Metaheuristic Implementations For The Blood Assignment Problem

Author: Prinolan Govender, Supervisor: Absalom Ezugwu

School of Mathematics, Statistics, and Computer Science, University of Kwazulu-Natal,
Private Bag Box X54001,
Durban 4000, South Africa

Abstract. The following paper implemented metaheuristic techniques (MT) to optimize the blood assignment problem (BAP) for South African blood banks. The study used Genetic algorithms (GA), Particle Swarm Optimization (PSO), Duel Algorithm (DA), Symbiotic Organism Search (SOS), and the Grey Wolf Optimizer (GWO). The objective function of the BAP tries to find ways of minimizing blood product wastage with regard to expiration and importation of blood products from external sources, whilst trying to maximising product delivery to patients in need. By minimizing the objective functions, the blood banks will incur lower operating costs in order to function and utilise resources more efficiently and effectively.

1. Introduction

Human blood inventory management is characterized by a string of factors which can prove to be complicated over time [1]. Blood products are usually received by voluntary donors, these products are then stored in ideal conditions for usage in the future. Blood is comprised of 4 components namely red blood cells (RBC), white blood cells (WBC), and platelets (PLT) which is immersed in a matrix of plasma. The following study focuses on whole blood (WB) units, which relates to all components of blood. In accordance to the ABO blood system [2], blood has four different blood groups namely A, B, AB and O. Each of these blood types have a Rhesus value (Rh) which can either be positive (+) or negative (-), the introduction of a Rh value therefore doubles the number of blood groups in humans leading to 8 different blood types. These blood groups are significant with regards to storage and distribution as mixing incompatible blood types can lead to blood clumping (also known as agglutination), which can be life threatening for patients [3]. The Blood assignment problem (BAP) is an optimization task, which tries to efficiently assign WB units to patients whilst trying to minimize the amount of importation and expiry within the blood bank. To confront the problem, this study employs metaheuristic (MT) implementations as well as an assignment policy which tries to satisfy the objective function of the BAP.

Demand for WB units can be broken into two main situations. Situation one involves a predetermined event such as a typical surgery which could be schedules days before. The scheduling process will also allow for sufficient WB units to be set aside (or imported if there is an insufficient amount). Situation two relates to demand for WB units that cannot be foreseen, this can be exemplified by people being exposed to sudden onsets of trauma which is in need of immediate attention and/or WB units. Blood is deemed as a perishable commodity due to its limited shelf life, coupled with the complexity of blood compatibility and the stochastic nature of daily blood demand the BAP can therefore be seen as a NP-Hard problem [4].

The following study is broken up into the following chapters. Chapter 2: literature review, Chapter 3: problem description, Chapter 4: methodology, Chapter 5: results and discussion, and Chapter 6: Conclusion.

2. Literature Review

Previous research that is specifically concerned with the BAP is relatively scarce. However, the studies conducted in this field have used different approaches and MTs to try and optimize the assigning of blood. The transfusion of blood and/or blood related products is a daily activity which occurs in most hospitals and clinics around the world. Before understanding blood compatibility, transfusion would often result to blood clumping and other negative effects upon the patient. Dating back to 1927, where the antigens A and B for blood types where discovered, later type O was established resulting in the widely known ABO blood group system [5]. Charpin [6] introduced a linear model for blood management, which incorporated a dynamic environment of blood products entering and exiting the system on a daily basis. The model also takes into account compatibility, and tries to find the best solution by attempting to match the supply to the demand for a day. Many factors contribute towards the assignment of blood to patients, this includes request time, urgency for the request, compatibility of blood types and quantity of blood [7].

The studies conducted by [2], [4] and [7] followed the same backbone in terms of producing an optimization model. The usage of a Multiple Knapsack algorithm was used to cross-match blood between compatible blood types, along with the Bottom-up technique to pull additional units of blood from compatible blood types. [2] Implemented multiple adaptations of a GA approach which included Genetic Algorithm (GA), Adaptive Genetic Algorithm (AGA), Simulated Annealing Genetic Algorithm (SAGA), Adaptive Simulated Annealing Genetic Algorithm (ASAGA) as well as Hill Climbing (HC) Algorithm. Results from the study showed that all implementations successfully achieved optimal fitness, with HC performing the best. The study conducted by [6] implemented 2 local searches namely GRASP and dynamic programming and generated supply for a day by adding the previous days remainder to the donations received in the day. The results showed that GRASP imports O^+ and O^- blood quite heavily within the first 50 days before eventually levelling out, whilst dynamic programming handles the event of demand exceeding supply more efficiently.

The BAP has generated unique potential solutions which take into account external factors that contribute towards the assignment of blood. Sapountiz [8] incorporated probability distribution by considering characteristics such as the management of the hospital, rules and regulations within the blood bank, and organising blood according to doctor’s preference. Table 1 represents the compatible blood types, “YES” indicates that a blood type is compatible, and “NO” implies that a blood type is not compatible.

Table 1

Blood Types	A+	A-	B+	B-	AB+	AB-	O+	O-
A+	YES	YES	NO	NO	NO	NO	YES	YES
A-	NO	YES	NO	NO	NO	NO	NO	YES
B+	NO	NO	YES	YES	NO	NO	YES	YES
B-	NO	NO	NO	YES	NO	NO	NO	YES
AB+	YES	YES	YES	YES	YES	YES	YES	YES
AB-	NO	NO	NO	YES	NO	YES	NO	YES
O+	NO	NO	NO	NO	NO	NO	NO	YES
O-	NO	NO	NO	NO	NO	NO	NO	YES

Table 1: Representation of blood compatibility

2.1. The South African population

South Africa is often called the "rainbow nation" due to its variety of culture and race. Race refers to the ethnicity of an individual, there exists four common races in South Africa, Black, White, Indian, and Coloured, with approximately 41 million citizens currently living in South Africa. The Pie chart labelled as "Figure 1" indicates the current percentage of races within the population.

Figure 1

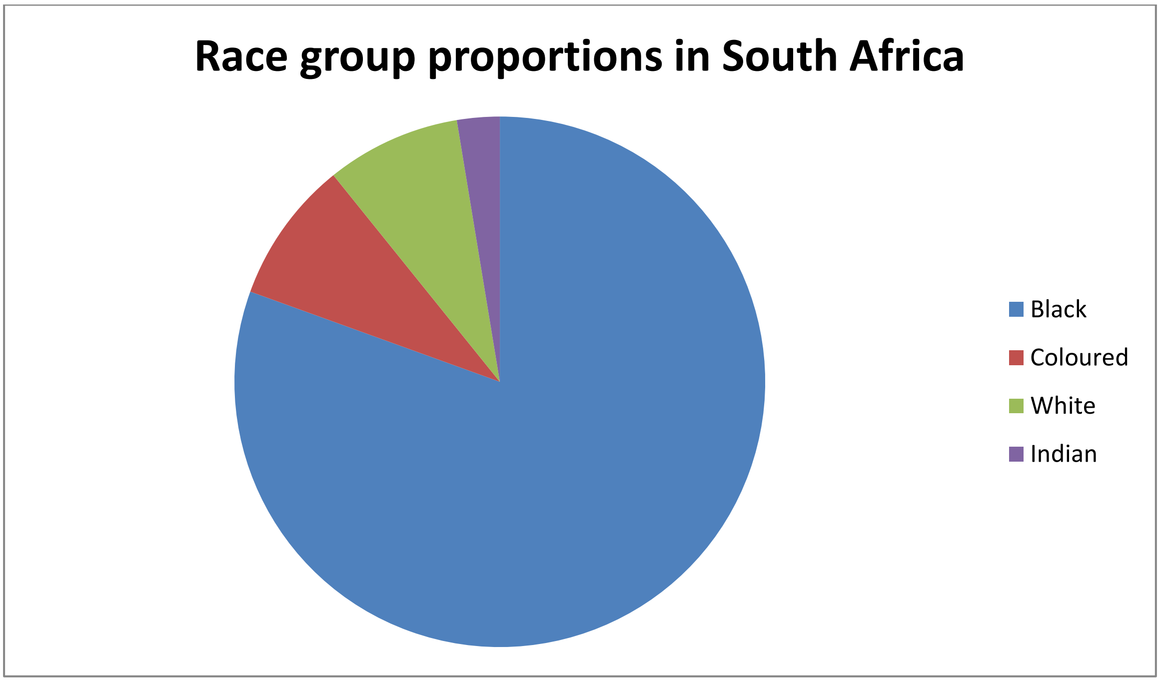

Figure 1: Pie chart representation of the current race group proportions in South Africa adapted from [3]

Figure 1 indicates that the current percentage of Black South African citizens is 80.6%, Coloured is 8.7%, 8.2% of White citizens, and finally the Indian population is 2.6%. The proportion of race in South Africa plays a large role with regards to HIV and AIDS, the disease is an acquired virus which attacks an individual's immune system [9], there is currently no cure for this virus. In South Africa an estimated 7 million people are currently affected with HIV, with an annual death toll of around 180 000 people and an estimated 64% of the infected individuals being of Black decent. Therefore this further justifies the screening process of blood for any blood related diseases and pathogen before it reaches its recipient. Due to the variety of culture in South Africa, the country also experiences a number of different public holidays. These holidays are derived from a variety of events, some of which are issued to honour the past of South African history whilst others are cultural based. In addition to public holidays, educational facilities such as schools and tertiary institutes take mid-term breaks. The importance of these dates relates to the social behaviour aspect that will be represented in the BAP model. In theory, individuals indulge in more dangerous activities during months with more breaks and public holidays, therefore leading to an increase in demand of blood and blood products.

Table 2

Blood Type	A^+	A^-	B^+	B^-	AB^+	AB^-	O^+	O^-
Proportion (%)	32	5	12	2	3	1	39	6

Table 2: Illustrates the proportion of blood types found in the South African Population adapted from [4]

2.2. Assumptions And Limitations

Due to the plethora of external factors which encumber the BAP, certain assumptions had to be introduced in order to create a mathematical model. This includes

I. The blood bank has an infinite supply of capital, and storage space. The external sources (for importing WB units) also have a limitless supply.
II. The time frame will be conducted over 365 days, with day 1 receiving no carryover of WB units from the previous day.
III. Validity of blood will be set at 30 days.
IV. All blood types will first fulfil requests associated with their blood types, from there the remainder from each blood group can contribute to other compatible blood types.

