Simulated annealing algorithm for solving pickup and delivery problem with LIFO, time duration and limited vehicle number

A Andriansyah¹*, N Prasanti², and H Y Sastra³

¹Laboratory of Industrial Computation and Optimization, Industrial Engineering, Universitas Syiah Kuala, Jl. Tgk. Syech Abdul Rauf No.7 Banda Aceh 23111 Aceh, Indonesia
²Industrial Engineering Department, Universitas Teuku Umar, Meulaboh, Indonesia
³Industrial engineering, Universitas Syiah Kuala, Jl. Tgk. Syech Abdul Rauf No.7 Banda Aceh 23111 Aceh, Indonesia
*E-mail: andriansyah@unsyiah.ac.id

Abstract. Nowadays, the process of delivering goods or documents from one place to another can be done faster by utilizing a courier service. This courier must be able to create a travel route to pick goods and deliver them with a short time. Route creation will be more difficult if there is a time window for each customer and the goods being transported are heavy or dangerous items. In this case, the courier company must implement a last-in-first-out rules. In addition, with the limitation of vehicles number, companies have to minimize the number of underserved customers. This problem called pickup and delivery problem with last-in-first-out, time duration, and limited vehicle number. In this study, to solve the problem, simulated annealing algorithm was used for some dataset. To explore the solution space for the problem, this algorithm will use pair relocation operator. In several experiments, the results show a good solution and short computation time.

1. Introduction

Vehicle Routing Problem (VRP) is the problem determining best routes with the aim of obtaining minimal transportation costs. Many VRP variants that have been developed in recent decades, for example VRP with time windows, VRP pickup and delivery and several other variants [1].

This study will discuss one of the VRP variants that are often encountered in life, namely the Pickup and Delivery Problem (PDP). This variant can usually be found in companies engaged in pickup and delivery services (door-to-door). This problem can be defined as follows: there is a set of customer locations that have the same pickup location and delivery location. Each location will be visited right once by one vehicle. Each transport vehicle assigned has a certain maximum capacity that must not be violated. Many studies discuss VRP with the characteristics described. For surveys on PDP, it is recommended to read [2], [3] and [4] who classify these problems in several types.

PDP has its variant. Technically, if the assigned vehicle only has a single door for loading-unloading access, it can be another problem. The problem is about the rules of goods that will be sent first. If the last transported item is a hazardous material, fragile, too heavy, it is prioritized to deliver the item first [5]. If there are no delivery rules, it might be done, but if you consume too much time because you have to do the demolition, this should be avoided [6]. This rule is called Last-In-First-Out (LIFO). An example of a route that shows this rule can be seen in figure 1(a), Figure 1(b) does not comply with the rules because the last item taken is goods from location P2, then the vehicle must prioritize location D2,
not location D1. In this paper, the problem is limited by the maximum duration for each vehicle traveling because usually the companies that deal with ensure the maximum time to arrive at the destination. It is called Pickup and Delivery Problem with LIFO and Time Duration (PDPLT).

![Diagram of PDPLT](image)

**Figure 1.** (a) PDP with LIFO policy, (b) PDP without LIFO policy

The assumption of PDP is unlimited number of vehicles and all customers can be served by the vehicles [6]-[8]. In fact, there is no company that have unlimited number of vehicles. This issue affected that not all customers can be served by the vehicles. Therefore, there should be best planning to minimize the number of unserved customer. [9] has discussed about this. The problem is called Pickup and Delivery Problem with LIFO, Time Duration and Limited Vehicle Number (PDPLTV). Mathematical formulation has been developed in that study.

Simulated annealing (SA) algorithm has been created for solving PDPLTV. This is a contribution of this study. For literature review, many algorithm has been developed for solving PDP. [10] developed Variable Neighborhood Search (VNS) algorithms for these problems and successfully implemented on several data samples with good performance. [5] developed the Branch and Bound algorithm for the same problem, [11] developed branch and cut algorithms and successfully completed sample data with up to 50 customer locations. The VNS algorithm for this problem with a successful tree representation resolves the TSPPDL problem developed by [12]. The same problem but the existence of distance constraints and solved by the two-stage Simulated Annealing (SA) algorithm with the VNS-Tabu Search (TS) developed by [13]. If the problem allows the LIFO rule to be violated, it is called the Traveling with Problem Pickup, Delivery, and Handling Cost (TSPD), but is subject to a penalty cost for the violation that has been committed. [14] developed branch and cut exact algorithms for these problems. The metaheuristic approach was developed by [15] uses the TS algorithm and successfully solves the problem.

