A Comparative Study Of Metaheuristic Approaches For The Blood Assignment Problem

Prinolan Govender, Absalom Ezugwu

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

Durban 4000, South Africa

**Abstract:** The blood assignment problem is an important real world optimization problem because of the continuous demand for blood transfusion during medical emergencies. However, the formulation of this problem faces a lot of challenges that stretches from manging critical blood shortages, limited shelf life or blood expiration, to the natural blood grouping that tends to incorporates additional constraints on the type of blood to be transfused to a patient. In addition, difficulty can also arise from blood banks not being able to meet their daily demands and blood products requiring importation from external sources. These challenges have serious consequences especially in the case where the demand for blood is very high. Therefore, there is the need to minimize blood product wastage with regard to expiration and importation, whilst maximising product delivery to patients in need. In this paper, five popular metaheuristic algorithms where employed to effectively minimize the operational costs of the blood transfusion centres. The computational results indicate that all the four proposed algorithms performed quite satisfactory from computational time, stock-piling and low importation level points of view.

.

**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. These blood components can be harvested from a single donation and be used for different medical cases [2]. The following study focuses on whole blood (WB) units which relates to all components of blood. In accordance to the ABO blood system (system which classifies blood based on the properties of red blood cells) [3], there 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 [4].

The demand for WB units can be exemplified using the following two main scenarios. The first scenario involves a predetermined event such as a typical surgery which could be scheduled days before in the case of hospital admission. The scheduling process will also allow for sufficient WB units to be set aside (or imported if there is an insufficient amount available in the blood bank). The second scenario relates to the demand for WB units that cannot be foreseen or that was not initially planned for as in the case of medical emergencies such road accidents or natural disasters. This can further be demonstrated by an individual who is exposed to sudden onsets of trauma, which is in need of immediate attention or WB unit’s transfusion. Blood preservation for onset supply in the case of unexpected demand can be a daunting task because blood product is considered a perishable commodity due to its limited shelf life, coupled with the complexity of blood compatibility and the stochastic nature of daily blood demand and supply. Therefore, it is necessary to devise more efficient and effective ways in which blood products can be stored and assigned to individuals in need.

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. The BAP is said to be an NP-hard problem [5]. The BAP is comprised of a plethora of components which becomes difficult to mathematically model, such components involve dealing with inadequate supply to meet daily demand, importing additional units, cross-matching blood, and expiring WB units once a unit has exceeded its shelf-life. It must be remembered that blood products must be voluntarily donated to blood banks, therefore there is no constant levels of supply, whilst demand for WB units occurs every day. The BAP is therefore studied with the effort to develop an adequate model which utilize WB units more efficiently and achieve its objective function.

Research relating to the BAP is relatively scarce and only a handful of related literature exists. However, the few studies which were conducted under the BAP generally follow similar scope of applying existing metaheuristic algorithms to solve the problem at hand. In this study, five different well-known nature inspired metaheuristic algorithms as well as a robust blood bank assignment policy are implemented to solve the BAP with the main goal of reducing cost and time of blood component distributions. In addition, the proposed blood assignment method generally seeks to minimize wastage of blood products by efficiently assigning blood to patient and preserving blood stock pile by allocating available blood to different blood types. A report conducted in Estonia, reviewed the various techniques and aspects of blood donations, and the costs associated with these methods [6]. This study will ignore any specific costs related to the BAP, as by minimizing the objective function, the blood bank would be deemed as optimal and efficient.

The method implemented in this study tries to reduce the randomness when generating datasets. Studies conducted in [3], [5] and [7] used fixed percentage bounds to generate values for demand and supply. However, in this paper a different approach was used when generating such values, by allocating a unique percentage bound to each month. The bounds conform to statistics taken from South African social behaviour with the attempt of generating values which mimic real-life monthly demand and supply for WB units. Therefore, to confront the problem at hand, this study employs the following metaheuristic algorithms namely, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Duelling Algorithm (DA), Grey Wolf Optimizer (GWO) and Symbiotic Organism Search (SOS) algorithm to solve the BAP. A metaheuristic is a problem-independent algorithmic framework which provides a set of strategies and guidelines which aid in developing heuristic optimization algorithms [8].

The remainder of this paper is organized as follows: Section 2 briefly review some of the existing literatures on BAP. Section 3 presents the BAP problem description and formulation. Section 4 presents the proposed metaheuristic algorithms and the new BAP assignment policy. Section 5 discusses the experimental configuration and computational results. Finally, concluding remarks and future directions are presented in Section 6.

**2. Related Work**

Studies conducted in the BAP topic have used different approaches and metaheuristics to try and optimize the assignment of blood. The transfusion of blood or blood related products is a daily activity which occurs in most hospitals and clinics around the world. The complexity associated with the understanding blood compatibility and transfusion using any blood type would often result in blood clumping and other negative effects upon the patient. Antigens A and B for blood types where on discovered in 1927 later type O was established resulting in the widely known ABO blood group system [9]. Bas [10] analysed another aspect of the blood banking system which focused on the donating process of blood. The study looked into donors, collection and screening of acquired blood units as well as finding a way of supplying an adequate supply of blood units to transfusion centres. A unique implementation was conducted by [11], who developed a web-based application for managing the information of blood donors and blood stock. Charpin [12] introduced a linear model for the blood management problem, 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 its daily demand. Many factors contribute towards the assignment of blood to patients, some of which includes request time, urgency for the request, compatibility of blood types and quantity of blood [13]. The management of blood can therefore be considered as a very diverse topic with different areas of consideration. 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.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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

The BAP has generated unique potential solutions which take into account external factors that contribute towards the assignment of blood. As such similar research topics relating to blood inventory have been conducted. In Kaveh’s [14] study of the blood banking supply chain allocation was considered. In this existing work, metaheuristic optimization, and graph partitioning were used to minimize the total distance between blood bank(s) and hospitals. The results from [14] showed pleasing results with regards to optimal points of view and computational time. Another study that dealt with the blood bank assignment problem was conducted by Sahin [15], who implemented several mathematical models to solve the location-allocation decision problems for blood banking services. In [9], real-world benchmark was used and the computational results showed that the proposed method was able to achieve successful results in resolving the management difficulties. Similarly, Sapountiz [16] 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 to address the BAP.

In recent years, studies conducted for the BAP by Adewumi [3], Olusanya [5] and Igwe [7] followed the same assignment patterns in terms of proposing optimization models for the blood assignment problem. The usage of a Multiple Knapsack model enabled cross-matching blood between compatible blood types, along with the Bottom-up technique to pull additional units of blood from compatible blood types. In [3], the implementation of 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 was considered. Results from the study showed that all implementations successfully achieved optimal fitness, with HC performing the best. The study conducted by [7] 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 method implemented in this study tries to reduce the randomness when generating datasets. Studies conducted by [3], [5] and [7] used fixed percentage bounds to generate values for demand and supply. However, the current study presented in this paper considered a different approach when generating such values, by allocating a unique percentage bounds to each month. The bounds conform to statistics taken from South African social behaviour with the attempt of generating values which mimic real-life monthly demand and supply for WB units. To confront the problem, this paper develops in five well-known global optimisation metaheuristic algorithms as well as a robust blood banking policy which tries to satisfy the main objective function of the BAP of minimising blood product wastage. The metaheuristics include, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Duelling Algorithm (DA), Grey Wolf Optimizer (GWO) and Symbiotic Organism Search (SOS).

**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 56 million citizens currently living in South Africa [17]. The Pie chart labelled as “Fig 1” indicates the current percentage of races within the population.