3. Problem Description

The demand for blood in a day must be met. If the demand exceeds to current supply on hand, the blood bank must then import additional units from external sources in order to fulfil the requests. Each day the blood bank receives WB units in the form of voluntary blood donations. The new supply of blood is then added to the existing supply (any units remaining from the previous day) of blood on hand, the new blood units enter a queuing system with the newer units being placed at the end of the queue. The purpose of the queuing technique allows for the oldest WB units to be used first which in turn minimises possible expiration of blood units. The total supply at the start of the day equates to the sum of the remainder from the previous day plus the total amount of donations received in the current day. Any units that exceed their shelf life are discarded from the blood bank. Typically a patient should receive their own blood type during transfusion, however if there is an inadequate supply of that specific blood type, this would then result in using compatible blood types. Before blood is pulled from other compatible blood groups, each blood type must fulfil all their blood type requests for the day, if there is any remainder after fulfilling such requests only then can it be distributed to other blood compatible groups. Overall the BAP can be summarized into 4 major components

I. Supply: Relates to the current stock of WB units on hand at any given moment.
II. Demand: Relates to both planned and unplanned events
III. Importation: Additional WB units are imported when demand exceeds supply for a day
IV. Expiring: WB units that have exceeded their shelf life are destroyed.

Figure 2

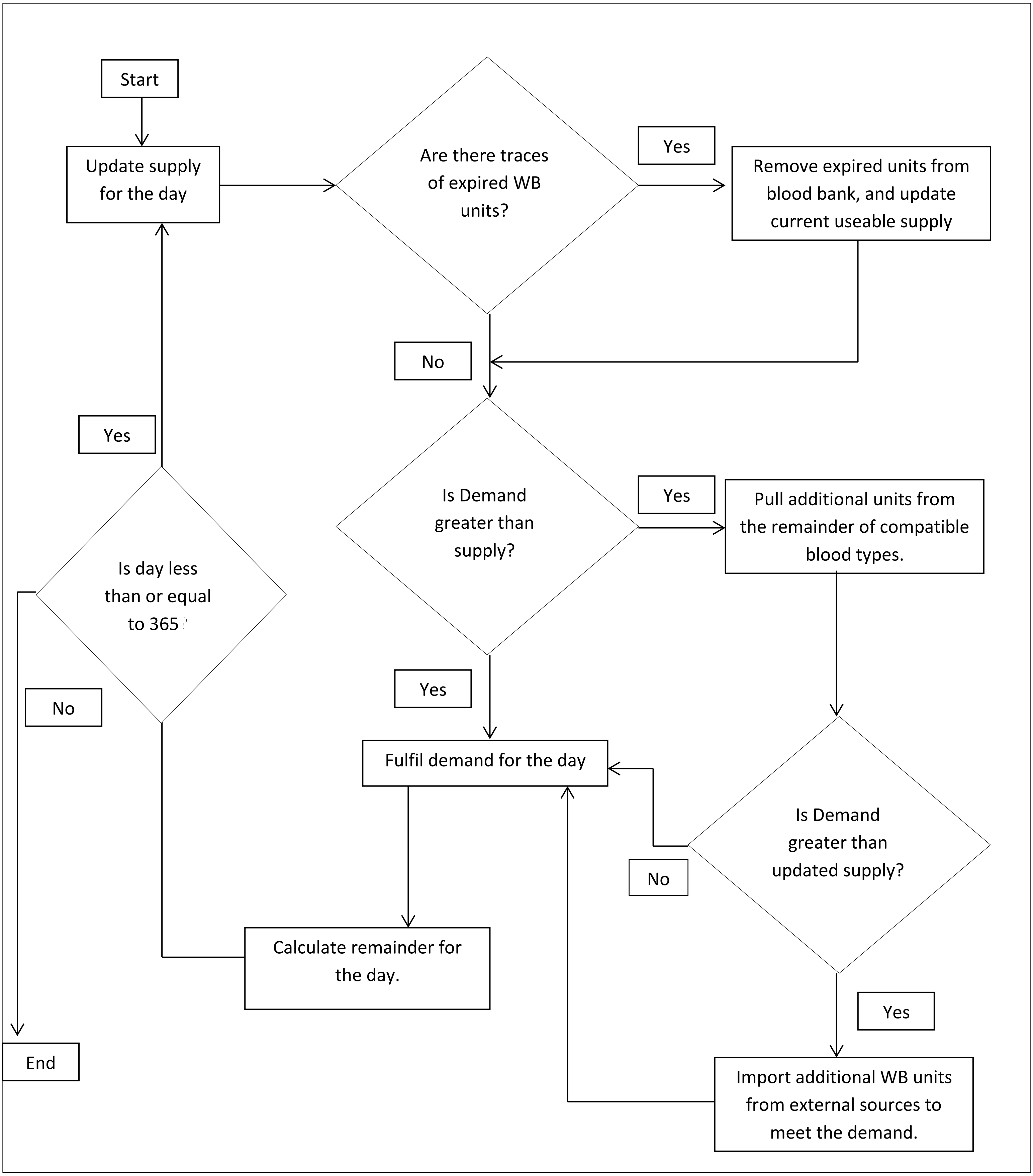
Start
Update supply for the day
Are there traces of expired WB units?
Yes
Remove expired units from blood bank, and update current useable supply
No
Is Demand greater than supply?
Yes
Pull additional units from the remainder of compatible blood types.
Yes
Is day less than or equal to 365
Yes
No
Fulfil demand for the day
Is Demand greater than updated supply?
No
Calculate remainder for the day.
Yes
Import additional WB units from external sources to meet the demand.
End

Figure 2: Flow diagram representing the daily processes that occur with the blood bank in this study

4. Methodology

As mentioned previously, the demand and supply for WB units follow a stochastic trend. In an ideal day the supply for each blood type would meet the exact demand level which in turn eliminates importation from additional units, as well as carrying over excess stock which opens the WB units to possible expiry. Therefore, several MH algorithms were implemented which randomly generates a demand and supply based on South African social trends with each implementation trying to find the best possible solution for the day. Upon the MH implementations, there are four aspects which are combined to offer a solution for the BAP, and consequently offer optimal WB unit assignment in relation to demand. The four components include the MH implementations, the blood compatibility process, expiring old WB units, importing additional WB units when demand exceeds supply for a day.

4.1 Objective function

The objective function for the BAP is given by equation (eq) 1. The aim relates to minimizing the total amount of importation of WB units, as well as to minimize the expiration experienced by the blood bank.

Let

I: Represent the amount of importation
E: Represent the amount of expiration
d: Represent the day

$$\text{Min } \sum_{d=1}^{n} (I_{Total} + E_{Total})\,(d) \qquad (1)$$

Where:

$1 \leq d \leq 365$

$$I_{total} = I_A^+(d) + I_A^-(d) + I_B^+(d) + I_B^-(d) + I_{AB}^+(d) + I_{AB}^-(d) + I_O^+(d) + I_O^-(d)$$

$$E_{total} = E_A^+(d) + E_A^-(d) + E_B^+(d) + E_B^-(d) + E_{AB}^+(d) + E_{AB}^-(d) + E_O^+(d) + E_O^-(d)$$

By minimizing the objective function stated in eq 1, the BAP will be successfully optimized.

4.2 Generating Demand And Supply

Due to confidentiality issues, it was not possible to use datasets from hospitals/clinics in this study. In order to test each implementation, values for both demand and supply had to be randomly generated. In order to generate more accurate values, this study incorporated South African social trends based from monthly statistics. Ideally the most adequate statistics would be related to monthly usage of blood products in the country, however these statistics could not be found. Instead this study will incorporate monthly holidays as well as school terms and breaks from other educational institutions. The ideology behind this method tries to show that the demand for blood has trends associated with a specific month, for example demand would be expected to have a higher value in a month like December due to many people being off from work, school and other institutions, as well as the rise of dangerous events such as drinking and driving and criminal activities. Reports have shown that South Africa experience an increase in the amount of drunk driving levels during Easter [10], therefore blood banks tend to stock-pile blood products for precautionary measures. Taking this and other social trends into account, it is possible to allocate each month with a specified percentage range for generating a value for demand. There were no significant events apart from occasional blood drives for generating values for supply, therefore the supply bounds will be set between 25-75%.

Table 3

Educational institutions terms	Start Month	End Month
1	January	March
2	April	June
3	July	September
4	October	December

Table 3: Represents the starting month and ending month of most educational institutions in South Africa [11].

Table 4

Date	Percentage bound (%)
1 January	New year’s day
21 March	Human Rights day
14 April	Good Friday
17 April	Family day
27 April	Freedom day
1 May	Workers day
16 June	Youth day
9 August	Woman’s day
24 September	Heritage day
16 December	Day of recognition
25 December	Christmas day
26 December	Boxing day

Table 4: Represents the public holidays in South Africa within a year

Using tables 3 and 4 from above, it is now possible to link each month with a unique percentage bound

Table 5

Month	Upper and lower percentage bounds (%)
January	35-85
February	25-50
March	25-75
April	65-90
May	25-75
June	35-85
July	65-90
August	25-75
September	10-50
October	25-75
November	25-75
December	65-90

Table 5: Illustrates the percentage bounded ranges used for generating demand.

Using the percentage bounds in table 5, it is now possible to generate demand, as well as supply using eq 2.

Let:

A: Represent the initial volume in a blood bank
d: Represent a day
m: Represent a month
b: Represent a blood type
Ub: Represent the upper percentage bound
Ul: Represent the lower percentage bound

$$(Supply_b/Demand_b)_d = A \cdot (\text{ rand } (Ub - Ul)_m) \qquad (2)$$

From eq 2, the supply or demand is generated by randomly selecting a percent between the upper and lower bounds depending on the month the system is currently in (this is established in accordance to the current day). This is then multiplied by the initial volume in a blood bank which generates a value for supply or demand.

Once a value has been generated, the value is then split into 8 sub values in accordance to table 2, this accurately represent the quantity in accordance to blood proportion in the South African population.

4.3 Updating blood supply

Updating stock of WB units has two components. Component 1 relates to daily donations received to the blood bank. Component 2 is the addition of the previous day's remainder added onto the new stock of the day. If the system is in the first day, the remainder equates to 0.

