Simulated Annealing Algorithm for Customer-centric Location Routing Problem

by

Eugene Sohn

Bachelor of Business Administration (BBA) in Supply Chain Management
Texas A&M University, 2014

SUBMITTED TO THE DEPARTMENT OF SUPPLY CHAIN MANAGEMENT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN SUPPLY CHAIN MANAGEMENT

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2018

©2018 Eugene Sohn. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature redacted

Signature of Author..................................................

Department of Supply Chain Management

Signature redacted

Certified by..........................................................

Mohammad Moshref-Javadi, PhD
Postdoctoral Associate, Center for Transportation and Logistics
Thesis Advisor

Signature redacted

Accepted by....................................................... 

Dr. Yossi Sheffi
Director, Center for Transportation and Logistics
Elisha Gray II Professor of Engineering Systems
Professor, Civil and Environmental Engineering
Simulated Annealing Algorithm for Customer-centric Location Routing Problem

by

Eugene Sohn

Submitted to the Department of Supply Chain Management on May 17, 2018
in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Supply Chain Management

ABSTRACT

In today’s world, the e-commerce market is growing rapidly and becoming more competitive. While many players in the industry are attempting to get their share of pie, consumers are demanding faster deliveries and free shipping. This market growth and change in consumer behavior provides an exciting opportunity for companies to compete. In order to meet the new consumer demand, companies need to find better ways to deliver faster. Faster delivery times can be achieved by using an optimization model to plan delivery network and operations. Typically, this optimization model has been based on minimizing cost. However, in the current market, lowest cost is not necessarily the best driver of sales as the consumer culture enters an era of instant gratification. We argue that minimizing customer waiting time will bring better performance and win over market share by providing the quickest delivery service that is expected by the majority of consumers. We propose solving the location routing problem (LRP) aiming at minimizing customer waiting time with capacitated depots and vehicles. We take two approaches to solve this problem: mathematical model and heuristic algorithm. The mathematical model obtains the optimal solution, but it has a limitation on the size of the problem due to the NP-hardness of the LRP. Therefore, we introduce three different variations of Simulated Annealing (SA) algorithm to solve the Capacitated Latency Location Routing Problem (CLLRP). According to the comparison results on a popular benchmark test, one of the designed SAs, the Iterative Simulated Annealing algorithm, consistently provides the best combination of performance and computation time compared to the other two SAs. Therefore, this specific algorithm is further compared to the mathematical model on some problem instances. The comparison results demonstrate that the proposed algorithm performs competitively with the algorithms in the literature and the mathematical model.

Thesis Advisor: Mohammad Moshref-Javadi, PhD
Title: Postdoctoral Associate, Center for Transportation and Logistics
ACKNOWLEDGEMENTS

Many individuals have helped me during my time at MIT and throughout the course of my research. I would like to extend my gratitude to the following individuals:

Dr. Mohammad Moshref-Javadi—For supporting me with your guidance and expertise and pushing me to deliver excellence.

My Classmate, Justin Yoon—For being my sounding board and making thesis more enjoyable.

My Classmate, Chun Him Yau—For your expertise in Python, dataframes, and numpy.

My Significant Other, Courtney Paige Stephen—For copyediting with a critical eye.

MIT SCM Faculties, Staffs, and Students—For making this brief moment we shared together worth remembering.
TABLE OF CONTENTS

1 INTRODUCTION ................................................................................................................. 7

2 LITERATURE REVIEW ........................................................................................................ 9

2.1 INCAPACITATED LOCATION-ROUTING PROBLEMS ....................................................... 9
2.2 CAPACITATED LOCATION-ROUTING PROBLEMS ...................................................... 10

3 METHODOLOGY ............................................................................................................... 14

3.1 PROBLEM DESCRIPTION AND MATHEMATICAL MODEL ............................................. 14
3.2 SIMULATED ANNEALING ................................................................................................. 17
  3.2.1 Initial Solution .......................................................................................................... 19
  3.2.2 Non-local Operators ............................................................................................... 21
  3.2.3 Local Operators ...................................................................................................... 22
  3.2.4 Termination Criteria ............................................................................................... 23
  3.2.5 Simulated Annealing Strategies ............................................................................... 24
  3.2.6 Parameter Settings .................................................................................................. 28

4 COMPUTATIONAL RESULTS ............................................................................................. 29

4.1 MATHEMATICAL MODEL RESULTS ............................................................................... 29
4.2 SIMULATED ANNEALING RESULTS ............................................................................. 31
  4.2.1 Comparison of SA Strategies .................................................................................. 31
  4.2.2 Comparison of SA3 with Mathematical Model ....................................................... 32

5 CONCLUSION .................................................................................................................... 34

6 REFERENCES ..................................................................................................................... 36
LIST OF FIGURES

Figure 1: Visualization of the Solution Representation .................................................. 20
Figure 2: Non-local Insertion Operator—Insert Customer 4 to Customer 7 .................. 21
Figure 3: Non-local Swap Operator—Swap Customer 4 with Customer 7 ..................... 22
Figure 4: Flip Operator—Flip Customer 4 to Customer 7 ........................................... 22
Figure 5: Local Insertion Operator—Insert Customer 4 to Customer 7 ....................... 23
Figure 6: Local Swap Operator—Swap Customer 4 with Customer 7 ......................... 23
Figure 7: Computing Time for Different Number of Depots and Vehicles ................. 30
Figure 8: Customer Waiting Time for Different Number of Depots and Vehicles .......... 30
Figure 9: Visual Representation of 10-4-4-5 Solution ..................................................... 33

LIST OF TABLES

Table 1: Different Heuristic Algorithms for Location Routing Problems ..................... 18
Table 2: Pseudocode for the Initial Solution ..................................................................... 20
Table 3: Adaptive Simulated Annealing (SA1) Pseudo Code ........................................ 24
Table 4: Sequential Simulated Annealing (SA2) Pseudo Code ..................................... 26
Table 5: Iterative Simulated Annealing (SA3) Pseudo Code ......................................... 27
Table 6: SA Algorithm Parameters .................................................................................. 28
Table 7: Three SAs compared with Moshref-Javadi and Lee (2016)'s result on Prins et al. (2006) Benchmark Problem Set ........................................................................... 32
Table 8: Computational Result Comparison Between Mathematical Model and SA3 ..... 33
Section 1 - Introduction

1 Introduction

In today’s competitive e-commerce market, delivering products in a timely manner is very important because leading e-commerce companies such as Amazon have set a two-day delivery as a standard. This two-day delivery window has become a new norm for customers. Such deliveries typically happen from depots to customers via a fleet of vehicles. As Moshref-Javadi and Lee (2016) stated, to design an efficient delivery system, the three key optimization variables are: depot locations, vehicle allocations (how many and what types of vehicles to use), and vehicle routings (the sequence of deliveries to customers for each vehicle). A well-designed distribution system will enhance the performance of companies with a competitive advantage by providing a high quality of service through faster delivery and thus increase their market share of ecommerce.

