

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/259998178>

On the performance of GRASP and dynamic programming for the blood assignment problem

Conference Paper · October 2013

DOI: 10.1109/GHTC.2013.6713684

CITATIONS

2

READS

65

3 authors, including:



Kevin Igwe

University of KwaZulu-Natal

6 PUBLICATIONS 6 CITATIONS

SEE PROFILE



Aderemi Adewumi

University of KwaZulu-Natal

94 PUBLICATIONS 349 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Call for paper - The 1st symposium on Metaheuristic, Matheuristic and hyperheuristic: Theory and Applications [View project](#)



Intelligent Math-heuristics Framework for Global Optimization [View project](#)

All content following this page was uploaded by [Aderemi Adewumi](#) on 08 March 2015.

The user has requested enhancement of the downloaded file.

On the Performance of GRASP and Dynamic Programming for the Blood Assignment Problem

Kevin Igwe, Micheal Olusanya and Aderemi Adewumi
School of Mathematics, Statistics and Computer Science
University of KwaZulu-Natal
South Africa Durban 4001

Email: 212553209@stu.ukzn.ac.za, 211560532@stu.ukzn.ac.za, adewumia@ukzn.ac.za

Abstract— Blood is very precious and vital in medicine; blood transfusion occurs every day during surgery or for other reasons. However, the continuous daily demands for blood transfusion in the hospitals have exerted more pressure on blood banks. For this reason the assignment of blood at blood transfusion centers requires optimization techniques in order to have efficient allocation as to reduce wastage and meet all the demands of the patient. The constraints of acquiring blood (cost), the blood grouping into types (Rhesus factors) and the limited shelf life of blood have pose a big challenge to blood transfusion centers. This paper models blood assignment problem in form of multiple knapsack problem and implement two local search optimization techniques, dynamic programming and GRASP, that use the model to facilitate efficient assignment of red blood units from available blood types to different request for blood of different types. Therefore, the result shows that GRASP handles data more efficiently over dynamic programming.

Index Terms— Optimization, Blood Transfusion, Blood assignment

I. INTRODUCTION

Blood transfusion centers are saddle with the responsibility of making sure that blood – the right type required, in the quantity it is required and at the time it is required [1] – is made available to hospitals at real time. For this reason, there is greater need to devise more efficient method for determining what blood type (from the available compatible types) should be assigned to a particular request from a hospital. There is also greater need to schedule these assignments in such a way that the hospitals' demand are met without incurring loss in terms of wastage or expiration of blood units. Otherwise, there may be shortage of blood in the bank; forcing the bank to rely on importation from outside the blood bank. The blood assignment problem is one of many optimization problems that seek minimization of wastage, maximization of profit and/or derivation of efficient method for placing 'objects' in different 'places'. Objects in this case are the blood¹ units while places are the different requested blood types. Precisely, blood assignment problem deals with finding an efficient method for assigning available blood to patients or allocating available blood to different blood types using cross matching to minimize wastage [2] and to reduce

the amount of blood importation from outside the blood bank[3]. The shelf life of blood is very short while acquisition of blood is very expensive. Therefore the acquired blood must be treated with optimum care. Many factors that may be taking into consideration when assigning blood to patient includes the request time, the quantity of blood requested [2] the urgency of the request [2], and most importantly the compatibility of bloods[2][3][4].

Blood transfusion is almost always inevitable activity in health care system. Its impact on saving millions of lives and on improvement of quality of life for patients suffering from chronicle disease is unequivocally seen. Blood transfusion centers have revived many lives dating back since 1939 [5]. Before the knowledge and understanding that blood types are not identical, blood transfusion has however caused many negative effects. The struggle to avert these negative effects is dated back to the nineteenth century. Landsteiner [6], from his observation, named the first two blood group antigens A and B. few other antigens were observed after this. Later in 1927, to avoid confusion, he named the antigens A, B, O and AB. This is widely known as ABO blood group system. A study conducted by [7] reported that despite the recent improvement in Blood Transfusion Centers to ensure quality and healthy blood transfusion, blood donation were still complicated by adverse reaction. The most common adverse reaction is presyncopal symptoms – weakness of muscles and some parts of the body, in other words vasovagal reactions of mild intensity [7].

Blood assignment problem have been tackled in many ways in the literature. Sapountzis[5] developed a model based on probability distribution. The paper, *Allocating Blood to Hospitals*, considered some characteristics such as the hospital's management, policies in the blood bank and the doctor's preferences in ordering and using blood. The focus is on minimization of the number of expiring units of blood. The objective function was represented as a utility minimization function.

$$U = 1 - e^{-ay} \quad (1)$$

The latter y denotes the number of units that expire for that blood group. Minimum utility is achieved when no units expire, while higher utility implies high expired unit.