Let

R: Represent the remainder
d: Represent a day
b: Represent a blood type

$$(Supply_b)_d = (Supply_b)_d + (R_b)_{d-1} \qquad (3)$$

Where $d \geq 1$

4.4 Expiring Blood Units

WB units are considered a perishable commodity due to its limited lifespan. The WB units can be frozen to prolong its lifespan, however this adds further costs incurred by the blood bank. This study neglects the use of frozen WB units, and sets expiration of these units to 30 days. This implies that a WB unit will be discarded if it is not used within 30 days of its first entry into the blood bank. The following algorithm states conditions that must be satisfied in order for expiry to occur. It is unlikely for expiry to occur when the daily demand and supply have similar levels or the daily demand exceeds the daily supply over a period of days, if this phenomenon occurs then it is unlikely for a unit of blood to be on the shelf for 30 days.

Algorithm 1

Input: S: Supply
D: Demand
Sumd: Demand summed over a specified period
d: day
E: Expiration for the day

```
If d >30 Then
    for integer i < 30
        Sumd += D_d
    End for loop
    If Sumd < (S_b)_{d-30} Then
        E = ((S_b)_{d-30}) – Sumd
    End if
End if
```

Algorithm 1 only occurs after day 30, due to WB units having a lifespan of 30 days. Therefore it’s impossible to have units expiring before this time, this also allows the implemented system to be more efficient.

4.5 Importing additional blood units

Importing additional blood units to meet the demand in a day results in added expenses incurred by the blood bank, logically minimizing importation results in lower expenses and utilising resources more effectively. Two conditions have to occur before importation can take place, these include

- Demand exceeding supply in a given day.
- Demand still exceeds supply after additional blood units are pulled from compatible blood types.

If these two conditions are satisfied, only the can additional blood units can be imported from external sources. In theory, importation should have higher levels in the first few starting days of a planning horizon, once an accumulation of certain blood types occur, importation starts to decline.

4.6 Bottom-up technique

When the WB units on hand cannot meet the demand for a day, additional units from other compatible blood types are used. Each blood type must fulfil their corresponding requests before distributing towards other compatible blood types. The bottom-up technique relates to a system which pulls from compatible blood types. Therefore it can be established that the remainder from a day is then split according to the number of possible compatible types. By implementing this technique, the blood bank will reduce importation of blood units, and utilises its resources more effectively. There can exist some medical cases which require the patients specific blood type, however this study has chosen to ignore this occurrence.

Table 6

Blood type	Can distribute to	Splitting
A^+	A^-, O^+, O^-	$\frac{(RemainderA+)}{3}$
A^-	O^-	$\frac{(RemainderA+)}{1}$
B^+	B^-, O^+, O^-	$\frac{(RemainderA+)}{3}$
B^-	O^-	$\frac{(RemainderB-)}{1}$
AB^+	A^+, A^-, B^+, B^-AB^-, O^+, O^-	$\frac{(RemainderA+)}{7}$
AB^-	A^-, B^-, O^-	$\frac{(RemainderA+)}{3}$
O^+	O^-	$\frac{(RemainderA+)}{1}$
O^-	N/A	N/A

Table 6: Represents the blood types and the compatible blood types it can distribute towards

The process of pulling blood is conducted according to the order blood proportions in South Africa (Table 2). The higher the proportion of a blood type results in that blood type distributing first. For example, if A^+ to pull from additional units, it would first pull from O^+ blood which has a proportion of 39%, if the demand is still not satisfied, more blood units will be pulled from O^- and finally if the demand still exceeds supply more units will be pulled from A^-. After conducting the bottom-up technique, if the demand still exceeds supply, then additional WB units will be imported. The act of pulling from compatible blood units in this manner tries and tries to maximize the storage of blood types which have the lowest proportion (have a higher rarity). More common blood types also have a higher percentage of resupply.

4.5 Individual representation

Due to the limited number of different blood types in humans, it is therefore possible to create a finite individual of size eight (eight blood types) with each segment in the individual of a type double to take into account a relevant value for supply, demand, importation and expiration. The following figure represents a typical individual.

Figure 3

Position 1	Position 2	Position 3	Position 4	Position 5	Position 6	Position 7	Position 8
A^+	A^-	B^+	B^-	AB^+	AB^-	O^+	O^-

Figure 3: Representation of an individual used in the MT implementations

Each segment from figure 3 stores a value which is calculated using the blood proportion percentage found in the South African population.

4.6 Genetic Algorithm

The GA implementation employs the use of populations consisting of individuals along with genetic operators such as selection, recombination and mutation [2] which was introduced by Koza in 1994 [12]. The genetic operators create diversity in a population in order to find the best possible solution to meet the demand for the day. This diversity also tries to eliminate premature convergence. The structure of the individual (figure 3) makes it possible to implement the genetic operators.

4.6.1 Selection

The process for selectin g specific individuals from a population was conducted using tournament selection. A specified tournament size is established, from here the 2 individual with the greatest fitness values are used for the next process.

4.6.2 Crossover

The 2 individuals that were previously selected are now subjected to the crossover. Conventionally a crossover operator would select n (where n > 0) random crossover points in each individual and swap the genes accordingly. Due to the unique nature of the individual, swapping random points would result in inaccurate readings based on the blood percentages in the population. For example, a case could arise where the A^+ segment (which has a relatively high percentage) could swap with a lower percentage segment such as O^-. Due to this possibility, this study implemented uniform crossover which selects n random points in both individual and swaps their corresponding values. After conducting the crossover method, the algorithm is now left with 2 newly formed individuals, each of these individuals have their fitness calculated, with the fittest individual being chosen and subjected to the next step. Figure C depicts the crossing over method.

Figure 4

Random crossover positions: 1, 3, 7

Original Individual 1

48	8	18	3	5	2	59	11

Original Individual 2

57	6	22	2	6	2	47	8

After crossing over

Offspring 1

57	8	22	3	5	2	47	11

Offspring 2

48	6	18	2	6	2	59	8

Figure 4: Depicts the crossing over effect between 2 individuals and the results obtained

4.6.3. Mutation

Mutation alters part of the individual to obtain a newer individual. This study used point mutation, the process randomly selects n number of points in the individual, and recalculates the value at that position, the recalculation process will only occur if that value in the supply individual does not equal to the value in the demand individual. For example, if position 5 would be selected, the algorithm would then generate a new value for supply by initial random amount, and multiplying it by 39% (proportion of the blood type in South Africa).

Once these steps have been completed, the individual is then placed into a new population (regeneration process), with the cycle continuing until the maximum generation size is met, or a solution is found for the day. Figure D illustrates the mutation process

Figure 5

Mutation position: 3, 4

Original Individual 1

48	8	18	3	5	2	59	11

After mutation process

Mutated Individual 1

48	8	20	5	5	2	59	11

Figure 5 illustrates the mutation process of an individual and the result obtained

4.7 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) is a population based metaheuristic which utilises a swarm of particles to perform is optimization process [4]. The algorithm begins by randomly distributing particles in a solution space and then begins its iterative process to try and locate the best solution. The optimization process relies on communication between particles in order to establish movement of the particles within the search space. The particles utilise both the experience of itself, as well as reachable neighbouring particles to guide the searching process. Given an n-dimensional space, each is characterized by a position vector $X_i = (X_{i1} \ldots X_{in})$ as well as a velocity vector $V_i = (V_{i1} \ldots V_{in})$. Both X_i and V_i make use of the following equations to iteratively update themselves.

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (P_i - X_i) + c_2 r_2 (P_g - X_i) \qquad (4)$$

$$X_i(t+1) = X(t) + V_i(t+1) \qquad (5)$$

Where
P_i: Represents the best personal best position
P_g: Represents the global best position
r_1, r_2: Represents random values between [0, 1]
c1, c2: Represents scaling parameters.
ω: Represents the inertia weight
t: Represents the iteration index

A further look at the parameters in correlation with equations (4) and (5) reveal that c_1 and c_2 exert random forces in the direction of both P_i and P_g, whilst the ω value aids in regulating the velocity which in turn helps to balance global and local searches. In this paper the PSO system was customized in order to conform to the BAP.

4.7.1. Particles and particle positions

The study conducted by [4] used a string representation (using letters corresponding to blood types) for the daily demand and supply, and transformed the string into values to represent certain components in equations (4) and (5). This study opts out of a string representation, instead using the individual representation in figure 3, it is possible to sum each segment to obtain a value which can relate to either X_i, P_i and P_g.

4.8. Duellist Algorithm

The Duellist Algorithm (DA) is based on the GA approach, which was inspired by human fighting and the ability of learning [13]. With the DA approach, all the individuals within a population are referred to as duellist, with the aspect of fighting to determine champions, winners and losers within the population. Unlike the GA approach which produce blind solutions (blind solutions relate to individuals being produced that may not be a better solution), the DA subjects loser individuals to learn from the winner which tries to minimize the blind effect. A winner between two individuals is based on the physical nature of an individual (fitness value) as well as a luck coefficient (LC), which is a randomly generated value. The DA implements several steps before conducting a duel between 2 individuals.

a) **Pre-Qualification**

If the duellist is below above a set fitness level, then the duellist is not selected to duel.

b) **Board of champions**

The board of champions aims at keeping the best duellist in the competition. The purpose of the champion is to train newer duellist to compete against each other. If the newer duellist has a better fitness than the original champion, then the duellist swaps positions with the champion.

c) **Duelling schedule**

The schedule between 2 duellists is set randomly, with each duellist using their fighting potential as well as LC to determine a winner. Conventionally, the higher the fighting potential and LC coefficient results in an individual having a greater chance of becoming a winner. In accordance to the BAP, the best solution is considered to be the individual with the lowest fitness value, therefore to adapt the DA in conjunction with the BAP, the inverse function of the randomly generated LC value is added to the fitness using the following equations and algorithm.