The goal of commercial companies in delivering products is to maximize profit. By minimizing costs, the companies will increase their profit and meet such goals. Another way for these companies to maximize their profit is by increasing their revenue. Based on the recent trends in consumer preference, Joerss, Neuhaus, and Schröder (2016) from McKinsey & Company states that nearly 25 percent of consumers are willing to pay significant premiums for getting same day or instant delivery. With this increase in consumer’s appetite for faster delivery, companies should focus on minimizing customers’ waiting time instead of minimizing cost of delivery. We believe that this will increase customer satisfaction and eventually lead to more profit for companies by increasing market share from a reputation of good delivery service. Due to these difference between the customer-centric system and server-centric system, the customer-centric systems should be carefully designed and optimized to achieve the minimum total waiting time of the system. In this direction of research, Moshref-Javadi and Lee (2016) have solved such a problem by introducing a solution for Latency Location-Routing Problem (LLRP) both mathematically and
heuristically. In their paper, they assume depots to have unlimited capacities while vehicles are capacitated.

The goal of this thesis is to model and solve the Latency Location-Routing Problem with capacitated depots with the objective of minimizing the total customer waiting time by optimally determining the location of depots, allocation of vehicles, and routing decisions concurrently. This problem is called the Capacitated Latency Location Routing Problem (CLLRP) and is a hybrid of two optimization problems: Facility Location Problem (FLP) and Cumulative Capacitated Vehicle Routing Problem (CCVRP). The FLP (Bramel & Simchi-Levi, 1997) determines the locations of depots, and the CCVRP (Ribeiro & Laporte, 2012) allocates vehicles to depots and finds the routes from depots to customers. The FLP and CCVRP are inter-related and therefore, should be solved simultaneously to achieve the optimized results. Due to the complexity of the CLLRP, the mathematical model (Mixed Integer Programming Model) can be used to solve only small-scale problems. Hence, we also propose a metaheuristic algorithm to solve large-scale problems efficiently. The model and algorithm are implemented and evaluated on several generated and adopted problems from the literature.
2 Literature Review

Many authors have studied different types of location routing problems, and we will separate them into two sub-sections: 1) incapacitated location routing problems and 2) capacitated location routing problems. Location routing problem (LRP), as previously mentioned, is finding optimal location of depots and routings of vehicles from the chosen depots to customers. Incapacitated LRP is a problem where we assume no capacity for either depots or vehicles. Capacitated LRP then is a problem where we assume capacity for depots and vehicles. This extra constraint in the capacitated LRP will always provide the same or a less optimal solution, but the results are more applicable for implementation because they more closely mimic the real-world constraints that companies face in their depot capacities.

2.1 Incapacitated Location-Routing Problems

Moshref-Javadi and Lee (2016) have solved Latency Location-Routing Problem (LLRP) mathematically and using two heuristic algorithms (Memetic Algorithm and Recursive Granular Algorithm). They use mixed integer linear programming (MILP) for less than nine customers and Memetic Algorithm or Recursive Granular Algorithm for larger instances. The model’s objective is to minimize the total customer waiting time, which is the sum of the arrival time of the vehicles at each customer. Moshref-Javadi and Lee achieve this objective by concurrently solving optimal location, allocation, and routing decisions. Furthermore, Wang, Du, and Ma (2014) have solved multi-objective location-routing for relief distribution problem with split delivery by utilizing two heuristic algorithms: non-dominating sorting genetic algorithm-II (NSGA-II) and non-dominated sorting differential algorithm (NSDE). The model’s objectives are to 1. minimize the maximum vehicle route traveling time (lowering the longest vehicle route traveling time), 2. minimize relief distribution cost, and 3. maximize the minimum route reliability. Both of the papers mentioned assume that depots have unlimited capacities.
Simulated Annealing Algorithm for Customer-centric Location Routing Problem

Derbel et al. (2012) have solved LRP with capacitated depots and an incapacitated vehicle for each depot with Genetic Algorithm (GA) and Iterated Local Search (ILS). The model's objective is to minimize the total cost of opening depots and delivering to customers by optimizing depot locations and vehicle routings to customers by using the GA and ILS algorithms. Derbel et al. (2012) assume that each depot has a capacity and cost while one incapacitated vehicle is assigned to each opened depot. Derbel et al. (2012) use ILS to refine the GA search through successive iterations and improve the gap to an optimal solution through using various search spaces. Jarboui, Derbel, Hanafi, and Mladenovic (2012) have also solved LRP with multiple capacitated depots and one incapacitated vehicle per depot using Various Neighborhood Search Heuristic (VNS). They use five neighborhood structures which are either routing or location type and use them in both perturbation and local search steps. The model's objective is to minimize the total cost for location and routing. They assume vehicles with unlimited capacity, which limits the application of the model in real life.

Unlike Moshref-Javadi and Lee (2016) and Wang, Du, and Ma (2014), Derbel et al. (2012) and Jarboui, Derbel, Hanafi, and Mladenovic (2012) have capacitated depots but have incapacitated vehicles. This paper will address the limitation of incapacitated depots and vehicles by setting up constraints for both depot and vehicle capacities.

2.2 Capacitated Location-Routing Problems

Escobar, Linfati, and Toth (2012) have solved the Capacitated Location-Routing Problem (CLRP) with a two-phase hybrid heuristic algorithm. The first phase is the Construction Phase, where it builds an initial feasible solution using an initial hybrid procedure followed by a splitting procedure to minimize the routing cost. The next phase is the Improvement Phase where a modified Granular Tabu Search (GTS) procedure is applied to improve the quality of the current solution with randomized perturbation procedure. The model's
objective is to minimize the sum of the opening costs for depots, the fixed cost of vehicles, and the variable traveling costs related to the routes. Escobar, Linfati, and Toth (2012) assume that each vehicle has a same fixed cost and capacity. Yu, Lin, Lee, and Ting (2010) have also solved the CLRP but with a simulated annealing (SA) heuristic method. To solve it, they generate an initial solution with a greedy algorithm, put the generated solution into the solution representation used in the SALRP algorithm, and use a standard SA procedure with a random neighborhood structure that includes an insertion move, swap move, and 2-opt move. The model’s objective is to minimize total cost of the opening cost of depots and routing costs (both variable and fixed cost) by determining which depots to open and which routes to use. Ting and Chen (2013) have solved the capacitated location-routing problem (CLRP) using a multiple ant colony optimization algorithm (MACO). Three colonies are used to solve this problem: location selection, customers assignment, and route construction. Ting and Chen’s objective function is to minimize total systems cost (fixed facility costs, transportation costs, and vehicle costs) by finding an optimal number and locations of depots and vehicle routes for each depot. Lopes et al. (2016) have solved the capacitated location routing problem using hybrid genetic algorithm. They use a metaheuristic that follows the standard Genetic Algorithm (GA) framework hybridized with local search procedures. The objective function is to minimize total costs by determining which depots to open and which routes to take using the hybrid genetic algorithm. Some of the main advantages of the proposed approach are intuitive chromosome representation and a simple framework.

It can be seen that there are several approaches to solve the capacitated location routing problem. Since the performance of the heuristic approaches highly depend on the design, parameters, and problem sizes, the goal is to design an efficient algorithm which can obtain near-optimal solution in reasonable time.
In addition to the general CLRP, some authors have focused on specific cases. For example, Rabbani et al. (2018) have solved hazardous waste location routing problem (HWLRP) with consideration of incompatibility among some kinds of wastes and minimizing total cost, transportation risk, and site risk by implementing two multi-objective evolutionary algorithms. To solve this problem, Nondominated Sorting Genetic Algorithm II (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO) algorithm have been employed. One of the main differences in assumptions in this paper is that vehicles are heterogenous as some wastes cannot mix together and need different types of vehicles to transport. Rath and Gutjahr (2014) have solved the Warehouse Location-Routing Problem using a “math-heuristic” technique for disaster relief. The model’s three objectives are to minimize fixed costs for depots and vehicles, minimize the transportation costs from plants to depots and warehousing costs, and maximize covered demand concurrently. In this model, Ruth and Gutjahr treat this problem as a single commodity problem as the composition of goods is homogenous. They also assume that both plants and possible depots exist with given capacities. Ruth and Gutjahr have proposed two solutions: 1) exact method for small instances by applying Adaptive Epsilon-Constraint Algorithm and 2) constraint pool heuristic for larger instances by using MILP as a backbone and Variable Neighborhood Search algorithm to optimize. Specific cases for location-routing problems will generally require different objective function or constraints.

