Strategic Optimization of Blood Supply Chain Management for Efficient Blood Bank Operations*

Simmera Ndlalane[1,†], Olumuyiwa Otegbeye[1,†] and Absalom E. Ezugwu[2,*,†]

[1]*School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, South Africa*
[2]*Unit for Data Science and Computing, North-West University, 11 Hoffman Street, Potchefstroom, 2520, North-West, South Africa*

Abstract

South Africa faces a critical shortage of blood donors, leading to substantial deficits in the national blood supply. Blood donations are vital for the treatment of life-threatening conditions, making it crucial to develop efficient models for the management of blood stocks. This paper presents a mathematical model to optimize blood donation and ensure sufficient supply to meet fluctuating demands. The model captures the complex interactions within the blood banking system, focusing on minimizing costs, reducing waste, and efficiently distributing blood units. Specifically, it addresses daily supply challenges by minimizing the need for emergency imports and reducing blood wastage due to expiration while meeting all demand requirements. The core objective is to minimize blood wastage and reduce the reliance on imported blood banks during emergencies. The proposed objective function incorporates variables such as emergency importation and expiration rates, and robust optimization techniques are applied to identify optimal solutions while satisfying operational constraints. Symbiotic Organism Search (SOS), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO) methods are utilized for optimization. Among these, SOS demonstrated superior performance, achieving the lowest levels of importation and wastage. However, the algorithms were unable to significantly reduce supply levels due to the accumulation of excess stock from the previous day, which carried over into the next day. This paper provides valuable information on blood supply management and highlights the potential for optimization techniques to improve efficiency and sustainability in blood banking.

Keywords

Blood banking, Blood importation, Blood wastage, SOS, GA, PSO, Blood demand, Blood emergency, Blood expiration, Blood assignment problems

1. Introduction

Blood is an essential fluid that delivers oxygen and nutrients to cells while removing carbon dioxide and waste products. The primary constituents of blood consist of red blood cells, responsible for oxygen and carbon dioxide transport; white blood cells, crucial for combating pathogens and aiding in immune response; platelets, which facilitate clot formation to prevent blood loss from injuries; and plasma, the fluid component transporting blood cells and platelets throughout the body [1]. Plasma also contains proteins, ions, nutrients, and wastes. Blood donation involves a voluntary process in which blood is extracted from donors and then transported to blood banks for storage until it is required for transfusion to patients in hospitals who require blood. Examples of situations where blood might be needed range from traumatic accidents, surgical procedures, childbirth, chronic disease, severe infections, blood-related conditions, and excessive blood loss, among others [2]. Therefore, ensuring an adequate blood supply is crucial for effective blood donation.

A blood bank is a center where blood collected from blood donation is stored and preserved for later use in blood transfusion. Key participants and entities within the blood supply chain process include blood donors, blood banks, hospitals, and patients. Figure 1 outlines the general interactions between each participant. In addition, it highlights the flow of blood from donation to utilization, emphasizing the crucial roles played by each entity.

HC@AIxIA 2024: 3rd AIxIA Workshop on Artificial Intelligence For Healthcare
*Corresponding author.
†These authors contributed equally.
2447056@students.wits.ac.za (S. Ndlalane); olumuyiwa.otegbeye@wits.ac.za (O. Otegbeye);
abstractomezugwu@nwu.ac.za (A. E. Ezugwu)

The operations in a blood bank include the collection of blood from donors, processing the blood, testing its health and specific properties, separating blood units based on blood type and other factors, and finally, storing these operations. As illustrated in Figure 1, the process of blood collection and distribution begins with the testing and screening phase, where blood collected by the South African National Blood Service (SANBS) undergoes rigorous testing to ensure its safety and suitability for transfusion. This includes screening for diseases such as HIV and other potential contaminants. Following successful screening, the blood collected is transferred to designated blood banks for further processing. Here, blood is subjected to processing techniques to preserve its quality and extend its shelf life. Once processed, blood is classified according to its blood type and subjected to additional screenings to confirm its safety for transfusion. Blood is quickly discarded to avoid potential harm if abnormalities or contaminants are detected during this stage. However, if the blood passes all necessary screenings and is deemed safe, it is sent to various hospitals and medical facilities according to their demands and needs. This ensures that hospitals have a steady blood supply to meet the demands of patients who require transfusions. The model derived later will focus only on the phases after donor acceptance and blood collection. It will not consider the initial step of donor evaluation or the risk of donor rejection, as these factors can vary over time and significantly impact the inflow of blood units to a hospital.

Alternatively, hospitals can transfer or export surplus blood to other facilities facing emergencies or experiencing lower demand. This collaborative approach helps optimize the distribution of blood resources, ensuring that they are used efficiently where they are most needed. Any surplus blood that remains after transfusion or exportation is stored in designated storage facilities. These blood reserves serve as a crucial backup to ensure that an adequate blood supply is always available, particularly during increased demand or emergencies. However, it is essential to note that despite these measures, blood units have a finite shelf life. If blood expires before it can be utilized, it is discarded to maintain the integrity and safety of the blood supply. Hospitals can import additional blood units from other facilities when they face a sudden surge in blood demand. This emergency measure helps to address immediate needs and ensures that patients receive timely and life-saving transfusions. These blood products use first-in-first-out (FIFO) and last-in-first-out (LIFO) methods to reach hospitals [2].

The demand for blood has increased worldwide, while there are low levels of blood donations. This means that the demand for blood exceeds the blood supply in hospitals. This poses a threat to patients in need of blood. Blood is being wasted through expiration when a specific type of blood is not in demand. There is insufficient blood stored for emergency events such as car accidents.

2. Related Work

This section presents a brief overview of preliminary studies and related works to further emphasise the current study's relevance.

According to Charpin *et al.* [1], blood is continuously required daily in hospitals for blood transfusion, emergencies, and treating diseases. Addressing shortages, handling limited shelf life, and navigating blood type mismatches present challenges in managing ongoing transfusion demands, as Govender and Ezugwu [3] highlighted in their research. An adequate blood supply is critical to ensure that lifesaving measures can be implemented when needed.