Algorithm 2

```
Input: DuellistA, DuellistB, LC

A(Luck) = [A(Fighting Capabilities) · (LC + (rand(0-1) · LC))] -1
B(Luck) = [B(Fighting Capabilities) · (LC + (rand(0-1) · LC))] -1

    If (A(Fighting Capabilities) + A(Luck) ≤ B(Fighting Capabilities) + B(Luck)) Then

        Winner = DuelistA
        Loser = DuelistB

    Else

        Winner = DuelistB
        Loser = DuelistA

End
```

d) **Duellist improvement**

After conducting the duel, the duellists are categorized either as a winner, loser or champion. The loser and winner are then treated to a form of learning in order to improve themselves. The loser learns from the winner, whilst the winner trains himself by randomly regenerating values for each segment only if that segment does not match the demand for a day, in hopes that the new result is better than the previous value. Since the demand and supply for a day follow the same individual representation, if the segment in the winner or loser individual matches the demand segment, then the segment does not change.

Figure 6

Demand for the day

21	3	8	1	2	1	26	5

Winner

20	2	10	0	2	1	27	4

Loser

18	3	8	4	3	1	26	2

Individuals after training

Winner

20	3	10	0	2	1	27	4

Loser

20	3	8	0	0	1	26	5

Figure 6: Illustrates how training occurs in the Loser individual to try and improve upon its original representation.

4.9. Symbiotic Organism Search

The Symbiotic organism search (SOS) algorithm simulates the interactive behaviour of organisms within nature [14]. The search is broken into 3 main categories namely Mutualism, Commensalism and Parasitism, each of these phases alters an individual(s) within a population attempting to obtain a better solution than its original representation.

Mutualism: Organisms interact with each other in a way that benefits both parties.

Let X_i and X_j represent 2 random individuals within a population, and MV represent the Mutual Vector.

$X_{inew} = X_i + rand(0, 1) \cdot (X_{best} – MV \cdot BF1)$ (6)

$X_{jnew} = X_j + rand(0, 1) \cdot (X_{best} – MV \cdot BF2)$ (7)

Where:

$MV = (X_i + X_j) / 2$ (8)

The value obtained from $(X_{best} – MV)$ tries to increase survival in the population, with all improved individuals replacing the original individuals.

Commensalism: Organisms interact with each other in a way that results in one organism benefit without harming or altering the other organism. Selection of two organisms is done randomly from the population, and have their fitness values evaluated, the fitter individual is labelled as X_i and the inferior individual is labelled as X_j.

$X_{inew} = X_i + rand(-1, 1) \cdot (X_{best} - X_j)$ (9)

X_i benefits from X_j by means of $(X_{best} - X_j)$ (10)

Parasitism: Organisms interact with each other in a way that benefits one organism (parasite) whilst harming the other organism (host). To evaluate a form of parasitism for the BAP, 2 individuals form a population are randomly selected, with each of its fitness values evaluated similar to the commensalism phase. Following the evaluation the fitter individual is labelled as the parasite, and the inferior as the host. The parasite then swaps segments of its representation with the host only if the value (from the host) improves its original solution. The following algorithm illustrates the parasitism procedure.

Algorithm 3

```
input: demand, host, parasite

    for (i to parasite length)
        if (parasite [i] != demand [i])

    // ensure that the parasite segment value does not match the value of the demand
    // segment for the day
            store diff1 = demand [i] – host [i]
            store diff2 = demand [i] – parasite [i]


        end if

    if (diff1 < diff2)
    Implies that the value of host is closer to the demand for the day, swap host and
    parasite segments.

    if (diff1 ≥ diff2)
    Implies parasite value already contains a value closer to the demand for the day.
    Therefore, do not replace value.

end
```

4.10 Grey Wolf Optimizer

The Grey wolf optimizer (GWO) was inspired from the canine family, with wolves being considered as apex predators (top of the food chain) [15]. The algorithm is moulded around a pack of wolves (the pack ranging between 5- 12 wolves), with each pack containing 3 key members the first, second and third also referred to as the alpha (α), beta (β) and delta (δ) respectively. Each of these wolves have their own tasks within the pack, for example the α is tasked with leading the pack, β wolf aids the α in decision making and is second in command, with δ wolves having to submit to α and β wolves. The lowest ranking wolves are referred to as omega wolves, which are considered the scapegoats of the pack. The GWO is defined as a predatory space of artificial wolves contained in a NxD where N is the number of wolves and D is the amount of variables of the BAP. The ith position of a wolf is represented by $X_i = (X_{i1,} X_{i2} \dots X_{iD})$ with X_{id} is the dth variable value of the ith artificial wolf. Each value for X is represented by the sum of the supply for the day, with the summed demand value representing the prey. According to [16] the hunting patterns of grey wolves follow a certain pattern.

- Tracking, chasing and moving towards the prey.
- Encircling, and harassing the prey until it stops moving
- Move forward to attack the prey.

Figure 7

Figure 7 represents 5 images: (A) Tracking prey, (B-D) pursuing, harassing and encircling and (E) situation before attack

Using this information, the GWO can be broken into individual components and mathematically modelled.

4.10.1 **Wolf pack hierarchy**

The fittest individual within the pack will be deemed as the α wolf, likewise the second and third fittest individual will be the β and δ wolf respectively, whilst the remaining wolves will be deemed as the ω. The GWO algorithm utilises this hierarchy in order to conduct the optimization process.

4.10.2 **Encircling the prey**

As mentioned previously, wolves encircle their prey during a hunt. The following equations are proposed for calculating the encircling behaviour.

Let A and C represent coefficient vectors.

$$\vec{C} = 2 \cdot \overrightarrow{r2} \quad (11)$$
$$\vec{A} = 2\vec{a} \cdot \overrightarrow{r1} \cdot \vec{a} \quad (12)$$

Where $\vec{a}$ is linearly decreased over a set number of iterations from 2 to 0. Whilst $\overrightarrow{r1}$, $\overrightarrow{r2}$ represent random vectors between 0 and 1

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (13)$$
$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (14)$$

Where t indicates the iteration, $\vec{X}_p$ is the position vector of the prey and $\vec{X}$ represents the position vector of the hunter (grey wolf).

4.10.3 **Hunting**

The hunting component utilises the alpha wolf to lead the hunt, the beta and delta wolf may part-take in the hunt occasionally. It is assumed that the alpha, beta and delta wolves have better knowledge regarding the prey than the omega wolves. Due to the alpha taking the lead in the hunt, we assign the best candidate solution to the alpha wolf, and in ascending order of fitness, allocate the remaining candidate solutions to the beta, delta and omega wolves. After allocation of candidate solutions, the wolves then iteratively update their positions using the following equations.

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (15)$$
$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (16)$$
$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (17)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 (\vec{D}_\alpha) \quad (18)$$
$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_1 (\vec{D}_\beta) \quad (19)$$
$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_1 (\vec{D}_\delta) \quad (20)$$

$$\vec{X}(t+1) = (\vec{X}_1 + \vec{X}_2 + \vec{X}_3) / 3 \quad (21)$$

4.10.4 **Exploitation (Attacking process)**

By decreasing the value of $\vec{a}$ over a set number of iterations, this mimics the process of a wolf approaching the prey. According to eq. (12), $\vec{a}$ is a component in calculating $\vec{A}$ which in turn decreases $\vec{A}$. The values for $\vec{A}$ lies between [-1, 1] with each position of the search agent lying between this specified range, if $\vec{A}$ is less than 1, this can be deemed as the wolf moving towards from the prey.

4.10.5 **Exploration (Searching for prey)**

As wolves search for prey, they tend to diverge from the pack, and then converge during an attack. The divergence pattern can be calculated using the value of $\vec{A}$ bounded between [-1, 1] this allows for global exploration to take place. Exploration is also favoured by component $\vec{C}$ which contains random values between [0, 2]. Component $\vec{C}$ allocates random weights to the prey in order to emphasize ($C > 1$) and deemphasize ($C < 1$). As mentioned previously, A is linearly decreased, by C is assigned random values to emphasize the exploration process. Component C can also be interpreted as a naturally occurring obstacles which occur during a hunt. Depending on the position of the wolf, C can randomly give the prey a weight which either makes it easier or harder for the wolf to catch the prey.

4.10.6 General implementation

The GWO starts with an initial random population of grey wolves (candidate solutions) which in correlation with the BAP, are represented as the supply of blood units for the day. The demand for WB units in a day is interpreted as the prey. The position of each wolf and prey equates to the sum of the values of each segment within the individuals representation (figure 3). Using the equations stated in (11-21), each candidate solution iteratively updates their position from the prey, with the parameter $\vec{a}$ being linearly decreased from 2 to 0 over a number of iterations. Candidate solutions tend to converge towards the prey when $|\vec{A}| < 1$ and diverge when $|\vec{A}| > 1$. The termination criteria terminates when the max number of iterations have been reached, or the supply equates to the demand for a day.

4.11 Datasets

Each MH implementation was subjected to 6 datasets, with each dataset serving as a situation that could possibly occur within the blood bank. As mentioned previously, due to confidentiality issues data could not be attained from South African hospitals/clinics, this resulted in datasets being randomly generated. Each dataset were allocated a specific percentage bound and initial blood volume amount. Note that the South Africa generated values (percentage bounds based on South African statistics) are represented by “SAGV”.

Table 7

Dataset	Initial Blood Volume	Demand bounds (%)	Supply bounds (%)
1	500	25-75	25-75
2	500	SAGV	25-75
3	500	75-100	25-50
4	500	25-50	75-100
5	5000	25-75	25-75
6	5000	SAGV	25-75

Table 7: Illustrates the datasets used within this study

Dataset 1: Serves as a control dataset. These percentage bounds where used in the study conducted by [4] and [6].

Dataset 2: Uses percentage bounds based from South African statistics for generating the value for demand. The use of this dataset tries to exemplify the idea of each month in South Africa should have a unique level of demand.

Dataset 3: Tests a situation within a blood bank where demand exceeds supply.

Dataset 4: Tests a situation within a blood bank where supply exceeds demand.

Dataset 5: Similar to dataset 1, but tests a situation with a larger volume of initial blood.

Dataset 6: Similar to dataset 2, but tests a situation with a larger volume of initial blood.

4.12 Parameter setting

- **GA**: Iterations=1000, Population size= 50, Mutation rate= 30%, Crossover rate=20%, Regeneration rate= 50%
- **PSO**: Iterations=1000, Swarm Size=50, $c_1 = c_2$= 1.7, ω= 0.715.
- **DA**: Iterations= 1000, Tournament size=50.
- **SOS**: Iterations=1000, Organisms=50.
- **GWO**: Iterations=1000, Pack size=50.

