DEVELOPING A SOFTWARE PROTOTYPE OF VEHICLE ROUTING PROBLEM WITH LOADING CONSTRAINTS USING GENETIC ALGORITHMS

Ahmad Rusdiansyah¹, Ira Prasetyaningrum¹, Budi Santosa², De-bi Cao³
¹Logistics and Supply Chain Management (L&SCM) Laboratory, Dept. of Industrial Engineering, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia
Email: arusdianz@gmail.com
²Optimization, Simulation and Data Mining (L&SCM) Laboratory, Dept. of Industrial Engineering, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia,
³Department of Administration Engineering, Faculty of Science and Technology, Keio University, Yokohama, Japan

ABSTRACT
In this research we consider the Vehicle Routing Problem with Loading Constraints (VRPLC). We attempt to integrate the classical Vehicle Routing Problem (VRP) and Container Loading Problem (CLP), which are NP-hard problem respectively, into a single model. The objectives are to minimize the traveling costs and to maximize the utilization of the container space. In this case, all cargos of a customer in the container should be located next to each other and all cargos can be unloaded without moving cargos for other customers. Thus, the cargos of last visited customers should be packed first and those for earlier visited customers should be loaded last. Accordingly, the VRPLC contains complex vehicle routing and packing constraints. To solve the problem, we develop a Genetic Algorithms Heuristic method considering cross-over and mutation operations. Based on the GA algorithms, we finally develop a software prototype that helps users to manage packing process in 3D visualization. Finally, we provide a numerical example to show how the software works.

KEY WORDS
Vehicle Routing Problem with Loading Constraints, Container Loading Problem, Genetic Algorithms, Software Prototype

1. Introduction

Vehicle Routing Problem (VRP) is a classical combinatorial optimization problem concerning the distribution of goods between depots and a set of customers. The basic VRP model is typically described as follows. A set of customers has to be served by a fleet of identical vehicles of limited capacity. The vehicles are initially located at a given depot. Each route begins at the depot, visits a subset of the customers and returns to the depot without violating the capacity constraints. The objective is to find a set of routes for the vehicles so that the total travelled distance is minimized.

The basic VRP model has been extended in the literature over the years. Some additional constraints have been incorporated including those of related to the quality of the service level and the characteristics of the customers and the vehicles. For examples, they are VRP with time windows (VRPTW), VRP with backhauls (VRP), VRP with Simultaneous Deliveries and Pickups (VRPSDP).

Some attempts also have been done to integrate the VRP with other problems. One of them is to integrate the VRP with Container Loading Problem (CLP). The CLP, which is an application of the well-known Bin-Packing Problem (BPP), is a three-dimensional problem that consists of arranging cargos of different sizes inside containers, in such a way so that the volume loaded is optimized. The cargoes are packed into containers and arranged appropriately in terms of its stability in order to minimize the unused space.

The requirement of integrating both problems occurred since the constraints of them are insufficient to satisfy practical needs in freight transportation. At one side, in the VRP, although the total weight of the cargos does not exceed the vehicle weight capacity, however, there is no guarantee that the loads could be feasibly mounted onto the containers. At the other side, in the CLP, the process of arranging cargos into containers does not concern the sequence of customers visited. Thus, some unnecessary reordering cargos of next visited customers is sometimes required when unloading cargos of visited customer. Accordingly, process of unloading of items takes too much time and increases risks of material handling.

The integrated problems will be referred to as the Vehicle Loading Problem with Loading Constraints (VRPLC).
The VRPLC can be specifically described as follows. Consider a central depot consolidating several cargos into trucks that have to be delivered to a set of geographically dispersed customers. The distance and the travel time between the depot and each customer locations, as well as between each pair of customers’ locations are also given. We assume that the shape of the cargos is rectangular where their quantities, weight, and dimensions (width, length and height) have been known when a delivery schedule is determined. The cargos have to be loaded onto a number of trucks whose have identical three-dimensional loading space and weight capacity container. The truck starts from the depot, visiting a set of customers, and returns to the depot. Due to the cargos will be delivered in several destinations of customers, thus, the loading process should be based on the sequence of visited customers. All cargos required by a customer must be placed on the same truck. Precisely, they should be arranged next to each other and can be unloaded without reordering cargos for the next visited customers. Moreover, the cargos of last visited customer should be packed first and those for earlier visited customer should be loaded last. This is called “Last In First Out” (LIFO) rule. Furthermore, in arranging cargos in the container, it is assumed that cargos can only be rotated by 90 degrees on the horizontal plane. For each truck, the total weight of the cargos should not exceed the truck weight capacity and the arrangement of the cargos into the loading space should be feasible. The objective of the VRPLC is to find a set of routes for the trucks so that minimizes the total travel distance and the unused space of each truck.

