International Journal of Service and Knowledge Management
International Institute of Applied Informatics
2023, Vol. 7, No. 2, IJSKM710

Solution Algorithm for Vehicle Routing Problem with Stochastic Demand

Masahiro Komatsu *, Ryota Omori *,
Tetsuya Sato *, Takayuki Shiina *

Abstract

The vehicle routing problem (VRP) determines a delivery route that minimizes the delivery cost. In this study, we consider the stochastic VRP with uncertainty and consider the variation in customer demand, which may cause a shortage of products during delivery. In this case, delivery vehicles have to return to the depot and replenish the products. We consider a model that minimizes the sum of the additional cost caused by the shortage and the normal delivery cost. In previous studies, the decomposition method using the L-shaped method was used. In this study, we improve the decomposition method to make it more efficient. In addition, we have improved the direct method of calculating the additional cost without the decomposition method by considering subtour elimination constraints. We have shown that the direct method is superior in terms of time-saving.

Keywords: vehicle routing problem, VRP, subtour elimination constraint, stochastic programming, stochastic demand

1 Introduction

In recent years, the demand for logistics and home delivery has been increasing, and solving the vehicle routing problem (VRP) (Toth and Vigo [1]) has become important. The VRP determines a delivery route that minimizes the delivery cost. In this study, we investigate the stochastic VRP with uncertainty and consider the variation in customer demand, which may cause a shortage of products during delivery. In this case, the delivery vehicles have to return to the depot and replenish the products. We consider a model that minimizes the sum of the additional cost caused by the shortage and the normal delivery cost.

In previous studies, the decomposition method using the L-shaped method was used. In this study, we improve the decomposition method to make it more efficient. Moreover, we improved the direct method of calculating the additional cost without using the decomposition method (Omori et al. [2]) based on subtour elimination constraints. We show that the direct method is superior in terms of time-saving. It is also shown new and previously unobtainable insights pertaining to subtour elimination constraints.

* Waseda University, Tokyo, Japan

2 Previous Studies

Laporte et al. [3] deal with a model that considers fluctuations in demand in a VRP with limited vehicle capacity. The additional cost is calculated by adding up the cost incurred whenever the cumulative delivery demand exceeds the vehicle capacity. Therefore, this model approximates the additional cost and does not minimize it. The optimality cut used in thier L-shaped algorithm was to eliminate a current non-optimal solution.

Omori et al. [2] formulated the model that customer demand fluctuates and a shortage occurs during delivery as a problem of minimizing the sum of the cost of delivery and the additional delivery cost caused by returning to the depot, based on the train operation planning of Giacco et al. [4]. Moreover, they showed that directly calculating the additional cost without the integer L-shaped decomposition method is superior in terms of time-saving.

Laporte et al. [5] formulated the algorithm for three time-varying VRP models and introduced a optimality cut in the L-shaped algorithm that includes the expected value of the objective function of the subproblem obtained at each iteration. In this paper, the optimality cut calculating the expectation of the two-stage problem is used and the cost can be accurately approximated.

Roberti and Toth [6] focused on the subtour elimination constraints in the asymmetric traveling salesman problem (ATSP) and introduced the flow formulation (ff) constraint and the Miller-Tucker-Zemlin (MTZ) constraint, which are polynomial-order subtour elimination constraints.

3 Stochastic Programming

3.1 Stochastic Programming

Stochastic programming deals with optimization under uncertainty. A stochastic programming with recorse is referred to as a two-stage stochastic programming. The basic two-stage stochastic linear programming with recourse (SPR) is formulated as follows.

$$
\begin{aligned}
\text{(SPR):min} \quad & E_{\tilde{\xi}}[c^T x + Q(x,\tilde{\xi})], && (1)\\
\text{s.t.} \quad & Ax = b, && (2)\\
& x \geq 0, && (3)\\
& Q(x,\xi) = \min\{q^T y(\xi) | W y(\xi) = h(\xi) - T(\xi)x, \\
& \qquad y(\xi) \geq 0\}, \qquad \xi \in \Xi, && (4)
\end{aligned}
$$

The first stage decisions are represented by x. We asuume the random vector $\tilde{\xi}$ is defined on a probability space. Let Ξ be the support of $\tilde{\xi}$. ξ is defined as the realization of random vector $\tilde{\xi}$.