Fig 1: Pie chart representation of the current race group proportions in South Africa adapted from [4]

Fig 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 [18], 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.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 [5]

**3. Problem Description**

The demand for blood in a day must be met. If the demand exceeds the current supply at hand, then 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 at 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 are any remaining units 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

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

Due to the plethora of external factors which encumber the BAP, certain assumptions had to be introduced in order to formulate the required mathematical model that would be adequate for the problem at hand. These assumptions are as follows:

1. The blood bank has an infinite supply of capital, and storage space. The external sources (for importing WB units) also have a limitless supply.
2. The time frame will be conducted over 365 days, with day 1 receiving no carryover of WB units from the previous day.
3. Validity of blood will be set at 30 days.
4. 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.

In Fig. 2, the flow diagram represents the daily process that occurs within the blood bank as modelled in this study, with each block representing the various potential activities that can be expected in the BAP.

Update supply for the day

Are there traces of expired WB units?

Yes

No

Remove expired units from blood bank, and update current useable supply

Is Demand greater than supply?

Pull additional units from the remainder of compatible blood types.

Yes

Import additional WB units from external sources to meet the demand.

Is Demand greater than updated supply?

Yes

Fulfil demand for the day

Calculate remainder for the day.

Start

No

Is day less than or equal to time period

Yes

End

No

No

Fig. 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 follows 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 metaheuristic 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 algorithms, 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 global optimisation implementations, the blood compatibility process, expiring old WB units, and 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

)*d* (1)

Where:

1 ≤ d ≤ 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 [19], 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%.

|  |  |  |
| --- | --- | --- |
| 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 [20].

|  |  |
| --- | --- |
| 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

|  |  |
| --- | --- |
| 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 Bu: Represent the upper percentage bound Bl: Represent the lower percentage bound

(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

(3)

Where d ≥ 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

**Begin**

**If** d > 30 Then

**for** integer i < 30 Sumd += Dd **End for loop**

**If** Sumd < (Sb)d-30 ThenE = ((Sb)d-30) – Sumd//supply on d-30 – Sumd over 30 days equates to expiry

**End if**

**End if End**

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.

|  |  |  |
| --- | --- | --- |
| Blood type | Can distribute to | Splitting |
| A+ | A-, O+, O- |  |
| A- | O- |  |
| B+ | B-, O+, O- |  |
| B- | O- |  |
| AB+ | A+, A-, B+, B-AB-, O+, O- |  |
| AB- | A-, B-, O- |  |
| O+ | O- |  |
| O- | 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.

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

Fig. 3: Representation of an individual used in the metaheuristic implementations

Each segment from fig. 3 stores a value which is calculated using the blood proportion percentage found in the South African population. By storing each value in a separate location, this allows us to manipulate the individual in later calculations.

**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 [21]. 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. A study conducted by [22] classified the GA process into 5 major components, namely: fitness function, population, selection, crossover and mutation. These 5 components were also incorporated in this study of the BAP. The structure of the individual (fig. 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.

Below is fig. 4 which represents the mechanics for the crossing over algorithm used within GA, to promote diversity within a population. Note that the highlighted segments from individuals 1 and 2 swap positions with each other, and this results in 2 newer offspring being created.

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 |

Fig. 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

Below is fig.5 which depicts the effect of mutation on an individual. The highlighted segments are positions that will be re-calculated randomly in order to produce a new individual.

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 |

Fig. 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 [5]. The PSO algorithm mimics behavioural patterns from animal groups that don’t have a specific leader, such as a flock of birds [23]. 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 Xi = (Xi1 . . . Xin) as well as a velocity vector Vi = (Vi1 . . . Vin). Both Xi and Vi make use of the following equations to iteratively update themselves.

(4)

(5)

Where Pi: Represents the best personal best position Pg: Represents the global best position r1, r2: 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 c1 and c2 exert random forces in the direction of both Pi and Pg, 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 Xi, Pi and Pg.

**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 [24]. 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**

Algorithm 2 was adapted from [24], and illustrates the process of establishing a winner and loser when conducting a duel between two individuals. Each duellist utilizes a luck coefficient which aids them during a duel.

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.

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 |

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

Fig. 6 represents illustrations of the individuals before and after training occurs. The individuals try to improve their initial representation in hopes of attaining a better solution. In fig. 6 the winner learns from the loser and vice versa, whilst taking into account the demand for a day. Both individuals try to attain a solution which best matches (or comes close) to the demand value for each segment.

**4.9. Symbiotic Organism Search**

The Symbiotic organism search (SOS) algorithm simulates the interactive behaviour of organisms within nature [16]. A notable advantage of the SOS algorithm is that it does not require any specific parameter tuning [26]. 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 Xi and Xj represent 2 random individuals within a population, and MV represent the Mutual Vector.

Xinew = Xi + rand(0, 1) ∙ (Xbest – MV ∙ BF1) (6)

Xjnew = Xj + rand(0, 1) ∙ (Xbest – MV ∙ BF2) (7)

Where:

MV = (Xi + Xj) / 2 (8)

The value obtained from (Xbest – 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 Xi and the inferior individual is labelled as Xj.

Xinew = Xi + rand(-1, 1) ∙ (Xbest - Xj) (9)

Xi benefits from Xj by means of (Xbest - Xj) (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 **Begin**

**for** (i to parasite length) **if** (parasite [i] is not equal to 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**

Algorithm 3 illustrates the parasitism process conducted in this study of the BAP. The parasite analyses the host and swaps segments if the host contains a better value than itself. The values for both the host and parasite are matched against the demand for the day (per blood type) in order to deduce the better value.

**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) [27]. 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 Xi = (Xi1, Xi2 … XiD) with Xid 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 [28] 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.



Fig. 7 represents 5 images: (A) Tracking prey, (B-D) pursuing, harassing and encircling and (E) situation before attack [27]

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.

= 2 . (11)

= 2 . . (12)

Where is linearly decreased over a set number of iterations from 2 to 0. Whilst, represent random vectors between 0 and 1

= | . p (t) – (t) | (13)

(t+1) = p(t) – . (14)

Where t indicates the iteration, p is the position vector of the prey and 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.

α = | 1 . α -| (15)

β = | 2 . β - | (16)

δ = | 3 . δ - | (17)

1 = α – 1 (α) (18)

2 = β – 1 (β) (19)

3 = δ – 1 (δ) (20)

= (1 + 2 + 3) / 3 (21)

4.10.4 **Exploitation** **(Attacking process)**

By decreasing the value of over a set number of iterations, this mimics the process of a wolf approaching the prey. According to eq. (12), is a component in calculating which in turn decreases. The values for lies between [-1, 1] with each position of the search agent lying between this specified range, if 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 bounded between [-1, 1] this allows for global exploration to take place. Exploration is also favoured by component which contains random values between [0, 2]. Component 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 being linearly decreased from 2 to 0 over a number of iterations. Candidate solutions tend to converge towards the prey when || < 1 and diverge when || > 1. The termination criteria terminates when the max number of iterations have been reached, or the supply equates to the demand for a day. The reduced amount of search parameters of the GWO implementation is an important advantage of the algorithm [29].

