A Linear Programming Model for a Blood Inventory Management at a Regional Blood Bank

Nour Alkhaja[1], Assia Chadly[1], Abdulrahman Almarzooqi[1], and Arwa Fikri [1]

[1]Khalifa University of Science, Technology and Research

May 16, 2020

Abstract

Models addressing the supply chain in blood banks are of goals, but the common theme is the tendency to focus on optimum demand meeting while minimizing the number of expired units. This paper provides a simplified LP model for inventory management. The objective function in this model formulation aims to minimize the cost and the number of wasted units, considering the unit limited shelf life. The implementation of the model is to be at a regional blood bank that receives scheduled orders. The parameters values were based on data collected from Abu Dhabi regional blood bank; it was then implemented using the Gurobi computational solver.

Keywords: blood, waste, collection plan, operations research, python, sensitivity analysis

1. Introduction

The blood supply chain is considered of a vital priority, and its study has occupied a large portion of operation research. Studies fro this supply chain provided analytical and simulation models. Since this problem comprises multiple process and products, on different decision-making levels, a few studies focused on addressing the supply chain as a whole while the majority focused on certain aspects of it. Those aspects include, but not limited to, the development of approximate ordering and issuing policies for inventory systems, distribution scheduling, and the analysis of multi-product system. They targeted those models to be implanted as decision rules within an individual hospital blood bank, or on a larger scale as a decision support system for hierarchical planning in a regional blood centre. Blood as a product has a limited shelf life of around 35 days; we can recognise four main blood groups which are determined by the presence or absence of two antigens A- and B- on the red blood surfaces, but the presence of a protein called the Rh factor that gives a total of eight major blood types (O +, O-, A+, A-, B+, B-, AB+, and AB-).

In blood transfusion systems, it is vital that patients are assigned to the correct blood types, since not all blood types are compatible with each other and mixing incompatible blood types can lead to agglutination. Collected blood units can be further broken down into separate products of blood: red cells, white cells, plasma and platelets. They have different usage in the medical treatment of patients and serve different functions in the human body. Blood shelf life considered to be of a critical factor in blood assignment-transfusion system; since blood units that are not used within their life span considered to be as wastage units ‘expired’. The complicated nature of this supply chain calls for a thorough review to previous works and necessitate a breakdown for its different stages within the following section.

2. Contribution

This paper provides an overview of the theory and the practice of the blood supply chain and inventory management. A linear programming model was developed to optimise the number of daily collected units at a regional blood bank. The study focused on challenging the complex nature of the blood supply chain by providing a simple model for inventory management while minimising cost and waste. The model addressed scheduled demand from a steady number of donors. This study based its parameters on data collected from Au Dhabi regional blood bank and implemented the model with the use of Gurobi solver.

3. Literature Review

Blood supply chain and inventory management are considered a significant field of operation research since the 1980s; attracting the focus of hundreds of researchers. Therefore, numerous publications and papers addressed the various aspects of the supply chain concerning this vital human-sourced perishable product. This supply chain could be broken down into four consecutive stages: collection, production, inventory and distribution towards the endpoint at hospitals (Osorio 2016). However, studies have shown that this system is far from linear and of a highly complex nature caused by unpredictability in both demand and supply, the limited shelf life of blood products, and the variations in issuing policies based on the hierarchy of the blood provider and how it operates. Those are few among many other influential factors contributing further to the complexity of the blood supply chain.

Gregory P.Prastacos, an early pioneer in this realm of operation research, provided in his paper (Prastacos 1984) a thorough, comprehensive explanation for the different stages of a blood supply chain; starting with blood collection either at the hospital itself or at donation points that feed a regional blood bank. Both the hospital and regional blood bank are stages within the decision-making pyramid, the following third level of this hierarchy would be the inter-regional blood centre which handles the coordination between several regional centres while also supervising the national blood program (Prastacos 1984).

Prastacos clarifies that :

> “at each of these levels, the decision-maker faces a wide range of operational, tactical, and strategic management problems”.

When modelling such a complex problem, we should be aware of two variables : (i) the limited shelf life of this product, and (ii) the difference between actual demand and usage. Optimum utilisation of assigned units is considered one of the most important criteria used to evaluate the performance of a blood provider, which is the number of wasted units (Prastacos 1984).

The number of needed blood unites usually is decided by each hospital based on a policy called (MSBOS), which stands for the maximum surgical blood order schedule. The number and type of ordered blood units, therefore, would rely on the classification of that procedure, the number of patients in the hospital, median estimated blood loss and transfusion index (Frank 2013), this policy was developed in the late 1970’s. There have been many attempts to revolutionise (MSBOS) since:

> “many new procedures have since been introduced, and surgical techniques have evolved such that blood loss is now less common”.

For example, the work presented by Steven M.Frank and his colleagues sought to propose a new categorisation of surgical procedures, it also introduced an algorithm for blood ordering that can be tailored to fit specific needs for different institutions. They managed to:

> “identify specific low–blood-loss procedures, for which we could eliminate blood orders, and thereby substantially reduce costs”.

