Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 6329203, 16 pages
https://doi.org/10.1155/2017/6329203

Research Article

Vehicle Routing Problems with Fuel Consumption and Stochastic Travel Speeds

Yanling Feng, Ren-Qian Zhang, and Guozhu Jia

School of Economics and Management, Beihang University, Beijing 100191, China

Correspondence should be addressed to Guozhu Jia; by1408109@buaa.edu.cn

Received 30 July 2016; Revised 8 December 2016; Accepted 15 December 2016; Published 29 January 2017

Academic Editor: Guillermo Botella-Juan

Conventional vehicle routing problems (VRP) always assume that the vehicle travel speed is fixed or time-dependent on arcs. However, due to the uncertainty of weather, traffic conditions, and other random factors, it is not appropriate to set travel speeds to fixed constants in advance. Consequently, we propose a mathematic model for calculating expected fuel consumption and fixed vehicle cost where average speed is assumed to obey normal distribution on each arc which is more realistic than the existing model. For small-scaled problems, we make a linear transformation and solve them by existing solver CPLEX, while, for large-scaled problems, an improved simulated annealing (ISA) algorithm is constructed. Finally, instances from real road networks of England are performed with the ISA algorithm. Computational results show that our ISA algorithm performs well in a reasonable amount of time. We also find that when taking stochastic speeds into consideration, the fuel consumption is always larger than that with fixed speed model.

1. Introduction

The vehicle routing problem (VRP) is one of the most important and studied combinatorial optimization problems [1]. It has gained great attentions from many researchers, especially in distribution and logistics management fields. More than fifty years have elapsed since Dantzig and Ramser [2] firstly introduced the problem in in 1959. They described a real-world application concerning the delivery of gasoline to service stations and proposed the first mathematical programming formulation and algorithmic approach. Following this seminal paper, hundreds of models and algorithms were proposed for the optimal and approximate solution of the different versions of VRP. Then in order to be better aligned with the real-world applications, many different VRP versions have been studied. The most common version is the CVRP [3], where each vehicle has limited load capacity. The VRP with Time Window (VRPTW) [4–6] aims to find optimal route sets with minimum total travel cost while serving each customer within specified time window. Other extended versions include VRP with backhauls [7], VRP with pickups and deliveries [8], and the multidepot VRP [9].

The deterministic VRP cannot cover all the situations in reality while considering stochastic VRP components. Consequently, Stochastic VRP are developed. For example, Ritzinger et al. [10] made a detailed review of the stochastic VRP. Mehrjerdi [11] combined chance constrained programming and multiple objective programming to obtain satisfactory solutions. Taş et al. [12, 13], Ehmke et al. [14], and Laporte et al. [15] put forward different heuristic methods for VPR with stochastic travel time and time windows. Marinakis et al. [16] developed a particle swarm algorithm for VRP with stochastic demands. For more stochastic demand results, one may refer to [17–19].

Based on the NP hardness of VRP, multiple heuristic algorithms have been put forward to solve this problem. Xiao and Konak [20] present simulating annealing algorithm to solve the green vehicle routing and scheduling problem with hierarchical objectives and weighted tardiness. Kondekar et al. [21] provide a mapreduce based hybrid genetic solution for solving large-scale vehicle routing problems in dynamic network with fluctuant link travel time. Neural network is also applied to solve stochastic multiconstraint problems with different time-scales; see Zhang et al. [22] and Meyer-Bäse et al. [23].

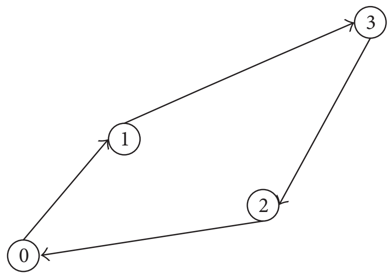


Figure 1: Illustration of a delivery.

Demir et al. [24] proposed an adaptive large neighborhood search algorithm (ALNS) to minimize the fuel consumption and the driving time with Pareto optimality. Garaix et al. [25] proposed a column generation algorithm for a dial-a-ride problem.

Due to the uncertain traffic factors, such as unexpected workload or bad weather, stochastic VRP has attracted more and more attentions. Cao et al. [26] proposed a partial Lagrange multiplier method. Ishigaki [27] considered a dynamic collection plan with stochastic demand and apply their search algorithm to an actual trash collection problem. Novel applications of the well-established problem can also be found in optical flow and vehicle systems [28–30].

In most literatures, travel speeds are assumed to be fixed or time-dependent (see, e.g., [31, 32]). However, in practice, due to the uncertainty of weather, traffic conditions, and other random factors, it is not appropriate to set travel speeds to be fixed constants in advance. The interest in stochastic VRP in this paper is motivated by both its practical relevance and its considerable difficulty: large VRP instances may be solved to optimality only in particular cases. Therefore, we study vehicle routing problems with stochastic travel speeds. Moreover, as each arc has limited speed and other random factors, the average speed of the same type of vehicles on the same arc approximately obeys normal distribution.

Figure 1 is an illustration of this VPR with stochastic average travel speed. Assume that a logistics company owns a fleet of trucks and one truck delivers goods for customers 1, 2, and 3. On the first day, the truck travels with an average speed of 25 m/s on arc(0, 1). On the second day, it rains heavily when it delivers goods for customer 1. The truck has to get over the poor road conditions caused by the heavy rain; as a result, the average speed it travels on arc(0, 1) becomes 15 m/s. Several days later, a traffic accident occurred on arc(0, 1), so it may travel with an average speed of 10 m/s. Practically, during a relatively long period of time, the average speed on arc(0, 1) is not fixed but with slight fluctuation. From this point of view, the average speed is a stochastic variable. Here we assume that the average speed on each arc follows normal distribution.

On the other hand, environmental issues have become worldwide problems. In fact, fuel consumption of traffic vehicles is a great contributor to CO_2 emissions which is the main culprit of global warming [33, 34]. As a result, we take the sum of the expected fuel consumption and total vehicle cost as the objective to evaluate different routes. It has been studied that the fuel consumed of a vehicle traveling along a route depends on many factors, which include distance, load, speed, road conditions, and vehicle types. Models can be found in Bektaş and Laporte [34] and Xiao et al. [35].

Table 1 summarizes and compares the formulation of VRP in the stream of uncertain VRP and green VRP.

Two main contributions of this paper are described as follows:

(1) Based on the fuel consumption model introduced by Bektaş and Laporte [34], the fuel consumption is a nonlinear function of travel speed, distance, and vehicle load. Due to the stochasticity of the travel speeds, we extend the model with stochastic speeds considered.

(2) An ISA (simulated annealing algorithm) is presented to solve large-scaled VRP problems.

The remainder of this paper is as follows. In Section 2, the description and formulation of this model is provided. In Section 3, we made a linearization of this problem. Then Section 4 introduces the improved simulated annealing algorithm with memory for optimizing the routing plan. In Section 5, the computational experiments are performed. Finally, Section 6 presents the conclusions and managerial insights.

2. Problem Statement and Model Formulation

Generally speaking, logistics companies tend to make budget decisions for a planning period; for example, they figure out the amount of the fuel consumption, the cost of keeping, and maintaining a fleet of vehicles in advance. Thus, we try to develop a fuel consumption model and algorithm to obtain more precise budgets. We use a digraph to describe the vehicle routing problems with fuel consumption and stochastic speeds (VRPFSV). Let a complete connected digraph $G = (N, A)$ be the logistics network with a node set $N = \{0, 1, \ldots, n\}$ and an arc set $A = \{(i, j) \mid i, j \in N,\ i \neq j\}$. Arc$(i, j)$ represents the path from node i to node j. The depot is denoted by node 0 and the other nodes in $N/\{0\}$ represent n customers with nonnegative demand q_i. The depot owns enough homogeneous vehicles with limited capacity Q, so the total demands of customers assigned to the same vehicle must be less than or equal to Q. Each route is finished by only one vehicle and each customer is served only once. The average travel speed v_{ij} on arc(i, j) is stochastic. After serving all the assigned customers, each vehicle has to return to the depot.