5. Results And Discussion

This section will provide line graphs for each MT in accordance to each dataset, as well as averages achieved per blood type and running time per dataset. The objective function relates to minimizing the overall importation and expiration of blood units, whilst ensuring that all blood demands are met within a day. A solution is found if the demand and supply for each blood type is identical, this results in no remainder (which could possibly expire) and no form of importation. However, due to the remainder of a previous day being added to the donations received by the current day it is unlikely that a solution could be found. Therefore, to evaluate the results of each MT 3 different aspects will be evaluated.

I. Running time for each MT
II. Average amounts for importation and expiration
III. Time taken (measured in days) when stock-piling occurs

The first 2 points are self-explanatory. The third point relates to stock-piling, stock-piling implies to a period when supply keeps increasing whilst demand levels remain approximately the same. The aim of stockpiling tries to minimize the amount of importation of certain blood types, usually blood types with a higher proportion in society results in a shorter period for stock-piling to occur, unlike smaller proportion blood types which may never experience stock-piling.

Table 8

MT	Variable	A^+	A^-	B^+	B^-	AB^+	AB^-	O^+	O^-
GA	**Supply**	40.00	6.25	15.00	2.50	3.75	1.25	48.75	8.75
	Demand	192.81	88.67	78.67	35.41	15.27	6.36	131.83	87.80
	Import	0.00	0.00	0.08	0.01	0.28	0.01	0.02	0.00
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PSO	**Supply**	162.7	7.25	61.35	3.26	16.16	12.90	47.51	9.94
	Demand	40.51	6.33	15.19	2.53	3.80	1.27	49.37	8.86
	Import	2.81	1.71	0.67	0.70	0.48	0.02	14.90	2.39
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DA	**Supply**	295.00	133.40	120.90	56.00	26.50	11.00	176.20	136.40
	Demand	39.27	6.14	14.72	2.45	3.68	1.23	47.86	8.59
	Import	0.45	0.03	0.17	0.00	0.25	0.02	0.35	0.02
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SOS	**Supply**	192.81	88.67	78.67	35.41	15.27	6.36	131.83	87.80
	Demand	40.00	6.25	15.00	2.50	3.75	1.25	48.75	8.75
	Import	0.00	0.00	0.08	0.01	0.28	0.01	0.02	0.00
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GWO	**Supply**	60.07	30.10	20.73	6.41	1.76	1.51	68.73	16.79
	Demand	40.32	6.30	15.12	2.52	3.78	1.26	49.14	8.82
	Import	1.36	0.01	1.07	0.09	2.02	0.26	1.14	0.00
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 8: Average results achieved for each metaheuristic implementation subjected to dataset 1 for each blood group measured in units.

Table 9

Metaheuristic	Time (Ms)	Time(Minutes)
GA	4748287	79.14
PSO	504291	8.40
DA	4899566	81.66
SOS	4776186	79.60
GWO	4752407	79.21

Table 9: Represents the running time per metaheuristic for dataset 1

Figure 8

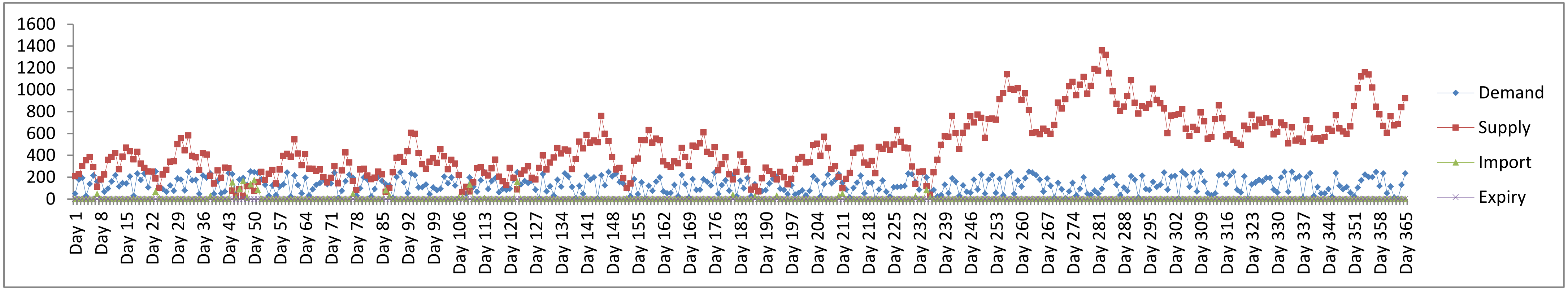

Figure 8: Represents a line graph over a period of 365 days for the GA implementation of dataset 1

Figure 9

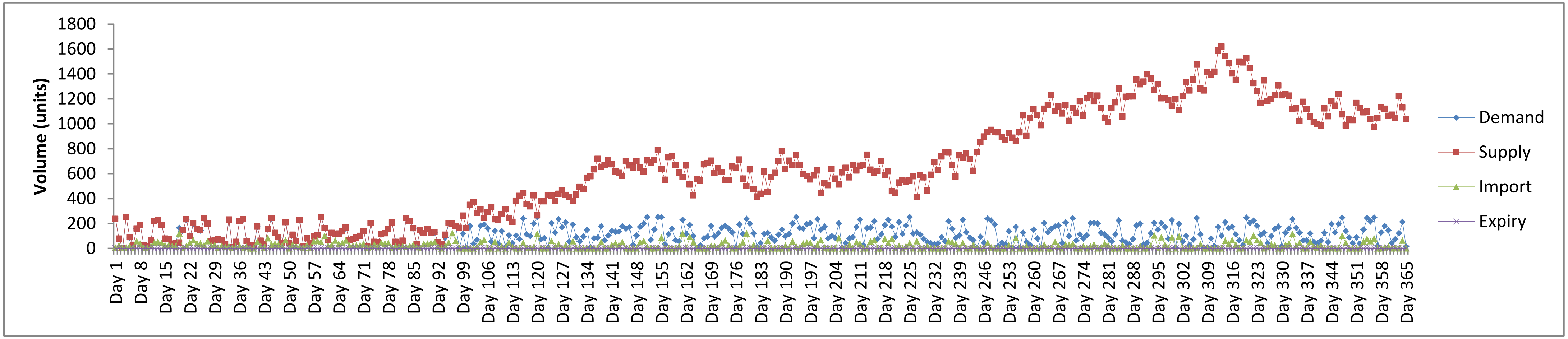

Figure 9: Represents a line graph over a period of 365 days for the PSO implementation of dataset 1

Figure 10

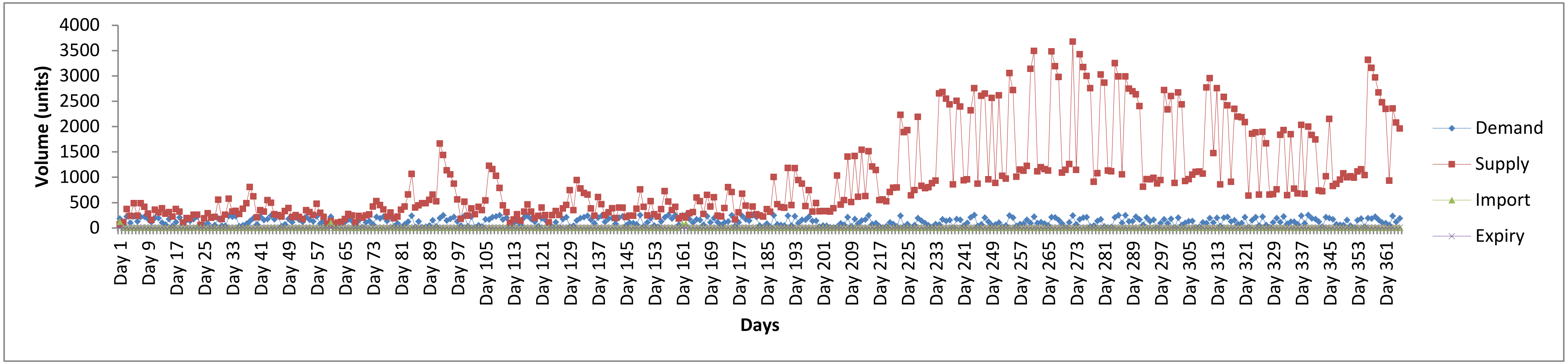

Figure 10: Represents a line graph over a period of 365 days for the DA implementation of dataset 1

Figure 11

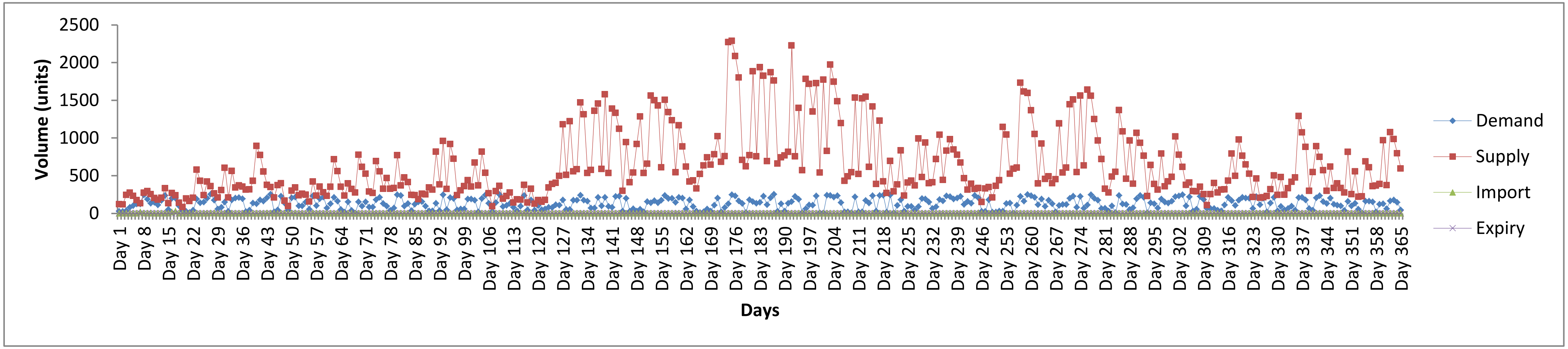

Figure 11: Represents a line graph over a period of 365 days for the SOS implementation of dataset 1