Blood compatibility is one of the most important factors in blood transfusions. Additionally, natural blood grouping restricts transfusion options due to blood compatibility. Karl Landsteiner's 1901 discovery identified the human blood system, the ABO system, comprising four primary groups. In 1940, Reid *et al.* [4] discovered that in total, there exist eight blood group classifications for white blood cells, including A^+, A^-, B^+, B^-, AB^+, AB^-, O^+ and O^-. Table 1 illustrates the compatibility pairs among the eight blood groups, with blood type AB represented as C. The first row of the table represents the blood donor's blood type, and the first column represents the recipient's blood type. "+" entries in the table indicate compatibility, signifying that the donor's blood type matches that of the recipient, allowing for a successful blood transfusion. The "-" entries in the table indicate an incompatibility between the donor

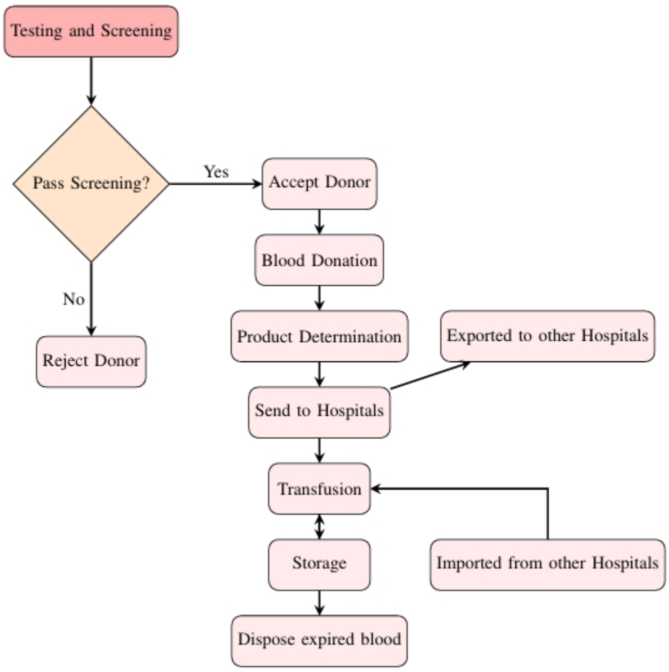

Figure 1: Flowchart of Blood Donation Process

and recipient. This concept forms the basis of the model formulation in this study. As is evident, O^- serves as the universal donor, while AB^+ acts as the universal recipient. During emergencies, patients are given blood type O^- since it can be administered to anyone, as noted by Charpin *et al.* [1].

Table 1
Blood Compatibility Chart

Recipient/Contributor	A^+	A^-	C^+	C^-	B^+	B^-	O^+	O^-
A^+	+	+	−	−	−	−	+	+
A^-	−	+	−	−	−	−	−	+
C^+	+	+	+	+	+	+	+	+
C^-	−	−	−	+	−	+	−	+
B^+	−	−	−	−	+	+	+	+
B^-	−	−	−	−	−	+	−	+
O^+	−	−	−	−	−	−	+	+
O^-	−	−	−	−	−	−	−	+

The shelf life of blood depends on the type of blood product and the temperature conditions. Whole blood lasts 30 days, red blood cells 24 days, plasma 12 months, and platelets 5 days. This study focuses on whole blood cells; thus, we will use 30 days for the expiration period in accordance with the study conducted in [3]. The assignment aims to allocate blood products to hospitals while minimizing the need for imports and mitigating the risk of expiration, as outlined by Charpin *et al.* [1] and Ezugwu *et al.* [5] in their investigation into optimal distribution strategies.

A model that describes the inflow and outflow processes of blood units is necessary to enhance the blood supply chain. In 1976, a planning model was developed by Cumming *et al.* [6] for donation collection and a basic model for distributing blood units to hospitals. Subsequently, a model was devised by Charpin *et al.* [1] that simplifies the blood assignment problem, focusing solely on red blood cells and excluding the Rhesus factor (with potential future inclusion). Govender and Ezugwu [3] later formulated an optimization objective function to efficiently allocate blood units to hospital patients while minimizing wastage due to expiry and reducing importation from external sources. Blood allocation for daily demand and the available supply follows the FIFO process, prioritizing the oldest blood units first and the newest last. The goal was to enhance the efficiency of the blood allocation procedure using the SANBS and demographic data.

The study by Dufourq *et al.* [7] addressed blood assignment problems; the study endeavours to optimize blood allocations to patients while minimizing blood importation without considering the expiration and emergency factors. The findings suggest that GA facilitated a more efficient distribution of blood importation, prioritizing fewer imports of more valuable types. According to the research undertaken by Govender and Ezugwu [3], which tried solving the blood assignment problem by minimising blood unit wastage importation while efficiently distributing blood units. The authors investigated six algorithms, including PSO, but the SOS algorithm slightly lowered the importation levels.

3. Methodology

To enhance the mathematical model introduced in prior research, which omitted considerations such as blood expiration and emergency demand, this study aims to refine it. Thus, the expanded model incorporates factors including the rate of blood expiration, the volume of blood expiring per unit time, the quantity of blood imported from other blood banks for each blood group, and the volume of blood exported to other blood banks for each blood group. The mathematical model maintains eight differential equations, each corresponding to a distinct blood type: A^+, A^-, B^+, B^-, AB^+, AB^-, O^+, and O^-. Each equation denotes the rate of change of the total blood units for the respective blood type, determined by subtracting the total available blood for transfusion from the total units transfused, adjusted for expired blood and accounting for both imported and exported blood units for emergency purposes at other hospitals.

Figure 2 shows the interactions between blood types A^+, A^-, B^+, B^-, AB^+, AB^-, O^+, and O^-, with AB simplified as C. The flowcharts depict how each type evolves through expiration, importation, exportation, and donations. Each blood type is color-coded, representing the system of differential equations governing changes in blood units. Inputs are added, and outputs are subtracted from the equations.

This study addresses the removal of expired units after transfusions or transfers and importation during emergencies. Importation increases blood availability, while exportation decreases it. The model represented by Equation 1, 2, 3, 4, 5, 6, 7, and 8 serves as a mathematical representation of the problem, accounting for the complexities of different blood types.