In the work of Angelis et al, [3] a multi-product, multi-period, multi-objective linear programming model was

¹ We adopt the system of referring to red blood cell as blood

developed and implemented to optimize blood donation – transfusion system in Rome. Angelis considered the urgency of request a very important criterion in determining the order of assignment of units of blood. The system makes use of three degree of urgency – not-urgent, urgent, and very urgent. The not urgent request can be delayed for at most eight days, the urgent for a day while very urgent request must be satisfied at real time. The model aims at optimizing the blood donation – transfusion system by minimizing the amount of waste and imports of the blood units. Each blood units has an expiry date, thus the operation of the system was scheduled to minimize the quantity of expiring units. Thus number of days a blood has stayed in the system was also considered an important factor. However, Angelis consider each blood type separately rather than all the eight blood types.

Angelis work has been used as a benchmark by Adewumi et al. [4]. However, a metaheuristic approach was implemented by Adewumi et al. The system uses Knapsack problem modeling approach to dynamically optimize the assignment of blood units while maximizing the profit. The same approach was used by Dufourq et al [2]. A knapsack problem approach is a well a known approach in optimization. It involves determining the optimal way of placing objects of different weights inside a bag in such a way that the capacity of the bag is not exceeded.

The rest of the paper is structured as follows; section 2 describes the model. The methodology is specified in section 3 while the result is discussed in section 4. Conclusion and possible improvements are specified in section 5.

II. MODELING THE BLOOD ASSIGNMENT SYSTEM

The model developed in this system is adapted from the work of Dufourq et al [2] and Adewumi et al [4]; hence the model makes use of Multiple Knapsack Problem (MKP) modeling approach. One among other important features adopted into this work is the blood cross matching which reduces the constraints exerted by Rhesus factors. The system, the optimization techniques and the assumptions made by the system are described in the following subsection.

A. The Blood Assignment System

The System assumes that the bank maintains blood unites of various types, A_{\pm} , B_{\pm} , AB_{\pm} , and O_{\pm} . It is a close resemblance of the systems developed in Adewumi et al [4]. The system, figure 1, takes a number of iteration t which represents the run for a day. It takes note of the available blood units at the beginning of each day, the donated blood units for a specific day and the number of request for the day. The available blood units are obtained from unassigned blood from the previous day added to the donated blood for the day. Thus the system gathers this information at the beginning of each day t . The request is made to the bank which in turn consults the algorithm at the back end to determine the best way to assign the available blood to the different requests for the day. It is assumed that the management issues out the blood units from the unsigned bloods of the previous day to minimize wastage. If for each day the demand cannot be met,

the blood transfusion center will result to blood importation. The quantity and type of the blood unit(s) required to be imported will be indicated by the system.

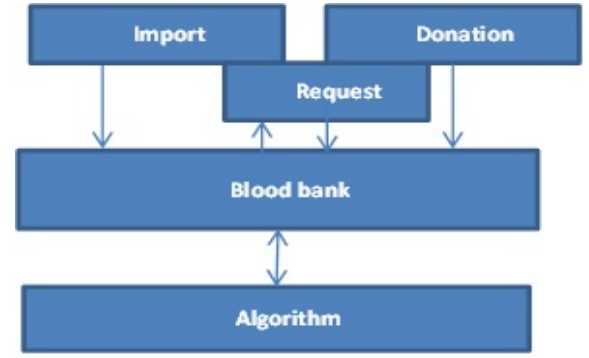


Figure 1: the Blood Assignment System

The value of the blood type used in the proposed model is calculated using the same approach in the work of Adewumi [4]. Maximization gives optimum assignment.

$$\begin{aligned}
 O^+ &= \frac{2}{27} = 0.07 & O^- &= \frac{1}{27} = 0.04 \\
 A^+ &= \frac{4}{27} = 0.15 & A^- &= \frac{2}{27} = 0.07 \\
 B^+ &= \frac{4}{27} = 0.15 & B^- &= \frac{2}{27} = 0.07 \\
 AB^+ &= \frac{8}{27} = 0.3 & AB^- &= \frac{4}{27} = 0.15
 \end{aligned}$$

The value of the blood is calculated based on the number of blood types that can donate blood to the type in consideration. The number is divided by the number of possible donations from one type to another. Thus the blood with least number indicates least choice of receiving from other blood types. Such blood type should have a higher volume of its type maintained in the system. Urgency of request is not considered.