Figure 12

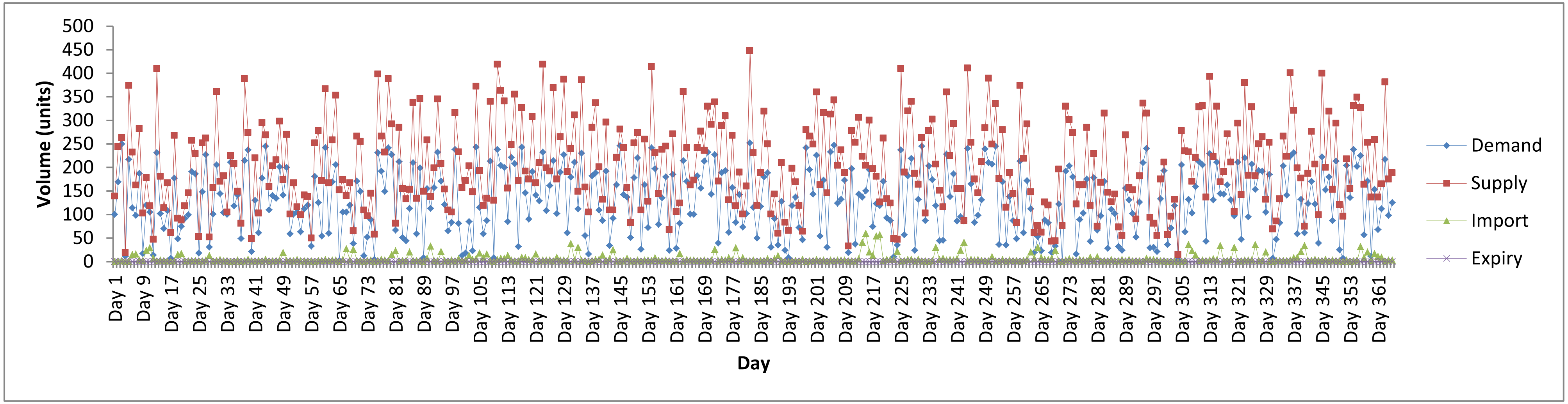

Figure 12: Represents a line graph over a period of 365 days for the GWO implementation of dataset 1

Evaluating results from dataset 1

Dataset 1 is used demand and supply percentage bounds ranging between 25-75% with an initial WB unit volume of 500 units. Looking at the running time for each algorithm shows that PSO was drastically faster than the other implementations with a time of 8.40 minutes, unlike the rest of the algorithms who's times ranged between 79-81 minutes. The GA implementation had a few moments of relatively large importation, and started stock-piling around day 235. The PSO implementation begins stock-piling around day 100, but had relatively high levels of importation for all blood types. DA had very low importation levels, and only started stock-piling around day 190. SOS started stock-piling around day 120, and did not experience any form of importation after this day, this implies that all blood types experienced stock-piling. Finally the GWO implementation had sporadic levels of both demand and supply throughout the 365 days therefore no form of stock-piling occurred and experienced very low levels of importation for A^- and no importation for O^- blood types. Overall the SOS implementation performed the best, even though it was only the second best with regards to stock-piling, the low importation levels coupled with its ability to not import any WB units after stock-piling justifies its selection. No algorithm experienced expiry, even though most implementations seem to have a much higher level of supply after stock piling, the 30 day shelf life makes it difficult for a particular WB unit to remain stagnant for the entire 30 days.

Table 10

MT	Variable	A^+	A^-	B^+	B^-	AB^+	AB^-	O^+	O^-
GA	**Supply**	25.72	4.12	9.65	1.67	2.32	0.90	31.31	5.68
	Demand	38.69	6.05	14.51	2.42	3.63	1.21	47.16	8.46
	Import	18.96	2.94	7.11	1.18	1.82	0.61	23.11	4.12
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PSO	**Supply**	33.26	8.13	16.11	5.07	0.97	2.54	47.98	12.35
	Demand	40.54	6.33	15.20	2.53	3.80	1.27	49.41	8.87
	Import	7.69	0.40	1.19	0.08	2.86	0.24	3.05	0.13
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DA	**Supply**	41.47	6.48	15.55	2.59	3.89	1.30	50.55	9.07
	Demand	48.83	16.24	17.84	6.15	2.19	1.73	56.38	18.24
	Import	7.07	0.68	2.85	0.28	2.06	0.26	8.13	0.91
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SOS	**Supply**	68.90	28.51	27.04	11.63	3.08	2.40	72.98	30.71
	Demand	37.16	5.81	13.93	2.32	3.48	1.16	45.29	8.13
	Import	1.20	0.01	0.39	0.01	1.51	0.10	0.68	0.00
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GWO	**Supply**	16.14	3.20	6.25	0.38	0.85	0.05	19.01	4.46
	Demand	21.06	3.29	7.90	1.32	1.97	0.66	25.66	4.61
	Import	6.43	1.14	2.34	1.10	1.15	0.63	8.10	0.97
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 10: Average results achieved for each metaheuristic implementation subjected to dataset 3 for each blood group measured in units.

Table 11

Metaheuristic	Time (Ms)	Time(Minutes)
GA	3872251	64.54
PSO	494639	8.24
DA	4203022	70.05
SOS	4765186	79.42
GWO	4623043	77.05

Table 11: Running time per metaheuristic for dataset 3

Figure 13

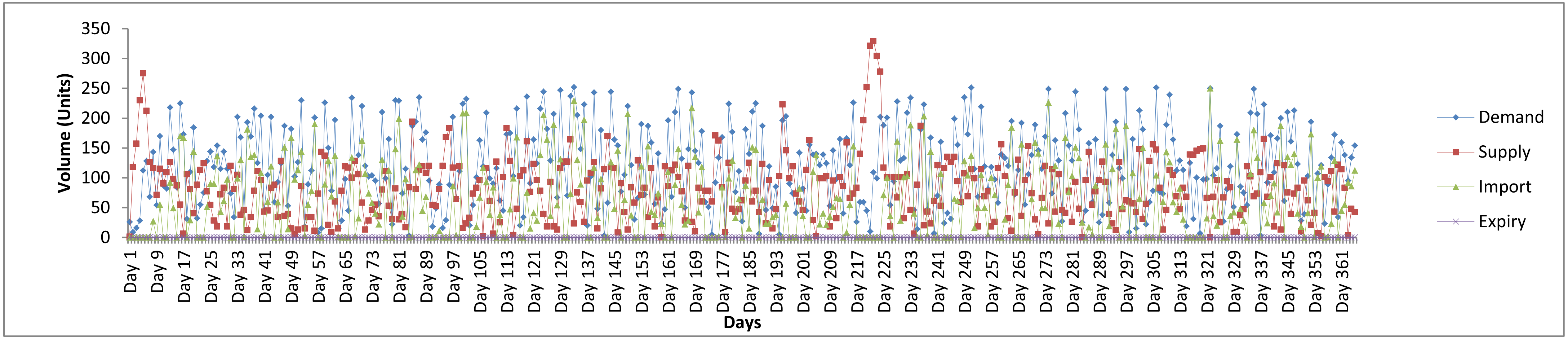

Figure 13: Represents a line graph over a period of 365 days for the GA implementation of dataset 3

Figure 14

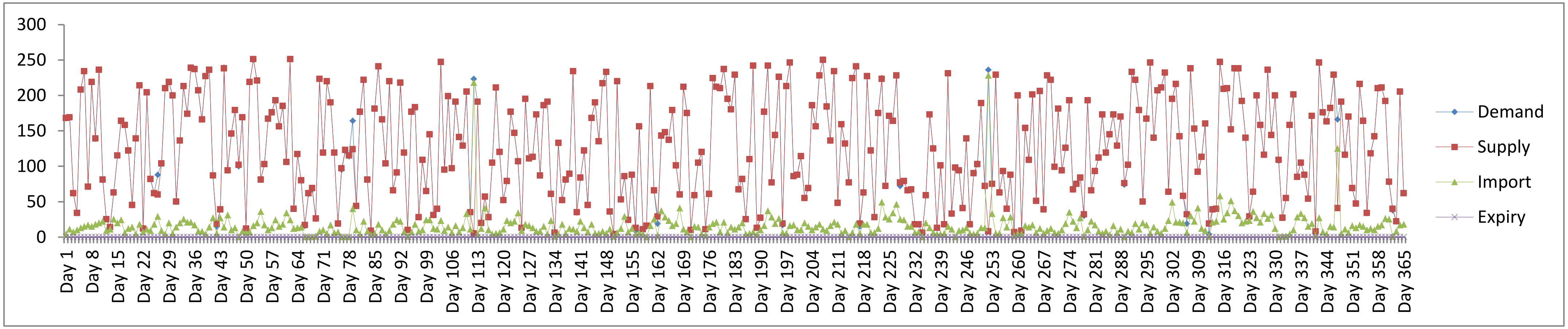

Figure 14: Represents a line graph over a period of 365 days for the PSO implementation of dataset 3

Figure 15

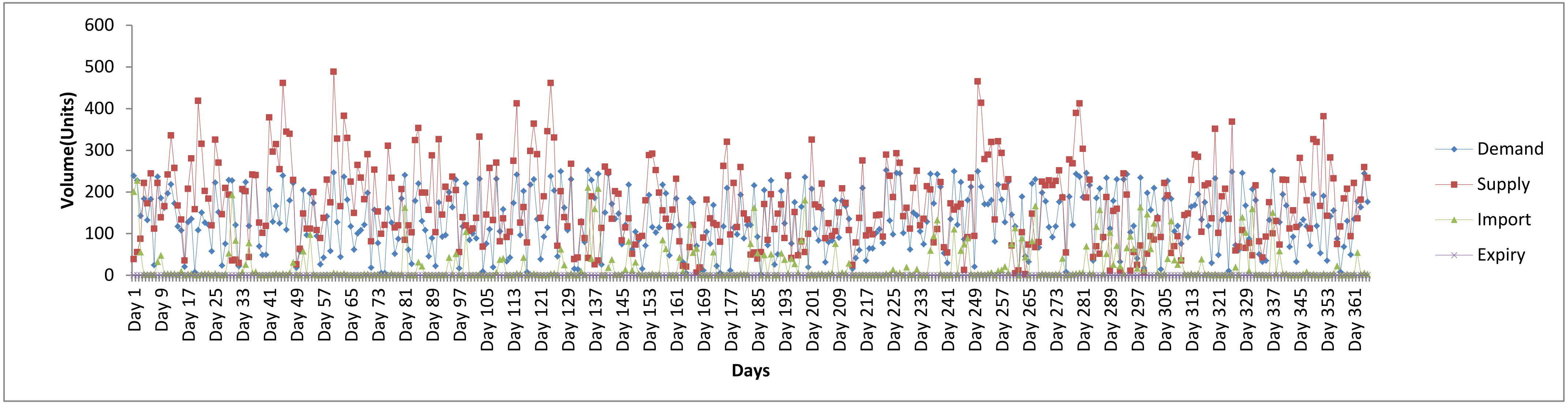

Figure 15: Represents a line graph over a period of 365 days for the DA implementation of dataset 3