Lastly, Farham, Sural, and Iyigun (2018) have provided an exact solution approach for location routing problem with time windows (LRPTW) based on branch-and-price for solving larger problem sizes and reducing solution times. The authors take a branch-and-price algorithm based on set-partitioning approach to solve and introduce two-stage heuristic for a significant saving in solving time while maintaining the solution quality. The authors’ objective function is to minimize total depot opening cost, vehicle fixed cost, and
traveling cost within allowable times for customers. In this problem, the authors assume homogeneity of vehicles. The key difference in this problem is that it minimizes cost while making sure the delivery happens within a specific time frame.

According to the literature review, there are papers that cover the latency location-routing problems (Moshref-Javadi and Lee), whereas some papers cover the conventional capacitated location-routing problems. However, there are no studies done on capacitated latency location-routing problems focusing on minimizing the waiting time of customers while adding capacity restrictions for both depots and vehicles. In this paper, we propose the capacitated latency location-routing problem (CLLRP), formulate the problem, introduce a heuristic algorithm, and evaluate the results.
3 Methodology

This section will explain two different methods used to solve the Capacitated Latency Location Routing Problem (CLLRP): mathematical model and simulated annealing algorithm. The advantages and disadvantages of each approach will be addressed.

3.1 Problem Description and Mathematical Model

Because the Location Routing Problem is a combination of two NP-hard problems (Facility Location Problem and Cumulative Capacitated Vehicle Routing Problem), it cannot find the optimal solution for real-world size problems in a reasonable time. However, it is important to construct the model and compare with the performance of the heuristic algorithm on small-size problems to guarantee the promising performance of the algorithm.

The assumptions of the model are the following:

- The number and locations of candidate depot ($N_f$) are known
- The number of depots to open ($N_g$) and vehicles ($N_v$) to use are predetermined
- The capacities of depots ($W_g$) and vehicles ($Q_k$) are pre-determined
- All of the demands are satisfied
- The travel time between customer $i$ and $j$ are symmetric

The following notations are used to formulate the problem:

**Indices**

$i, j, u$ Represent customers, totally $N_c$ customers  
$k$ Represents vehicle  
$g$ Represents candidate depots, totally $N_f$

**Sets**

$K$ Set of vehicles, $|K|  
$G$ Set of candidate depots, $|G| = N_f  
$V'$ Set of customers, $|V'| = N_c  
$V$ Set of all customers and candidate depots $|V| = N = N_c + N_f$

**Parameters**

14
Section 3 - Methodology

Variables

\( t_i^k \quad \text{Arrival time of vehicle } k \text{ at customer } i \)

\( x_{ij}^k \quad \text{1 if vehicle } k \text{ traverses arc } (i,j) \text{ from customer } i \text{ to customer } j; \) otherwise, 0

\( f_{gi} \quad \text{1 if customer } i \text{ is supplied from depot } g; \text{ otherwise, 0} \)

\( z_g \quad \text{1, if facility } g \text{ is open; otherwise 0} \)

The mathematical formulation of the CLLRP is the following:

Minimize:

\[
\sum_{k \in K, i \in V} t_i^k
\]  \hspace{1cm} (1)

s.t.

\[
\sum_{i \in V} f_{gi} q_i \leq W_g \quad \forall g \in G
\]  \hspace{1cm} (2)

\[
\sum_{j \in V} x_{ij}^k = \sum_{j \in V} x_{ji}^k \quad \forall i \in V, \forall k \in K
\]  \hspace{1cm} (3)

\[
\sum_{k \in K, j \in V', i \neq j} x_{ij}^k = 1 \quad \forall i \in V'
\]  \hspace{1cm} (4)

\[
\sum_{g \in G} f_{gj} = 1 \quad \forall j \in V'
\]  \hspace{1cm} (5)

\[
\sum_{i \in V, j \in V'} x_{ij}^k q_j \leq Q_k \quad \forall k \in K
\]  \hspace{1cm} (6)
Simulated Annealing Algorithm for Customer-centric Location Routing Problem

\[ \sum_{g \in G, i \in V'} x_{gi}^k = 1 \forall k \in K \]  \hspace{1cm} (7)

\[ \sum_{u \in V'} x_{ui}^k + \sum_{u \in V \setminus \{i\}} x_{ui}^k \leq 1 + f_{gi} \forall i \in V', \forall k \in K, \forall g \in G \]  \hspace{1cm} (8)

\[ t_i^k + c_{ij} - (1 - x_{ij}) M \leq t_j^k, \forall i \in V, \forall j \in V', \forall i \neq j, \forall k \in K, \forall g \in G \]  \hspace{1cm} (9)

\[ \sum_{i \in V} f_{gi} \leq M z_g \forall g \in G \]  \hspace{1cm} (10)

\[ \sum_{i \in V} f_{gi} \geq z_g \forall g \in G \]  \hspace{1cm} (11)

\[ \sum_{g \in G} z_g = N_g \]  \hspace{1cm} (12)

\[ t_i^k \geq 0, \forall i \in V, \forall k \in K \]

\[ z_g \in \{0,1\} \forall g \in G \]  \hspace{1cm} (13)

\[ x_{ij}^k \in \{0,1\} \forall i \in j \]

In this model, the objective function (1) is to minimize the total customer waiting time, which is the sum of the arrival time of the vehicles at customers. Constraint (2) ensures that the sum of customers’ demand for each of the depots are below the depot’s capacity. Constraints (3 and 4) enforce flow continuity in routes. Constraint (5) ensures that each customer is assigned to one depot only. Constraint (6) guarantees that total loads for each vehicle does not exceed vehicle’s capacity. Constraint (7) ensures that each customer is assigned to exactly one route, and each route starts from exactly one depot. Constraint (8) makes sure customer is assigned to a depot if the route is connecting the depot to the customer. Latency at each node is calculated using constraints (9). Constraint (10) guarantees that a facility must be open if a customer is assigned to that facility. Constraint
Section 3 - Methodology

(11) makes sure all opened depots are used. Constraint (12) ensures that agreed number of depots are opened. Finally, constraint (13) addresses the ranges and types of the variables. Because the Location Routing Problem is a combination of two NP-hard problems (Facility Location Problem and Cumulative Capacitated Vehicle Routing Problem), it is also an NP-hard problem and is only applicable for small-scale problem. Therefore, in the next section we introduce the metaheuristic approach to solve the CLLRP on small and large-scale problems.

3.2 Simulated Annealing

We introduce simulated annealing heuristic algorithm to solve large-sized problems. Simulated Annealing (SA) is a heuristic algorithm introduced by Kirkpatrick, Gelatt, and Vecchi (1983) and is inspired by the annealing process. Annealing is a process in metallurgy where metals are cooled slowly so that the atoms randomly distribute over a longer period of time to increase size of crystals and reduce defects. The atoms move around more quickly when the temperature is high, and they slow down as the temperature cools off. Similar to the atoms in annealing process, SA accepts a solution more easily when the temperature is high and is stricter when the temperature is low. Therefore, when the temperature is high, SA is able to escape local optimum in order to seek global optimum as the “temperature” of the algorithm decreases.