Given a first stage decision x, the realization of random vector ξ of $\tilde{\xi}$ is observed. The second stage data ξ become known. Then, the second stage decision $y(\xi)$ must be taken so as to satisfy the constraints $Wy(\xi) \geq \xi - Tx$ and $y(\xi) \geq 0$. The second stage decision $y(\xi)$ is assumed to cause a penalty of q. The objective function contains a deterministic term $c^\top x$ and the expectation of the second stage objective. The symbol $E_{\tilde{\xi}}$ represents the mathematical expectation with respect to $\tilde{\xi}$, and the function $Q(x,\xi)$ is called the recourse function in state ξ. The value of the recourse function is given by solving a second stage linear programming problem.

3.2 L-shaped Method

It is assumed the random vector $\tilde{\xi}$ has a discrete distribution with finite support $\Xi = \{\xi^1, ..., \xi^K\}$ with $\mathrm{Prob}(\tilde{\xi} = \xi^k) = p^k (k = 1, ..., K)$. A paticular realization ξ of the random vector $\tilde{\xi}$ is called a scenario. The deterministic equivalent problem (DEP) for (SRP) is formulated as follows.

$$
\begin{aligned}
\text{(DEP):min} \quad & c^T x + \mathscr{Q}(x), && (5)\\
\text{s.t.} \quad & Ax = b, && (6)\\
& x \geq 0, && (7)\\
& \mathscr{Q}(x) = \sum_{k=1}^{K} p^k Q(x, \xi^k), && (8)\\
& Q(x, \xi^k) = \min\{q^T y(\xi^k) | W y(\xi^k) = h(\xi^k) - T(\xi^k) x, \\
& \qquad y(\xi^k) \geq 0\}, \qquad k = 1, ..., K, && (9)
\end{aligned}
$$

To solve (DEP), an L-shaped method based on Benders decomposition has been used. In the L-shaped method, we solve the following (Master) problem. θ is defined as the upper bound for the expected recorse function such that $\theta \geq \sum_{k=1}^{K} p^k Q(x, \xi^k)$. Let (x^*, θ^*) be the optimal solution of the (Master) problem. If the second stage problem is infeasible, the feasible cut is added to the (Master) problem. If the second stage problem is feasible and $\theta^* < p^k Q(x^*, \xi^k)$, the optimality cut is added to the (Master) problem.

$$
\begin{aligned}
\text{(Master):min} \quad & c^T x + \theta, && (10)\\
\text{s.t.} \quad & Ax = b, && (11)\\
& x \geq 0, && (12)\\
& \theta \geq 0, && (13)
\end{aligned}
$$

4 Model

4.1 Problem Description

A depot, customers, and delivery routes are represented in a directed graph, $G = (V, A)$. We denote multiple arcs from node i to j, which represent customers or depots, as $(i, j, z), z \in Z$. After a delivery vehicle leaves the depot, it visits each customer once and returns to the depot, satisfying the demand of customers. However, as the customer demand stochastically fluctuates, a shortage may occur during delivery. When this shortage happens, the vehicle has to return to the depot and replenish the products. We minimize the sum of the additional cost for this shortage and the normal delivery cost. To calculate the additional cost, we define multiple arcs connecting the depot to customers and customers to other customers. There are two types of multiple arcs: those that go through the depot and those that do not go through the depot. The former arcs denote delivery vehicles that return to the depot and replenish products.

4.2 Notations

Variable

x_{ij}	0-1 variable indicating whether the vehicle moves from customer i to customer j
w^s_{ijz}	0-1 variable indicating whether the vehicle moves from customer i to customer j through arc z in scenario s. (When $z = 1$, the vehicle does not go through the depot. When $z = 2$, it goes through the depot, as shown in Figure 1.)
g^s_{ijz}	In scenario s, when the vehicle passes through edge (i, j, z), the total volume of deliveries made to customer i after the last departure of the depot.

Set

V	The set of depot and all customers $\{0, 1, ..., n\}$, where 0 represents a depot
V_0	The set of all customers$\{1, ..., n\}$
A	The set of all arcs connecting the depot to the customer and the customer to the depot
A_z	The set of arcs z connecting the depot to the customer and the customer to the depot (When $z = 1$, the vehicle goes through the depo. When $z = 2$, it goes through the depot, $A = A_1 \cup A_2$).

