendoftimeperiodt,producedintimeperiodr" <
endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*r = 0; *r < *T; *r = *r + 1){
for(*t = 0; *t < *T; *t = *t + 1){
if(*r <= *t){
*val = cplex.getValue(Ie[*p][*i][*r][*t]);
outFile << "Ie" << "p" << *p+1 << "" << "i" << *i+1 << "" << "r" << *r+1 <<
"" << "t" << *t + 1 << "," << *val << endl;
}
}
}
}
}

    outFile << endl;
outFile << "Lost − salequantityofproductpfordemandintimeperiodtandretaileri." <<
endl;

    for(*p = 0; *p < *P; *p = *p + 1){
for(*i = 0; *i < *I; *i = *i + 1){
for(*t = 0; *t < *T; *t = *t + 1){
*val = cplex.getValue(Lk[*p][*i][*t]);
outFile << ”LK” << ”p” << *p + 1 << ”” << ”i” << *i + 1 << ”” << ”t” <<
*t + 1 << ”,” << *val << endl;
}
}
}

    //closethefile
outFile.close();

    }
catch(IloException&ex){
cerr << ”Error : ” << ex << endl;
}
catch(...){
cerr << ”Error” << endl;
}

    }

    voidfreeData(){

    template < typenameT >
T * *AllocateDynamic2DArrayCSTYLE(intnRows, intnCols){

    int * m = newint(0);

    T * *dynamicArray;

    dynamicArray = (T * *)calloc(nRows, sizeof(T*));
```

$for(*m = 0; *m < nRows; *m = *m + 1)$

$\{$

$dynamicArray[*m] = (T*)calloc(nCols, sizeof(T));$

$\}$

$return(dynamicArray);$

$free(m);$

$\}$

$template < typenameT >$

$voidFreeDynamic2DArrayCSTYLE(T * *dArray, intnRows, intnCols)\{$

$int * m = newint(0);$

$for(*m = 0; *m < nRows; *m = *m + 1)$

$\{$

$free(dArray[*m]);$

$\}$

$free(dArray);$

$free(m);$

$\}$

$int * create_{i}nt_{v}ector(intdim)$

$\{$

$int * ptr;$

$if((ptr = (int*)calloc(dim, sizeof(int))) == NULL)\{$

$printf(" : Memoriainsuficiente");$

$exit(8);$

}

$returnptr;$

$free(ptr);$

}

$longdouble * create_double_vector(intdim)$

{

$longdouble * ptr;$

$if((ptr = (longdouble*)calloc(dim, sizeof(longdouble))) == NULL)\{$

$printf(: Memoriainsuficiente");$

$exit(8);$

}

$returnptr;$

$free(ptr);$

}

$template < typenameT >$

$T * * * AllocateDynamic3DArrayCSTYLE(intnRows, intnCols, intnDep)\{$

$int * m = newint(0);$

$int * i = newint(0);$

$T * * * dynamicArray;$

$dynamicArray = (T * **)calloc(nRows, sizeof(T * *));$

$for(*m = 0; *m < nRows; *m = *m + 1)$

{

```
dynamicArray[*m] = (T * *)calloc(nCols, sizeof(T*));
for(*i = 0; *i < nCols; *i = *i + 1)
{
dynamicArray[*m][*i] = (T*)calloc(nDep, sizeof(T));
}
}
return(dynamicArray);

    free(m);
free(i);
}

    template < typenameT >
voidFreeDynamic3DArrayCSTYLE(T * * * dArray, intnRows, intnCols, intnDep){

    int * m = newint(0);
int * i = newint(0);

    for(*m = 0; *m < nRows; *m = *m + 1)
{
for(*i = 0; *i < nCols; *i = *i + 1)
{
free(dArray[*m][*i]);
}
free(dArray[*m]);
}
free(dArray);

    free(m);
free(i);
}

    template < typenameT >
T * AllocateDynamic1DArrayCSTYLE(intnRows){

    T * dynamicArray;

    dynamicArray = (T*)calloc(nRows, sizeof(T));

