THE CAPACITATED VEHICLE ROUTING PROBLEM WITH DRONES CONSIDERING STOCHASTIC DEMANDS AND RESTRICTED RETURN TRIP

Tengkuo Zhu
The University of Texas at Austin, Texas.
Graduate Research Assistant
Civil, Architectural, and Environmental Engineering
The University of Texas at Austin
Austin, Texas 78712
Email: zhutengkuo@utexas.edu

Stephen D. Boyles
The University of Texas at Austin, Texas
Associate Professor
Civil, Architectural, and Environmental Engineering
The University of Texas at Austin
Austin, Texas 78712
Email: sboyles@austin.utexas.edu

Word Count: 6818 words + 2 table(s) × 250 = 7318 words

Submission Date: August 3, 2020

ABSTRACT

This paper investigate a new last-mile routing problem called capacitated vehicle routing problem with drones with stochastic demand (CVRPDSD). CVRPDSD considers the randomness of the customers' demand and aims to find a set of coordinated routes of both the truck and the drone while minimizing the total route cost, which consists of the a priori cost and the expected recourse cost. To model the recourse action in CVRPDSD, a new recourse strategy is proposed, which is modified from the classical recourse policy in stochastic vehicle routing problem. The expected recourse cost of each individual route under the new recourse strategy is calculated as a closed form mathematical equation. A large neighborhood search method is presented to solve the problem. Numerical experiments on public available VRPLIB datasets illustrate that on average the addition of UAVs can reduce 21.5% of the total overall cost. Additional sensitivity analyses further indicate that UAV capacity has a significant effect on the total cost.

Keywords: VRP with drones, stochastic demand, large neighborhood search

INTRODUCTION

The Covid-19 pandemic is exacting a terrible human toll and menacing the economy in different industry sectors. However, the e-commerce industry continues to grow during the pandemic, perhaps even accelerating as in-person shopping is restricted. According to (*20*) and (*4*), in 2020 there has been a 129% year-over-year growth in U.S. & Canadian e-commerce orders as of April 21 and an 146% growth in all online retail orders. The growth of these numbers indicates an increasing need for efficient goods transportation methods in logistics and reverse logistics that requires no human interaction.

The use of unmanned aerial vehicles (UAVs), or drones (throughout this paper these terms are used interchangeably) for this purpose has received increasing attention, and may represent the future of the logistics. This newly emerged delivery method uses pilotless aircraft to deliver packages autonomously and can be remotely controlled through a ground controlled station. A number of large organizations such as Amazon and Google have announced their initiatives in drone delivery. According to (*10*), the global drone market is currently worth 14 billion dollars and this number is estimated to grow to 43 billion dollars by 2024.

Compared to traditional internal combustion engine-based truck delivery, drone delivery has the potential to significantly reduce the delivery cost and time. In general, without the costly labor, drone delivery can significantly lower the laboring cost. It is also more environmentally friendly than petroleum-fuel powered vehicles. However, despite the recently progress and investment on drone delivery, several challenges still exist such as the limited flight range, payload capacity and safety issue when flying across downtown districts and/or other densely populated urban areas. The truck-drone delivery method, which is proposed in (*17*), provides one solution by allowing delivery trucks and drones cooperates and accomplish delivery task independently.

During the vehicle's operation, congestion and variability in demand and travel times affects these industries on three major service dimensions: (i) travel time; (ii) reliability; and (iii) cost. For example, traffic condition may vary and cause uncertainty in the travel times. In many pick-up services, the demands of the customers are revealed only upon arrival of the vehicles at their locations. Taken these factors into the input data of routing problem leads to the so-called stochastic vehicle routing problem (SVRP). To model the stochastic VRP, two approaches are commonly used, chance constrained programming (CCP) and stochastic programming with recourse (SPR). The CCP versions of the SVRP usually minimizes the sum of planned route costs, while ensuring that the probability of route failure does not exceed a given threshold.

In this paper, we consider a PICK-UP ONLY capacitated vehicle routing problem with drones with stochastic demand (CVRPDSD). With a known probabilistic description of the demand, we aim to design a set of coordinated routes of multiple trucks and drones while minimizing the total route cost, which consists of the *a priori* cost and the expected recourse cost. Besides, although most of the drone routing problem only involves one single truck and one UAV, in this research, multiple trucks, each of which is equipped with one UAV, can be used to fulfill the customers demand. Based on the author's knowledge, this is the first research to address the stochastic routing problem with drones. The recourse strategy is also proposed in this research. Based on this strategy, this research derives the mathematical formulation of the expected recourse cost and adopts a large neighborhood search (LNS) method to obtain the optimal *a priori* routes that minimize the total route cost.

The main contributions of this paper are:

- The CVRPDSD is introduced and investigated. It is a *pick-up only* problem which consid-

ers uncertainty in customer demand, and the operation of delivery method that involves multiple trucks and UAVs. The problem aims to find a set of routes that has the minimal overall route cost which includes the *a priori* planning cost and expected recourse cost.
- The truck-drone recourse policy during the operation process is proposed, along with necessary modifications to the classical VRP recourse strategy. A closed-form mathematical equation is derived which can be used to calculate the expected recourse cost of a truck-drone coordinated route.
- A hybrid large neighborhood search heuristic, which also integrates constraint programming (CP) modelling, is proposed to solve CVRPDSD.
- The numerical analysis based on the tests on public available VRPLIB data sets indicates that UAV capacity has a significant effect on the overall route cost.

