Simulated annealing with restart strategy for the blood pickup routing problem

V F Yu*, T Iswari1,2,3, N M E Normasari1,2, A M S Asih2 and H Ting4
1Department of Industrial Management, National Taiwan University of Science and Technology, Taipei, Taiwan
2Department of Mechanical and Industrial Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia
3Program Studi Teknik Industri, Universitas Katolik Parahyangan, Bandung, Indonesia
4Department of Information and Finance Management, National Taipei University of Technology, Taipei, Taiwan

*vincent@mail.ntust.edu.tw

Abstract. This study develops a simulated annealing heuristic with restart strategy (SA_RS) for solving the blood pickup routing problem (BPRP). BPRP minimizes the total length of the routes for blood bag collection between a blood bank and a set of donation sites, each associated with a time window constraint that must be observed. The proposed SA_RS is implemented in C++ and tested on benchmark instances of the vehicle routing problem with time windows to verify its performance. The algorithm is then tested on some newly generated BPRP instances and the results are compared with those obtained by CPLEX. Experimental results show that the proposed SA_RS heuristic effectively solves BPRP.

1. Introduction
The need for blood rapidly increases recently. However, the supply of blood steadily decreases [1] With these trends, it is necessary to find an effective and efficient way to manage blood supply chain. In the US, operation of blood service includes collection, storage and processing of blood and its components[2].

Blood collection involves two essential components: donors (suppliers) and blood bank (depot). In the collection process, to take a pint of blood (450-ml specialized package) from a donor takes about 10 minutes. After the collection, the blood samples are sent to the blood bank via vehicles equipped with temperature-controlled boxes as a temporary storage while the blood samples are in transit. Donated blood must be delivered to a blood processing center, called blood bank, within 6 hours. Thus an efficient vehicle routing to pickup the blood is needed.

The efficiency of blood collection planning/supply center and related issues have been studied [3] since 1970s, but routing aspects of the problem have rarely been studied. Therefore, this research introduces the Blood Pickup Routing Problem (BPRP), which is an extension of the well-known Vehicle Routing Problem with Time Windows (VRPTW). BPRP covers not only the typical characteristics of VRPTW but also problem rich constraints such as the spoilage time of blood and time restrictions at the depot.
Since VRPTW is NP-hard, BPRP is also NP-hard. Therefore, exact solution approaches may not be suitable for practical cases. Considering the success of Simulated Annealing (SA) on solving hard combinatorial optimization problems [4], this study develops a meta-heuristic based on Simulated Annealing (SA) for solving BPRP of practical sizes. More specifically, this research modified the Simulated Annealing with Restart Strategy (SA_RS) of Lin and Yu [11] to solve BPRP. SA has been successfully applied to many vehicle routing related problems [12],[13],[7],[8],[14],[11], including BPRP [10]. Therefore, this study improves the SA for BPRP by incorporating the restart strategy to avoid getting trapped in local optimum by exploring larger portion of search space.

The rest of this paper is organized as follows. Section 2 reviews relevant literature. Section 3 describes the problem. The proposed SA_RS is discussed in Section 4. Section 5 presents computational study. Section 6 provides conclusions and future research directions.

2. Literature Review
There are not many studies devoted to blood collection planning[15]. Blood collection planning is firstly studied by Cumming et al. [16]. They studied the forecasting of blood collection by developing population based Markovian model to decrease the gap between demand and supply of blood. Yi [2] introduced the vehicle routing aspect on blood issue. With the federal’s regulation on blood goods, the study seeks an efficient way to decide which sites to shuttle blood and how these vehicles should be routed to provide a sufficient amount of platelets to meet daily production requirements at the minimum transportation cost possible. The computational results reveal significant reduction of transportation cost of nearly 60%, on average. Doerner, et al. [17] study the blood collection process of the Austrian Red Cross blood program and address multiple pickups at single site. Similar to Yi [2], they assume known and constant collection with the goal to find minimum cost tours. The resulting cost reduction ranges from 6% to 12% depending on the problem complexity.

Literature on heuristic approaches for VRPTW is extensive. Recently, Rabbani et al. [18] develop a Genetic Algorithm (GA) to solve VRP with multi-middle depots for perishable food delivery. The results prove that the GA is promising. Song and Ko [19] propose a priority-based heuristic (PBH) to find an efficient route to deliver perishable food product. The priority in PBH is computed based on distance and density of the node. The result shows that PBH can solve the problem efficiently. Rabbani, et al. [20] develop and compare GA and SA on solving VRPTW for delivering perishable products. The results reveal that SA produces better solutions than GA. Osvald and Stirn [21] develop a heuristic based on tabu search to solve fresh vegetables distribution. The result shows that the algorithm can improve the solution up to 47%. Doerner et al. [17] propose a heuristic based on the savings and greedy procedure to solve VRP with time windows. The results show that the heuristic find solutions in a reasonable time.

