A BLOOD DISTRIBUTION SYSTEM:
AN APPLICATION TO THE TURKISH RED CRESCENT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ATIL KURT

IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

JANUARY 2018

Approval of the thesis

A BLOOD DISTRIBUTION SYSTEM:
AN APPLICATION TO THE TURKISH RED CRESCENT

Submitted by **ATIL KURT** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Science** _____________

Prof. Dr. Yasemin Serin
Head of Department, **Industrial Engineering** _____________

Prof. Dr. Meral Azizoğlu
Supervisor, **Industrial Engineering Dept., METU** _____________

Assoc. Prof. Dr. Ferda Can Çetinkaya
Co-supervisor, **Industrial Engineering Dept., Çankaya University** _____________

Examining Committee Members:

Prof. Dr. Ömer Kırca
Industrial Engineering Dept., METU _______________

Prof. Dr. Meral Azizoğlu
Industrial Engineering Dept., METU _______________

Assist. Prof. Dr. Banu Lokman
Industrial Engineering Dept., METU _______________

Assoc. Prof. Dr. Mustafa Alp Ertem
Industrial Engineering Dept., Çankaya University _______________

Assist. Prof. Dr. Serhat Gül
Industrial Engineering Dept., TED University _______________

Date: 29.01.2018

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Atıl KURT

Signature :

ABSTRACT

A BLOOD DISTRIBUTION SYSTEM:
AN APPLICATION TO THE TURKISH RED CRESCENT

KURT, Atıl

Ph.D. Department of Industrial Engineering

Supervisor: Prof. Dr. Meral AZİZOĞLU

Co-Supervisor: Assoc. Prof. Dr. Ferda Can ÇETİNKAYA

January 2018, 139 pages

In this study, we consider the blood distribution system in Turkey and focus on Central Anatolian Regional Blood Center. Our problem differs from the classical distribution problems as it resides irradiation centers, urgent demands, vehicle availability and traveling time restrictions. In this study, we considered two problems.

To address the first problem, we develop a mixed integer linear program with two objectives: maximizing the demand satisfaction and minimizing total time travelled by the vehicles. We propose two decomposition-based heuristic solution approaches. The results of our experiments have revealed that the model cannot solve even small sized instances in reasonable times; however, the heuristic solution approaches are appropriate for solving complex real life problems.

Second problem proposes several demand satisfaction options by taking into account the irradiation centers, urgent demands, and product availability. To address the problem, we develop a mixed integer linear program with the objective of maximizing the weighted demand satisfaction, and propose a hybrid genetic algorithm. The results of our experiments have revealed that the model cannot solve even small sized instances in reasonable times; however, the hybrid genetic algorithm is appropriate for solving complex real life problems.

Keywords: Blood distribution system, Mathematical model, Heuristic approach, Genetic algorithm

ÖZ

KAN DAĞITIM SİSTEMİ: TÜRK KIZILAYI İÇİN BİR UYGULAMA

KURT, Atıl

Doktora, Endüstri Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Meral AZİZOĞLU

Ortak Tez Yöneticisi: Doç. Dr. Ferda Can ÇETİNKAYA

Ocak 2018, 139 sayfa

Bu çalışmada Orta Anadolu Bölge Kan Merkezini'ne odaklanarak Türkiye'deki kan dağıtım sistemini ela aldık. Problemimiz ışınlama merkezlerini, acil talepleri, araç ulaşabilirlik ve seyahat süresi gibi kısıtları dikkate aldığından literaturedeki klasik dağıtım sistemlerinden farklı bir yapıya sahip olup iki tip problem incelenmiştir.

Birinci problem iki amaçlı bir karışık tamsayılı doğrusal programlama modeli geliştirmiş olup ağırlıklandırılmış karşılanan talep miktarının en çoklanmayı ve toplam gezinti süresini en azlamayı amaçlamıştır. İki ayrıştırma-tabanlı sezgisel çözüm yaklaşımı geliştirdik. Deneylerimizin sonuçları modelin küçük boyutlu sayılabilecek problemleri bile çözemediğini gösterdi; ancak sezgisel çözüm yaklaşımlarının karmaşık gerçek yaşam problemleri için daha uygundur.

İkinci problem ışınlama merkezlerini, acil talepleri ve ürün elverişliliğini dikkate alan ve farklı talep sağlama yöntemlerini önermektedir. Problem çözümü için bir toplam ağırlıklandırılmış karşılanan talep miktarını en çoklanmasını amaçlayan karmaşık tamsayılı doğrusal programlama modeli geliştirilmiştir ve bir melez genetik algoritma önerilmiştir. Deneylerimizin sonuçları modelin küçük boyutlu sayılabilecek problemleri bile çözemediğini gösterdi; ancak sezgisel çözüm yaklaşımlarının karmaşık gerçek yaşam problemleri için daha uygundur.

Anahtar Kelimeler: Kan dağıtım Sistemi, Matematiksel model, Sezgisel yaklaşım**,** Genetik algoritma

ACKNOWLEDGEMENTS

I would like to express my great gratitude to my supervisors Prof. Dr. Meral Azizoğlu and Assoc. Prof. Dr. Ferda Can Çetinkaya for their guidance leadership, interest, patience and encouragement during this thesis. It was a great chance and opportunity to work with them.

I am also grateful to Assist. Prof. Dr. Banu Lokman, Assoc. Prof. Dr. Mustafa Alp Ertem, Prof. Dr. Ömer Kırca and Asst. Prof. Dr. Serhat Gül for accepting to review this thesis and their valuable contributions.

I am always indebted to my family, I am especially grateful to my parents Behzat Kurt and Sultan Kurt for their endless and unconditional love. I also would like to thank my brothers Sadık Kurt, Yusuf Kurt and Volkan Kurt, and my sister Binnaz Kılıç for their support in my whole life

I wish to express my appreciation to all Central Anatolian Regional Blood Center and Blood Services General Directorate of the Turkish Red Crescent personnel for their help throughout this study.

I would sincerely thank to all my friends who have contributed indirectly to this thesis. I especially would like to extend my special thanks to Beyza Delibaş and Beata Trochanowska for their friendship, and morale support during this study. I also thank to Mesut Koç for his help on my research.

I also want to thank all METU and Çankaya University personnel for their help and friendship throughout this study. I especially thank to Yağmur Sönmez Demir and Efe Çiftci for their extensive help.

Finally, I would like to thank TÜBİTAK (The Scientific and Technological Research Council of Turkey) for supporting this study through a doctorate study scholarship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The blood products supply chain management is more crucial than those of the ordinary goods due to the vital importance of blood for human beings. The associated supply chain has several different characteristics as stated by Pierskalla (2005).

1. Blood is a perishable product and its sub-products have different shelf lives.
2. The blood supply is not known with certainty.
3. The demands for blood products are not known with certainty.
4. Interactive decisions must be made at the strategic design and, at operational and tactical levels.
5. The entire supply chain should be examined as a whole system but not as a subsystem of some larger systems.
6. Generalized theoretical research can be derived via the practical real life problems.

Blood supply chain systems of ten different countries are studied by Rock et al. (2000). As stated in the study, in the USA, the blood is supplied by different organizations: 85 percent of the blood products are supplied by non-profit organizations like America’s Blood Centers and American Red Cross and, the rest are supplied by for-profit organizations or hospitals. The blood products supply chains in England, Scotland, Italy, Switzerland, France, and Canada are organized by national non-profit organizations. In Norway, all blood centers are managed within the hospitals, whereas in South Africa, private organizations play a major role.

The blood supply chain in Turkey is organized by the Turkish Red Crescent (TRC), a non-profit organization. The organization is responsible to supply the whole blood, from the safe donors, extract the required blood product derivatives like red blood cells, plasma, and platelets and distribute them to the hospitals at the required times. The timely delivery of the blood products is crucial particularly for the urgent demand and emergency cases. The origin of TRC is Ottoman Assistance Association for the injured and sick soldier, which is founded in 1868. Service activity areas of TRC are blood, disaster intervention, healthcare, social benefits, youth, and education in all places that the human needs. Some historical events about blood supply system in Turkey are as follows:

- First transfusion is made in 1938,
- First blood centers are established in 1957,
- Blood donation organization is established in 1974,
- Blood Service Directorate of TRC is established in 1983,
- In 2007, the TRC is assigned to meet blood demands in Turkey and 15 Regional Blood Centers (RBC) are established.

RBCs, Blood Centers (BC), and all laboratories in TRC have ISO 9001:20001 Quality Management System Certificate.

The Turkish Red Crescent distributes the blood to the demand points using the following six channels:

i. Regional Blood Centers (RBC)
ii. Temporary Regional Blood Centers (TRBC)
iii. Blood Centers (BC)
iv. Blood Stations (BS)
v. Mobile Units (MU)
vi. Transfusion Centers/Hospitals (H)

There are four main blood subproducts which are supplied by the TRC. These are red blood cells, plasma, platelets, and apheresis platelets. The Turkish Red Crescent also considers the ABO group system and Rhesus factor for each supplied blood product while meeting the blood demand in Turkey. The detailed information about the type of supplied blood products are discussed in Chapter 2.

Our main focus will be on the distribution system of the supply chain in TRC. Inspired from a practical problem, we consider the routing of the vehicles to the hospitals. We consider two blood distribution problems in this study.

1.1 Problem 1

The poor allocation of the blood products from the Central Anatolian RBC to the hospitals has posed a main challenge in our study. We recognize that developing an efficient blood distribution model is critical for academicians and practitioners due to the complexity of the timely allocation and distribution. We consider the routing of the vehicles to the hospitals and two objectives in hierarchy: maximum demand satisfaction and minimum total time travelled. We construct a mathematical model and propose two heuristic algorithms.

To the best of our knowledge, our study is the first attempt that considers irradiation centers, urgent demands and blood products availability. Including irradiation centers leads to precedence structure, and product availability leads to the consideration of the partial demand satisfaction. Hence, we consider the allocation of the blood products along with their distribution to the hospitals and add a new category, *distribution - product allocation*, to the blood products literature.

1.2 Problem 2

Problem 1 considered the blood products allocations and distribution problem with urgent demands, product availabilities, and irradiation centers. The promising results observed in Problem 1 and our continuing collaboration with the Turkish Red Crescent have been the main motivation of this problem. We recognize that the demand may not be satisfied fully due to the limited capacities of the Turkish Red Crescent, hence define new transportation options that use the vehicles of the hospitals in addition to the vehicles of the organization.

In the current blood distribution system of the Central Anatolian RBC, the demand of hospitals is satisfied by its own vehicles. Due to the capacity and maximum travel time constraints of the vehicles, the demand of all hospitals cannot be satisfied by this way. To increase the amount of demand satisfaction, we propose several options to distribute the blood products. Satisfying the demand of the hospitals by the vehicles of the RBC is the first option (O1) of our proposed distribution system, and we call it as the *current operation*. Satisfying the demand of a hospital via another hospital where the hospital takes its demand from another hospital visited by the vehicle of the RBC is our second option (O2), and we call it as the *transfer service*. Satisfying the demand of a hospital directly from the RBC by the hospital's vehicle visiting the RBC is the third option (O3), and we call it as the *self service*. We develop a mixed integer linear program with the objective of maximizing the demand satisfaction and a propose a hybrid genetic algorithm to solve the problem.

1.3 Chapter Organization

The thesis is organized as follows. Chapter 2 describes the blood supply chain system in Turkey and blood products. We present Problem 1 in Chapter 3. Problem 2 is discussed in Chapter 4. Finally, our conclusion and future works are given in Chapter 5.

CHAPTER 2

PROBLEM ENVIRONMENT

Our main focus will be on the blood distribution system in Turkey. In this chapter, we give general information about the blood products and the blood supply systems in Turkey.

2.1 Blood Products

The main product of the TRC is the whole blood that is taken directly from the human beings, *donors*, and this blood is used in different cases when treating patients. The whole blood once taken from the donors should be *transfused* (the process of giving the blood to the human body) or *centrifuged* (the process of separating the components) in one day as its shelf life is one day. Following the centrifugation process, three main products, namely *red blood cells*, *plasma*, and *platelets* are obtained. The different components have different shelf lives and the separation enables one to adapt the use of the blood products to the specific needs of a patient. The treatment cases and shelf lives of the products are summarized in Table 2.1.

Table 2.1: Usage and shelf lives of the blood components

Product	**Treatment**	**Shelf Life**
Red Blood Cells	Surgery with major blood loss	42 days
	Treatment of anaemic patients	
	Premature infants	
Platelets	Major blood loss	5 days
	Cancer treatment	
Plasma	Blood loss and curbs in surgery	2 years
	Treatment of liver disease	
	Treatment of burn injuries	

Figure 2.1 below shows the structure of the blood products and the detailed explanations for the products are given below.

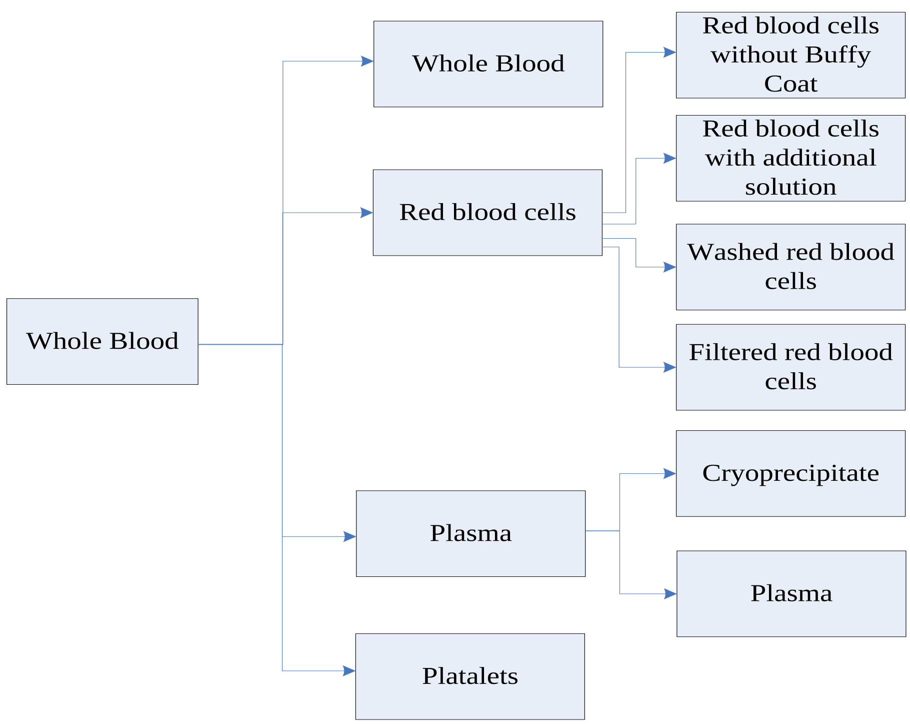

Figure 2.1: Structure of the blood products

1. Red blood cells: Main duty of Red blood cells is delivering oxygen to the body tissues. The following four subproducts can be obtained from red blood cells.
    - Red blood cells without Buffy Coat: Produced by removing Buffy Coat and some part of its plasma from red blood cells.
    - Red blood cells with additional solution: Produced by adding a proper solution to red blood cells.
    - Washed red blood cells: Produced by washing of red blood cells with isotonic solution.
    - Filtered red blood cells: Produced by removing leucocytes from the red blood cells.

    Each subproduct has a shelf life of 42 days.

2. Plasma: Plasma is a liquid composite that holds blood cells in the whole blood. After the centrifugation process, plasma is frozen and its temperature is reduced to under -30 degree in one hour.
    - Once plasma is centrifuged at high speed and concentrated up to 40 ml, it is called cryoprecipitate.

    Plasma itself and cryoprecipitate, each has a shelf life of 2 years.

3. Platelet: It is a composite that includes high incidence platelet ingredients of whole blood. Main duty of the platelets is to stop bleeding vessel injuries. The whole blood should be centrifuged in 24 hours; otherwise it loses its platelet properties. Platelet has a shelf life of 5 days.

Some blood products are directly obtained from the donors using apheresis device. The device filters the blood to take the desired components and sends the other components back to the donor. Turkish Red Crescent gets platelets via apheresis device and calls the obtained product as Aphaeresis platelets.

Irradiation is a process that ionizes the blood components to destroy all living leukocytes. The products that are subjected to the irradiation process are referred to as irradiated products. TRC is also responsible for the irradiation of the blood products.

The products are safely transported to the hospitals in special boxes without any need for the refrigerator, in about 2 hours.

There are many issues while making transfusion. The most important issues are the ABO group system and Rhesus factor, as the patient's blood type and transfused blood type should coincide. Although some blood types can receive blood of different types, this is not applicable in today's world. This can be applied in too rare cases such as war, earthquake, and huge disaster. Therefore, ABO group system and Rhesus factors for blood products should be considered in the blood supply chain system.

There are four groups in ABO group system: A, B, AB, and 0. A and B are the names of the two proteins in the blood. If a person's blood carries only protein A, then blood type is A. If it carries only protein B, then blood type is B. If it carries both of them, then blood type is AB. If it carries none of them, then blood type is 0.

There are two groups in the Rhesus factor: negative and positive. The name of the blood type is determined by Antigen D in the blood. If a person's blood carries Antigen D, then blood type is called as positive, otherwise blood type is called as negative. There are other antigens in the human body which are C, c, E and e. These antigens are not generally considered while transfusing, but they have to be considered for some patients while transfusing blood.

ABO group system and Rhesus factor have to be considered while transfusing red blood cells, platelets, and whole blood. Although only ABO group system has to be considered at the plasma transfusion, the Turkish Red Crescent also considers Rhesus factor while plasma's transfusion.

In this study we consider four major blood types (platelets, aphaeresis platelets, red blood cells and plasmas), with four blood groups and two Rhesus factors for a total of 32 different products.

2.2 Foundations for Blood Supply Chain System in Turkey

In Turkey, the blood supply chain is organized by a government institution, the Turkish Red Crescent (TRC) having five types of service centers: Regional Blood Centers (RBC), Temporary Regional Blood Centers (TRBC), Blood Centers (BC), Blood Stations (BS), Mobile Units (MU) and Transfusion Centers/Hospitals (H). TRBC, BC, BS and MU take blood from donors. Hospitals can take blood from donors if RBC permits. BSs and MUs send their blood to BC, and BC sends it to RBC. The assignment between centers is fixed, and deliveries of blood are done according to these assignments. Finally, RBC sends blood products to hospitals and TRBCs are assigned to RBC. There is also possibility that blood products can be transferred within the RBCs and hospitals.

2.2.1 Regional Blood Center (RBC)

The flow of blood from donor to patient is coordinated by the RBCs. The responsibilities of the RBCs are listed below:

- to test blood
- to decompose the blood in its sub products
- to store them in appropriate conditions
- to make usable blood

- to distribute blood products to hospitals.

Figure 2.2: Picture of unit blood

There are seventeen RBCs in Turkey. Their names and amount of collected blood in 2016 are stated in Table 2.2. The number of blood centers, blood stations, and provinces that are in one RBC region can be seen in Appendix A.

Table 2.2: Regional Blood Centers in Turkey

Number	Name of the RBC	Location	Collected blood amount in 2016 (*Unit blood)
1	Aegean RBC	İzmir	345,435
2	Europe RBC	İstanbul- Bağcılar	238,531
3	Central Anatolian RBC	Ankara	216,642
4	Central Mediterranean RBC	Adana	191,242
5	East Mediterranean RBC	Gaziantep	153,851
6	South Marmara RBC	Bursa	135,507
7	North Marmara RBC	İstanbul - Kartal	123,563
8	West Black Sea RBC	Düzce	114,452
9	Inner Anatolian RBC	Kayseri	107,922
10	West Mediterranean RBC	Antalya	103,665
11	Central Black Sea RBC	Samsun	92,923
12	West Anatolian RBC	Eskisehir	79,180
13	East Black Sea RBC	Trabzon	58,101
14	South West RBC	Malatya	57,787
15	East Anatolian RBC	Erzurum	51,874
16	South Anatolian RBC	Diyarbakir	39,066
17	South East RBC	Van	32,024

*Unit blood is shown in Figure 2.2.

There is a blood center in each RBC except the ones located in Istanbul-Kartal, Izmir, Ankara, and Malatya. The blood centers in these four RBCs are located at different places. The location of RBCs on Turkey map can be seen in Figure 2.3.

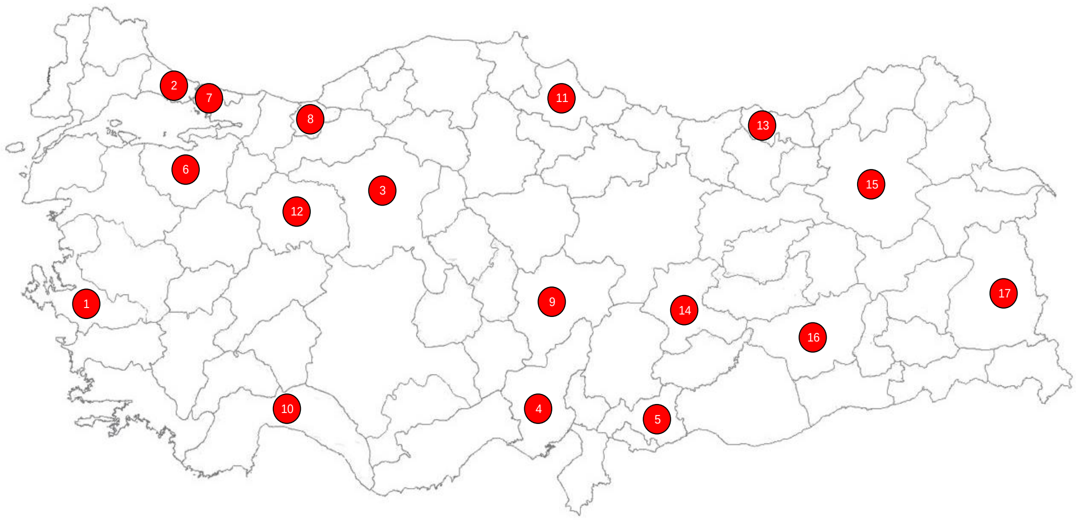

Figure 2.3: Locations of the RBCs on Turkey's Map

To satisfy the needs of the hospitals, each RBC generally uses the bloods which are donated in their own region. In addition to this, the RBCs can transfer blood from the other RBCs. They use special vehicles and planes to transfer blood products.

Each donated blood in a RBC has to be tested and used according to the results of these tests. A sample from each donated blood is sent to one of the RBCs which are located at Istanbul, Ankara, Izmir and Erzurum. The allocation of RBCs to these four RBCs is fixed. These blood samples are sent by planes of Turkish Airlines. Turkish Airlines offers free service for the Turkish Red Crescent. The results of these tests are sent to the RBC in one day.

The RBCs in Turkey do not make any estimation for the number of donated and needed blood. However, the ministry of health determines yearly and monthly goals for blood donation. The blood donation organizations are planned according to these goals. The RBCs have safety stock levels to manage the unpredicted demand for blood products.

The RBCs are also responsible to satisfy blood product demands of the hospitals. They use a software program to coordinate blood flow between RBCs and hospitals. A hospital sends its demand to a RBC via this software program. The RBC determines amount of demand to be satisfied and sends this information via the software program. If the RBC cannot satisfy all demand of the hospital, then the RBC allows hospitals to receive blood directly from the donors. The satisfied demands are sent by the vehicles of the RBC. The distribution of the demand is done at predetermined time slots and vehicle routes. Hospitals send urgent demands to the RBC. There are three ways of supplying urgent demands. First way is sending directly to the hospitals. The RBC must always hold a vehicle in the facility for the urgent demands. Second way is the hospital sends a vehicle such as ambulance, commercial vehicle, bike, etc. to receive blood from the RBC. Third way is sending demand by the vehicle routes. In this case, vehicle first visits the hospital which has urgent demand, then visits the other hospitals according to the vehicle route.

2.2.2 Temporary Regional Blood Centers (TRBC)

The TRBCs can receive blood donation and transfuse blood to patients. They do not need to take permission from the RBC to receive blood donations. The TRBCs are also hospitals and they use their own donated blood. They do not send their donated blood to RBCs or other hospitals. They can also demand blood from the RBCs once needed. There are 25 TRBCs in Turkey by the end of the 2016. The Turkish Red Crescent wants to reduce the number of TRBC to zero since the Turkish Red Crescent wants to control each donated and transfused blood.

2.2.3 Blood Centers (BC)

They are responsible for receiving blood donations and store blood and send them to the RBC. They are organizing the mobile units. Collected bloods at blood stations and mobile units are sent to the BCs. They are stored at BCs since BCs have suitable depots to store the blood. BCs normally do not distribute blood. However, BCs can distribute blood to some hospitals due to farness of hospitals to the RBCs. This distribution of blood products is also controlled by RBCs. There are 65 BCs in Turkey and each of them is assigned to one RBC. Location of the BCs in Turkey map can be seen at Figure 2.4. Red points on maps shows the RBCs and the points which have the same colors show the BCs that belong to the RBCs

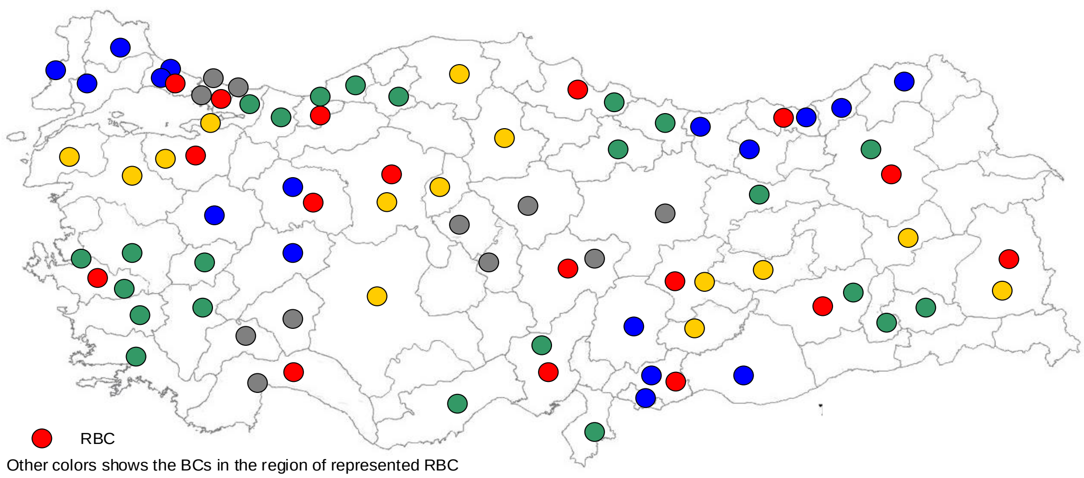

Figure 2.4: Location of the BCs on Turkey's map

Some provinces in Turkey do not have any BC. These are depended to other BC. The assignment of these provinces can also be seen in the Figure 2.5. Province which does not have BC is shown by root of arrow and depended province is shown by arrow head on the map.

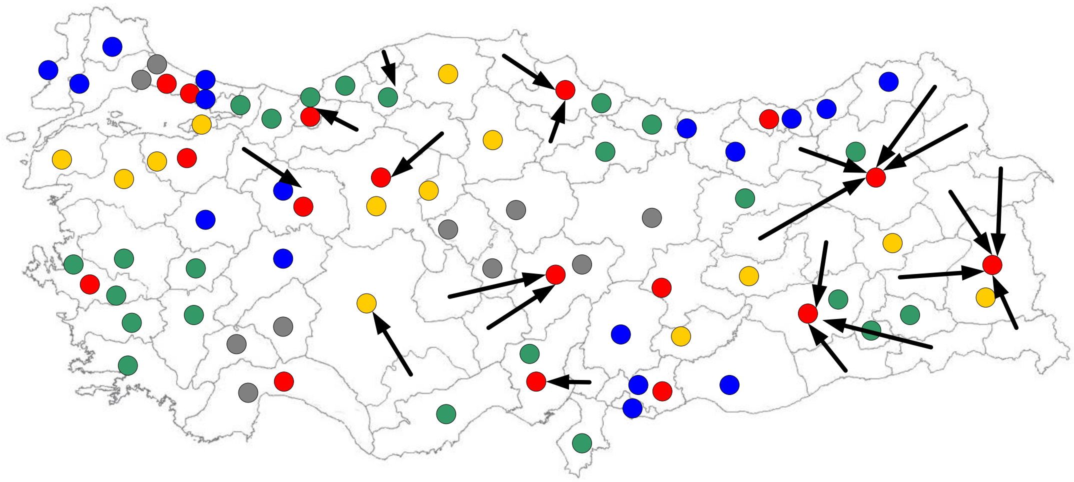

Figure 2.5: Provinces which does not have BC and depends on BC located in other province

2.2.4 Blood Stations (BS)

Blood stations are responsible for taking blood donations. They send their donated bloods to assigned BC. Like the BCs, they are organized and managed by the BCs. The only difference of BS from mobile units is its fixed location. There are 44 BSs in Turkey.

2.2.5 Mobile Units (MU)

Mobile units take bloods from the donors and send them to the BCs. They are organized and planned by the assigned BC. Their assignment to the BC is fixed.

BCs have special teams to make operation plans for the mobile units. Mobile units have to operate according to these plans. Mobile unit locations can be private or public institutions and special vehicles in the crowded public places. The factors used in planning the mobile units are listed below.

- Amount of blood can be collected (potential blood)
- Physical conditions to take blood
- Permission of institutions or municipality

Mobile units normally stay one day in a donation point. They can rarely visit two points in one day. They can also stay at a donation point more than one day depending on the quantity of potential blood collection. Special vehicles at mobile units return to the BC or parked to next position of mobile units if their position changes. If mobile unit location is far from the city center (such as some districts), then mobile unit teams will stay at this location. A mobile unit is set in about half an hour.