2.1. Assumptions and Notations. In this section, the assumptions and notations used in the model formulation are listed. First, we assume the following:

(1) Each demand must be satisfied and each customer is served only once.

(2) The depot owns enough homogenous vehicles.

(3) Each vehicle must departure from the depot and after having served its customers it must return to the depot.

(4) Time window constraints are not considered here.

Table 1: A comparison of existing vehicle routing models.

	Reference	Stochastic travel speed	Stochastic service time	Stochastic demand	Objective function	Weight	Time window	Independent decision variables	Solution approach
Uncertain VRP	Figliozzi [36]	No	Yes	No	Vehicle cost; travel time	No	Yes	Routes; service time	New metaheuristic
	Errico et al. [37]	No	Yes	No	Cost	No	Yes	Routes; service time	Column generation; branch-price-and-cut algorithms
	Marinaki and Marinakis [38]	No	No	Yes	Expected length	Yes	No	Route	Glowworm swarm optimization
	Sarasola et al. [39]	No	No	Yes	Total distance; lateness penalty	No	No	Route	Variable neighborhood search
	Zhang et al. [19]	No	No	Yes	Travel time; earliness penalty; lateness penalty	No	Yes	Route	Dynamic programming
Green VRP	Bektaş and Laporte [34]	No	No	No	Driver cost; operational cost; emissions cost	Yes	Yes	Route; start time	CPLEX solver
	Jabali et al. [40];	No	Yes	No	Driver cost; fuel cost; emissions cost	No	Yes	Routes; speed; route start time	Tabu search
	Xiao et al. [35]; Kuo [31]	No	No	No	Fuel consumption	Yes	No	Routes	Simulated annealing
	Franceschetti et al. [41]	No	Only two fixed periods	No	Emissions cost; driver wage	Yes	Yes	Routes; speed; departure time	MIP & DSOP algorithm
	Xiao and Abdullah [42]	No (time-dependent speed)	No	No	CO_2 emissions; travel time; travel distance	Yes	Yes	Routes; travel distance in each period; time	Dynamic programming
	This paper	Yes	No	No	Fuel consumption; vehicle cost	Yes	No	Routes; vehicle load on each arc	MIP & improved simulated annealing

Then the notations used in the VRPFSV formulation are listed as follows:

n: total number of customers.

i: index of nodes (the depot is represented by 0).

f: fixed cost of using a vehicle.

Q: the capacity of a vehicle.

N: set of nodes including the depot, $i \in N$.

A: set of arcs formed by all pairs of nodes, $(i, j) \in A$, $\forall i \in N$, $j \in N$, $i \neq j$.

l_{ij}: distance between node i and node j.

v_{ij}: stochastic average travel speed on arc(i, j).

Two decision variables are as follows:

x_{ij}: binary variable indicating whether arc(i, j) is traveled. If arc(i, j) is traveled by a vehicle, then $x_{ij} = 1$; otherwise, $x_{ij} = 0$.

Q_{ij}: load carried from node i to node j.

2.2. Formulation of the Model of VRPFSV. In this section, a mixed integer linear programming model is developed to

Table 2: Values and notations of parameters.

Notation	Description	Typical values
W	Curb-weight (kilogram)	6350
Ξ	Fuel-to-air mass ratio	1
K	Engineer friction factor (kilojoule/rev/liter)	0.2
N	Engineer speed (meter/second^2)	33
V	Engineer displacement (liters)	5
G	Gravitational constant (meter/second^2)	9.81
C_d	Coefficient of aerodynamic drag	0.7
P	Air density (kilogram/meter^3)	1.204
A	Frontal surface area (meter^2)	3.912
C_r	Coefficient of rolling resistance	0.01
n_{tf}	Vehicle drive train efficiency	0.4
H	Efficiency parameter for diesel engines	0.9
c	Fuel and CO_2 emissions cost per liter (£)	1.4
K	Heating value of a typical diesel fuel (kilojoule/gram)	44
Ψ	Conversion factor (gram/second to liter/second)	737
$v^{\min}$	Lower speed limit (meter/second)	5
$v^{\max}$	Upper speed limit (meter/second)	25

Source: Demir et al., 2012 [24].

construct the VRPFSV. The VRPFSV is an extension of the classical VRP and assumes that average travel speed v_{ij} on arc(i, j) is a nonnegative random variable. Furthermore, the average travel speed v_{ij} on arc(i, j) is assumed to obey normal distribution with mean u_{ij} and variance σ_{ij}^2.

The fuel consumption model applied in this paper is based on the comprehensive model formed by Barth and Boriboonsomsin [33]. It is given by

$$Z_{ij}\left(v_{ij}\right) = \frac{\lambda\left(kNV + w\gamma\alpha_{ij}v_{ij} + \gamma\alpha_{ij}f_{ij}v_{ij} + \beta\gamma v_{ij}^3\right)d_{ij}}{v_{ij}}, \quad (1)$$

where $\lambda = \xi/\kappa\psi$ and $\gamma = 1/1000\eta_{tf}\eta$ are constants decided by different fuel properties, $\beta = 0.5C_d\rho A$, and w are vehicle characteristics related constants. α_{ij} is a constant associated with the road characteristics and acceleration, $\alpha_{ij} = \tau_{ij} + g\sin\theta_{ij} + gC_r\cos\theta_{ij}$, where τ_{ij} is acceleration and θ_{ij} is the road angle inclination. The reference values of all parameters are given in Table 2. Assuming that acceleration and road inclination are zeros, then the fuel consumption on arc(i, j) can be rewritten as

$$Z_{ij}\left(v_{ij}\right) = \left(\frac{\omega_1}{v_{ij}} + \omega_2 + \omega_3 f_{ij} + \omega_4 v_{ij}^2\right)d_{ij}, \quad (2)$$

where $\omega_1 = \lambda kNV$, $\omega_2 = \lambda w\gamma gC_r$, $\omega_3 = \lambda\gamma gC_r$, $\omega_4 = \lambda\beta\gamma$, and the price of the fuel $c = 1.4$£$/L$. Equation (2) establishes a good correlation between fuel consumption and travel speed, weight, and distance.

In our research, the service cost includes two components: the expected fuel consumption and the fixed vehicle cost.