B. The MKP approach

MKP [12][4] involves m knapsacks of various capacities c_1, c_2, \dots, c_m and n objects. Each of the n has a profit p_i ; the weight of the i^{th} object takes j values, $1 \leq j \leq m$. Thus the objective is formulated as $\sum_{i=1}^n w_{ij} x_i \leq C_j$ for $j = 1, 2, \dots, m$ that produces the maximum profit $p(\vec{x}) = \sum_{i=1}^n p_i x_i$. In the case of blood assignment problem, the blood units stored in the blood bank as n objects that must be assigned to m (knapsacks) requests of different types (capacities)

The model is formulated as below,

$$\begin{aligned}
 &\text{minimize } \sum_{t=1}^n I_{Total}(t) \\
 &\text{where } I_{Total}(t) = I_{O+}(t)I_{O-}(t) + I_{A+}(t) + I_{A-}(t) + I_{B+}(t) \\
 &\quad + I_{B-}(t) + I_{AB+}(t) + I_{AB-}(t)
 \end{aligned} \tag{2}$$

C. Representation strategy

The system represents feasible solution to assignment problem as a sequence of genes – integer values representing the blood types. This approach was inspired by the

representation use in [2]. However, the genes represents the number of blood units received from a particular type, see *vector 2*.

2	3	0	54	7	76	0	7	5	3	7	0	...	3
---	---	---	----	---	----	---	---	---	---	---	---	-----	---

Vector 1. Showing a sample of chromosome which representation the receiving part

Give	A+	A-	O+	O-	A-	O-	B+	B-	O+	O-
Re- ceive	A+	A+	A+	A+	A-	A-	B+	B+	B+	B+

B-	O-	A+	A-	B+	B-	AB+	AB-	O+
B-	B-	AB+	AB+	AB+	AB+	AB+	AB+	AB+

O-	A-	B-	AB-	O-	O+	O-	O-
AB+	AB-	AB-	AB-	AB-	O+	O+	O-

Vector 2. Interpretation of chromosome representation

The first two numbers in the chromosome represents the number of red blood units assigned from A+ to A+ and from A- to A+ respectively.

III. METHODOLOGY

Two local search single-solutions based heuristic, Dynamic Programming and Greedy Randomize Adaptive Search Procedure (GRASP), were used in an attempt to optimize the blood assignment in the blood bank. Heuristics are known to perform than brute force method especially in combinatorial optimization problems. The implementation strategies used are discussed below.

A. Dynamic Programming

Dynamic programming is used to solve combinatorial optimization problem by dividing the problem variables into a family of sub- problems that differs one from another in the- hand side [8]. For the case of the blood assignment problem using MKP, the value at the left hand side must be less than or equal to the value at the right hand side. Another adjustment done on the model in other to adapt dynamic programming technique is to relax the constraints specified by typical multiple Knapsack problem definition that an object is either placed totally into the knapsack or not placed at all if the capacity of the knapsack is exceeded. This is specified as $x_{ij} \in \{0,1\}$. $x_{ij} = 1$, implies object i is placed into knapsack j while $x_{ij} = 0$ implies that object i is not in knapsack j . Relaxed constraint is specified as $0 \leq x_{ij} \leq 1$ which implies that fraction of the object (available blood) is allowed into the knapsack (Request of different types).

B. Greedy Randomized Adaptive Search Procedure (GRASP)

GRASP is an optimization technique that seeks combination of greedy choice towards best solution and randomness of search in other to explore the neighborhoods. GRASP can be implemented as being totally greedy; thus always choose the best solution. The greediness can also be reduced using a parameter, α , which determines the level of greediness of the algorithm. Thus, it can be total random, $\alpha = 0$, or totally greedy, $\alpha = 1$. GRASP algorithm have a construction phase and optional Local Search phase.

For the case of blood assignment, a purely greedy algorithm was implemented. The algorithm is given below.

Start: GRASP(numberOfIteration)
Solution \leftarrow *ConstructionPhase()*
LSsolution \leftarrow *LocalSearchPhase(Solution)*
Solution \leftarrow *updateSolution(solution,LSsolution)*
numberOfIteration reached
stop

A local search operator was defined to alter the genes in order to improve the solution.