Many VRP papers have been available in the literature. The reader is suggested to read these following publications on the VRP as well as its derived problems: Toth and Vigo [1] and Cordeau et al.[2]. There are also many articles discussed the CLP such as Chen et al [3], Pisinger [4], Eley [5], Martello [6] and Faroe et al.[7]. Although the VRP and the CLP have been discussed in the literature over the years, unfortunately, there are only a very few papers on VRPLC such as Gendreau et al. [8] and Gendreau et al.[9]. The first paper discussed the VRP with two-dimensional loading constraints (VRP-2LC) while the latter discussed the VRP with three-dimensional loading constraints (VRP-3LC). Two-dimensional loading problems usually arise when the cargos have the same height and the three-dimensional loading problems occur when dealing with cargos having different height, width and length. Gendreau et al. [8] developed a tabu search heuristic for solving VRP-2LC. Some other CRP-2LC works such as Iori et al. [2003] as reported in Gendreau et al. [10] and Doerner[9].To our knowledge, Gendreau et al. [10] is the one and only paper discussing the VRP with three-dimensional loading constraints (VRP-3LC). These findings motivate us to develop a VRPLC model (which can cover both VRP-2LC and VRP-3LC) and the algorithms to solve the problem.

Since the VRP and the CLP have been known respectively as NP-hard problems, consequently, the VRPLC is also an NP-hard problem. We thus preferred to develop a heuristic method than an exact method to solve the VRPLC. Specifically, we use Genetic Algorithms method since the algorithms have been recognized its effectiveness to to solve such complex problems.

The organization of this paper is described as follows. In section 2 we introduce a formulation of VRPLC. In section 3 we describe the heuristic of genetic algorithms. We then briefly describe the design of our software prototype in section 4 based on the GA heuristic. In section 5 some numerical experiments are performed. We conclude the discussion in section 6.

2. Model Formulation

The VRPLC can be defined on a complete and undirected graph $G=(V, E)$, where $V=\{0,...,n\}$ is the vertex set and $E$ is the arc set. Vertex set $V$ consists of vertex subset $\{1,...,n\}$ corresponding to the customers, and vertex 0 corresponding to the depot. Each customer $i \{i=1,...,n\}$ is associated with a set of cargos which need to be loaded into a number of identical containers. The number of containers available in the depot to serve is $K$ indexed $\{k=1,...,K\}$. $L_k, W_k, H_k$ and $B_k$ respectively are parameters indicating the length, width, height, and weight capacity of container $k$. The total number of cargos of all customers is $M$ which is indexed by $\{m=1,...,M\}$. $p_{mi}$, $q_{mi}$, $r_{mi}$, $I_{mi}$ indicate respectively the length, width, height, and weight of cargo $m$ which belongs to customer $i$. We assume that all cargos of a retailer will be loaded to the same container. We also assume that we uplift the cargos using forklift which have weight capacity $f$.

Each arc $(i,j) \in E$ refers to a nonnegative traveling cost between vertex $i$ and vertex $j$, $d_{ij}$, $i,j \in V$. The traveling distance matrix is assumed to satisfy the triangle inequality, $d_{ij} \leq d_{iv} + d_{vj}, \forall i,j,r \in V$. We assume that traveling distance symmetrical: $d_{ij}=d_{ji}, \forall i, j \in V$. There are some other variables used in the formulation as follows:

$$\pi_{mik}$$ A binary variable which is equal to
1 if cargo number $m$ which belong to customer $i$ is placed in container $k$; otherwise it is equal to 0.

$$\omega_{mi}$$ A binary variable which is equal to
1 if cargo number $i$ which belong to
\((a_{mi}, b_{mi}, c_{mi})\) customer \(i\) is placed in column \(\delta\); otherwise it is equal to 0.