Consider arc(i, j); if it is traveled by a vehicle with average speed of v_{ij}, then it has an expected fuel consumption; according to (2), the expected fuel consumption is given in

$$E\left[F_{ij}\left(v_{ij}\right)\right] = \int_{v_{ij}^{\min}}^{v_{ij}^{\max}} l_{ij}\left(\frac{w_1}{v_{ij}} + w_2 + w_3 Q_{ij} + w_4 v_{ij}^2\right) \cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij}. \quad (3)$$

Consequently, the mathematical formulation of VRPFSV can be expressed as follows:

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} cE\left[F_{ij}\left(v_{ij}\right)\right]x_{ij} + f\sum_{j=1}^{n} x_{0j} \quad (4)$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} = 1, \quad \forall i \in N/\{0\} \quad (5)$$

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = 0, \quad \forall i \in N \quad (6)$$

$$\sum_{j=1, j\neq i}^{n} Q_{ji} - \sum_{j=1, j\neq i}^{n} Q_{ij} = D_i, \quad \forall i \in N/\{0\} \quad (7)$$

$$Q_{ij} \le Qx_{ij}, \quad \forall (i, j) \in A \quad (8)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \quad (9)$$

$$Q_{ij} \ge 0, \quad \forall (i, j) \in A. \quad (10)$$

In this optimization model, the two variable sets are x_{ij} and Q_{ij}. Equation (4) is the objective function which consists of the expected fuel consumption and the total vehicle cost. $x_{0j} = 1$ indicates a departure of a newly used vehicle from the depot; as any used vehicle will trigger a fixed cost, the sum of x_{0j} represents the total vehicle cost. Constraint (5) represents that the vehicle visits each node (except the depot) only once. Equation (6) represents the conservation of flow constraints. Equation (7) ensures that the demand of each customer must be met and it also eliminates subtours. Equation (8) guarantees that the vehicle load cannot exceed the vehicle capacity Q. Equation (9) indicates that x_{ij} is a binary variable. Equation (10) is the nonnegative constraint of vehicle load.

3. Linearization and Solution by CEPLEX

Since the VRP is NP-hard and this VRPFSV model includes stochastic travel speed, it is at least NP-hard. In this section, we transform the original model into a linear one, and then small instances can be solved easily by CEPLEX 12.6.2.

This VRPFSV model is nonlinear due to the existence item $x_{ij}Q_{ij}$ in the objective. In fact, as the Hessian matrix of the objective is nonpositive, the VRPFSV model is even not convex. Fortunately, we manage to make the linearization and obtain the linear model which is equivalent to the original one.

For any given arc(i, j), consider its expected fuel consumption:

$$\begin{aligned} E\left[F_{ij}\left(v_{ij}\right)\right] &= x_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} l_{ij}\left(\frac{\omega_1}{v_{ij}}+\omega_2+\omega_3 Q_{ij}+\omega_4 v_{ij}^2\right) \\ &\cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij} \\ &= x_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} l_{ij}\left(\frac{\omega_1}{v_{ij}}+\omega_2+\omega_4 v_{ij}^2\right) \\ &\cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij} + x_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} l_{ij}\omega_3 Q_{ij} \\ &\cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij} \\ &= x_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} l_{ij}\left(\frac{\omega_1}{v_{ij}}+\omega_2+\omega_4 v_{ij}^2\right) \\ &\cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij} + Q_{ij}x_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} l_{ij}\omega_3 \\ &\cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij}. \end{aligned} \tag{11}$$

In fact, the total fuel consumption can be reduced as follows:

$$\begin{aligned} E\left[F_{ij}\left(v_{ij}\right)\right] &= x_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} l_{ij}\left(\frac{w_1}{v_{ij}}+w_2+w_4 v_{ij}^2\right) \\ &\cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij} + Q_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} l_{ij}w_3 \\ &\cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij}. \end{aligned} \tag{12}$$

Property 1. In the VRPFSV, (11) is equivalent to (12), subjecting to constraint (8).

Proof of Property 1. In order to prove this property, we only need to prove that the product of Q_{ij} and x_{ij} is equivalent to Q_{ij}. In fact, x_{ij} is a binary variable; if $x_{ij} = 1$, then $x_{ij}Q_{ij} = Q_{ij}$, else if $x_{ij} = 0$, due to constraints (8) and (10), we get $Q_{ij} = 0$. Consequently, $x_{ij}Q_{ij} = 0 = Q_{ij}$. Thus, $x_{ij}Q_{ij}$ is equivalent to Q_{ij} which indicates that (11) is equivalent to (12).

According to (12), objective (4) is converted to the equivalent linear form:

$$\begin{aligned} &\sum_{i=1}^{n}\sum_{j=1}^{n} cE\left[F_{ij}\left(v_{ij}\right)\right]x_{ij} + f\sum_{j=1}^{n} x_{0j} \\ &= \sum_{i=0}^{n}\sum_{j=0}^{n} cx_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} l_{ij}\left(\frac{\omega_1}{v_{ij}}+\omega_2+\omega_4 v_{ij}^2\right) \\ &\cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{n} cQ_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} l_{ij}\omega_3 \\ &\cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij} + f\sum_{j=1}^{n} x_{0j}. \end{aligned} \tag{13}$$

Let

$$\begin{aligned} C_1 &= cl_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} \left(\frac{\omega_1}{v_{ij}}+\omega_2+\omega_4 v_{ij}^2\right) \\ &\cdot \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij} \end{aligned} \tag{14}$$

$$C_2 = cl_{ij}\int_{v_{ij}^{\min}}^{v_{ij}^{\max}} \omega_3 \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-(v_{ij}-u_{ij})^2/2\sigma_{ij}{}^2} dv_{ij}. \tag{15}$$

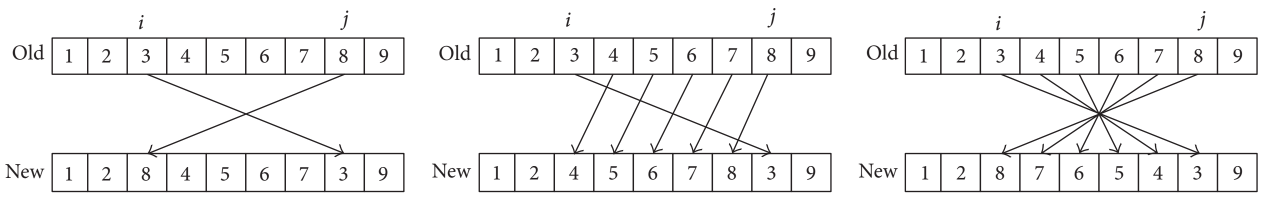
As a result, objective (4) can be transformed into $\sum_{i=1}^{n}\sum_{j=1}^{n} cE[F_{ij}(v_{ij})]x_{ij} + f\sum_{j=1}^{n} x_{0j} = C_1\sum_{i=0}^{n}\sum_{j=0}^{n} x_{ij} + C_2\sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij} + f\sum_{j=1}^{n} x_{0j}$, which is a linear objective function about x_{ij} and Q_{ij} now. Moreover, constraints (5)–(10) are linear about the decision variables, so we established a linear mixed integer programming model for VRPFSV.

Then the following intractable problem is how to calculate C_1 and C_2, as it is unable to get the closed form of the integral in (14). In the following section, we use numerical integration to obtain and store the value of C_1 and C_2 on every arc in advance.

Now all the expressions in the model are linear, so we got the coefficient value in the objective. It is possible to optimally solve problem VRPFSV by the existing solver CPLEX (version 12.6.2) for small sized problem. The linearization plays a significant role in finding optimal solution and reducing computation time. □

4. ISA Algorithm for Large-Scaled VRPFSV

For large-scaled instances, the existing solver will go expired, as the computation process will cost a very large amount of time. We develop an algorithm based on SA algorithm. This ISA algorithm includes four parts, construction of the initial solution, generating neighborhood solutions, local search, and replacement of the current best solution. And theoretically, the SA algorithm is guaranteed to converge to the global optimization solution with probability one [43].



(a) Swap (b) Relocation (c) 2-opt

Figure 2: Three exchange rules.

4.1. Construction of Initial Solution. In the string model, the unique depot is encoded as 0 and the customers are encoded as a series of positive numbers. The string vector which represents the initial route is formatted as $S = \{0 \ \text{index} \ \cdots \ \text{index} \ 0\}$ and the length of S is $n + 2$.