1) The Local search operator

The local search operates is based on the level of importation of each blood type, see *Vector 3* below. The operator randomly chooses an index in the range of the chromosome length. The value of the blood type which donated the quantity of blood at the specified index is determined. Another index with the corresponding blood type is chosen. If blood importation is required at any of the receiving ends of the two chosen index (same blood type), the quantity of a blood is randomly chosen from 0 to the number of required importation. This quantity is added to the end that requires importation and subtracted from the other end. The subtraction is done in other not to violate the constraints. In the vector 3, it can be seen from the interpretation of chromosome representation given in vector 2, the two chosen index, 1 and 4 correspond to blood of type A-, donating blood to A+ and A- respectively. Assuming A+ requires importation of 3 more blood units, a random number is generated between 0 and 3 (inclusive) – say 2 is generated. The number is then added to the index 1 and also subtracted from the quantity in the index position 4. Thus resulting in the chromosome represented in *vector 4*. This operation is carried out randomly a number of time before the chromosome is return for fitness evaluation.

2	3	0	54	7	76	0	7	5	3	7	0	...	3
---	---	---	----	---	----	---	---	---	---	---	---	-----	---

Vector 3. Showing the operation of the local search

2	5	0	54	5	76	0	7	5	3	7	0	...	3
---	---	---	----	---	----	---	---	---	---	---	---	-----	---

Vector 4. Showing the result of the local search

Algorithm	DataSet	A+	A-	B+	B-	AB+	AB-	O+	O-
DP	Data 1	0	2	0	8	0	0	179	227
	Data 2	0	2	0	8	0	0	707	743
	Data 3	0	9	0	9	0	5	1381	2059
	Data 4	0	0	0	4	1	0	143	604
	Data 5	0	34	0	17	0	0	2219	1825
GRASP	Data 1	0	0	0	4	0	0	474	478
	Data 2	0	49	0	23	0	0	0	1025
	Data 3	0	0	0	4	0	0	1772	1758
	Data 4	0	0	0	4	0	0	428	618
	Data 5	0	22	0	7	0	0	335	1129

Table 1: comparison of the algorithm

IV. RESULT AND DISCUSSION

Simulation was performed on a regular PC. The program were written in Java and run on Netbeans IDE 7.2 environment. The program assume a 90 days of test run. The data used for the model was randomly generated using the same specification found in [4]. For Dataset 1-3, the amount of requests and donations are both equal to randomly generated amounts between 25-75% of the initial volumes of each blood type. For the case of Data 4, the amounts of Donations are randomly generated between 30-70% of the initial volume while the requests are randomly generated between 25-75% of the initial volume.

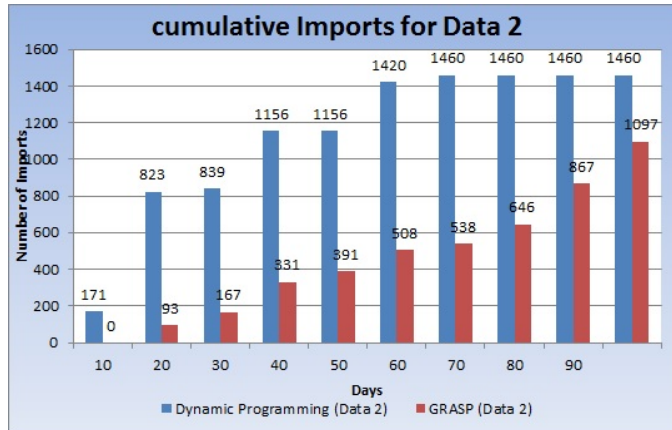


Figure 2: level importation for each blood type using both the Dynamic Programming and GRASP algorithm on data 2.

Data 5 uses 30-75% for Requests and 25-75% for donations. The initial volumes are 500,1000,2000,1000, and another 1000 for data 1,2,3,4 and 5 respectively. Figure 2 shows the amount of importation for each blood type using data 2 on both algorithms. The result is a positive one since the bloods with low values are not being imported. However, the importation of the blood units of type O+ and O- is very high compared to that in the algorithm implemented in Adewumi et al [4], see figure 4.

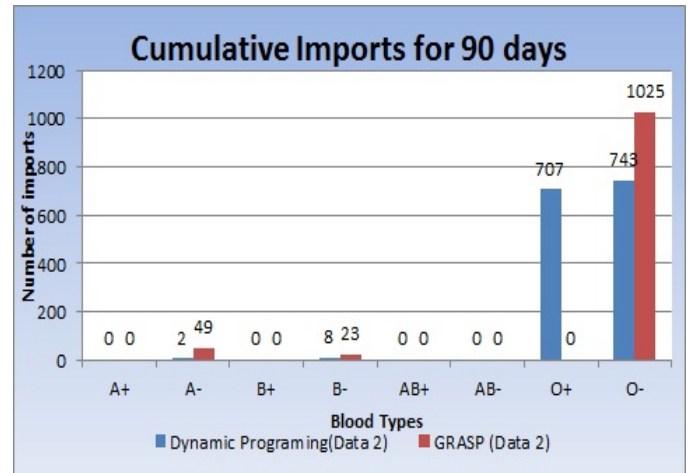


Figure 3: cumulative imports from day 1 to 90 showing performance of both the Dynamic Programming and GRASP algorithms.