    return(dynamicArray);
}

    template < typenameT >
voidFreeDynamic1DArrayCSTYLE(T * dArray, intnRows){
free(dArray);
}
```

References

Abuobidalla, O., Chen, M., & Chauhan, S. (2019). A matheuristic method for planning railway freight transportation with hazardous materials. *Journal of Rail Transport Planning & Management*, *10*, 46–61.

Abuobidalla, O. A., Chen, M., & Chauhan, S. S. (2021). Modelling and solving an integrated freight train scheduling and trip planning problem with hazardous materials. *International Journal of Rail Transportation*, *9*(3), 256–289.

Ahumada, O., & Villalobos, J. R. (2009). Application of planning models in the agri-food supply chain: A review. *European journal of Operational research*, *196*(1), 1–20.

Amiri-Aref, M., Klibi, W., & Babai, M. Z. (2018). The multi-sourcing location inventory problem with stochastic demand. *European Journal of Operational Research*, *266*(1), 72–87.

Banasik, A., Kanellopoulos, A., Claassen, G., Bloemhof-Ruwaard, J. M., & van der Vorst, J. G. (2017). Assessing alternative production options for eco-efficient food supply chains using multi-objective optimization. *Annals of Operations Research*, *250*(2), 341–362.

Beemsterboer, B., Teunter, R., & Riezebos, J. (2016). Two-product storage-capacitated inventory systems: A technical note. *International Journal of Production Economics*, *176*, 92–97.

Brandenburg, M., & Rebs, T. (2015). Sustainable supply chain management: A modeling perspective. *Annals of Operations Research*, *229*(1), 213–252.

Cunha, P. S., Raupp, F. M., & Oliveira, F. (2017). A two-stage stochastic programming model for periodic replenishment control system under demand uncertainty. *Computers & Industrial Engineering*, *107*, 313–326.

Dekker, R., Bloemhof, J., & Mallidis, I. (2012). Operations research for green logistics–an overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*, *219*(3), 671–679.

Demir, E., Bektaş, T., & Laporte, G. (2014). A review of recent research on green road freight transportation. *European Journal of Operational Research*, *237*(3), 775–793.

Duan, Q., & Liao, T. W. (2013). A new age-based replenishment policy for supply chain inventory optimization of highly perishable products. *International Journal of Production Economics*, *145*(2), 658–671.

Eskandarpour, M., Dejax, P., Miemczyk, J., & Péton, O. (2015). Sustainable supply chain network design: An optimization-oriented review. *Omega*, *54*, 11–32.

García-Flores, R., Higgins, A., Prestwidge, D., & McFallan, S. (2014). Optimal location of spelling yards for the northern australian beef supply chain. *Computers and Electronics in Agriculture*, *102*, 134–145.

Gholami-Zanjani, S. M., Jabalameli, M. S., & Pishvaee, M. S. (2021). A resilient-green model for multi-echelon meat supply chain planning. *Computers & Industrial Engineering*, *152*, 107018.

Golini, R., Moretto, A., Caniato, F., Caridi, M., & Kalchschmidt, M. (2017). Developing sustainability in the italian meat supply chain: an empirical investigation. *International Journal of Production Research*, *55*(4), 1183–1209.

Huang, H., He, Y., & Li, D. (2018). Pricing and inventory decisions in the food supply chain with production disruption and controllable deterioration. *Journal of Cleaner Production*, *180*, 280–296.

Jain, A., Groenevelt, H., & Rudi, N. (2011). Periodic review inventory management with contingent use of two freight modes with fixed costs. *Naval Research Logistics (NRL)*, *58*(4), 400–409.

Juneja, V. K., & Marks, H. M. (2006). Growth kinetics of salmonella spp. pre-and post-thermal treatment. *International Journal of Food Microbiology*, *109*(1-2), 54–59.

Martel, A. (2003). Policies for multi-echelon supply: Drp systems with probabilistic time-varying demands. *INFOR: Information Systems and Operational Research*, *41*(1), 71–91.

Mochizuki, M., & Hattori, T. (1987). Kinetic study of growth throughout the lag phase and the exponential phase of escherichia coli. *FEMS Microbiology Ecology*, *3*(5), 291–296.

Mogale, D., Kumar, M., Kumar, S. K., & Tiwari, M. K. (2018). Grain silo location-allocation problem with dwell time for optimization of food grain supply chain network. *Transportation Research Part E: Logistics and Transportation Review*, *111*, 40–69.

Mohammed, A., & Wang, Q. (n.d.). An investment evaluation of a rfid-enabled meat supply chain: a multi-criteria approach. In *MATEC Web of Conferences* (Vol. 70, p. 06003).

Mohammed, A., Wang, Q., & Li, X. (2015). A cost-effective decision-making algorithm for integrity of HMSCs: A multi-objective approach. In *2015 IEEE 10th International Conference on Industrial and Information Systems ICIIS* (pp. 106–110).

Mohebalizadehgashti, F., Zolfagharinia, H., & Amin, S. H. (2020). Designing a green meat supply chain network: A multi-objective approach. *International Journal of Production Economics*, *219*, 312–327.

Rahbari, M., Hajiagha, S. H. R., Mahdiraji, H. A., Dorcheh, F. R., & Garza-Reyes, J. A. (2021). A novel location-inventory-routing problem in a two-stage red meat supply chain with logistic decisions: evidence from an emerging economy. *Kybernetes*.

Resat, H. G., & Turkay, M. (2019). A discrete-continuous optimization approach for the design and operation of synchromodal transportation networks. *Computers & Industrial Engineering*, *130*, 512–525.

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*(2), 345–355.

Tendall, D., Joerin, J., Kopainsky, B., Edwards, P., Shreck, A., Le, Q. B., … Six, J. (2015). Food system resilience: defining the concept. *Global Food Security*, *6*, 17–23.

Vo, T. L. H., & Thiel, D. (2011). Economic simulation of a poultry supply chain facing a sanitary crisis. *British Food Journal,Emerald, 2011, 113 (8), 011-1030*.

Workie, E., Mackolil, J., Nyika, J., & Ramadas, S. (2020). Deciphering the impact of covid-19 pandemic on food security, agriculture, and livelihoods: A review of the evidence from developing countries. *Current Research in Environmental Sustainability*, 100014.