Continuous variables indicating the coordinates of the front-left bottom (FLB) corner of cargo \(m\) which belong to customer \(i\).

\((l_{am}, l_{bm}, l_{cm})\) A binary variable indicating whether the length side of cargo \(m\) is parallel to the X-axis, Y-axis or Z-axis. For example, the value of \(l_{bm}=1\) if length side of cargo \(m\) is parallel to the Y-axis, otherwise it is equal to 0.

\((w_{am}, w_{bm}, w_{cm})\) A binary variable indicating whether the width side of cargo \(m\) is parallel to the X-axis, Y-axis or Z-axis. For example, the value of \(w_{bm}=1\) if width side of cargo \(i\) is parallel to the Y-axis, otherwise it is equal to 0.

\((h_{aj}, h_{aj}', h_{aj}''')\) A binary variable indicating whether the height side of cargo \(m\) is parallel to the X-axis, Y-axis or Z-axis. For example, the value of \(h_{bm}=1\) if height side of cargo \(m\) is parallel to the Y-axis, otherwise it is equal to 0.

\(\alpha\) Weight value of the portion of CLP

\(\beta\) Weight value of the portion of VRP

The following is a complete formulation of the VRPLC:

\[
\text{Min } \alpha \cdot \left( \sum_{k \in K} L_k \cdot W_k \cdot H_k - \sum_{m \in M} \sum_{i \in I} \sum_{k \in K} p_{mi} \cdot q_{mi} \cdot r_{mi} \right) + \beta \cdot \sum_{i \in I} \sum_{j \in V} d_{ij} \cdot X_{ij} \tag{1}
\]

Subject to

\[
\sum_{i \in I} X_{ij} = 1 \quad \forall j \in V \setminus \{0\} \tag{2}
\]

\[
\sum_{j \in V} X_{ij} = 1 \quad \forall i \in I \setminus \{0\} \tag{3}
\]

\[
\sum_{i \in S} \sum_{j \in S} X_{ij} \leq |S| - 1 \quad \forall S \subset V \tag{4}
\]

\[
\sum_{i \in I} \sum_{\pi \in \mathbb{P}} \pi_{mik} = 1 \quad \forall m \in M \quad \forall i \in I \tag{5}
\]

\[
(a_{mi} + p_{mi} \cdot l_{am} + q_{mi} \cdot w_{am} + r_{mi} \cdot h_{am}) \cdot \pi_{mik} \leq L_k \quad \forall m \in M, \forall i \in I, \forall k \in K \tag{6}
\]

\[
(b_{mi} + p_{mi} \cdot l_{bm} + q_{mi} \cdot w_{bm} + r_{mi} \cdot h_{bm}) \cdot \pi_{mik} \leq W_k \quad \forall m \in M, \forall i \in I, \forall k \in K \tag{7}
\]

\[
(c_{mi} + p_{mi} \cdot l_{cm} + q_{mi} \cdot w_{cm} + r_{mi} \cdot h_{cm}) \cdot \pi_{mik} \leq H_k \quad \forall m \in M, \forall i \in I, \forall k \in K \tag{8}
\]

\[
\sum_{i \in I} \sum_{m \in M} \omega_{mi} \cdot \pi_{mik} \leq B_k \quad \forall k \in K \tag{9}
\]

\[
\sum_{i \in I} \sum_{m \in M} (\omega_{mi} \cdot \omega_{mik}) = 1 \quad \forall m \in M \quad \forall i \in I \tag{10}
\]

\[
\sum_{i \in V} \sum_{m \in M} (\omega_{mi} \cdot \omega_{mik}) \leq f \quad \forall \delta \tag{11}
\]

\[
l_{am}, l_{bm}, l_{cm}, w_{am}, w_{bm}, w_{cm}, h_{am}, h_{bm}, h_{cm}, \pi_{mik}, \omega_{mik} \in \{0,1\} \tag{12}
\]

\[
a_{mi}, b_{mi}, c_{mi} \geq 0 \tag{13}
\]

\[
X_{ij} \in \{0,1\} \quad \forall i, j \in V \tag{14}
\]