3. Problem Description
BPRP is characterized by a network consisting of a blood processing facility (depot) and fixed donation sites (customers) accumulating donated blood. Donation sites to be visited are pre-determined and assumed to produce a fixed amount of blood bags. The blood bank is responsible for managing the collection process which includes dispatching of shuttles to the donation sites and bringing the donated blood back to the blood bank for further processing.

The perishability of blood must be considered for practical purpose. The requirement as to the age of the blood bags when they reach the blood bank varies from blood bank to blood bank depending on the facility’s requirement. But, a standard age of at most 6 hours should be adhered [2].

The objective of the BPRP is to find the set to minimum cost routes to bring back all collected blood bags to the blood bank without waste, that is, within the shelf-life. Moreover, BPRP has the following typical VRPTW attributes and assumptions:

1. Each donation site can be visited within a specific time window pertaining to the opening and ending time of the donation event;
2. Supply at each site is deterministic;
3. The blood bank operates within a specific time window as well;
4. The blood bank administers the scheduling of the shuttles which are assumed to have homogeneous capacity;
5. Total amount of supplies for collection in a route should not exceed the capacity of the shuttle;
6. Each donation site can be visited only once;
7. Service time is assumed to be negligible.
8. Age of the blood starts at the time of donation.

More formally, BPRP can be defined on a complete graph \( G = (N, A) \) where \( N = \{v_0, v_1, \ldots, v_n\} \) is the node set and \( A = \{(v_i, v_j): v_i, v_j \in N, i \neq j\} \) is the arc set. Node \( v_0 \) represents the depot and the remaining nodes in \( N \) represent a subset \( N' \) of supplier nodes (donation sites). Each supplier node is associated with a supply \( p_i \) and a time window \([o_i, e_i]\). Each node operates from \( o_i \) to \( e_i \). Each vehicle route is restricted with a maximum allowable route length, \( T_{\text{max}} \), which represents the spoilage of the blood. Each arc \((v_i, v_j): i \neq j\) is associated with a finite travel time \( t_{ij} \). The goal of BPRP is to find a set of least-cost routes to collect total supply produced at each node using a fleet of homogenous vehicles.

4. SA_RS for BPRP

SA is popular heuristic for solving combinatorial optimization problems. SA is first proposed by Metropolis, et al. [22]. SA usually begins by randomly generated an initial solution. Then a new solution is taken from the neighborhood of the current solution. The new solution replaces the current solution if it is better, and then the process continues from the new current solution. SA accepts a worse solution with small probability. Thus, SA can escape to be trapped at local optima. To increase the diversity of SA to explore a larger solution space, this study develops the SA with restart strategy. Restart strategy is a well-known strategy to diversify SA procedure [11]. The following sections give more detailed explanation of the SA_RS for BPRP.

4.1. Initial Solution

The initial solution for SA_RS is generated by the Nearest Neighborhood algorithm [23] The vehicle starts from depot to visit the nearest unvisited customer, then continue to the next nearest unvisited customer until all customers are visited. The vehicle then returns to the depot. The vehicle will return to the depot when the total collected blood exceeds the capacity of the vehicle, the arrival time violates time window or the age of blood exceeds the spoilage time. The algorithm then continues to construct a new route and performs the same method until all customers have been visited.

4.2. Neighborhood

SA_RS uses a random neighborhood structure that includes swap, insert, and reverse operations. The swap operation is executed by randomly selecting two points in different routes of the current solution and then swapping their positions. The insertion operation inserts a randomly selected point from one route into a randomly selected position of another route of the current solution. The reverse operation is performed by randomly selecting two points from a route in the current solution and then reversing the sequence between them, including those two selected points. The probabilities of choosing swap, insert, and reverse operations are 0.3, 0.4, and 0.3, respectively. These probabilities are determined based on an preliminary experiment.

