Vendor Managed Inventory Problem for Blood Platelets: a Two Stage Deep Reinforcement Learning Approach

Andres F. Osorio[a], Juan D. Carvajal[b]

[a] *afosorio@icesi.edu.co*
[b] *juan030698@hotmail.com*

Abstract

This paper presents an application of Deep Reinforcement techniques

1. Introduction

2. Literature Review

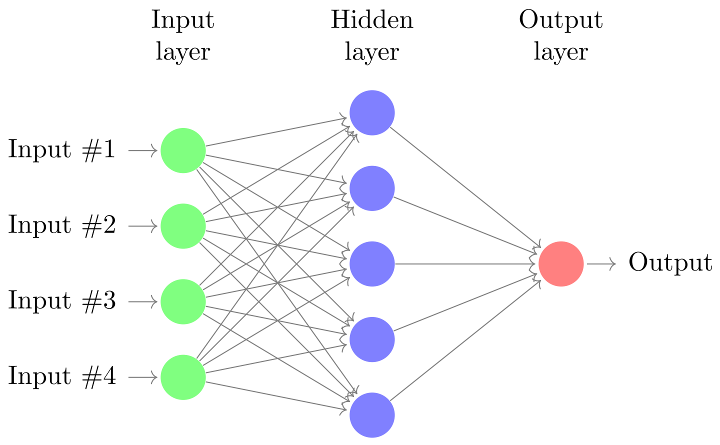
3. Problem Definition

3.1. Context

In general terms, the supply chain of human blood products its very complicated. Blood products, specially platelets, have very short expiration periods (around 5 days). Because of this, blood managing entities , like donor centers, face great issues estimating and planning production of these vital products. These issues can be of two types, the first one, its about expiration levels, if the donor center produces too much platelets (overestimating the demand), these limited and costly to produce items will go to waste, generating high costs for the entity, and providing no benefit to final users.The second kind of issue, is related to product stock outs (underestimating the demand).Its clear that stock outs are a much more important problem for the donor centers, because these can result in increased risk of death for those needing the platelets.The gravity of this problems is only increased when multiple hospitals or medical centers, with different demand distributions are dependent on the supplies from the donor center.The basic decision for a donor center consists of two parts, the first one, its about how much to produce. Human blood can be transformed and separated into its basic constituents (platelets among these)using a process called blood fractionation,usually performed centrifuging the blood.This process is carried out using specially designed blood bags,which can be single, double, triple or quadruple.Only the last two (the most expensive ones) can be used to extract platelets, so the donor center must know how many of these bags to use in the total donor pool.The second problem its about how much to deliver to each medical center, given its demand distribution and the current inventory of the donor center. This problem is essentially a particular case of a Vendor Managed Inventory System, where a distribution center has to supply each of its customers based on individual needs and current capacity.

3.2. Modeling

To model such a system,a 4 medical center distribution has been considered, but the model should be easily scaled to an arbitrary number of medical centers.The medical centers will be referred from know on as hospitals. Each hospital has an internal inventory, visible to the donor center, this inventory is represented by a 5-tuple:



4. Methodology

4.1. Vendor Managed Inventory

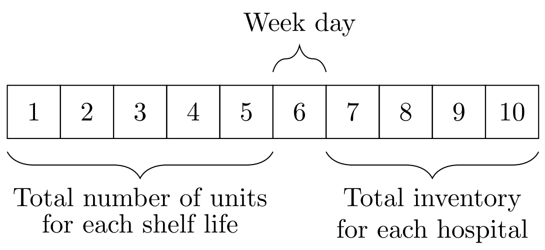
Opposing to the traditional vendor-client systems, Vendor managed inventory systems design the supply chain architecture from the supplier to the customers.Clients must provide relevant information like inventories and sales, and is the responsibility of the vendor to define the size and date of the shipments. These systems often come with a tight information system coupling between clients and vendors, so the information is always readily available. The VMI model is going to be leveraged in order to reduce the general levels of stock-outs and expired product in the blood supply chain, more specifically the platelets supply chain.

4.2. Deep Reinforcement Learning Framework

To leverage the VMI model, the blood distribution centers (the vendors for this specific case), need a way to define optimal or near optimal policies to make shipments to the customers(hospitals and medical facilities in need of platelets).Considering that in real-world systems, demand is stochastic and often seasonal, most classic inventory policies can't be used. Also, for this specific case, the vendor does not have unlimited production capacity, because raw materials for blood products come from a variable pool of donors. Because of this,the best approach to find the shipping policies are heuristics. The chosen method to generate such policies is Deep Reinforcement Learning.This framework is based on a reward system across various simulations of the actual system, that must be represented in a way the deep learning framework can understand.

4.2.1. Data Representation

Typical deep reinforcement systems are modeled as a set of states and a set of actions. A state correspond to a tuple of numbers that represent certain characteristics of the system at a given time. Similarly, the action space of a system can be represented as a tuple, where each element maps directly to a well defined action. Both state and action spaces can be discrete or continuous.For this specific case, both are discrete. The case study system state is represented by a 10-tuple, each position of the tuple represents a characteristic of the system in the following form:


Week day
1
2
3
4
5
6
7
8
9
10
Total number of units for each shelf life
Total inventory for each hospital

Position 1 to 5: Total number of platelets from 0 to 4 remaining days of shelf life. Items with 0 days of shelf life will expire in the current time step. Items with 4 days of remaining shelf life have just been produced.