Many algorithms were also developed to address the collection stage in terms of collection methodology, which determines whether to keep collected unite as a whole or separate it into different products. A discard rate (Osorio 2016) is included before such a decision is made to account for invalid blood samples.

Another aspect of blood optimisation modelling is the role of historical data; such data can help in testing models before implementation, shaping formulation, and performance comparison. At the collection stage, such data would help in deciding the proportion of required blood donations from different blood types, and the most blood products in demand (Prastacos 1984).

Analytical models at the collection stage sought to solve the configuration of collection points, policies and methods; focusing on measuring the impact of different factors on the performance of a blood bank during the collection stage, such as staff allocation and donor scheduling (Osorio 2016) . Ghandforoush and Sen (Ghandforoush 2010) used a nonlinear integer algorithm to support daily planning of platelet production, Alfonso and Xie (Alfonso 2013) presented a mathematical model for collection planning to minimise products obtained from external suppliers.

The next stage would be about the handling of the collected blood units, although sometimes specific collection methodologies enable the collection of platelets directly from the donor. Models at this stage are focused either on one type of product or multiple products; with the aim to reduce cost, optimise storage capacity, and minimise shortages and wasted unites. Haijema and van Dijk (van 2009) developed a Markovian model to support decision making on production and inventories of platelets. Multiple periods, special periods such as weekends and different types of demand are included in the model. Dynamic programming and local search algorithms were used as solution methods, depending on the problem size.

Blood assignment from inventory would potentially be the last stage of this supply chain unless scheduled deliveries from a regional blood bank to affiliated local hospitals is required. Blood assignment problems BAP are solved using mathematical models. De Angelis and Ricciardi (Angelis 2001) modelled the Blood assignment problem as a multi-product, multi-period, multi-objective linear programming model. They used this method to minimise the quantity of blood imported from outside the system and stabilise the quantities assigned daily. This model addressed only one of the blood groups (A+-, B+-, AB+-, O+-) at a time; they solved the problem of optimal scheduling according to urgency and availability without influencing the demand and supply flows. This was done by introducing three degrees of urgency: very urgent requests that must be satisfied immediately, urgent requests that can be satisfied by the day after, and low-urgency requests that can be met within eight days. It has been applied to the Italian Red Cross (CRI) blood donation-transfusion system in Rome and showed satisfying results.

A similar model was used by Adewumi and Budlender (Adewumi 2012) to optimise the blood assignment problems, by dynamically determining the assignment of blood with the use of multiple Knapsack algorithms and compare its performance with simpler assignment models. The idea of cross-matching between blood types proved to achieve better results with the MKA algorithm by reducing shortages in comparison with Simple Assignment algorithms.

An integer programming model was proposed by Sapountzis (Sapountzis 1984) considered the allocation of blood units from a regional Blood Transfusion Service BTS to the hospitals in that area; the studied case was Glasgow and West of Scotland BTS which serves around 68 hospitals in the West of Scotland. The blood shelf life in Scotland is about 28 days, which means blood units that are not used by that date are said to be expired and should be returned to the Blood bank. However, in Glasgow and West Scotland BTS, it is not permitted to receive the expired units from hospitals since it would lower the safety standards. Therefore, to reduce the wastage of the blood without increasing the workload of the BTS, it is important to allocate blood units to the hospitals based on their activity. The BTS transfuse required blood units based on either routine orders or urgent, fresh blood requests (usually for paediatrics or cardiac surgery) to fulfil each hospital's demands, the main objective of this model was to minimise the expected number of expired units at each hospital. By comparing the mathematical programming model with the manual system (that does not take into account the characteristics of each hospital), a reduction percentage of 6.1% and 3.6% respectively for the number of expired units was achieved.

Olusanya (Olusanya 2015) proposed an efficient new method by combining different techniques, the queue and particle swarm optimisation (PSO) with multiple knapsack problem to address the challenges accompanying the BAP. The PSO, multiple knapsack assignment method which is used to handle the cross-matching of blood types to satisfy the requested units and stabilise the stored blood types in the bank, whereas queuing technique was used to monitor the expiration date of each blood type. The objective was to optimise the assignment of blood types and minimise wasted units and importation from external sources which could be very expensive. By using the above techniques, the total number of blood units imported from external resources were significantly minimised with no wastage.

3.1 Study Scope

One could argue that all stages of the blood supply chain are equally important, but a counter-argument could shift the focus towards inventory management since it plays a central role at this chain.

As has been discussed earlier, blood collection and inventory could take place either at the hospital or at the regional blood centre. The hospital blood bank operates as an inventory location, to meet the local demand; it will handle storage and issuing transfusion requests for unique blood types for a random number of units (Prastacos 1984). Once a request is received, a matching number of units are identified and removed from free inventory and placed on reserve inventory until they are either used or returned. Unutilised units during their shelf life will be invalid and discarded.

Blood management in regional blood banks is considered more challenging because of the large number of participants in the system. We can distinguish between two types of regional systems: centralised and decentralised. In a centralised inventory management system, the regional blood bank supervises the collection, storage, and assignment blood orders to hospitals. This system is based on a scheduled ordering system, while the decentralised system process daily orders from each hospital blood bank which sets its inventory levels. Studies proofed that centralised systems outperform decentralised ones in the number of perished inventory units and shortages, two important criteria for the performance of a blood bank, as well as reducing the average per unit (Prastacos 1984).