Parameter

c_{ijz}	The travel cost $(c_{ijz} = c_{jiz})$ when passing through the arc z from customer i to customer j.
d^s_j	The stochastic demand for customer j in scenario s
p^s	The probability of scenario s $(\sum_{s \in \Xi} p^s = 1)$
Q	The capacity of the vehicle
C	The lower bound of vehicle's inventory

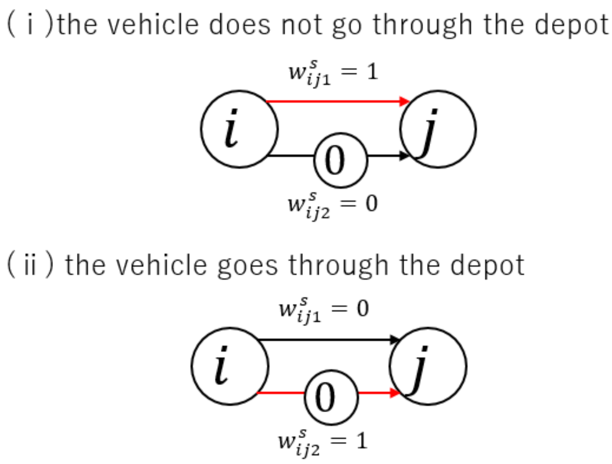

Figure 1: variable w^s_{ijz}

4.3 Formulation

$$\min \quad \sum_{(i,j)\in A_1} c_{ij1}x_{ij} + \sum_{s\in\Xi} p^s \sum_{(i,j,z)\in A} (c_{ij2}-c_{ij1})w^s_{ij2}, \tag{14}$$

$$\text{s.t.} \quad \sum_{i\in V\setminus\{j\}} x_{ij} = 1, \qquad \forall j \in V, \tag{15}$$

$$\sum_{k\in V\setminus\{j\}} x_{jk} = 1, \qquad \forall j \in V, \tag{16}$$

$$x_{ij} + x_{ji} \le 1, \qquad \forall i \in V, \forall j \in V\setminus\{i\}, \tag{17}$$

$$\sum_{i\in S}\sum_{j\notin S}(x_{ij}+x_{ji}) \ge 2, \qquad \forall S \subset V, |S| \ge 3, \tag{18}$$

$$x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A_1, \tag{19}$$

$$\sum_{k\in V\setminus\{j\}} \sum_{z:(j,k,z)\in A} g^s_{jkz} = d^s_j + \sum_{i\in V_0\setminus\{j\}} g^s_{ij1}, \qquad \forall s \in \Xi, \forall j \in V_0, \tag{20}$$

$$g^s_{ij1} \le (Q-C)w^s_{ij1}, \qquad \forall s \in \Xi, \forall (i,j) \in A_1, \tag{21}$$

$$g^s_{ij2} \le Qw^s_{ij2}, \qquad \forall s \in \Xi, \forall (i,j) \in A_2, \tag{22}$$

$$w^s_{ij1} + w^s_{ij2} = x_{ij}, \qquad \forall s \in \Xi, \forall (i,j,z) \in A, \tag{23}$$

$$w^s_{ijz} \in \{0,1\}, \qquad \forall s \in \Xi, \forall (i,j,z) \in A. \tag{24}$$

The first term in objective function (14) defines the normal delivery cost, the second term is the expected value of the additional cost owing to the shortage of products, and the sum of these two terms is minimized. Equations (15) and (16) are constraints that visit each customer once, (17) is a constraint that eliminates subtour between two points, (18) is a constraint that eliminates subtour among three or more points, (20)-(23) are constraints in the total delivery volume, and (19) and (24) are binary conditions with respect to the variables.

4.4 Total Delivery Volume Constraints

As shown in Figure2, Left side of equation (20) indicates the total delivery volume up to customer j after the vehicle leaves the depot. The first term of right side indicates the demand for customer j. The second term of right side indicates the total delivery volume up to customer i when the vehicle does not go through the depot.

When the vehicle does not go through the depot, if the total delivery volume up to customer i is G (as shown in Figure3), then $g^s_{ij1} = G$ and $g^s_{ij2} = 0$. Therefore, the total delivery volume up to customer j is $d^s_j + G$. When the vehicle goes through the depot, $g^s_{ij1} = 0$ and $g^s_{ij2} = G$. Therefore, the total delivery volume up to customer j is d^s_j.

Inequalities (21) and (22) are constraints on the upper bound of g^s_{ij1} and g^s_{ij2}. Inequality (21) indicates that when the vehicle moves from i to j, the total delivery volume up to i must be less than or equal to $(Q-C)$. Inequality (22) indicates that the total delivery volume up to i must be less than or equal to Q.