3.1. Mathematical Model

$$\begin{aligned}\frac{dV_{O^-}}{dt} =&Q_{O^-} - \left(D_{O^-O^-} + D_{O^-O^+} + D_{O^-A^-} + D_{O^-A^+} + D_{O^-B^-} + D_{O^-B^+} + D_{O^-C^-} + D_{O^-C^+}\right) + I_{O^-} - \\ &\alpha_1 V_{O^-} - E_{O^-},\end{aligned} \tag{1}$$

$$\frac{dV_{O^+}}{dt} =Q_{O^+} - \left(D_{O^+O^+} + D_{O^+A^+} + D_{O^+B^+} + D_{O^+C^+}\right) + I_{O^+} - \alpha_2 V_{O^+} - E_{O^+}, \tag{2}$$

$$\frac{dV_{A^-}}{dt} =Q_{A^-} - \left(D_{A^-A^-} + D_{A^-A^+} + D_{A^-C^-} + D_{A^-C^+}\right) + I_{A^-} - \alpha_3 V_{A^-} - E_{A^-}, \tag{3}$$

$$\frac{dV_{A^+}}{dt} =Q_{A^+} - \left(D_{A^+A^+} + D_{A^+C^+}\right) + I_{A^+} - \alpha_4 V_{A^+} - E_{A^+}, \tag{4}$$

$$\frac{dV_{B^-}}{dt} =Q_{B^-} - \left(D_{B^-B^-} + D_{B^-B^+} + D_{B^-C^-} + D_{B^-C^+}\right) + I_{B^-} - \alpha_5 V_{B^-} - E_{B^-}, \tag{5}$$

$$\frac{dV_{B^+}}{dt} =Q_{B^+} - \left(D_{B^+B^+} + D_{B^+C^+}\right) + I_{B^-} - \alpha_6 V_{B^+} - E_{B^+}, \tag{6}$$

$$\frac{dV_{C^-}}{dt} =Q_{C^-} - \left(D_{C^-C^-} + D_{C^-C^+}\right) + I_{C^-} - \alpha_7 V_{C^-} - E_{C^-}, \tag{7}$$

$$\frac{dV_{C^+}}{dt} =Q_{C^+} - D_{C^+C^+} + I_{C^+} - \alpha_8 V_{C^+} - E_{C^+}. \tag{8}$$

For x representing an element of A^+, A^-, B^+, B^-, AB^+, AB^-, O^+, and O^-, where AB is denoted as C, and for n as an element of $\{1, 2, 3, 4, 5, 6, 7, 8\}$, the explanation of each term in this model is as follows:

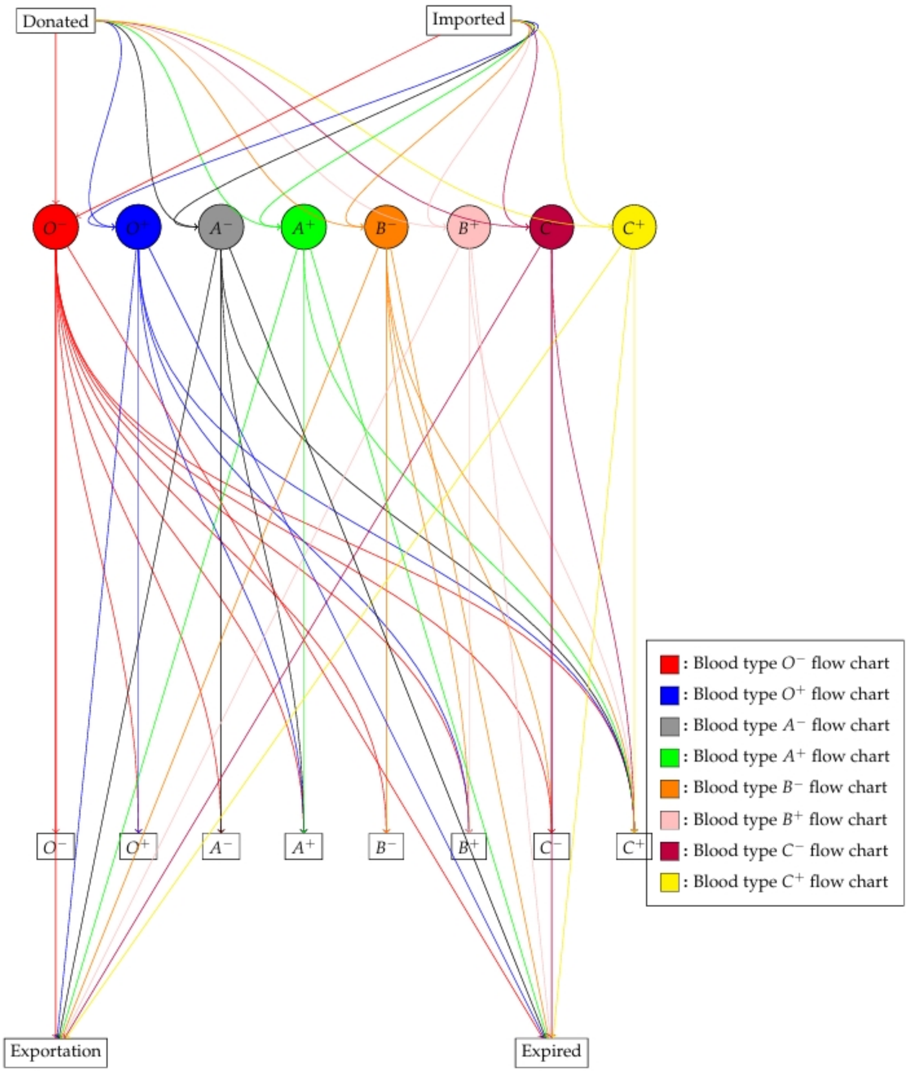

Figure 2: Blood type flowchart

V_x denotes the amount of blood available for donation, while $\frac{dV_x}{dt}$ represents the rate of change of the blood type V_x over time. Q_x indicates a source of blood from donations, and the terms D_{xy} refer to various diffusion rates between different blood types. The term I_x represents an external blood input from other hospitals, often in emergencies. α_n stands for the expiration rate of blood units for each blood type, and $\alpha_n V_x$ accounts for the degradation or removal process due to expiration, proportional to the concentration of V_x. Finally, E_x represents the exportation of blood to other hospitals to meet their emergency demands. The number of blood units of a particular blood type imported (I_x) and exported (E_x) on any given day will depend on the number of emergencies in the hospital. The blood units indicated by I_x will be imported into the hospital to address emergencies, while those represented by E_x will be exported to other hospitals to meet their emergency needs. Additionally, the volume of blood transfused during emergencies is critical in determining the net inflow and outflow of blood units within the healthcare system.

3.2. Objective Function

The equation

$$\zeta = \min\left(\sum_{t=1}^{p} E(t) + \sum_{t=1}^{p} I(t)\right), \tag{9}$$

subject to

$$1 \leq t \leq 365,$$

denotes the formulation of the model objective function and aims to minimize the amount of blood imported and expired for each blood type. It will be tested with the dataset, and its results will be compared to those reported in previous studies.