Aside from perishability and the disparity between supply and demand, modelling blood inventory performance is also characterised by other features such as the utilisation of blood resources, and age of blood when transfused to patients (Prastacos 1984).

Optimisation is used to support decisions concerning the required number of donors (either in total or broken down by blood group) and the associated collection and production methods, to minimise the production cost. During the production stage, once the collection process is over, the units are placed in quarantine to assure their validity and then added to the inventory. Outdated units are removed daily, and the inventory levels are revised and ready to process and issue new orders, which are divided as either regular or urgent.

In each inventory management model, several assumptions are to be made. Osorio and Brailsford (Osorio 2016) proposed an integrated simulation-optimisation model to support strategic and operational decision making. They used an integer linear optimisation model to run over a planning horizon; it was used to support daily decision such as target number of donors, collection methods, and production planning. A discrete event simulation (DES) were then used to represent flows through the supply chain. In this way, the model can be used in two modes: strategic level to help blood bank managers to evaluate unique resources, allocation policies, and at operational level to set daily collection and production targets. The model output of both cases is considered as standard key performance indicators: stockouts, outdoes, several donors, and production costs.

The ILP model is supposed to calculate the optimal required number of donors by running over a 7-day planning horizon. At the same time, the DES would account for the uncertainty in supply and demand, based on probability distribution fitted from historical data routinely collected in all blood centres. The system state is updated at the begging of each day where the ILP algorithm will run the calculations for the next seven days. The DES will handle simulations of the centre's operations on that day, distributing blood to several demand points. The proposed method can be classified as an "alternative Optimisation-based Simulation (IOS)". The ILP model optimised a cost function composed of production costs, penalties for expired unites, number of stockouts and violation of the blood groups proportionality constraints (Osorio 2016).

Other studies focused on addressing inventory in regional blood banks, focused on optimising the number of daily collections. But it also clarified that determining the optimal size of total units to be collected (overall eight blood types) to meet demand in time is a complex process. Therefore, an exact solution to this problem is impractical to obtain (Prastacos 1984).

In this paper, we developed a linear programming model of inventory management at a regional blood bank. The model explained within the following section is focused on offering a rather simplified linear model to meet daily demand and supply.

4. Problem formulation and discussion

The model is developed for a regional blood bank that would receive scheduled demand from affiliated hospitals. Using linear programming formulation to optimize the number of collected units to meet the daily demand. The model assumes the collection and demand are for whole blood units, for all eight types of blood; scheduled demand also is assumed to be of equal urgency to all hospitals and across all blood types. The constraints compel the model to keep the collection rate adequate to prevent the inventory from falling below a certain minimum level while taking into consideration storage capacity and compelling the collection not exceed this capacity. The objective function is designed to minimize the overall costs and wastage.

Other assumptions were also decided upon to help in guiding the formulation. For example, the initial inventory is set to be null. FIFO policy is assumed to be implemented to keep the inventory fresh, the shelf life of whole blood units is set to expire after 35 days (approximately five weeks) which is the assumed time horizon of the model. We also considered the unit cost includes the cost of collection, handling and storage. As it has been indicted before, not all collected samples are registered in the inventory; some samples are deemed invalid after collection. We assume that only 90% of collected units are to be included in the inventory. The feed to the blood bank is assumed to be available at certain quantities available for each blood type, from regular donors. The time horizon will span over five weeks; the collection is made daily based on scheduled weekly demand, collection proportion will vary based on demand for each blood type, week by week. We assume the weekly demand is about 700 blood units in total for all blood types, based on information received from the regional blood bank in Abu Dhabi.

Variables are denoted by indices that help in tracking the time when the collected unit is registered to the inventory; the first index (i) is used to denote the blood type, second index (t) is used to assign the week during the planning horizon period, and the third index (b) is used to indicate the day of the week. We also consider all variables to be of a positive value.

For each index, the range will be as the following :

$i \in \{1, 2, 3..8\}$ since there are eight types of blood

$t \in \{1, 2, 3..5\}$ since the planning horizon is five weeks

$d \in \{1, 2, 3..7\}$ since there are 7 days in each week

4.1 Parameters

- $G_{i,t,b}$: available donors for blood type i during week t at day b.
- $D_{i,t,b}$:demand for blood type i during week t at day b.
- $H_{i,t,b}$: daily collected units for blood type i available during week t at day b for hospital w.
- M_i : minimum inventory level of available whole blood units for blood type i.
- $F_{i,t,b}$: maximum capacity level of available whole blood units for blood type i during week t at day b.
- L : cost of collected whole blood unit, fixed for all blood types across all days of the week.
- Q : penalty cost for expired units.

4.2 The decision making variables

- $C_{i,t,b}$: inventory level for blood type i during week t at day b.
- $Y_{i,t,b}$: expired units of blood type i during week t at day b.
- $X_{i,t,b}$: collected units for blood type i during week t at day b.