2.2.6 Transfusion Centers/Hospitals

The hospitals transfuse bloods to patients. In order to do this, they demand the blood from the RBC, make necessary tests, and store on their depot. They can take blood from other sources upon the permission of the RBC. In Turkey, there are about 1156 hospitals that take service from the Turkish Red Crescent.

The assignment of the hospitals to the RBC is fixed. Hospitals can demand blood at any time. There are two types of blood product demand: routine demand and urgent demand. Routine demands are for usually the hospital's inventory for future use, i.e., the RBC does not need to satisfy it soon as possible. If the demand is routine, bloods are distributed at predetermined time slots. Distribution of the demand must be done according to distance and accessibility of hospitals to the RBC. The RBC sends the amount of blood product units, to the hospitals. If the RBC gives a promise, then they have to transport it in 24 hours. Hospitals rarely receive blood from the other hospitals. Urgent demands are demands of blood product that have to be used in very short time such as injured people in accident. In urgent demand, blood is supplied as soon as possible by a vehicle.

Hospitals have to keep safety stock for the red blood cells and plasmas. They cannot determine a safety stock for platelets due to its short shelf life.

TRBC, BC, BS and mobile units take blood from donors. Hospitals can take blood from donors if the RBC permits. Blood stations and mobile units send their blood to BC, and BC sends it to the RBC. The assignment between centers is fixed, and deliveries of blood are done according to these assignments. Finally, the RBC sends blood products to hospitals and TRBCs which are assigned to the RBC. There is also possibility that blood products can be transferred between the RBCs and hospitals.

Figure 2.6 shows the blood flow between these centers. It can be observed from Figure 2.6 that TRBC, BC, BS and mobile units take blood from donors. Hospitals can take blood from donors if the RBC permits. Blood stations and mobile units send their blood to BC, and BC sends it to the RBC. The assignment between centers is fixed, and deliveries of blood are done according to these assignments. Finally, the RBC sends blood products to hospitals and TRBCs which are assigned to the RBC. There is also possibility that blood products can be transferred between the RBCs and hospitals.

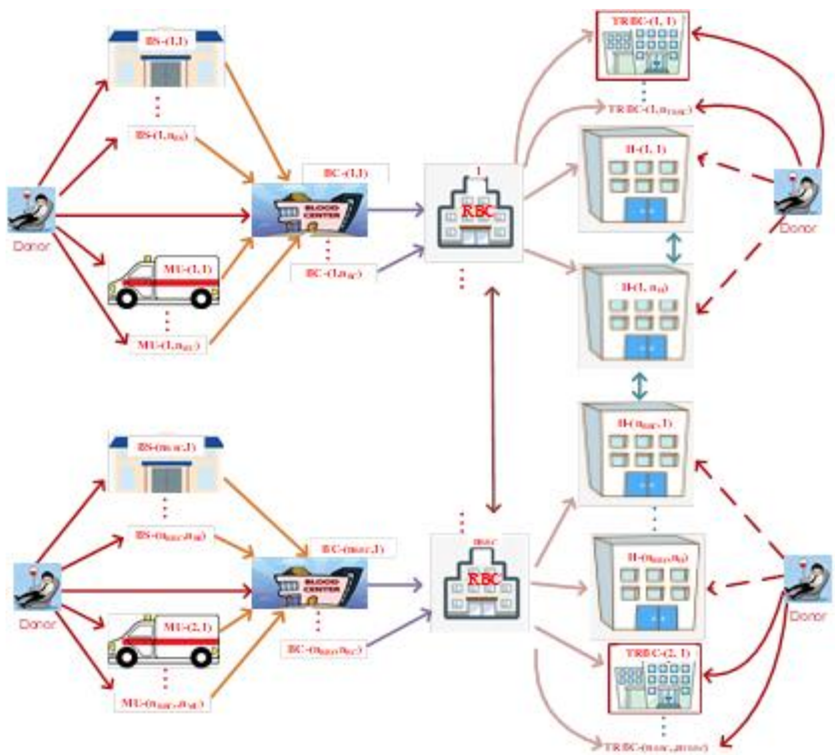

Figure 2.6: Blood flow structure

2.3 Information Flow Structure

RBCs are responsible for each BCs, BSs, mobile units, and hospitals in their region. These relations between these centers are done according to the regulations. If a hospital wants to have service from a RBC, they have to make contract with the RBC and should obey to the rules of RBC. The BCs are responsible from BSs and mobile units. They decide on their working plan. The following figure shows the information flow structure of the blood supply chain in Turkey.

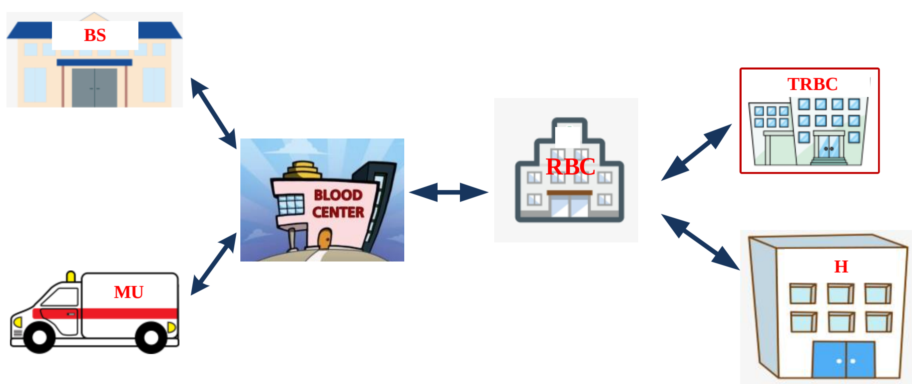

Figure 2.7: Information flow structure in Turkey

2.4 Collection of Blood

Blood donation is taken at BC, BS and mobile units. Bloods taken by BSs and mobile units are sent to the BC. Assignment of BSs and mobile units to BC is fixed. Then, BCs send collected bloods to the RBC. Detailed information about collection of blood is listed as follows:

- Taken blood from BS and mobile units are collected to the BC in the following two ways. First way, vehicle collects bloods by a route that starts at the BC, visits blood donation points and returns to the BC. Second way, collecting blood via mobile units that bring bloods to the BC when they are done.
- Vehicle collects bloods by a route in the following situations.
    - Only close points are visited.
    - If blood inventory is not enough to satisfy demand, then blood donation points are visited. Platelet is important since it must be disassembled from whole blood in 24 hours. Otherwise, whole blood loses its platelet features.
    - Supply of bloods which have negative Rhesus factor is hard. Therefore, a vehicle visits blood donation points if there are considerable amount of negative Rhesus blood.
- Collected bloods are held on special boxes and refrigerators. These holding conditions do not change the shelf life of blood.
- There is no defined time window for collecting the bloods from BS and mobile units.
- Vehicles collecting bloods can also distribute bloods to hospital, i.e., collection and distribution of blood can be done by the same route.

There are 108 blood donation gain vehicles in Turkey at the end of year 2016. Only one of them is owned by the Turkish Red Crescent and others are rental.

2.5 Distribution of Blood

The distribution management of the blood products is done by the RBCs. In general, RBCs are responsible for distributing the bloods. In some cases, the BCs can distribute blood to hospitals according to their location under the management of the RBC. Information about blood distribution system in Turkey is listed below.

- Hospitals should determine their safety stocks and they should manage their inventory accordingly. However, most of the hospitals do not have any stock.
- Hospitals can demand blood any time, but demands are distributed according to hospital's location.
- There are fixed routes for hospitals. Vehicles visits according to these routes.
- Start times of vehicles routes are fixed (the available times of the vehicles are known and not subject to change).
- More than one vehicle can be used at the same time to distribute the blood.
- Urgent demands are directly supplied from the RBC. If a vehicle has urgent and routine demands, then vehicle first visits the hospital which has urgent demand and then visits the other hospitals.
- Bloods are distributed in the special boxes and thermal protectors to hold blood in suitable temperature conditions.

2.6 Blood Flow Structure of Central Anatolian RBC

To understand the blood flow structure of the Turkish Red Crescent in a clearer way, we focus on the Central Anatolian RBC that is responsible for seven provinces; namely Ankara, Konya, Çorum, Kastamonu, Kırıkkale, Çankırı, and Karaman. There are 5 BC and 6 BS in the RBC. There are 128 hospitals which are served by Central Anatolian RBC; 66 from Ankara, 32 from Konya, 9 from Çorum, 6 from Kastamonu, 4 from Kırıkkale, 2 from Çankırı, and 4 from Karaman. Five of them are also TRBC; 4 from Ankara and 1 from Konya. The blood flow structure of the Central Anatolian RBC can be seen in the following figure.

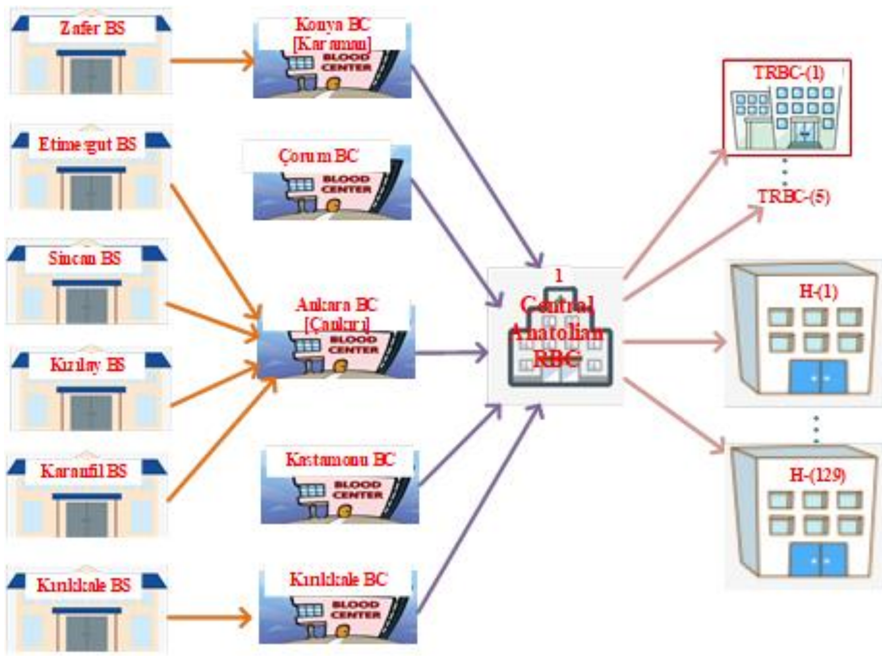

Figure 2.8: Blood flow structure of the Central Anatolian RBC

CHAPTER 3

PROBLEM 1 (P1): SOLUTION APPROACHES TO THE PROBLEM OF BLOOD DISTRIBUTION IN THE TURKISH RED CRESCENT

In this chapter, we discuss about our first problem which is introduced in Chapter 1. We start with discussing the literature and then we define the problem. Then, we give its mathematical model and present proposed algorithms. Finally, we discuss the results of our computational experiment.

3.1 Literature Review

Belien and Force (2012) present an extensive review of the literature on the supply chain management of the blood products. The review classifies the studies according to the type of blood product, problem type, solution method, hierarchical level, type of approach (exact versus heuristic), performance measures, and practical implementation/case studies. Another review of the blood supply chain is provided by Osorio et al. (2015), where only quantitative models are considered and the main features of each model pertinent to the supply chains are presented.

In this section, we present a review of the literature on supply chain management of blood products based on the problem type. In our review, we categorize the studies as inventory management, location and allocation, and distribution problems and we list the studies in Table 3.1 which includes the problem type, blood product type, hierarchical level, solution method, and practical implementation/case studies.

Inventory management studies

Gunpinar and Centeno (2015) studied the blood inventory management problem at individual hospital level. They considered uncertain demand and proposed a robust integer programming. A mathematical model for age-based transhipment problem for blood banks inventory structure was developed by Wang and Ma (2015). They used simulation operating scenarios and concluded that age-based policy under first-in-first-transhipment reduces the expired rate efficiently. Puranam et al. (2017) studied multi-period inventory management problem for red blood cells. They proposed a dynamic algorithm to minimize the total inventory cost. Daskin and Coullard (2002) studied the inventory management problem of platelets for the local blood bank in the Chicago area and solved the problem by the Lagrangian-based approach.

Location and location-allocation studies

Rusman and Rapi (2014) worked on blood bank location model for Makassar City. Their aim was to determine the location of blood banks so as to minimize the sum of blood bank opening cost and delivery cost from blood banks to hospitals. Chaiwuttisak et al. (2016) studied location-allocation problem for Thailand blood supply chain. Kaveh and Ghobadi (2017) solved p-median location-allocation problem, where a number of blood centers are located and allocated to hospitals to minimize total distance. Sha and Huang (2012) studied multi-period location-allocation problem for temporary blood locations to collect blood products in big earthquake cases. Another multi-period location-allocation problem studied by Zahiri et al. (2014) considered different blood collection scenarios. Jabbarzadeh et al. (2014) determined the location of permanent and temporary blood collection facilities, distribution scheduling of temporary blood collection facilities, and blood inventory levels at blood centers.

Jacobs et al. (1996) studied the location problem for mid-Atlantic region of American Red Cross by developing collection and distribution models. Alfonso et al. (2013) focused on the planning of the mobile collection units in Auvergne-Loire Region in France in order to minimize the total number of imported red blood cells from other regions. In another study, Alfonso et al. (2014) studied the allocation of different resource types to blood mobile and fixed site units. Ramezanian and Behboodi (2017) worked on the location of permanent and temporary donation centers to design the blood supply chain in Tehran, Iran.

Sahin et al. (2007) studied a hierarchical location-allocation problem for the Turkish Red Crescent. They basically determined the location of regional blood centers and blood centers and the allocation of the demand points to the pairs of regional blood centers and blood centers with the aim of minimizing the total weighted distance. Arvan et al. (2015) studied a supply chain network design problem by determining the location of donation points and blood banks. A stochastic bi-objective supply chain problem in disaster was studied by Fahimnia et al. (2017). Hsieh (2014) studied a multi-objective integrated inventory management and location-allocation problem for the blood banks.

Distribution studies

Distribution problems for blood products are rarely studied in the literature. Gregor et al. (1982) studied the distribution strategies to determine the number of vehicles used and inventory levels. They had three evaluation criteria: the number of emergency orders, average response time to an emergency order, the number of routine surgeries postponed. Hemmelmayr et al. (2009) studied the distribution problem of the blood products for Australian hospitals so as to minimize the total travel time. Stochastic nature of the same problem was considered by Hemmelmayr et al. (2010). Salehipour and Sepehri (2012) also studied blood distribution problem with the objective of minimizing the total waiting time.

Sahinyazan et al. (2015) focused on a selective vehicle routing problem for blood mobiles by considering the shuttles which collect blood from the bloodmobiles. Another vehicle routing problem for blood mobiles was studied by Gunpinar and Centeno (2016) by minimizing the number of blood mobiles and total distance travelled.

Table 3.1: Summary of the studies in our literature review

Reference	Problem type	Type of blood products	Hierarchical level	Solution methodology
Alfonso et al. (2013) *	Location	Red blood cell	RBC	IP
Alfonso et al. (2015) *	Location	Red blood cell	RBC	IP
Chaiwuttisak et al. (2016) *	Location Allocation	Whole blood	Supply Chain	IP
Daskin and Coullard (2002)	Inventory Location	Platelets	RBC	Lagrangian Relaxation
Fahimnia et al. (2017)	Location Allocation	Whole blood	Supply Chain	ε-constrainted and Lagrangian Relaxation
Gregor et al. (1986) *	Distribution	Blood products	RBC	Simulation
Gunpinar and Centeno (2015)	Inventory	Red blood cell, platelets	Individual Hospital	IP
Günpinar and Centeno (2016)	Distribution	Whole blood	Supply	IP
Hummelayr et al. (2009) *	Distribution	Other/Unknown	RBC	IP and VNS
Hummelayr et al. (2010) *	Distribution	Other/Unknown	RBC	IP and VNS
Jabbarzadeh et al. (2014) *	Location Allocation	Whole blood	RBC	IP
Jacobs et al. (1996) *	Location	Whole blood	Supply	IP
Kaveh and Ghodabi (2017) *	Location Allocation	Blood products	Supply Chain	Enhanced colliding bodies algorithm
Puranam et al. (2017)	Inventory	Red blood cell	Supply Chain	Dynamic programming
Ramezanian and Behboodi (2017) *	Location Allocation	Whole blood	Supply Chain	IP
Rusman and Rapi (2014) *	Location Allocation	Whole blood	RBC	IP
Sahin et al. (2007) *	Location Allocation	Whole blood	Supply Chain	IP
Sahinyazan et al. (2015)	Distribution	Whole blood	Supply	IP based heuristic
Selahipour and Sepehri (2012) *	Distribution	Blood products	RBC	IP and hybrid heuristic
Sha and Huang (2012) *	Location Allocation	Whole blood	RBC	Lagrangian Relaxation based
Wang and Ma (2015)	Inventory	Blood products	RBC	Simulation
Zahiri et al. (2014)	Location Allocation	Whole blood	Supply Chain	IP
Arvan et al. (2015) *,**	Location Allocation	Whole blood, blood products	Supply Chain	IP
Hsieh (2014) **	Inventory	Whole blood	Supply	Genetic algorithm
Our Problem *,**	Distribution	Blood products	RBC	Decomposition based heuristic

IP: Integer Programming, VNS: Variable Neighborhood Search
*: practical application, **: multi-objective

In this study, we consider the allocation of the blood products along with their distribution to the hospitals. Hence we consider a new category, *distribution - product allocation*. Different from all other studies, we take irradiation centers into account in our distribution network, which leads to precedence relations between nodes. We inspired from Sahin et al.'s (2007) study, which considers a practical situation in the Turkish Red Crescent and studies the hierarchical location-allocation problem there.

3.2 Our problem - the blood distribution system

Our observations on the blood supply chain system in Turkey have motivated us to deal with the blood distribution system. We were inspired by a practical application and tested the performance of our solution approaches in the real life problem instances.

In the current blood distribution system, the RBCs should determine the amounts to be shipped to the hospitals by considering its stock levels, demand and locations of the hospitals, and deadlines of the demand. Currently, the amount to be shipped to each hospital and the route to be followed by each vehicle are determined by the worker and driver experiences without any systematic approach. Generally, the vehicles first visit the urgent demand points and then the routine (non-urgent) demand points, thereby leading to inefficient routes. Furthermore, some issues like availability of the products and the vehicles, vehicle capacities, length of the distribution period and maximum travel time of the vehicles are not considered by the RBC while preparing the distribution plan.

Our problem can be viewed as a multi-period blood distribution problem where the urgent demands and routine demands are considered simultaneously. Blood products can be irradiated or non-irradiated (normal), and are distributed from the RBC in three different periods of a day. Hence, we solve three sequential single-period vehicle scheduling problems, starting from period 1, and apply rolling-time horizon approach where the solution of the vehicle scheduling problem for a distribution period is used as an input for the next distribution period. We treat the unsatisfied routine demand of a distribution period as a new demand for the next distribution period.

In the blood distribution system of a single period, we have a set of vehicles available at the RBC to deliver blood products through a set of routes to a set of demand points, i.e., hospitals. The irradiated products are first delivered to the irradiation centers and then to the hospitals. We call these irradiation centers and hospitals as *nodes* and define a *vehicle route* as a sequence of visited nodes. Figure 3.1 illustrates a distribution system in a period for a case where there are eight hospitals, one irradiation center, and two vehicles. The first vehicle is loaded with the normal demand of hospitals 3, 6 and 7. It first visits hospital 7 and then goes to the hospitals 6 and 3 in its route and returns to the RBC. The second vehicle is loaded with the normal demand of hospitals 1 and 2 and the blood products to be irradiated for the hospitals 2 and 3. This second vehicle first visits hospital 1, and then goes to the irradiation center at which the blood products for hospitals 2 and 3 are irradiated. Then, the second vehicle visits hospital 3 and unloads the irradiated products of this hospital. Finally, hospital 2 is visited by the second vehicle, and the normal and irradiated products of this hospital are unloaded. The route of the second vehicle finishes at the RBC. As it is seen from the figure, the demand of hospitals 4, 5 and 8 are not satisfied within the distribution period.

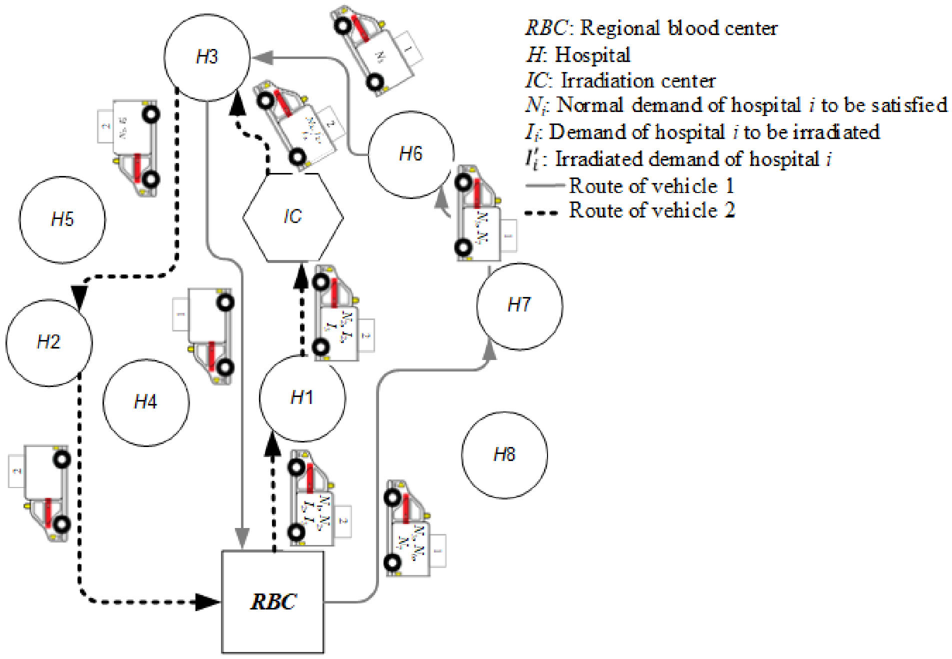

Figure 3.1: Distribution system in a period

3.3 Mathematical model

In this section, we present the mathematical programming formulation of our blood distribution problem. We solve the problem over a distribution period of pre-specified length. We assume that there are $N + 1$ nodes, where one node is for the RBC, IC nodes are for the irradiation centers and the remaining $N - IC$ nodes are for the hospitals. Among these nodes ($i = 0,1, \dots, N$), $i = 0$ is the regional blood center (RBC), $i = 1,2, \dots, IC$ are for the irradiation centers, and $i = IC + 1, IC + 2, \dots, N$ are for the hospitals. U is the set of hospitals with urgent demands (there are $|U|$ hospitals in U).

The travel time between nodes i and j is T_{ij} time units. a_i and b_i are the quantity independent and per unit loading and unloading times for node i, respectively. w_i is the weight of hospital i indicating its relative importance. DD_i is the deadline for hospital i.

There are P product types, and A_p units of product p ($p = 1,2,..,P$) are available at the RBC. The amount of normal product p demanded by hospital i is D_{ip}, and the amount of irradiated product p demanded by hospital i is ID_{ip} units.

There are V vehicles, carrying blood products, with identical speed and different capacity. Carrying capacity of vehicle v ($v = 1,2,..,V$) is C_v units. Vehicle v becomes available at time EA_v and remains continuously available. Each vehicle starts and ends its route at the RBC. The maximum travel time of vehicle v is MT_v time units. Hence, the maximum of the MT_v values is an upper bound on the length of the distribution period.

We make the following additional assumptions to construct the mathematical model:

- All parameters are known with certainty and not subject to any change, i.e., the system is deterministic and static.
- Each vehicle can visit each node.
- Each node can be visited by more than one vehicle.
- Blood products to be irradiated are first delivered to the irradiation centers and then to the hospitals.
- Irradiation centers are always available.
- Urgent demand of a hospital should be satisfied fully before the deadline. However, routine demand of the hospitals might be unsatisfied or partially satisfied.

Our problem is to deliver blood products (normal and irradiated) via a set of vehicles to a set of hospitals within the distribution period. Thus, N is an upper bound on the number of positions (nodes) visited on each vehicle route, where k is an index for positions ($k = 1,2,\dots,N$) in this route.

***Decision variables*:**

$$X_{ikv} = \begin{cases} 1 & \text{if vehicle } v \text{ visits node } i \text{ at position } k \text{ of its route} \\ 0 & \text{otherwise} \end{cases}$$

$R_{ipv} =$ Amount of normal product p carried to hospital i by vehicle v

$IR_{ijpv} =$ Amount of irradiated product p processed by irradiation center j ($j = 1,\dots,IC$) and carried to hospital i ($i = IC+1, IC+2,\dots,N$) by vehicle v

$$Y_{ijv} = \begin{cases} 1 & \text{if irradiation center } j \text{ is used to process the demand of hospital } i \text{ carried by vehicle } v \\ 0 & \text{otherwise} \end{cases}$$

$S_{ikv} =$ Arrival time of vehicle v to node i at its k^{th} position of its route

$T'_{ij} =$ Travel time between nodes i and j

***Objective function*:**

We consider two objectives in hierarchy: first maximizing the total weighted demand satisfied, $z_1 = \sum_{v=1}^{V}\sum_{p=1}^{P}\sum_{i=IC+1}^{N} w_i\left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpk}\right)$ and then minimizing the total travel time, $z_2 = \sum_{j=0}^{N}\sum_{i=0}^{N} T'_{ij}$.