**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, c1 = c2= 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.

1. Running time for each MT
2. Average amounts for importation and expiration
3. 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.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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.

|  |  |  |
| --- | --- | --- |
| 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

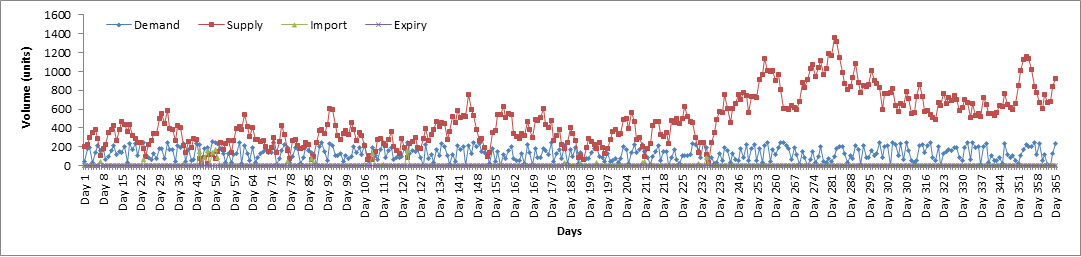


Fig. 8: Represents a line graph over a period of 365 days for the GA implementation of dataset 1

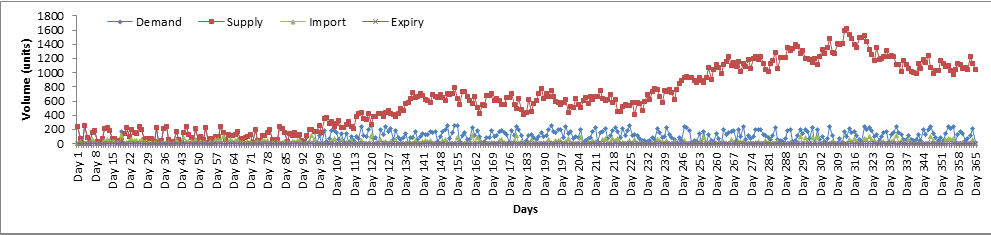


Fig. 9: Represents a line graph over a period of 365 days for the PSO implementation of dataset 1

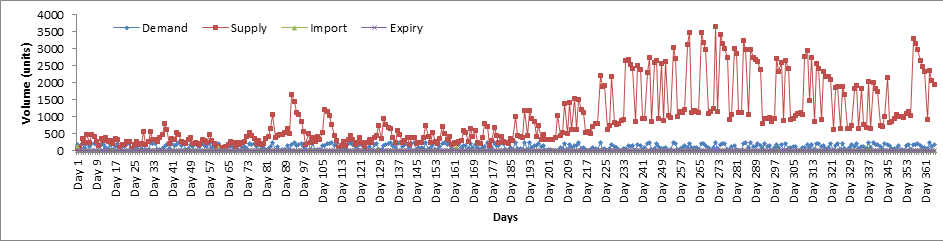


Fig. 10: Represents a line graph over a period of 365 days for the DA implementation of dataset 1

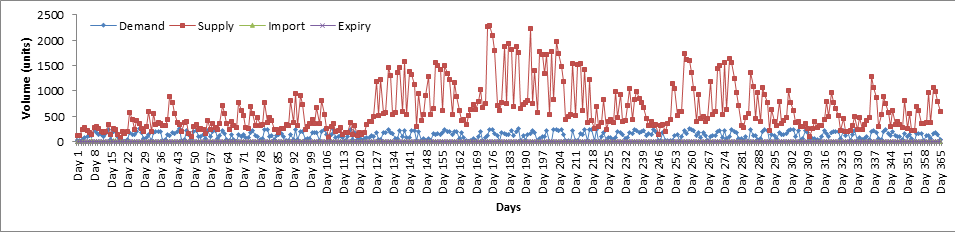


Fig. 11: Represents a line graph over a period of 365 days for the SOS implementation of dataset 1

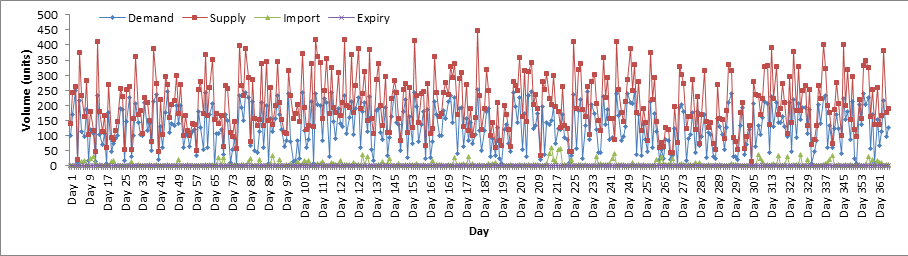
****

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

**Evaluating results from dataset 1**

Fig. 8-12 represents line graphs obtained when each metaheuristic algorithm was subjected to 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.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MT** | **Variable** | **A+** | **A-** | **B+** | **B-** | **AB+** | **AB-** | **O+** | **O-** |
| **GA** | **Supply** | 157.87 | 26.43 | 59.48 | 20.51 | 13.08 | 9.35 | 192.73 | 39.42 |
|  | **Demand** | 19.67 | 3.07 | 7.38 | 1.23 | 1.84 | 0.61 | 23.98 | 4.30 |
|  | **Import** | 0.07 | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 | 0.09 | 0.00 |
|  | **Expiry** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **PSO** | **Supply** | 278.14 | 3.57 | 101.07 | 1.79 | 24.84 | 10.51 | 25.50 | 4.98 |
|  | **Demand** | 19.28 | 3.01 | 7.23 | 1.20 | 1.81 | 0.60 | 23.50 | 4.22 |
|  | **Import** | 1.29 | 0.71 | 0.09 | 0.31 | 0.23 | 0.04 | 5.60 | 0.96 |
|  | **Expiry** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **DA** | **Supply** | 27.40 | 8.53 | 10.29 | 3.27 | 1.90 | 1.07 | 35.90 | 9.99 |
|  | **Demand** | 20.01 | 3.13 | 7.51 | 1.25 | 1.88 | 0.63 | 24.39 | 4.38 |
|  | **Import** | 2.61 | 0.09 | 0.81 | 0.03 | 0.60 | 0.12 | 1.38 | 0.13 |
|  | **Expiry** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **SOS** | **Supply** | 222.87 | 43.48 | 76.14 | 24.06 | 19.19 | 9.34 | 294.67 | 52.15 |
|  | **Demand** | 19.08 | 2.98 | 7.15 | 1.19 | 1.79 | 0.60 | 23.25 | 4.17 |
|  | **Import** | 0.11 | 0.00 | 0.03 | 0.00 | 0.01 | 0.00 | 0.08 | 0.00 |
|  | **Expiry** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **GWO** | **Supply** | 29.48 | 15.30 | 10.10 | 3.90 | 0.85 | 0.86 | 35.87 | 7.73 |
|  | **Demand** | 19.96 | 3.12 | 7.48 | 1.25 | 1.87 | 0.62 | 24.33 | 4.37 |
|  | **Import** | 0.72 | 0.01 | 0.52 | 0.05 | 1.02 | 0.17 | 0.57 | 0.00 |
|  | **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 2 for each blood group measured in units.

|  |  |  |
| --- | --- | --- |
| Metaheuristic | Time (Ms) | Time(Minutes) |
| GA | 3983352 | 66.38 |
| PSO | 561452 | 9.35 |
| DA | 4113223 | 68.55 |
| SOS | 4665054 | 77.75 |
| GWO | 4525144 | 75.41 |