4.3 The constraints

1. Resources constraint states that the number of daily collected units for any blood type shouldn't exceed the number of available units from regular donors:

$X_{i,t,b} \leq G_{i,t,b}$

2. Demand constraint states that the demand shouldn't exceed available resources, i.e capacity:

$D_{i,t,b} \leq G_{i,t,b}$

3. Inventory equilibrium constraint define how the daily inventory is updated everyday. It's set to equal collected units in the same day and any left units from previous weeks minus the demand and the identified expired units :

$C_{i,t,b} = 0.9 \times X_{i,t,b} + C_{i,t-1,b} - D_{i,t,b} - Y_{i,t,b}$

4. Minimum inventory constraint is defined to prevent shortages in inventory by always keeping the inventory above a minimum level:

$C_{i,t,b} \leq M_{i,t,b}$

5. Constraint of valid inventory is defined to identify expired units daily. The expired units at any point of time is a unit in inventory that was collected at the week $t-5$ during the first day of that week :

$Y_{i,t,b} = C_{i,t-5,1}$

6. Inventory capacity constraint at any point is defined as the inventory level plus collected units and it should be less than the maximum capacity:

$F_{i,t,b} = C_{i,t,b} + X_{i,t,b} \leq G_{i,t,b}$

7. Perished units should be less than 1% of all the collected units:

$Y_{i,t,b} \leq 0.01 \times X_{i,t,b}$

4.4 The objective function

The objective function is designed to minimize cost, wasted units, and to keep inventory level from exceeding demand:

$Z = Q \sum_i^8 \sum_t^5 \sum_b^7 Y_{i,t,b} + L \sum_i^8 \sum_t^5 \sum_b^7 X_{i,t,b} + \sum_i^8 \sum_t^5 \sum_b^7 C_{i,t,b}$

5. Discussion on existing data:

To ensure realistic results, we sourced parameters values from the regional blood bank in Abu Dhabi. Costs incurred for collection, handling, and storage of units are estimated to be around 800 AED for a 450 ml blood packet. A weekly demand of 700 blood units that include all blood types is regularly provided to hospitals affiliated with the regional blood bank. A minimum inventory of 100 unit of each blood type is kept for emergencies. The most recent data indicates that a total demand of 708 blood units with variant shares for each blood type: 220 blood units for O-, 53 for O+, 98 for A-, 98 for A+, 88 for B-, 88 for B+, 53 for AB+, and 10 for AB-.

The blood type O- is a universal donor; therefore, extra quantities are always kept by blood banks for that specific blood type since it can cover shortages for other blood types.

The data acquired from Abu Dhabi regional blood bank also stated the minimum inventory levels for each blood type as the following: 6 blood units for O+, 11 for A- and A+ each, 10 for B- and B+ each, 1 for AB- since there are fewer people with AB- blood type, and 6 for AB+.We considered the feed to the blood bank is stable from regular donors.

We considered the feed to the blood bank is stable from regular donors. According to the blood bank, the frequency of people donating blood is different for each day in the week, with Sundays registering significantly higher donation levels. Wednesdays are the lowest in donations while stable donation levels are registered for Mondays, Tuesdays and Thursdays. To quantify that, we have set some percentages for each day, which the blood bank approved, and these are: 33% of blood units are collected on Sunday, 10% on Monday, Tuesday, and Thursday, 30% on Wednesday, 5% on Friday, and 2% on Saturday.

The main purpose of this model is to optimise the number of daily collected units to meet the scheduled demand and avoid waste. As per the blood bank in Abu Dhabi, keeping a waste level of 1% was among their KPIs.

Code in Gurobi (Anaconda)

For our project, and as part of the course requirement, we have used Gurobi to optimize our problem. This being said, and since we are dealing with 8 excel sheets (1 per blood type), we had to call the following libraries:

```python
from __future__ import print_function
import pandas as pd
from array import array
from pandas import ExcelWriter
from pandas import ExcelFile
from ortools.linear_solver import pywraplp
import gurobipy as grb
from gurobipy import *
import numpy as np
```

Figure 1: Libraries used in the code

After that, we have introduced our model as bldInv (Blood Inventory), along with the minimum inventory for each blood type before introducing the supply as a dictionary that reads from every excel file.

```python
bldInv= grb.Model()

T= ['W1','W2','W3','W4','W5'] #Time in weeks
B=['D1','D2','D3','D4','D5','D6','D7'] #Time in days
BT= ['O-','O+','A-','A+','B-','B+','AB+','AB-'] #Blood Types
T=range(1 , 36)

Min_inv= {'O-': 25, 'O+': 6, 'A-': 11, 'A+': 11, 'B-': 10, 'B+': 10, 'AB-': 1, 'AB+': 6}

supply= {}
for i in BT:
    for t in T:
        if i=='O-':
            ON= pd.read_excel('Desktop/ONegative.xlsx')
            supply[i, t]= ON.Supply[t-1]
        elif i=='O+':
            OP= pd.read_excel('Desktop/OPositive.xlsx')
            supply[i, t]= OP.Supply[t-1]
        elif i=='A+':
            AP= pd.read_excel('Desktop/APositive.xlsx')
            supply[i, t]= AP.Supply[t-1]
        elif i=='A-':
            AN= pd.read_excel('Desktop/ANegative.xlsx')
            supply[i, t]= AN.Supply[t-1]
        elif i=='B+':
            BP= pd.read_excel('Desktop/BPositive.xlsx')
            supply[i, t]= BP.Supply[t-1]
        elif i=='B-':
            BN= pd.read_excel('Desktop/BNegative.xlsx')
            supply[i, t]= BN.Supply[t-1]
        elif i=='AB+':
            ABP= pd.read_excel('Desktop/ABPositive.xlsx')
            supply[i, t]= ABP.Supply[t-1]
        elif i=='AB-':
            ABN= pd.read_excel('Desktop/ABNegative.xlsx')
            supply[i, t]= ABN.Supply[t-1]
```