The problems most similar to the research in this paper were examined by [6], but did not consider the limited vehicle number and resolved using the TS algorithm. [7] discusses the problems considering time windows, but does not limit the time duration for each vehicle and solve by branch and cut and price algorithm. In the following year [16] developed the Genetic Algorithm (GA) -Local Search algorithm (LS) to solve large-scale cases. The last paper that discussed about PDPLTV is [9], but solved by mathematical model that implemented in optimization software.

2. Methods

2.1. Model Formulation

A set of nodes $N$ and a set of arcs $A$ that incorporated in a network $G$. $N = O \cup P \cup D$ which $O = \{0, 2n + 1\}$is the depot, $P = \{1, 2, ..., n\}$ is pickup nodes and $D = \{n + 1, ..., 2n\}$ is delivery node. Each node $i \in P$ has a request $q_i > 0$ that should be sent to $(i + n) \in D$. That is represented by request $q_{i+n} = -q_i$. The opening time $e_i$ and closing time $l_i$ are conducted with each point $i \in O$. They are also used as the maximum duration that the vehicle must depart from the depot and return to the depot. There is travel time $t_{ij}$ that is associated with each link $(i, j) \in A$. The set $K = \{1, 2, ..., v\}$ is also used in this problem for describing vehicles. Each $k \in K$ has a vehicle capacity $Q$. The problem is formulated into integer linear programming (ILP) model and use several decision variables. The variables that are related to vehicle load and vehicle travel time is $L^k_i$ and $T^k_i$ as a non-negative continuous decision variable. $x^k_{ij}$ is a decision variable. 1 if the vehicle travels and 0 to vice versa. PDPLTV is represented in a three-
index integer linear programming. The formulations are based on [17] and [18]. The completed ILP model can be seen in [9].

2.2. Simulated Annealing Algorithm
The construction algorithm used in this study is the Insertion Heuristic (IH) algorithm. This algorithm forms a solution by inserting one by one the customer's location between the arcs formed and then the route with the minimum insertion cost is selected.

SA algorithm is an algorithm that belongs to a heuristic improvement where the resulting solution is near optimal. This algorithm uses random techniques to explore the solution space. The SA algorithm is found based on the analogy of the annealing process of a material. The material is heated at a certain temperature to its melting point, then cooled to the speed of cooling slowly to a certain temperature to find the best atomic structure. This cooling speed is commonly called a cooling schedule [19]. The implementation of the SA algorithm is always to accept a better solution and allow it to accept a worse solution with a certain probability. The probability function can be seen in equation 1.

\[ P = e^{-\Delta/T_c} \]  \hspace{1cm} (1)

With \( P \) is the probability of acceptance, \( \Delta \) is the difference between the evaluation of the new solution \( f(\sigma_B) \) and the evaluation of the current solution \( f(\sigma_c) \) and \( T_c \) are the current temperature. The solution produced by the SA algorithm is very dependent on the parameters used such as the initial temperature \( T_0 \), the final temperature \( T_f \) and the cooling schedule \( \alpha \). To find the new temperature \( T_f \) in the cooling process can use equation 2.

\[ T_b = \alpha \cdot T_c \]  \hspace{1cm} (2)