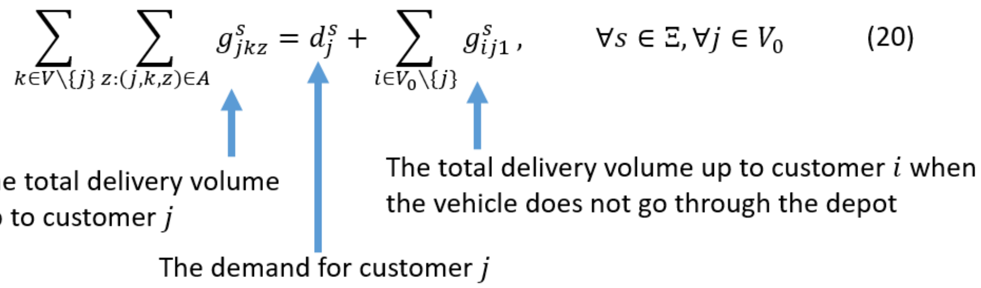

Figure 2: Clarification of Equation (20)

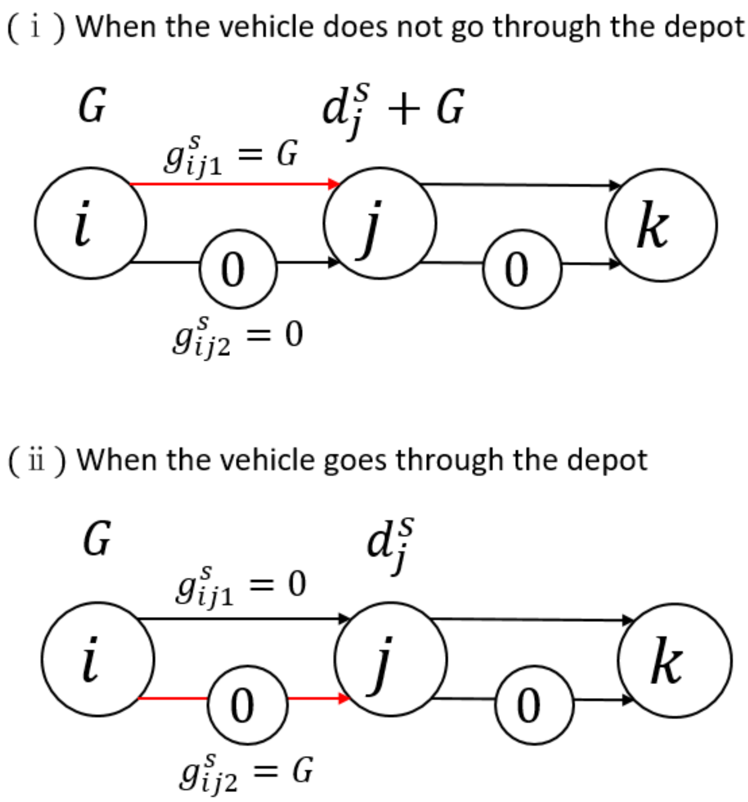

Figure 3: Calculation of Total Delivery Volume

5 Decomposition Method

5.1 Problem Description

In this section, the model formulated by Omori et al. [2] is solved as two-stage stochastic programming using the integer L-shape method. First, the master problem is solved, and then, the feasible solution is obtained by solving subproblems based on the obtained solution. Compared with the conventional decomposition method, the optimality cut is modified to include q^r (as described below) based on the method proposed by Laporte et al. [5]. Moreover, we improved the algorithms of Omori et al. [2] and Laporte et al. [5].

5.2 Notations

Variable

θ	The upper bound of the additional cost caused by the shortage
θ^s	The upper bound of the additional cost caused by the shortage in scenario s

Set

S^r	The set of (i,j) such that $x_{ij} = 1$ in the r-th feasible solution.

Parameter

q^r	The expectation of the r-th two-stage problem
N	The number of feasibility cuts

5.3 Formulation

Master Problem

$$\min \quad \sum_{(i,j)\in A_1} c_{ij1}x_{ij} + \theta, \tag{25}$$

$$\text{s.t.} \quad \sum_{i\in V\setminus\{j\}} x_{ij} = 1, \quad \forall j \in V, \tag{26}$$

$$\sum_{k\in V\setminus\{j\}} x_{jk} = 1, \quad \forall j \in V, \tag{27}$$

$$x_{ij} + x_{ji} \le 1, \quad \forall i \in V, \forall j \in V\setminus\{i\}, \tag{28}$$

$$\sum_{i\in S}\sum_{j\notin S}(x_{ij} + x_{ji}) \ge 2, \quad \forall S \subset V, |S| \ge 3, \tag{29}$$

$$\theta \ge q^r \left(\sum_{(i,j)\in S_r} x_{ij} - n \right), \quad \forall r = 1,2,...,N \tag{30}$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in A_1. \tag{31}$$