Then make S feasible for VRPFSV; that is, assign n customers to appropriate number of vehicles. Usually, vehicle load contributes a lot to the fuel consumption. For example, if a vehicle carries a heavy load starting from the depot, it manages to serve more customers. On the contrary, if the initial load is small, more vehicles will be needed because of the capacity limit. According to vehicle capacity, string S is transformed to feasible solution with the following procedures. First, start from the second character of S, and then accumulate the demands of nodes successively. If the cumulative demands exceed the vehicle capacity, start a new vehicle; that is, insert a 0 into string S before the current character and then reset the cumulative demands to 0. Repeat the above steps until the end of S. Take $n = 10$ as an example, assuming that $S = \{0 \ 3 \ 4 \ 1 \ 7 \ 5 \ 9 \ 8 \ 10 \ 6 \ 2 \ 0\}$ is the initial string. As we do not know how many zeros are needed to be inserted into S initially, we set the length of VRPX to the largest likely length 21. Because the cumulative demands of 3, 4, and 1 are less than the capacity and the cumulative demands of 3, 4, 1, and 7 exceed the capacity, 0 is inserted between 1 and 7. Continue checking until the end of S; then we can obtain $\text{VRPS} = \{0 \ 3 \ 4 \ 1 \ 0 \ 7 \ 5 \ 0 \ 9 \ 8 \ 10 \ 0 \ 6 \ 2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0\}$, which represents ten customers being visited by four vehicles.

4.2. Generate Neighborhood Solutions. Exchange rule: three commonly used exchange rules swap, relocation, and 2-opt (see Figure 2) are used to configure the neighborhood solutions. After exchanging, we get a neighborhood solution, and then make it feasible following steps of Section 4.1. Finally, the neighborhood solutions are obtained.

4.3. Local Search and Update the Current Best Solution. The objective function value is the criterion to evaluate which solution is better. If the total cost becomes smaller, the newly generated solution is accepted. Otherwise, the solution is accepted with certain probability. As poor solutions maybe accepted at certain probability, so the best of the output cannot be guaranteed. Consequently, we use a memory array $\text{VRPS}^{\text{best}}$ in the algorithm to record the best solution. Only when a better solution appears, will $\text{VRPS}^{\text{best}}$ be updated. After continuous improvement, we can get the best solution in all searched neighborhoods.

Furthermore, it is not so easy to calculate the vehicle load on each arc, so once a route is generated, the load will be calculated in a reverse order. More details are listed in in Appendix B.

This algorithm adopts adaptive cooling processing, and the temperature cooling coefficient τ can be set to $A/(A + \mathit{loop})$, where A is the accepted solution numbers of the current temperature and *loop* is the total loop numbers. In this way, if the number of accepted solutions is small, that is, the current solution is close to optimal solution, the coefficient may lead to smaller search scope which may cost less time. On the other hand, if the number of accepted solutions is big, a relatively big cooling coefficient will help expand the search scope.

The main steps of this improved simulated annealing algorithm are as follows.

Step 1. Initialize parameters. Input customers with demand D and distance matrix L, vehicle capacity Q, fixed single vehicle cost f, speed distribution parameters of each arc u and σ^2, the end temperature T_{end}, and the number of inner loops *loop*.

Step 2. Generate initial string S according to the method introduced in Section 4.1.

Step 3. Generate initial temperature. Randomly change S to its neighbor for 1000 times; the maximum objective deviation is selected as the initial temperature.

Step 4. Whether the inner loop number is reached, if it is, go to Step 8. Otherwise, generate new solutions and transform it into feasible solution.

Step 5. Calculate the deviation between the current cost and the former one and accept or reject it according to Metropolis rule.

Step 6. Update the best so far solution.

Step 7. Whether the improvement is less than 0.01, if it is, go to Step 9.

Step 8. Whether the end temperature is reached, if so, go to Step 9. Otherwise, drop the temperature and go to Step 4.

Step 9. Stop and output the best solution. The pseudocode of the improved annealing algorithm is listed in Appendix A.

In Step 3, we use self-adaptive method to obtain the best initial temperature. As too high initial temperature will increase the computational time and too low initial temperature will trap in local optimal solution. Here, we generate the maximum temperature adaptively according to the problem; that is, randomly change the incumbent solution for 1000 times. The maximum cost deviation between two neighboring solution is selected as the initial temperature.

Moreover, the temperature drops adaptively, under certain temperature; if the number of solutions being accepted is large, then the temperature drops very quickly. Else, if the number of accepted solutions is small, the temperature drops relatively slowly.

In Step 5, let C_i and C_{now} be the objective value of the ith iteration and the best objective of the accepted routes, and $\Delta C = C_i - C_{\text{now}}$. VRPS^{now} and VRPS^{new} denote, respectively, the minimum cost route among the accepted ones and the ith route. If $\Delta C < 0$, then replace VRPS^{now} with VRPS^{new}. Else, replace VRPS^{now} with VRPS^{new} if $\text{random}(0, 1) < (T_{\text{now}} - T_{\text{end}})/(T_0 - T_{\text{end}})$.

The algorithm terminates on two conditions: the current temperature is below or equal to the end temperature T_{end}, or the improvement of the best solution is less than 0.01.

Theory 2. In the VRPFSV, the time complexity of the proposed SA algorithm is $O(n)$.

Proof. In the outer temperature loop, the number of temperatures in the cooling procedure is $\log_\tau(T_0/T_{\text{end}})$; in the internal loop, the number of loops is loop, and each loop has a calculation of cost function C with time complexity $O(n)$. Thus, the total time complexity becomes $O(n \text{ loop } \log_\tau (T_0/T_{\text{end}}))$. As loop, τ, T_0, and T_{end} are all given constants, Theory 2 is proved. □

5. Computational Experiments of the ISA Algorithm

In this section, the results of computational applications of ISA algorithm are presented.

First, we introduce the Pollution-Routing Problem Instance Library (PRPLIB) (http://www.apollo.management.soton.ac.uk/prplib.htm). This library contains nine different-scale groups of instances of the Pollution-Routing Problem (PRP). Each group consists of 20 different instances. These instances are based on real distances collected from UK cities. The first number in the file name after UK shows the number of nodes contained in the instance. The second is the order number of the instance within the group. The problem consists of n nodes, and the format of each file includes data about number of customers, vehicle capacity, city name and demand, distance matrix, and minimum and maximum speed level.

As there are no suitable benchmark problems of this model, we modified the benchmark problems of the PRPLIB to test our algorithm. Suppose that v_{ij} is the average travel speed on arc(i, j) and it obeys normal distribution with mean parameter u_{ij} and variance parameter σ_{ij}.

In these examples, speed limit of all arcs is 5 m/s~25 m/s. For each arc, the integer mean parameters u_{ij} are generated randomly from interval [5, 25] and set $\sigma_{ij} = 1$. At the same time, the other parameters stay unchanged. The ISA algorithm was implemented in MATLAB and executed on an Intel 2.0 GHz processor with 1.59 G of RAM.

5.1. Quality and Efficiency of the ISA Algorithm.

Tables 2 and 3 compared results of the optimal solutions found by CPLEX, SA, and ISA algorithm when n is small. The SA and ISA algorithm are performed 10 times for each instance. The average and best objective value of the 10 runs are listed in the tables, respectively. We label the metrics associated with each objective as follows: the first column is the instance ID, and the second and third columns are the optimal objective value and computation time of solutions solved by the solver CPLEX. Avg-Obj is the average total cost found by our SA or ISA algorithm in 10 runs. Best-Obj is the best objective of the 10 runs. Dev. is the relative deviations between the best objective obtained in 10 runs and the optimal cost solved by CPLEX or SA; that is, $\text{Dev} = ((\text{Obj}^{\text{ISA best}} - \text{Obj}^i)/\text{Obj}^i) \times 100\%$, $i = \text{CEPLX}$ or SA best. Avg-CPU time is the average run time (CPU) of the SA or ISA algorithm.