4.3. SA_RS Parameters

The SA with restart strategy begins with five parameters \( I_{\text{iter}}, T_0, N_{\text{non-improving}}, \text{Max_restart}, \) and \( \alpha \). \( I_{\text{iter}} \) denotes the number of iterations the search proceeds at a particular temperature. \( T_0 \) represents the initial temperature. \( N_{\text{non-improving}} \) is the maximum allowable number of reductions in temperature during which the best objective function value is not improved. The algorithm will restart at the initial temperature if \( N_{\text{non-improving}} \) is reached. \( \text{Max_restart} \) is the maximum number of restarts. Finally, \( \alpha \) is the coefficient controlling the cooling mechanism. Boltzmann function is used to calculate the probability of acceptance.
4.4. SA_RS Procedure
This study implements the restart strategy to prevent the solution being trapped in the local solution. The restart strategy allows the algorithm to restart from the initial temperature if the number of non-improvement solution have been reached in the temperature reductions. This mechanism helps the algorithm to explore more in the solution area. Figure 1 shows the flowchart of SA_RS algorithm.

![Flowchart of SA_RS](image)

**Figure 1.** Flowchart of SA_RS.

5. Computational Results
The proposed SA_RS is implemented in C++ and tested on small BPRP instances to evaluate its performance.

5.1. Parameter Setting
The best parameter combination is determined by a two-level ($2^k$) factorial design, which has been widely used to determine parameter setting for heuristics [24]. The initial parameters used for the pilot experiment are: $I_{iter} = 3000$, $\alpha = 0.95$, $T_0 = 10$, $N_{non-improving} = 100$, and $Max_{restart} = 10$. The best parameters for BPRP are: $I_{iter} = 9000$, $\alpha = 0.95$, $T_0 = 10$, $N_{non-improving} = 200$, and $Max_{restart} = 10$. These parameter values are used for all computational study hereafter.

5.2. Results on VRPTW Instances
Table 1 compares the performances of SA and SA_RS on solving in Solomon’s VRPTW instances. SA_RS provides better solution for all instances. The average percentage gap is -1.80% and the largest gap is -11.67% for instance C203. However, SA_RS needs slightly longer computational time than SA. This shows that the proposed SA_RS can solve VRPTW effectively.

**Table 1. Comparison basic SA and SA_RS in VRPTW.**

| Instance | SA | SA_RS | CPU Time (s) | CPU Time (s) | Gap (%) |
|----------|----|-------|--------------|--------------|--------|
| C101     | 828.94 | 50.24 | 828.94 | 57.73 | 0.00% |
| C102     | 828.94 | 49.26 | 828.94 | 59.28 | 0.00% |
| C103     | 828.06 | 57.36 | 828.06 | 59.39 | 0.00% |
| C104     | 824.78 | 49.82 | 824.78 | 61.09 | 0.00% |
| C105     | 828.94 | 48.95 | 828.94 | 55.89 | 0.00% |
| C106     | 828.94 | 49.19 | 828.94 | 55.69 | 0.00% |
| C107     | 828.94 | 49.27 | 828.94 | 56.21 | 0.00% |
| C108     | 828.94 | 48.43 | 828.94 | 54.43 | 0.00% |
| C109     | 828.94 | 48.36 | 828.94 | 53.98 | 0.00% |
| C201     | 621.64 | 63.35 | 591.56 | 47.67 | -4.84% |
| C202     | 621.64 | 62.87 | 591.56 | 54.35 | -4.84% |
| C203     | 669.27 | 54.55 | 591.17 | 49.13 | -11.67% |
| C204     | 625.06 | 72.63 | 591.17 | 52.37 | -5.42% |
| C205     | 626.63 | 65.59 | 588.88 | 54.19 | -6.02% |
| C206     | 621.01 | 68.49 | 588.49 | 53.07 | -5.24% |
| C207     | 588.29 | 62.98 | 588.29 | 47.34 | 0.00% |
| C208     | 588.32 | 55.17 | 588.32 | 49.55 | 0.00% |
| R101     | 1672.08 | 61.27 | 1649.86 | 72.46 | -1.33% |
| R102     | 1491.22 | 59.73 | 1487.32 | 81.52 | -0.26% |
| R103     | 1243.6 | 68.71 | 1229.09 | 71.58 | -1.17% |
| R104     | 1030.59 | 59.07 | 1014.22 | 70.89 | -1.99% |
| R105     | 1405.99 | 65.12 | 1389.93 | 78.24 | -1.14% |
| R106     | 1275.85 | 77.59 | 1273.43 | 69.39 | -0.19% |
| R107     | 1118.63 | 80.08 | 1092.19 | 72.72 | -2.36% |
| R108     | 993.45 | 68.31 | 973.49 | 73.74 | -2.01% |
| R109     | 1186.79 | 60.76 | 1177.44 | 68.70 | -0.79% |
| R110     | 1134.84 | 61.28 | 1113.68 | 76.10 | -1.87% |
| R111     | 1103.16 | 61.04 | 1076.42 | 72.96 | -2.42% |
| R112     | 985.71 | 61.88 | 971.26 | 73.23 | -1.47% |
| R201     | 1199.58 | 64.64 | 1182.07 | 72.67 | -1.46% |
| R202     | 1078.94 | 57.94 | 1063.38 | 69.14 | -1.44% |
| R203     | 914.44 | 56.23 | 912.24 | 64.65 | -0.24% |
| R204     | 756.45 | 74.43 | 743.50 | 68.34 | -1.71% |
| R205     | 1012.95 | 65.66 | 1006.89 | 63.84 | -0.60% |
| R206     | 931.91 | 70.11 | 911.40 | 62.41 | -2.20% |
| R207     | 847.02 | 68.13 | 809.36 | 68.33 | -4.45% |
| R208     | 732.74 | 54.75 | 727.52 | 57.36 | -0.71% |
| R209     | 908.19 | 56.02 | 890.01 | 69.12 | -2.00% |
| R210     | 920.47 | 56.71 | 920.47 | 71.68 | 0.00% |
| R211     | 795.21 | 63.99 | 769.19 | 65.93 | -3.27% |
| RC101    | 1690.83 | 63.20 | 1673.24 | 70.92 | -1.04% |
| RC102    | 1506.78 | 66.31 | 1501.96 | 73.17 | -0.32% |
| RC103    | 1355.85 | 78.58 | 1297.57 | 69.20 | -4.30% |
| RC104    | 1200.85 | 73.95 | 1172.10 | 80.78 | -2.39% |
| RC105    | 1585.61 | 69.69 | 1547.43 | 79.59 | -2.41% |
| RC106    | 1435.31 | 64.53 | 1396.60 | 72.14 | -2.70% |
| RC107    | 1285.04 | 61.90 | 1258.94 | 71.25 | -2.03% |
| RC108    | 1154.92 | 62.71 | 1147.26 | 70.09 | -0.66% |
| RC201    | 1314 | 58.80 | 1294.99 | 69.85 | -1.45% |
| RC202    | 1127.6 | 69.40 | 1123.55 | 64.22 | -0.36% |
5.3. Results on BPRP Instances