Table 11: Represents the running time per metaheuristic for dataset 2

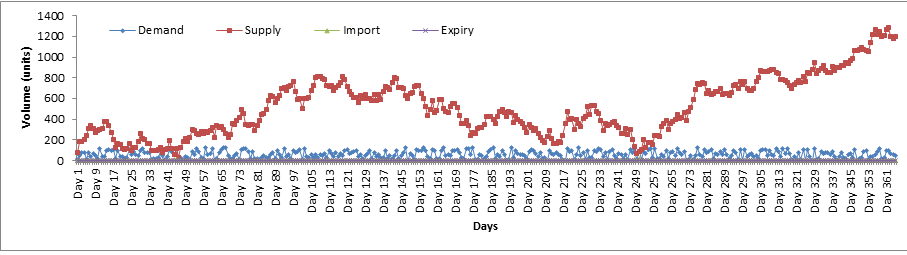


Fig. 13: Represents a line graph over a period of 365 days for the GA implementation of dataset 2

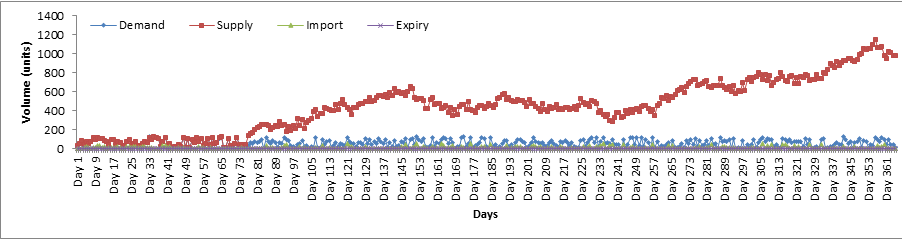


Fig. 14: Represents a line graph over a period of 365 days for the PSO implementation of dataset 2

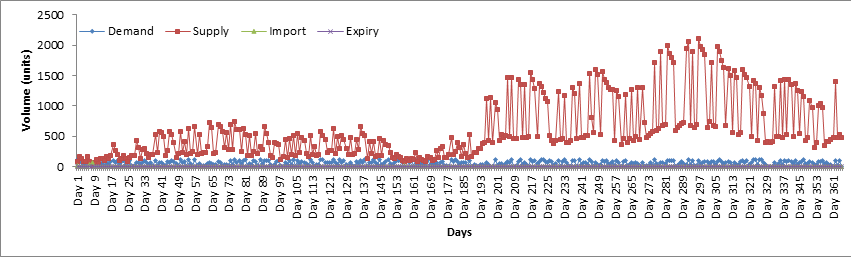


Fig.15: Represents a line graph over a period of 365 days for the DA implementation of dataset 2

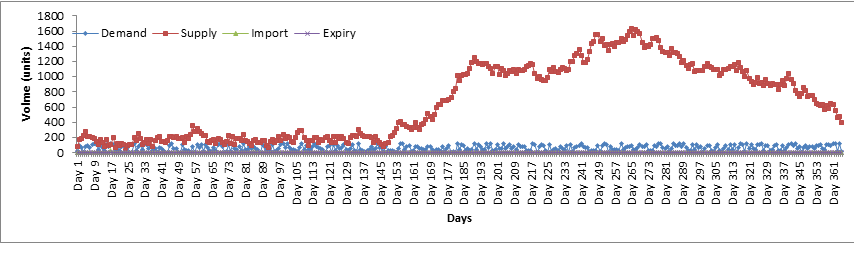


Fig. 16: Represents a line graph over a period of 365 days for the SOS implementation of dataset 2

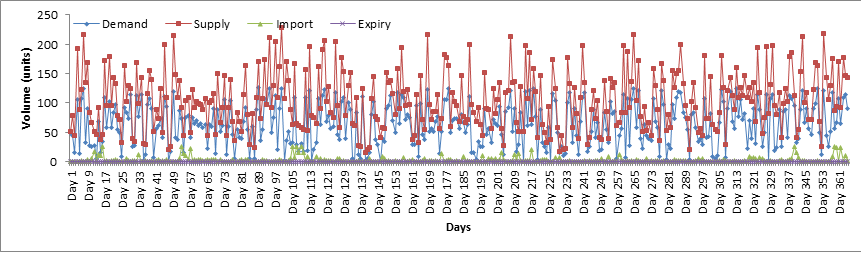


Fig. 17: Represents a line graph over a period of 365 days for the GWO implementation of dataset 2

**Evaluating results from dataset 2**

Fig. 13-17 represents line graphs obtained when each metaheuristic algorithm was subjected to dataset 2.Dataset 2 has been established to test the implications of SAGV. These values are generated by allocating specific percentage ranges (table 5) to each month in relation to monthly schooling terms and public holidays. The graph curvatures (fig.13-17) follow similar trends as compared to the results achieved from dataset 1. The results achieved indicate a lower value for the average demand generated, which is expected as certain months have much lower percentage bounds as compared to other months. The comparison between SAGV demand and the set percentage bounds used in previous research will be further discussed in section 5.1.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 12: Average results achieved for each metaheuristic implementation subjected to dataset 3 for each blood group measured in units.

|  |  |  |
| --- | --- | --- |
| 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 13: Running time per metaheuristic for dataset 3

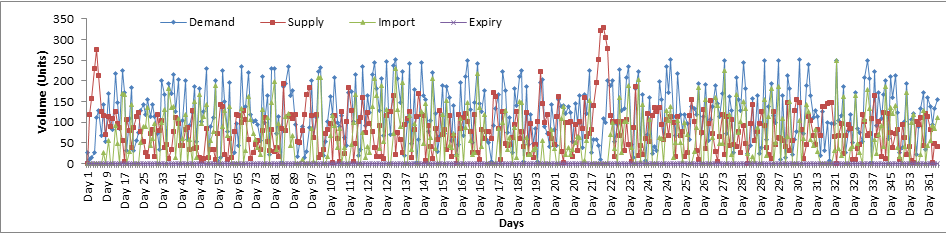


Fig. 18: Represents a line graph over a period of 365 days for the GA implementation of dataset 3

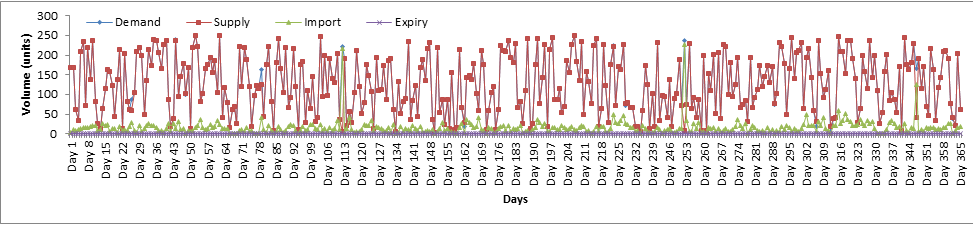


Fig. 19: Represents a line graph over a period of 365 days for the PSO implementation of dataset 3

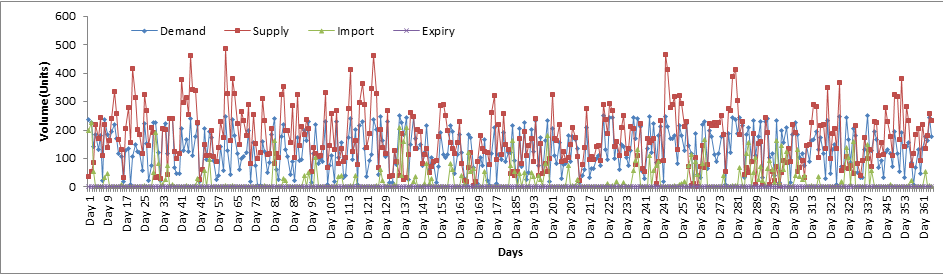


Fig. 20: Represents a line graph over a period of 365 days for the DA implementation of dataset 3