3.2.1. Expiration

Equation 10 calculates the remaining amount of blood after it expires 30 days post-issuance, rendering it unsuitable for hospital use due to health reasons:

$$E_x(t) = \sum_{t=31}^{\infty} R_x(t), \tag{10}$$

subject to $t > 30$. Where $R_x(t)$ determines the amount of blood that remains after it has been issued.

$$R_x(t) = S_x(t) - D_x(t), \tag{11}$$

subject to

$$S_x(t) \geq D_x(t), 1 \leq d \leq 365.$$

If the demand over the period after 30 days is less than the supply before the 30-day expiration, the expired amount will be the difference between this supply and the demand, as the blood will surpass the 30-day expiration period. This can be mathematically represented as follows:

When $t > 30$, if

$$\left(\sum_{t=30}^{p} D_x(t)\right) < S_x(t-30),$$

then,

$$E_x(t) = S_x(t-30) - \left(\sum_{t=30}^{p} D_x(t)\right),$$

where $1 \leq t \leq 365$.

3.2.2. Importation

Hospitals frequently encounter situations where there is an urgent need for blood due to emergencies. In such cases, hospitals can initially import blood from other compatible blood types and then resort to importing blood from other hospitals when needed. Equation 12 calculates the amount of blood imported for each type.

$$\begin{aligned} I(t) = & I_{A^+}(t) + I_{A^-}(t) + I_{B^+}(t) + I_{B^-}(t) + \\ & I_{AB^+}(t) + I_{AB^-}(t) + I_{O^+}(t) + I_{O^-}(t) \end{aligned} \tag{12}$$

Blood banks and hospitals can import blood from other compatible blood types when there is a shortage of a particular blood type, which can be represented as:

If

$$D_x(t) > S_x(t), \tag{13}$$

then

$$I_y(t) = D_x(t) - S_x(t). \tag{14}$$

Figure 3.2.2 represents the vector representation of an individual in the population. Each segment represents a distinct value derived from the blood data set, facilitating individual manipulation in subsequent calculations.

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

Figure 3: Grid representation of an individual in a population

3.3. Methods

In this research, Metaheuristic Algorithms such as Symbiotic Organism Search (SOS) [8], Genetic Algorithm (GA) [9], and Particle Swarm Optimization (PSO) [10] have been selected based on their proven performance and suitability for the problem at hand. The decision to use SOS and PSO is supported by the study conducted by Govender and Ezugwu [3], which focused on the Blood Assignment Problem and concluded that SOS outperformed other metaheuristic implementations, while PSO was the fastest in producing results. Additionally, the GA algorithm has been chosen based on the findings by Dufourq *et al.* [7], where GA not only outperformed other algorithms under investigation but also provided a more efficient distribution of blood importation with fewer imports from high-value blood types. These algorithms were selected for their efficiency, speed, and ability to achieve superior results in similar problem domains.

3.3.1. Symbiotic Organism Search

The SOS and GA algorithms have demonstrated superior performance compared to other algorithms examined in each study. The algorithm incorporates three phases inspired by real-world biological interactions: mutualism, commensalism, and parasitism phases.

Mutualism Phase: An organism X_j is selected to pair with organism X_i. Together, these organisms engage in mutation to enhance the survival probabilities of organisms within the ecosystem. The offspring solutions X_i^{new} and X_j^{new} are computed based on the mutualistic symbiosis between the parent organisms X_i and X_j. The calculations are defined as:

$$X_i^{\text{new}} = X_i + \text{rand}(0,1) * (X_{\text{best}} - \text{Mutualism\_Vector} * \text{BF}_1), \tag{15}$$

$$X_j^{\text{new}} = X_j + \text{rand}(0,1) * (X_{\text{best}} - \text{Mutualism\_Vector} * \text{BF}_2), \tag{16}$$

$$\text{Mutualism\_Vector} = \frac{X_i + X_j}{2}. \tag{17}$$

Commensalism Phase: Randomly selecting two organisms, X_i and X_j, from the ecosystem, we modify organism X_i with the assistance of organism X_j. The resulting child solution, derived from this modification through commensal symbiosis between organisms X_i and X_j, is expressed as:

$$X_i^{\text{new}} = X_i + \text{rand}(-1,1) * (X_{\text{best}} - X_j). \tag{18}$$

Parasitism Phase: Organism X_i is randomly selected, and a Parasite Vector is created by duplicating X_i and modifying some dimensions. Then, another organism X_j is chosen as the host and is replaced by the parasite vector, which usually has a better fitness value than X_j. However, if X_j has a higher fitness, it becomes immune to the Parasite Vector, preventing it from surviving in the ecosystem.

3.3.2. Genetic Algorithm

This algorithm utilizes recombination and mutation to generate new chromosomes, akin to biological reproduction. The mutation alters genes within the chromosome. GA aims to evolve the optimal chromosome for solving a given problem. The algorithm comprises three main components: natural selection, mutation, and crossover.

Natural Selection: In nature, individuals with superior survival traits survive for longer periods. Consequently, over time, the population becomes dominated by genes from these superior individuals, while genes from inferior individuals diminish. Species with high survival rates thrive, whereas those with low survival rates perish. This is the theory of natural selection.

Crossover: During the crossover operation, two individuals combine genetic material to create diverse offspring. The parent strings yield children strings based on a chosen crossover point. With a crossover probability p_c, $100 \cdot p_c\%$ of the population undergoes crossover, while $100 \cdot (1 - p_c)\%$ remains unchanged. Common methods include single-point and double-point crossovers.

Mutation: Mutation introduces random variations into the genetic search process, thereby preventing the population from becoming stuck in local optima. It enhances diversity within the population by operating at the bit level: during reproduction, each bit in the offspring has a small probability of mutation, typically denoted as mutation probability p_m.