5.1.1. Parameter Settings.

As different parameter settings may influence the performance of ISA heuristic, we test different parameter values in advance. Furthermore, the initial temperature and temperature drop factor are generated adaptively; we only need to determine the appropriate values of *loop* and T_{end}. We change one parameter at a time to observe how that parameter affects the solution. It can be noticed from Figure 3 that using a larger *loop* or a smaller T_{end} may improve the solution quality a little but increase the computation time significantly.

By trading off the effectiveness and the efficiency of the algorithm, we found the values of $loop = 100$ and $T_{\text{end}} = 0.1$ are appropriate parameters combination.

As analyzed above, we set $loop = 100$, $T_{\text{end}} = 0.1$, $c = 1.4$, and $f = 1$. ω_1, ω_2, ω_3, ω_4 are set to the same value as mentioned in Section 2.2; that is, $\omega_1 = 1.0176 \times 10^{-3}$, $\omega_2 = 5.3360 \times 10^{-5}$, $\omega_3 = 8.4032 \times 10^{-9}$, and $\omega_4 = 1.4122 \times 10^{-7}$. In order to compare the results obtained by SA and ISA, we set the same parameter with the same value both in SA and in ISA. Thus, the parameters in SA is as follows: $T_0 = 1000$, $loop = 100$, and $T_{\text{end}} = 0.1$.

In Table 3, the CPLEX is always superior to SA and ISA in both objective and computation time, and the largest relative deviation between ISA and CPLEX is 5.88%. Table 4 depicts in most cases that the ISA manage to obtain the optimal solution and the computation time turns out to be less than CPLEX. Bold numbers indicate that, in four instances, the ISA algorithm finds the optimal solution. In all cases, ISA consumes much less time than classic SA while obtaining the same solution.

It is shown in Table 5 that when $n = 25$, the largest deviation of the ISA and CPLEX is 1.46%, and the ISA obtains

Table 3: Comparison results when $n = 10$.

Instance ID	CPLEX		Solutions found by SA in 10 runs			Solutions found by ISA in 10 runs			Dev. from CPLEX	Dev. from SA
	Obj (£)	CPU time (s)	Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)	Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)		
UK10_01	99.63	1.39	99.63	99.63	33.01	99.96	**99.63**	13.61	0.00%	0.00%
UK10_02	124.46	0.33	124.46	124.46	22.46	124.46	**124.46**	17.17	0.00%	0.00%
UK10_03	100.81	0.47	100.97	100.97	32.81	101.35	100.97	5.36	0.15%	0.00%
UK10_04	100.18	0.81	106.07	106.07	33.37	106.07	106.07	4.57	5.88%	0.00%
UK10_05	107.19	1.22	109.69	109.69	42.67	109.69	109.69	22.07	2.33%	0.00%

Table 4: Comparison results when $n = 15$.

Instance ID	CPLEX		Solutions found by SA in 10 runs			Solutions found by ISA in 10 runs			Dev. from CPLEX	Dev. from SA
	Obj (£)	CPU time (s)	Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)	Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)		
UK15_01	161.86	18.42	159.59	159.59	33.01	160.16	159.59	13.51	1.40%	0.00%
UK15_02	109.19	0.8	109.87	109.87	22.46	109.89	109.87	25.05	0.62%	0.00%
UK15_03	167.96	46.66	167.96	167.96	32.81	170.22	**167.96**	6.43	0.00%	0.00%
UK15_04	172.26	83.89	172.55	172.26	33.37	174.52	172.26	7.91	0.00%	0.00%
UK15_05	166.73	0.88	166.73	166.73	42.67	166.95	**166.73**	7.25	0.00%	0.00%

Table 5: Comparison results when $n = 25$.

Instance ID	CPLEX		Solutions found by SA in 10 runs			Solutions found by ISA in 10 runs			Dev. from CPLEX	Dev. from SA
	Obj (£)	CPU time (s)	Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)	Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)		
UK25_01	153.24	4.16	155.25	155.11	61.85	156	**153.24**	22.63	0.00%	−1.21%
UK25_02	185.74	41.39	189.7	187.6	52.25	189.6	**185.74**	25.05	0.00%	−1.01%
UK25_03	105.6	16.62	107.46	106.44	51.69	108.2	105.59	20.25	0.01%	−0.80%
UK25_04	134.63	9.73	138.38	136.26	51.90	137.6	136.59	31.36	1.46%	0.24%
UK25_05	187.95	24.97	189.68	189.15	53.63	192.2	**187.95**	76.98	0.00%	−0.63%

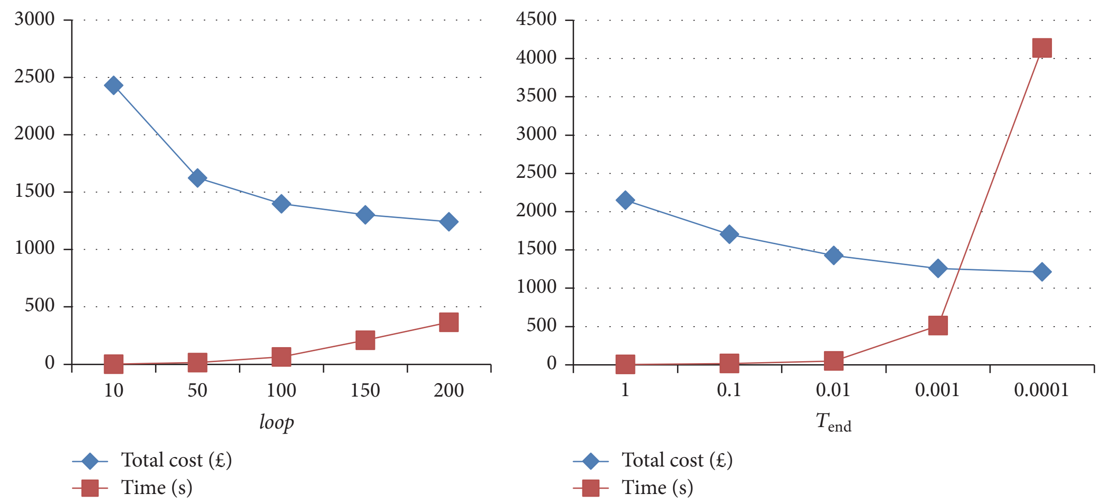


FIGURE 3: Sensitive analysis of *loop* and T_{end}.

the near optimal solution in less time than both CPLEX and ISA. For three instances the ISA managed to get the optimal solution.

In Table 6, results from the ISA outperform the solver CPLEX. As can be seen, the biggest deviation between ISA and CPLEX is 5.25% and the longest running time of ISA is less than 60 seconds, while solver CPLEX costs more than 8 hours. On the other hand, ISA costs less time and obtain better solutions than SA. These results highlight the high quality and time advantage of the proposed SA algorithm for large-scaled problems.

For large-scaled problems, if we set the time limit of CPLEX solver to 12 hours, all the computing process terminated without any solution. Thus, Tables 7 and 8 only perform the ISA and SA results. As can be seen, the longest running time of ISA is less than 100 seconds, which illustrates that the proposed algorithm performs well especially for large-scaled problems.

5.1.2. The Effect of Fixed Cost and Variable Cost. As our interest lies in the robustness of the final approach of the problem, we consider different fixed cost and variable cost values based on average over ten runs for each instance.

Take $n = 200$ as an example. First let $c = 1$, and variable c values are tested over the ranges $f \in \{0.01, 0.1, 10, 100, 1000\}$. The total cost and number of vehicles are listed in Table 9.