The SA algorithm begins by creating an initial solution. This solution can be developed randomly or by using a simple algorithm such as nearest neighborhood. SA algorithm then performs operators to the initial solution to develop a new solution and compare with the current best solution to see if an improvement has been made. If the new solution has lower latency than the current best solution, the best solution will be updated with the new solution. If the new solution is not better, then the new solution can still be accepted with
a probability determined by Boltzmann function $e^{-\Delta/(kT)}$, where $\Delta$ is the latency of new solution – current solution, $k$ is the predetermined constant, and $T$ is the current temperature. The main reason for accepting a worse solution is avoiding a local optimum and moving on to a solution that will lead to the global optimum.

Besides the simulated annealing algorithm, there are other heuristic algorithms that are popular to solve combinatorial optimization problem such as location routing problem. Yu et.al (2010) solved Capacitated Location Routing Problem (CLRP) with Simulated Annealing. Moshref-Javadi and Lee (2016) solved Latency Location Routing Problem (LLRP) with two different heuristic algorithm such as Memetic Algorithm (MA) and Recursive Granular Algorithm (RGA). Ardjmand et al. (2015) solved LRP for hazardous materials using Genetic Algorithm (GA). Escobar et al (2013) solved CLRP using Modified Granular Tabu Search. Lopes et al. (2016) solved CLRP using Evolutionary Algorithm. Lastly, Ting and Chen (2013) solved CLRP using Multiple Ant Colony Algorithm (MACA). Table 1 shows a summary of different heuristic algorithms that people have used previously.

<table>
<thead>
<tr>
<th>Year</th>
<th>Author</th>
<th>Heuristic Algorithm</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>Yu et al.</td>
<td>Simulated Annealing</td>
<td>Capacitated Location Routing Problem</td>
</tr>
<tr>
<td>2013</td>
<td>Escobar et al.</td>
<td>Modified Granular Tabu Search</td>
<td>Capacitated Location Routing Problem</td>
</tr>
<tr>
<td>2016</td>
<td>Lopes et al.</td>
<td>Evolutionary Algorithm</td>
<td>Capacitated Location Routing Problem</td>
</tr>
<tr>
<td>2013</td>
<td>Ting and Chen</td>
<td>Multiple Ant Colony Algorithm</td>
<td>Capacitated Location Routing Problem</td>
</tr>
<tr>
<td>2016</td>
<td>Moshref-Javadi</td>
<td>Memetic Algorithm or Recursive</td>
<td>Incapacitated Latency Location Routing</td>
</tr>
<tr>
<td></td>
<td>and Lee</td>
<td>Granular Algorithm</td>
<td>Problem</td>
</tr>
<tr>
<td>2013</td>
<td>Jarboui et al.</td>
<td>Various Neighborhood Search</td>
<td>Incapacitated Location Routing Problem</td>
</tr>
<tr>
<td>2015</td>
<td>Ardjmand et al.</td>
<td>Genetic Algorithm</td>
<td>LRP for Hazardous Materials</td>
</tr>
</tbody>
</table>

Although the concept of simulated annealing (SA) algorithm is simple, it can be applied in various ways to solve different type of optimization. In this paper, we propose three different types of SA: Adaptive Simulated Annealing (SA1), Sequential Simulated Annealing (SA2), and Iterative Simulated Annealing (SA3). Each of these strategies will be explained in details in Section 3.2.5. However, they use the same heuristic to generate an initial solution.
3.2.1 Initial Solution
When selecting depots, we use probabilistic centrality method. There is a list of candidate depots, $G$, a number of depots to open, $N_g$, list of customers, $V'$, and location of customers and depots. For each customer, the closest depot to that customer will gain a popularity point. After finding the closest depot for all customers, the depots will have different amounts of popularity points with the sum being equal to the number of customers. With these popularity points, each candidate depot will have a unique probability of getting selected (popularity points of that candidate depot/ sum of all popularity points). This method is very important because it allows depots with low popularity a chance to be selected. This will result in a more diverse initial solution, which will lead to a higher probability of finding the optimal solution. Once a depot has been selected, that depot will no longer have popularity points. Thus, the sum of all popularity points will also decrease by that of depot. The remaining depots will have adjusted a probability and a new depot will be selected until $N_g$ depots have been selected. If there is not $N_g$ number of candidate depots with popularity points, then a random candidate depot will receive a popularity point until there is an $N_g$ number of candidate depots with points.

Once $N_g$ number of depots have been selected, then it will go through a similar process to decide how many vehicles to allocate to each depot. First, each selected depot will be allocated with one vehicle. If there are more vehicles left, each selected depot will have a probability of getting assigned to a vehicle based on the point system described above. However, in this case, chosen depots will remain until all vehicles have been assigned. Once all vehicles have been assigned to a depot, then each vehicle will take turn to assign the closest customer from its current location. This will be done until all customers have been allocated.
Simulated Annealing Algorithm for Customer-centric Location Routing Problem

Table 2: Pseudocode for the Initial Solution

<table>
<thead>
<tr>
<th>Initial Solution—Nearest Neighborhood with Centrality Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Input $N_g, N_f, N_v, N_c, C_{ij}$</td>
</tr>
<tr>
<td>2: Count closest customers $N_c$ to each depot $N_f$</td>
</tr>
<tr>
<td>3: Each depot has weighted probability of being selected based on the number of closest customers to the total customers.</td>
</tr>
<tr>
<td>4: Open $N_g$ depots based on the probability of each depot</td>
</tr>
<tr>
<td>5: Assign vehicles $N_v$ to the opened depots ($R_g$)</td>
</tr>
<tr>
<td>6: While customers assigned to vehicle $&lt; N_c$</td>
</tr>
<tr>
<td>7: For $j = 0$ to $N_v$</td>
</tr>
<tr>
<td>8: New_node ← nearest node to Current_node</td>
</tr>
<tr>
<td>9: customer assigned +=1</td>
</tr>
<tr>
<td>10: End For</td>
</tr>
<tr>
<td>11: End While</td>
</tr>
<tr>
<td>12: Calculate the total latency</td>
</tr>
</tbody>
</table>

Figure 1 shows a solution representation after the initial solution has been formulated. Each row is a vehicle and its route that begins from a depot and delivers to customers. For example, in the first row in Table 1 Figure 1, Vehicle 1 leaves Depot 1 to deliver to Customer 1, Customer 2, Customer 3, and Customer 4. This matrix will be used in Simulated Annealing algorithm to find more optimal solutions.

<table>
<thead>
<tr>
<th>Vehicle 1</th>
<th>Depot 1</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle 2</td>
<td>Depot 2</td>
<td>C5</td>
<td>C6</td>
<td>C6</td>
<td>C8</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 1: Visualization of the Solution Representation
Proposed Simulated Annealing algorithms will contain two non-local operators and three local operators. These operators are essential in finding an optimal solution.

3.2.2 Non-local Operators
Non-local operators allow shuffling customers in different routes. This shuffling allows the algorithm to leave local optimum and find global optimum. Two non-local operators are non-local insertion and non-local swap. To continue with the table above, we will apply non-local insertion and non-local swap to the initial solution. Figure 2 demonstrates using non-local insertion operator from Customer 4 to Customer 7. The picture on the left shows the original solution, and the picture on the right shows the new solution after Customer 4 has been inserted before Customer 7.