Subproblem

$$\min \quad \theta^s = \sum_{(i,j,z)\in A} (c_{ij2} - c_{ij1})w^s_{ij2}, \tag{32}$$

$$\sum_{k\in V\setminus\{j\}} \sum_{z:(i,j,z)\in A} g^s_{jkz} = d^s_j + \sum_{i\in V_0\setminus\{j\}} g^s_{ij1}, \quad \forall s \in \Xi, \forall j \in V_0, \tag{33}$$

$$g^s_{ij1} \le (Q-C)w^s_{ij1}, \quad \forall s \in \Xi, \forall (i,j) \in A_1, \tag{34}$$

$$g^s_{ij2} \le Qw^s_{ij2}, \quad \forall s \in \Xi, \forall (i,j) \in A_2, \tag{35}$$

$$w^s_{ij1} + w^s_{ij2} = x_{ij}, \quad \forall s \in \Xi, \forall (i,j,z) \in A, \tag{36}$$

$$w^s_{ijz} \in \{0,1\}, \quad \forall s \in \Xi, \forall (i,j,z) \in A. \tag{37}$$

(30) defines the newly introduced optimality cut. When x_{ij} is the same as the rth feasible solution, the value of $\sum_{(i,j)\in S_r} x_{ij}$ is $n+1$, and the right-hand side is q^r. Thus, for the r-th feasible solution, (30) provides a lower bound, q^r, for θ of the additional cost. The algorithm of the L-shaped method considering the integer constraint is shown below. The algorithm is based on Omori et al. [2], Laporte et al. [3], Laporte et al. [5], and Laporte et al. [7].

.

Step 0 Set the number of iterations $\nu := 0$ and the cost of the best-known solution $\bar{z} := \infty$.
Step 1 Set $\nu := \nu + 1$ and solve the relaxation problem except for (18). Let (x^ν, θ^ν) be the optimal solution of master problem.
Step 2. Check whether the obtained optimal solution satisfies (18), and if not, add a cut and return to Step 1. Otherwise, if $cx^\nu + \theta^\nu \geq \bar{z}$, then $\bar{z}$ is the optimal value, and the algorithm terminates. Otherwise, go to Step 3.
Step 3. Solve the subproblem for each scenario, $Q(x^\nu) := \sum_{s\in\Xi} p^s\theta^s$, $z^\nu := \sum_{(i,j)\in A_1} c_{ij1}x_{ij} + Q(x^\nu)$. If $z^\nu \leq \bar{z}$, then $\bar{z} := z^\nu$.
Step 4. If $\theta^\nu < Q(x^\nu)$, set $r := r+1$, and add an optimality cut to the master problem. Otherwise, $\bar{z}$ is the optimal value, and the algorithm terminates.
Step 5 Let $(x^{\nu\prime}, \theta^{\nu\prime})$ be the optimal solution obtained by solving the master problem in opposite order. Replace (x^ν, θ^ν) with $(x^{\nu\prime}, \theta^{\nu\prime})$ and perform steps 3 and 4 again. After adding the optimality cuts to the master problem, return to Step 1.

Figure 4: Proposed algorithm

The conventional L-shaped method has been improved; i.e., two feasible solutions are obtained for each solution of the master problem. As symmetry is assumed for the delivery cost between each customer, the costs of normal delivery for one delivery route and the reverse order will be the same. However, the additional costs may be different. Therefore, whenever we solve the master problem, subproblems based on the obtained solution and the solution that follows the opposite order need to be solved. Therefore, two additional optimality cuts are obtained. Based on this algorithm, the number of times the master problem is solved can be approximately halved, thus expediting calculation.