From Table 9, we notice that the total cost increase as the fixed cost increases. The number of vehicles increases as the vehicle cost increases, but the increase is not obvious. It only changes slightly when the difference between f and c is large enough. The reason is that our heuristic algorithm always tends to accept the solution with the minimum number of vehicles with a given depots visiting sequence.

5.2. Compared Results between Models with and without Stochastic Speed. In order to observe the effects brought by

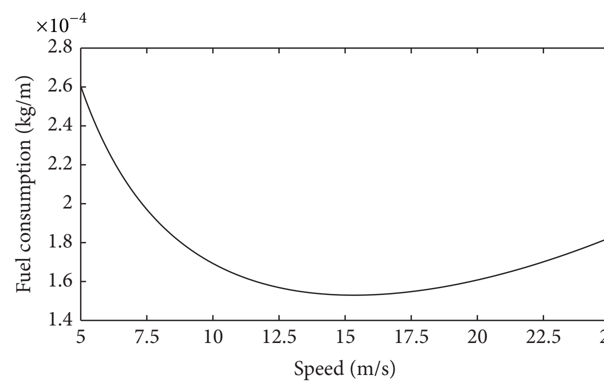


FIGURE 4: Relationship between fuel consumption and travel speed.

the stochastic speeds, we focus on the compared results when $n = 10$. In the stochastic model, we set the means of average speed on all arcs to the given value in the first column of Table 10. In the fixed speed model, the speed of each arc is set to the given value, that is, the corresponding mean in the stochastic model. Table 10 lists the objectives obtained by the ISA algorithm.

As can be seen from Table 10, over the speed interval [5, 25], cost of model with fixed speed declines first; when it reaches 15 m/s, the minimum cost is obtained; then the cost begins to increase until the end of the speed interval.

In fact, fuel consumption takes up a large proportion in the total cost, so the trend of it (Figure 4) is in line with the trend of the total cost. Generally, costs obtained by the stochastic speed model is larger than that with fixed speed over the speed interval (the bold numbers in Table 10 are relatively high cost of the stochastic speed model), but when the speed is close to the end points of the interval, model with fixed speed will obtain smaller cost. The reason of this is that the integral interval will be ineffective once speeds exceed the two interval ends.

Table 6: Comparison results when $n = 50$.

Instance ID	CPLEX		Solutions found by SA in 10 runs			Solutions found by ISA in 10 runs			Dev. from CPLEX	Dev. from SA
	Obj (£)	CPU time (s)	Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)	Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)		
UK50_01	314.75	55276.10	347.99	340.31	87.54	353.42	329.05	55.41	4.54%	−3.31%
UK50_02	317.82	30042.20	363.70	356.72	86.72	356.37	334.5	39.71	5.25%	−6.23%
UK50_03	325.77	35527.60	354.47	349.60	86.05	359.56	338.58	41.7	3.93%	−3.15%
UK50_04	418.24	55624.80	444.30	437.77	87.65	436.86	425.8	28.03	1.81%	−2.73%
UK50_05	352.95	46121.40	371.80	366.43	84.67	386.03	359.88	31.88	1.96%	−1.79%

Table 7: Results of $n = 100$.

Instance ID	Solutions found by SA in 10 runs Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)	Solutions found by ISA in 10 runs Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)	Dev. from SA
UK100_01	786.44	774.13	155.56	765.4	718.92	99.39	−7.13%
UK100_02	774.64	744.66	153.76	727.98	709.02	57.02	−4.79%
UK100_03	690.51	685.67	154.38	682.02	647.96	93.33	−5.50%
UK100_04	663.71	659.64	157.92	683.57	642.06	27.75	−2.67%
UK100_05	629.12	617.56	156.52	612.15	571.8	92.49	−7.41%

Table 8: Results of $n = 200$.

Instance ID	Solutions found by SA in 10 runs Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)	Solutions found by ISA in 10 runs Avg-Obj (£)	Best-Obj (£)	Avg-CPU time (s)	Dev. from SA
UK200_01	1413.97	1387.87	327.66	1388.82	1307.69	40.31	−6.13%
UK200_02	1372.61	1326.84	284.24	1303.79	1249.02	57.18	−6.23%
UK200_03	1387.10	1363.50	371.36	1336.83	1264.7	48.27	−7.81%
UK200_04	1316.76	1244.31	292.16	1295.71	1242.86	36.85	−0.12%
UK200_05	1524.22	1503.30	293.40	1497.39	1381.58	42.39	−8.81%

Table 9: Test results of different fixed vehicle cost.

	Values of f				
	1000	100	10	0.1	0.01
N	28	28	29	29	29
Total cost (£)	28975.39	4459.221	1474.35	1154.432	968.6605

Table 10: Compared results of $n = 10$.

Given speed value (m/s)	Objective value (£) Stochastic speed model	Fixed speed model
5	92.42	201.39
7	158.4	160.77
9	**141.34**	140.16
11	**129.78**	129.1
13	**124.03**	123.57
15	**124.46**	121.7
17	**122.74**	122.48
19	**125.56**	125.35
21	**130.13**	129.95
23	132.95	136.04
25	71.21	143.47

6. Conclusion

A new variant of the VRP with fuel consumption is presented, where the travel speed is considered to be stochastic. To solve this problem, an improved simulated annealing algorithm is proposed and presented in this paper. The proposed ISA is effective for solving the fuel VRP especially for large-scale problems. Experimental results showed that when the value of the expected speed is not close to the end points of speed limit, total travel cost of the stochastic model is always larger than that of the fixed speed model. In order to make reliable logistics routing decisions, managers should take the stochasticity of speed into consideration.

Several extensions are possible for further research. One worth mentioning extension is when average travel speed follows other forms of distribution, and another one would be the time window version or traffic congestion version of this problem.

Appendix

A. The Pseudocode of the ISA Algorithm

See Algorithm 1.

B. Calculation of the Cost Function C(VRPS)

See Algorithm 2.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research of Ren-Qian Zhang is supported by NSFC under 71571006 and 71271010, the research of Guozhu Jia is supported by NSFC under 71372007, and the research of Yanling Feng is supported by China Scholarship Council under 201606020056.

Main: Improved Simulated Annealing algorithm with memory for VRPFSV
Input:
(1) n customers with demand D and distance matrix L
(2) vehicle capacity Q, single vehicle cost f, speed distribution parameters u and σ^2 of each arc
(3) simulated annealing parameters: T_{end}, *loop*
Output: VRPS route with lowest fuel and vehicle cost
Begin
(1) Initiating: read data file and generate random route S_0 based on the above-mentioned string character. String length: $N = n + 2$
(2) $S = S_0$ //accepted as incumbent solution
(3) Heating: randomly change the incumbent solution S to its neighbor for 1000 times. The maximum cost deviation between two adjacent solutions is selected as the initial temperature, $T_0 = \max(\Delta C)$
(4) $T = T_0$
(5) **While** $T > T_{\text{end}}$ **do begin**
(6) **For** $k = 1$ **to** *loop* **do begin**
(7) Changing S to S^{new} //changing S to S^{new} by one exchange rule randomly
(8) Convert S^{new} to VRPS^{new} //Convert S^{new} to feasible solution
(9) Calculate total cost $C(\text{VRPS}^{\text{new}})$
(10) $\Delta C = C(\text{VRPS}^{\text{new}}) - C(\text{VRPS})$ //calculating cost deviation
(11) Accept or reject VRPS^{new} according to the probability of Metropolis algorithm
(12) Update VRPS with VRPS^{new} //Update VRPS
(13) **If** $\text{VRPS}^{\text{new}} < \text{VRPS}^{\text{best}}$
(14) Update $\text{VRPS}^{\text{best}}$ with VRPS^{new} //Update $\text{VRPS}^{\text{best}}$
(15) **end if**
(16) **End for**
(17) **If** $\Delta C < 0.001$
(18) **Terminate the ISA heuristic**
(19) **End if**
(20) Count the number of accepted solutions A, $\tau = A/(A + loop)$ //calculating τ
(21) $T = T\tau$
(22) **End while**
(23) **Output** $\text{VRPS}^{\text{best}}$
End