Figure 16

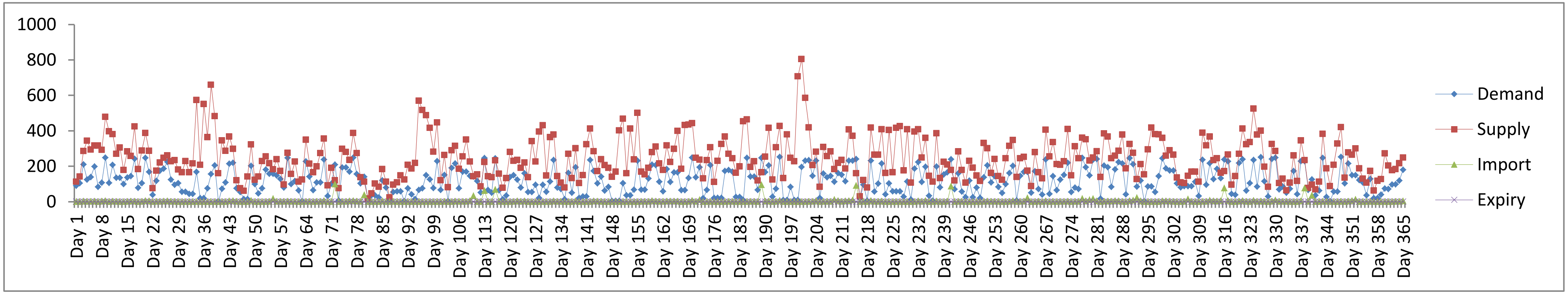

Figure 16: Represents a line graph over a period of 365 days for the DA implementation of dataset 3

Figure 17

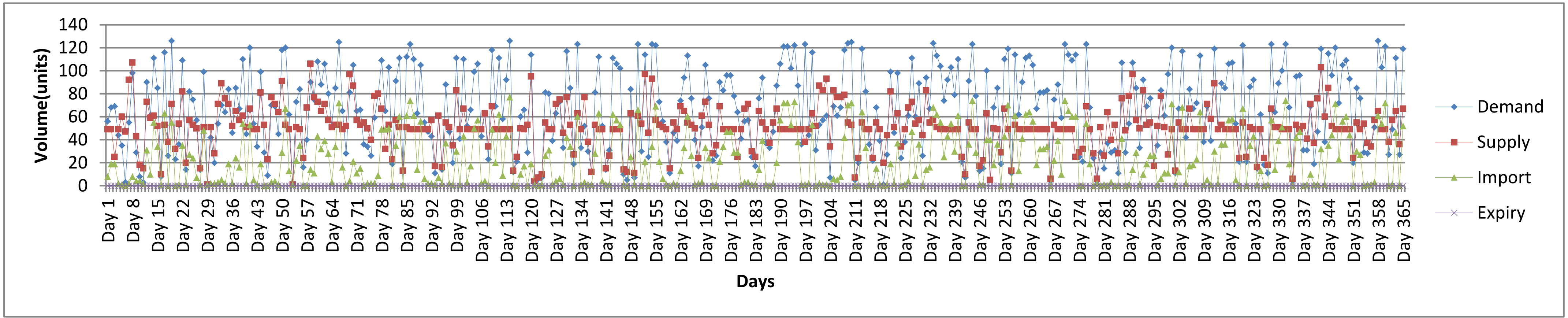

Figure 17: Represents a line graph over a period of 365 days for the GWO implementation of dataset

Evaluating results from dataset 3

Dataset 3 tests a situation where demand exceeds supply for WB units. In these sets of results it would be expected for a large volume of importation to occur, however due to stock-piling it is possible for the implementations to experience brief periods of no importation. In comparison to dataset 1 (control dataset), all implementations experienced an increase in average importation, with relatively similar running times. Even though the SOS algorithm did have a higher importation as compared to dataset 1, it still imported a far lower amount as compared to the rest of the algorithms. The SOS implementation also ended up with a higher supply in relation to the demand which occurred due to stock-piling and using other compatible blood types to meet demand. A closer look at figures (13-17) and it is noticeable that when supply exceeds demand in the beginning days the supply tends to increase more due to stock-piling, however due to a higher demand percentage bound, the supply slowly decreases until importation is inevitable.

Table 12

MT	Variable	A^+	A^-	B^+	B^-	AB^+	AB^-	O^+	O^-
GA	**Supply**	5205.15	816.14	1952.03	320.22	485.21	159.66	6334.86	1136.78
	Demand	383.20	59.87	143.70	23.95	35.92	11.97	467.02	83.82
	Import	1.88	0.31	0.70	0.10	0.28	0.06	2.31	0.39
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PSO	**Supply**	3331.25	80.35	1264.40	40.95	307.45	106.53	503.23	108.92
	Demand	397.69	62.14	149.13	24.86	37.28	12.43	484.68	86.99
	Import	46.06	14.62	5.68	5.70	6.45	0.40	129.94	19.75
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DA	**Supply**	3442.04	1749.02	1325.24	674.90	310.58	108.34	2190.95	1773.78
	Demand	396.28	61.92	148.60	24.77	37.15	12.38	482.97	86.69
	Import	2.87	0.15	0.57	0.00	1.91	0.03	2.24	0.09
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SOS	**Supply**	2404.66	1184.73	882.15	438.40	238.29	86.86	1639.28	1220.06
	Demand	397.86	62.17	149.20	24.87	37.30	12.43	484.89	87.03
	Import	0.00	0.00	0.00	0.00	0.62	0.00	0.00	0.00
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GWO	**Supply**	637.34	252.61	230.45	70.28	19.65	17.99	771.21	161.06
	Demand	424.51	66.33	159.19	26.53	39.80	13.27	517.38	92.86
	Import	7.73	0.21	6.15	0.39	20.15	0.93	5.22	0.00
	Expiry	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Average results achieved for each metaheuristic implementation subjected to dataset 5 for each blood group measured in units.

Table 13

Metaheuristic	Time (Ms)	Time(Minutes)
GA	4422080	73.70
PSO	504499	8.40
DA	4203022	70.05
SOS	5805914	96.76
GWO	4385240	73.09

Running time per metaheuristic for dataset 5

Figure 18

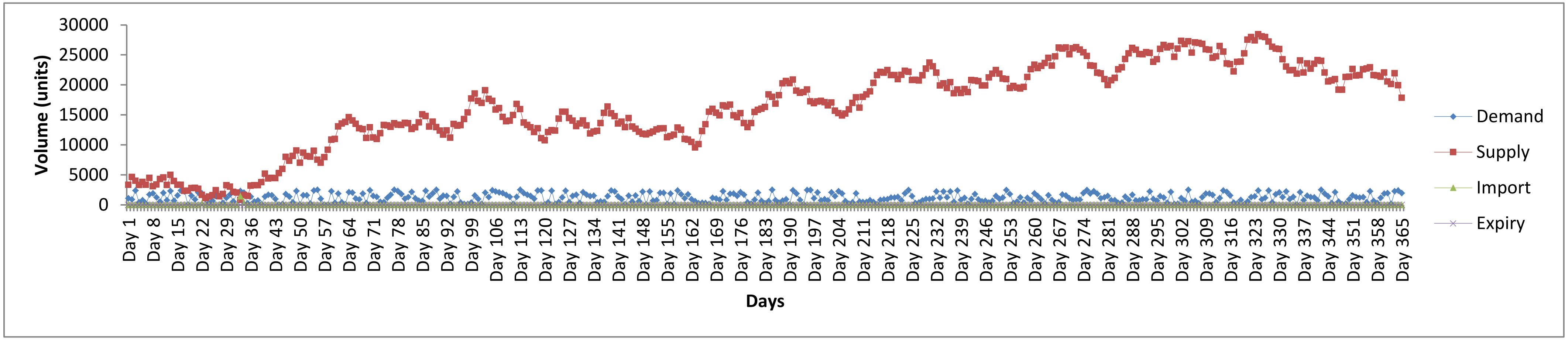

Figure 18: Represents a line graph over a period of 365 days for the GA implementation of dataset 5

Figure 19

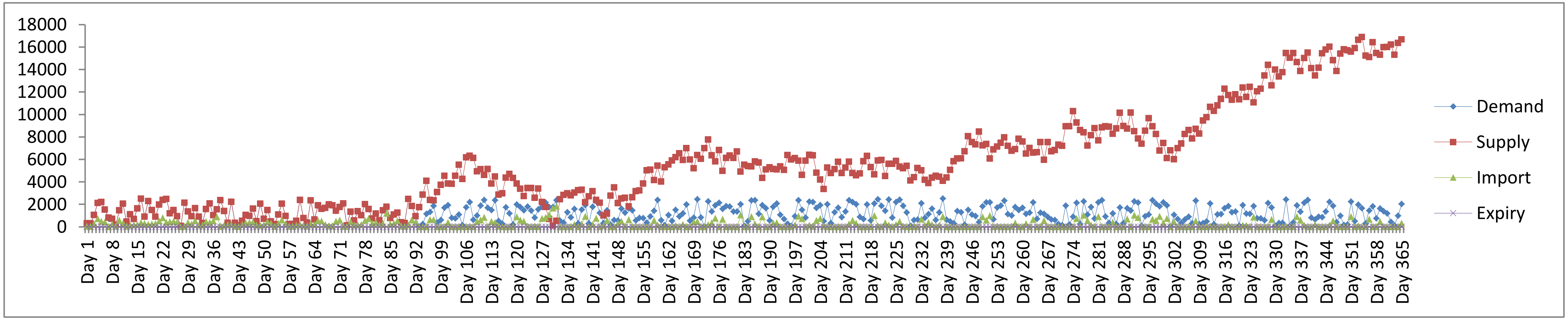

Figure 19: Represents a line graph over a period of 365 days for the PSO implementation of dataset 5