6 Subtour Elimination Constraint

6.1 Constraints Based on Cuts

Inequality (38), which is a subtour elimination constraint, is usually used. As this is a constraint on all the subsets of two or more points, it requires $2^{n+1} - (n+1) - 2$ constraint formulas. If we add all of them for a problem with large n, the number of constraint formulas will be huge. Therefore, we solve the problem without enumerating the subtour elimination constraints and determine whether the obtained solution satisfies the subtour elimination constraints; if not, we add a cut and solve the problem again. This reduces the computation time.

$$\sum_{i\notin S}\sum_{j\in S}(x_{ij} + x_{ji}) \geq 2, \qquad \forall S \subset V, |S| \geq 2 \tag{38}$$

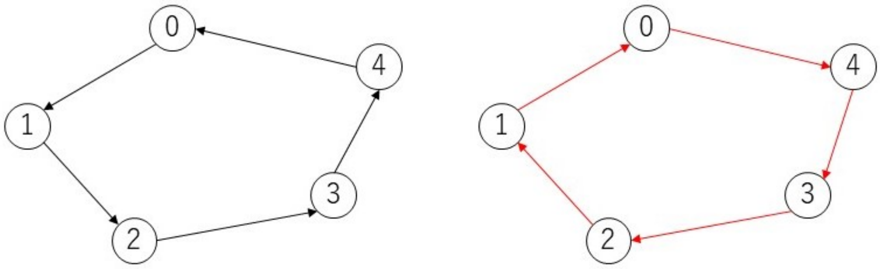

Figure 5: One route and the opposite route

6.2 ff Constraint

The ff constraint (Gavish and Graves [8]) considers a problem that starts at a depot, collects one object from each point, and brings n objects back to the depot. These constraints play the role of subtour elimination constraints because if a subtour exists, it is not possible to bring n objects back to the depot. The constraints are defined in (39)-(42). y_{ij} is the number of objects possessed by the vehicle while moving from customer i to customer j (it is 0 when there is no movement from customer i to customer j).

$$y_{ij} \leq n x_{ij}, \qquad \forall (i,j) \in A_1 \tag{39}$$

$$\sum_{k \in V \setminus \{j\}} y_{jk} - \sum_{i \in V \setminus \{j\}} y_{ij} = 1, \qquad \forall j \in V_0 \tag{40}$$

$$\sum_{k \in V_0} y_{0k} - \sum_{i \in V_0} y_{i0} = -n, \tag{41}$$

$$\sum_{j=1}^{n} y_{0j} = 0, \tag{42}$$

Inequality (39) is the constraint on the upper bound of y_{ij}, and (40) is the constraint that the difference between the number of objects in possession after and before visiting a customer is 1 for all the customers. (41) is the constraint that the difference between the number of objects leaving the depot and returning to the depot is $-n$. (42) is the constraint that the number of goods when leaving the depot is zero. The number of constraint (39) and (40) are $n(n+1)$ and n, respectively. Therefore, the total number of constraints is n^2+2n+2.

6.3 MTZ Constraints

The MTZ constraint (Miller et al. [9]) defines a weight, $u_i (\in \mathbb{R})$, of each customer except for the depot. Inequality (43) is the MTZ constraint.

$$u_i - u_j + (n+1)x_{ij} \leq n, \qquad \forall i, j \in V_0, \tag{43}$$

When subtours exist, there is at least one subtour that does not contain a depot. If the number of points that constitutes the subtour is k, then adding the equations in (43) for the branches that make up the subtour results in $(n+1)k \leq nk$, which contradicts. Therefore, by adding this constraint, we can eliminate the subtour. Inequality (43) includes $n(n-1)$ formulas.

7 Numerical Experiments

7.1 Experimental Environment

The experimental environment is as follows.
OS: Windows10 64bit
CPU: Core i7-3770S(3.10GHz)8.00GB
Modeling language: AMPL
Solver: CPLEX12.10.0.0.

7.2 Experimental Data

Customers are assumed based on a 2-dimensional space $[0,100]^2$, and the coordinates of the depot are $(50,50)$. The coordinates of the customer are randomly determined using uniform distribution. The delivery cost between the depot and the customer and between the customers is proportional to the distance between the two points.

$$c_{ij1} = \sqrt{|x_i - x_j|^2 + |y_i - y_j|^2},$$
$$c_{ij2} = \sqrt{|x_i - x_0|^2 + |y_i - y_0|^2} + \sqrt{|x_j - x_0|^2 + |y_j - y_0|^2}.$$

The vehicle capacity Q and the lower bound of the vehicle inventory C are fixed at the following values:

$$Q = 500,$$
$$C = 0.$$

We define α as the ratio of the expected value of total demand to vehicle capacity. When the expected value of demand for all the customers is $\bar{d}$, α is represented by the equation defined below. By changing the value of α, we can change the value of customer demand.