Algorithm 1

Input:
(1) string vector VRPS and fuel consumption function F about travel speed v.
(2) single vehicle cost: f, vehicle capacity: Q, demand matrix D, speed matrix V, distance matrix L.
Output: total cost of all vehicles, including fixed cost and expected fuel consumption
Begin
(1) $Totalcost = 0$, $ExpectF = 0$
(2) n = find the last nonzero index of VRPS; //find the real end of the string
(3) **For** $i = n$ **to** 1 **do begin** //start from the last customer of the last vehicle
(4) $L = L(\text{VRPS}(i), \text{VRPS}(i+1))$; //distance of arc($\text{VRPS}(i), \text{VRPS}(i+1)$)
(5) $u = V(\text{VRPS}(i), \text{VRPS}(i+1))$; //the mean of the average speed of this arc
(6) **If** $\text{VRPS}(i+1) == 0$ //find a used vehicle
(7) $W = 0$; //reset the load W to zero
(8) $m = m + 1$; //update the number of vehicles
(9) **Else**
(10) $W = W + D(\text{VRPS}(i+1))$; //update the vehicle load
(11) **End If**
(12) $ExpectF = ExpectF + quadl(F(v, W, L, u), v^{\min}, v^{\max})$; //update the expectation of fuel consumption
(13) **End For**
(14) $TotalC = mf + ExpectF$;
(15) **Return** $TotalC$
End

Algorithm 2

References

[1] J. K. Lenstra and A. H. G. R. Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.

[2] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 2, pp. 80–91, 1959.

[3] P. Toth and D. Vigo, "Models, relaxations and exact approaches for the capacitated vehicle routing problem," *Discrete Applied Mathematics*, vol. 123, no. 1–3, pp. 487–512, 2002.

[4] S. Kassem and M. Chen, "Solving reverse logistics vehicle routing problems with time windows," *International Journal of Advanced Manufacturing Technology*, vol. 68, no. 1-4, pp. 57–68, 2013.

[5] R. Spliet and G. Desaulniers, "The discrete time window assignment vehicle routing problem," *European Journal of Operational Research*, vol. 244, no. 2, pp. 379–391, 2015.

[6] R. Baldacci, A. Mingozzi, and R. Roberti, "Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints," *European Journal of Operational Research*, vol. 218, no. 1, pp. 1–6, 2012.

[7] R. Tavakkoli-Moghaddam, A. R. Saremi, and M. S. Ziaee, "A memetic algorithm for a vehicle routing problem with backhauls," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 1049–1060, 2006.

[8] A. Hoff, I. Gribkovskaia, G. Laporte, and A. Løkketangen, "Lasso solution strategies for the vehicle routing problem with pickups and deliveries," *European Journal of Operational Research*, vol. 192, no. 3, pp. 755–766, 2009.

[9] S. Allahyari, M. Salari, and D. Vigo, "A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem," *European Journal of Operational Research*, vol. 242, no. 3, pp. 756–768, 2015.

[10] U. Ritzinger, J. Puchinger, and R. F. Hartl, "A survey on dynamic and stochastic vehicle routing problems," *International Journal of Production Research*, vol. 54, no. 1, pp. 215–231, 2016.

[11] Y. Z. Mehrjerdi, "A multiple objective stochastic approach to vehicle routing problem," *The International Journal of Advanced Manufacturing Technology*, vol. 74, no. 5–8, pp. 1149–1158, 2014.

[12] D. Taş, M. Gendreau, N. Dellaert, T. van Woensel, and A. G. de Kok, "Vehicle routing with soft time windows and stochastic travel times: a column generation and branch-and-price solution approach," *European Journal of Operational Research*, vol. 236, no. 3, pp. 789–799, 2014.

[13] D. Taş, N. Dellaert, T. van Woensel, and T. de Kok, "The time-dependent vehicle routing problem with soft time windows and stochastic travel times," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 66–83, 2014.

[14] J. F. Ehmke, A. M. Campbell, and T. L. Urban, "Ensuring service levels in routing problems with time windows and stochastic travel times," *European Journal of Operational Research*, vol. 240, no. 2, pp. 539–550, 2015.

[15] G. Laporte, F. Louveaux, and H. Mercure, "Vehicle routing problem with stochastic travel times," *Transportation Science*, vol. 26, no. 3, pp. 161–170, 1992.

[16] Y. Marinakis, G.-R. Iordanidou, and M. Marinaki, "Particle Swarm Optimization for the vehicle routing problem with stochastic demands," *Applied Soft Computing Journal*, vol. 13, no. 4, pp. 1693–1704, 2013.

[17] M. Dror, G. Laporte, and P. Trudeau, "Vehicle routing with stochastic demands: properties and solution frameworks," *Operations Research Society of America. Transportation Science Section. Transportation Science*, vol. 23, no. 3, pp. 166–176, 1989.

[18] D. J. Bertsimas, "A vehicle routing problem with stochastic demand," *Operations Research*, vol. 40, no. 3, pp. 574–585, 1992.

[19] J. Zhang, W. H. Lam, and B. Y. Chen, "On-time delivery probabilistic models for the vehicle routing problem with stochastic demands and time windows," *European Journal of Operational Research*, vol. 249, no. 1, pp. 144–154, 2016.

[20] Y. Xiao and A. Konak, "A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness," *Applied Soft Computing Journal*, vol. 34, pp. 372–388, 2015.

[21] R. Kondekar, A. Gupta, G. Saluja, R. Maru, A. Rokde, and P. Deshpande, "A MapReduce based hybrid genetic algorithm using island approach for solving time dependent vehicle routing problem," in *Proceedings of the International Conference on Computer and Information Science (ICCIS '12)*, pp. 263–269, IEEE, Kuala Lumpur, Malaysia, June 2012.

[22] J. Zhang, X. Zhao, and X. He, "A minimum resource neural network framework for solving multiconstraint shortest path problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 8, pp. 1566–1582, 2014.

[23] A. Meyer-Bäse, G. Botella, and L. Rybarska-Rusinek, "Stochastic stability analysis of competitive neural networks with different time-scales," *Neurocomputing*, vol. 118, pp. 115–118, 2013.

[24] E. Demir, T. Bektaş, and G. Laporte, "An adaptive large neighborhood search heuristic for the pollution-routing problem," *European Journal of Operational Research*, vol. 223, no. 2, pp. 346–359, 2012.

[25] T. Garaix, C. Artigues, D. Feillet, and D. Josselin, "Optimization of occupancy rate in dial-a-ride problems via linear fractional column generation," *Computers & Operations Research*, vol. 38, no. 10, pp. 1435–1442, 2011.

[26] Z. Cao, H. Guo, J. Zhang, D. Niyato, and U. Fastenrath, "Improving the efficiency of stochastic vehicle routing: a partial lagrange multiplier method," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3993–4005, 2015.

[27] A. Ishigaki, "An application to stochastic vehicle-routing problem in a waste collection," in *Proceedings of the 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI '16)*, pp. 1151–1156, Kumamoto, Japan, July 2016.

[28] L. Schulze and A. Wüllner, "The approach of automated guided vehicle systems," in *Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI '06)*, pp. 522–527, IEEE, Shanghai, China, June 2006.