Step of proposed LS in this paper are given as follows:
0. Control parameters initialization \( T_0, T_f, \alpha, l, E \).
1. Generate initial solution \( \sigma_0 \) using insertion.
2. Set current temperature \( T_c \) equal to initial temperature \( T_0 \), \( e = 1 \), dan current solution \( \sigma_c \) equal to initial solution \( \sigma_0 \).
3. Using pair relocation operator, generate current solution
4. Collect the set of new solutions \( \sigma_B \).
5. Calculate \( f(\sigma_B) \).
6. Take \( f(\sigma_B) \) minimum objective.
7. Calculate difference \( f(\sigma_B) \) and \( f(\sigma_c) \) or \( \Delta \). If \( \Delta < 0 \), continue to 12, otherwise, 9.
8. Generate random number \( r \) between 0-1.
9. Calculate \( \exp(-\Delta/T_c) \). If \( r < \exp(-\Delta/T_c) \), continue to 12, otherwise, 11.
10. Set \( e = e + 1 \). Back to 4.
11. Set \( \sigma_c = \sigma_B \).
12. \( e = E ? \). If yes, continue to 14. otherwise, 11.
13. \( T_c \leq T_a ? \). If yes, continue to 17, otherwise L15.
14. \( T_b = \alpha \cdot T_c \).
15. \( T_c = T_b \), back to 4.
16. \( \sigma_a = \sigma_c \).
17. Stop.

3. Result
Intel® Core™ i5-3470 3.20 GHz (4CPUs) 4 GB RAM, under Windows 10 operating system are used to the computational experiment. The SA algorithm used is implemented in the programming language MATLAB™.
[6] generates the instance that the passenger and package request randomly generated the following uniform distributions. These instances are used in this study for the computational experiments. Each request has pickup and delivery nodes. Parameter $\alpha = 40$ and $\beta = 60$, $v = 2$, $n = 7$ are used in this experiment, same with [9].

Table 2 shows that SA algorithm is able to complete several instances. The solutions offered are similar to solutions in previous studies. But at some data, only shows optimal local solutions. This is because the SA algorithm is probabilistic. Each time (when running) will show different results. In addition, the selection of the appropriate SA parameters will also affect the quality of the solution produced. However, the SA algorithm is able to provide a solution with a shorter time. This is because the SA algorithm does not need to explore all the solutions from problem space.

**Table 1.** Computational results from [9]

| No | Instance     | Objective | Unserved Customer | Solution       | Time (s) |
|----|--------------|-----------|-------------------|----------------|----------|
| 1  | lc101_20_0   | 251,33    | 2                 | Optimal global | 121      |
| 2  | lc102_20_0   | 363,33    | 4                 | Optimal global | 249      |
| 3  | lc103_20_0   | 353,33    | 4                 | Optimal global | 18       |
| 4  | lc104_20_0   | 379,33    | 5                 | Optimal global | 1        |
| 5  | lc105_20_0   | 349,33    | 4                 | Optimal global | 5        |
| 6  | lc106_20_0   | 243,33    | 2                 | Optimal global | 182      |
| 7  | lc107_20_0   | 368,67    | 5                 | Optimal global | 6        |
| 8  | lc108_20_0   | 371,33    | 4                 | Optimal global | 815      |

**Table 2.** Computational results from SA

| No | Instance     | Objective | Unserved Customer | Solution       | Time (s) |
|----|--------------|-----------|-------------------|----------------|----------|
| 1  | lc101_20_0   | 265,33    | 2                 | Optimal local  | 4,2      |
| 2  | lc102_20_0   | 383,21    | 4                 | Optimal local  | 6,3      |
| 3  | lc103_20_0   | 359,33    | 4                 | Optimal local  | 1,2      |
| 4  | lc104_20_0   | 379,33    | 5                 | Optimal global | 0,3      |
| 5  | lc105_20_0   | 349,33    | 4                 | Optimal global | 0,8      |
| 6  | lc106_20_0   | 255,25    | 2                 | Optimal local  | 5,3      |
| 7  | lc107_20_0   | 388,63    | 5                 | Optimal local  | 2,2      |
| 8  | lc108_20_0   | 400,33    | 4                 | Optimal local  | 10,2     |

4. Conclusion
This study proposed SA algorithm for solving the PDPLTV. Minimize total travel time and number of unserved customers are the objective of this study. The contribution of this paper is considering the limited vehicle number. The results shown by the algorithm are a good solution, but are still local optimal. This is caused the probabilistic of algorithm. Future work can improve this algorithm, or can use other algorithms to improve the quality of the solution. To improve this algorithm, other operator can be chosen. The operator is very important, because this tool work to explore space of solution. Using other parameter value is another option. For example, number of iterations, bigger number can produce good quality solution, but increase running time. Future work, this algorithm can be implemented with large scale of data.