Among the solutions with the maximum satisfied demand, we select the one with the minimum total travel time. In doing so, we create the following composite objective function where the second objective is weighted by a sufficiently small positive coefficient ε.

$$\text{Max } z = z_1 - \varepsilon z_2 = \sum_{v=1}^{V} \sum_{p=1}^{P} \sum_{i=IC+1}^{N} w_i \left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpk} \right) \\ -\varepsilon \sum_{j=0}^{N} \sum_{i=0}^{N} T'_{ij} \tag{3.1}$$

The coefficient ε should be set small enough so that the amount of satisfied demand should not decrease even for the largest possible value of the total travel time. That is,

$$z_1^* - \varepsilon z_2^{max} \geq z_1^* - 1 - \varepsilon z_2^{min} \tag{3.2}$$

where z_2^{min} and z_2^{max} are the minimum and maximum possible total time values, respectively. This follows;

$$1 \geq \varepsilon z_2^{max} - \varepsilon z_2^{min} \tag{3.3}$$

$$\varepsilon \leq \frac{1}{z_2^{max} - z_2^{min}} \, . \tag{3.4}$$

As z_2^{min} and z_2^{max} values are not known, we use the respective lower and upper bounds, in their places. On the total time travelled, we use a lower bound LB_2 and an upper bound UB_2 in place of z_2^{min} and z_2^{max}, respectively, and set ε according to the following equation

$$\varepsilon = \frac{1}{UB_2 - LB_2 + 1}. \tag{3.5}$$

It is clear that UB_2 is the total of the maximum travel times of the vehicles. Thus, the following formula is used to obtain UB_2:

$$UB_2 = \sum_{v=0}^{V} (MT_v - EA_v). \tag{3.6}$$

Recall that we assume at least one hospital must be visited in a vehicle route and this hospital is the closest one to the RBC and the vehicle immediately returns to the RBC. Thus, the following formula is used to obtain LB_2:

$$LB_2 = \min_{j=IC+1, IC+2 \ldots, n} \{ T_{0j} + T_{j0} \} \tag{3.7}$$

***Constraints*:**

$$\sum_{i=1}^{N} X_{ikv} \leq 1 \qquad \text{for } k = 1,2,\dots,N;\ v = 1,2,\dots,V \tag{3.8}$$

$$\sum_{i=1}^{N} X_{i,k+1,v} \leq \sum_{i=1}^{N} X_{ikv} \qquad \text{for } k = 1,2,\dots,N;\ v = 1,2,\dots,V \tag{3.9}$$

$$\sum_{v=1}^{V} \sum_{k=2}^{N} X_{0kv} = \sum_{i=0}^{N} \sum_{v=1}^{V} X_{i2v} \tag{3.10}$$

$$\sum_{k=1}^{N} k \cdot X_{ikv} \leq \sum_{k=1}^{N} k \cdot X_{0kv} \qquad \text{for } i = 1,2,\dots,N;\ v = 1,2,\dots,V \tag{3.11}$$

$$\sum_{p=1}^{P} \left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv} \right) \leq \left(\sum_{p=1}^{P} \left(D_{ip} + ID_{ip} \right) \right) \sum_{k=1}^{N} X_{ikv}$$

$$\text{for } i = IC+1, IC+2, \dots, N;\quad v = 1,2,\dots,V \tag{3.12}$$

$$\sum_{v=1}^{V} R_{ipv} \leq D_{ip} \qquad \text{for } i = IC+1, IC+2, \dots, N;\quad p = 1,2,\dots,P \tag{3.13}$$

$$\sum_{v=1}^{V} \sum_{j=1}^{IC} IR_{ijpv} \leq ID_{ip} \qquad \text{for } i = IC+1, IC+2, \dots, N;\quad p = 1,2,\dots,P \tag{3.14}$$

$$\sum_{v=1}^{V} \sum_{i=IC+1}^{N} \left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv} \right) \leq A_p \qquad \text{for } p = 1,2,\dots,P \tag{3.15}$$

$$\sum_{p=1}^{P} \sum_{i=IC+1}^{N} \left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv} \right) \leq C_v \qquad \text{for } v = 1,2,\dots,V \tag{3.16}$$

$$\sum_{k=1}^{N} X_{ikv} + \sum_{k=1}^{N} X_{jkv} \geq 2\left(Y_{ijv} + \sum_{k=1}^{N} X_{ikv} - 1 \right)$$

$$\text{for } i = IC+1, IC+2, \dots, N; j = 1,2,\dots,IC;\quad \sum_{p=1}^{P} ID_{ip} > 0\ ; v = 1,2,\dots,V \tag{3.17}$$

$$S_{i1v} \geq (T_{0i} + EA_v)X_{i1v} \qquad \text{for } i = 1,2,\dots,N;\ v = 1,2,\dots,V \tag{3.18}$$

$$S_{jkv} \geq S_{i,k-1,v} + \left(a_i + b_i \sum_{p=1}^{P}\left(R_{ipv} + \sum_{j'=1}^{IC} IR_{ij'pv}\right)\right) + T_{ij} -$$
$$\left[a_i + b_i \sum_{p=1}^{P}\left(D_{ip} + ID_{ip}\right) + \max_{v'}\{MT_{v'}\}\right] \times (2 - X_{i,k-1,v} - X_{jkv})$$

$$\text{for } i = IC+1, IC+2,\dots,N;\ v = 1,2,\dots,V;$$

$$j = 1,2,\dots,N; i \neq j; k = 1,2,\dots N \tag{3.19}$$

$$S_{jkv} \geq S_{i,k-1,v} + \left(a_i + b_i \sum_{i'=1}^{N} \sum_{p=1}^{P} IR_{i'ipk}\right) + T_{ij} -$$
$$\left[a_i + b_i \sum_{i'=1}^{N} \sum_{p=1}^{P} ID_{ip} + \max_{v'}\{MT_{v'}\}\right] \times (2 - X_{i,k-1,v} - X_{jkv})$$

$$\text{for } i = 1,2,\dots,IC; j = 1,2,\dots,N; i \neq j;$$

$$k = 1,2,\dots N;\ v = 1,2,\dots,V \tag{3.20}$$

$$\sum_{k=1}^{N} k \cdot X_{ikv} \geq \sum_{k=1}^{N} k \cdot X_{jkv} - N\left(1 - Y_{ijv}\right)$$

$$\text{for } i = IC+1, IC+2,\dots,N;\ \sum_{p=1}^{P} ID_{ip} > 0;$$

$$j = 1,2,\dots,IC;\ v = 1,2,\dots,V \tag{3.21}$$

$$S_{ikv} \leq DD_i \qquad \text{for } i = IC+1, IC+2,\dots,N; k = 1,2,\dots N;$$

$$v = 1,2,\dots,V \tag{3.22}$$

$$S_{0kv} \leq MT_v \qquad \text{for } k = 1,2,\dots,N; v = 1,2,\dots,V \tag{3.23}$$

$$\frac{\sum_{p=1}^{P} IR_{ijpv}}{\sum_{p=1}^{P} ID_{ip}} \leq Y_{ijv} \qquad \text{for } i = IC+1, IC+2,\dots,N; \sum_{p=1}^{P} ID_{ip} > 0;$$

$$j = 1,2,\dots,IC; v = 1,2,\dots,V \tag{3.24}$$

$$T'_{ij} \geq T_{ij}\left(X_{i,k-1,v} + X_{jkv} - 1\right) \quad \text{for } i,j = 0,1,\dots,N; j \neq i;\ k = 2,3,\dots,N;$$

$$v = 1,2, \dots, V \quad (3.25)$$

$$T'_{0j} \geq T_{0j} X_{j1v} \qquad \text{for } j = 1,2, \dots, N;\ v = 1,2, \dots, V \quad (3.26)$$

$$\sum_{v=1}^{V} R_{ipv} = D_{ip} \qquad \text{for } i = IC + 1, IC + 2, \dots, N; p = 1,2, \dots, P;$$

$$i \in U \quad (3.27)$$

$$\sum_{v=1}^{V} \sum_{j=1}^{IC} IR_{ijpv} = ID_{ip} \qquad \text{for } i = IC + 1, IC + 2, \dots, N; p = 1,2, \dots, P;$$

$$i \in U \quad (3.28)$$

$$X_{ikv}, Y_{ijv} \in \{0,1\} \qquad \text{for } i = 0,1, \dots, N;\ k = 1,2, \dots, N;$$

$$v = 1,2, \dots, V; j = 1,2, \dots, IC \quad (3.29)$$

$$S_{ikv}, T'_{ij} \geq 0 \qquad \text{for } i, j = 0,1, \dots, N;\ k = 1,2, \dots, N;$$

$$v = 1,2, \dots, V \quad (3.30)$$

$$R_{ipv}, IR_{ijpv} \geq 0 \text{ and integer} \quad \text{for } i = IC + 1, IC + 2, \dots, N; j = 1,2, \dots, IC;$$

$$p = 1,2, \dots, P;\ v = 1,2, \dots, V \quad (3.31)$$

Constraint set (3.8) guarantees that each vehicle visits at most one node in any position on its route. Constraint set (3.9) ensures that if a node is assigned to a position, then there must be an assignment to the previous position. Constraint set (3.10) states that the total number of vehicles departing from the RBC is equal to the total number of vehicles used. Constraint set (3.11) guarantees the RBC is visited as the last node by all vehicles. Constraint set (3.12) ensures that the demand of a hospital can be satisfied by a vehicle that visits the hospital. Constraint sets (3.13) and (3.14) state that the amount of products carried to the hospitals does not exceed the normal and irradiated demand, respectively. Constraint set (3.15) guarantees that the amount of products carried is limited by the amount available at the RBC. The capacities of the vehicles are considered in Constraint set (3.16). Constraint set (3.17) ensures that if the demand of a hospital is processed by an irradiation center, then the hospital and the irradiation center should be visited by the same vehicle. Constraint sets (3.18), (3.19), and (3.20) calculate the arrival time of vehicle v visiting node i in the first position, the next node after leaving node i, and the next node after leaving irradiation center i, respectively. Constraint set (3.21) guarantees that a hospital with demand for the irradiated products should be visited after the irradiation center serves to this hospital's demand. Constraint set (3.22) ensures that the arrival time of the vehicle to hospital i should not exceed the deadline imposed by the hospital. Constraint set (3.23) guarantees that the arrival time of the vehicle to the RBC does not exceed its maximum travel time. Constraint set (3.24) assigns hospital i to irradiation center j if the hospital i is served by irradiation center j. Constraint set (3.25) calculates the travel time between two successive nodes visited by a vehicle. Constraint set (3.26) calculates the travel time between the RBC and the first visited node. Constraint sets (3.27) and (3.28) guarantee that the urgent demands and irradiated urgent demands are satisfied. Constraint sets (3.29), (3.30), and (3.31) represent the binary, non-negativity, and integrality restrictions, respectively.

We now present some properties of the optimal solution whose incorporation (via additional constraints) may reduce the size of the search space, hence improve the efficiency of the MILP. Below, we state each of these properties and their associated constraints.

Property 1: If any hospital's irradiated product demand is satisfied by any vehicle, then this vehicle cannot visit the RBC in the second position of its route.

$$X_{02v} \leq 2 - Y_{ijv} - \sum_{k'=1}^{K} X_{i,k',v} \quad \text{for } i = IC+1, IC+2, \dots, N;$$

$$j = 1,2, \dots, IC; v = 1,2, \dots, V \qquad (3.32)$$

Property 2: If any hospital's irradiated product demand is satisfied by any vehicle, then this vehicle cannot visit this hospital in the first position. In other words, the irradiation center is visited before all hospitals served by that center.

$$X_{i1v} \leq 1 - Y_{ijv} \qquad \text{for } i = IC+1, IC+2, \dots, N;$$

$$j = 1,2, \dots, IC; v = 1,2, \dots, V \tag{3.33}$$

Property 3: The position of an irradiation center in the visiting sequence of the vehicle is less than the number available positions minus one plus the number of hospitals served by this irradiation center.

$$\sum_{k=1}^{K} kX_{jkv} \leq K - \left(1 + \sum_{i=IC+1}^{N} Y_{ijv}\right)$$

$$\text{for } j = 1,2, \dots, IC; v = 1,2, \dots, V \tag{3.34}$$

Property 4: A vehicle cannot visit the RBC at position 1.

$$\sum_{v=1}^{V} X_{01v} = 0 \tag{3.35}$$

Any hospital cannot be visited at the final position.

$$\sum_{v=1}^{V} \sum_{i=IC+1}^{N} X_{iKv} = 0 \tag{3.36}$$

Any irradiation center cannot be visited at position $K-1$.

$$\sum_{v=1}^{V} \sum_{j=1}^{IC} X_{j,K-1,v} = 0 \tag{3.37}$$

If the earliest arrival time of a vehicle to any hospital at any position is greater than the deadline of the hospital, then the hospital cannot be visited after this position.

$$\sum_{k'=k}^{K} X_{ik'v} = 0 \text{ for } i = IC+1, IC+2, \dots, N; v = 1,2, \dots, V;$$

$$k = 1,2, \dots, K; EA_v + \sum_{l=1}^{k} A_{[l]} - a_i > DD_i \qquad (3.38)$$

where $A_i = \min_j \{T_{ji} + a_i\}$ and A_i values are arranged as $A_{[1]} \leq \cdots \leq A_{[L]}$.

We add (3.35) through (3.38) as each equates to zero and obtain the following constraint set.

$$\sum_{v=1}^{V} X_{01v} + \sum_{v=1}^{V} \sum_{i=1}^{N} X_{iKv} + \sum_{v=1}^{V} \sum_{j=1}^{IC} X_{j,K-1,v} + \sum_{v=1}^{V} \sum_{i=IC+1}^{N} \sum_{k=1 \text{ and } (EA_v + \sum_{l=1}^{k} A_{l-a_i}) > DD_i}^{K} \sum_{k'=k}^{K} X_{ik'v} = 0 \qquad (3.39)$$

Our mathematical model is explained by the objective function (3.1) and the constraint sets (3.8) through (3.34), and the constraint (3.39).

3.4 Proposed heuristic algorithms

The size of the MILP model increases exponentially with an increase in the number of nodes or vehicles. We verify by computational tests that the model cannot solve even small sized problem instances. Recognizing this, we present two decomposition-based heuristic algorithms. Below are the detailed descriptions of our heuristic algorithms.

3.4.1 Decomposition-based heuristic algorithms

Our heuristic algorithms that decompose the problem into sub-problems involve five phases: *Finding Eligible Vehicles for Hospitals*, *Finding an Initial Solution (Vehicle Schedule)*, *Improvement by Inserting Nodes to Vehicle Routes*, *Building a New Vehicle Schedule by Perturbing the Current Solution*, and *Improvement by Swapping (Pairwise Interchanging) Nodes*. The algorithms only vary by their phase flows. We use a predetermined critical load ratio (*CLR*) to improve the chance of assigning one vehicle to the hospital. The heuristics start with critical load ratio (*CLR*) of 0. *NI* is the number of iterations used by the algorithms.

We round up all product quantities (R_{ipv} and IR_{ijpv} values) returned by the heuristic algorithm to their upper integer values. If the solution is not feasible with rounded R_{ipv} and IR_{ijpv} values, we reduce the shipment to a hospital with the minimum weight by one unit until feasibility is maintained.

The flowcharts of the heuristic algorithms 1 and 2 are given in Figures 3.2 and 3.3, respectively.

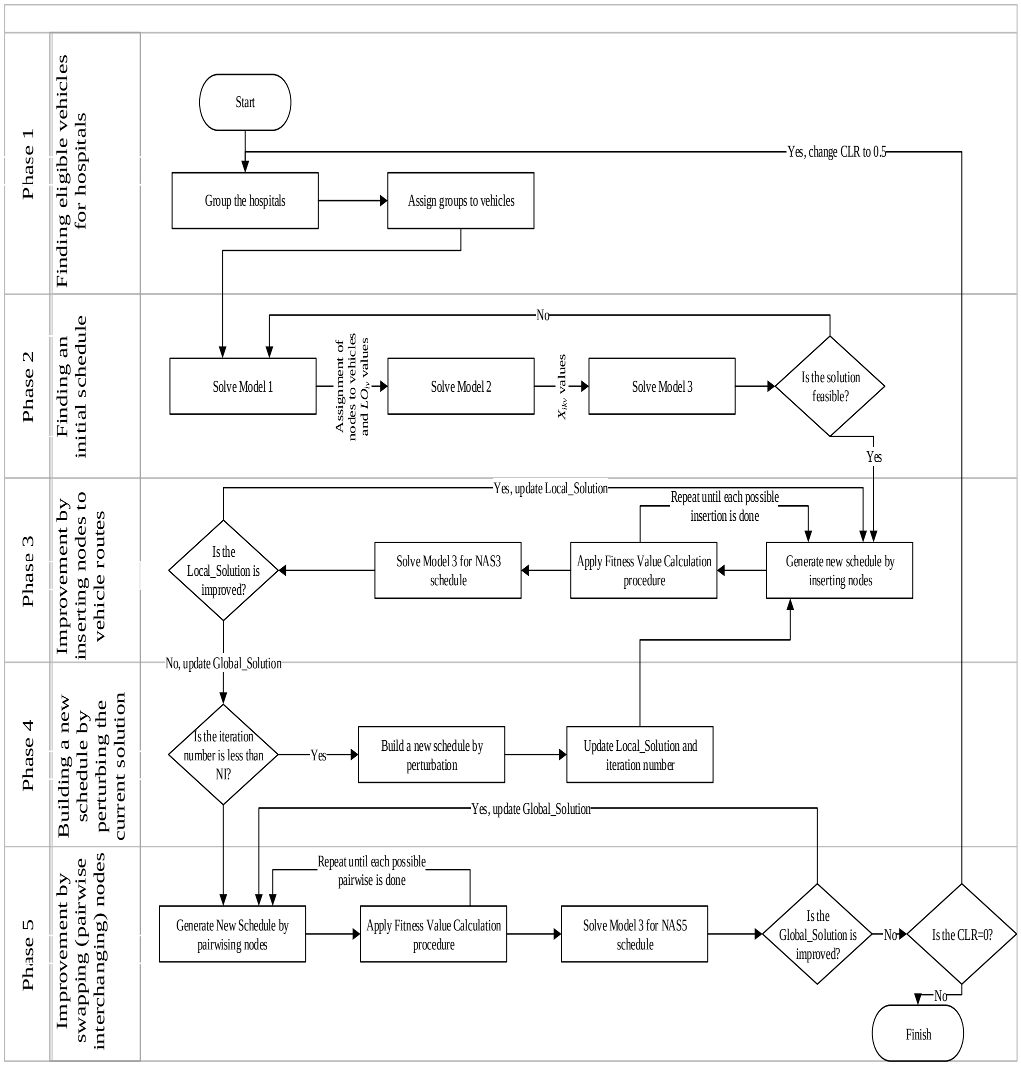

Figure 3.2: Flowchart of the Heuristic Algorithm 1

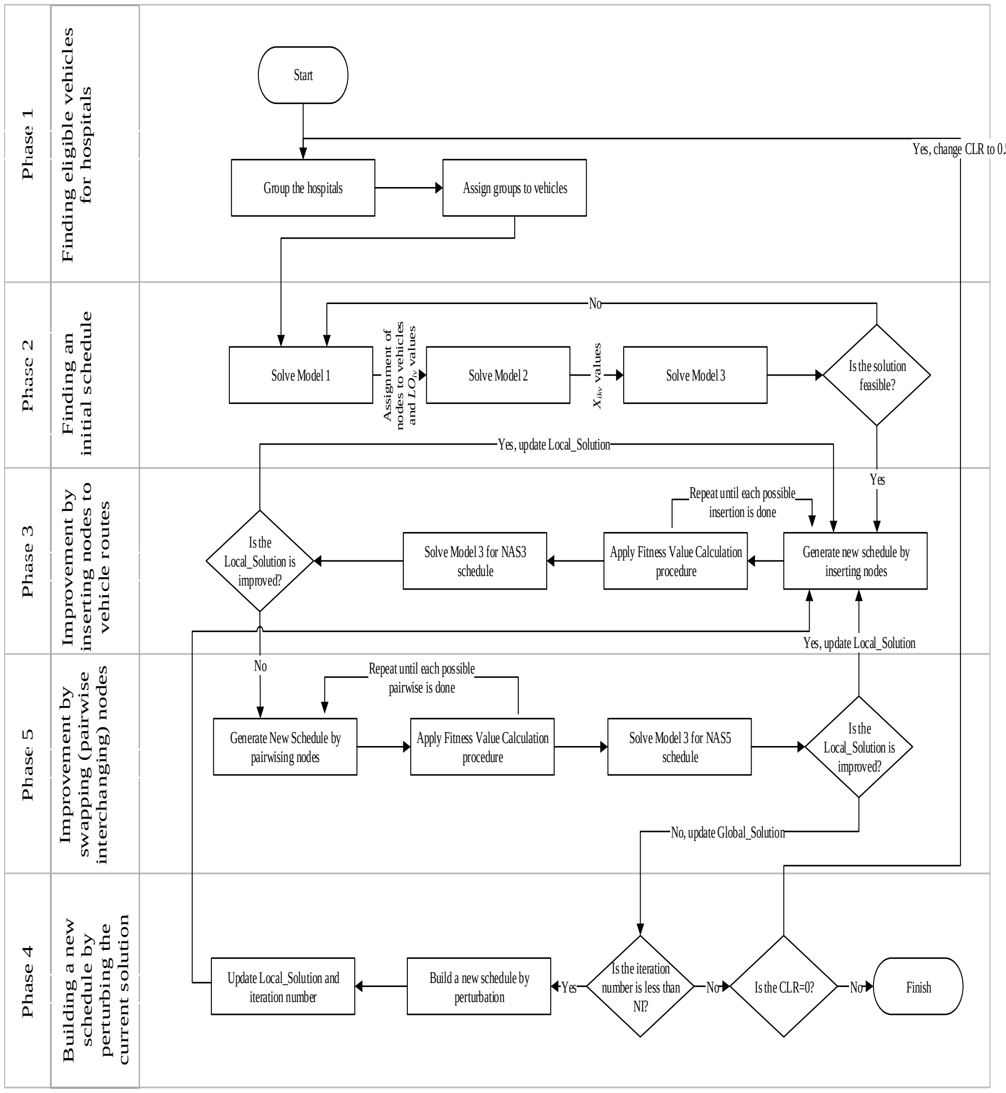

Figure 3.3: Flowchart of the Heuristic Algorithm 2

Once we relax the integrality requirements on all binary variables, the resulting linear programming (LP) model makes product assignments to the hospitals. To describe this LP model, so called Model M1, we use the same parameters and decision variables of our original model and introduce a new parameter (Z_{iv}) and two decision variables (UC_v^+, UC_v^-) as defined below:

$$Z_{iv} = \begin{cases} 1 & \text{if hospital } i \text{ can be assigned to vehicle } v \\ 0 & \text{otherwise} \end{cases}$$

UC_v^+ = Capacity over-usage (deviation) of vehicle v from the average capacity usage of the vehicles

UC_v^- = Capacity under-usage (deviation) of vehicle v from the average capacity usage of the vehicles

Model M1 becomes:

$$\text{(M1) Max } z = \sum_{v=1}^{V} \sum_{p=1}^{P} \sum_{i=IC+1}^{N} w_i \left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpk} \right) - \varepsilon \sum_{v=1}^{V} (UC_v^+ + UC_v^-) \tag{3.40}$$

Subject to

Constraints (3.13, 3.14, 3.15, 3.16, 3.27, 3.28) and

$$\sum_{p=1}^{P} \left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv} \right) (1 - Z_{iv}) = 0 \quad \text{for } i = IC+1, IC+2, \dots, N;\ v = 1,2,\dots,V \tag{3.41}$$

$$\sum_{p=1}^{P} \sum_{i=IC+1}^{N} \left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv} \right) - UC_v^+ + UC_v^- = \frac{\sum_{v'=1}^{V} \sum_{p=1}^{P} \sum_{i=IC+1}^{N} \left(R_{ipv'} + \sum_{j=1}^{IC} IR_{ijpv'} \right)}{V} \quad \text{for } v = 1,2,\dots,V \tag{3.42}$$

$$R_{ipv}, IR_{ijpv}, UC_v^+, UC_v^- \geq 0 \quad \text{for } i = IC+1, IC+2, \dots, N;\ p = 1,2,\dots,P;\ j = 1,2,\dots,IC;\ v = 1,2,\dots,V \tag{3.43}$$

Our primary objective (3.40) is to maximize the total weighted demand satisfied. The total capacity usage deviation is minimized as the secondary objective weighed by a very small positive number $\varepsilon = 0.001$. Constraint set (3.41) ensures that a vehicle cannot visit a hospital if it is not in the hospital's set of eligible vehicles. Constraint set (3.42) calculates the capacity usage deviations of the vehicles. Constraint set (3.43) ensures non-negativity of the variables.

Note that Model M1 can be solved in polynomial time since it is an LP. It returns R_{ipv} and IR_{ijpv} values explicitly and, the assignment of nodes to the vehicles and hospitals to the irradiation centers implicitly.

Given the allocation of the products to the hospitals and vehicles by Model M1, the new problem turns into V independent single-machine sequence-dependent scheduling problems with additional precedence and fixed-position restrictions. We construct a mixed integer linear programming model (MILP) where the minimization of the total tardiness is the primary objective and the minimization of the total travel time is the secondary objective by weighing it with a very small positive number $\varepsilon = 0.001$.

We use R_{ipv} and IR_{ijpv} values returned by Model M1 to find the total loading and unloading times (LO_{iv}) for each node as:

$$LO_{iv} = a_i + b_i \sum_{i'=IC+1}^{N} \sum_{p=1}^{P} IR_{i'ipk} \quad \text{for } i = 1,2,\dots,IC; \quad v = 1,2,\dots,V \tag{3.44}$$

$$LO_{iv} = a_i + b_i \sum_{p=1}^{P} \left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv} \right) \quad \text{for } i = IC+1, IC+2,\dots,N; \quad v = 1,2,\dots,V \tag{3.45}$$

We define the following additional sets, parameters and decision variables for our MILP Model M2.

Additional sets and parameters:

$NO_v =$ Set of nodes (including the RBC) assigned to vehicle v

$|NO_v| =$ Total number of nodes assigned to vehicle v

$$Y_{ij} = \begin{cases} 1 & \text{if irradiation center } j \text{ is used to process the demand of hospital } i \\ 0 & \text{otherwise} \end{cases}$$

$LO_i =$ Total loading and unloading time for node i

Additional decision variables:

$$XX_{ik} = \begin{cases} 1 & \text{if vehicle visits node } i \text{ at position } k \\ 0 & \text{otherwise} \end{cases}$$

$S_{ik} =$ Arrival time of a vehicle to node i at position k

$TA_i =$ Tardiness of node i

Model M2 becomes:

(M2) Min $z = \sum_{i \in NO_v} TA_i + \epsilon \sum_{j \in NO_v} \sum_{i \in NO_v} T'_{ij}$ (3.46)

Subject to

$$\sum_{i \in NO_v} XX_{ik} = 1 \qquad \text{for } k = 1,2, \ldots, |NO_v| \tag{3.47}$$

$$\sum_{k=1}^{|NO_v|} XX_{ik} = 1 \qquad \text{for } i \in NO_v \tag{3.48}$$

$$S_{i1} \geq (T_{0i} + EA_v) XX_{i1} \qquad \text{for } i \in NO_v \tag{3.49}$$

$$S_{jk} \geq S_{i,k-1} + LO_i + T_{ij} - (LO_i + T_{ij} + MT_v) \times \left(2 - XX_{i,k-1} - XX_{jk}\right) \text{ for } i,j \in NO_v; i \neq j; \; k = 1,2, \ldots, |NO_v| \tag{3.50}$$

$$\sum_{k=1}^{|NO_v|} S_{ik} - DD_i \leq TA_i \quad \text{for } i \in NO_v; i \neq 0 \tag{3.51}$$

$$S_{0v_k} - MT_v \leq TA_0 \tag{3.52}$$

$$\sum_{k=1}^{|NO_v|} k \cdot XX_{ik} \geq Y_{ij} \sum_{k=1}^{|NO_v|} k \cdot XX_{jk} \quad \text{for } i \in NO_v; i \neq 0;$$

$$j = 1,2, \dots, IC; i \neq j \tag{3.53}$$

$$XX_{0,|NO_v|} = 1 \tag{3.54}$$

$$T'_{ij} \geq T_{ij}\left(X_{i,k-1} + XX_{jk} - 1\right) \text{ for } i, j \in NO_v; j \neq i;$$

$$k = 2,3, \dots, |NO_v| \tag{3.55}$$

$$T'_{0j} \geq T_{0j} XX_{j1} \qquad \text{for } j \in NO_v \tag{3.56}$$

$$XX_{ik} \in \{0,1\} \qquad \text{for } i \in NO_v;\ k = 1,2, \dots, |NO_v| \tag{3.57}$$

$$S_{ik} \geq 0 \qquad \text{for } i \in NO_v;\ k = 1,2, \dots, |NO_v| \tag{3.58}$$

The objective (3.46) is to minimize the total tardiness while selecting the solution with minimum total travel time where we set ε to 0.001. Constraint sets (3.47) and (3.48) represent the assignment constraints for node and positions in the vehicle route, respectively. The arrival time of a vehicle to node i in the first position and visiting node j immediately are found by constraint sets (3.49) and (3.50). The tardiness values of the hospitals and the RBC are calculated by constraint sets (3.51) and (3.52). Constraint set (3.53) guarantees that the hospitals with irradiated demands should be visited after the irradiation center. Constraint set (3.54) assigns the RBC to the last position. Constraint set (3.55) calculates the travel time between two successively visited nodes, whereas the constraint set (3.56) calculates the travel time between the RBC and the first visited node. Constraint sets (3.57) and (3.58) represent the binary and non-negative variables, respectively.

For each vehicle route, if the total tardiness returned by Model M2 is zero, then it is an optimal solution for the first objective function z_1 of the original problem for continuous R_{ipv} and IR_{ijpv} values; otherwise, the solution of Model M2 provides a promising route for this vehicle.

Finding an optimal solution to Model M2 might require exponential effort due to the existence of the binary variables. Recognizing this fact, we aim approximate solutions, hence we develop a heuristic procedure. Recall that, Model M2 is similar to the single-machine tardiness problem with sequence-dependent setup times for which an *Iterative Local Search* (ILS) algorithm is proposed by Arroyo et al. (2009). We modify this algorithm in the following ways, and use its solution instead of solving Model M2 directly:

- We set the RBC to the end of the sequence.
- We set the objective function value to a sufficiently large number if any precedence constraint between an irradiation center and a hospital is not satisfied.
- We set the due date of the irradiation centers to a sufficiently large number and the due date of the RBC to the maximum travel time.
- We consider the total tardiness and the total travel time simultaneously.
- If the deadlines of all assigned hospitals are greater than the maximum travel time of the vehicles, then we consider only the total travel time and use the nearest-neighbor heuristic (*NNH*) that is proposed for solving the travelling salesperson problem.
- We use different number of iterations (*NIMILS*) and runs (*NRMILS*) and "α" value (*AlphaMILS*) in our modified ILS algorithm.

One can observe that given the vehicle routes, our original problem can be modelled as an LP. Using this fact, we take vehicle routes, i.e., XX_{ik} values, returned by the modified ILS algorithm and transform them to X_{ikv} values to be used as parameters for Model M3 below:

$$\text{(M3) Max } z = \sum_{v=1}^{V}\sum_{p=1}^{P}\sum_{i=IC+1}^{N} w_i\left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpk}\right) \quad (3.59)$$

Subject to

Constraints (3.13, 3.14, 3.15, 3.16, 3.22, 3.23, 3.27, 3.28) and

$$\sum_{p=1}^{P}\left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv}\right)\left(1 - \sum_{k=1}^{|NO_v|} X_{ikv}\right) = 0$$

$$\text{for } i = IC+1, IC+2, \dots, N;\ v = 1,2,\dots,V \quad (3.60)$$

$$S_{i1v} \geq (T_{0i} + EA_v)X_{i1v} \qquad \text{for } i = 1,2,\dots,N; v = 1,2,\dots,V \quad (3.61)$$

$$S_{jkv} \geq S_{i,k-1,v} + a_i + b_i \sum_{p=1}^{P}\left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv}\right) + T_{ij}$$

$$\text{for } i = IC+1, IC+2, \dots, N; j = 0,1,\dots,N;$$

$$i \neq j; v = 1,2,\dots,V; k = 1,2,\dots,|NO_v|;$$

$$X_{jkv} = X_{i,k-1,v} = 1 \quad (3.62)$$

$$S_{jkv} \geq S_{i,k-1,v} + a_i + b_i \sum_{i'=IC+1}^{N}\sum_{p=1}^{P} IR_{i'ipv} + T_{ij}$$

$$\text{for } i = 1,2,\dots,IC;\ j = 1,2,\dots,N;$$

$$v = 1,2,\dots,V;\ i \neq j;$$

$$k = 1,2,\dots,|NO_v|;$$

$$X_{jkv} = X_{i,k-1,v} = 1 \quad (3.63)$$

$$\sum_{p=1}^{P} IR_{ijpv} = 0 \qquad \text{for } j = 1,2,\dots,IC; i = IC+1, IC+2, \dots, N;$$

$$v = 1,2,\dots,V;$$

$$\sum_{k=1}^{|NO_v|} k \cdot X_{ikv} \leq \sum_{k=1}^{|NO_v|} k \cdot X_{jkv} \tag{3.64}$$

$$R_{ipv}, IR_{ijpv}, S_{ikv} \geq 0 \qquad \text{for } i = 0,1,\dots,N;\ p = 1,2,\dots,P;$$

$$j = 1,2,\dots,IC;\ v = 1,2,\dots,V;$$

$$k = 1,2,\dots,|NO_v| \tag{3.65}$$

Objective function (3.59) is maximizing the total weighted demand satisfied. Constraint set (3.60) guarantees that if the vehicle does not visit a hospital, then it cannot deliver any product to that hospital. The arrival time of a vehicle to node i if this node is visited in the first position of its route, if node j is immediately visited after hospital i, and if node j is immediately visited after irradiation center i, are calculated in constraint sets (3.61), (3.62), and (3.63), respectively. Constraint set (3.64) ensures the amount of irradiated product p carried to hospital i from irradiation center j by vehicle v should be zero, if irradiation center j is visited after hospital i. Constraint set (3.65) represents non-negative variables.