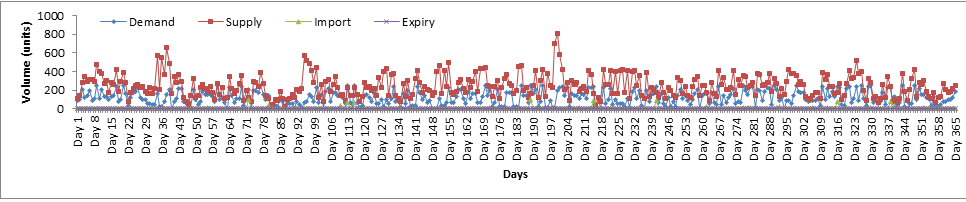


Fig. 21: Represents a line graph over a period of 365 days for the SOS implementation of dataset 3

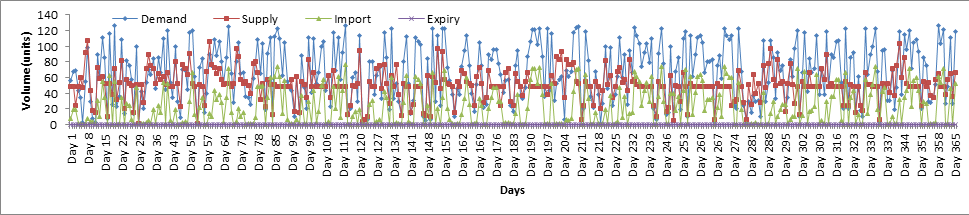


Fig. 22: Represents a line graph over a period of 365 days for the GWO implementation of dataset 3

**Evaluating results from dataset 3**

Fig. 18-22 represents line graphs obtained when each metaheuristic algorithm was subjected to 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 fig. 18-22 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.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MT | Variable | A+ | A- | B+ | B- | AB+ | AB- | O+ | O- |
| GA | Supply | 2124.89 | 335.79 | 791.17 | 152.07 | 195.07 | 66.28 | 2583.42 | 469.82 |
|  | Demand | 20.62 | 3.22 | 7.73 | 1.29 | 1.93 | 0.64 | 25.13 | 4.51 |
|  | Import | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|  | Expiry | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| PSO | Supply | 3584.50 | 6.29 | 1350.75 | 2.52 | 328.10 | 100.46 | 49.15 | 8.80 |
|  | Demand | 20.65 | 3.23 | 7.74 | 1.29 | 1.94 | 0.65 | 25.17 | 4.52 |
|  | Import | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|  | Expiry | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| DA | Supply | 1893.13 | 470.49 | 715.33 | 169.36 | 175.23 | 56.17 | 506.27 | 461.14 |
|  | Demand | 19.71 | 3.08 | 7.39 | 1.23 | 1.85 | 0.62 | 24.02 | 4.31 |
|  | Import | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|  | Expiry | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| SOS | Supply | 3419.58 | 565.67 | 1308.53 | 209.98 | 326.20 | 98.90 | 4172.27 | 746.18 |
|  | Demand | 18.84 | 2.94 | 7.07 | 1.18 | 1.77 | 0.59 | 22.96 | 4.12 |
|  | Import | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 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 | 3977.10 | 617.69 | 1498.53 | 257.23 | 363.27 | 118.53 | 4843.29 | 868.76 |
|  | Demand | 20.31 | 3.17 | 7.61 | 1.27 | 1.90 | 0.63 | 24.75 | 4.44 |
|  | Import | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|  | Expiry | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

Table 14: Average results achieved for each metaheuristic implementation subjected to dataset 4 for each blood group measured in units.

|  |  |  |
| --- | --- | --- |
| Metaheuristic | Time (Ms) | Time(Minutes) |
| GA | 5220157 | 87.00 |
| PSO | 500968 | 8.34 |
| DA | 5135744 | 85.60 |
| SOS | 5978164 | 99.63 |
| GWO | 5998772 | 99.97 |

Table 15: Running time per metaheuristic for dataset 4

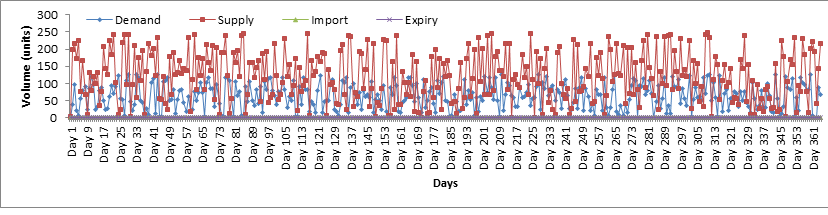


Fig. 23: Represents a line graph over a period of 365 days for the GA implementation of dataset 4

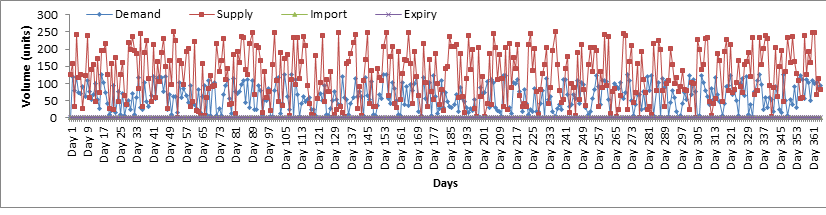


Fig. 24: Represents a line graph over a period of 365 days for the PSO implementation of dataset 4

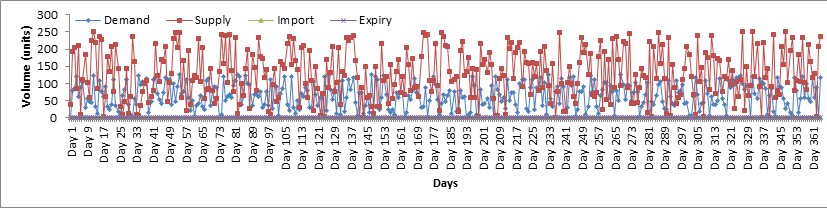


Fig. 25: Represents a line graph over a period of 365 days for the PSO implementation of dataset 4

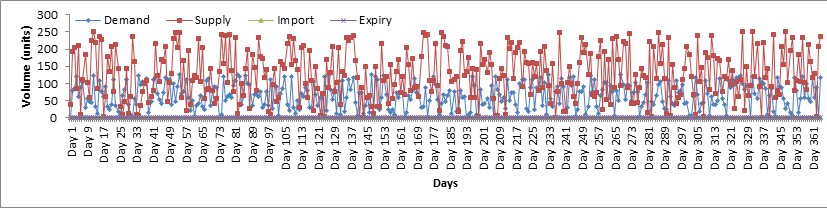


Fig. 26: Represents a line graph over a period of 365 days for the SOS implementation of dataset 4

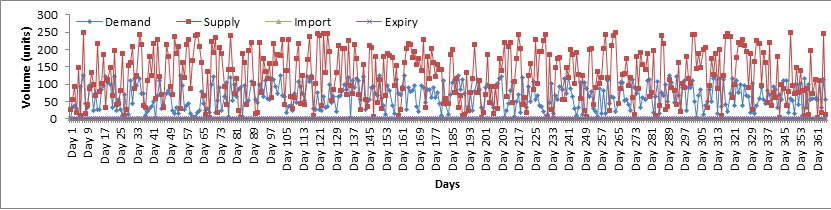


Fig. 27: Represents a line graph over a period of 365 days for the GWO implementation of dataset 4