3.3.3. Particle Swarm Optimization

The PSO algorithm creates global memory for the whole population of particles by recording the best-ever position and the corresponding fitness value. This is computed on every iteration of the algorithm. In a PSO system, particles traverse a multi-dimensional search space by "flying" around until they reach a relatively stable state or until computational constraints are met [10]. In a multi-dimensional space, let X represent the positions of m particles, expressed as $X = [X_1, \dots, X_j, \dots, X_m]$. At time step t, the position of the j-th particle, X_k^t, is determined by its previous position and current velocity, denoted as $X_k^t = X_k(X_k^{t-1}, V_k^t)$. The neighborhood of a particle X_k, $N(X_k)$, includes all particles X_j that are "near" X_k. The best previous position of X_k is defined as $X_k^{t^*}$, satisfying $f(X_k^{t^*}) > f(X_k^t)$. Each particle X_k updates its state according to the equations:

$$V_{k+1}^t = wV_k^t + c_1 r_1 (X_k^{t^*} - X_k^t) + c_2 r_2 (g_k - X_k^t), \tag{19}$$

$$X_{k+1}^t = V_{k+1}^t + X_k^t. \tag{20}$$

The global best position found by the swarm from the beginning of the search up to the current iteration k is represented by the term g_k is calculated using the equation:

$$g_k = best_{t,k}\{X_k^t, t = 1, 2, .., n, k = 1, 2, .., k\}. \tag{21}$$

3.4. Dataset Summary

The study uses real-world blood datasets to implement state-of-the-art metaheuristic algorithms that include SOS, GA, and PSO. Data sourced from the Enugu National Blood Transfusion Service in Nigeria, spanning from 2010 to 2018, was adapted from the Nigerian Enugu blood bank's records over a decade (2009-2019). These datasets detail monthly distributions of blood units across various blood types. Ethical concerns have been carefully considered throughout the process of collecting data. The confidentiality and anonymity of the individuals in the data set have been strictly maintained.

4. Experiment, Results and Discussion

In this section, we describe a series of experiments conducted to evaluate the practicality of the proposed mathematical model and the efficiency of the GA, SOS, and PSO optimization algorithms. The experiments were performed on a computing platform equipped with an Intel Core i3 CPU running at 1.20 GHz, 8 GB of RAM, and the Windows 11 operating system. All three algorithms were implemented using Python.

Different population sizes of 50, 100, and 150 were tested during the experiments. These population sizes were chosen to maintain consistency with a study in [5], which used the same population sizes. By adhering to these values, we ensure a fair comparison with existing research, enabling a more accurate evaluation of our results relative to the established findings. The supply values were set within constant percentage bounds ranging from 0 to 100%, with the initial blood volume capped at 300 units. The selection of parameters for the SOS, PSO, and GA algorithms is consistent with the implementation used in study [5], which worked with the same dataset. Table 4 shows the parameters used to apply the algorithms.

Table 2
Parameter Configuration for SOS, PSO, and GA Algorithms

Parameter	SOS	PSO	GA
Population Size (N)	50/100/150	50/100/150	50/100/150
Crossover Rate (Cr)			0.07
Mutation Rate (m)			0.03
Personal Learning Coefficient (c_1)		1.7	
Global Learning Coefficient (c_2)		1.7	
Inertia Weight (w)		0.715	
Inertia Weight Damping Ratio ($wdamp$)		0.99	
Maximum Iterations	1000/1500/2000/3000	1000/1500/2000/3000	1000/1500/2000/3000

The discrepancy between demand and supply must be zero for an optimal solution, but no such solution was found due to the excess supply. This occurred because leftover blood units from the previous day were carried over, increasing the total supply. As a result, the algorithms couldn't meet the required optimal supply. Additionally, import values should be minimal, but this condition wasn't satisfied due to high import quantities. However, the expiration values, which should approach zero, were met. All supply, import, and expiration conditions must be satisfied for a valid solution.

The SOS algorithm achieved the lowest importation levels compared to GA and PSO, as shown in Tables 3 and 5, which aligns with our goal to minimize imports. Additionally, from Tables 3, 4, and 5, the PSO algorithm consistently minimized blood expiration, with values very close to zero across all population sizes. In contrast, other algorithms still had blood units expiring after 30 days. The different population sizes investigated show trends in monthly blood volume (imported versus expired) for a population of 50 monitored over a108 months(2010–2018).

Table 3
Best overall solution across all months for N = 50.

MA	Metric	O^+	O^-	A^+	A^-	B^+	B^-	AB^+	AB^-
SOS	Supply	52.3378	82.0033	4.6288	2.8539	72.4485	79.1469	5.5448	31.8383
	Import	0.3812	0.0000	0.0000	1.4168	0.0000	0.1493	0.0000	0.0000
	Expiry	0.8947	0.7583	1.0100	0.0000	0.0000	0.4466	0.0000	2.1598
GA	Supply	66.1899	26.3545	38.4704	33.0664	43.4723	62.5835	80.0120	40.2313
	Import	0.0000	2.2822	0.0000	8.5202	0.0000	6.5266	0.0000	0.0000
	Expiry	7.0944	0.0000	0.0000	7.5624	0.0000	0.0000	0.0000	0.1411
PSO	Supply	85.3712	44.1428	11.6032	34.3623	22.5331	26.5761	57.9227	38.8552
	Import	0.0000	0.0000	0.0000	0.2374	0.0000	0.0000	0.0759	0.0000
	Expiry	0.0000	0.0000	0.0000	0.0000	0.0322	0.0000	0.00098	0.0000

Table 4
Best overall solution across all months for N = 100.

MA	Metric	O^+	O^-	A^+	A^-	B^+	B^-	AB^+	AB^-
SOS	Supply	94.0303	55.2482	89.6114	51.9018	67.7487	61.5071	16.2537	5.3548
	Import	0.0000	0.3944	0.0000	0.0000	0.0000	0.0000	0.1091	0.0000
	Expiry	2.0318	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
GA	Supply	63.7027	59.9290	74.8799	77.2995	42.9237	41.8013	20.9964	46.2272
	Import	18.3455	0.0000	0.0000	0.0000	2.6478	0.0000	10.0721	1.4992
	Expiry	5.4329	11.7177	0.0000	0.0000	7.7886	0.0000	2.1490	0.0000
PSO	Supply	98.4776	97.3294	57.3443	15.2266	60.3158	66.5277	89.5080	23.5496
	Import	0.0011	0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0001
	Expiry	0.0000	0.0000	0.0000	0.0007	0.0000	0.0084	2.9875	0.0000

The comparison of the SOS, GA, and PSO algorithms in managing blood supply, import, and expiration across all months (Tables 3, 4, and 5) reveals distinct performance patterns. For a population size of 50, the SOS algorithm delivers the highest supply for O^-, with minimal import and near-zero expiration,

Table 5
Best overall solution across all months for N = 150.