Figure 2: Model and initializing the data

Then, we declared our decision variables as stated in the formulation part, and set our initial inventory to 0 for all blood types, then defined our objective function.

```python
#Declaring Decision Variables
Exp_units = bldInv.addVars(BT, T , obj=800, name='Expired_Units')
Coll_units= bldInv.addVars(BT, T, obj=800,name='Collected_Units')
Inv= bldInv.addVars(BT, T, obj=800, name='Units_in_Inventory')
Inv['O-',0]=0
Inv['O+',0]=0
Inv['A-',0]=0
Inv['A+',0]=0
Inv['B-',0]=0
Inv['B+',0]=0
Inv['AB-',0]=0
Inv['AB+',0]=0
#Objective Function
bldInv.modelSense = grb.GRB.MINIMIZE
```

Figure 3: Decision variables and objective function declaration

After the objective function, all the constraints were implemented as shown below:

```python
#Constraints

#Inventory Equilibruim Constraint
for i in BT:
    for t in T:
        inv_constr = bldInv.addConstr(0.9*Coll_units[i,t] + Inv[i,t-1] - supply[i,t] - Exp_units[i,t]
                                      == Inv[i,t], name='Inventory Equilibrium'+i+str(t))

#Inventory Minimum Constraint
for i in BT:
    for t in T:
        Max_Inv_constrt= bldInv.addConstr(grb.quicksum(Inv[i,t] for i in BT) <= 800,
                                          name='Maximum Capacity Constraint'+i+str(t))

#Demand Constraint
for i in BT:
    for t in T:
        Min_Inv_cnstrt= bldInv.addConstr(Inv[i,t] >=  Min_inv[i], name='Minimum Inventory Level'+i+str(t))

#Capacity Constraint
for t in T:
    Capacity_cnstrt= bldInv.addConstr((grb.quicksum(Inv[i,t] +  Coll_units[i,t] for i in BT) <=  800),
                                      name='Maximum Capacity'+i+str(t))

#Expired Units Constraint
for i in BT:
    for t in T:
        Exp_units_constrt= bldInv.addConstr(Exp_units[i,t] - 0.01 * Coll_units[i,t] <= 0,
                                            name='Perishability Minimum')
```

Figure 4: Constraints of the problem

Afterwards, we called the function, optimized the model, and made sure the results were exported to the corresponding excel files.

```python
bldInv.write('Blood Inventory Model.lp')
f = open('Blood Inventory Model.lp', 'r')
print (f.read())
f.close()
bldInv.optimize()
```

```python
ON['Collected_O-']=[Coll_units['O-',t].X for t in T]
OP['Collected_O+']=[Coll_units['O+',t].X for t in T]
AN['Collected_A-']=[Coll_units['A-',t].X for t in T]
AP['Collected_A+']=[Coll_units['A+',t].X for t in T]
BN['Collected_B-']=[Coll_units['B-',t].X for t in T]
BP['Collected_B+']=[Coll_units['B+',t].X for t in T]
ABN['Collected_AB-']=[Coll_units['AB-',t].X for t in T]
ABP['Collected_AB+']=[Coll_units['AB+',t].X for t in T]
```

```python
ON.to_excel('Desktop/ONegative.xlsx')
OP.to_excel('Desktop/OPositive.xlsx')
AN.to_excel('Desktop/ANegative.xlsx')
AP.to_excel('Desktop/APositive.xlsx')
BN.to_excel('Desktop/BNegative.xlsx')
BP.to_excel('Desktop/BPositive.xlsx')
ABN.to_excel('Desktop/ABNegative.xlsx')
ABP.to_excel('Desktop/ABPositive.xlsx')
```

Figure 5: Optimization of the model and data export to Excel

6. Results

After running the code, we got the following result:

```
Optimize a model with 1155 rows, 840 columns and 4752 nonzeros

Coefficient statistics:
  Matrix range     [1e-02, 1e+00]
  Objective range  [8e+02, 8e+02]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 8e+02]
Presolve removed 814 rows and 560 columns
Presolve time: 0.05s
Presolved: 341 rows, 280 columns, 1368 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    5.4577744e+06   3.555556e-04   0.000000e+00      0s
       1    5.4577778e+06   0.000000e+00   0.000000e+00      0s

Solved in 1 iterations and 0.06 seconds
Optimal objective  5.457777778e+06
```