Figure 2: Non-local Insertion Operator – Insert Customer 4 to Customer 7

Figure 3 depicts non-local swap where Customer 4 from Vehicle 1 is swapped with Customer 7 from Vehicle 2.
3.2.3 Local Operators

Local operators help optimize locally for the given solution. It allows improvements by shuffling customers within the routes. Three local operators are flip, local insertion, and local swap. Figure 4 shows using flip operator to Customer 4 to Customer 7. The picture on the left and the first row represent the original solution, and the picture on the right and the last row represent the new solution after flip operator has been applied to Customer 4 and Customer 7.

Figure 5 shows local insertion operator where Customer 4 has been inserted before Customer 7.
Section 3 - Methodology

Lastly, Figure 6 shows local swap operator where Customer 4 and Customer 7 have been swapped.

These defined operators will be used with different strategies in the three proposed Simulated Annealing algorithms and compare the performance of each algorithm.

3.2.4 Termination Criteria
All three proposed algorithms will terminate once the current temperature, $T$, reaches final temperature, $T_f$. After each full iteration, current temperature will decrease by a factor of $\alpha$, a number ranging from 0 to 1.
3.2.5 Simulated Annealing Strategies

As mentioned briefly, the three different simulated annealing algorithms are Adaptive Simulated Annealing (SA1), Sequential Simulated Annealing (SA2), and Iterative Simulated Annealing (SA3). The strategy of finding optimal solution for these three algorithms differ by their unique ways of using the defined operators. Adaptive Simulated Annealing uses intelligent algorithm to prioritize more successful operators. Sequential Simulated Annealing simply runs five operators in a sequential order. Iterative Simulate Annealing runs local operators within non-local operators.

Adaptive Simulated Annealing (SA1)

SA1 has three prime operators: flip, non-local insertion, and non-local swap. Each operator begins with an equal weight of 1/3. However, after each 50 iterations, the algorithm will assess which prime operators performed well and assign more weight to such operators. This intelligent algorithm function helps the model run more efficiently by prioritizing effective operators over ineffective operators. After each prime operator, if the new solution is accepted, then the model will perform three local operators (flip, local swap, and local insertion) to the new solution N-local times. Once the prime operators have run, the algorithm will run depot swap operator with a probability of (1/number of Customers). Lastly, before lowering the temperature, the model will repeat N-prime_operator times.

Table 3 is a detailed pseudo code of SA1.

Table 3: Adaptive Simulated Annealing (SA1) Pseudo Code

<table>
<thead>
<tr>
<th>Adaptive Simulated Annealing (SA1) Pseudo Code:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> T₀, Tᵣ, α, β, over_capacity, K, N-prime_operator, N-local, weight_prime_operator_1, weight_prime_operator_2, weight_prime_operator_3, reward_prime_1, reward_prime_2, reward_prime_3, count_prime_1, count_prime_2, count_prime_3, and feasibility</td>
</tr>
<tr>
<td><strong>Step 1:</strong> Generate an initial solution, X, by performing Nearest Neighborhood with Centrality.</td>
</tr>
<tr>
<td><strong>Step 2:</strong> Let weight_prime_operator_1, 2, and 3 = 1.0/3.0; T=T₀;</td>
</tr>
<tr>
<td><strong>Step 3:</strong> Let reward_prime_1, 2, and 3 and count_prime_1, 2, and 3 = 0.</td>
</tr>
<tr>
<td><strong>Step 4:</strong> Generate r₁ = random number between 0 and 1.</td>
</tr>
<tr>
<td>If r₁&lt;weight_prime_operator_1, generate a new solution Y from X by random flip operation and count_prime_1+=1.</td>
</tr>
</tbody>
</table>
Section 3 - Methodology

Else If \( r1 < \text{weight}_{\text{prime\_operator}}(1+2) \), generate a new solution \( Y \) from \( X \) by random non-local insertion operation and \( \text{count}_{\text{prime\_2}} += 1 \).
Else, generate a new solution \( Y \) from \( X \) by random non-local swap operation and \( \text{count}_{\text{prime\_3}} += 1 \).

**Step 5:** If the feasibility of \( Y == \text{true} \), measure the latency of \( Y \).

**Step 5.1:** If the latency of \( Y \leq \text{the latency of } X \), then \( Y \) is the new best solution and new current solution and \( \text{reward}_{\text{prime}}(1, 2, \text{or } 3) += 1 \).
Else, generate \( r2 = \text{random } (0,1) \)
If \( r2 < \exp(-A/KT) \), \( X = Y \) (\( Y \) becomes the new current solution)

**Step 6:** If feasibility of \( Y != \text{true} \), over_capacity = the amount of demand that is over the capacity, generate \( r2 = \text{random } (0,1) \)
If \( r2 < \exp(-(\Delta + \beta \cdot \text{over\_capacity})/KT) \), \( X = Y \) (\( Y \) becomes the new current solution)

**Step 7:** If \( Y \) became the new solution, then run three local operators on affected vehicles (flip, local swap, and local insertion). Repeat Step 5 and Step 6.

**Step 8:** repeat Step 7 \( N\)-local times.

**Step 9:** repeat Step 4 through Step 8 \( N\)-prime\_operator times.

**Step 10:** generate \( r2 = \text{random } (0,1) \)
If \( r2 < 1.0/\text{number of customers} \), then perform a random depot swap

**Step 10.1:** If feasibility of \( Y == \text{true} \), measure the latency of \( Y \).

**Step 10.2:** If latency of \( Y \leq \text{latency of } X \), then \( Y \) is the new best solution and new current solution (\( X = Y \)).
Else, generate \( r2 = \text{random } (0,1) \)
If \( r2 < \exp(-\Delta/\text{KT}) \), \( X = Y \) (\( Y \) becomes the new current solution)

**Step 10.3:** If feasibility of \( Y != \text{true} \), over_capacity = the amount of demand that is over the capacity, generate \( r2 = \text{random } (0,1) \)
If \( r2 < \exp(-(\Delta + \beta \cdot \text{over\_capacity})/KT) \), \( X = Y \) (\( Y \) becomes the new current solution)

**Step 11:** If solution \( Y \) is accepted, repeat Step 7.

**Step 12:** \( T = \alpha \cdot T \);

**Step 13:** Every 50 iterations from step 4, measure the effectiveness of each prime operator and modify the weight.
\[
\text{Weight\_prime\_operator\_1} = \text{weight\_prime\_operator\_1} + \text{reward\_prime\_1}/\text{count\_prime\_1}
\]
\[
\text{Weight\_prime\_operator\_2} = \text{weight\_prime\_operator\_2} + \text{reward\_prime\_2}/\text{count\_prime\_2}
\]
\[
\text{Weight\_prime\_operator\_3} = \text{weight\_prime\_operator\_3} + \text{reward\_prime\_3}/\text{count\_prime\_3}
\]

**Step 14:** If \( T < T_f \), then terminate the heuristic. Else, go back to **Step 3**

*Sequential Simulated Annealing (SA2)*

SA2 is a simpler algorithm than SA1. For each temperature, it runs two non-local operators and three local operators \( N\)-operator times. It then runs depot swap operator with a probability of \( (1/\text{number of Customers}) \) before reducing the temperature. Table 4 demonstrates a detailed pseudo code.
Simulated Annealing Algorithm for Customer-centric Location Routing Problem