MA	Metric	O^+	O^-	A^+	A^-	B^+	B^-	AB^+	AB^-
SOS	Supply	71.3746	19.2465	70.2420	97.6066	3.1203	80.2222	44.7775	51.3909
	Import	1.2239	1.9949	0.0000	0.0000	0.2681	0.0000	0.0000	0.0000
	Expiry	1.7547	0.5083	1.8952	0.2479	0.0110	1.4982	0.0000	1.2795
GA	Supply	61.3059	20.9250	84.9485	50.3067	53.6114	84.3721	70.1356	84.0563
	Import	10.1416	0.0000	2.4322	2.3351	0.0450	4.5331	3.9506	15.8105
	Expiry	0.0000	0.0000	0.0000	0.0000	2.8422	0.0000	1.5940	5.0178
PSO	Supply	30.9199	76.9317	99.9976	33.3497	27.4404	29.6979	37.2713	15.6569
	Import	0.0047	0.0005	13.7263	0.0000	0.0000	0.0000	0.0000	0.0000
	Expiry	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

especially for A^-, B^+, and AB^+, making it the most efficient. The GA algorithm shows good supply for AB^+, lower import for O^-, but struggles with expiration rates for O^+, A^-, and AB^-. The PSO algorithm minimises import and expiration, particularly for O^+, though zero importation is impractical. For a population size of 100, SOS offers a high supply for most blood types with low imports and expirations but struggles slightly with O^+. GA performs well for A^-, with low import for AB^-, though expiration for AB needs improvement. PSO achieves high supply for O^+ and O^-, while minimizing import and expiration rates, making it the most effective at maintaining supply with minimal waste. At a population size of 150, SOS maintains a high supply for A^- with low import and expiration rates. GA provides a strong supply for A^+ and B^-, but shows less efficient management for expiration, especially for AB^+ and AB^-. PSO delivers the highest supply for O^-, maintaining minimal import and expiration across all blood types.

In summary, SOS demonstrates the most consistent performance in supply management, while PSO excels in minimizing import and expiration. GA shows room for improvement in managing expiration rates.

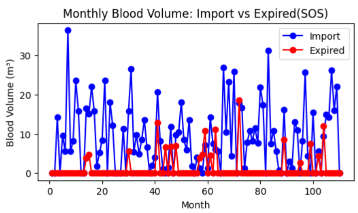

Figure 4: Trends in blood units for a population of 50 for the SOS algorithm.

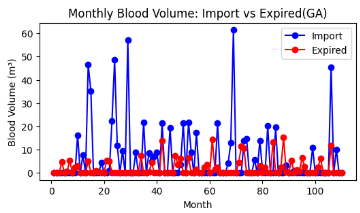

Figure 5: Trends in blood units for a population of 50 for the GA algorithm.

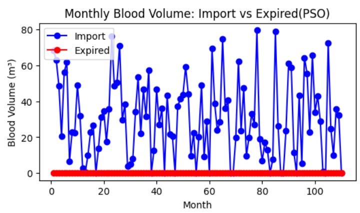

Figure 6: Trends in blood units for a population of 50 for the PSO algorithm.

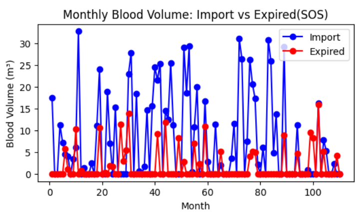

Figure 7: Trends in blood units for a population of 100 for the SOS algorithm.

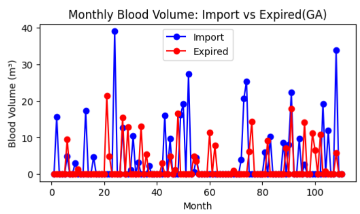

Figure 8: Trends in blood units for a population of 100 for the GA algorithm.

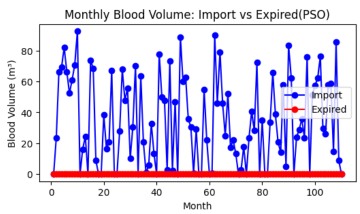

Figure 9: Trends in blood units for a population of 100 for the PSO algorithm.

When the population is initially set to 50, Figures 4, 5, and 6 reveal that imported blood volume exhibits high variability with occasional spikes, while expired volume remains consistently low. Notably, the PSO model shows the highest peak in imported blood, followed by GA, with SOS exhibiting lower peaks. Despite the algorithmic differences, trends indicate that imported blood volume fluctuates

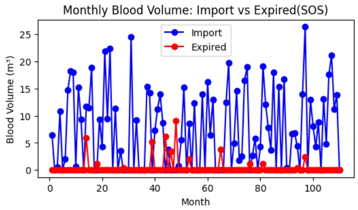

Figure 10: Trends in blood units for a population of 150 for the SOS algorithm.

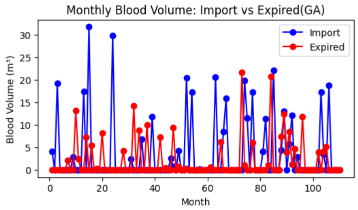

Figure 11: Trends in blood units for a population of 150 for the GA algorithm.

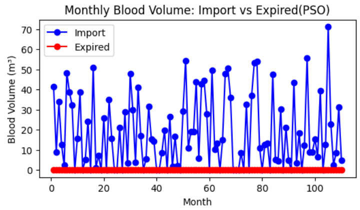

Figure 12: Trends in blood units for a population of 150 for the PSO algorithm.

significantly over time, whereas PSO maintains a zero expired volume, indicating SOS's efficiency in this scenario.

As the population increases to 100, a correlation between importation and expiration is anticipated. In Figure 7, SOS shows significant fluctuations in importation and more minor variations in expired blood. Conversely, Figure 8 depicts both volumes experiencing frequent volatility, yet the expired volume remains low. The PSO model in Figure 9 illustrates more significant import variability while the expired volume remains near zero. Overall, GA proves to be the most efficient at this population size.

With a population size of 150, blood imports display similar fluctuations to those seen in populations of 50 and 100, but with slightly lower peaks. In Figure 10, the SOS algorithm shows that expired blood remains low relative to imports, suggesting a negative correlation between expired units and population size. Figure 11 shows more frequent, less pronounced peaks in imports for GA, while expired units occur more frequently than in SOS, indicating less efficient usage. The PSO model in Figure 12 reflects a slight decrease in imports compared to smaller populations, with expired units remaining low. In this case, SOS again emerges as the most efficient.

Overall, across varying population sizes, the SOS algorithm consistently demonstrates superior efficiency in managing blood imports and minimizing expired units.