Since BPRP is a new problem, we generate 10 small BPRP benchmark instances from classical VRP benchmark instances. The depot and the first 20 customers of the VRP instances are used to generate BPRP instances, along with their coordinates, and the customer demands. Time windows, vehicle capacity, and spoilage time are then added to the instances. We set the time window for all nodes to be [60, 660]. The time window represents the opening and closing time of the nodes. Vehicle capacity ranges from 50 to 140 to maintain the feasibility. The spoilage time is set to be 360s. CPLEX solver is used to solve these instances for comparison purpose. The maximum running time of CPLEX is 5 hours.

To evaluate the solution quality of the proposed SA_RS, this research compares results from both SA and SA_RS to the CPLEX results. As can be seen in Table 2, both SA and SA_RS find optimal solutions to 5 out of the 10 instances, and better solutions for the remaining instances. The comparative results reveal that the proposed algorithms can provide good BPRP solutions in a reasonable amount of time.

Table 2. Small Comparative Results on BPRP Instances.

| Instance | SA | SA_RS |
|----------|----|-------|
| RC203    | 989.44 | 62.55 | 962.98 | 67.16 | -2.67% |
| RC204    | 819.84 | 59.31 | 807.92 | 63.69 | -1.45% |
| RC205    | 1206.34 | 72.89 | 1185.98 | 66.90 | -1.69% |
| RC206    | 1109.64 | 66.39 | 1096.09 | 65.45 | -1.22% |
| RC207    | 1005.02 | 68.78 | 994.20 | 67.16 | -1.08% |
| RC208    | 831.75 | 68.54 | 812.13 | 65.66 | -2.36% |
| Average  | 1013.86 | 62.46 | 997.00 | 65.14 | -1.80% |

5.3. Results on BPRP Instances

Since BPRP is a new problem, we generate 10 small BPRP benchmark instances from classical VRP benchmark instances. The depot and the first 20 customers of the VRP instances are used to generate BPRP instances, along with their coordinates, and the customer demands. Time windows, vehicle capacity, and spoilage time are then added to the instances. We set the time window for all nodes to be [60, 660]. The time window represents the opening and closing time of the nodes. Vehicle capacity ranges from 50 to 140 to maintain the feasibility. The spoilage time is set to be 360s. CPLEX solver is used to solve these instances for comparison purpose. The maximum running time of CPLEX is 5 hours.