Table 4: Sequential Simulated Annealing (SA2) Pseudo Code

Sequential Simulated Annealing (SA2):

**Input:** $T_0$, $T_f$, $\alpha$, $\beta$, over_capacity, $K$, N-operator, and feasibility

**Step 1:** Generate an initial solution, $X$, by performing Nearest Neighborhood with Centrality.

**Step 2:** Let $T=T_0$

**Step 3:** generate a new solution $Y$ from $X$ by random non-local swap operation

**Step 3.1:** If feasibility == true, measure the latency of $Y$.

**Step 3.2:** If latency of $Y \leq$ latency of $X$, then $Y$ is the new best solution and new current solution.

**Step 3.3:** If feasibility !=true, over_capacity = the amount of demand that is over the capacity, generate $r_1 = \text{random (0,1)}$

**Step 3.4:** Repeat Step 3 N-operator times

**Step 4:** repeat Step 3 with non-local insertion operation.

**Step 5:** generate a new solution $Y$ from $X$ by random flip operation

**Step 5.1:** If feasibility == true, measure the latency of $Y$.

**Step 5.2:** If latency of $Y \leq$ latency of $X$, then $Y$ is the new best solution and new current solution.

**Step 5.3:** If feasibility !=true, over_capacity = the amount of demand that is over the capacity, generate $r_1 = \text{random (0,1)}$

**Step 5.4:** If $r_1 < \exp(-\Delta/KT)$, $X = Y$ (Y becomes the new current solution)

**Step 5.5:** Repeat Step 5 through 7 N-operator times

**Step 6:** Repeat Step 5 with random local insertion operation.

**Step 7:** Repeat Step 5 with random local swap operation.

**Step 8:** Repeat Step 5 with random local swap operation.

**Step 9:** repeat Step 5 through 7 N-operator times

**Step 9.1:** If feasibility == true, measure the latency of $Y$.

**Step 9.2:** If latency of $Y \leq$ latency of $X$, then $Y$ is the new best solution and new current solution.

**Step 9.3:** If feasibility !=true, over_capacity = the amount of demand that is over the capacity, generate $r_1 = \text{random (0,1)}$

**Step 9.4:** If $r_1 < \exp(-(\Delta + \beta \times \text{over_capacity})/KT)$, $X = Y$ (Y becomes the new current solution)

**Step 10:** $T = \alpha \times T$

**Step 11:** If $T<T_f$, then terminate the heuristic. Else, go back to Step 3

Iterative Simulated Annealing

SA3 is a similar algorithm to SA1. For each temperature, it runs two non-local operators with three local operators within each non-local operator N-local_operator times. It then repeats the non-local operator N-non_local_operator times. Before the temperature
Section 3 - Methodology

decreases, the model runs depot swap operator with a probability of \((1/\text{number of Customers})\) with three local operators if the solution from depot swap is accepted. Table 5 is the model’s detailed pseudo code.

Table 5: Iterative Simulated Annealing (SA3) Pseudo Code

<table>
<thead>
<tr>
<th>Iterative Simulated Annealing (SA3):</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> (T_0, T_f, \alpha, \beta, \text{over_capacity}, K, \text{N_non_local_operator}, \text{N_local}, \text{and feasibility})</td>
</tr>
<tr>
<td><strong>Step 1:</strong> Generate an initial solution, (X), by performing Nearest Neighborhood with Centrality.</td>
</tr>
<tr>
<td><strong>Step 2:</strong> Let (T = T_0)</td>
</tr>
<tr>
<td><strong>Step 3:</strong> generate a new solution (Y) from (X) by random non-local swap operation.</td>
</tr>
<tr>
<td><strong>Step 3.1:</strong> If feasibility of (Y == \text{true}), measure the latency of (Y).</td>
</tr>
<tr>
<td><strong>Step 3.2:</strong> If latency of (Y \leq \text{latency of } X), then (Y) is the new best solution and new current solution ((X = Y)).</td>
</tr>
<tr>
<td>Else, generate (r_1 = \text{random (0,1)})</td>
</tr>
<tr>
<td>If (r_1 &lt; \exp(-\Delta/KT)), (X = Y) ((Y) becomes the new current solution)</td>
</tr>
<tr>
<td><strong>Step 3.3:</strong> If feasibility of (Y != \text{true}), (\text{over_capacity} = \text{the amount of demand that is over the capacity}), generate (r_1 = \text{random (0,1)})</td>
</tr>
<tr>
<td>If (r_1 &lt; \exp(-(\Delta + \beta \cdot \text{over_capacity})/KT)), (X = Y) ((Y) becomes the new current solution)</td>
</tr>
<tr>
<td><strong>Step 4:</strong> If the new solution from <strong>Step 3</strong> has been accepted, generate a new solution (Y) from (X) by random flip operation on affected vehicles.</td>
</tr>
<tr>
<td><strong>Step 4.1:</strong> If feasibility of (Y == \text{true}), measure the latency of (Y).</td>
</tr>
<tr>
<td><strong>Step 4.2:</strong> If latency of (Y \leq \text{latency of } X), then (Y) is the new best solution and new current solution ((X = Y)).</td>
</tr>
<tr>
<td>Else, generate (r_1 = \text{random (0,1)})</td>
</tr>
<tr>
<td>If (r_1 &lt; \exp(-\Delta/KT)), (X = Y) ((Y) becomes the new current solution)</td>
</tr>
<tr>
<td><strong>Step 4.3:</strong> If feasibility of (Y != \text{true}), (\text{over_capacity} = \text{the amount of demand that is over the capacity}), generate (r_1 = \text{random (0,1)})</td>
</tr>
<tr>
<td>If (r_1 &lt; \exp(-(\Delta + \beta \cdot \text{over_capacity})/KT)), (X = Y) ((Y) becomes the new current solution)</td>
</tr>
<tr>
<td><strong>Step 5:</strong> Repeat <strong>Step 4</strong> with random local insertion operation.</td>
</tr>
<tr>
<td><strong>Step 6:</strong> Repeat <strong>Step 4</strong> with random local swap operation.</td>
</tr>
<tr>
<td><strong>Step 7:</strong> Repeat <strong>Step 4</strong> through <strong>Step 6</strong> N-local times</td>
</tr>
<tr>
<td><strong>Step 8:</strong> Repeat <strong>Step 3</strong> with non-local insertion operation through <strong>Step 7</strong>.</td>
</tr>
<tr>
<td><strong>Step 9:</strong> Repeat <strong>Step 3</strong> through <strong>Step 8</strong> N-prime-operator times.</td>
</tr>
<tr>
<td><strong>Step 10:</strong> generate (r_2 = \text{random (0,1)})</td>
</tr>
<tr>
<td>If (r_2 &lt; 1.0/\text{number of customers}), then perform a random depot swap</td>
</tr>
<tr>
<td><strong>Step 10.1:</strong> If feasibility of (Y == \text{true}), measure the latency of (Y).</td>
</tr>
<tr>
<td><strong>Step 10.2:</strong> If latency of (Y \leq \text{latency of } X), then (Y) is the new best solution and new current solution ((X = Y)).</td>
</tr>
<tr>
<td>Else, generate (r_1 = \text{random (0,1)})</td>
</tr>
<tr>
<td>If (r_1 &lt; \exp(-\Delta/KT)), (X = Y) ((Y) becomes the new current solution)</td>
</tr>
<tr>
<td><strong>Step 10.3:</strong> If feasibility of (Y != \text{true}), (\text{over_capacity} = \text{the amount of demand that is over the capacity}), generate (r_1 = \text{random (0,1)})</td>
</tr>
</tbody>
</table>
Simulated Annealing Algorithm for Customer-centric Location Routing Problem