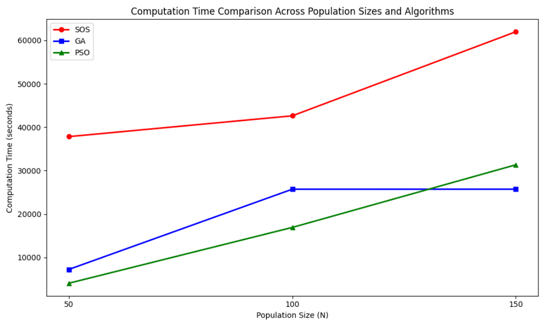

Figure 13: Comparison of Computation Times for SOS, GA, and PSO Algorithms Across Increasing Population Sizes

The computation times for the SOS, GA, and PSO algorithms at different population sizes ($N \in \{50, 100, 150\}$) reveal distinct trends. Additionally, Figure 13 shows that SOS is always higher compared to GA and PSO, and there is a positive relationship between population size and computational time. Its time significantly increases with larger populations, indicating its sensitivity to dataset size, which suggests it may not be the most efficient for higher populations. The GA algorithm also shows a rise in computation time with increasing population size, but this increase is more moderate than that of SOS. While GA requires more time than PSO, it remains faster than SOS, making it a better balance between accuracy and efficiency. In contrast, the PSO algorithm consistently exhibits the lowest computation times at each population size, scaling efficiently with a relatively linear increase in time. Thus, PSO is highly suited for managing larger populations while maintaining computational efficiency. In summary, all algorithms show increased computation times as population size grows. However, SOS has the most substantial rise, GA exhibits moderate scaling, and PSO is the most scalable and efficient for larger datasets.

Additionally, different iteration counts of 1000, 1500, 2000, and 3000 were investigated to analyze how increasing the number of iterations would impact the results. Most prior studies utilized 1000 iterations as a standard, providing a baseline for comparison. The selection of these four specific iteration counts was deliberate, following a systematic progression that allows for a thorough examination of the effects of increased iterations on convergence and solution quality. Starting with 1000 iterations, a commonly accepted threshold in the literature, ensures consistency with existing studies. The increments of 500 allow for a gradual exploration of the effects of increased computational effort on the results. The results for iteration 1000 are represented by Table 3 which was used initially before changing the iteration sizes.

Table 6
Best overall solution across all months for iteration = 1500.

MA	Metric	O^+	O^-	A^+	A^-	B^+	B^-	AB^+	AB^-
SOS	Supply	64.0454	46.8071	54.3417	14.0263	5.3065	89.6529	97.4574	62.4392
	Import	0.0000	0.0000	0.8161	0.2179	0.0000	0.8953	0.0228	0.0000
	Expiry	0.5999	0.0000	0.0122	0.2850	0.8730	0.7395	0.0437	2.5337
GA	Supply	29.5575	81.7788	75.5449	77.7300	11.0796	85.8349	97.8943	23.6903
	Import	4.0783	0.0000	0.0000	5.1850	2.8365	0.0000	0.0000	1.7617
	Expiry	5.4661	3.2693	0.0000	1.9745	2.1263	7.2586	0.0000	15.2805
PSO	Supply	27.2534	41.1694	75.0992	33.6941	100.0000	63.4012	88.5958	74.7302
	Import	0.0000	0.0000	13.2614	0.0000	0.0000	0.0000	0.0000	0.0000
	Expiry	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 7
Best overall solution across all months for iteration = 2000.

MA	Metric	O^+	O^-	A^+	A^-	B^+	B^-	AB^+	AB^-
SOS	Supply	39.7338	31.7426	24.0939	41.3134	19.2563	51.3032	68.0335	1.1295
	Import	2.7445	0.5518	1.0493	0.0000	0.0903	1.1284	0.0000	0.2013
	Expiry	0.8038	0.4107	0.0933	1.8271	0.4450	0.0000	0.0000	1.0459
GA	Supply	56.5495	73.0082	87.9078	3.2571	69.3914	74.4764	46.3198	99.7527
	Import	3.3600	3.8645	9.0355	2.9689	0.0000	8.9388	2.2576	0.0000
	Expiry	5.1377	1.3377	0.0000	1.9421	10.8324	0.0000	0.0000	0.0000
PSO	Supply	22.8763	28.8392	100.0000	0.0000	9.9644	100.0000	36.5501	32.9503
	Import	0.0000	0.0000	0.0000	1.8936	0.0000	0.0000	0.0000	0.0000
	Expiry	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 8
Best overall solution across all months for iteration = 3000.

MA	Metric	O^+	O^-	A^+	A^-	B^+	B^-	AB^+	AB^-
SOS	Supply	3.3447	89.7159	87.1079	74.9300	4.1742	20.4870	98.0572	5.3403
	Import	1.3726	0.0000	0.8773	0.0000	0.0000	0.0000	1.2245	0.2152
	Expiry	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
GA	Supply	39.0868	4.9251	11.2667	56.5289	66.4698	74.2616	57.1878	68.7403
	Import	0.0000	0.0000	9.4337	6.8285	1.0444	3.3390	0.3088	0.0000
	Expiry	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PSO	Supply	59.5217	46.8161	68.6812	56.6590	26.7479	0.0000	100.0000	26.5916
	Import	0.0000	0.0000	0.0000	0.0000	0.0000	2.4892	0.0000	0.0000
	Expiry	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Increasing the number of iterations from 1000 to 1500, 2000, and 3000 significantly improved the results across the algorithms used (SOS, GA, and PSO) in terms of importation and expiration metrics. Both the SOS and PSO algorithms exhibited a marked reduction in importation values, achieving near-zero expiration rates, particularly at higher iterations, which aligns with the goal of minimizing non-zero importation and eliminating expiration. In contrast, the GA algorithm showed less sensitivity to iteration increases, maintaining low importation levels but failing to reach zero expiration consistently. Overall, the findings suggest that higher iterations enhance the performance of most algorithms, particularly SOS and PSO, in optimizing blood supply management. Table 6 and 8 show that the SOS algorithm outperformed the other algorithm as it has the smallest values for both importation and expiration. But for Table 7, the PSO algorithm has the smallest values for both factors.

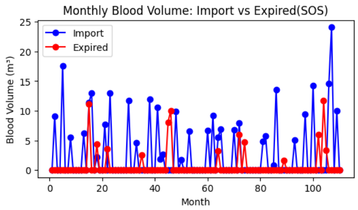

Figure 14: Trends in blood units at iteration 1500 for the SOS algorithm.

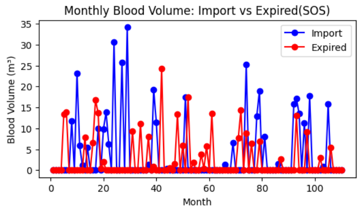

Figure 15: Trends in blood units at iteration 1500 for the GA algorithm.