$$\alpha = \frac{n\bar{d}}{Q}.$$

The value of the demand independent of the scenario and the customer is given by the following equation

$$d_j^s = \text{round}\left(\text{Uniform}\left(\frac{1}{2}\bar{d}, \frac{3}{2}\bar{d}\right)\right), \qquad \forall s \in \Xi, \forall j \in V_0.$$

7.3 Comparison Among Decomposition Methods

We compare the computational efficiency of the conventional decomposition method proposed by Omori et al. [2](Table 1), our decomposition method (Table 2), and the direct solution method without decomposition (Table 3). In the three methods, the subtour elimination constraints between two points (17) are included in advance, while the subtour elimination constraints between three or more points (18) are excluded from the relaxation problem, and cuts are added when the constraints are violated. ”SEC cuts” in Tables 1 and 2 indicate the number of cuts for subtour elimination constraints. ”Master” in the table indicates the number of times the master problem is solved.

Table 1: conventional decomposition method

α	customer	scenario	SEC cuts	optimality cuts	Master	time (seconds)
1	10	3	5	24	27	3
1	10	5	7	38	42	8
1	10	7	5	6	9	1
1	20	3	27	976	989	937
1	20	5	23	690	701	558
1	20	7	23	570	581	460
1	30	3	46	770	790	979
1	30	5	30	86	98	47
1	30	7	30	1004	1014	1251

Table 2: proposed decomposition method

α	customer	scenario	SEC cuts	optimality cuts	Master	time (seconds)
1	10	3	5	24	15	3
1	10	5	7	38	23	7
1	10	7	5	6	6	1
1	20	3	27	978	502	533
1	20	5	23	692	357	337
1	20	7	23	570	296	293
1	30	3	46	770	405	542
1	30	5	30	86	55	35
1	30	7	30	1004	512	842

Table 3: direct solution method

α	customer	scenario	SEC cuts	Master	time (seconds)
1	10	3	0	1	1
1	10	5	0	1	1
1	10	7	0	1	1
1	20	3	0	1	4
1	20	5	0	1	12
1	20	7	0	1	17
1	30	3	0	1	32
1	30	5	0	1	48
1	30	7	0	1	37

As listed in Tables 1, 2, and 3, the proposed decomposition method is faster than the conventional decomposition method and slower than the direct decomposition method. This is because constraint (20) acts like a subtour elimination constraint. Both decomposition methods generate several cuts of the subtour elimination constraint to solve the problem, but the direct solving method does not generate any cuts of the subtour elimination constraint. Therefore, the direct method is faster in terms of computation time.

7.4 Comparison Among Subtour Elimination Constraints

We compare the computational efficiency of the normal subtour elimination, ff constraints, and MTZ constraints. For the conventional subtour elimination constraint, the subtour elimination constraints between two points (17) are included in advance, and the subtour elimination constraints between three or more points (18) are excluded from the relaxation problem; cuts are added when the constraints are violated, as shown in the experiment of the comparison among decomposition methods. SEC cuts indicate the number of cuts for subtour elimination constraints.

We obtain the result that the normal subtour elimination constraint is faster for most problems. This is because constraint (20) on the total delivery volume acts like a subtour elimination constraint.

Table 4: Comparison Among Subtour Elimination Constraints

α	customer	scenario	normal SEC cuts	normal SEC (seconds)	ff (seconds)	MTZ (seconds)
1	10	3	0	1	1	1
1	10	5	0	1	1	1
1	10	7	0	1	1	1
1	20	3	0	4	3	13
1	20	5	0	12	9	12
1	20	7	0	27	14	19
1	30	3	0	32	21	55
1	30	5	0	4	51	83
1	30	7	0	144	395	223
2	10	3	0	4	7	6
2	10	5	0	38	44	38
2	10	7	0	72	71	68
2	20	3	0	3291	500	553
2	20	5	0	3068	3881	4988
2	20	7	0	3152	7916	5258
2	30	3	2	2722	3885	10962
2	30	5	0	2643	7288	9331
2	30	7	0	5598	18382	13475

7.5 Considerations

The reason that constraint (20) based on the total delivery volume acts like a subtour elimination constraint has been explained. Let us assume that we have a subtour of 1→2→3→1 (Figure 6).