Model M3 is an LP model, hence can be solved in polynomial time. After solving M3, we modify the objective function (3.59) by adding the term $\sum_{j=0}^{N}\sum_{i=0}^{N} T'_{ij}$, which is a constant since vehicle routes are known in advance. That is, we have

$$\text{Max } z = \sum_{v=1}^{V}\sum_{p=1}^{P}\sum_{i=IC+1}^{N} w_i\left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpk}\right) -$$

$$\varepsilon \sum_{j=0}^{N}\sum_{i=0}^{N} T'_{ij} \tag{3.66}$$

We now give the detailed description of five phases of the heuristic algorithms.

Phase 1: Finding eligible vehicles for hospitals

In this phase, we aim to find a good initial solution. In doing so, we group the hospitals by their locations and assign them to the vehicles. We get the inspiration from Ochi et al. (1998).

We first estimate the angle for each hospital pairs where the origin point of the angle is taken as the RBC, using the cosine law ($\text{Angle}_{ij} = arc\cos\{(T_{0i}^2 + T_{0j}^2 - T_{ij}^2)/(2T_{oi}T_{oj})\}$). Based on the angle values among all hospital pairs, we determine two leading hospitals with the maximum angle. For each unselected hospital, we determine the sum of the angles from the leading hospitals. We select the hospital with the maximum angle sum as the next leading one. We continue until all V leading hospitals are selected.

We assign the remaining hospitals to V groups of leading hospitals by considering the *critical angle* $(= 180/(V + 1))$ and the *critical time* (*CT*), which is found by multiplying the *time ratio for eligibility set* (*TRES*) and the average travel time between RBC and hospitals. If the angle between an unassigned hospital and a leading hospital is smaller than the critical angle or the travel time between the RBC and this unassigned hospital is smaller than the critical time, then we assign this hospital to the leading hospital group.

For each hospital group, we find the maximum travel time of the hospitals in the group from the RBC. We sort the hospital groups in descending order of those maximum travel times. In case of ties, we give priority to the group with the highest total demand. For each vehicle, we determine the total available time which is the difference between MT_v and EA_v. Similarly, we sort the vehicles in descending order of their total available times. In case of ties, we give priority to the vehicle with the larger capacity.

Starting from the first hospital group in the sorted list, we assign each hospital group to the vehicle in the same position of the sorted list of vehicles. For each hospital i in a group of hospitals, we set its eligibility parameter Z_{iv} to 1 if this group is assigned to vehicle v. This process is carried out for all hospitals with routine demand; however, all vehicles are assumed to be eligible for all irradiation centers and hospitals with urgent demand.

Phase 2: Finding an initial solution

We observe that the solutions of Model M1 and M2 may not be feasible for Model M3 due the deadline and maximum travel time constraints. To ensure a feasible solution for Model M3, we propose this phase.

We solve Model M1 and calculate the loading ratio (LR_{iv}) for each hospital and vehicle pair as follows:

$$LR_{iv} = \frac{\sum_{p=1}^{P}\left(R_{ipv}+\sum_{j=1}^{IC} IR_{ijpv}\right)}{\sum_{v'=1}^{V}\sum_{p=1}^{P}\left(R_{ipv'}+\sum_{j=1}^{IC} IR_{ijpv'}\right)} \tag{3.67}$$

where the numerator is the total amount of product carried by vehicle v to hospital i, and the denominator is the total amount of product carried by all vehicles serving hospital i.

Among all the hospital and vehicle pairs, we take pair (i', v') which gives the minimum loading ratio. If this ratio is smaller than the critical load ratio (*CLR*), then we set vehicle v' as ineligible for hospital i'. We resolve Model M1 and continue the process above until the minimum loading ratio is greater than *CLR*.

We use the modified ILS algorithm, in place of solving Model M2, to calculate the required parameters for Model M3. Next, we solve Model M3. If the solution is infeasible, then we select a vehicle route with the maximum tardiness value, and determine a hospital which is the furthest hospital (found heuristically) from the other hospitals on the same vehicle route. Otherwise, we apply Phase 2 until a feasible solution for Model M3 is found. Finally, we set this feasible solution as the Local_Solution and the Global_Solution since this feasible solution is the first solution to the original problem.

Phase 3: Improvement by inserting nodes to the vehicle routes

This phase tries to improve the solution by inserting additional nodes to the current vehicle routes since the value of our primary objective may increase as the number of assigned nodes increases.

We start by selecting a random node i' and a vehicle v' where i' is not already assigned to v' and this node-vehicle pair may improve the Local_Solution. Among all the possible positions in the vehicle route v', we randomly select *IPOS* number of insertion positions and calculate the travel time increase (*TTI*) for each selected position k as follows:

$$TTI_k = T_{f(k-1,v'),i'} + T_{i',f(k,v')} - T_{f(k-1,v'),f(k,v')} \qquad (3.68)$$

where $f(k, v')$ is the hospital which is assigned to position k of vehicle v'. Then, we select the position k' which gives the minimum *TTI* value, and update our vehicle route v' by inserting node i' to position k'. Furthermore, we calculate the fitness value with the procedure given below and continue the process above until all possible node-vehicle pairs are examined.

We sort the solutions obtained by the process above in descending order of their fitness values, and put the first *NAS*3 number of solutions into a temporary set of solutions. Starting from the best solution in this temporary set, we solve Model M3 and apply the unprofitable node elimination procedure given in Section below for all the members of the temporary set. For the first time when the objective function value returned by Model M3 for one of the solutions in the temporary set is better than that of the current Local_Solution, we replace the current Local_Solution with the new solution and repeat Phase 3 without solving the remaining solutions from the temporary set. If the objective function values returned by Model M3 for all solutions in the temporary set are not better than those of the current Local_Solution, we update the Global_Solution if the objective function value of the current Local_Solution is better than that of the current Global_Solution.

Fitness value calculation

Solving Model M3 many times may require excessive amount of computation time. In place of solving Model M3, we propose a method to find a powerful estimate (fitness) on its objective function value. In doing so, we first calculate the estimated values of R_{ipv} by dividing hospital i's demand for normal product p to the total number of vehicles visiting hospital i, and IR_{ijpv} by dividing the hospital i's irradiated demand for product p to the number of irradiated center visited before hospital i. Using these estimated R_{ipv} and IR_{ijpv} values, we check the product availability constraints, the vehicle capacity constraints, and the time related (deadline and maximum travel time) constraints. If one of these constraints are not satisfied, we calculate the amount of violations:

- *Product availability violation*: Sum of all violation amounts due to the product availability constraints is as given below:

$$PAV = \sum_{p=1}^{P} \max\left\{\sum_{v=1}^{V} \sum_{i=IC+1}^{N}\left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv}\right) - A_p\, , 0\right\} \quad (3.69)$$

- *Vehicle capacity violation*: Maximum of the difference between the total demand satisfied and the total vehicle capacity, and the sum of all vehicle capacity violation amounts, is as given below:

$$VCV = \max\left\{\begin{matrix}\sum_{v=1}^{V}\left(\sum_{p=1}^{P}\sum_{i=IC+1}^{N}\left(R_{ipv}+\sum_{j=1}^{IC} IR_{ijpv}\right)-C_v\right), \\ \sum_{v=1}^{V}\max\left\{\sum_{p=1}^{P}\sum_{i'=IC+1}^{N}\left(R_{i'pv}+\sum_{j=1}^{IC} IR_{i'jpv}\right)-C_v\,,0\right\}\end{matrix}\right\} \quad (3.70)$$

where i' is the set of hospitals served by one vehicle only.

- *Time related violation*: Maximum of the sum of the maximum deadline violations of all vehicle routes, sum of the maximum travel time violations of all vehicle routes and the difference between the total required time and the total vehicle availability time for all vehicle routes, is as given below:

$$TRV = \max\left\{\begin{matrix}\sum_{v=1}^{V}\max_{\forall i'}(MA_{i'}-DD_{i'},0) \\ \sum_{v=1}^{V}\max(MR_v-MT_v,0) \\ TTR-\sum_{v=1}^{V}(MT_v-EA_v)\end{matrix}\right\} \quad (3.71)$$

where

- i' is the set of hospitals which are visited by vehicle v.
- $MA_{i'}$ (MR_v) is the minimum arrival time of a vehicle to hospital i' (RBC), which includes EA_v values, sum of travel and constant loading/unloading times at all nodes that are visited before hospital i'(RBC), and all variable loading/unloading times of nodes that are served only by that vehicle and visited before hospital i' (RBC).
- TTR is the sum of all travel times, constant and variable loading/unloading times of all vehicle routes.

We calculate the estimated value for the objective function as follows:

$$\sum_{v=1}^{V}\sum_{p=1}^{P}\sum_{i=IC+1}^{N} w_i\left(R_{ipv}+\sum_{j=1}^{IC} IR_{ijpv}\right)-\varepsilon\sum_{j=0}^{N}\sum_{i=0}^{N} T'_{ij}$$
$$-\max\left\{\min_i(w_i)\times PAV, \min_i(w_i)\times VCV, \min_i\left(\frac{w_i}{b'_i}\right)\times TRV\right\} \quad (3.72)$$

where T'_{ij} is the travel time between nodes i and j,

$$b_i' = \begin{cases} b_i + b_{i'} & \text{if irradiation center } i' \text{ is used to process the demand of hospital } i \\ b_i & \text{otherwise} \end{cases},$$

and $\sum_{j=0}^{N} \sum_{i=0}^{N} T_{ij}'$ is a constant since vehicle routes are known in advance.

Procedure for deleting unprofitable node

If Model M3 returns a solution such that any hospital is visited with no service, we drop this hospital and resolve Model M3. If no such hospital exists, then we terminate the procedure.

Phase 4: Building a new vehicle schedule

The current Global_Solution may be a local optimal so that no further improvements could be achieved. In order not to get stuck in a local optimal solution, the current solution is perturbed either by removing a node from a vehicle route or interchanging two nodes in different vehicle routes.

We first calculate PV_{kv} values for each node k in vehicle route v of the Global_Solution, assuming that node k is between nodes i and j, as follows:

$$PV_{kv} = \min_{i'=IC+1,\dots,N} \left(\frac{w_{i'}}{b_{i'}}\right) \times \left(T_{ik} + T_{kj} + a_k - T_{ij}\right) - w_k \times PC_{kv} \qquad (3.73)$$

where PC_{kv} is the amount of products carried to node k by vehicle v, and the maximum weight of the hospitals which are visited after an irradiation center by vehicle v is used as the weight for this irradiation center.

If the Global_Solution is updated in the current iteration, then we calculate the difference between the maximum travel time and the arrival time to the RBC for each vehicle, select the vehicle(s) giving the minimum difference, determine a node with the maximum PV_{kv} value among vehicle(s) with the minimum difference, and remove this node from its vehicle route. Otherwise, we randomly select two vehicles, determine a node with the maximum PV_{kv} value in each vehicle, and pairwise interchange these two nodes. Finally, we replace the Local_Solution with the solution obtained by the process above.

Phase 5: Improvement by swapping (pairwise interchanging) nodes

This phase improves the solution by swapping two nodes in three ways, (1) between the vehicle routes, (2) within a vehicle route, and (3) between a node in a vehicle route and a node which is not in any vehicle route.

We randomly select two nodes that may provide better solution by using three ways mentioned above. We update our vehicle routes by interchanging the selected nodes and calculate the fitness value by the Fitness Value Calculation procedure and continue the process above until all possible node pairs are examined.

We sort the solutions obtained by the process above in descending order of their fitness values, and put the first *NAS*5 solutions into a temporary set. Starting from the best solution in this temporary set, we solve Model M3 and apply the unprofitable node elimination procedure for all the members of the temporary set. For the first time when the objective function value returned by Model M3 for one of the solutions in the temporary set is better than that of the current Local_Solution, we replace the current Local_Solution with the new solution. We repeat Phase 5 in Heuristic Algorithm 1 (Phase 3 in Heuristic Algorithm 2) without solving the remaining solutions from the temporary set, if the objective function values returned by Model M3 for all solutions in the temporary set are not better than those of the current Local_Solution, we update the Global_Solution if the objective function value of the current Local_Solution is better than that of the current Global_Solution.

The selection of the parameters (*TRES*, *IPOS*, *NAS*3, *NAS*5, *NI*, *NIMILS*, and *NRMILS*) used in our heuristic algorithms is discussed in the computational experiments section.

3.5 Computational experiments

In this section, we describe the computational experiment designed to evaluate the performance of the solution approaches. The heuristic algorithms are coded at the C++ platform, and C++ CPLEX application of IBM ILOG CPLEX optimization studio V12.6.2 is used to solve the mathematical models in our heuristic algorithms and the MILP model under the time limit of one hour. All computational experiments are conducted on a personal computer with Intel Xeon CPU E5-2650 2GHz (2 Processor) and 128 GB RAM under Windows 10 operating system.

3.5.1 Parameter settings

We use the real problem instances which are encountered the Central Anatolian RBC between January 4 and February 4, 2016. We observe that the product availabilities of this period are relatively smaller than the total demand of the hospitals. Therefore, this period is more critical period in a year, so it was selected. For this period, we solve 360 problem instances where the number of nodes changes between 10 and 55.

We now explain the way followed to obtain the data set.

- Demand quantities are received from the RBC data base. We categorize them according to the daily distribution slots.
- Urgent demands should be satisfied in one hour.

- Traveling times between nodes are determined via the geographic information system database of BASARSOFT Company, Ankara.
- Constant and variable parts of the loading and unloading times are estimated by using worker experiences.
- Weights of hospitals are determined as follows.
    - We classify the hospitals as standard (normal) hospitals, children hospitals, hospitals that cannot take blood from donor, large hospitals, hospitals sending special donors, and temporary RBCs.
    - We ask the experienced RBC staff to assign weights (between 0 and 10) to hospital classes and take the averages of the assigned weights.
    - Some hospitals that send their donors to the RBC have higher weights. A hospital's demand can be more than the quantity obtained from their donors. In order to deal with this issue, we separate hospitals demand in two parts: one only includes the quantities obtained from donors and second is the difference between its original demand and quantities obtained from donors. We give the weight of 10 for the first part, and for the second part we give its original weight. We create a dummy node for the hospital.
- If a hospital has both urgent and routine demand, then we duplicate the node of this hospital.
- We use 3, 4, 5, and 6 vehicles in our problem instances.
- All vehicles are identical in capacity with 300 units of blood products, and are ready at the beginning of the distribution period.
- The length of a distribution period is 3 hours, and this period defines the maximum travel time allowed to each vehicle.
- The problem instances are grouped by the number of nodes. Each group has 10 problem instances.

The number of nodes and product ranges are given in Table 2.

Table 3.2: Number of nodes and product types in our data set

Group	1	2	3	4	5	6	7	8	9
N	10-14	15-19	20-24	25-29	30-34	35-39	40-44	45-49	50-54
P	8-19	8-23	13-23	15-21	16-22	17-24	20-27	19-30	20-28

3.5.2 Performance measures

To evaluate the performance of the heuristic algorithms, we consider two types of solutions provided by CPLEX for MILP. These are the best integer solution and the best non-integer solution (some of the variables are non-integer) obtained by MILP within one hour time limit. For the first objective function value, we separate the instances into following two sets.

1. The instances with known optimal solutions- two cases may occur.
   i. CPLEX returns the optimal solution in one hour.
   ii. CPLEX cannot return the optimal solution in one hour, but the difference between best integer solution and best non-integer solution returned by CPLEX is less than 0.3. This implies that the optimal solution for the first objective function is found, as it can take only integer values or has a fractional part of 0.5 (since hospital weights used in the objective function are the multiples of 0.5). In this case, we cannot conclude that the resulting solution is optimal for the second objective function.
2. The instances with unknown optimal solutions.

On the other hand, for the first objective function, we use the following performance measure for the set of problem instances whose optimal solution is obtained by the MILP:

$$\text{Percent Error} = PE = \frac{z_1^B - z_1^H}{z_1^H} \times 100 \qquad (3.74)$$

where z_1^B is the optimum value of the first objective function returned by the MILP model, and z_1^H is the value of the first objective function returned by the heuristic algorithm.

For the set of the problem instances whose optimal solution is unknown, we use the following performance measure for the first objective function:

$$PE = \frac{z_1^N - z_1^H}{z_1^H} \times 100 \tag{3.75}$$

where z_1^N is the value of the non-integer solution returned by the MILP model. The number of optimal solutions obtained is another performance measure for the heuristic algorithms. The optimality of the heuristic algorithms is assured when the first objective function value is equal to the best non-integer objective function value of the MILP.

To evaluate the performance of the heuristic algorithms for the second objective function, we only use the set of problem instances with known optimal solution for the first objective function. We use the following performance measure for the second objective:

$$PE = \frac{z_2^H - z_2^M}{z_2^M} \times 100 \tag{3.76}$$

where z_2^M is the value of the second objective returned by the MILP model, and z_2^H is the value of the second objective function returned by the heuristic algorithm.

The CPU time is another performance measure used for the heuristic algorithms and the MILP model.

3.5.3 Parameter tuning for the heuristic algorithms

In this section, we discuss the selection of parameters (*TRES*, *IPOS*, *NAS*3, *NAS*5, *NI*, *NIMILS*, and *NRMILS*) used in our heuristic algorithms. Three levels of each parameter are tested. We use Group 1 and Group 9 defined in Table 3.3, i.e., small (10-14 nodes) and large (50-54 nodes) problem instances. We set the number of vehicles to 3 and 6.

We change the level of each parameter while fixing the levels of other parameters at the same level, as in Salvietti et al. (2014) and Nearchou and Lagodimos (2013). Table 3.3 reports the values of the performance measures for each level of each parameter.

Table 3.3: Parameter tuning results for the heuristic algorithms

<table>
<tr><th rowspan="2">Parameter</th><th rowspan="2">Levels</th><th colspan="5">Heuristic Algorithm 1</th><th colspan="5">Heuristic Algorithm 2</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>*</th><th>**</th><th>1</th><th>2</th><th>3</th><th>*</th><th>**</th></tr>
<tr><td rowspan="3">Time ratio for eligibility set (TRES)</td><td>Values</td><td>0.1</td><td>0.5</td><td>1</td><td></td><td></td><td>0.1</td><td>0.5</td><td>1</td><td></td><td></td></tr>
<tr><td>PE</td><td>4.4</td><td>2.8</td><td>2.7</td><td>1</td><td rowspan="2">0.5</td><td>3.4</td><td>3.6</td><td>3.2</td><td>1</td><td rowspan="2">1</td></tr>
<tr><td>CPU</td><td>3.2</td><td>2.7</td><td>3.1</td><td>0.5</td><td>1.5</td><td>1.7</td><td>1.3</td><td>1</td></tr>
<tr><td rowspan="3"># of insertion positions (IPOS)</td><td>Values</td><td>1</td><td>3</td><td>5</td><td></td><td></td><td>1</td><td>3</td><td>5</td><td></td><td></td></tr>
<tr><td>PE</td><td>3.1</td><td>2.9</td><td>2.9</td><td>3, 5</td><td rowspan="2">3</td><td>3.8</td><td>3.6</td><td>3.0</td><td>5</td><td rowspan="2">5</td></tr>
<tr><td>CPU</td><td>2.5</td><td>1.7</td><td>1.8</td><td>3</td><td>1.5</td><td>1.7</td><td>1.8</td><td>1</td></tr>
<tr><td rowspan="3"># of alternative schedules for Phase 3 (NAS3)</td><td>Values</td><td>5</td><td>10</td><td>15</td><td></td><td></td><td>5</td><td>10</td><td>15</td><td></td><td></td></tr>
<tr><td>PE</td><td>3.0</td><td>3.0</td><td>2.8</td><td>15</td><td rowspan="2">5</td><td>3.3</td><td>3.3</td><td>3.3</td><td>5, 10, 15</td><td rowspan="2">10</td></tr>
<tr><td>CPU</td><td>0.9</td><td>1.4</td><td>2.3</td><td>5</td><td>2.5</td><td>1.6</td><td>2.1</td><td>10</td></tr>
<tr><td rowspan="3"># of alternative solutions for Phase 5 (NAS5)</td><td>Values</td><td>10</td><td>20</td><td>30</td><td></td><td></td><td>5</td><td>10</td><td>15</td><td></td><td></td></tr>
<tr><td>PE</td><td>3.0</td><td>3.0</td><td>3.0</td><td>10, 20, 30</td><td rowspan="2">10</td><td>3.2</td><td>3.3</td><td>3.3</td><td>5</td><td rowspan="2">5</td></tr>
<tr><td>CPU</td><td>1.4</td><td>1.5</td><td>1.4</td><td>10, 30</td><td>1.5</td><td>1.5</td><td>1.6</td><td>5, 10</td></tr>
<tr><td rowspan="3"># of iterations for the main algorithm (NI)</td><td>Values</td><td>4</td><td>6</td><td>9</td><td></td><td></td><td>3</td><td>4</td><td>6</td><td></td><td></td></tr>
<tr><td>PE</td><td>3.1</td><td>3.0</td><td>3.0</td><td>6, 9</td><td rowspan="2">4</td><td>3.3</td><td>3.3</td><td>3.3</td><td>3, 4, 6</td><td rowspan="2">3</td></tr>
<tr><td>CPU</td><td>1.1</td><td>1.5</td><td>2.2</td><td>4</td><td>1.3</td><td>1.6</td><td>1.9</td><td>3</td></tr>
<tr><td rowspan="3"># of iterations for the Modified ILS Alg. (NRMILS)</td><td>Values</td><td>5</td><td>10</td><td>100</td><td></td><td></td><td>5</td><td>10</td><td>100</td><td></td><td></td></tr>
<tr><td>PE</td><td>3.3</td><td>3.2</td><td>3.5</td><td>10</td><td rowspan="2">10</td><td>3.3</td><td>3.3</td><td>3.3</td><td>5, 10, 15</td><td rowspan="2">5</td></tr>
<tr><td>CPU</td><td>1.5</td><td>1.4</td><td>1.5</td><td>10</td><td>1.5</td><td>1.6</td><td>1.8</td><td>5</td></tr>
<tr><td rowspan="3"># of runs for the Modified ILS Alg. (NRMILS)</td><td>Values</td><td>2</td><td>5</td><td>10</td><td></td><td></td><td>2</td><td>5</td><td>10</td><td></td><td></td></tr>
<tr><td>PE</td><td>3.2</td><td>3.0</td><td>3.2</td><td>5</td><td rowspan="2">2</td><td>3.3</td><td>3.3</td><td>3.3</td><td>2, 5, 10</td><td rowspan="2">2</td></tr>
<tr><td>CPU</td><td>1.5</td><td>1.8</td><td>1.6</td><td>2</td><td>1.5</td><td>1.7</td><td>2.0</td><td>2</td></tr>
<tr><td rowspan="3">Alpha value for Modified ILS Alg. (AlphaMILS)</td><td>Values</td><td>0.4</td><td>0.6</td><td>0.8</td><td></td><td></td><td>0.4</td><td>0.6</td><td>0.8</td><td></td><td></td></tr>
<tr><td>PE</td><td>3.4</td><td>3.4</td><td>3.4</td><td>0.4, 0.6, 0.8</td><td rowspan="2">0.8</td><td>3.2</td><td>3.3</td><td>3.2</td><td>0.4, 0.8</td><td rowspan="2">0.8</td></tr>
<tr><td>CPU</td><td>0.8</td><td>0.8</td><td>0.8</td><td>0.4, 0.6, 0.8</td><td>1.4</td><td>1.4</td><td>1.4</td><td>0.4, 0.6, 0.8</td></tr>
</table>

* Best value(s) for performance criteria
** Selected value for the heuristic algorithm

Table 3.3 shows that all percent deviations of the heuristic algorithms are very small (almost all are below 3.5%) and not much sensitive to the selected levels of the parameters. Hence, we base our parameter selection on the CPU times (select the level that gives the minimum CPU time) and we use the percent deviation as tie breaker (if any percent deviation cannot resolve ties, as for the *NAS5* value of Heuristic Algorithm 1 and *AlphaMILS* value for both algorithms, we make a random selection). We make a single exception for the *IPOS* value of Heuristic Algorithm 2 where we select the level that leads to the smallest percent deviation (3%) and highest CPU time, as the first and second smallest CPU times were due to the levels with relatively too high percent deviation values (above 3.5%).

3.5.4 Discussion of the results

In this section, we discuss the results of the experiments for two heuristic algorithms and the MILP model. We summarize the results in Table 3.4.

Table 3.4: Summary for discussion of the results

Difference between the best integer and the non-integer solution by the MILP	Number of problem instances	Objective function 1	Objective function 2
Less than 0.3	55	HAs obtain the optimal solution for all instances	See Table 3.5
Greater than 0.3	305	See Table 3.6	----

We first discuss the performance of the heuristic algorithms for the problem instances whose optimal solutions are known. We observe that the MILP model obtains optimal solutions for the first objective function for 55 out of 360 problem instances. Heuristic Algorithms 1 and 2 also return optimal solutions for the first objective function for all 55 problem instances. Table 3.5 reports on the CPU times of the heuristic algorithms and their performance relative to the second objective function.

Note that Heuristic Algorithms 1 and 2 have average CPU times of 3.29 and 4.57 seconds, respectively. We observe that the increase in the number of vehicles does not affect the CPU times of the heuristic algorithms; however, the number of nodes has a significant effect on the CPU times. From Table 3.5, we observe that our heuristic algorithms provide better solutions for the second objective function with negative average *PE*s. Overall performance error of the Heuristic Algorithms 1 and 2 are -29.09 and -24.59 percent, respectively, i.e., our heuristic algorithms provide about 30 percent better results than the MILP. We observe that *PE* is affected by the number of vehicles since it significantly reduces as the number of vehicles increases. We may also conclude that the number of nodes does not have any effect on the *PE* since we did not observe any structural behaviour of *PE*s when the number of nodes increases. We also report the performance of these heuristic algorithms' best one which is obtained by taking the minimum value of the second objective function returned by the heuristic algorithms. Best of these heuristics provides better results accordingly, but its contribution to the second objective is about 1.5 percent better than Heuristic Algorithm 1.

Table 3.5: Performance of heuristic algorithms – the instances with known optimal solutions for the first objective

			Heuristic Algorithm 1		Heuristic Algorithm 2		Heuristic Algorithm MIN{HA1, HA2)	
N	*V*	Number of Optimum Solutions Obtained	Avg. *PE*	Avg. CPU (Sec.)	Avg. *PE*	Avg. CPU (Sec.)	Avg. *PE*	Avg. CPU (Sec.)
10-14	3	10	-18.89	2.26	-16.70	3.34	-19.97	5.60
15-19		2	-34.77	0.71	-23.99	0.80	-34.77	1.52
Avg. (Sum)		**(12)**	**-21.53**	**2.01**	**-17.91**	**2.92**	**-22.44**	**4.92**
10-14	4	10	-25.37	2.75	-20.73	3.56	-27.22	6.31
15-19		6	-24.00	5.15	-20.49	6.93	-26.49	12.09
Avg. (Sum)		**(16)**	**-24.86**	**3.65**	**-20.64**	**4.82**	**-26.95**	**8.47**
10-14	5	10	-30.91	2.94	-26.35	3.65	-32.78	6.60
15-19		4	-26.77	5.57	-25.96	8.22	-27.73	13.79
Avg. (Sum)		**(14)**	**-29.73**	**3.69**	**-26.24**	**4.96**	**-31.33**	**8.65**
10-14	6	10	-37.01	3.08	-28.56	4.17	-37.50	7.24
15-19		3	-52.61	5.27	-51.42	9.42	-55.14	14.69
Avg. (Sum)		**(13)**	**-40.61**	**3.58**	**-33.84**	**5.38**	**-41.57**	**8.96**
Avg. (Sum)		**(55)**	**-29.09**	**3.29**	**-24.59**	**4.57**	**-30.54**	**7.86**

Table 3.6: Performance of heuristic algorithms for the first objective – unsolved instances by MILP.