Position 6: Represent the day of the week from 1 to 7, Monday is the first day of the week.

Position 7 to 10: Represent the sum of the inventory for each of the 4 hospitals connected to this vendor.

Now it's clear that the state space for this system is discrete.

For the action space, the scheme is very similar, the deep reinforcement agent works with discrete actions represented by a tuple.The proposed system has the need to produce two types of actions simultaneously , so we have the need to encode these two values within a single index of the action vector. The first type of action has to do with how many units of platelets are extracted from the collected blood for the next time step. This action can be expressed as a percentage of the total donor pool for the current time step (day).Because these percentage is a continuous quantity, we chose to discretize it into 10% intervals,this means we have 11 values from 0% to 100%. The second kind of action is related to how much product to ship to the hospitals and medical centers (as a total). Considering the demand distribution for all the hospitals, we chose a value of 99 as an absolute maximum for this number, this means we have 100 possible values for this kind of action (from 0 to 99). If we multiply the possibilities of the two types of actions (11 x 100), we get a total of 1100 possible actions for this system. Here we can note that the action space for this system is discrete, because the system can perform one and only one action in a given time step, and the chosen action is well defined as a binary option.This means that a typical decision for this agent is a one-hot encoded vector, where the index of the one is the value for the action. This is important, because other variations of deep reinforcement learning allow the decision to be continuous, this means the agent could choose to execute several actions with values between 0 and 1.The resulting vector can be represented as follows:

1	2	3	4	$\cdots$	1096	1097	1098	1099	1100

To extract both actions based on the index of the vector, these basic operation can be performed:

$$Production\ \% = 10((n-1) \mod 11)$$

$$Total\ Shipment = \lfloor \frac{n-1}{11} \rfloor$$

4.3. Goal Programming Model

$$Q(s_t, a_t) \longleftarrow (1-\alpha).Q(s_t, a_t) + \alpha.(r_t + \gamma. \max Q(s_{t+1}, a)) \tag{1}$$

4.4. Integer Programming Model

Objective Function

$$\min \quad \sum_{h \in H} (I_{h0}CV + F_h CF) \tag{2}$$

Constraints

$$I_{h0} = \max(0, I_{h1} + X_{h1} - D_h) \tag{3}$$

$$F_h = \max(0, D_h - \sum_{r \in R} (I_{hr} + X_{hr})) \tag{4}$$

$$F_h \leq \frac{1}{\|H\|} \sum_{h \in H} (F_h) \tag{5}$$

$$A_r = \sum_{h \in H} (X_{hr}) \tag{6}$$

$$\frac{\sum_{r \in R} (I_{hr} + X_{hr})}{D_h} = \frac{\sum_{r \in R} (I_{h+1r} + X_{h+1r})}{D_{h+1}} \quad \forall h < 4 \tag{7}$$

Constraints 2 and 3 can be linearised as follows:
Linearisation Equation 2

$$-I_{h1} - X_{h1} + D_h \leq M * YI_{h0} \ \ \forall \ \ h \tag{8}$$

$$I_{h1} + X_{h1} - D_h \leq M(1 - YI_{h0}) \ \ \forall \ \ h \tag{9}$$

$$I_{h0} \geq 0 \ \ \forall \ \ h \tag{10}$$

$$I_{h0} \geq I_{h1} + X_{h1} - D_h \ \ \forall \ \ h \tag{11}$$

$$I_{h0} \leq M(1 - YI_{h0}) \ \ \forall \ \ h \tag{12}$$

$$I_{h0} \leq I_{h1} + X_{h1} - D_h + M * YI_{h0} \ \ \forall h \tag{13}$$

Linearisation Equation 3

$$-D_h + \sum_{r \in R}(I_{hr} + X_{hr}) \leq M * YF_h \ \ \forall h \tag{14}$$

$$D_h - \sum_{r \in R}(I_{hr} + X_{hr}) \leq M * (1 - YF_h) \ \ \forall h \tag{15}$$

$$F_h \geq 0 \ \ \forall \ \ h \tag{16}$$

$$F_h \geq D_h - \sum_{r \in R}(I_{hr} + X_{hr}) \ \ \forall \ \ h \tag{17}$$

$$F_h \leq M(1 - YF_h) \ \ \forall \ \ h \tag{18}$$

$$F_h \leq D_h - \sum_{r \in R}(I_{hr} + X_{hr}) + M * YF_h \ \ \forall h \tag{19}$$

Once the model has been run, the inventory in each hospital is updated as follows:

$$I_{hr} = \max(0, I_{hr+1} + X_h - \max(0, D_h - \sum_{r \in R}(I_{hr} + X_{hr}))) \ \ \forall h, \ r \neq 0, 4 \tag{20}$$

$$I_{h4} = \max(0, X_4 - \max(0, D_h - \sum_{r \in R}(I_{hr} + X_{hr}))) \tag{21}$$

5. Case study

6. Results

References