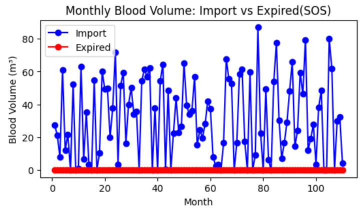

Figure 16: Trends in blood units at iteration 1500 for the PSO algorithm.

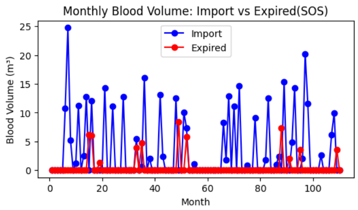

Figure 17: Trends in blood units at iteration 2000 for the SOS algorithm.

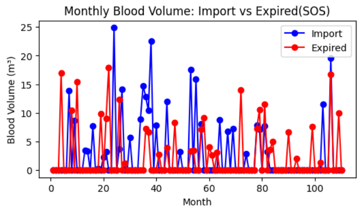

Figure 18: Trends in blood units at iteration 2000 for the GA algorithm.

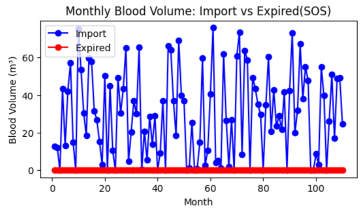

Figure 19: Trends in blood units at iteration 2000 for the PSO algorithm.

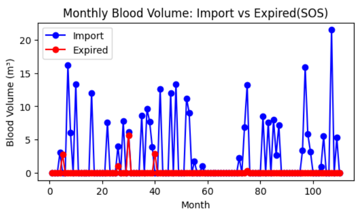

Figure 20: Trends in blood units at iteration 3000 for the SOS algorithm.

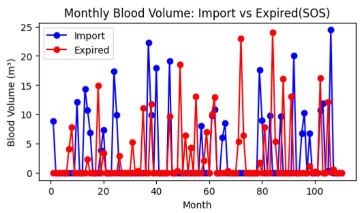

Figure 21: Trends in blood units at iteration 3000 for the GA algorithm.

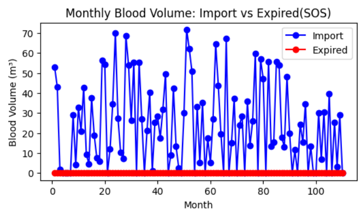

Figure 22: Trends in blood units at iteration 3000 for the PSO algorithm.

The analysis of blood unit trends across different algorithms and iterations (Figures 14–22) suggests that the SOS algorithm may be the most effective in balancing blood imports and minimizing expirations over time. At iterations 1500, 2000, and 3000, SOS (Figures 14, 17, and 20) shows relatively stable import rates with consistently low expiration levels, indicating an efficient management of blood inventory. In contrast, the GA (Figures 15, 18, and 21) exhibits high and frequent import peaks across all iterations, which could lead to unnecessary over-importation without a proportional decrease in expirations. Similarly, the PSO algorithm (Figures 16, 19, and 22) maintains high import levels but with fewer expirations, indicating limited success in minimizing imports. Overall, the SOS algorithm emerges as the most suitable for minimizing both blood imports and expirations as iterations increase, demonstrating more excellent stability and efficiency in blood unit management compared to GA and PSO.

These results are quite similar to those obtained in the study by Ezugwu *et al.* [5], which used the same dataset and parameters. The results in this study are improved because expiration was also included in the objective function, and different rhesus factors were considered. Lower levels were reported compared to their study, with significantly lower supply levels, even though an optimal solution was not found. This discrepancy between blood supply and demand arose from updating the supply with the remainder of the previous day. Additionally, the algorithms in this study required less computational time than the earlier study. However, both studies suggest that the SOS algorithm outperformed all tested algorithms.

5. Conclusion and Future Work

The optimal assignment of the blood problem presented in this paper aims to find the best solution to the global supply and demand of blood by considering factors such as blood expiration and the importation of blood during emergencies. Notably, blood is constantly in high demand worldwide, making a reliable supply essential. The heuristic algorithms used in this study effectively optimised certain critical choice variables relevant to the developed model. Overall, these algorithms made significant progress in addressing the problem.

To enhance their effectiveness, it is essential to introduce variables that account for the remaining blood from the previous day, as carrying over the remainder did not yield satisfactory results. Future work should focus on minimizing the remainder to reduce or eliminate carryover. Additionally, factors such as seasonal variations, patient demographics by supplying blood based on the demographics of the patient population, and the investigation of other algorithms should also be considered.

References

[1] J. Charpin, A. Adewumi, M. Bashe, D. Hirpa, B. Okeke, M. S. Tchamga, E. Ujeneza, Optimal assignment of blood in a blood banking system, Mathematics in Industry Study Group, University of University of KwaZulu-Natal, Westville, South Africa (2011).

[2] A. Ekaputri, R. D. Kusumastuti, Simulation of blood inventory management: Case of the indonesian red cross jakarta branch, SIMULATION, Department of Management, Faculty of Economics and Business, Universitas Indonesia 12 (2022) 31–2022.

[3] P. Govender, A. E. Ezugwu, A comparative study of metaheuristics for blood assignment problem, in: Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence, School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Westville, South Africa, 2018, pp. 1–8.

[4] M. E. Reid, C. Lomas-Francis, M. L. Olsson, The blood group antigen factsbook, Academic press, 2012.

[5] A. E. Ezugwu, O. Otegbeye, P. Govender, J. O. Odo, computational intelligence approach to dynamic blood allocation with abo-rhesus factor compatibility under real-world scenario, IEEE access, School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Westville, South Africa 8 (2020) 97576–97603.

[6] P. D. Cumming, K. E. Kendall, C. C. Pegels, J. P. Seagle, J. F. Shubsda, A collections planning model for regional blood suppliers: description and validation, Management Science 22 (1976) 962–971.

[7] E. Dufourq, M. Olusanya, A. Adewumi, Studies in metaheuristics for the blood assignment problem, Proceedings of the Operation Research Society of South Africa (ORSSA'12), School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Westville, South Africa (2012).

[8] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Computers & Structures 139 (2014) 98–112.

[9] J. H. Holland, Genetic algorithms, Scientific american 267 (1992) 66–73.

[10] E. Ozcan, C. K. Mohan, Particle swarm optimization: surfing the waves, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), volume 3, IEEE, 1999, pp. 1939–1944.