**Evaluating results from dataset 4**

Fig. 23-27 represents line graphs obtained when each metaheuristic algorithm was subjected to dataset 4. Dataset 4 was implemented to test a situation within a blood bank were daily supply heavily exceeds daily demand. This tests how well the implementation policy deals with this situation, it would be expected for expiry to occur, however this did not occur. The lack of expiry is due to the FIFO issuing policy coupled with the WB shelf life of 30 days, it is therefore unlikely for a certain amount of WB units to be within storage for a period longer than 30 days. None of the MT implementations being subjected to dataset 4 experienced expiry or importation, therefore it can be established that the PSO system offers the best results when we compare the running times.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

Table 16: Average results achieved for each metaheuristic implementation subjected to dataset 5 for each blood group measured in units.

|  |  |  |
| --- | --- | --- |
| 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 |

Table 17: Running time per metaheuristic for dataset 5

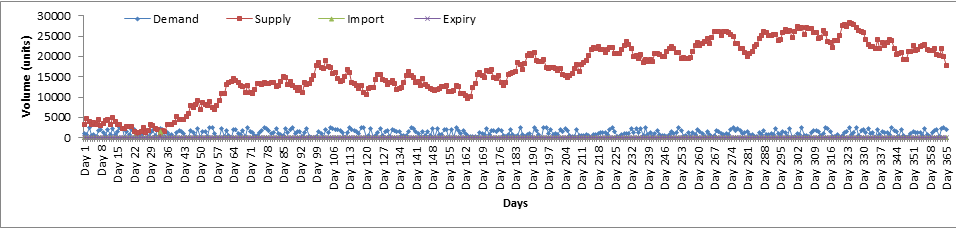


Fig. 28: Represents a line graph over a period of 365 days for the GA implementation of dataset 5

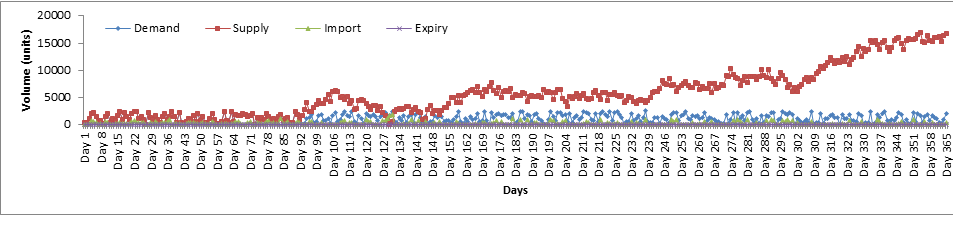


Fig. 29: Represents a line graph over a period of 365 days for the PSO implementation of dataset 5

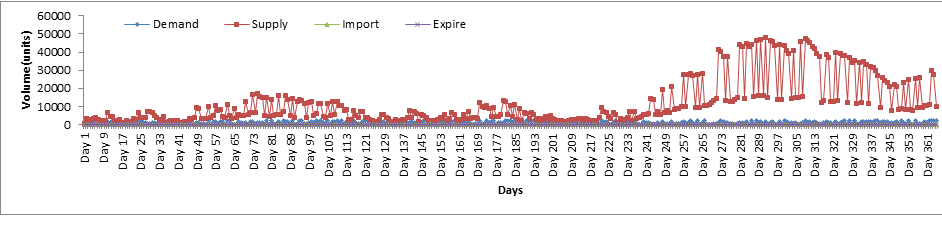


Fig. 30: Represents a line graph over a period of 365 days for the DA implementation of dataset 5

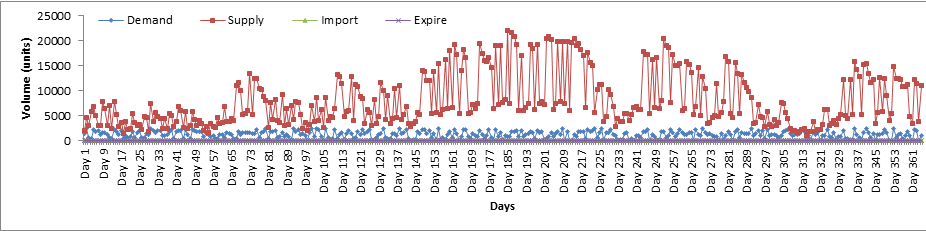


Fig. 31: Represents a line graph over a period of 365 days for the SOS implementation of dataset 5

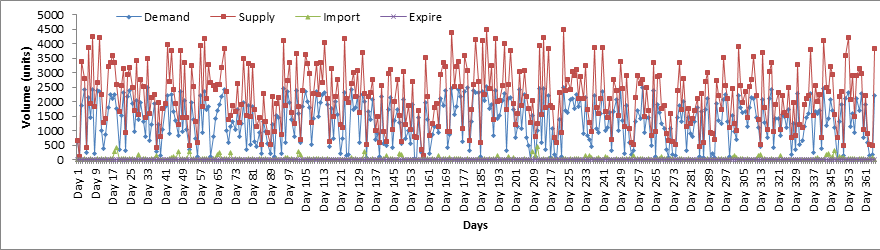
****

Fig. 32: Represents a line graph over a period of 365 days for the GWO implementation of dataset 5

**Evaluating results from dataset 5**

Fig. 28-32 represents line graphs obtained when each metaheuristic algorithm was subjected to 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.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MT** | **Variable** | **A+** | **A-** | **B+** | **B-** | **AB+** | **AB-** | **O+** | **O-** |
| **GA** | **Supply** | 17293.71 | 2700.27 | 6485.94 | 1082.13 | 1619.56 | 536.04 | 21076.53 | 3779.62 |
|  | **Demand** | 345.93 | 54.05 | 129.72 | 21.62 | 32.43 | 10.81 | 421.60 | 75.67 |
|  | **Import** | 0.73 | 0.11 | 0.28 | 0.05 | 0.07 | 0.03 | 0.90 | 0.16 |
|  | **Expiry** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **PSO** | **Supply** | 10808.48 | 59.98 | 4047.42 | 24.02 | 1011.76 | 334.79 | 467.84 | 83.99 |
|  | **Demand** | 351.56 | 54.93 | 131.84 | 21.97 | 32.96 | 10.99 | 428.47 | 76.90 |
|  | **Import** | 0.29 | 17.44 | 0.11 | 6.96 | 0.03 | 0.01 | 136.02 | 24.41 |
|  | **Expiry** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **DA** | **Supply** | 15915.12 | 7070.80 | 5962.89 | 2671.17 | 1466.52 | 489.37 | 7503.44 | 7096.10 |
|  | **Demand** | 344.48 | 53.82 | 129.18 | 21.53 | 32.29 | 10.76 | 419.83 | 75.35 |
|  | **Import** | 0.90 | 0.00 | 0.24 | 0.00 | 0.29 | 0.00 | 0.88 | 0.00 |
|  | **Expiry** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **SOS** | **Supply** | 5157.35 | 914.22 | 2439.90 | 430.83 | 549.99 | 206.36 | 7621.29 | 1389.61 |
|  | **Demand** | 354.62 | 55.41 | 132.98 | 22.16 | 33.25 | 11.08 | 432.20 | 77.57 |
|  | **Import** | 0.00 | 0.00 | 0.00 | 0.00 | 0.18 | 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** | 830.88 | 360.82 | 292.25 | 106.79 | 14.26 | 23.18 | 963.91 | 205.18 |
|  | **Demand** | 349.75 | 54.65 | 131.16 | 21.86 | 32.79 | 10.93 | 426.26 | 76.51 |
|  | **Import** | 4.83 | 0.04 | 7.61 | 0.35 | 38.30 | 1.14 | 7.45 | 0.00 |
|  | **Expiry** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