		Heuristic Algorithm 1			Heuristic Algorithm 2			Heuristic Algorithm MAX{HA1, HA2)		
N	*V*	# of Hits*	Avg. *PE*	Avg. CPU (Sec.)	# of Hits*	Avg. *PE*	Avg. CPU (Sec.)	# of Hits*	Avg. *PE*	Avg. CPU (Sec.)
15-19	3	7	0.38	5.29	7	0.38	7.66	7	0.38	12.95
20-24		5	0.84	8.83	5	0.73	13.72	5	0.73	22.56
25-29		1	1.49	15.65	5	1.60	24.61	5	1.22	40.26
30-34		0	5.20	21.81	0	4.85	33.02	0	4.64	54.83
35-39		0	7.38	33.40	0	6.31	49.52	0	5.64	82.91
40-44		0	6.41	53.51	0	6.31	81.99	0	5.79	135.50
45-49		0	8.10	58.69	0	8.57	111.16	0	7.84	169.85
50-54		0	23.95	69.17	0	22.38	121.58	0	21.74	190.75
Avg. (Sum)		**(13)**	**6.88**	**34.01**	**(17)**	**6.55**	**56.63**	**(17)**	**6.14**	**90.64**
15-19	4	3	0.76	5.83	3	0.76	7.02	3	0.76	12.85
20-24		8	0.06	12.45	10	0.00	16.50	10	0.00	28.94
25-29		6	0.17	16.72	8	0.45	28.71	8	0.11	45.42
30-34		1	2.33	29.10	1	1.75	43.45	1	1.57	72.55
35-39		1	2.41	39.15	1	2.22	63.20	1	2.01	102.36
40-44		0	2.79	64.39	0	2.60	84.84	0	2.10	149.23
45-49		0	5.22	75.28	0	4.52	136.79	0	4.36	212.07
50-54		0	6.99	95.16	0	6.78	149.99	0	5.89	245.15
Avg. (Sum)		**(19)**	**2.74**	**45.21**	**(23)**	**2.52**	**71.12**	**(23)**	**2.21**	**116.33**
15-19	5	5	0.51	5.76	5	0.51	8.24	5	0.51	14.00
20-24		7	0.09	13.99	10	0.00	20.68	10	0.00	34.67
25-29		7	0.20	18.89	9	0.27	39.66	9	0.19	58.55
30-34		4	0.53	31.28	9	0.03	57.68	9	0.03	88.96
35-39		3	1.38	48.88	6	0.28	76.68	7	0.21	125.56
40-44		2	1.38	70.20	2	0.73	136.01	3	0.65	206.21
45-49		0	2.37	92.21	0	1.46	140.67	0	1.16	232.88
50-54		0	3.41	98.83	0	3.72	194.36	0	2.62	293.19
Avg. (Sum)		**(28)**	**1.27**	**49.70**	**(41)**	**0.89**	**88.25**	**(43)**	**0.68**	**137.95**
15-19	6	6	0.44	6.15	6	0.44	11.77	6	0.44	17.92
20-24		8	0.33	16.35	8	0.33	24.30	8	0.33	40.65
25-29		8	0.35	21.82	9	0.08	48.42	10	0.00	70.24
30-34		7	0.31	37.42	10	0.00	71.71	10	0.00	109.12
35-39		5	0.56	45.78	8	0.03	111.45	8	0.02	157.23
40-44		3	0.86	72.58	6	0.28	149.62	6	0.25	222.20
45-49		1	0.92	106.26	1	0.92	166.22	2	0.66	272.48
50-54		0	1.80	119.43	0	1.16	223.98	0	1.01	343.40
Avg. (Sum)		**(38)**	**0.71**	**55.06**	**(48)**	**0.40**	**104.41**	**(50)**	**0.33**	**159.46**
Avg. (Sum)		**(98)**	**2.92**	**45.95**	**(129)**	**2.61**	**80.09**	**(133)**	**2.36**	**126.04**

* # of Hits: Number of times the heuristic algorithm obtains the optimal solution

The performance of the heuristic algorithms on the first objective function for the unsolved instances is presented in Table 3.6. We report the number of optimum solutions obtained by the heuristic algorithms and their average CPU times and *PE*s. From Table 3.6, we observe that Heuristic Algorithms 1 and 2 return optimal solutions for 98 and 129 out of 305 problem instances, respectively. We also observe that the *PEs* from the non-integer solutions returned by the MILP of Heuristic Algorithms 1 and 2 on the first objective are 2.92 and 2.61, respectively. It means *PE* from the optimal solution on the first objective function is less than 2.92 and 2.61 percent, respectively. The average CPU times of these algorithms are 45.95 and 80.09 seconds, whereas those instances could not be solved in an hour by the CPLEX. Hence, our algorithms provide better results than the CPLEX in relatively short CPU times.

We may also conclude that the number of vehicles has a significant effect on both CPU time and *PE* of the heuristic algorithms. CPU times increase and *PEs* decrease as the number of vehicles increases. In addition, the number of nodes has also effect on the performance measures; both CPU time and *PE* increases as the number of nodes increases. We also report the best solution of the heuristic algorithms which is given in the last three columns of Table 3.6. Taking the best of the heuristic algorithms returns better results where the total number of optimum solutions increased to 133 and the *PE* reduced to 2.36. On the other hand, the average CPU time increased to 2 minutes. We may conclude that the user can use both heuristic algorithms if one desires better results in longer CPU times.

From our experiments, we can conclude that the MILP does not perform well since it does not provide the optimal solution for any problem instance and also it does not obtain an integer feasible solution for two problem instances within 1 hour of CPU time. Moreover, its relative gap between best non-integer and best integer solution is too high. On the other hand, our heuristic algorithms provide high quality solutions in relatively small CPU times. Heuristic Algorithms 1 and 2 (on the first objective function) return optimal solutions for 153 and 184 problem instances (out of 360), respectively. Once the best solution of two heuristic algorithms is used, the optimal solutions for 188 (more than half) problem instances solved for the first objective function are reached.

We also analyze the effects of the number of hospitals served, and the number of vehicles on the maximum weighted satisfied demand (z_1) and the total time travelled (z_2) values and report the results in Table 3.7.

Table 3.7: Objective function values of the Heuristic Algorithm Max{HA1, HA2}

V	3		4		5		6	
N	Avg. z_1	Avg. z_2	Avg. z_1	Avg. z_2	Avg. z_1	Avg. z_2	Avg. z_1	Avg. z_2
10-14	645.9	118.5	645.9	119.4	645.9	118.4	645.9	115.4
15-19	634.7	168.3	634.7	169.1	634.7	172.5	634.7	172.4
20-24	1113.5	199.8	1120.5	223.1	1120.5	215.3	1118.7	220.0
25-29	1163.6	205.1	1177.8	227.4	1176.9	233.9	1179.1	245.0
30-34	1973.2	220.7	2036.6	269.3	2070.9	311.3	2071.6	313.6
35-39	2120.4	223.8	2205.0	283.0	2251.4	306.5	2258.9	310.4
40-44	3434.4	188.4	3574.6	237.1	3628.2	305.4	3642.5	325.1
45-49	4086.9	204.6	4265.7	289.8	4403.7	359.2	4427.0	375.7
50-54	4007.8	188.3	4249.9	280.0	4391.1	353.4	4463.7	396.2
Avg.	**2131.1**	**190.8**	**2212.3**	**233.1**	**2258.1**	**264.0**	**2271.3**	**274.8**

We observe from Table 3.7 that any increase in z_1 value leads to an increase in z_2 value. That is, to satisfy more demand, one has to pay more for the distribution cost. Note from the table that, increasing the number of vehicles increases z_1 and z_2 values. The increases are more significant when the number of vehicles is small. When the number of nodes is between 50-54 and the number of vehicles increases from 3 to 4, the total satisfied demand increases to 4250 from 4008 and total time travelled increases to 280 from 188. For the same case, when the number of vehicles increases from 5 to 6, the total satisfied demand increases to 4464 from 4391 and total time travelled increases to 396 from 353. Using the results of Table 3.7, the managers may analyze the trade-offs between number of vehicles (along with z_2 value) and z_1 value and select the best solution according to their preferences.

3.5.5 Sensitivity analysis

In this section we aim to analyze the effects of some critical parameters on the maximum total weighted satisfied demand amount (z_1), total travel time (z_2) and on the CPU times. We select the maximum travel times for vehicles (MT_v), availability of products (A_p), and weight of the hospitals (w_i) as critical parameters for the vehicles, products and hospitals, respectively.

For the effects of the MT_v, we select two levels: 180 minutes to represent low and 240 minutes to represent high travel times. To see the effect of product availabilities, i.e., A_p values, we select two levels: the original product availabilities and twice of the original product availabilities for the second level to represent low and high availability levels, respectively. To see the effects of hospital weights we use two levels. The low weight level is represented by the original data where the weights are distributed between 0.5 and 10. To find the high weight level instances, we take the weights of the low weight instances and double the ones that are below 5; hence, the resulting weights are distributed between 1 and 10.

We select two problem combinations (Group 1– small sized instances with 10-14 nodes and Group 9– large sized instances with 50-54 nodes) from Table 3.2 and report the associated results in Table 3.8. We set the number of vehicles to 3.

Note from Table 3.8 that the parameters do not have any significant effect on the CPU times of the heuristic algorithms. Recall that the problem size parameters (number of vehicles and number of hospitals) caused very significant increase on the CPU times. Furthermore, for large sized instances, the z_1 values increase from 4010.4 to 4250.4 when the maximum travel times increase from 180 to 240 minutes. This is due to the fact that by increasing travel times, more room becomes available for demand satisfactions. This in turn increases z_2 values from 210.4 to 297.9 as more nodes are visited due to more available time. We observe that increasing the maximum travel time does not have a significant effect on the z_1 and z_2 values for the small sized problem instances.

Table 3.8: Sensitivity analysis based on the parameters

		Original	High MT_v	Twice A_p	Modified w_i
z_1	Group 1	645.9	645.9	658.1	1060.1
	Group 9	4010.4	4250.4	4204.8	5686.3
z_2	Group 1	125.3	124.9	125.9	126.1
	Group 9	210.4	297.9	201.9	197.3
CPU	Group 1	4.2	4.5	4.2	4.3
	Group 9	168.4	158.9	164.8	171.2

We also observe that z_1 values slightly improve when the product availability increases. The improvement is more significant for the large-sized instances. It is less than expected and we can conclude that the availability of product is not a highly binding constraint for satisfying the demand of hospitals.

Table 8 also shows that when the some hospital weights are doubled, the z_1 values increase from 645.9 to 1060.1 (about 65%) and from 4010.4 to 5686.3 (about 40%) for the small and large-sized instances, respectively. However, the associated z_2 values remain almost the same because the routes followed by the vehicles may not be affected from the hospital weights.

CHAPTER 4

PROBLEM 2 (P2): NEW DISTRIBUTION STRATEGIES FOR BLOOD DISTRIBUTION PROBLEM FOR THE TURKISH RED CRESCENT

In this chapter, we discuss our second problem that presents new distribution strategies for the Turkish Red Crescent. We extend some of our results for the first problem and develop some new approaches. We first define the problem. Then, we give its mathematical model and present heuristic algorithm. Finally, we discuss the results of our computational experiment.

4.1 Literature Review

The blood products literature goes back to 1960s. Since then, an enormous amount of research considering different aspects of blood products supply chain has been developed. Several review papers have also been published, two most noteworthy are due to Belien and Force (2012) and Osorio et al. (2015).

Osorio et al. (2015) consider quantitative models and present their main characteristics based on their relevance to the following stages of the supply chain: collection, production, inventory and delivery. Belien and Force (2012) classify the blood products literature under different classification fields as type of blood products, solution method, hierarchical levels, type of problems, type of approaches, performance measures, and practical implementation or case studies. We discuss the related literature based on type of the problems (one of the categories in Belien and Force (2012)). The position of the papers for the other categories given in Table 4.1.

Inventory management studies

Hosseinifard and Abbasi (2018) considered the inventory centralization of the blood products at hospital level. They showed that the centralization of some hospitals so as to satisfy the demands of the others, helps to reduce the outdates and shortages at all places. Dillion et al. (2017) studied the inventory planning of red blood cells at individual hospital levels. They studied stochastic aspects of the problem and proposed stochastic models so as to find the review period length and target inventory level.

Ensafian and Yaghaubi (2017) considered platelets at the RBC level and proposed mathematical models to find optimal (minimum cost) inventory levels. They also defined a biobjective problem that trade-offs between the total cost and the freshness of the delivered products. Attari et al. (2017) also considered two objectives: total cost and maximum unsatisfied demand among all hospitals, and proposed multi-choice goal programming technique to handle their trade-offs.

Location and allocation-Planning for collection

Ghasemi and Bashiri (2017) studied a blood supply network to find the locations of the blood mobile facilities and blood donation sites and, the inventory levels at the blood centers. To handle the stochastic demand, a two stage model that reduces the total wastage and holding costs is developed. Ramezanian and Behboodi (2017) considered location and allocation strategies in the blood supply network with demand and cost uncertainties. They first discussed a deterministic model so as to reduce shortages and harmful damages, and then incorporated uncertainties using a robust optimization approach. Zahiri and Pishvaee (2017) considered blood group incompatibility in their blood supply network. They gave a mathematical model with two objectives: total cost and maximum unsatisfied demand. To handle the uncertainties, they proposed robust probabilistic approaches. Khalilpourazari and

Khamseh (2017) developed a multi objective model for the blood supply chain design in earthquakes. They considered the magnitude of the earthquakes, and different transportation models with variable speed and capacity. Zahiri and Pishvaee (2017) and Khalilpourazari and Khamseh (2017) discussed real world applications in Iran, of their studies.

Location and allocation- studies which consider distribution

Daskin et al. (2002) introduced a distribution center location problem. They considered working inventory cost, safety stock inventory cost at the distribution centers, and transportation cost from suppliers, to the distribution centers. They formulate the problem as a nonlinear integer model and propose a lagrangian relaxation algorithm for its solution. Shen et al. (2003) considered a joint location – inventory problem with a single supplier and multiple retailers. They proposed a set covering integer model that assigns the retailers to the distribution centers, and proposed a column generation algorithm for its solution. Yegül (2016) introduced a new echelon, so called regional transfusion centers, and proposed a nonlinear integer model to find the location of the regional blood and transfusion enters. They presented several decomposition and simulated annealing based heuristic algorithms to solve their real life application in the Turkish Red Crescent. Şahin et al. (2007) addressed the location problem in the Turkish Red Crescent, at a regional level. They defined two levels of hierarchy, where level 1 includes regional blood centers and level 2 include blood centers, blood stations and mobile units. They developed several mathematical models to solve the problems at both levels.

Kaveh and Ghobadi (2017) considered the problem of allocating the blood centers to the hospitals so as to minimize the total distance between the centers and hospitals. They proposed a graph partitioning based algorithm and a metaheuristic algorithm using a new neighborhood structure. Chaiwuttisak et al. (2016) considered a location-allocation problem with two types of service facilities: blood donation room only and donation room with a distribution center. They presented an integer programming model to improve the blood products supply while reducing the transportation costs. Kaveh and Ghobadi (2017) and Chaiwuttisak et al. (2016) presented case studies in Iran and Thailand, respectively.

Fahimnia et al. (2017) considered the problem of finding the number of facilities, blood collection and transportation quantities and inventory levels in case of disasters. They developed a hybrid solution approach that combines ε-constrained and Lagrangian relaxation ideas. Their aim is to minimize supply chain costs while maintaining timely supply of blood.

Distribution studies

Distribution problems for blood products are rarely studied in the literature. Gregor et al. (1982) studied the distribution strategies to determine the number of vehicles used and inventory levels. They had three evaluation criteria: the number of emergency orders, average response time to an emergency order, the number of routine surgeries postponed. Hemmelmayr et al. (2009) studied the distribution problem of the blood products for Australian hospitals so as to minimize the total travel time. Stochastic nature of the same problem was considered by Hemmelmayr et al. (2010). Salehipour and Sepehri (2012) also studied blood distribution problem with the objective of minimizing the total waiting time.

In this study, we extend the Problem 1 with considering two more distribution strategies. we consider the allocation of the blood products to the hospitals along with their distribution (O1), via another hospital (O2), and directly from RBC (O3). Hence we extended the category given in Problem 1, distribution - product allocation.

Moreover, we also consider irradiation centers in our distribution network different than the other studies, which leads to precedence relations between nodes.

Table 4.1: Summary of the studies in our literature review

Reference	Problem type	Type of blood products	Hierarchical level	Solution methodology
Daskin et al. (2002)	Inventory Location	Platelets	RBC	Lagrangian Relaxation
Dillon et al. (2017)	Inventory	Red Blood cells	RBC	IP
Fahimnia et al. (2017)	Location Allocation	Whole blood	Supply Chain	ε-constrained and Lagrangian Relaxation
Ghasemi and Bashiri (2017)	Location Allocation	Whole blood	RBC	Robust IP
Shen et al. (2003)	Inventory Location	Platelets	RBC	IP
Yegül (2016)	Location Allocation	Blood products	Supply Chain	Non-linear IP and Heuristic algorithms
Chaiwuttisak et al. (2016) *	Location Allocation	Whole blood	Supply Chain	IP
Gregor et al. (1982) *	Distribution	Blood products	RBC	Simulation
Hummelayr et al. (2009) *	Distribution	Other/Unknown	RBC	IP and VNS
Hummelayr et al. (2010) *	Distribution	Other/Unknown	RBC	IP and VNS
Hosseinifard and Abbasi (2018)	Inventory	Blood products	RBC	What-if scenario analysis
Kaveh and Ghodabi (2017) *	Location Allocation	Blood products	Supply Chain	Enhanced colliding bodies algorithm
Ramezanian and Behboodi (2017) *	Location Allocation	Whole blood	Supply Chain	IP
Sahin et al. (2007) *	Location Allocation	Whole blood	Supply Chain	IP
Selahipour and Sepehri (2012) *	Distribution	Blood products	RBC	IP and hybrid heuristic
Khalilpourazari and Khamseh (2017) *,**	Location Allocation	Whole blood	Supply Chain	IP
Zahiri and Pishvaee (2017) *,**	Location Allocation	Whole blood	Supply Chain	Robust and Fuzzy IP
Attari et al. (2017) *,**	Inventory	Blood products	RBC	IP
Ensafian and Yaghaubi (2017) *,**	Inventory	Platelets	RBC	IP
Problem 1 *,**	Distribution	Blood products	RBC	Decomposition based heuristic
Problem 2 *	Distribution	Blood products	RBC	Hybrid genetic algorithm

IP: Integer Programming, VNS: Variable Neighborhood Search
*: practical application, **: multi-objective

4.2 The blood distribution Problem

In the current blood supply chain system of TRC, there exists one way to satisfy the demand of the hospitals. Demand is only satisfied by the vehicles of the RBCs. Due to the vehicle and time limitations, the demand of all hospitals cannot be satisfied by this way. This have motivated us to propose new distribution strategies for distribution of the blood products. We have inspired from first problem (Problem P1) and we proposed two new strategies to their problem, where demand is only satisfied by the vehicles of the RBC, and we call this option as the *RBC service* (Option O1). In addition to Option O1, we propose new strategies: *transfer service* (Option O2) and *self-service* (Option O3).

In the blood distribution system, we have a set of vehicles available at the RBC to deliver blood products through a set of routes to a set of demand points, i.e., hospitals. We also considered the irradiated demand of the hospitals with our distribution strategies. Irradiation of the products in the options O1 and O2 is the responsibility of RBC, i.e., the RBC vehicles first carry the products to the irradiation centers for irradiation process, and then delivers the irradiated products to the hospitals. There are many irradiation centers, so we should choose the irradiation center and sequence it effectively in the vehicle route. However, irradiation of the products in the Option O3 is the responsibility of the hospital with the irradiated product demand.

We also considered the most of the issues given in Problem P1, but our second study differs in some ways. The comparison between the Problem P1 and Problem P2 is given in Table 4.2.

Table 4.2: Comparison of the problem characteristics of Problem P1 and P2

Problem Environment	Problem P1	Problem P2
RBC service (O1)	Yes	Yes
Transfer service (O2)	No	Yes
Self service (O3)	No	Yes
Urgent demands	Have to be satisfied	Satisfied with high weights
Vehicle capacity and availability constraints	Yes	Yes
Product availability constraints	Yes	Yes
Irradiation centers	Yes	Yes
Number of objective functions	Multiple	Single

Figure 4.1 illustrates a distribution system in a period for a case where there are eight hospitals, one irradiation center, and two RBC vehicles. Normal demand of the hospitals 3, 6 and 7 is loaded to the first vehicle of the RBC. This vehicle sequentially visits the hospitals 7, 6, 3 and returns to the RBC. The second vehicle of the RBC is loaded with the normal demand of the hospitals 1, 2 and 5, and the irradiated demand of the hospitals 2, 3 and 5. This second vehicle first visits the hospital 1, and then goes to the irradiation center at which the blood products for the hospitals 2, 3 and 5 are irradiated. Then, the second vehicle visits the hospital 3 and unloads the irradiated products of this hospital. Finally, the hospital 2 is visited by the second vehicle, and the normal and irradiated products of the hospitals 2 and 5 are unloaded. The route of the second vehicle finishes at the RBC. A vehicle of the hospital 5 visits the hospital 2 and takes (normal and irradiated) demand of the hospital 5. Moreover, a vehicle of the hospital 4 directly visits the RBC to take its normal demand. Figure 2 shows that the demand of the hospital 8 is not satisfied within the distribution period.

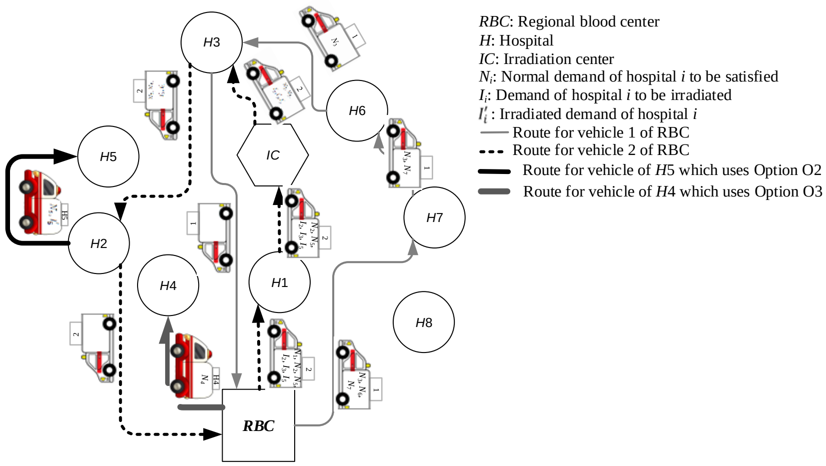

Figure 4.1: Distribution system

4.3 Mathematical model

In this section, we present the mathematical programming formulation of our blood distribution problem over a distribution period of pre-specified length. To do this, we will follow the notation given by Problem P1 and add new parameters, indices and decision variables. We assume that there are $N+1$ nodes, where one node is for the RBC, IC nodes are for the irradiation centers and the remaining $N-IC$ nodes are for the hospitals. Among these nodes ($i=0,1,\dots,N$), $i=0$ is the regional blood center (RBC), $i=1,2,\dots,IC$ are for the irradiation centers, and $i=IC+1, IC+2,\dots,N$ are for the hospitals. U is the set of hospitals with urgent demands (there are $|U|$ hospitals in U).

The travel time between nodes i and j is T_{ij} time units. a_i and b_i are the quantity independent and per unit loading and unloading times for node i, respectively. DD_i is the deadline for hospital i.

The hospitals have different weights that represents their relative importance. We use relatively high weights for the hospitals whose demand is urgent. Hospitals weights are also dependent on the demand satisfaction option. The most profitable option is satisfying the demand with the RBC's vehicles (Option O1) since it is likely to be the most preferred option by the hospitals. We use the transfer service (Option O2) as the second profitable option. The weight of this option depends on the two hospitals: hospital whose demand is satisfied and hospital giving the transfer service. This weight is smaller than the weight of the Option O1. Option O3 is the least preferred option for a hospital since this hospital must use its own vehicles to satisfy the demands. Therefore, the smallest weight is for the Option O3. Thus, we have the following relations for the weights of the hospitals:

$$w_i \geq \alpha_{il} \geq \beta_i$$

Where w_i is the weight of hospital i where demand is satisfied by the vehicles of the RBC (Option O1), α_{il} is the weight of hospital i whose demand is satisfied by the transfer hospital l, and β_i is the weight of hospital i whose demand is satisfied by the vehicles of this hospital.

There are P product types, and A_p units of product p ($p = 1,2,..,P$) are available at the RBC. The amount of normal product p demanded by hospital i is D_{ip}, and the amount of irradiated product p demanded by hospital i is ID_{ip} units.

There are V vehicles, carrying blood products, with identical speed and different capacity. Carrying capacity of vehicle v ($v = 1,2,..,V$) is C_v units. Vehicle v becomes available at time EA_v and remains continuously available. Each vehicle starts and ends its route at the RBC. The maximum travel time of vehicle v is MT_v time units. Hence, the maximum of the MT_v values is an upper bound on the length of the distribution period.

Each hospital cannot be used to satisfy the demand of another hospital due to some restrictions such as the lack of a depot area or a special refrigerator. Therefore, we use a subset CT of the hospitals which can be used as the set of transfer points to satisfy the demand of the other hospitals. We also assume that a hospital in this subset must have demand. Moreover, each hospital cannot use each hospital to satisfy its demands due to some restrictions such as the distance between two hospitals. Then, we use DC_i and DH_i to represent the set of hospitals that can be used to satisfy the demand of hospital i and the set of hospitals whose demand can be satisfied by hospital i, respectively. Some hospitals cannot directly visit the RBC due to lack of vehicles and long distance between the hospital and the RBC. Thus, we use another subset SD to identify the hospitals which can satisfy its demand by the Option O3.

We make the following additional assumptions to construct the mathematical model:

- All parameters are known with certainty and not subject to any change, i.e., the system is deterministic and static.
- Each vehicle can visit each node.
- Each node can be visited by more than one vehicle.
- Blood products to be irradiated are first delivered to the irradiation centers and then to the hospitals.
- Irradiation centers are always available.
- If a hospital directly satisfies its demand from the RBC by its vehicle, then there should not be any transfer with the RBC vehicles or any hospital.
- A hospital cannot be used as the transfer point to satisfy the demand of another hospital whose demand is not urgent if the demand of the hospital giving the transfer service is not fully satisfied. This case is not valid while satisfying the urgent demands.

Our problem is to deliver blood products (normal and irradiated) via three options. Thus, N is an upper bound on the number of positions (nodes) visited on each vehicle route, where k is an index for positions ($k = 1,2,\dots,K$) in this route. Based on the problem characteristics and assumptions mentioned above, we develop a mixed integer linear programming (MILP) model, which is an extension of the model given in Problem P1.