If r1 < \exp(-(\Delta + \beta * over\_capacity) / KT), X = Y (Y becomes the new current solution)

**Step 10.4:** If feasibility of Y == true, perform Step 4 through Step 7

**Step 11:** T = \alpha*T;

**Step 12:** If T<Tf, then terminate the heuristic. Else, go back to Step 3

### 3.2.6 Parameter Settings

All three simulated annealing algorithms will use six parameters: T0, Tf, \alpha, \beta, over\_capacity, and K. T0 represents the initial temperature, and Tf represents final temperature. The algorithm will begin at T0 and after each iteration, the temperature will be reduced by T = T* \alpha until T reaches Tf. \alpha represents the rate of cooling. \beta measures the weight of the over capacity penalty, while over\_capacity measures the total amount of demand that is over capacity in a solution for both vehicles and depots. Finally, K is the Boltzmann constant used in the probability function \(e^{-\Delta/(kT)}\) to determine whether to accept a new solution or not. A well-designed algorithm requires a well-thought-out parameter in order for the model to work effectively, and it is crucial to run the algorithm with different parameters to determine an optimal parameter. After several experiments with different parameters, the Table 6 shows the most effective parameter found during the experiment.

<table>
<thead>
<tr>
<th>Table 6: SA Algorithm Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SA1 Parameters</strong></td>
</tr>
<tr>
<td>T0</td>
</tr>
<tr>
<td>Tf</td>
</tr>
<tr>
<td>\alpha</td>
</tr>
<tr>
<td>\beta</td>
</tr>
<tr>
<td>K</td>
</tr>
<tr>
<td>N-prime_operator</td>
</tr>
<tr>
<td>N-local_operator</td>
</tr>
</tbody>
</table>

28
4 Computational Results
The proposed mathematical model and three Simulated Annealing algorithms are implemented in Python 2.7 and run on a PC with an Intel Core i5-6300U (2.40GHz). Gurobi is used to code the mathematical model and test it on randomly generated problems ranging from five to twelve customers with two to four depots and vehicles. Each problem set includes five different randomized instances. In order to assess the proposed SAs, the algorithms are compared on Prins et al (2006). This benchmark is popular for testing algorithms designed for capacitated location routing problems. Since this paper measures latency, we will compare with the results in Moshref-Javadi and Lee (2016) in which they relax the depot capacity constraint. After comparing the three SAs on Prins et al. (2006) problem set, we use the best SA algorithm among the three to compare with the mathematical model on the generated problem set.

4.1 Mathematical Model Results
The mathematical model solves several randomly generated problem instances ranging from five to twelve customers and depots and vehicles ranging from two to four. Figure 7 shows the average computing time of five trials from five to twelve customers. Computing time was capped at 5000 seconds. Figure 7 shows that for the problems with twelve customers, it gets nearly impossible to optimally solve them with reasonable computing power and time, indicating the computation complexity of the location routing problem. In Figure 7, D4V4 means the parameter has four depots and four vehicles. All depots in this mathematical problem instances have a capacity of 140 demands, and all vehicles have a capacity of 70 demands.
Figure 7: Computing Time for Different Number of Depots and Vehicles

Figure 8 shows the values of the total waiting time of the optimal solutions. Unlike the exponential increase of computing time, the total customer-waiting time linearly increases as the size of the problem increases. This figure also shows that to reduce the total customer waiting time, companies can also choose to invest more in resources such as more depots and vehicles. However, adding more resources comes with higher cost, which is not the focus of this research.

Figure 8: Customer Waiting Time for Different Number of Depots and Vehicles
Section 4 - Computational Results

As mentioned above, it is computationally inefficient to use the mathematical model for larger instances. This is where heuristic algorithm compensates for the inadequacies of the mathematical model.

4.2 Simulated Annealing Results
The simulated annealing algorithms are compared on the Prins et al. (2006) benchmark problem set. This benchmark is popular for testing algorithms designed to solve location routing problems.

4.2.1 Comparison of SA Strategies
All three SAs perform competitively when compared with Moshref-Javadi and Lee (2016)’s result. For each problem set, 30 trials were run for the three SAs. The best and average of customer-waiting time and average computation have been recorded, and the results are summarized in Table 7. Although SA1 is adaptive, it is likely adapting too quickly and missing out on operators that could have brought an improvement. SA2 has the most number of best solutions, but the algorithm also takes the longest time to solve, especially on large-sized problems. For example, for 50 and 100 customer nodes problem instances, SA2 takes 66% longer to run the algorithm, but the best solution SA2 found is typically better than SA3 by only less than two percent. SA2 takes a lot more computation time due to its unbiased approach in operators. Unlike SA1 and SA3, SA2 performs local operators on any random route, while SA1 and SA3 only applies the local operators to routes that have been changed by non-local operators. Therefore, SA2 will overall perform more local operators than SA1 and SA3, resulting in longer computation time. SA3 performs similarly to SA2, but with shorter computation time. It also captures both targeting local operators while having unbiased approach in its non-local operators. This mixed strategy turns out to be the best in combination of both computation time and best solution. Thus, SA3 is picked to compare with the results of the mathematical model.
4.2.2 Comparison of SA3 with Mathematical Model

Because SA3 is the most favored algorithm among the three algorithms, this algorithm will compare with the results in the mathematical model. Specifically, four depots and four vehicles versus two depots and two vehicles for ten to twelve customers instances will be compared. The comparison results are shown in Table 8, and the table shows that SA3 is able to solve the same problems as mathematical model in considerably shorter computation time. In fact, for the problem instance 12-2-2-2, SA3 found a better solution as the mathematical model were not able to find an optimal solution within 5000 seconds.

Therefore, the results indicate that the proposed SA algorithm is an efficient method for solving the CLLRP. Figure 9 shows the visual representation of problem instance 10-4-4-5's best solution. Green boxes are opened depots. Red box is a depot not used. Each color of the line represents a vehicle and the direction the vehicle is go. For example, Purple line shows that a vehicle is leaving Depot 5 to deliver to Customer 8 and Customer 7.

4.2.2 Comparison of SA3 with Mathematical Model

Because SA3 is the most favored algorithm among the three algorithms, this algorithm will compare with the results in the mathematical model. Specifically, four depots and four vehicles versus two depots and two vehicles for ten to twelve customers instances will be compared. The comparison results are shown in Table 8, and the table shows that SA3 is able to solve the same problems as mathematical model in considerably shorter computation time. In fact, for the problem instance 12-2-2-2, SA3 found a better solution as the mathematical model were not able to find an optimal solution within 5000 seconds.