Figure 20

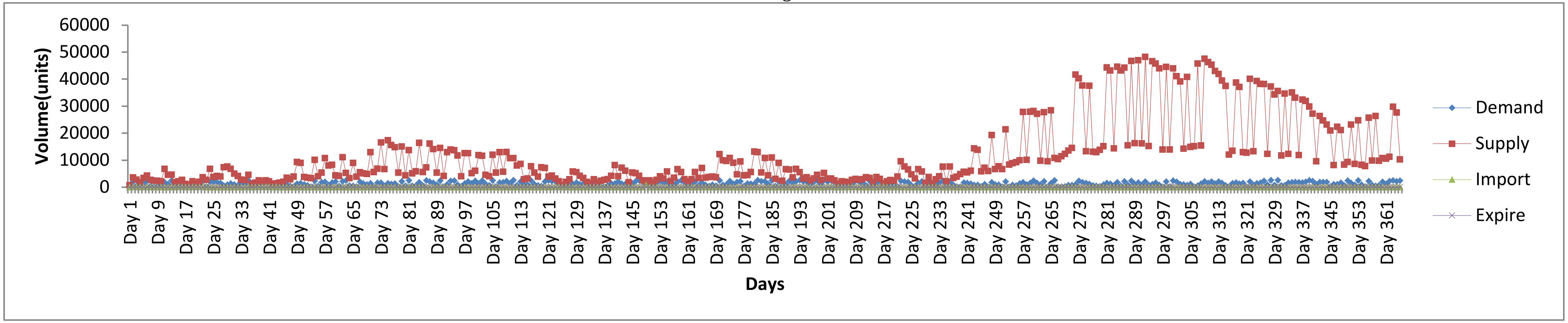

Figure 20: Represents a line graph over a period of 365 days for the DA implementation of dataset 5

Figure 21

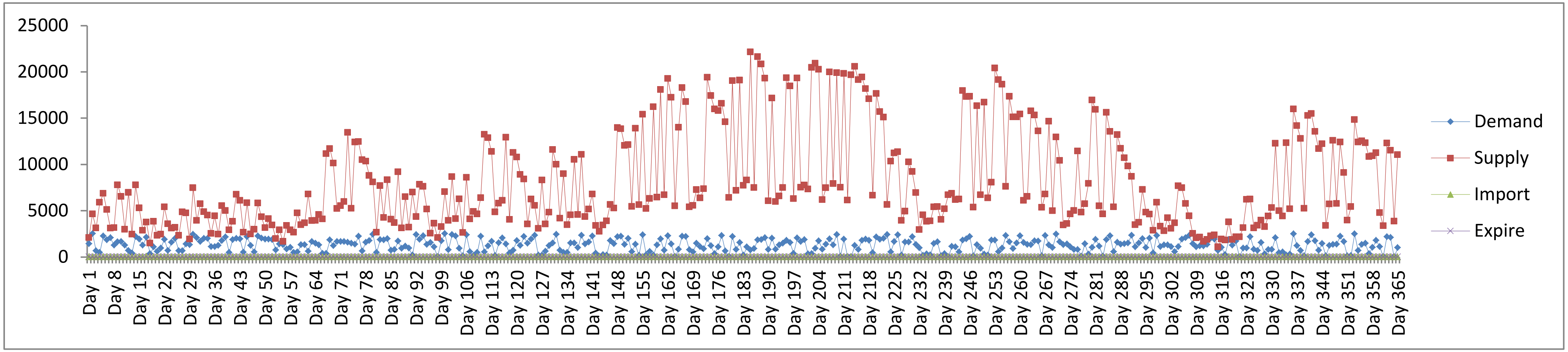

Figure 21: Represents a line graph over a period of 365 days for the SOS implementation of dataset 5

Figure 22

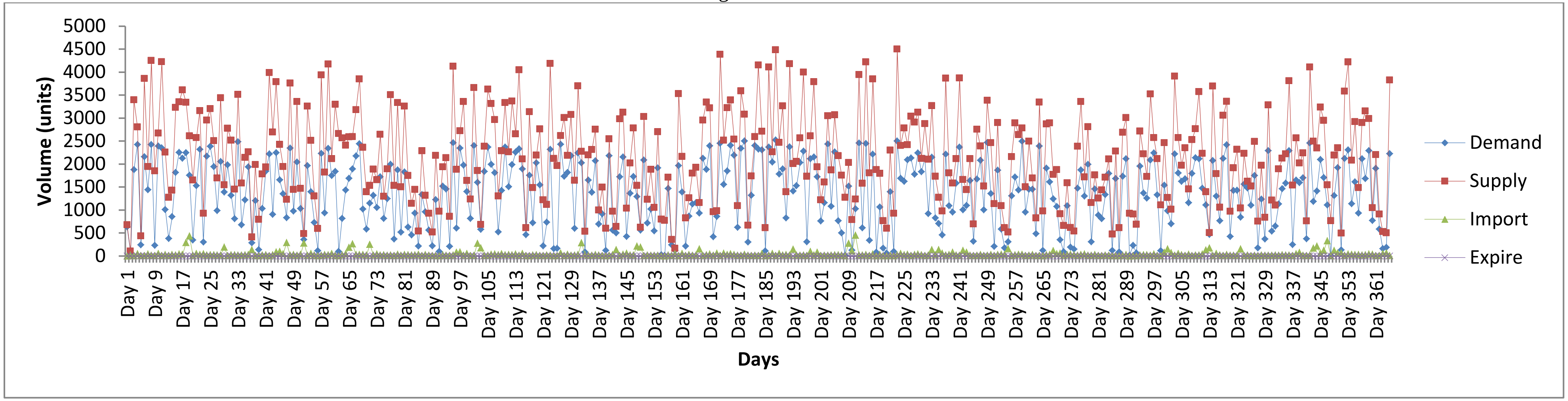

Figure 22: Represents a line graph over a period of 365 days for the GWO implementation of dataset 5

Dataset 5 used percentage bounds similar to that of dataset 1, however it also used a much larger initial blood volume of 5000 units. The results show that all implementations reached a day where stock-piling occurs except the GWO. The GWO failed to stock-pile based on the fact that it tries to find the best solutions for a daily basis. To elaborate further the GWO may satisfy the day's requests for a specific day without any remainder, however the next day may need to import additional units if an adequate solution cannot be found. The GA begun stock-piling around day 35 which is relatively fast, however it still experienced importation of lower proportion blood types at specific periods, whilst the PSO begun stock-piling around day 92, but experienced the largest amounts of importation throughout the time period. DA took the longest to begin stock-piling, but did not experience any importation after this event occurred. SOS performed the best with majority of the days receiving an adequate supply to meet the demand and store for later usage, in addition the only blood type which experienced importation for the SOS algorithm was AB^+. These results indicate that the SOS algorithm can manage a higher volume of blood much more efficiently.

(Still need to add in the remaining results and conclusion)

References

[1]: Hesse. S, Coullard. C, Daskin. M, Hurter. A, (2005), A case study in platelet inventory management, Evanston, Illinois

[2]: Adewumi. A, Budlender. N, Olusanya. M, (2012), Optimizing the assignment of blood in a blood banking system: Some initial results, University of KwaZulu Natal, School of Mathematics, statistics and computer science, Durban, South Africa.

[3]: South African National Blood Service, What's your type, http://www.sanbs.org.za, [Accessed on 17/05/2018].

[4]: Olusanya. O, Arasomwan. MA, Adewumi AO, (2014), Particle swarm optimization algorithm for optimizing assignment of blood in blood banking system, University of KwaZulu Natal, School of Mathematics, statistics and computer science, Durban, South Africa.

[5 NEW] Reid. M.E, Lomas-Francis. C, (2004), The blood group antigen Fac-Book, (3rd ed), Academic Press

[6]: Charpin J.P.F, Adewumi. A.O, (2011), Blood Assignment, University of KwaZulu Natal, School of Mathematics, statistics and computer science, Durban, South Africa.

[7]: Igwe. K, Olusanya. M, Adewumi. A.O, (2015), On the performance of GRASP and dynamic programming for the blood assignment problem, University of KwaZulu Natal, School of Mathematics, statistics and computer science, Durban, South Africa.

[8]: Sapountzis C, (1989), Allocating blood to hospitals, Pireas graduate school of industrial studies, Greece

[9]: Lewis. R, (1992), Life, New York: State University of New York, C. Brown publishers

[10]: https://www.news24.com/SouthAfrica/News/easter-road-death-toll-up-by-14-from-last-year-20180417, visited: [04-06-2018]

[11]: http://www.schoolterms.co.za/2018.html, visited: [04-06-2018]

[12]: Ragalo. A, Pillay. N, (2017), An investigation of dynamic fitness measures for genetic programming, Expert systems with applications, 52-72.

[13]: Biyanto. T.R, Fibrianto H.Y, Nugroho G, Listijorini E, Budiati T, Huda H, (2016), Duelist Algorithm: An algorithm by how duelist improve their capabilities in a duel, Indonesia.

[14]: Nama. S, Apu K.S, Ghosh S, (2016), Improved symbiotic organisms search algorithm for solving unconstrained function optimization, Department of Mathematics, National Institute of Technology Agartala, Tripura, India.

[15]: Mirjalili S, Mirjalili S.M, Lewis. A, Grey Wolf Optimizer, (2014) School of Information and Communication Technology, Griffith University, Nathan, Brisbane, Australia and Department of Electrical Engineering, Faculty of Electrical, and Computer Engineering, Shahid Beheshti University, Tehran, Iran

[16]: Shoghian. S, Maryam. K, (2012), A Comparison among Wolf Pack Search and Four other Optimization Algorithms, faculty of Electrical Engineering Control Department University of Tabriz , Iran.