In this case, we enumerate (20) for each customer as follows:

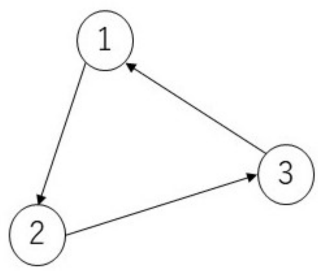

Figure 6: subtour (1→2→3→1)

$$g^s_{121} + g^s_{122} = d^s_1 + g^s_{311}, \tag{44}$$
$$g^s_{231} + g^s_{232} = d^s_2 + g^s_{121}, \tag{45}$$
$$g^s_{311} + g^s_{312} = d^s_3 + g^s_{231}. \tag{46}$$

If there are no arcs that return to the depot, $w^s_{121}, w^s_{231}, w^s_{311} = 1$, $w^s_{122}, w^s_{232}, w^s_{312} = 0$, and $g^s_{122}, g^s_{232}, g^s_{312} = 0$. Substituting this into (44)-(46), we obtain the following equations.

$$g^s_{121} = d^s_1 + g^s_{311}, \tag{47}$$
$$g^s_{231} = d^s_2 + g^s_{121}, \tag{48}$$
$$g^s_{311} = d^s_3 + g^s_{231}. \tag{49}$$

Based on $d^s_1, d^s_2, d^s_3 > 0$, these three equations do not hold at the same time and are contradictory. Therefore, we cannot have a subtour without arcs that return to the depot.

Conversely, consider the case where there is an arc that returns to the depot. For example, when moving from customer 1 to customer 2, we go through the depot. Then, $w^s_{122}, w^s_{231}, w^s_{311} = 1$, $w^s_{121}, w^s_{232}, w^s_{312} = 0$, and $g^s_{121}, g^s_{232}, g^s_{312} = 0$. Substituting this into (44)-(46), we obtain the following equations.

$$g^s_{122} = d^s_1 + g^s_{311}, \tag{50}$$
$$g^s_{231} = d^s_2, \tag{51}$$
$$g^s_{311} = d^s_3 + g^s_{231}. \tag{52}$$

Solving these three equations, we obtain $g^s_{231} = d^s_2$, $g^s_{311} = d^s_3 + d^s_2$, and $g^s_{122} = d^s_1 + d^s_3 + d^s_2$, which are consistent. Therefore, when there is an arc that returns to a depot, there is a possibility that a subtour can be formed.

Therefore, when (20) is included in the solution, there is no subtour that does not have an arc that returns to the depot, and (20) acts like a subtour elimination constraint. However, (20) is not a complete subtour elimination, therefore, it is necessary to include a subtour elimination constraint.

8 Conclusion and Future Tasks

The proposed decomposition method can be improved compared with the conventional decomposition method, but the direct method is even faster in computation time. As for the subtour elimination constraints in the direct method, the normal subtour elimination constraints are often faster than the ff and MTZ constraints. Constraint (20) in the total delivery volume makes it difficult to create a subtour.

We consider a method that sequentially adds a subtour elimination constraint between two points as a cut. To make the model more realistic, we can enhance it, so that when a shortage occurs, the vehicle goes back and forth between the customer and the depot where the shortage has occurred.

References

[1] Toth, P., Vigo, D.: Vehicle Routing Problems, Methods, and Applications. Second Edition, Society for Industrial and Applied Mathematics (2014)

[2] Omori, R., Shiina, T.: New Formulation for the Vehicle Routing Problem with Stochastic Demands, 9th International Congress on Advanced Applied Informatics (IIAI-AAI), 723–728 (2020)

[3] Laporte, G., Louveaux, F., Hamme, L.: An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands, Operations Research, 50, 415–423 (2002)

[4] Giacco, G.L., D'Ariano, A., Pacciarelli, D.: Rolling stock rostering optimization under maintenance constraints, Journal of Intelligent Transportation Systems, 18(1), 95–105 (2014)

[5] Laporte, G., Louveaux, F., Mercure, H.: The vehicle routing problem with stochastic travel times, Transportation Science, 26, 161–170 (1992)

[6] Roberti, R., Toth, T.: Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison, EURO Journal on Transportation and Logistics, 1, 113–133 (2012)

[7] Laporte, G., Louveaux, F.: The integer L-shaped method for stochastic integer programs with complete resource, Operations Research Letters, 13, 133–142 (1993)

[8] Gavish, B., Graves, S.C.: The traveling salesman problem and related problems, Working Paper OR-078-78, Operations Research Center, MIT, Cambridge, MA(1978)

[9] Miller, C.E., Tucker, A.W., Zemlin, R.A.: Integer programming formulations and travelling salesman problems, Journal of the ACM, 7, 326–329 (1960)

[10] Shiina, T.: Stochastic Programming, Asakura Publishing (2015) (in Japanese)