[29] G. Botella, A. García, M. Rodríguez-Álvarez, E. Ros, U. Meyer-Baese, and M. C. Molina, "Robust bioinspired architecture for optical-flow computation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 4, pp. 616–629, 2010.

[30] S. Mafrica, A. Servel, and F. Ruffier, "Towards an automatic parking system using bio-inspired 1-D optical flow sensors," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES '15)*, pp. 96–103, Yokohama, Japan, November 2015.

[31] Y. Kuo, "Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem," *Computers & Industrial Engineering*, vol. 59, no. 1, pp. 157–165, 2010.

[32] J. Qian and R. Eglese, "Finding least fuel emission paths in a network with time-varying speeds," *Networks*, vol. 63, no. 1, pp. 96–106, 2014.

[33] M. Barth and K. Boriboonsomsin, "Energy and emissions impacts of a freeway-based dynamic eco-driving system," *Transportation Research Part D: Transport and Environment*, vol. 14, no. 6, pp. 400–410, 2009.

[34] T. Bektaş and G. Laporte, "The pollution-routing problem," *Transportation Research Part B: Methodological*, vol. 45, no. 8, pp. 1232–1250, 2011.

[35] Y. Xiao, Q. Zhao, I. Kaku, and Y. Xu, "Development of a fuel consumption optimization model for the capacitated vehicle routing problem," *Computers and Operations Research*, vol. 39, no. 7, pp. 1419–1431, 2012.

[36] M. Figliozzi, "Vehicle routing problem for emissions minimization," *Transportation Research Record*, no. 2197, pp. 1–7, 2010.

[37] F. Errico, G. Desaulniers, M. Gendreau, W. Rei, and L.-M. Rousseau, "A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times," *European Journal of Operational Research*, vol. 249, no. 1, pp. 55–66, 2016.

[38] M. Marinaki and Y. Marinakis, "A glowworm swarm optimization algorithm for the vehicle routing problem with stochastic demands," *Expert Systems with Applications*, vol. 46, pp. 145–163, 2016.

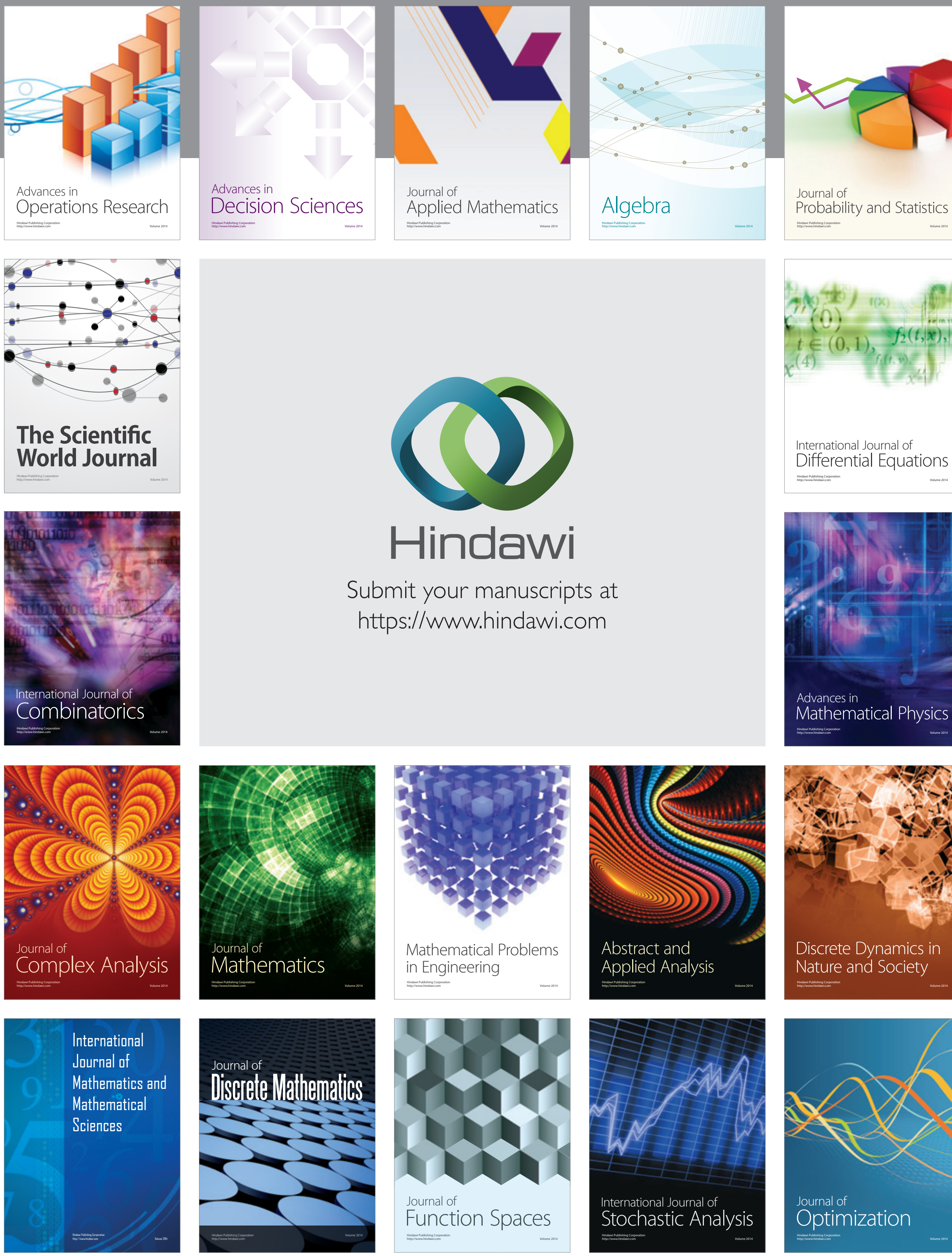
[39] B. Sarasola, K. F. Doerner, V. Schmid, and E. Alba, "Variable neighborhood search for the stochastic and dynamic vehicle routing problem," *Annals of Operations Research*, vol. 236, no. 2, pp. 425–461, 2016.

[40] O. Jabali, T. van Woensel, and A. G. de Kok, "Analysis of travel times and CO_2 emissions in time-dependent vehicle routing," *Production and Operations Management*, vol. 21, no. 6, pp. 1060–1074, 2012.

[41] A. Franceschetti, D. Honhon, T. Van Woensel, T. Bektaş, and G. Laporte, "The time-dependent pollution-routing problem," *Transportation Research Part B: Methodological*, vol. 56, pp. 265–293, 2013.

[42] Y. Xiao and K. Abdullah, "A genetic algorithm with exact dynamic programming for the green vehicle routing & scheduling problem," *Journal of Cleaner Production*, 2016.

[43] V. Granville, M. Krivánek, and J.-P. Rasson, "Simulated annealing: a proof of convergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 652–656, 1994.


Advances in
Operations Research
Advances in
Decision Sciences
Journal of
Applied Mathematics
Algebra
Journal of
Probability and Statistics
The Scientific
World Journal
International Journal of
Differential Equations
Hindawi
Submit your manuscripts at
https://www.hindawi.com
International Journal of
Combinatorics
Advances in
Mathematical Physics
Journal of
Complex Analysis
Journal of
Mathematics
Mathematical Problems
in Engineering
Abstract and
Applied Analysis
Discrete Dynamics in
Nature and Society
International
Journal of
Mathematics and
Mathematical
Sciences
Journal of
Discrete Mathematics
Journal of
Function Spaces
International Journal of
Stochastic Analysis
Journal of
Optimization