Therefore, the results indicate that the proposed SA algorithm is an efficient method for solving the CLLRP. Figure 9 shows the visual representation of problem instance 10-4-4-5’s best solution. Green boxes are opened depots. Red box is a depot not used. Each color of the line represents a vehicle and the direction the vehicle is go. For example, Purple line shows that a vehicle is leaving Depot 5 to deliver to Customer 8 and Customer 7.
### Section 4 - Computational Results

Table 8: Computational Result Comparison Between Mathematical Model and SA3

<table>
<thead>
<tr>
<th>Problem Instance</th>
<th>Customers</th>
<th>Vehicle Capacity</th>
<th>Depot</th>
<th>Open</th>
<th>Math Best</th>
<th>Math Time (seconds)</th>
<th>SA3 Best</th>
<th>Average</th>
<th>SA3 Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-4-4-1</td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>335.85</td>
<td>201.73</td>
<td>335.85</td>
<td>350.70</td>
</tr>
<tr>
<td>10-4-4-2</td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>311.36</td>
<td>189.93</td>
<td>311.36</td>
<td>323.29</td>
</tr>
<tr>
<td>10-4-4-3</td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>229.18</td>
<td>164.33</td>
<td>229.17</td>
<td>241.53</td>
</tr>
<tr>
<td>10-4-4-4</td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>300.48</td>
<td>105.01</td>
<td>300.48</td>
<td>305.68</td>
</tr>
<tr>
<td>10-4-4-5</td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>410.80</td>
<td>1615.89</td>
<td>410.80</td>
<td>410.80</td>
</tr>
<tr>
<td>11-4-4-1</td>
<td>11</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>431.02</td>
<td>296.37</td>
<td>431.02</td>
<td>435.33</td>
</tr>
<tr>
<td>11-4-4-2</td>
<td>11</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>409.13</td>
<td>2732.94</td>
<td>409.13</td>
<td>455.73</td>
</tr>
<tr>
<td>11-4-4-3</td>
<td>11</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>347.75</td>
<td>1582.21</td>
<td>347.75</td>
<td>351.83</td>
</tr>
<tr>
<td>11-4-4-4</td>
<td>11</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>352.06</td>
<td>5000.00</td>
<td>354.27</td>
<td>361.43</td>
</tr>
<tr>
<td>11-4-4-5</td>
<td>11</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>320.14</td>
<td>252.46</td>
<td>320.14</td>
<td>326.05</td>
</tr>
<tr>
<td>12-4-4-1</td>
<td>12</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>502.76</td>
<td>2506.55</td>
<td>510.50</td>
<td>511.26</td>
</tr>
<tr>
<td>12-4-4-2</td>
<td>12</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>419.54</td>
<td>1378.80</td>
<td>419.54</td>
<td>421.57</td>
</tr>
<tr>
<td>12-4-4-3</td>
<td>12</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>469.81</td>
<td>257.62</td>
<td>469.81</td>
<td>482.16</td>
</tr>
<tr>
<td>12-4-4-4</td>
<td>12</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>527.50</td>
<td>5000.00</td>
<td>544.35</td>
<td>552.67</td>
</tr>
<tr>
<td>12-4-4-5</td>
<td>12</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>4</td>
<td>320.14</td>
<td>124.70</td>
<td>320.14</td>
<td>326.63</td>
</tr>
<tr>
<td>10-2-2-1</td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>636.73</td>
<td>414.34</td>
<td>663.65</td>
<td>695.67</td>
</tr>
<tr>
<td>10-2-2-2</td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>501.86</td>
<td>466.19</td>
<td>518.66</td>
<td>575.74</td>
</tr>
<tr>
<td>10-2-2-3</td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>509.46</td>
<td>530.72</td>
<td>509.46</td>
<td>554.22</td>
</tr>
<tr>
<td>10-2-2-4</td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>607.91</td>
<td>805.47</td>
<td>610.02</td>
<td>640.00</td>
</tr>
<tr>
<td>10-2-2-5</td>
<td>10</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>705.46</td>
<td>1385.43</td>
<td>718.57</td>
<td>775.25</td>
</tr>
<tr>
<td>11-2-2-1</td>
<td>11</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>815.19</td>
<td>5000.00</td>
<td>836.54</td>
<td>902.80</td>
</tr>
<tr>
<td>11-2-2-2</td>
<td>11</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>783.97</td>
<td>5000.00</td>
<td>783.97</td>
<td>806.20</td>
</tr>
<tr>
<td>11-2-2-3</td>
<td>11</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>606.27</td>
<td>5000.00</td>
<td>615.49</td>
<td>701.94</td>
</tr>
<tr>
<td>11-2-2-4</td>
<td>11</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>543.13</td>
<td>2451.40</td>
<td>560.47</td>
<td>573.81</td>
</tr>
<tr>
<td>11-2-2-5</td>
<td>11</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>662.55</td>
<td>3711.22</td>
<td>684.39</td>
<td>726.39</td>
</tr>
<tr>
<td>12-2-2-1</td>
<td>12</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>702.49</td>
<td>5000.00</td>
<td>752.75</td>
<td>878.79</td>
</tr>
<tr>
<td>12-2-2-2</td>
<td>12</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>783.51</td>
<td>5000.00</td>
<td>741.05</td>
<td>793.54</td>
</tr>
<tr>
<td>12-2-2-3</td>
<td>12</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>798.43</td>
<td>5000.00</td>
<td>798.43</td>
<td>958.40</td>
</tr>
<tr>
<td>12-2-2-4</td>
<td>12</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>853.99</td>
<td>5000.00</td>
<td>886.14</td>
<td>980.68</td>
</tr>
<tr>
<td>12-2-2-5</td>
<td>12</td>
<td>70</td>
<td>140</td>
<td>5</td>
<td>2</td>
<td>640.29</td>
<td>5000.00</td>
<td>640.29</td>
<td>694.71</td>
</tr>
</tbody>
</table>

**Figure 9:** Visual Representation of 10-4-4-5 Solution
5 Conclusion

In this paper, we considered a customer-centric location routing problem, so-called the Capacitated Latency Location Routing Problem (CLLRP). The CLLRP determines the optimal depot locations and allocation and routing of vehicles to customers. Unlike a typical LRP, this paper aims at minimizing the total customer waiting time, instead of minimizing the total costs. Furthermore, this paper considers capacity constraints on both depots and vehicles compared to the incapacitated latency location routing problem (LLRP) in Moshref-Javadi and Lee (2016). In addition to a mathematical model, three different variations of simulated annealing algorithms were proposed based on the strategies to use the operators of the algorithms. SA1 adaptively chooses the operators, SA2 sequentially applies the non-local and local operators, and SA3 employs the local operators iteratively after the non-local operators. These algorithms use the nearest neighborhood and probabilistic centrality algorithm to find an initial solution which is then improved using five different operators. The algorithms were compared on Prins et al. (2006) benchmark problem set and the best performing SA was then compared with the mathematical model. Comparing with the LLRP results in Moshref-Javadi and Lee (2016), the proposed SAs show promising results with low gap. Overall, SA2 performed the best in finding a solution with minimum waiting time; however, it also took longer time to compute as the customer size increased. SA3 performed as well as SA2 while having lower computation time.

In this research, the costs of facilities and transportation are not considered in the optimization. However, in real world applications, companies have to meet the budget requirement for projects. As a future research, one can consider multi-objective optimization with a focus on both cost and waiting time. Furthermore, this research assumes that all demands are satisfied. However, this assumption is not always true in real life, and
Section 5 - Conclusion

companies have to make a decision on who to serve. This could be another possible future research consideration.
6 REFERENCES


https://doi.org/10.1080/00207543.2014.942010


https://doi.org/10.1016/j.cor.2012.05.008


https://doi.org/10.1016/j.cor.2012.05.009


http://www.jstor.org/stable/1690046


www.elsevier.com/locate/caie