To evaluate the solution quality of the proposed SA_RS, this research compares results from both SA and SA_RS to the CPLEX results. As can be seen in Table 2, both SA and SA_RS find optimal solutions to 5 out of the 10 instances, and better solutions for the remaining instances. The comparative results reveal that the proposed algorithms can provide good BPRP solutions in a reasonable amount of time.
[1] Lee W-C and Cheng B-W 2011 An intelligent system for improving performance of blood donation. Journal of Quality 18(2) 173-185
[2] Yi J 2003 Vehicle routing with time windows and time-dependent rewards: A problem from the american red cross. Manufacturing & Service Operations Management 5(1) 74-77
[3] Dumas M B and Rabinowitz M 1977 Policies for reducing blood wastage in hospital blood banks. Management Science 23(10) 1124-1132
[4] Alfa A S, Heragu S S and Chen M 1991 A 3-opt based simulated annealing algorithm for vehicle routing problems. Computers & Industrial Engineering 21(1-4) 635-639
[5] Van Breedam A 1995 Improvement heuristics for the vehicle routing problem based on simulated annealing. European Journal of Operational Research 86(3) 480-490
[6] Kuo Y 2010 Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. Computers & Industrial Engineering 59(1) 157-165
[7] Yu V F, Lin S-W, Lee W and Ting C-J 2010 A simulated annealing heuristic for the capacitated location routing problem. Computers & Industrial Engineering 58(2) 288-299
[8] Lin S-W, Yu V F and Lu C-C 2011 A simulated annealing heuristic for the truck and trailer routing problem with time windows. Expert Systems with Applications 38(12) 15244-15252
[9] Xiao Y, Zhao Q, Kaku I and Xu Y 2012 Development of a fuel consumption optimization model for the capacitated vehicle routing problem. Computers & Operations Research 39(7) 1419-1431
[10] Iswari T, Yu V F and Asih A M S 2016 Simulated annealing for the blood pickup routing problem. International Journal of Information and Management Sciences 27(4) 317-327
[11] Lin S-W and Yu V F 2015 A simulated annealing heuristic for the multiconstraint team orienteering problem with multiple time windows. Applied Soft Computing 37 632-642
[12] Lin S-W, Ying K-C, Lee Z-J and Hsi F-H. Applying simulated annealing approach for capacitated vehicle routing problems. 2006 IEEE International Conference on Systems, Man and Cybernetics, 2006, 639-644.
[13] Lin S W, Yu V F and Chou S Y 2009 Solving the truck and trailer routing problem based on a simulated annealing heuristic. Computers & Operations Research 36(5) 1683-1692
[14] Lin S W and Yu V F 2012 A simulated annealing heuristic for the team orienteering problem with time windows. European Journal of Operational Research 217(1) 94-107
[15] Beliën J and Forcé H 2012 Supply chain management of blood products: A literature review. European Journal of Operational Research 217(1) 1-16
[16] Cumming P D, Kendall K E, Pegels C C, Seagle J P and Shubsda J F 1976 A collections planning model for regional blood suppliers: Description and validation. Management Science 22(9) 962-971
[17] Doerner K F, Gronalt M, Hartl R F, Kiechle G and Reimann M 2008 Exact and heuristic algorithms for the vehicle routing problem with multiple interdependent time windows. Computers & Operations Research 35(9) 3034-3048
[18] Rabbani M, Farshbaf-Geranmayeh A and Haghjoo N 2016 Vehicle routing problem with considering multi-middle depots for perishable food delivery. Uncertain Supply Chain Management 4(3) 171-182
[19] Song B D and Ko Y D 2016 A vehicle routing problem of both refrigerated-and general-type vehicles for perishable food products delivery. Journal of Food Engineering 169 61-71
[20] Rabbani M, Ramezankhani M-J, Farrokhi-Asl H and Farshbaf-Geranmayeh A 2015 Vehicle routing with time windows and customer selection for perishable goods. International Journal of Supply and Operations Management 2(2) 700-719
[21] Osvald A and Stirn L Z 2008 A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. Journal of food engineering 85(2) 285-295
[22] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E 1953 Equation of state calculations by fast computing machines. The journal of chemical physics 21(6) 1087-1092
[23] Pop P C, Zelina I, Lupşe V, Sitar C P and Chira C 2011 Heuristic algorithms for solving the generalized vehicle routing problem. *International Journal of Computers Communications & Control* 6(1) 158-165

[24] Coy S P, Golden B L, Runger G C and Wasil E A 2001 Using experimental design to find effective parameter settings for heuristics. *Journal of Heuristics* 7(1) 77-97

**Acknowledgments**

This research was partially supported by the Ministry of Science and Technology of the Republic of China (Taiwan) under grants MOST 103-2221-E-011-062-MY3 and MOST 106-2410-H-011-002-MY3. This support is gratefully acknowledged.