Table 18: Average results achieved for each metaheuristic implementation subjected to dataset 6 for each blood group measured in units.

|  |  |  |
| --- | --- | --- |
| Metaheuristic | Time (Ms) | Time(Minutes) |
| GA | 4321070 | 72.017 |
| PSO | 503324 | 8.38 |
| DA | 4302052 | 71.70 |
| SOS | 5906312 | 98.43 |
| GWO | 4362574 | 72.70 |

Table 19: Running time per metaheuristic for dataset 6

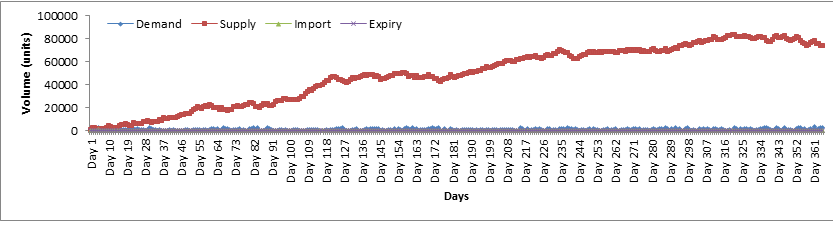
****

Fig. 33: Represents a line graph over a period of 365 days for the GA implementation of dataset 6

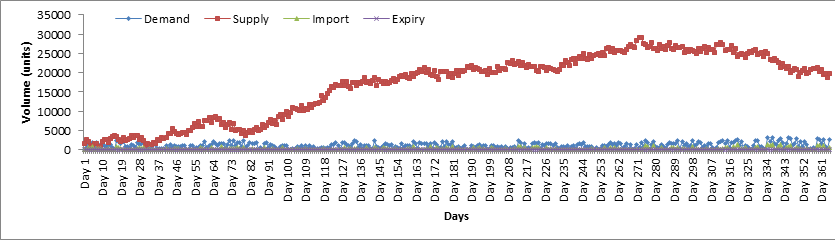
****

Fig. 34: Represents a line graph over a period of 365 days for the PSO implementation of dataset 6

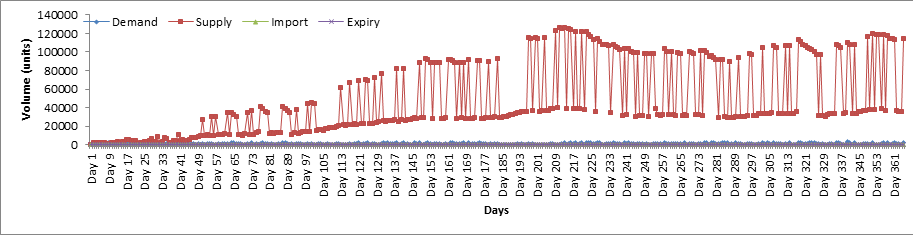
****

Fig. 35: Represents a line graph over a period of 365 days for the DA implementation of dataset 6

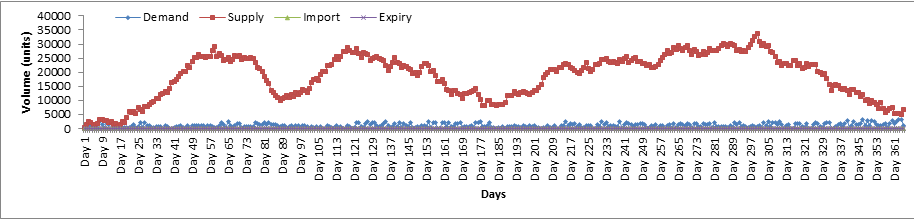
****

Fig. 36: Represents a line graph over a period of 365 days for the SOS implementation of dataset 6

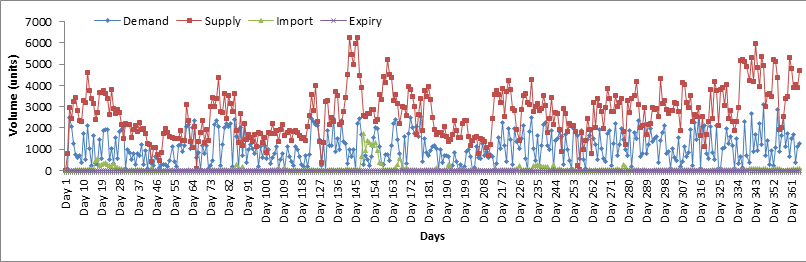
****

Fig. 37: Represents a line graph over a period of 365 days for the GWO implementation of dataset 6

**Evaluating results from Dataset 6**

Fig. 33-37 represents line graphs obtained when each metaheuristic algorithm was subjected to dataset 6. Dataset 6 represents SAGV for demand and supply generation coupled with an increase in the initial volume of WB units (5000 units). After stock-piling occurs, majority of the MT implementations store large amounts of stock with very minimal importation after stock piling occurred. The only MT not to achieve stock-piling was the GWO implementation, therefore the GWO system did encounter importation at certain points within the 365 day time frame. A noticeable trend is the shortened period of time for stock-piling to occur as compared to dataset 1 (control dataset), due to the larger volumes remaining units occur more frequently which aids in the stock-piling process.

**5.1 Comparison between generating demand bounds**

Research conducted by [4] used constant percentage bounds ranging between 27-75% in order to generate values for demand. These bounds were used across the entire testing range (365 days). This study allocated specific percentage ranges to each month in hopes of generating more accurate demand levels in accordance to South African monthly schooling terms and public holidays. Ideally, the best source for generating demand percentage bounds would preferably be statistics based around actual demand for WB units within South Africa, however these statistics could not be located. Therefore the second option was used. For easier identification, Dem1 will refer to demand generated using the constant percentage bounds between 25-75% and Dem2 will represent SAGV for demand. The average demand generated for Dem2 is lower due to the varying percentage ranges, a month will be exposed to a very low percentage bound which in turn reduces the overall average, unlike Dem1 which has a constant percentage range. The following figures depict the demand curvature for each MT implementation.