Dataset	Model	Time	O+	O-	A+	A-	B+	B-	C+	C-	Total
Data 1	Simple Assignment	0.265	10	116	55	51	104	19	21	3	379
	MKA (GA)	89.069	69	175	0	37	0	10	0	0	291
	MKA (AGA)	90.872	61	183	0	37	0	10	0	0	291
Vol=500	MKA (SAGA)	91.453	64	178	0	37	0	10	0	0	289
	MKA (ASAGA)	93.888	62	180	0	37	0	10	0	0	289
	MKA (HC)	0.903	61	182	0	38	0	10	0	0	291
Data 2	Simple Assignment	0.219	83	180	414	0	428	36	0	11	1152
	MKA (GA)	321.559	66	345	0	8	0	15	0	0	434
	MKA (AGA)	327.209	66	345	0	8	0	16	0	0	435
Vol=1000	MKA (SAGA)	321.166	66	345	0	8	0	16	0	0	435
	MKA (ASAGA)	330.206	66	346	0	8	0	14	0	0	434
	MKA (HC)	2.295	66	340	0	8	0	17	0	0	431
Data 3	Simple Assignment	0.627	0	86	824	121	1715	32	0	5	2783
	MKA (GA)	1315.915	663	520	234	79	175	89	0	0	1760
	MKA (HC)	8.067	643	522	246	72	176	93	0	0	1752
Data 4	Simple Assignment	0.575	0	12	183	54	0	2	0	0	251
Vol=1000	MKA (ASAGA)	344.292	0	19	0	0	0	0	0	0	19
Data 5	Simple Assignment	0.575	664	71	1278	155	703	83	32	36	3022
Vol=1000	MKA (ASAGA)	367.361	1492	405	441	179	271	76	0	4	2868

Figure 4: Results from Adewumi et al [4]

The result shows that Dynamic programming imports blood of type O+ and O- heavily within the first 50 days before being stable while GRASP imports gradually as the days increases. One significant advantage shown on GRASP over Dynamic Programming is that it handles Data 5 more efficiently. Thus if request is more than donation, the algorithm will handle the situation better than dynamic programming. However, the overall result when compared to the population based algorithm implemented in Duforq et al [2] and Adewumi et al [4] shows that the population based algorithm balances the importation; thus fewer burdens on blood type O-.

V. CONCLUSION

In this paper, we have shown the performance of the two local search algorithm implemented. The result is a positive one since only the blood units which can be used as universal donor are being imported. Improvement can be done in this area by hybridizing the GRASP and the developed local search algorithm with population based optimization technique or Tabu search.

REFERENCES

- [1] V. Hemmelmayr, K. F. Doerner, R.F. Hartl, and M.W.P. Savelsbergh, Delivery Strategies for Blood Products Supplies *OR Spectrum, Volume 31, Number 4, October 2009* , pp. 707-725(19)
- [2] E. Dufourq, M.O Olusanya, and A.O Adewumi, "Studies in Metaheuristics for Blood Assignment Problem," in *the Proceedings of the 2012 ORRSA Annual Conference* i6 -19 September 2012, pp 107-116
- [3] D. V Angelis, N. Ricciardri, and G. Storchi, (2001). "Optimizing Blood Assignment in a Donation Transfusion System". *International Transactions in Operational Research*. 8(2):183- 192
- [4] A. Adewumi, N. Budlender and M.O. Olusanya, "Optimizing the Assignment of Blood in a Blood Banking System: Some Initial Results," in *the Proceedings of IEEE World Congress on Computational Intelligence* June, 10-15, 2012 - Brisbane, Australia.
- [5] C. Sapountzis, (1989, May), Allocating Blood to Hospitals *The Journal of the Operational Research Society*, Vol. 40, No. 5 (May, 1989), pp. 443-449 Available: <http://www.jstor.org/stable/2583616> .
- [6] M. E. Reid and C. Lomas-Francis, "The Blood Group Antigen Fac-Book," Elsevier Ltd 2004,
- [7] C. Pathak, M. Pujani, S. Pahuja, and M. Jain Adverse reactions in whole blood donors: an Indian scenario *PMCJournal Blood Transfus*. 2011 January; 9(1): 46–49.
- [8] C. Wilbaut, S. Hanafi, A. Freville and S. Balev., *Tabu search: global intensification using dynamic*. Contro and Cybernetics. Vol. 35 (2006) N0. 3