Acknowledgments
The authors would like to thank to the Universitas Syiah Kuala for laboratory facilities of Industrial Engineering and partial financial assistance under Laboratory Grant of Fiscal Year 2019 Number: 305/UN11.2/PP/PNBP/SP3/2019 Date May 3, 2019. Thanks, and high appreciation to Rector, Head of Integrated Laboratory and Head of LPPM Universitas Syiah Kuala.

References
[1] Toth P and Vigo D 2014 Vehicle Routing: Problems, Methods, and Applications (SIAM)
[2] Berbeglia G, Cordeau J-F, Gribkovskaia I and Laporte G 2007 Static pickup and delivery problems: a classification scheme and survey TOP 15 1–31
[3] Parragh S N, Doerner K F and Hartl R F A survey on pickup and delivery problems: Part I: Transportation between customers and depot Journal für Betriebswirtschaft 2008
[4] Parragh S, Doerner K and Hartl R 2008 A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations Journal für Betriebswirtschaft 58 81–117
[5] Carrabs F, Cerulli R and Cordeau J-F 2007 An Additive Branch-and-Bound Algorithm for the Pickup and Delivery Traveling Salesman Problem with LIFO or FIFO Loading INFOR: Information Systems and Operational Research 45 223–238
[6] Benavent E, Landete M, Mota E and Tirado G 2015 The Multiple Vehicle Pickup and Delivery Problem with LIFO Constraints European Journal of Operational Research 243 752–762
[7] Cherkesly M, Desaulniers G and Laporte G 2014 Branch-Price-and-Cut Algorithms for the Pickup and Delivery Problem with Time Windows and Last-in-First-Out Loading Transportation Science 49 752–766
[8] Furtado M G S, Munari P and Morabito R 2017 Pickup and Delivery Problem with Time Windows: A New Compact Two-Index Formulation Operations Research Letters 45 334–341
[9] Andriansyah, Prasanti N and Sentia P D 2018 Pickup and delivery problem with LIFO, time duration, and limited vehicle number MATEC Web Conf. 204 07003
[10] Carrabs F, Cordeau J-F and Laporte G 2007 Variable Neighborhood Search for The Pickup and Delivery Traveling Salesman Problem with LIFO Loading INFORMS J. on Computing 19 618–632
[11] Cordeau J-F, Iori M, Laporte G and González J J S A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with LIFO loading Networks 55 46–59
[12] Li Y, Lim A, Oon W-C, Qin H and Tu D 2011 The tree representation for the pickup and delivery traveling salesman problem with LIFO loading European Journal of Operational Research 212 482–496
[13] Cheang B, Gao X, Lim A, Qin H and Zhu W 2012 Multiple pickup and delivery traveling salesman problem with last-in-first-out loading and distance constraints European Journal of Operational Research 223 60–75
[14] Battarra M, Erdoğan G, Laporte G and Vigo D 2010 The Traveling Salesman Problem with Pickups, Deliveries, and Handling Costs Transportation Science 44 383–399
[15] Erdog’can G, Battarra M, Laporte G and Vigo D 2012 Metaheuristics for the traveling salesman problem with pickups, deliveries and handling costs Computers & Operations Research 39 1074–1086
[16] Cherkesly M, Desaulniers G and Laporte G 2015 A Population-based Metaheuristic for the Pickup and Delivery Problem with Time Windows and LIFO Loading Comput. Oper. Res. 62 23–35
[17] Furtado M G S, Munari P and Morabito R 2017 Pickup and delivery problem with time windows: A new compact two-index formulation Operations Research Letters 45 334–341
[18] Cherkesly M, Desaulniers G and Laporte G 2014 Branch-Price-and-Cut Algorithms for the Pickup and Delivery Problem with Time Windows and Last-in-First-Out Loading Branch-Price-and-Cut Algorithms for the Pickup and Delivery Problem with Time Windows and Last-in-First-Out Loading Transportation Science, Article in Advance 1–15
[19] Yang X S 2010 *Engineering Optimization An Introduction with Metaheuristic Application* (New Jersey: John Wiley & Sons, Inc)