The objective function is to minimize the total unused space of the container(s) used and traveling distance. The indegree and outdegree constraint (2) and (3) impose that exactly one arc enters and each vertex associated with a customer, respectively. Constraint (4) is used to eliminate subtours. Constraint (5) guarantees that each cargo will be packed into exactly one container. Constraints (6)-(8) ensure that all cargos packed into a container fit within the physical dimensions of the container. Constraint (9) ensures that the total weight of all cargos packed into a container not exceed the weight capacity of the container. Constraint (10) guarantees that each cargo will be put in the exactly one column of container \(k\). Constraint (11) ensures that all cargos that will be loaded into a column of a container will not exceed the weight capacity of the forklift.

3. Genetic Algorithms

GA is an adaptive heuristic search method based on population genetics. The representation of the solution space consists of encoding significant features of a solution as a chromosome. It is made up of a sequence of genes capturing the basic characteristic of a solution.

3.1. Chromosome Representation

Specifically, in the VRPLC model, a chromosome represents a sequence of cargos’ index, which is represented as a \(gen\), grouped by customer index. The chromosome should fulfill all constraints of VRPLC. For example, Figure 1 shows a representation of chromosome. Container #1 consists of all cargos of customer #2 (Cargo #3, #4, #5), customer #4 (Cargo #7, #8) and #6 (Cargo #
Step 1: There are three main steps to generate population which can be described as follows.

**Step 1:** We generate the first two chromosomes in population in order to build a sequence of customers to be visited. We apply two different Traveling Salesman Problem (TSP) methods. The first chromosome is generated using Nearest Neighborhood method while the second chromosome is generated using the famous Clarke and Wright’s Savings Heuristics. The rest chromosomes, which also consider the sequence of customers, are generated randomly. Note that, in this case, we have not yet considered weight and dimension constraints. In this step, we have produced the first population.

After having a population where each chromosome ordered by the sequence of customers to be visited, we then apply 3-BP-L heuristics explained in [11] to determine the X,Y,Z coordinates of all cargos in the container(s). For each chromosome, we calculate the fitness function value, which is the objective function, as formulated in (1). Sort the chromosomes in decreasing order of the value of the fitness function.

**Step 2:** In order to produce offsprings that will form the next generation, we recombine the genes of selected parents. The process so called Crossover process combines the characteristics of chromosomes to potentially create offspring with better fitness. In this context, we use Order Crossover type (OX) which is shown in Figure 2.

![Figure 1: Example of Representation of Chromosome](image)

![Figure 2: Illustration of OX](image)

We also consider another type of recombination operation called Mutation. Mutation operation consists of randomly modifying gene(s) of a single chromosome at a time. We further explore the solution space. This operation is also to ensure genetic diversity. Specifically, in our model, we do mutation process by rotating the cargos by 90 degrees on the horizontal plane. In this context, the occurrence of mutation is generated with probability that we determine beforehand. We then again apply 3-BP-L algorithms to determine the X,Y,Z coordinates of all cargos in the container(s) and calculate the fitness function value for each chromosome.

**Step 3:** After Step 2, we now have 3 populations: initial population, A population resulted from the Crossover Operation, and a population resulted from the Mutation Operation. Within these three populations, we compare the fitness function value of each chromosome of a population to the correspondent chromosome of other two populations. We then choose the best chromosomes which produce lower fitness function value to generate a new population of the first generation.

We repeat Step 2 until the number of generation reached. Finally, we obtain the solution, which is the first chromosome of the last population.