***Decision variables*:**

$$X_{ikv} = \begin{cases} 1 & \text{if vehicle } v \text{ visits node } i \text{ at position } k \text{ of its route} \\ 0 & \text{otherwise} \end{cases}$$

$R_{ipv} =$ Amount of normal product p carried to hospital i by vehicle v

$IR_{ijpv} =$ Amount of irradiated product p processed by irradiation center j ($j = 1,\dots,IC$) and carried to hospital i ($i = IC+1, IC+2,\dots,N$) by vehicle v

$FR_{ilpv} =$ Amount of normal product p carried to hospital l by vehicle v to satisfy the normal demand of hospital i

$FIR_{ijlpv} =$ Amount of irradiated product p processed by irradiation center j and carried to hospital l by vehicle v to satisfy the irradiated product demand of hospital i

$DR_{ip} =$ Amount of product p is used to satisfy the total demand of hospital i by Option O3.

$$Y_{ijv} = \begin{cases} 1 & \text{if irradiation center } j \text{ is used to process the demand of} \\ & \text{hospital } i \text{ carried by vehicle } v \\ 0 & \text{otherwise} \end{cases}$$

$S_{ikv} =$ Arrival time of vehicle v to node i at its k^{th} position of its route

$$DV_i = \begin{cases} 1 & \text{if hospital } i \text{ uses the self-service Option O3 to satisfy its total demand} \\ 0 & \text{otherwise} \end{cases}$$

$$UT_{ijv} = \begin{cases} 1 & \text{If hospital } j \text{ is used to satisfy the demand of hospital } i \text{ that has urgent demand by vehicle } v \\ 0 & \text{otherwise} \end{cases}$$

$$WS_{ip} = \begin{cases} 1 & \text{if all demand of hospital } i \text{ for product } p \text{ is satisfied} \\ 0 & \text{otherwise} \end{cases}$$

***Objective function*:**

Our objective function (4.1) maximizes the total weighted demand satisfied the Options O1, O2 and O3.

$$\text{Max } z = \sum_{v=1}^{V} \sum_{p=1}^{P} \sum_{i=IC+1}^{N} w_i \left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpk}\right) + \sum_{l \in CT} \sum_{v=1}^{V} \sum_{p=1}^{P} \sum_{i=IC+1}^{N} \alpha_{il} \left(FR_{ilpv} + \sum_{j=1}^{IC} FIR_{ijlpk}\right) + \sum_{p=1}^{P} \sum_{i=IC+1}^{N} \beta_i DR_{ip} \quad (4.1)$$

***Constraints*:**

$$\sum_{i=1}^{N} X_{ikv} \leq 1 \qquad \text{for } k = 1,2,\ldots,K;\ v = 1,2,\ldots,V \qquad (4.2)$$

$$\sum_{i=1}^{N} X_{i,k+1,v} \leq \sum_{i=1}^{N} X_{ikv} \qquad \text{for } k = 1,2,\ldots,K-1; v = 1,2,\ldots,V \qquad (4.3)$$

$$\sum_{k=1}^{K} X_{ikv} \leq 1 \qquad \text{for } i = IC+1, IC+2 \ldots, N;\ v = 1,2,\ldots,V \qquad (4.4)$$

$$\sum_{v=1}^{V} \sum_{k=2}^{N} X_{0kv} = \sum_{i=0}^{N} \sum_{v=1}^{V} X_{i2v} \qquad (4.5)$$

$$\sum_{k=1}^{K} kX_{ikv} \leq \sum_{k=1}^{K} kX_{0kv} \qquad \text{for } i = 1,2,\ldots,N;\ v = 1,2,\ldots,V \qquad (4.6)$$

$$\sum_{p=1}^{P}\left(R_{ipv}+\sum_{j=1}^{IC} IR_{ijpv}\right) \leq \left(\sum_{p=1}^{P}\left(D_{ip}+ID_{ip}\right)\right)\sum_{k=1}^{K} X_{ikv}$$

for $i = IC+1, IC+2, \ldots, N;$

$v = 1,2,\ldots,V$ (4.7)

$$\sum_{p=1}^{P}\left(FR_{ilpv}+\sum_{j=1}^{IC} FIR_{ijlpv}\right) \leq \left(\sum_{p=1}^{P}\left(D_{ip}+ID_{ip}\right)\right)\sum_{k=1}^{K} X_{lkv}$$

for $l \in CT;\ i \in DH_l; v = 1,2,\ldots,V$ (4.8)

$$\sum_{p=1}^{P}\left(FR_{ilpv}+\sum_{j=1}^{IC} FIR_{ijlpv}\right) \leq \sum_{p=1}^{P}\left(D_{ip}+ID_{ip}\right) UT_{ilv}$$

for $i, l = IC+1, IC+2, \ldots, N;$

$v = 1,2,\ldots,V$ (4.9)

$$\sum_{v=1}^{V} R_{ipv}+\sum_{v=1}^{V}\sum_{l\in DC_i} FR_{ilpv} \leq D_{ip}\,(1-DV_i)$$

for $i = IC+1, IC+2, \ldots, N;$

$p = 1,2,\ldots,P$ (4.10)

$$\sum_{v=1}^{V}\sum_{j=1}^{IC} IR_{ijpv}+\sum_{v=1}^{V}\sum_{l\in DC_i}\sum_{j=1}^{IC} FIR_{ijlpv} \leq ID_{ip}\,(1-DV_i)$$

for $i = IC+1, IC+2, \ldots, N;$

$p = 1,2,\ldots,P$ (4.11)

$$\sum_{i=IC+1}^{N}\left(\sum_{v=1}^{V}\left(R_{ipv}+\sum_{l\in DC_i} FR_{ilpv}+\sum_{j=1}^{IC}\left(IR_{ijpv}+\sum_{l\in DC_i} FIR_{ijlpv}\right)\right)+ DR_{ip}\right) \leq A_p$$

for $p = 1,2,\ldots,P$ (4.12)

$$\sum_{p=1}^{P}\sum_{i=IC+1}^{N}\left(R_{ipv}+\sum_{l\in DC_i} FR_{ilpv}+\sum_{j=1}^{IC}\left(IR_{ijpv}+\sum_{l\in DC_i} FIR_{ijlpv}\right)\right) \leq C_v$$

for $v = 1,2, \dots, V$ (4.13)

$Y_{ijv} \leq \sum_{k=1}^{K} X_{jkv}$ for $i = IC + 1, IC + 2, \dots, N;$

$j = 1,2, \dots IC; v = 1,2, \dots, V$ (4.14)

$S_{i1v} \geq (T_{0i} + EA_v)X_{i1v}$ for $i = 1,2, \dots, N;\ v = 1,2, \dots, V$ (4.15)

$$S_{jkv} \geq S_{i,k-1,v} + \left(a_i + b_i \sum_{p=1}^{P}\left(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv} + \sum_{l \in DH_i}\left(FR_{lipv} + \sum_{j=1}^{IC} FIR_{ljipv}\right)\right)\right) + T_{ij} - M_1\left(2 - X_{i,k-1,v} - X_{jkv}\right)$$

for $i = IC + 1, IC + 2, \dots, N;$

$j = 0,1, \dots, N;\ i \neq j; k = 1,2, \dots K;$
$v = 1,2, \dots, V$ (4.16)

$$S_{jkv} \geq S_{i,k-1,v} + \left(a_i + b_i \sum_{i'=IC+1}^{N} \sum_{p=1}^{P}\left(IR_{i'ipk} + \sum_{l=IC+1}^{n} FIR_{lii'pv}\right)\right) + T_{ij} - M_2 \times \left(2 - X_{i,k-1,v} - X_{jkv}\right)$$

for $i = 1,2, \dots, IC;\ j = 0,1, \dots, N;\ i \neq j;$

$k = 1,2, \dots K;\ v = 1,2, \dots, V$ (4.17)

$$\sum_{k=1}^{K} k \cdot X_{ikv} \geq \sum_{k=1}^{K} k \cdot X_{jkv} - K\left(1 - Y_{ijv}\right)$$

for $i = IC + 1, IC + 2, \dots, N;$

$j = 1,2, \dots, IC;\ v = 1,2, \dots, V$ (4.18)

$S_{ikv} \leq DD_i$ for $k = 1,2, \dots K;\ v = 1,2, \dots, V; i \in U$ (4.19)

$S_{0kv} \leq MT_v$ for $k = 1,2, \dots, K; v = 1,2, \dots, V$ (4.20)

$$\frac{\sum_{p=1}^{P}\left(IR_{ijpv}+\sum_{l\in DH_i} FIR_{ljipv}\right)}{\sum_{p=1}^{P}\left(ID_{ip}+\sum_{l\in DH_i} ID_{lp}\right)} \le Y_{ijv} \text{ for } i = IC+1, IC+2, \dots, N;$$

$$j = 1,2,\dots,IC; v = 1,2,\dots,V \qquad (4.21)$$

$$DR_{ip} \le \left(D_{ip} + ID_{ip}\right)DV_i \qquad \text{for } i \in SD;\ p = 1,2,\dots,P \qquad (4.22)$$

$$DR_{ip} = 0 \qquad \text{for } i \notin SD;\ p = 1,2,\dots,P \qquad (4.23)$$

$$\sum_{v=1}^{V} R_{ipv} + \sum_{v=1}^{V}\sum_{j=1}^{IC} IR_{ijpv} \ge \left(D_{ip} + ID_{ip}\right) WS_{ip}$$

$$\text{for } i \in CT;\ p = 1,2,\dots,P \qquad (4.24)$$

$$\sum_{v=1}^{V}\left(FR_{ilpv} + \sum_{j=1}^{IC} FIR_{ijlpv}\right) \le \left(\sum_{v=1}^{V}\left(D_{ip} + ID_{ip}\right)\right)WS_{lp}$$

$$\text{for } l \in CT;\ i \in DH_l;\ p = 1,2,\dots,P; i \notin U$$

(4.25)

$$S_{ikv} \ge S_{jkv} + T_{ji} - \left(MT_v + T_{ji}\right)\left(1 - UT_{ijv}\right)$$

$$\text{for } j \in CT;\ i \in DH_j; i \in U; k = 1,2,\dots K;$$

$$v = 1,2,\dots,V \qquad (4.26)$$

$$UT_{ijv} = 0 \qquad \text{for } j \notin CT; i \notin DH_j;\ i \notin U;$$

$$v = 1,2,\dots,V \qquad (4.27)$$

$$X_{ikv}, Y_{ijv}, DV_i, WS_{ip}, UT_{ii'v} \in \{0,1\}$$

$$\text{for } i, i' = 0,1,\dots,N; k = 1,2,\dots,K;$$

$$v = 1,2,\dots,V; j = 1,2,\dots,IC;$$
$$p = 1,2,\dots,P; \tag{4.28}$$

$$S_{ikv}, R_{ipv}, IR_{ijpv}, DR_{ip}, FR_{ilpv}, FIR_{ijlpv} \geq 0$$

$$\text{for } i,l = 0,1,\dots,N;\ p = 1,2,\dots,P;$$

$$j = 1,2,\dots,IC;\ v = 1,2,\dots,V \tag{4.29}$$

$R_{ipv}, IR_{ijpv}, DR_{ip}, FR_{ilpv}, FIR_{ijlpv}$ are integer

$$\text{for } i,l = 0,1,\dots,N;\ p = 1,2,\dots,P;$$

$$j = 1,2,\dots,IC; v = 1,2,\dots,V \tag{4.30}$$

where M_1 in (4.16) and M_2 in (4.17) are sufficiently large positive numbers and can be obtained by the following formulas:

$$M_1 = a_i + b_i \sum_{p=1}^{P}\left(D_{ip} + ID_{ip} + \sum_{l\in DH_i}\left(D_{lp} + ID_{lp}\right)\right) + MT_v$$

$$M_2 = a_i + b_i \sum_{i'=IC+1}^{N} \sum_{p=1}^{P}\left(ID_{i'p} + \sum_{l\in DH_{i'}} ID_{lp}\right) + MT_v$$

Constraint set (4.2) ensures that each vehicle can visit at most one node in any position on its route. Constraint set (4.3) ensures that if a node is assigned to a position, then there must be an assignment to the previous position. Constraint set (4.4) ensures each vehicle can visit a node at most one position on its route. Constraint set (4.5) states that the total number of vehicles departing from the RBC is equal to the total number of vehicles used. Constraint set (4.6) guarantees the RBC is visited as the last node by all vehicles. Constraint set (4.7) ensures that the demand of a hospital can be satisfied by a vehicle that visits the hospital. Constraint set (4.8) guarantees that any demand of a hospital (i) can be satisfied from another hospital (l) if the vehicle visits the hospital (l). Constraint set (4.9) links UT_{ilv} values with amount of satisfied demand via O2. Constraint sets (4.10) and (4.11) ensure the amount of total satisfied demand of a hospital by the Options O1 and O2 should not exceed its total (normal and irradiated) demand if the hospital does not use the Option O3. Constraint set (4.12) guarantees that the amount of products carried is limited by the amount available at the RBC. The capacities of the vehicles are considered in Constraint set (4.13). Constraint set (4.14) ensures that if the demand of a hospital is processed by an irradiation center, then the hospital and the irradiation center should be visited by the same vehicle. Constraint sets (4.15), (4.16), and (4.17) calculate the arrival time of vehicle v visiting node i in the first position, the next node after leaving node i, and the next node after leaving irradiation center i, respectively. Constraint set (4.18) guarantees that a hospital with demand for the irradiated products should be visited after the irradiation center serves to this hospital's demand. Constraint set (4.19) ensures that the arrival time of the vehicle to hospital i should not exceed the deadline imposed by the hospital. Constraint set (4.20) guarantees that the arrival time of the vehicle to the RBC does not exceed its maximum travel time. Constraint set (4.21) assigns hospital i to irradiation center j if the hospital i is served by irradiation center j. Constraint set (4.22) ensures that if a hospital uses the option O3, then the amount of demand satisfied by the Option O3 should not exceed the total (normal and irradiated) demand of this hospital. Constraint set (4.23) guarantees that a hospital cannot satisfy its demand by the Option O3 if it is not in the set of *SD*. Constraint set (4.24) controls the demand of a hospital is fully satisfied, where the hospital is used as a transfer point for another hospital that uses the Option O2. Constraint set (4.25) guarantees that a hospital can satisfy its demand by the Option O2 if the demand of the hospital used as a transfer point is fully satisfied. Constraint set (4.26) calculates the arrival time of the products to hospital i which has urgent demand if hospital j is used to satisfy the demand of hospital i in the route vehicle v. Constraint set (4.27) guarantees that decision variable UT_{ijv} (that controls the satisfaction of urgent demands by the Option O2) takes value zero if hospital j is giving transfer service to an hospital that is not covered in CT or hospital i is not in DH_j or the demand of hospital i is not urgent. Constraint sets (4.28), (4.29), and (4.30) represent the binary, non-negative, and integer variables.

We now present some properties of the optimal solution whose incorporation (via additional constraints) may reduce the size of the search space, hence improve the efficiency of the MILP. We use the same optimal properties given for Problem P1. In addition, we provide the following property and its associated constraint.

Property: If any hospital's total demand (normal and irradiated) for a product is greater than the availability of this product, then the hospital's demand cannot be fully satisfied. We use the following constraint to improve the efficiency of the model.

$$WS_{ip} = 0 \qquad \text{for } i = IC+1, IC+2, \dots, N; p = 1,2,\dots,P;$$

$$D_{ip} + ID_{ip} > A_p$$

(4.31)

Our mathematical model is explained by the objective function (4.1) and the constraint sets (4.2) through (4.30), the constraints (3.32), (3.33), (3.34) and (3.39) given by given for Problem P1, and the constraint (4.31).

A feasible solution for Problem P1 is also feasible for our problem P2 and some constraints (such as deadline and urgent demand constraints) in their problem are relaxed or not considered in our problem. This follows that the optimal solution of the first objective function value of their Problem P1 is a lower bound for our problem $P2$. We state this result formally by the following property.

Property: $z^*(P2) \geq z^*(P1)$

Proof: Note that any solution that is feasible for Problem $P1$ is also feasible for Problem $P2$. Thus, an optimal solution of $P1$ is also a feasible solution for $P2$. Hence, the optimal solution of $P2$ is greater than or equal to the optimal value of primary objective function at $P1$, i.e., $z^*(P2) \geq z^*(P1)$. ■

4.4 Proposed Solution Procedure - Hybrid Genetic Algorithm

We observe from computational tests that the MILP cannot solve medium and large sized problem instances since the size of the MILP drastically increase with an increase in the number of nodes and vehicles. Therefore, we propose a hybrid genetic algorithm (HGA) to solve the problem. Our hybrid genetic algorithm includes three stages: Iterative local search (ILS) based algorithm for initialization, genetic algorithm for improvement, and a linear programming model is solved for its finalization. The first two stages of the algorithm aim to find good vehicle routes and final stage allocates products to hospitals with considering the vehicle routes. We now give the detailed descriptions of each part used in our hybrid genetic algorithm.

4.4.1 Initialization: Iterative local search based algorithm

Our iterative local search based initialization algorithm involve four phases: *Finding an Initial Schedule, Improvement by Inserting Nodes to Vehicle Routes, Building a New Vehicle Schedule,* and *Improvement by Swapping (Pairwise Interchanging) Nodes*. Let *NI* be the number of iterations used by the algorithm and *NIS* be the number of initial schedules. The flowchart of the iterative local search based initialization algorithm is given in Figure 4.2.

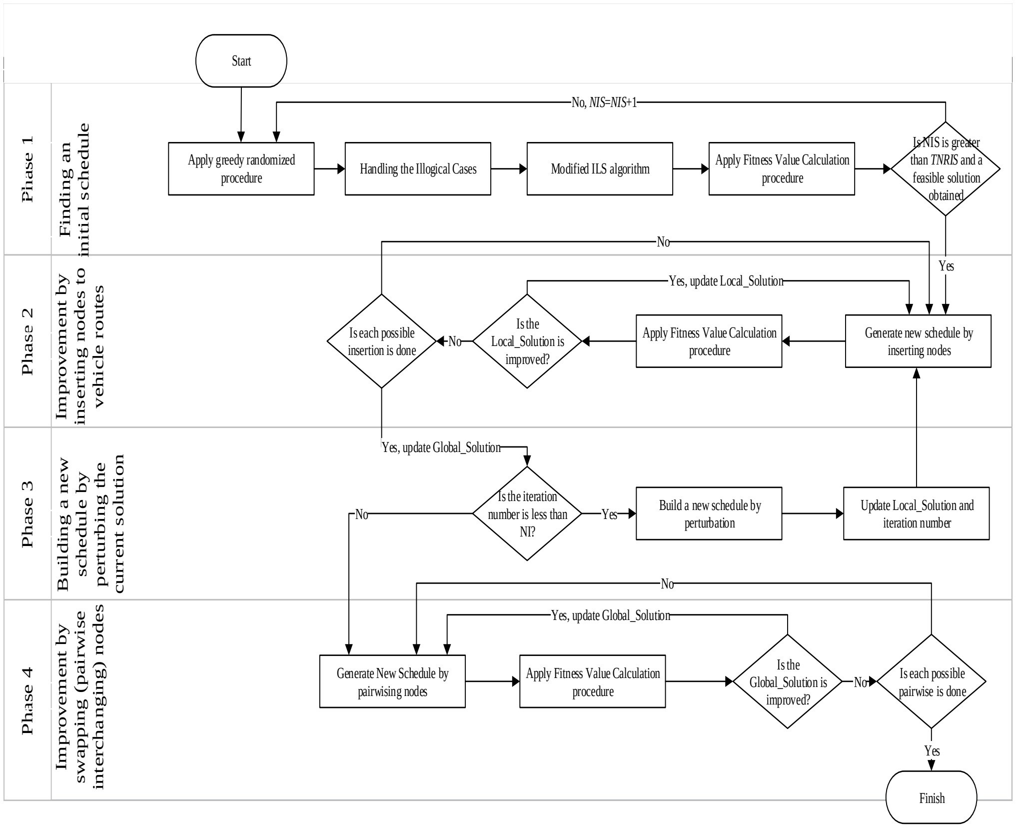

Figure 4.2: Flow of the iterative local search based initialization algorithm

We now give the detailed description of four phases of the iterative local search based initialization algorithm.

Phase 1: Finding an Initial Solution

This phase finds a feasible solution to MILP model given in Section 4.3 by relaxing the integrality requirements on variables R_{ipv}, IR_{ijpv}, FR_{ilpv}, FIR_{ijlpv}, and DR_{ip}.

We first apply a greedy randomized initialization procedure given below to assign the nodes to the vehicles and hospital set satisfying the demand by using Option O3. This initialization procedure may end up with some illogical cases. We handle these illogical cases by the rules explained below. Next, we calculate the loading and unloading times for each node by assuming all demand of this hospital is satisfied.

$$LO_{iv} = \begin{cases} a_i + b_i \sum_{i'=IC+1, i' \in v_i}^{N} \left(\sum_{p=1}^{P} ID_{i'p} \right), & i = 1,2,..,IC;\ i \in v_i; \\ & v = 1,2,\dots,V \\ a_i + b_i \sum_{p=1}^{P} \left(D_{ip} \right), & i = IC+1, IC+2, \dots, N; \\ & i \in v_i; v = 1,2,\dots,V \end{cases} \tag{4.32}$$

where v_i represents the nodes assigned to vehicle v.

We use the modified iterative local search (MILS) algorithm given for Problem P1 which is originally proposed by Arroyo et al. (2009) to solve single-machine tardiness problem with sequence-dependent setup times.

Finally, we calculate the fitness value with the procedure and given below to obtain a solution for the problem. We repeat this initial solution generation procedure for *TNRIS* (Total number of initial schedules) times and until we get at least one feasible solution. Then, we start with a feasible solution (vehicle routes and assignment of hospitals using the Option O3) which gives the best fitness value and set this feasible solution as the Local_Solution and the Global_Solution.

Greedy Randomized Initialization Procedure

The stepwise description of the greedy randomized procedure is given below.

Step 1. Calculate the potential normal demand PD_i, potential irradiated demand (PID_i) for each irradiation center and hospital:

$$PD_i = \sum_{p=1}^{P} D_{ip} \qquad \text{for } i = IC+1, IC+2, \dots, N \tag{4.33}$$

$$PID_i = \begin{cases} \sum_{p=1}^{P} ID_{ip}, & i = IC+1, IC+2, \dots, N \\ \dfrac{\sum_{j=IC+1}^{N} PID_j}{BV}, & i = 1,2,\dots,IC \end{cases} \tag{4.34}$$

Step 2. For each vehicle v, set the position index to zero, i.e., $k_v = 0$.

Step 3. Randomly select a vehicle v' between 1 and V.

Step 4. Calculate the potential gain ratio GR_i to be assigned to the selected vehicle v'.

$$GR_i = \begin{cases} \dfrac{(w_i-\beta_i)(PD_i+PID_i)}{T_{0i}+a_i+b_i(PD_i+PID_i)} & i = IC+1, IC+2, \dots, N; \\ & k_{v\prime} = 0 \\ \dfrac{(w_i-\beta_i)(PD_i+PID_i)}{T_{\left(ch_{k_{v'}-1,v'}\right),i}+a_i+b_i(PD_i+PID_i)} & i = IC+1, IC+2, \dots, N; \\ & k_{v'} > 0 \\ \dfrac{(PID_j)\times \max\limits_{j=IC+1,IC+2,\dots,N1}\{w_j\}}{T_{0i}+a_i+b_i(PID_i)} & i = 1,2,\dots,IC;\ k_{v\prime} = 0 \\ \dfrac{\max\limits_{j=IC+1,IC+2,\dots,N}\{w_j\}\times(PID_i)}{T_{\left(ch_{k_{v'}-1,v\prime}\right),i}+a_i+b_i(PID_i)} & i = 1,2,\dots,IC;\ k_{v'} > 0 \end{cases} \quad (4.35)$$

where ch_{kv} represents the node that are assigned to position k of set v.

Step 5. Calculate the maximum (*Gmax*) and minimum (*Gmin*) gain for the vehicle v'.

$$Gmax = \max_{i=1,2,\dots,N} GR_i$$

(4.36)

$$Gmin = \min_{i=1,2,\dots,N} GR_i$$

(4.37)

Step 6. Randomly select a node i' between 1 and N.

Step 7. If the following conditions satisfy

$$GR_{i'} > Gmax - PA \times (Gmax - Gmin)$$

(4.38)

$$DD_{i'} \geq \sum_{j=1}^{k_{v'}-1}\left(T_{ch_{j,v'},ch_{j+1,v'}} + a_{ch_{j,v'}} + b_{ch_{j,v'}} \times \left(PD_{ch_{j,v'}} + PID_{ch_{j,v'}}\right)\right) + EA_{v'} + T_{0,ch_{1,v'}} + T_{ch_{k_{v'},v'},i'} \quad (4.39)$$

$$MT_{v'} \geq \sum_{j=1}^{k_{v'}-1}\left(T_{ch_{j,v'},ch_{j+1,v'}} + a_{ch_{j,v'}} + b_{ch_{j,v'}} \times \left(PD_{ch_{j,v'}} + PID_{ch_{j,v'}}\right)\right) + T_{ch_{k_{v'},v'},i'} + a_{i'} + b_{i'} \times (PD_{i'} + PID_{i'}) + EA_{v'} + T_{0,ch_{1,v'}} + T_{i',0} \quad (4.40)$$

where *PA* is a parameter for our algorithm that represents the assignment ratio; then, we assign node i' to vehicle v' and update the following parameters:

$$PD_{i'} = (1 - AR)PD_{i'} \quad (4.41)$$

$$PID_{i'} = (1 - AR)PID_{i'} \quad (4.42)$$

$$k_{v'} = k_{v'} + 1 \quad (4.43)$$

$$ch_{k_{v'},v'} = i' \quad (4.44)$$

where *AR* is the assignment ratio which is a constant parameter represents the ratio of hospital's demand is satisfied by the assigned vehicle, and go to Step 3.

Otherwise (i.e., one of the conditions 4.38, 4.39 and 4.40 are not satisfied), then we do not assign node i' to vehicle v' and we repeat Step 6 until each node has been selected. If each node is selected, then we assign -1 to all remaining positions in vehicle v'.

Step 8. Go to Step 3 until each position of vehicle is filled.

Step 9. If a hospital is not assigned to any position of a vehicle but it can use Option O3 to satisfy its demand, then, we assign the hospital to hospital set which use Option O3, with a probability *PFD* (probability of direct visit), and repeat this step for each hospital.

Step 10. If a node is not assigned to any vehicle or set above, then we assign it to the set of hospitals which are not visited and not using the Option O3 to satisfy their demand.

Step 11. Put node zero (RBC) to the first -1 in each vehicle list.

We can call each solution obtained by schedule as chromosome.

Handling the Illogical Cases

As we stated before, the greedy randomized initialization procedure may provide some illogical cases. In this section, we investigate these cases.

Case 1: There is not any assigned hospital after an irradiation center in a vehicle list. We delete the irradiation center.

Case 2: There exists two irradiation center in a vehicle list and

- a hospital with urgent demand between two irradiation center and it has irradiated demand. We keep both irradiation centers.
- a hospital with routine demand between two irradiation centers and hospital has irradiated demand. We will delete the last scheduled irradiation center node. If the hospital does not have any irradiated demand, then we will delete irradiation center which creates more travelling time.

Case 3: There exists a hospital assigned to set of hospitals using Option O3 to satisfy its demand but it is not in the set of *SD*. We delete the hospitals from the list.

Case 4: A hospital assigned more than one to the list for set of hospitals using Option O3 and hospital set which is not visited and do not use Option O3. We keep one of them and delete the others.

Case 5: A hospital with urgent demand is assigned more than one to a list for vehicle routes, then we keep the first node assigned to vehicle and delete the others.

Case 6: A hospital is assigned to a vehicle, then we will delete it from the set of hospitals using Option O3 and hospital set which is not visited.

Case 7: A hospital with routine demand is assigned more than one to a list for vehicle, then we will delete the hospital which needs more travelling time.

Case 8: A hospital is not assigned to any vehicle list and cannot use Option O2 to satisfy its demand but it is in the set of *SD*. We assign it to the set of hospitals using Option O3.

Fitness value calculation

We may solve the LP given in Section 4.4.3 to evaluate the value of the objective function for a given vehicle route. However, we observe that solving this LP model for many times may require excessive amount of computation time. In place of solving this model, we propose the following method to find a powerful estimate (fitness) on its objective function value.

Firstly, we control the deadline and maximum travel time constraints only using the earliest available time of vehicles, travelling times, and constant loading and unloading times. If any of these time constraints are violated due to vehicle routes, then the solution is infeasible and we terminate the procedure with an objective function value which is the sum of the time constraints violations. Otherwise, we apply the following procedure to find the allocation of products to the hospitals.

In the MILP model given in Section 4.3, we have several binary variables. When we fix the values of some binary variables, we may easily solve the LP model in Section 5.3 and determine the allocation of the products to the hospitals. For this purpose, we use the following methods.

(1) [Setting UT_{ijv}values] UT_{ijv}'s are binary decision variables which control the hospitals with urgent demands to use the Option O2 to satisfy their demand. We first assume that only the hospitals, which are not assigned to any vehicle or hospital set using Option O3, can use Option O2. We secondly assume that only one hospital and vehicle pair can be used. Then, we use the following procedure to find UT_{ijv} values.

- For each visited hospital and vehicle pair, we calculate minimum arrival time (MA_{jv}) from the RBC to hospital *j* via vehicle *v*. The minimum arrival time (MA_{jv}) is the sum of the earliest available time of vehicles (EA_v), travel time between nodes visited before hospital j and fixed loading and unloading times (a_i) of the nodes visited before hospital j.
- Then, we find the time differences DBD_{ijv} between deadline of hospital *i* and the sum of the minimum arrival time (MA_{jv}) to hospital *j* and the time between hospitals *i* and *j*.

$$DBD_{ijv} = DD_i - \left(MA_{jv} + T_{ji}\right) \tag{4.45}$$

- Finally, we set UT_{ijv} to 1 for hospital i to satisfy its demand from hospital j and vehicle v pair providing the maximum DBD_{ijv} value.

(2) [Modifying α_{il} values] In our original model given in Section 4.3, we consider that a hospital i cannot be used to satisfy the demand of a specific product from

another hospital l if the demand of hospital l is not fully satisfied for that product. In order to deal with this consideration, we modify the weight (α_{il}) of satisfying the demand of hospital i from hospital l. The modified weight should be less than the weight of the hospital l which is used as a transfer point. By this way, we firstly satisfy the demand of the hospital with higher weight. The modified weight should consider the their original weights of hospital i and l. We modify our weights α'_{il} as:

$$\alpha'_{il} = \begin{cases} \frac{\alpha_{il}}{\max\limits_{i'}(\alpha_{i'l})+\varepsilon} w_l, \ w_l \leq \alpha_{il} \\ \frac{\alpha_{il}}{\max\limits_{i'}(\alpha_{i'l})+\varepsilon} \alpha_{il}, \ w_l > \alpha_{il} \end{cases} \tag{4.46}$$

where ε is the minimum weight, which is 0.5, in our problem instances.

(3) [Modifying DV_i values] If a hospital is not assigned to any vehicle and cannot satisfy its demand from another hospital by Option O2, then we add this hospital to the hospital set which satisfy its demand by Option O3 if it is in the set of *SD*.

Step 1. Calculate the estimated values:

- R_{ipv} by dividing hospital i's demand for normal product p to the total number of vehicles visiting hospital i,
- IR_{ijpv} by dividing the hospital i's irradiated demand for product p to the number of irradiated center visited before hospital i,
- FR_{ilpv} by dividing the hospital i's demand for normal product p to the number of hospitals which can be used as a transfer point for hospital i,
- FIR_{iljpv} by dividing the hospital i's irradiated demand for product p to the number of irradiated centers visited before hospital l which can be used as a transfer point for hospital i,

- DR_{ip} by summing the hospital i's normal and irradiated demand for product p.

  In order to calculate the FR_{ilpv} and FIR_{iljpv} values for hospital i with urgent demand. We use the hospitals and vehicle pairs which provide the UT_{ijv} values as one.

Step 2. [Update of the allocation amounts when the time constraints are violated]

a. Set $v = 1$.

b. Calculate the arrival time of vehicle v to the hospitals and the RBC.

c. Check the feasibility of deadline constraints with urgent demands (according to the sequence of hospitals visited by vehicle v). If one of them is violated, then

   c.1. Find the least profitable allocation that gives the following value

$$\min_i \left(\frac{w_i}{b_i}, \min_l \left(\frac{\alpha'_{li}}{b_i} \right), \frac{w_i}{b_i + b_j}, \min_l \left(\frac{\alpha'_{li}}{b_i + b_j} \right) \right) \tag{4.47}$$

   where i is the visited hospital and j is the irradiation center visited and used for satisfying the irradiation requirement of this hospital i, before a hospital (not inclusive) where the deadline constraint is violated.

   c.2 If the total violation is greater than the allocation amount multiplied by the variable loading and unloading time (b_i), then set the allocation amount to zero, update the violation amount, and go to Step 2.c.1. While updating the allocation amount, firstly we select product type if product availability constraint is violated, if there is no any product type violation, then we randomly select a product type to update allocation.