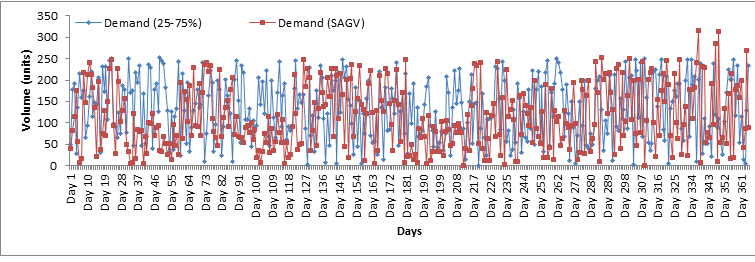


Fig. 38: Represents a comparison between demand generations for GA

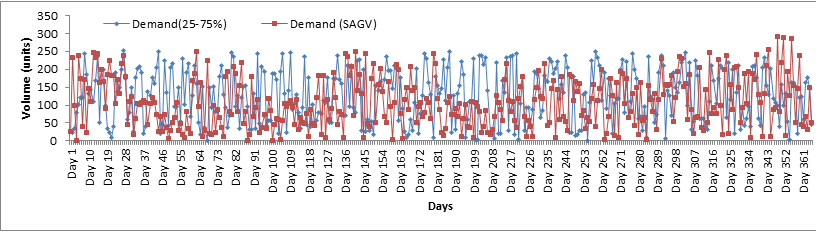


Fig. 39: Represents a comparison between demand generations for PSO

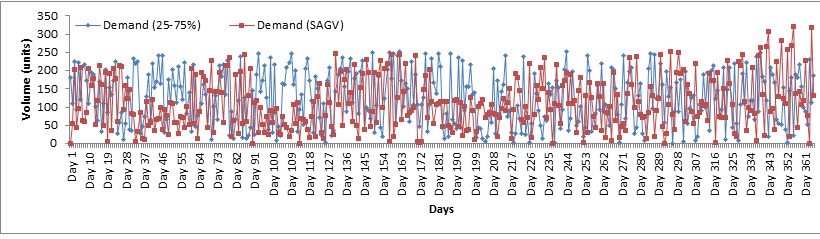


Fig. 40: Represents a comparison between demand generations for DA

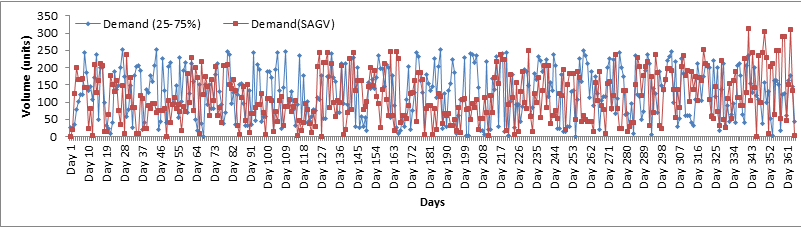


Fig. 41: Represents a comparison between demand generations for SOS

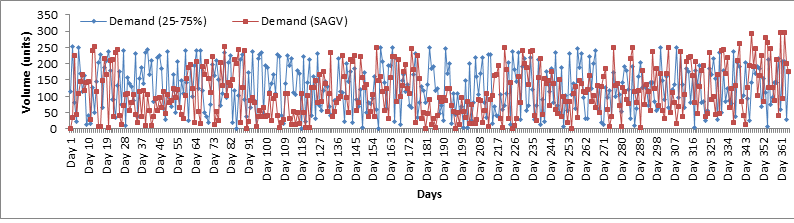


Fig. 42: Represents a comparison between demand generations for GWO

With reference to fig. 38-42, these graphs represent a comparison between demand generated using set percentage bounds between 25-75% (which were used in studies conducted by [2], [4] and [7]) in comparison to demand generated using SAGV. Dem1 depicts a stochastic curvature across 365 days. Looking at dem2, it is noticeable that certain periods experience a lower demand curvature. For example around days 28-55 represent the month of February, according to table 5, February has a lower demand for WB units as compared to December (day 334) which has a much higher demand curvature. Even though the statistics used for generating these demands may not be entirely accurate, it still provides some form of stability which can be improved upon in future literature. By implementing this technique, it reduces the randomness in generating demand values which in turn provides more accurate results.

6. Conclusion

The following paper presented the implementation of 5 metaheuristic algorithms namely GA, PSO, DA, SOS and GWO to work the BAP with the objective of minimizing the total amount of importation and expiration experienced by a blood bank. Upon implementing the metaheuristic algorithms, generating demand values for datasets implemented a formula that took into account South African public holidays and schooling terms in order to generate more accurate values. All metaheuristic implementations produced efficient results, however the SOS system provided the best results with almost no form (or very little) importation across all datasets. Determining a superior algorithm was determined by how quick an algorithm began stock-piling. Stock-piling enabled the algorithm to incur fewer importations, and due to WB’s life-span of 30 days, expiration was unlikely to occur. Expanding upon the aspect of expiry, none of the metaheuristic algorithms experienced this scenario due to the FIFO issuing system. It must be noted that the worst performing algorithm was the GWO. The GWO found good solutions on particular days, but experienced very high levels of demand at random periods, and could not sufficiently stock-pile adequate levels of WB units over a period of time. With regards to GA, DA, and PSO, these algorithms also produced satisfactory results very similar to the SOS algorithm, but simply took much longer to experience stock-piling, which in turn raised their importation levels. In terms of computational time, most algorithms experienced running times exceeding 60 minutes, except the PSO algorithm which was much faster averaging less than 10 minutes per dataset. Even though stock-piling drastically reduced importation, some blood types (mostly the rarer blood types) still experienced importation. It must be remembered that fig 8-37 represent a cumulative value for supply and demand for that specific day, therefore the blood types with a higher rarity may never experience stock-piling within the time period.

An important aspect of this study for the BAP relates to the random generation of demand values. As mentioned, this study incorporated values which conformed to South African public holidays and schooling terms (which in turn points at South African social behaviour discussed in section 4.2). With reference to fig. 38-42, the graphs illustrate trends in demand levels. When using a fixed percentage bounds (25-75%), the demand levels are stochastic which should not be the case, as the demand for WB units should have higher levels during certain periods in a year. The introduction of generating demand using unique percentage bounds allocated to each month manage to successfully give a more accurate reading for demand levels. Due to the fixed percentage bounds being constant throughout the time period, and the SAGV demand generation having much lower demand values at certain periods, the overall average of the fixed percentage approach is much higher in comparison to the SAGV approach. Even though real-world data could not be located, the ideology can be expanded upon in future literature with blood management problems, or any other perishable inventory problem.

Further work can improve upon the generation of demand values by utilising more accurate statistics based from a nation. The use of more assumptions can be introduced to try and change the mathematical model, as well as using real-life data instead of randomly generated data. The following study has tried to bridge the gap between previous literatures by means of exploring new MT implementations as well as try to contribute towards the study of blood related topics, as well as other inventory problems relating to perishable items.

**References**

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

[2]: Groth. S, (2005), Manual on the management, maintenance and use of blood cold chain equipment, World health organization, Geneva

[3]: 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.

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

[5]: 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.

[6]: Alloja. J, Espenberg. K, Kiivet. R. A, (2012), The analysis of cost-effectiveness of the optimal system of the blood service, Tartu, Estonia

[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]: Sorensen. K, (2013), Metaheuristic-the metaphor exposed, International transaction in operation research, Antwerp, Belgium, pp 1-16

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

[10]: Bas. S, Carelio. G, Lanzarone. E, Yalcindag. S, (2016), Management of blood donation system: Literature review and research perspectives, Health care systems engineering for scientists and practitioners, Lyon, France, pp. 121-132

[11]: Khaing. P.P, Rattanaroengchai. W, Tupcullai. S, (2016), Blood bank management system, Bangkok, Thailand.

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

[14]: Kaveh. A, Ghobadi. M, (2016), A multistage algorithm for blood banking supply chain allocation problem, University of Science and technology, Iran.

[15]: Sahin. G, Surel. H, Meral. S, (2007), Locational analysis for regionalization of Turkish Red Crescent blood services, University of Florida, United States of America

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

[17]: Mid year population estimates report, (2017),Statistics South Africa, pp 1

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

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

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

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

[22]: Carr. J, (2014), An introduction to genetic algorithms,

[23]: Rini. D.P, Shamsuddin. S.M, Yuhaniz. S.S, (2011), Particle swarm optimization: Technique, System and Challenges, International journal of computer applications (Vol 14, No: 1), Malaysia.

[24]: 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.

[25]: 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.

[26]: Prakash. S, Rajathy. R, (2015), Implementation of Symbiotic Organism Search algorithm for Extracting Maximum Power from the PV system under Partially Shaded Condition, International science press (vol 8, No:5), Pondicherry, India, pp. 1871-1880

[27]: 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

[28]: 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.

[29]: Precup. R.E, David. R.C, Szedlak-Stinean. A, Petriu E.M, Dragan. F, An easily understandable grey wolf optimizer and its application to fuzzy controller tuning, (2017), Algorithms (vol 10, No: 68), Ottawa, Canada