### 4. Design of Software

We have developed a software prototype based on our GA heuristics. The software is called “RouteLoad”. There are three main menus which are “Master Data”, “Process”, and “GA Parameters”. The interface of the software can be seen in Figure 3.
4.1. Master Data

The Master Data menu contains the following sub menu:

- **Container Dimension**: This is the facility for maintaining data of types of containers and their dimensions including length, width, height, weight capacity.
- **Forklift Weight Capacity**: This is the facility for inputting the maximum weight capacity of the forklift.
- **Cargo/Box Dimension**: This is the facility for maintaining data of types of cargos and their dimensions including length, width, height and weight.
- **Customer Data**: This is the facility for maintaining data of customers including Name, Address, Phone, Fax, Contact Person, Location coordinates (x,y).

4.2. Process

The Process menu contains the following steps which should be done sequentially.

- **Step 1: Container and Forklift Selection**: This is the facility for selecting the type of container from the database. Only one type is allowed.
- **Step 2: Customer Selection**: This facility is used to choose the customers whose loads will be loaded to the containers.
- **Step 3: Cargo Selection per Customer**: This facility is used to choose the types and number of cargos of each customers which will be loaded to the containers.
- **Step 4: Routing and Loading Process**: This facility is used to process the loading and to determine the number of containers required as well as the routes of each truck. The arrangement of packing in the containers will be shown in X,Y,Z coordinates and in 3D graphics. Users can also follow the arrangement of cargo packing for each container truck step by step. The routes for each container truck can also be seen visually.

4.3. GA Parameters: This is the facility for determining the GA Parameters including: number of generation, convergency, number of population, number of crossover operations, crossover level, number of mutation per generation, mutation level.

5. Numerical Example

In order to demonstrate how the software is used, we give an example. Suppose that a distribution company will distribute cargos to 13 dispersed located customers using container trucks. The truck use a standard “20’ dry steel” container which dimension (length, width and height) is 589 cm, 235 cm and 239 cm respectively. The weight capacity of the container is 24000 Kgs. To uplift the cargos, we use a forklift having weight capacity 350 kgs. Each customer requests a number of cargos in different size. There are total 25 cargos will be loaded.

We assume that the weight of all cargos types is less than the forklift weight capacity. We also assume that the total weight of the cargos uplifted by the forklift should not less than the the forklift weight capacity. Table 1 shows customer data including Customer’s name, Customer’s...
The solution can be described as follows. The container trucks required are two units. The first container truck will go through this following tour: Depot → Customer2 → Customer1 → Customer6 → Customer3 → Customer5 → Customer4 → Depot, while the second container truck will go through this following tour: Depot → Customer7 → Customer13 → Customer12 → Customer8 → Customer10 → Depot. The tours can be seen visually in figure 5.

Figures 6 and 7 show the arrangement of cargos packed in Container#1 and Container#2 respectively. We may see in these figures that the cargos are packed based on the sequence of visited customers. All cargos required by a customer were placed on the same truck and arranged next to each other so that they were unloaded without reordering cargos for the next visited customers. The cargos are packed in stable conditions. We may see also that the cargos of the last visited customer should be packed first and those for earlier visited customers should be loaded last. Accordingly, unnecessary movements when unloading cargos have been eliminated. Moreover, the visual 3D output will guide users (i.e. packagers) easily when doing loading and unloading process of the cargos.

8. Conclusion

In this research we have developed a software prototype of Vehicle Routing Problem with Loading Constraints (VRPLC). The VRPLC is an integrated model between the classical Vehicle Routing Problem (VRP) and Container Loading Problem (CLP). The objective of the VRPLC is to find a set of routes for the trucks so that minimizes the total travel distance and the unused space.
of each container truck. We have solved the VRPLC, which is known as a NP Hard problem, using Genetic Algorithms method. We have demonstrated that the software can fulfill effectively all requirements required by VRPLC. Moreover, the 3D visual feature of the software can help users to load and unload the cargos. Thus, the loading and unloading process can be performed effectively. As our future research, we will examine the performance of our proposed heuristics. We plan to test the algorithm using standard data tests of VRPLC.

Acknowledgements

The authors would like to thank you to the research grant received from the PHK-A3 Program of the Department of Industrial Engineering, Institut Teknologi Sepuluh Nopember (ITS) Surabaya. We also acknowledge the support from the department for the research travel grant received.

References