   Otherwise, reduce the allocation amount by the ratio of violation value to the variable loading and unloading time (b_i), and go to

Step 2.b. until all deadline and maximum travel time constraints in vehicle v are satisfied.

d. Otherwise, set $v = v + 1$, and go to Step 2.b.

Step 3. [Update of the allocation amounts when the product availability constraints are violated]

a. Set $p = 1$.

b. For product type p, calculate the total allocation.

c. Check the feasibility of product availability constraint. If the constraint is violated, then

c.1. Find the least profitable allocation (hospital which uses Option O1 or Option O3, or hospital pair which uses Option O2) that gives the following value:

$$\min_{i}\left(w_i, \min_{l}(\alpha'_{li}), \beta_i\right) \tag{4.48}$$

where hospital i having the allocation for product p.

c.2. If the total violation is smaller the allocation amount, then change the allocation amount to the amount of violation, set $p = p + 1$, and go to Step 3.b.

Otherwise, set the allocation amount to zero, update the violation amount, and go to Step 3.c.1.

d. Otherwise, set $p = p + 1$, and go to Step 3.b.

Step 4. [Update of the allocation amounts when the vehicle capacity constraints are violated]

a. Set $v = 1$.

b. Calculate the total allocation for vehicle v.

c. Check the feasibility of vehicle capacity constraint. If the constraint is violated, then c.1. Find the least profitable allocation (hospital which uses Option O1 or hospital pair which uses Option O2) that gives the following value:

$$\min_i \left(w_i, \min_l (\alpha'_{li}) \right) \tag{4.49}$$

where hospital i visited by vehicle v.

c.2. If the total violation is smaller the allocation amount, then change the allocation amount to the amount of violation, set $v = v + 1$, and go to Step 4.b.

Otherwise, set the allocation amount to zero, update the violation amount, and go to Step 4.c.1.

d. Otherwise, set $v = v + 1$, and go to Step 4.b.

Step 5. [Procedure for deleting unprofitable node]

If Step 4 returns a solution such that any hospital is visited with no service, we drop this hospital and go to Step 1; otherwise go to Step 6.

Step 6. Calculate the objective function value with the original weights and updated allocation amounts.

Phase 2: Improvement by inserting nodes to the vehicle routes

This phase tries to improve the solution by inserting additional nodes to the current vehicle routes since the value of our objective may increase as the number of assigned nodes increases. We consider the set of hospitals using Option O3 by setting its set as another vehicle (V+1). We use similar procedure for this set while inserting nodes. We eliminate some insertion of hospitals which may lead the Local_Solution to be an infeasible solution and whose potential gain with insertion is less than the average potential gain of all hospitals.

$$\sum_{p=1}^{P} \left(D_{ip} + ID_{ip} \right) - \sum_{v=1}^{V} G_{iv} \tag{4.50}$$

where G_{iv} is the amount of weighted demand already gained from hospital i assigned to vehicle *v* in the current Local_Solution.

We start by selecting a random node i' and a vehicle v' where i' is not already assigned to v' and this node-vehicle pair may improve the Local_Solution. If the selected vehicle is $V + 1$ (set for hospitals using Option O3) and hospital i' is proper for it, then we assign node i' to vehicle $V + 1$ and drop hospital i' from the other vehicles if it assigned to it. Otherwise, we randomly select *IPOS* number of insertion positions and calculate the travel time increase (*TTI*) for each selected position k as follows:

$$TTI_k = T_{f(k-1,v'),i'} + T_{i',f(k,v')} - T_{f(k-1,v'),f(k,v')} \tag{4.51}$$

where $f(k, v')$ is the hospital which is assigned to position *k* of vehicle v'. Then, we select the position k' which gives the minimum *TTI* value, and update our vehicle route v' by inserting node i' to position k'. Furthermore, we calculate the fitness value with the Fitness Value Calculation Procedure.

If the objective function value returned by the insertion is better than *API* (*API* is the acceptable percentage increase, and is used to reduce the solution time for the algorithm) times the objective function value of the current Local_Solution, then we replace the current Local_Solution with the new solution and repeat Phase 2. Otherwise, we select another node-vehicle pair, and repeat the procedure. We continue the process above until all possible node-vehicle pairs are examined. Finally, we update the Global_Solution if the objective function value of the current Local_Solution is better than that of the current Global_Solution.

Phase 3: Building a new vehicle schedule

The current Global_Solution may be a local optimal so that no further improvements could be achieved. In order not to get stuck in a local optimal solution, the current solution is perturbed either by removing a node from a vehicle route or interchanging two nodes in different vehicle routes.

We first calculate PV_{kv} values for each node k in vehicle route v of the Global_Solution, assuming that node k is between nodes i and j, as follows:

$$PV_{kv} = \min_{i'=IC+1,\dots,N} \left(\frac{w_{i'}}{b_{i'}}\right) \times \left(T_{ik} + T_{kj} + a_k - T_{ij}\right) - w_k \times PC_{kv} + \beta_k \sum_{p=1}^{P} \left(D_{kp} + ID_{kp}\right) \quad (4.52)$$

where PC_{kv} is the amount of products carried to node k by vehicle v, the maximum weight of the hospitals which are visited after an irradiation center by vehicle v is used as the weight for this irradiation center and the term "$\beta_k \sum_{p=1}^{P} \left(D_{kp} + ID_{kp}\right)$" is not considered in Equation 4.52 if node k is not in the subset SD including the hospitals satisfying their demand by the Option O3.

If the Global_Solution is updated in the current iteration, then we calculate the difference between the maximum travel time and the arrival time to the RBC for each vehicle, select the vehicle(s) giving the minimum difference, determine a node with the maximum PV_{kv} value among vehicle(s) with the minimum difference, and remove this node from its vehicle route. Otherwise, we randomly select two vehicles, determine a node with the maximum PV_{kv} value in each vehicle, and pairwise interchange these two nodes. Finally, we replace the Local_Solution with the solution obtained by the process above.

Phase 4: Improvement by swapping (pairwise interchanging) nodes

This phase improves the solution by swapping two nodes in four ways, (1) between the vehicle routes, (2) within a vehicle route, and (3) between a node in a vehicle route and a node which is not in any vehicle route, and (4) between a node in a vehicle route and a hospital which is in the set of hospitals using Option O3.

We randomly select two nodes that may provide better solution by using four ways mentioned above. We update our vehicle routes by interchanging the selected nodes and we drop the hospital i' from the its assigned vehicles if selected i' will be inserted to the set of hospitals using Option O3. Then, we calculate the fitness value by Fitness Value Calculation Procedure.

If the objective function value returned by the pairwise interchanging is better than *API* times the objective function value of the current Local_Solution, then we replace the current Local_Solution by the new solution and repeat Phase 4. Otherwise, we select another node-vehicle pair and repeat the procedure. We repeat the process above until all possible node pairs are examined. Finally, we update the Global_Solution if the objective function value of the current Local_Solution is better than that of the current Global_Solution.

The selection of the parameters (*AR*, *PA*, *PFD*, *TNRIS*, *AIP*, *NI*, *NIMILS*, *NRMILS*, and *AlphaMILS*) used in the iterative local search algorithm is discussed in the computational experiments section.

4.4.2 Improvement: Genetic Algorithm

To improve the initial solution, we propose a genetic algorithm. Genetic algorithms are first developed by Holland (1975). They use a mechanism that is close to the surviving way of the vital populations, where good individuals are usually from the combination of two good organisms, so called parents.

We give the flowchart of the genetic algorithm in Figure 4.3. As can be observed from the figure, our algorithm generates the initial population from the solution of the iterative local search based algorithm and improves it via crossover and mutation operators.

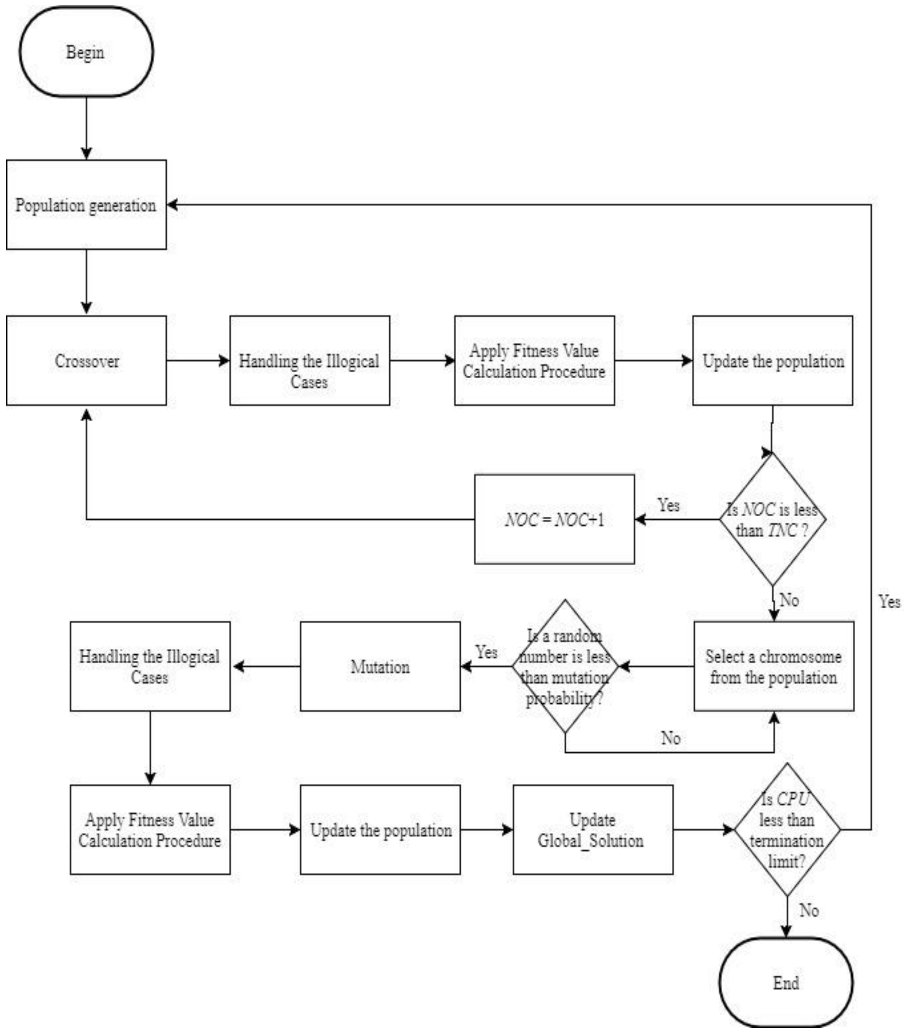

Figure 4.3: Flow of Genetic Algorithm

NOC: Number of crossover; *TNC*: Total number of crossover;

Chromosome Representation

Our chromosome representation scheme is similar to the ones used for the traveling salesman problem. The chromosome, so-called *genes*, reside $N \times (V + 2)$ components. The first $N \times V$ genes are for the vehicles, the next N genes are for the hospitals using Option O3 and the last N genes are for the hospitals with no assignments. We use positive integers for the nodes and "-1" for the unassigned positions.

To illustrate our representation scheme, consider 8-node and 2-vehicle instance. The routes 0-2-4-6-0 and 0-1-6-3-0 for vehicles 1 and 2, respectively. Option O3 is used by hospitals 5 and 7. Hospital 8 does not have any assignment. The resulting $32\ (= 8 \times (2 + 2))$ chromosomes are tabulated below.

Table 4.3: The chromosome representation for the example instance

Route for Vehicle 1 (Set 1)	2	4	6	-1	-1	-1	-1	-1
Route for Vehicle 2 (Set V)	1	6	3	-1	-1	-1	-1	-1
Set of hospitals using O3 (Set $V + 1$)	5	7	-1	-1	-1	-1	-1	-1
Unassigned Nodes (Set $V + 2$)	8	-1	-1	-1	-1	-1	-1	-1

Finding the Initial Population

Our initial population with the predetermined population size (PS) is formed as follows:

- The iterative local search algorithm (ILS) is solved to generate the first member.
- The greedy randomized algorithm, handling the illogical cases, and the modified ILS algorithm are solved successively to get the first half of the population.

- To generate the other members of the population, we use the ILS solution. We apply pairwise interchanges to two different genes and use the Handling the Illogical Cases Procedure, and the modified ILS algorithm.

Evaluating the Chromosome

To evaluate the performance of the chromosomes, we use the Fitness Value Calculation Procedure. We consider the infeasible solutions as well, as they may provide some better results with crossover and mutation operators.

Selecting the Parents

We select two chromosomes as parents for the generation of two children using the following ranking procedure.

Step 1. List the solutions in nonincreasing order of their fitness values.

Step 2. For each task in the list define a selection probability (SP_i) as follows:

$$SP_i = \frac{PS - \text{rank of } i + 1}{PS(PS+1)/2} \tag{4.58}$$

where SP_i values are higher for better chromosomes.

Step 3. Generate two random numbers from $U[0,1]$

Step 4. Select parents according to the generated numbers and SP_i values.

Crossover Operator

Crossover process is performed over two randomly selected parents. We start with randomly selecting two cut points between 1 and N. The genes between the selected cut points are preserved while other genes are crossed.

We illustrate the process on an example instance with following selected parents given in Table 4.4.

Table 4.4: Two selected parents for crossover

Parent 1

2	4	6	-1	-1	-1	-1	-1
1	6	3	-1	-1	-1	-1	-1
5	7	-1	-1	-1	-1	-1	-1
8	-1	-1	-1	-1	-1	-1	-1

Parent 2

3	7	-1	-1	-1	-1	-1	-1
1	8	4	-1	-1	-1	-1	-1
5	-1	-1	-1	-1	-1	-1	-1
6	2	-1	-1	-1	-1	-1	-1

First, we select two random numbers (a is the small number and b is the big number). Then, we preserve the genes between two random numbers (inclusive) of each gene set of parents 1 and 2, and create two lists (List 1 and List 2) for each parent. List 1 starts with gene one plus big random number of last set in Parent 1 (gene b+1 of set V +2) then continues with the ones that are not preserved for Parent 1. List 2 is created similarly. An example for the preserved genes and lists are given in Figure 4.4.

We create Offspring 1 using the preserved genes from Parent 1 and List 2. We put List 2 to the remaining genes of the preserved chromosome. We apply the same method for Offspring 2. After creating the offspring, we modify the chromosomes where the genes with “-1” must be at the end of sets. The crossover methodology is summarized in Figure 4.4.

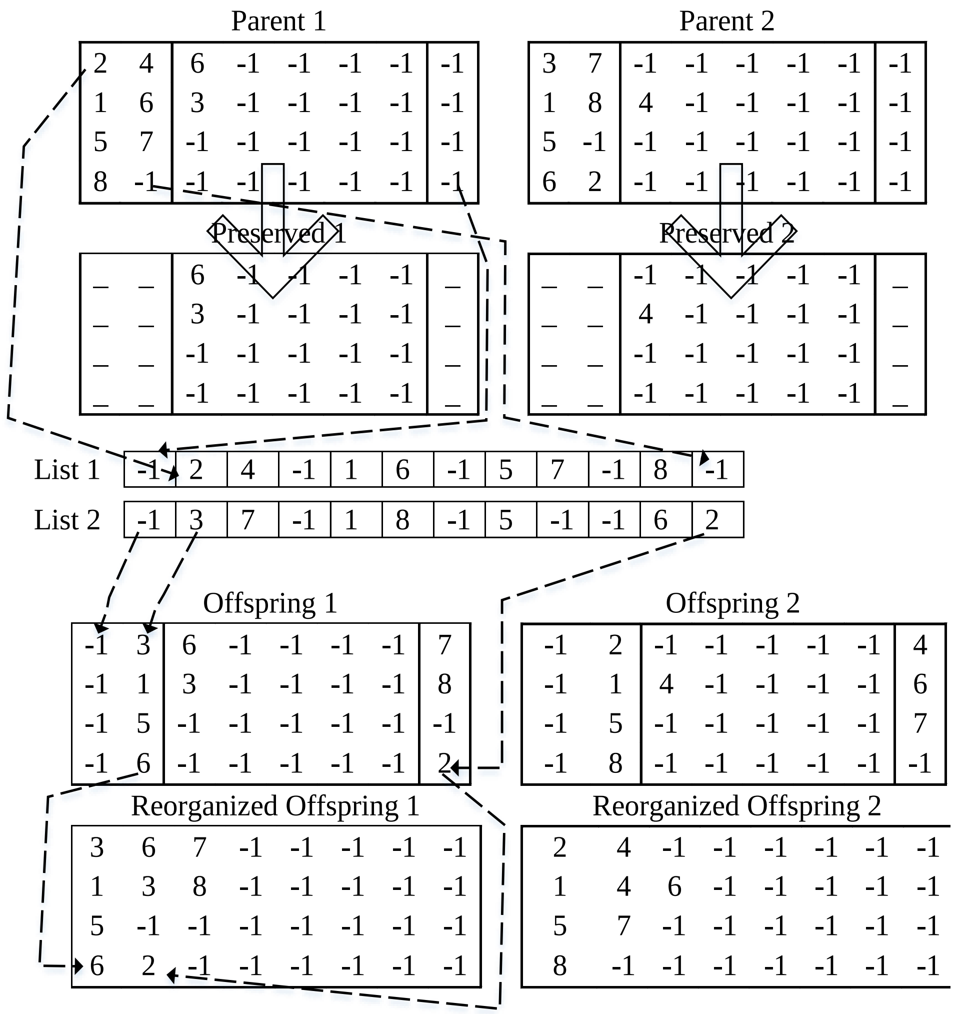

Figure 4.4: Illustration of the Crossover Operation

Resolving Offspring 1 and 2 may not give a feasible schedule for our problem, in such a case we must use the procedure of Handling the Illogical Cases.

Mutation Operator

Mutation operation for offspring is applied over the crossover operation with the hope of further improvement. The operation is performed only if a randomly generated probability value is greater than the predetermined mutation probability (*MP*). Below is the stepwise description of the mutation operation.

Step 1. Chromosome Insertion.

i. Eliminate some insertion for vehicles, nodes, and positions considering the deadline constraint of the nodes.
ii. Randomly select a vehicle v' between 1 and V.
iii. Calculate the potential gain for insertion (*PGI*) for each node and position. If inserted node i is in the set for V+1, then

$$PGI_{ik} = \frac{(w_i - \beta_i)\sum_{p=1}^{P}(D_{ip} + ID_{ip})}{T_{(ch_{k-1,v'}),i} + T_{i,(ch_{k,v'})} - T_{(ch_{k-1,v'}),(ch_{k,v'})} + a_i + b_i(PD_i + PID_i)} \quad \text{for } k = 1, \dots, K \tag{4.58}$$

If inserted node i is in the set for V+2, then

$$PGI_{ik} = \frac{\left(w_i - \max_{l}\{\alpha_{il}\}\right)\sum_{p=1}^{P}(D_{ip} + ID_{ip})}{T_{(ch_{k-1,v'}),i} + T_{i,(ch_{k,v'})} - T_{(ch_{k-1,v'}),(ch_{k,v'})} + a_i + b_i(PD_i + PID_i)} \quad \text{for } k = 1, \dots, K \tag{4.59}$$

iv. Select a node and position pair which provides maximum *PGI* value and does not violate the maximum travel time for vehicle v'.
v. Insert the selected node to the selected position of vehicle v'.
vi. Go to Step 2 until each vehicle is selected.
vii. Update the schedule and terminate the mutation operator.

If any node is inserted to a vehicle, then we stop the mutation operation. Otherwise, we apply Step 2.

Step 2. Pairwise interchange of two genes.

i. Calculate total travel time for the chromosome.
ii. Repeat the following procedure N times:
   a. Randomly select two genes from chromosome.
   b. Swap (pair-wise interchange) the two genes.
   c. Calculate the total travel time for the new chromosome.
   d. Calculate the change in the total travel time (difference between new and existing chromosomes).
iii. Select the genes which provide maximum gain in the total travel time (only consider the swaps which reduce the total travel time).
iv. Swap the selected genes.
v. Update the schedule and terminate the mutation operator.

If any node is swapped, then stop. Otherwise, we apply Step 3.

Step 3. Gene change

i. Select randomly one gene.
ii. Generate a number randomly: it is -1 with probability of 0.5, and it is between 1 and N with probability of 0.5.
iii. Replace the gene with the randomly selected different gene.
iv. Update the schedule and terminate the mutation operator.

We need to use Handling the Illogical Cases Procedure since resolving the chromosome does not give a reasonable solution.

Termination Procedure

We set a termination limit of $CTL \times N \times V$ seconds for the genetic algorithm, where CTL is predetermined constant. We set an upper limit of TNC operations, for the number of crossover operations.

The values of the parameters (*AR*, *PA*, *PFD*, *PS*, *MP*, *TNC*, *CTL*, *NIMILS*, *NRMILS*, and *AlphaMILS*) are set via a parametric analysis discussed in Section 4.5.3.

4.4.3 Finalization: A Linear Programming Model

Given the vehicle routes and the hospitals served directly from the RBC, we find the allocation of blood types using an LP model. Our LP model uses the assumed values (UT_{ilv}, α'_{il}, and DV_i) discussed in the Fitness Value Calculation Procedure and stated as below:

$$\text{Max } z_1 = \sum_{v=1}^{V} \sum_{p=1}^{P} \sum_{i=IC+1}^{N} w_i \big(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpk}\big) + \sum_{l \in CT} \sum_{v=1}^{V} \sum_{p=1}^{P} \sum_{i=IC+1}^{N} \alpha'_{il} \big(FR_{ilpv} + \sum_{j=1}^{IC} FIR_{ijlpk}\big) + \sum_{p=1}^{P} \sum_{i=IC+1}^{N} \beta_i DR_{ip} \tag{4.60}$$

Subject to

Constraints (10, 11, 12, 13, 15, 19, 20, 22) and

$$\sum_{p=1}^{P} \big(FR_{ilpv} + \sum_{j=1}^{IC} FIR_{ijlpv}\big) = 0 \quad \text{for } v = 1,2,\dots,V \ or\ l \notin DC_i\ or\ l \notin CT\ or \sum_{k=1}^{N} X_{lkv} = 0 \text{ or } (UT_{ilv} = 0 \text{ and } i \in U) \tag{4.61}$$

$$\sum_{p=1}^{P} \big(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv}\big)\big(1 - \sum_{k=1}^{N} X_{ikv}\big) = 0 \quad \text{for } i = IC+1, IC+2, \dots, N;\ v = 1,2,\dots,V; \tag{4.62}$$

$$S_{jkv} \geq S_{i,k-1,v} + a_i + b_i \sum_{p=1}^{P} \big(R_{ipv} + \sum_{j=1}^{IC} IR_{ijpv} + \sum_{l \in DH_i} \big(FR_{lipv} + \sum_{j=1}^{IC} FIR_{ljipv}\big)\big) + T_{ij} \quad \text{for } i = IC+1, IC+2, \dots, N;\ j = 0,1,\dots,N;\ i \neq j;\ v = 1,2,\dots,V;\ X_{jkv} = X_{i,k-1,v} = 1 \tag{4.63}$$

$$S_{jkv} \geq S_{i,k-1,v} + a_i + b_i \sum_{i'=IC+1}^{N} \sum_{p=1}^{P} \left(IR_{i'ipk} + \sum_{l \in DH_{i'}} FIR_{lii'pv} \right) + T_{ij}$$

for $i = 1,2, \dots, IC; j = 1,2, \dots, N;$
$k = 1,2, \dots, N;\ v = 1,2, \dots, V;$
$X_{jkv} = X_{i,k-1,v} = 1$ (4.64)

$$\sum_{p=1}^{P} IR_{ijpv} + \sum_{l \in DH_i} FIR_{liipv} = 0$$

for $j = 1,2, \dots IC; v = 1,2, \dots, V;$
$i = IC + 1, IC + 2, \dots, N;$
$\sum_{k=1}^{N} k \cdot X_{ikv} < \sum_{k=1}^{N} k \cdot X_{jkv}$ (4.65)

$$S_{ikv} \geq S_{jkv} + T_{ji}$$

for $j \in CT;\ i \in DH_j;\ i \in U;$
$k = 1,2, \dots N;\ v = 1,2, \dots, V;$
$UT_{ijv} = 1$ (4.66)

$$R_{ipv}, IR_{ijpv}, S_{ikv}, FR_{ilpv}, FIR_{ijlpv}, DR_{ip} \geq 0$$

for $i = 0,1, \dots, N;\ p = 1,2, \dots, P;$
$l \in DC_i;\ j = 1,2, \dots, IC;$
$v = 1,2, \dots, V;$ (4.67)

Objective function (4.60) maximizes the total weighted demand satisfied. Constraint set (4.61) guarantees that the demand of a hospital cannot be satisfied from a transfer point (another hospital) if the transfer point is not in the set for hospital or the vehicle does not visit the transfer point, or the corresponding UT_{ilv} value is zero for urgent demands. Constraint set (4.62) guarantees that if the vehicle does not visit a hospital, then it cannot deliver any product to that hospital. The arrival time of a vehicle to node j if node j is immediately visited after hospital i, and if node j is immediately visited after irradiation center i, are calculated in constraint sets (4.63) and (4.64), respectively. Constraint set (4.65) ensures the amount of irradiated product p carried to hospital i from irradiation center j by vehicle v should be zero, if irradiation center j is visited after hospital i. Constraint set (4.66) calculates the arrival time of the products to the hospitals with urgent demands and uses another hospital to satisfy the demand. Constraint set (4.67) represents non-negative variables.

We round all product quantities (R_{ipv}, IR_{ijpv}, FR_{ilpv}, FIR_{ijlpv}, and DR_{ip} values) returned by the model to the nearest integer values. If the solution is not feasible with rounded R_{ipv}, IR_{ijpv}, FR_{ilpv}, FIR_{ijlpv}, and DR_{ipv} values, we reduce the shipment to a hospital with the minimum weight (where the modified weights are used) by one unit until feasibility is maintained. Finally, we use the original hospital weights to obtain the resulting objective function value.

4.5 Computational Experiments

In this section, we describe the computational experiments designed to evaluate the performance of the solution approaches. The hybrid genetic algorithm is coded at the C++ platform, and C++ CPLEX application of IBM ILOG CPLEX optimization studio V12.6.2 is used to solve the mathematical models in our heuristic algorithm and the MILP model under the time limit of two hours. All computational experiments are conducted on a personal computer with Intel Xeon CPU E5-2650 2GHz (2 Processor) and 128 GB RAM under Windows 10 operating system.

4.5.1 Parameter settings

We use the same 360 real problem instances given for Problem P1. Those problem instances were encountered the Central Anatolian RBC between January 4 and February 4, 2016. In addition to the data in their problem instances, we have used the following methods of obtaining the additional parameters which are needed for our problem.

- Set of hospitals (*CT*) that can be used to satisfy the demand of other hospitals (*CT*) must have special refrigerator and enough storage space to keep the products. Therefore, each hospital in the responsibility of the Central Anatolian RBC cannot be in the set *CT*. The hospital set *CT* has been determined by a manager from the TRC who knows the facilities of each hospital.
- Set of hospital using Options O2 and O3 are determined by using the travelled time among the hospitals and hospital and RBC, respectively. A hospital can take its demand from another hospital if the travelled time between two hospitals is less than 20 minutes. A hospital can take its demand directly from the RBC if its traveled time to RBC is less than 20 minutes. These travel time limits are determined intuitively and experimentally. We run our randomly generated problem instances with different travel time limits from 5 to 30 minutes, and then we observed that our objective function value does not increase much after 20 minutes but CPU time to obtain the optimal solution drastically increases.
- Weight for satisfying demand by Option O2 (α_{il}) and satisfying demand by Option O3 (β_i) are determined with the help of a manager from the Turkish Red Crescent. The manager claims that the original weights of the hospitals will be reduced maximum ten and twenty percent if Options O2 and O3 are used, respectively, by considering the travel times among the hospitals and hospital and the RBC. Then, we create the weights α_{il} and β_i by the following formulas.

$$\alpha_{il} = w_i \left(1 - \frac{0.10 \times T_{il}}{\max_l T_{il}}\right) \tag{4.68}$$

$$\beta_i = w_i \left(1 - \frac{0.20 \times T_{i0}}{\max_i T_{i0}}\right) \tag{4.69}$$

With the above formulas, we also preserve the relative importance of the hospitals.

- In order to consider the urgent demands in our problem, we modified the weights of the hospitals with urgent demands. We modify its original weight by adding maximum relative weight in the problem instances which is ten in our problem instances.
- We use 2, 3, 4, and 5 vehicles in our problem instances.
- The length of a distribution period is 2 hours, and this period defines the maximum travel time allowed to each vehicle.
- There exist 2 irradiation centers.

The number of nodes and product ranges are given in Table 4.5.

Table 4.5: Number of nodes and product types in our data set

Group	1	2	3	4	5	6	7	8	9
N	10-14	15-19	20-24	25-29	30-34	35-39	40-44	45-49	50-54
P	8-19	8-23	13-23	15-21	16-22	17-24	20-27	19-30	20-28

4.5.2 Performance measures

To evaluate the performance of the heuristic algorithms, we consider the best non-integer solution (some of the variables are non-integer) obtained by MILP within two hours of time limit.

We use the following performance measure for each problem instances since best integer value and best non-integer value obtained by the MILP are same in the optimal solutions:

$$\text{Percent Error} = PE = \frac{z^N - z^H}{z^H} \times 100 \qquad (4.70)$$

where z^N is the value of the non-integer solution returned by the MILP model, and z^H is the value of the objective function returned by the heuristic algorithm. We use the best integer value returned by the MILP mode to evaluate its solution quality which is calculated as follows:

$$PE = \frac{z^N - z^B}{z^B} \times 100 \tag{4.71}$$

where z^B is the value of the best integer solution returned by the MILP model. The number of optimal solutions obtained is another performance measure for the heuristic algorithm. The optimality of the heuristic algorithm is assured when the objective function value is equal to the best non-integer objective function value of the MILP.