Figure 6: Code's result

The results on the excel files are as shown below:

	Supply	Week_Nbr	Day_Nbr	Collected_O-
Day1	73	W1	D1	109
Day 2	22	W1	D2	24
Day3	22	W1	D3	24
Day4	66	W1	D4	73
Day5	22	W1	D5	24
Day6	11	W1	D6	12
Day7	4	W1	D7	4
Day8	73	W2	D1	81

Figure 7: Collected units output for O-

	Supply	Week_Nbr	Day_Nbr	Collected_O+
Day1	18	W1	D1	27
Day 2	5	W1	D2	6
Day3	5	W1	D3	6
Day4	16	W1	D4	18
Day5	5	W1	D5	6
Day6	3	W1	D6	3
Day7	1	W1	D7	1
Day8	18	W2	D1	20

Figure 8: Collected units output for O+

	Supply	Week_Nbr	Day_Nbr	Collected_A-
Day1	32	W1	D1	48
Day 2	10	W1	D2	11
Day3	10	W1	D3	11
Day4	29	W1	D4	32
Day5	10	W1	D5	11
Day6	4	W1	D6	4
Day7	3	W1	D7	3
Day8	32	W2	D1	36

Figure 9: Collected units output for A-

	Supply	Week_Nbr	Day_Nbr	Collected_A+
Day1	32	W1	D1	48
Day 2	10	W1	D2	11
Day3	10	W1	D3	11
Day4	29	W1	D4	32
Day5	10	W1	D5	11
Day6	4	W1	D6	4
Day7	3	W1	D7	3
Day8	32	W2	D1	36

Figure 10: Collected units output for A+

	Supply	Week_Nbr	Day_Nbr	:ollected_B-
Day1	29	W1	D1	43
Day 2	9	W1	D2	10
Day3	9	W1	D3	10
Day4	26	W1	D4	29
Day5	9	W1	D5	10
Day6	4	W1	D6	4
Day7	2	W1	D7	2
Day8	29	W2	D1	32

Figure 11: Collected units output for B-

	Supply	Week_Nbr	Day_Nbr	:ollected_B+
Day1	29	W1	D1	43
Day 2	9	W1	D2	10
Day3	9	W1	D3	10
Day4	26	W1	D4	29
Day5	9	W1	D5	10
Day6	4	W1	D6	4
Day7	2	W1	D7	2
Day8	29	W2	D1	32

Figure 12: Collected units output for B+

	Supply	Week_Nbr	Day_Nbr	ollected_AB
Day1	18	W1	D1	27
Day 2	5	W1	D2	6
Day3	5	W1	D3	6
Day4	16	W1	D4	18
Day5	5	W1	D5	6
Day6	3	W1	D6	3
Day7	1	W1	D7	1
Day8	18	W2	D1	20

Figure 13: Collected units output for AB+

	Supply	Week_Nbr	Day_Nbr	ollected_AB
Day1	4	W1	D1	6
Day 2	1	W1	D2	1
Day3	1	W1	D3	1
Day4	3	W1	D4	3
Day5	1	W1	D5	1
Day6	0	W1	D6	0
Day7	0	W1	D7	0
Day8	4	W2	D1	4

Figure 14: Collected units output for AB-

We have taken a screenshot of the first 8 days since we assumed that we are starting with no inventory and we should satisfy the inventory equilibrium constraint. Hence, collected units on the first day of operations should be the highest. Eventually, the same numbers will be repeating themselves all over the remaining 27 days while the inventory was kept at exactly its minimum for all blood types.

The results seem pretty logical, and when we have checked them with the blood bank from which we got the initial data, they have confirmed that they collected a little more each day for each blood type, which is why they have a waste of about 3% on all blood types. Although our model doesn't thoroughly deal with perished units, it includes some measures directly related to it; that can be seen via the perishability constraint, and inventory equilibrium.

Sensitivity analysis

Sensitivity analysis is an important measure to see how variables react to certain changes in the model. We have used Gurobi to have that report as well, and the code was as follows:

```
print('Sensitivity Analysis (SA)\nObjVal', bldInv.ObjVal)
bldInv.printAttr(['X', 'Obj','SAObjLow','SAObjUp'])
bldInv.printAttr(['X', 'RC','LB','SALBLow','SALBUp','UB','SAUBLow','SAUBUp'])
bldInv.printAttr(['Sense', 'Slack','Pi','RHS','SARHSLow','SARHSUp'])
```

Figure 15: Sensitivity Analysis code segment

After running this segment of code, the second line of code resulted in the following:

Sensitivity Analysis (SA)

ObjVal 5457777.7777777845

Variable	X	Obj	SAObjLow	SAObjUp
Collected_Units[O-,1]	108.889	800	80	1e+100
Collected_Units[O-,2]	24.4444	800	80	1520
Collected_Units[O-,3]	24.4444	800	80	1520
Collected_Units[O-,4]	73.3333	800	80	1520
Collected_Units[O-,5]	24.4444	800	80	1520
Collected_Units[O-,6]	12.2222	800	80	1520
Collected_Units[O-,7]	4.44444	800	80	1520
Collected_Units[O-,35]	4.44444	800	-720	1520
Units_in_Inventory[O-,1]	25	800	0	1e+100
Units_in_Inventory[O-,2]	25	800	0	1e+100
Units_in_Inventory[O-,3]	25	800	0	1e+100
Units_in_Inventory[O-,4]	25	800	0	1e+100
Units_in_Inventory[O-,5]	25	800	0	1e+100
Units_in_Inventory[O-,6]	25	800	0	1e+100
Units_in_Inventory[O-,7]	25	800	0	1e+100
Units_in_Inventory[O-,35]	25	800	-888.889	1e+100

Figure 16: Sensitivity analysis results part 1

The results were similar to all variables regardless of the blood type, as the same number were repeated until day 34. If we attempt to interpret the results, we could see that the coefficient of the collected units could decrease by 720 or increase by the same, and that wouldn't impact our optimal solution. On the other hand, the coefficient of the units in inventory could decrease to 0 or increase to infinity and that wouldn't change our optimal solution. That is logical because we are trying to maintain a fixed inventory for all blood types. Regarding the very first line of the sensitivity report, having an upper limit of infinity makes sense because we are already collecting in excess vis-à-vis other days. Yet, the allowable decrease on the 35th day is big, and that could be interpreted in the sense that we already have enough inventory to satisfy that day's need, therefore if we decrease or increase its coefficient by 720 up and down, nothing will change when it comes to the optimal solution; and the same thing applies to the inventory.

The second line of code gave the following output:

Variable	X	RC	LB	SALBLow	SALBUp	UB	SAUBLow	SAUBUp	
Collected_Units[O-,1]	108.889	0	0	-1e+100	108.889	1e+100	108.889	1e+100	
Collected_Units[O-,2]	24.4444	0	0	-1e+100	24.4444	1e+100	24.4444	1e+100	
Collected_Units[O-,3]	24.4444	0	0	-1e+100	24.4444	1e+100	24.4444	1e+100	
Collected_Units[O-,4]	73.3333	0	0	-1e+100	73.3333	1e+100	73.3333	1e+100	
Collected_Units[O-,5]	24.4444	0	0	-1e+100	24.4444	1e+100	24.4444	1e+100	
Collected_Units[O-,6]	12.2222	0	0	-1e+100	12.2222	1e+100	12.2222	1e+100	
Collected_Units[O-,7]	4.44444	0	0	-1e+100	4.44444	1e+100	4.44444	1e+100	
Units_in_Inventory[O-,1]	25	0	0	-1e+100	25	1e+100	25	1e+100	
Units_in_Inventory[O-,2]	25	0	0	-1e+100	25	1e+100	25	1e+100	
Units_in_Inventory[O-,3]	25	0	0	-1e+100	25	1e+100	25	1e+100	
Units_in_Inventory[O-,4]	25	0	0	-1e+100	25	1e+100	25	1e+100	
Units_in_Inventory[O-,5]	25	0	0	-1e+100	25	1e+100	25	1e+100	
Units_in_Inventory[O-,6]	25	0	0	-1e+100	25	1e+100	25	1e+100	
Units_in_Inventory[O-,7]	25	0	0	-1e+100	25	1e+100	25	1e+100	

Figure 17: Sensitivity analysis results part 2

The reduced cost, as the amount by which the value of the objective function will decrease if we increase a variable by 1unit, is 0 for all variables according to the figure above. That is logical since the reduced cost is always 0 if the optimal value is positive, and that is the case in our project. The lower bound and upper bound are set to infinity for there were no restrictions on that initially in the formulation. Other than that, the largest and lowest bound values at which our optimal solution would remain optimal are equal to our optimal values.

The output of the third line of the code was the below:

Constraint	Sense	Slack	Pi	RHS	SARHSLow	SARHSUp
Inventory EquilibriumO-1	=	0	888.889	73	-25	406
Inventory EquilibriumO-2	=	0	888.889	22	3.55271e-15	599
Inventory EquilibriumO-3	=	0	888.889	22	3.55271e-15	599
Inventory EquilibriumO-4	=	0	888.889	66	1.42109e-14	503
Inventory EquilibriumO-5	=	0	888.889	22	3.55271e-15	599
Inventory EquilibriumO-6	=	0	888.889	11	1.77636e-15	626
Inventory EquilibriumO-7	=	0	888.889	4	0	636
Maximum Capacity ConstraintO-1	<	720	0	800	80	1e+100
Maximum Capacity ConstraintO-2	<	720	0	800	80	1e+100
Maximum Capacity ConstraintO-3	<	720	0	800	80	1e+100
Maximum Capacity ConstraintO-4	<	720	0	800	80	1e+100
Maximum Capacity ConstraintO-5	<	720	0	800	80	1e+100
Maximum Capacity ConstraintO-6	<	720	0	800	80	1e+100
Maximum Capacity ConstraintO-7	<	720	0	800	80	1e+100
Minimum Inventory LevelO-1	>	0	800	25	-0	47
Minimum Inventory LevelO-2	>	0	800	25	3	47
Minimum Inventory LevelO-3	>	0	800	25	3	91
Minimum Inventory LevelO-4	>	0	800	25	-0	47
Minimum Inventory LevelO-5	>	0	800	25	3	36
Minimum Inventory LevelO-6	>	0	800	25	14	29
Minimum Inventory LevelO-7	>	0	800	25	21	98
Maximum CapacityAB-1	<	370	0	800	430	1e+100
Maximum CapacityAB-2	<	641.111	0	800	158.889	1e+100
Maximum CapacityAB-3	<	641.111	0	800	158.889	1e+100
Maximum CapacityAB-4	<	485.556	0	800	314.444	1e+100
Maximum CapacityAB-5	<	641.111	0	800	158.889	1e+100
Maximum CapacityAB-6	<	683.333	0	800	116.667	1e+100
Maximum CapacityAB-7	<	702.222	0	800	97.7778	1e+100