The CPU time is another performance measure used for the heuristic algorithm and the MILP model.

4.5.3 Parameter tuning for the heuristic algorithm

In this section, we discuss the selection of the parameters used by our heuristic algorithm. We use Groups, 1, 5 and 9 from Table 4.5 and set the number of vehicles to 2 and 5. Five random instances are used for each parameter combination.

Table 4.8 reports the values of the parameters used in the initialization part. For each value of the tested design parameters, we keep the other parameters at fixed values. The fixed values are also reported in Table 4.6. We give the PE values and CPU time for each tested level, and base our selection on the best trade-off between the PE values and CPU times. The selected values are also given in the table.

Table 4.6: The results of the parameter tuning- Initialization part

Parameter	Levels	Initialization part					
		1	2	3	Best	Selected	Fixed
Assignment ratio (*AR*)	Values	0.6	0.8	1.0			
	PE	2.7	2.5	2.2	1.0	1	0.8
	CPU	1.1	1.2	1.2	0.6		
Assignment probability (*PA*)	Values	0.3	0.5	0.7			
	PE	2.4	2.5	2.4	0.3, 0.7	0.7	0.5
	CPU	1.1	1.1	1.1	0.3, 0.5, 0.7		
Probability of direct visit (*PFD*)	Values	0.3	0.5	0.7			
	PE	2.4	2.5	2.2	0.7	0.7	0.5
	CPU	1.2	1.2	1.1	0.7		
Total number of initial schedule (*TNRIS*)	Values	10	20	40			
	PE	2.5	2.5	2.3	40	40	20
	CPU	1.2	1.2	1.1	40		
Number of insertion positions (*IPOS*)	Values	1	3	5			
	PE	2.4	2.5	2.4	1, 5	5	3
	CPU	1.2	1.2	0.9	5		
Acceptable percentage increase (*AIP*)	Values	1.000	1.001	1.002			
	PE	2.3	2.5	2.4	1.000	1.002	1.001
	CPU	1.8	1.2	0.9	1.002		
Number of iterations for the main algorithm (*NI*)	Values	3	4	6			
	PE	2.5	2.5	2.3	6	3	3
	CPU	1.2	1.2	1.4	3 ,4		
# of iterations for the Modified ILS Alg. (*NIMILS*)	Values	10	50	100			
	PE	2.5	2.2	2.4	50	50	10
	CPU	1.2	1.1	1.0	100		
# of runs for the Modified ILS Alg. (*NRMILS*)	Values	2	5	10			
	PE	2.7	2.5	2.5	5, 10	10	5
	CPU	1.1	1.2	1.1	2, 10		
Alpha value for Modified ILS Alg. (*AlphaMILS*)	Values	0.4	0.6	0.8			
	PE	2.5	2.5	2.5	0.4, 0.6, 0.8	0.6	0.6
	CPU	1.2	1.2	1.2	0.4, 0.6, 0.8		

For the improvement stage of the algorithm, we first decide on the termination limit (*CTL*). We test four values (1, 3, 6 and 9) for the *CTL* values. Table 4.7 reports on the performance values of each value of *CTL*.

Table 4.7: The results of the parameter tuning–*CTL* value

Value	1	3	6	9
Avg. PE	2.23	2.18	2.11	2.06
Avg. CPU (Min)	2.09	5.74	11.38	17.01

After analyzing the performance results we set *CTL* value to 6, as it leads to reasonable CPU times with good *PE* values. The *PE* values for *CTL* values of 6 and 9 are 2.11 % and 2.06 %, respectively, hence are very close.

For the other parameters' tunings, we set a constant termination limit and use *PE* for the selection criterion. We use three levels for each parameter and set the other parameters to the fixed values given in table.

Table 4.8: The results of the parameter tuning –Improvement part

Parameter	Levels	Improvement part				
		1	2	3	Selected	Fixed
Assignment ratio (*AR*)	Values	0.6	0.8	1.0		
	PE	2.15	2.20	2.20	1	0.8
Assignment probability (*PA*)	Values	0.3	0.5	0.7		
	PE	2.21	2.14	2.18	0.5	0.5
Probability of direct visit (*PFD*)	Values	0.4	0.6	0.8		
	PE	2.19	2.21	2.19	0.8	0.6
Population size (*PS*)	Values	5	10	20		
	PE	2.25	2.14	2.21	10	10
Mutation Probability (*MP*)	Values	0.1	0.5	0.9		
	PE	2.20	2.14	2.22	0.5	0.1
Total number of crossover (*TNC*)	Values	40	80	120		
	PE	2.14	2.13	2.17	80	40
# of iterations for the Modified ILS Alg. (*NIMILS*)	Values	10	50	100		
	PE	2.21	2.19	2.21	50	10
# of runs for the Modified ILS Alg. (*NRMILS*) (*NRMILS*)	Values	2	5	10		
	PE	2.21	2.20	2.19	10	2
Alpha value for Modified ILS Alg. (*AlphaMILS*)	Values	0.4	0.6	0.8		
	PE	2.13	2.21	2.14	0.4	0.6

The table also includes the parameter value fixed while selecting the other parameters. The selected value of the parameter is the one that leads to the smallest *PE* value.

4.5.4 Discussion of the results

In this section, we discuss the results of the experiment for the MILP model and the heuristic algorithm.

We observe that the MILP returns optimal solution only for the small-sized problem instances with up to 25 nodes, and report the related CPU times in Table 4.9. We set a termination limit of 2 hours for the MILP model.

Table 4.9: The CPU times of the MILP model – small-sized instances

N	*V*	CPU (Sec.) Avg.	CPU (Sec.) Max.	# of solved in CPU limit of 2 hours
10-14		3033	7200	6
15-19	2	7200	7200	0
20-24		7200	7200	0
Avg., *Max*., (sum)		**5811**	***7200***	**(6)**
10-14		137	641	10
15-19	3	6670	7200	2
20-24		7200	7200	0
Avg., *Max*., (sum)		**4669**	***7200***	**(12)**
10-14		103	452	10
15-19	4	3509	7200	7
20-24		6865	7200	1
Avg., *Max*., (sum)		**3492**	***7200***	**(18)**
10-14		105	250	10
15-19	5	2608	7200	8
20-24		6105	7200	3
Avg., *Max*., (sum)		**2939**	***7200***	**(21)**
Avg., *Max*., (sum)		**4228**	***7200***	**(57)**

Note from the table that even for the small-sized problem instances. The majority of the problems could not be solved in our termination limit of two hours.

We observe that the performance of the MILP highly depends on the problem size parameters. Any increase in N and V leads to significant changes in CPU times. Note that when there are 10-14 nodes, the average CPU time decreases from 3033 to 105 seconds. When the number of vehicles increases from 2 to 5. When the average number of vehicles is 4, the average CPU times increases from 103 to 6865 seconds when the number of nodes increases from 10-14 to 20-24.

We report the PE values of the heuristic algorithm for the small-sized problem instance whose optimal solutions are returned by the MILP, in Table 4.10.

Table 4.10: Performance of heuristic algorithm – the instances with known optimal solutions

N	*V*	# of instances	Avg. CPU (Sec.)	*PE* (%)		# of optimum solutions
				Avg.	Max.	
10-14	2	6	148.90	0.00	0.00	6
10-14	3	10	231.85	0.00	0.03	9
15-19		2	337.59	0.00	0.00	2
10-14	4	10	308.98	0.00	0.00	10
15-19		7	422.57	0.00	0.00	7
20-24		1	485.19	0.00	0.00	1
10-14	5	10	386.30	0.00	0.00	10
15-19		8	534.53	0.00	0.00	8
20-24		3	617.95	0.02	0.06	2
Avg., *Max*., (sum)		**(57)**	**358.12**	**0.00**	***0.06***	**(55)**

The HGA uses the parameters that are fine tuned in the previous section. Accordingly, we set the termination limit for the instances with N nodes and V vehicles to $N \times V \times 6$ seconds.

Note from the table that the HGA obtains optimal solution for the 55 out of 57 instances. The gaps of the unsolved instances are 0.03 % and 0.06 %. These values show the outstanding performance of our HGA over small-sized problem instances.

Table 4.11 reports on the performance the MILP model and the HGA over all problem sets, residing 360 problem instances.

Table 4.11: Performance of heuristic algorithm and the MILP model – over all problem set

		PE (%)			
		MILP model		HGA	
N	*V*	Avg.	Max	Avg.	Max.
10-14		0.39	1.66	0.17(6)*	0.60
15-19		1.09	3.46	0.80	1.64
20-24		6.50	17.89	4.00	11.42
25-29		8.19	22.12	4.23	15.74
30-34	2	10.77	27.80	4.44	10.47
35-39		10.62	18.65	5.14	8.17
40-44		9.75	16.07	4.96	8.59
45-49		15.62	23.76	6.58	9.67
50-54		16.65	22.27	6.86	12.37
10-14		0.00	0.00	0.00(9)*	0.03
15-19		0.35	2.10	0.03(6)*	0.16
20-24		5.77	23.87	2.47	14.96
25-29		5.26	13.74	2.18	11.83
30-34	3	8.05	12.51	2.61	8.12
35-39		8.92	14.81	2.55	3.48
40-44		20.23	46.41	3.13	5.14
45-49		17.82(2)**	25.28	4.69	7.83
50-54		23.00(1)**	32.55	5.01	8.91
10-14		0.00	0.00	0.00(10)*	0.00
15-19		0.03	0.20	0.00(10)*	0.00
20-24		3.92	12.85	0.80(1)*	4.25
25-29		3.64	9.20	0.66	2.39
30-34	4	7.66	12.29	1.80	6.67
35-39		11.92	20.78	1.61	2.98
40-44		12.06	25.21	2.14	3.95
45-49		20.20(4)**	24.84	2.89	4.38
50-54		21.98(1)**	27.99	3.47	5.73
10-14		0.00	0.00	0.00(10)*	0.00
15-19		0.03	0.30	0.00(10)*	0.00
20-24		2.94	8.32	0.49(5)*	4.33
25-29		2.22	5.70	0.28(5)*	1.54
30-34	5	7.28	11.26	1.07	5.23
35-39		12.57	37.23	1.03	3.21
40-44		11.58	17.33	1.25	2.58
45-49		19.73(4)**	32.77	2.21	5.39
50-54		22.45(1)**	31.65	2.42	4.42
Avg., *Max*., (sum)		**9.14(13)****	***46.41***	**2.28(72)***	***15.74***

*Number of times HGA equals to the upper bound
**Number of times no feasible solution is found in 2 hours.

We observe that the MILP fails to return any feasible solution of 13 out of 360 problem instances, after 2 hours of execution.

From Table 4.11, we also observe that heuristic algorithm returns optimal solutions for 72 out of 360 problem instances. We also observe that the *PE* from the non-integer solutions returned by the MILP of heuristic algorithm is 2.28 percent. It means *PE* from the optimal solution is less than 2.28 percent. Recall that the CPU times of the heuristic are fixed to $CTL \times N \times V$ seconds, whereas those instances could not be solved in two hours by the MILP. Hence, our algorithms provide much better results than the MILP in relatively short CPU times.

We can see from Table 4.11 that the PE of both MILP model and HGA increases when the number of nodes increases since problem size increases. Note that when there are 2 vehicles, the *PE* values for the MILP model are 0.39 and 16.65, for $N = 10 - 14$ and $N = 50 - 54$, respectively. For the HGA, the respective *PE* values are 0.60 and 12.37 for $N = 10 - 14$ and $N = 50 - 54$. This is due to the fact that the number of integer variables increases and the differences between integer solution and the relaxation of the problem getting bigger. Number of vehicle does not significantly affect the PE of the MILP model but it significantly reduces the PE of the heuristic algorithm. It may be due to the number of alternative solution increases when the number of vehicles increases.

We can conclude that the MILP does not perform well since most of the problem instances are not optimally solved by the MILP model in two hours. Moreover, the MILP model cannot obtain an integer feasible solution for 13 problem instances within 2 hours of CPU time and its relative gap between best non-integer and best integer solution is too high. On the other hand, our heuristic algorithm provides high quality solutions in relatively small pre-determined CPU times. Heuristic algorithm returns optimal solutions for 72 problem instances (out of 360). Hence, we can suggest user to use the heuristic algorithm to solve the problem of any size.

We also analyze the effects of the number of hospitals served, and the number of vehicles on the maximum weighted satisfied demand (z) and report the results in Table 4.12.

Table 4.12: Objective function values of the heuristic algorithm

V	2	3	4	5
N	Avg. z	Avg. z	Avg. z	Avg. z
10-14	655.99	656.90	656.90	656.90
15-19	641.89	645.54	645.65	645.65
20-24	1104.78	1132.76	1142.33	1143.73
25-29	1215.29	1239.25	1247.73	1252.73
30-34	2038.22	2076.31	2097.36	2111.18
35-39	2247.37	2308.56	2328.20	2339.60
40-44	3531.62	3601.83	3639.41	3659.55
45-49	4226.65	4332.67	4403.79	4440.29
50-54	4350.45	4446.57	4512.60	4554.71
Avg.	**2223.59**	**2271.15**	**2297.11**	**2311.59**

Note from the table that, increasing the number of vehicles increases z value. The increases are more significant when the number of vehicles is small. When the number of nodes is between 45-49 and the number of vehicles increases from 2 to 3, the total satisfied demand increases to 4332.67 from 4226.65. For the same case, when the number of vehicles increases from 4 to 5, the total satisfied demand increases to 4440.29 from 4403.79. Using the results of Table 4.12, the managers may analyze the trade-offs between number of vehicles and z value and select the best solution according to their preferences.

4.5.5 Sensitivity analysis

In this section we aim to analyze the effects of some critical parameters on the maximum total weighted satisfied demand amount (z). We select the maximum travel times for vehicles (MT_v), availability of products (A_p), and weight of the hospitals (w_i) as critical parameters for the vehicles, products and hospitals, respectively.

For the effects of the MT_v, we select two levels: 120 minutes to represent low and 180 minutes to represent high travel times. To see the effect of product availabilities, i.e., A_p values, we select two levels: the original product availabilities and twice of the original product availabilities for the second level to represent low and high availability levels, respectively. To see the effects of hospital weights we use two levels. The low weight level is represented by the original data where the weights are distributed between 0.5 and 10. To find the high weight level instances, we take the weights of the low weight instances and double the ones that are below 5; hence, the resulting weights are distributed between 1 and 10.

We select three problem combinations (Group 1– small sized instances with 10-14 nodes, Group 5 – medium sized instances with 30-34 nodes, and Group 9– large sized instances with 50-54 nodes) from Table 4.5 and report the associated results in Table 4.13. We set the number of vehicles to 2.

Note from Table 4.13 that the z values increase from 4676.93 to 4803.22 when the maximum travel times increase from 120 to 180 minutes for large sized instances. This is due to the fact that by increasing travel times, more room becomes available for demand satisfactions. We also observe that increasing the maximum travel time does not have a significant effect on the z values for the small and medium sized problem instances.

Table 4.13: Sensitivity analysis based on the parameters

		Original	High MT_v	Twice A_p	Modified w_i
	Group 1	923.67	925.50	942.19	1510.84
z	Group 5	2173.56	2229.34	2208.16	3489.09
	Group 9	4676.93	4803.22	5115.04	6619.08

We also observe that z values slightly improve when the product availability increases. The improvement is more significant for the large-sized instances. It is less than expected and we can conclude that the availability of product is not a highly binding constraint for satisfying the demand of hospitals.

Table 4.13 also shows that when the some hospital weights are doubled, the z values increase from 623.67 to 1510.84 (about 64%), from 2173.56 to 3489.09 (about 60%), and 4676.93 to 6619.08 (about 41%) for the small, medium, and large-sized instances, respectively.

CHAPTER 5

CONCLUSIONS AND FUTURE RESEARCH ISSUES

In this study, we consider the blood distribution system in Turkey and focus on the Central Anatolian Regional Blood Center. We studied two problems for the blood distribution system in Central Anatolian Regional Blood Center.

To the best of our knowledge, our first problem is the first attempt that considers irradiation centers, urgent demands and blood products availability. Including irradiation centers leads to precedence structure where product availability may lead to partial satisfaction of the demands of some customers. Hence, we study the allocation of the demand along with their distribution to the hospitals.

We aim to maximize the amount of blood products carried to the hospitals for each distribution period of specified length. Among the alternative optimal solutions of the problem, we select the one that minimizes the total time travelled.

We model the Problem 1 as a MILP model and improve its efficiency using the properties of the optimal solutions. We find that the MILP model cannot be solved within our termination limit of one hour. Having experienced the difficulty of attaining optimal solutions through the MILP, we propose two decomposition-based heuristic procedures. We collect and organize real data to test the performances of the MILP and heuristic procedures. We analyze the effects of the problem size parameters (number of hospitals, number of vehicles) on the difficulty of the solutions and on the maximum satisfied demand and total time travelled values. We find that our heuristics return high quality solutions at reasonable times and can be used to solve large sized problem instances. We also observe that the maximum satisfied demand and total time travelled are sensitive to the number of vehicles, number of the hospitals, and weight of the hospitals. We study the trade-off between the problem size parameters and maximum satisfied demand and total time travelled values, we recommend the managers to select the number of vehicles that best fulfils their demand satisfaction and travel cost concerns.

Our second problem extends the first problem with considering two more demand satisfaction options. We aim to maximize the amount of blood products carried to the hospitals for each distribution period of specified length.

We model the Problem 2 as a MILP model and improve its efficiency using the properties of the optimal solutions. We find that the MILP model cannot be solved within our termination limit of two hours. Having experienced the difficulty of attaining optimal solutions through the MILP model, we propose a hybrid genetic algorithm with three stages: initialization based on iterative local search algorithm, improvement based on genetic algorithm and finalization is based on a linear programming. We collect and organize real data to test the performances of the MILP and heuristic procedures. We analyze the effects of the problem size parameters (number of hospitals, number of vehicles) on the difficulty of the solutions and on the maximum satisfied demand and total time travelled values. We find that our heuristic algorithm returns high quality solutions at reasonable times and can be used to solve large sized problem instances. We also observe that the maximum satisfied demand is sensitive to the number of vehicles, number of the hospitals, and weight of the hospitals. We study the trade-off between the problem size parameters and maximum satisfied demand, we recommend the managers to select the number of vehicles that best fulfils their demand satisfaction concerns.

Our motivation in the study was to improve the distribution system of the TRC. We hope our study opens new research directions for theoretical and practical concerns. For theoretical purposes, the development of exact procedures using decomposition ideas might be worthwhile. The practitioners may extend our study to multiple RBCs, full demand satisfactions, and time windows on service times for hospitals and visits times of irradiation centers. Moreover, a strategic problem of opening new irradiation centers and defining their locations might be worth studying.

REFERENCES

1. Alfonso, E., Augosto, V. and Xie, X., 2013. Tactical planning of bloodmobile collection systems. In: Automation Science and Engineering (CASE), 2013 IEEE International Conference on, 26-31, IEEE.

2. Alfonso, E., Augosto, V. and Xie, X., 2015. Mathematical programming models for annual and weekly bloodmobile collection planning. IEEE Transactions on Automation Science and Engineering 12(1), 96-105.

3. Arroyo, J.E.C, Nunes G.V.P. and Kamke, E.H., 2009. Iterative local search heuristic for the single machine scheduling problem with sequence dependent setup times and due dates. In Hybrid Intelligent Systems, Ninth International Conference on Hybrid Intelligent Systems 1, 505-510, IEEE.

4. Arvan, M., Moghaddam, R.T. and Abdollahi, M., 2015. Designing a bi-objective, multi-product supply chain network for blood supply. Uncertain Supply Chain Management 3(1), 57-68.

5. Attari, M.Y.N., Pasandide, S.H.R, Agaie, A. and Akhavan Niaki, S.T., 2017. Presenting a stochastic multi choice goal programming model for reducing wastages and shortages of blood products at hospitals. Journal of Industrial and Systems Engineering 10, 81-96.

6. Belien, J. and Force H., 2012. Supply chain management of blood products: a literature review. European Journal of Operational Research 217(1), 1-16.

7. Chaiwuttisak, P., Smith, H., Wu, Y., Potts, C., Sakuldamrongpanich, T. and Pathomsiri, S., 2016. Location of low-cost blood collection and distribution centres in Thailand. Operations Research for Health Care 9, 7-15.

8. Daskin, M., Coullard, C. and Shen, Z.J., 2002. An inventory-location model: formulation, solution algorithm and computational results. Annals of Operations Research 110(1), 83-106.

9. Dillon, M., Oliveira, F. and Abbasi, B., 2017. A two-stage stochastic programming model for inventory management in the blood supply chain, International Journal of Production Economics, 187, 27-41.

10. Ensafian H. and Yaghoubi, S., 2017. Robust optimization model for integrated procurement, production and distribution in platelet supply chain. Transportation Research Part E: Logistics and Transportation Review 103, 32-55.

11. Fahimnia, B., Jabbarzadeh, A., Ghavamifar, A. and Bell, M., 2017. Supply chain design for efficient and effective blood supply in disasters. International Journal of Production Economics 183, 700-709.

12. Ghasemi, E. and Bashiri, M. 2017. A Selective Covering-Inventory-Routing problem to the location of bloodmobile to supply stochastic demand of blood. 13th International Conference on Industrial Engineering.

13. Gregor, P.J., Forthofer, R.N. and Kapadia, A.S., 1982. An evaluation of inventory and transportation policies of a regional blood distribution system. European Journal of Operational Research 10(1), 106-113.

14. Gunpınar, S. and Centeno, G., 2015. Stochastic integer programming models for reducing wastages and shortages of blood products at hospitals. Computers and Operations Research 54, 129-141.

15. Gunpinar, S. and Centeno, G., 2016). An integer programming approach to the bloodmobile routing problem. Transportation Research Part E 86, 94-115.

16. Hemmelmayr, V., Doerner, K.F., Hartl, R.F. and Savelsbergh, M.W.P., 2009. Delivery strategies for blood product supplies. OR Spectrum 31(4), 707-725.

17. Hemmelmayr, V., Doerner, K.F., Hartl, R.F. and Savelsbergh, M.W.P., 2010. Vendor managed inventory for environments with stochastic product usage. European Journal of Operational Research 202(3), 686-695.

18. Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The university of Michigan Press, Ann Arbor, Michigan, USA.

19. Hosseinifard, Z. and Abbasi, B., 2016. The inventory centralization impacts on sustainability of the blood supply chain. Computers and Operations Research. Articles in Advance.

20. Hsieh, C.L., 2014. An evolutionary-based optimization for a multi-objective blood banking supply chain model. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, Cham, 511-520.

21. Jabbarzadeh, A., Fahimnia, B. and Seuring, S., 2014. Dynamic supply chain network design for the supply of blood in disasters: a robust model with real world application. Transportation Research 70, 225-244.

22. Jacobs, D.A., Silan, M.N. and Clemson, B.A 1996. An analysis of alternative locations and service areas of American Red Cross blood facilities. Interfaces 26(3), 40-50.

23. Jyrki S. R., and Spens, M. K., 2006. Using simulation to increase efficiency in blood supply chains. Management Research News 29, 801–819.

24. Kaveh, A. and Ghobadi, M., 2017. A multistage algorithm for blood banking supply chain allocation problem. International Journal of Civil Engineering 15(1), 103-112.

25. Khalilpourazari, S. and Khamseh, A.A. 2017. Bi-objective emergency blood supply chain network design in earthquake considering earthquake magnitude: a comprehensive study with real world application. Annals of Operations Research, 1-39.

26. Nearchou, A.C. and Lagodimos, A.G., 2013. Heuristic solutions for the economic manpower shift planning problem. European Journal of Industrial Engineering 7, 657-686.

27. Ochi, L.S., Vianna, D.S., Drummond, L.M.A. and Victor, A.O., 1998**.** An evolutionary hybrid metaheuristic for solving the vehicle routing problem with heterogeneous fleet. In European Conference on Genetic Programming. Springer, Berlin, Heidelberg, 187-195.

28. Osorio, A.F., Brailsford, S.C. and Smith, H. K., 2015. A structured review of quantitative models in the blood supply chain: a taxonomic framework for decision-making. International Journal of Production Research 53(24), 7191-7212.

29. Pierskalla, W.P., 2005. Supply chain management of blood banks. Operations Research and Health Care. Springer, US, 103-145.

30. Puranam, K., Novak, D.C., Lucas, M.T. and Fung, M. 2017. Managing blood inventory with multiple independent source of supply. European Journal of Operational Research 259(2), 500-511.

31. Ramezanian, R. and Behboodi, Z. 2017. Blood supply chain network design under uncertainties in supply and demand considering social aspects. Transportation Research Part E: Logistics and Transportation Review 104, 69-82.

32. Rock, G., Akerblom, O., Berseus, O., Herve, P., Jacobs, P., Kelly, T., MacPherson, J., Nydegger, U., Segatchian, G., Urbaniak, S. and Valboseni, M., 2000. The supply of blood products in 10 different systems or countries. Transfusion Science 22(3), 171-182.

33. Rusman, M. and Rapi, A., 2014. Blood banks location model for blood distribution planning in Makassar City. In Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference 2014.

34. Sahin, G., Sural, H. and Meral, S., 2007. Locational analysis for regionalization of Turkish Red Crescent blood service. Computer and Operations Research 34(3), 692-704.

35. Sahinyazan, F.G., Kara, B.Y. and Taner, M.R., 2015. Selective vehicle routing for a mobile blood donation system. European Journal of Operational Research 245(1), 22-34.

36. Salehipour, A. and Sepehri, M.M., 2012. Exact and heuristic solutions to minimize total waiting time in the blood products distribution problem. Advances in Operations Research 2012, 1-25.

37. Salvietti, L., Smith, N.R. and Cárdenas-Barrón, L.E., 2014. A stochastic profit-maximizing economic lot scheduling problem with price optimization. European Journal of Industrial Engineering 8, 193-221.

38. Sha, Y. and Huang, J., 2012. The multi-period location-allocation problem of engineering emergency blood supply systems. System Engineering Procedia 5, 21-28.

39. Shen, Z.J.M, Coullard, C. and Daskin, M.S., 2003. A Joint Location-Inventory Model. Transportation Science 37(1), 40-55.

40. Wang, K.M. and Ma, Z.J. 2015. Age-Based policy for blood transhipment during blood shortage. Transportation Shortage Part E: Logistics and Transportation Review 80, 166-183.

41. Yegül, M., 2016. Blood supply network design. Ph.D. thesis, Middle East Technical University.Ankara, Turkey

42. Zahiri, B., Mousazadeh, M. and Bozorgi-Amiri, A., 2014. A robust stochastic approach for blood collection and distribution network design. International Journal Research in Industrial Engineering 3(2), 1-11.

43. Zahiri, B. and Pishvaee, M.S., 2017. Blood supply chain network design considering blood group compatibility under uncertainty. International Journal of Production Research 55(7), 2013-2033.

APPENDIX

Table A1. Information about RBC in Turkey

Regional Blood Center	Number of BC	Number of BS	Provinces which are in the responsibility of RBC
Aegean RBC	7	10	İzmir, Muğla, Aydın, Uşak, Manisa, Denizli
North Marmara RBC	5	1	İstanbul
Central Anatolian RBC	5	6	Ankara, Konya, Çorum, Kırıkkale, Kastamonu, Çankırı, Karaman
Central Mediterranean RBC	3	8	Adana, Mersin, Hatay, Osmaniye
East Mediterranean RBC	4	5	Gaziantep, Şanlıurfa, Kahramanmaraş, Kilis
South Marmara RBC	4	3	Bursa, Balıkesir, Yalova, Çanakkale
West Black Sea RBC	5	6	Düzce, Sakarya, Kocaeli, Zonguldak, Karabuk, Bolu, Bartın
Inner Anatolian RBC	5	1	Kayseri, Nevşehir, Kırşehir, Sivas, Yozgat, Aksaray, Niğde
West Mediterranean RBC	3	2	Antalya, Burdur, Isparta
Central Black Sea RBC	3	2	Samsun, Tokat, Ordu, Sinop, Amasya
West Anatolian RBC	3	1	Eskişehir, Kütahya, Afyon, Bilecik
East Anatolian RBC	2	3	Erzurum, Erzincan, Tunceli, Ardahan, Bayburt, Kars
East Black Sea RBC	5	0	Trabzon, Rize, Artvin, Giresun, Gümüşhane
South Anatolian RBC	3	2	Diyarbakir, Siirt, Batman, Şırnak, Bingöl, Mardin
South West RBC	2	0	Malatya, Elazığ, Adıyaman
South East RBC	2	0	Van, Muş, Hakkari, Ağrı, Iğdır, Bitlis
Europe RBC	3	4	İstanbul, Edirne, Kırklareli, Tekirdağ
Total	64	54	

RBC: Regional Blood Center, BC: Blood Center, BS: Blood Stations

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Kurt, Atıl
Nationality: Turkish (TC)
Date and Place of Birth: 1 September 1986, Malatya
Marital Status: Single
Phone: +90 538 326 45 63
Fax: +90 312 233 10 26
email: atil.kurt.ie@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
MS	Cankaya University Industrial Engineering	2012
BS	Cankaya University Industrial Engineering	2008
High School	Seyranbağları High School, Ankara	2003

WORK EXPERIENCE

Year	Place	Enrollment
2009- Present	Cankaya University Ind. Eng. Dept.	Expert
2007 July	Arçelik Refrigerator Factory	Intern Engineering Student
2006 July	Hidromek	Intern Engineering Student

FOREIGN LANGUAGES

High-intermediate English

HOBBIES

Latino dances, reading, and movies