Figure 18: Sensitivity analysis results part 3

From the above, it appears that both Maximum capacity constraints have shadow prices (Pi) of 0; therefore, they are considered nonbinding constraints. Likewise, the inventory equilibrium and minimum inventory level constraints have non-zero shadow prices and 0 slack; and are therefore considered binding constraints. Actually, the slack value of 0 for the inventory equilibrium is evident since it's an equality.

7. Conclusion and Future work

This model proved to be able to solve the inventory management problem for a perishable product despite its simplistic approach. The algorithm could be further expanded to include other types of blood products.The sensitivity analysis showed which parameters hold the largest effect on the blood bank performance and what are the intervals in which the blood bank could play to keep its collection plan optimal. This project could further be improved by implementing an inventory update that would help to track the inventory and regularly check whether the collected units were consumed totally or not. This leads to another constraint in which the perished units are kept to a minimum so that the blood bank meets its KPIs. A notion towards integer programming model would prove to be also a possible future enhancement for the current model.

Acknowledgement :

We would like to deeply thank Dr. Andrei for his support throughout this project, from its initial formulation to the implementation of the code and its results. Warm thanks go to Abu Dhabi blood bank as well, mainly Dr. Naima Oumeziane for her stimulating collaboration, and her tireless answers to our endless questions and concerns regarding data and results.

References

1.Adewumi, Aderemi, et al. "Optimizing the Assignment of Blood in a Blood Banking System: Some Initial Results." *2012 IEEE Congress on Evolutionary Computation*, 2012, doi:10.1109/cec.2012.6256633.

2.Angelis, V. De, et al. "Optimizing Blood Assignment in a Donation-Transfusion System." *International Transactions in Operational Research*, vol. 8, no. 2, 2001, pp. 183–192., doi:10.1111/1475-3995.00258.

3.Bellabdaoui, A., et al. "Optimisation Des Niveaux De Stock Dans Les Centres De Transfusion Sanguine Au Maroc." *Transfusion Clinique Et Biologique*, vol. 21, no. 1, 2014, pp. 15–22., doi:10.1016/j.tracli.2013.10.004.

4.Dijk, Nico Van, et al. "Blood Platelet Production: a Novel Approach for Practical Optimization." *Transfusion*, vol. 49, no. 3, 2009, pp. 411–420., doi:10.1111/j.1537-2995.2008.01996.x.

5.Ensafian, Hamidreza, and Saeed Yaghoubi. "Robust Optimization Model for Integrated Procurement, Production and Distribution in Platelet Supply Chain." *Transportation Research Part E: Logistics and Transportation Review*, vol. 103, 2017, pp. 32–55., doi:10.1016/j.tre.2017.04.005.

6.Frank, Steven M., et al. "Optimizing Preoperative Blood Ordering with Data Acquired from an Anesthesia Information Management System." *Anesthesiology*, vol. 118, no. 6, 2013, pp. 1286–1297., doi:10.1097/aln.0b013e3182923da0.

7.Jefferies, Leigh, et al. "A Team Approach To Decrease Wasted Blood Products." *Laboratory Medicine*, vol. 27, no. 12, Jan. 1996, doi:10.1093/labmed/27.12.833.

8.Pereira, A. "Blood Inventory Management in the Type and Screen Era." *Vox Sanguinis*, vol. 89, no. 4, 2005, pp. 245–250., doi:10.1111/j.1423-0410.2005.00700.x.

9.Prastacos, Gregory P. "Blood Inventory Management: An Overview of Theory and Practice." *Management Science*, vol. 30, no. 7, 1984, pp. 777–800., doi:10.1287/mnsc.30.7.777.

10.Sapountzis, Costas. "Allocating Blood to Hospitals from a Central Blood Bank." *European Journal of Operational Research*, vol. 16, no. 2, 1984, pp. 157–162., doi:10.1016/0377-2217(84)90070-5.

11.Osorio, Andres F., et al. "Simulation-Optimization Model for Production Planning in the Blood Supply Chain." *Health Care Management Science*, vol. 20, no. 4, Apr. 2016, pp. 548–564., doi:10.1007/s10729-016-9370-6.