LITERATURE REVIEW

Due to the recent development on related techniques and the urgent desire of a new delivery method that is both efficient and environmental friendly, research on the application of UAV in logistics is receiving heavy attention in the operations research community. Some of research focuses on drone-only delivery such as (*30*). However, in this section we focus on research into the delivery methods combining a truck and a UAV, a so-called truck-UAV tandem delivery method. This method not only represents the transition from the traditional truck-only delivery to UAV delivery, but in itself is an highly efficient way to accomplish delivery tasks. This method combines the strength of truck delivery (high payload, unlimited travel range) and UAV delivery (low labor cost, fast and efficient) while addressing their corresponding shortcomings (limited flight range of a UAV, volatile travel times of trucks due to congestion).

(*17*) introduced a truck-UAV tandem delivery method, the flying sidekick traveling salesman problem (FSTSP). It assumes that the truck serves as a UAV hub that can launch and retrieve the UAV at customer nodes while both vehicles deliver parcels independently. A mixed integer linear programming (MILP) formulation and a truck-first-drone-second heuristic are also proposed. Since then, different exact and heuristic algorithms are presented to solve this problem. (*34*) proposed an iterative method to solve FSTSP where the the first stage divides all the nodes into two exclusively mutated node set and the second stage optimizes the route by serving nodes in one set by truck and serving nodes in another set by UAV, while (*7*) and (*8*) present a neighborhood search method on FSTSP. There are also several studies on a slightly different problem, the traveling salesman problem with drones (TSPD) where the UAV could be retrieved at its launched node, which is prohibited in FSTSP. Examples include (*3*) and (*19*).

A natural extension of FSTSP or TSPD involves the coordination of multiple trucks and drones. This is called vehicle routing problem with drones (VRPD). (*5*) studied the routing problem with a truck and several drones and a cluster-based approach is used to solve this problem. (*32*) proposed a more complicated problem which involves cooperation of multiple trucks and drones. In this problem, a drone can travel with the first truck to a drone hub and then travel with a different truck to continue its delivery task. This interchangeable property of drones makes the problem more challenging to solve. An arc-based integer programming model is proposed and a branch-and-price algorithm is used to solve the problem. Due to the complication of this problem, most research on this topic uses meta-heuristics, primarily using neighborhood search methods. (*27*) adopts a variable neighborhood search (VNS) while (*26*) used a hybrid VNS-tabu search to solve a variant of VRPD called vehicle routing problem with drones and en route operations (VRPDERO).

In VRPDERO drones can be launched and retrieved en route on some discrete points on arc. Besides, (*23*) used a large neighborhood method to solve VRPD.

(*31*) is the first paper that addresses the uncertainty of customers' demands. It considered a multi-depot variant of the CVRP with Poisson distributed demands. A modified algorithm originally proposed by Clarke and Wright (*6*) was used to solve the problem. (*2*) derived a closed form expressions to compute the expected length of an *a priori* route. (*14*) presented an integer L-shaped method for stochastic programs with recourse. The same L-shaped algorithm is also used in (*12*) to solve the single vehicle CVRPSD. A new set of optimality cuts is derived to bound the recourse cost. More recently, (*21*) used a hybrid method that combines local branching and Monte Carlo sampling to solve single-vehicle CVRPSD. Several researchers have focused on finding alternative recourse policy, such as an optimal restocking policy which enables preventive return trips to the depot ((*33*), (*24*)), or rule based policy((*25*)). Most of the research on SVRP with recourse focus on the single-vehicle case due to the inherent complexity of the problem. The CCP modeling approach is common; see, for instance, (*29*) and more recently (*9*) and (*18*).

In this paper we focus on multiple vehicle routing, considering stochastic demand with restricted return trips. Note that in CVRPDSD a failure does not necessarily mean a return trip; when a failure happens at a UAV node the truck need not return to depot. We limit the maximum number of return trips to one, while having no limitation on the number of failures along a route. We believe this assumption is reasonable, as it makes little managerial sense to plan distribution routes that need a large number of return trips, and the total expected demand of a single route should not very much exceed the capacity the vehicle. These two properties imply that for an *a priori* planned route, the number of potential return trips is small. Similar assumptions are common in the literature, as in (*11*) or (*16*).

To our knowledge, no published research has considered or modeled randomness in customers' demand in vehicle routing problem with drones. This research fills this gap by first propose a modified recourse strategy for VRPD, derives a mathematical formulation of the expected recourse cost and then solves the problem within a large neighborhood search framework that integrates constraint programming modelling.

PROBLEM DESCRIPTION AND MATHEMATICAL FORMULATION

In this section, the problem description and basic assumptions of CVRPDSD are presented first. Then the recourse policy in the situation of failure is introduced. Finally, this section presents the calculation process of the expected recourse cost and the total route cost

Problem description

The CVRPDSD is defined on an undirected, complete graph $G = (V,E)$, with a vertex set V consisting of a depot site $I = \{0\}$ and a set of customers $J = v_1, v_2, ..., v_c$. The vertex set is thus $V = I \cup J$. The edge set $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ contains the edges connecting vertices of V. Each edge (v_i, v_j) is associated with two non-negative travel time τ_{ij} and τ'_{ij}, which corresponds to the travel time needed for the truck and UAV to travel from node i to node j, respectively.

In this paper, the CVRPDSD is defined to be a pick-up only problem where the logistics company aims to find a set of coordinated routes such that each customer is served exactly once by either a truck or a drone while minimizing the total expected route cost. The customer's demand is stochastic with a given probability distribution, while the actual demand is only revealed when the vehicle visit the customer's location. Thus, a route failure might happens during the delivery

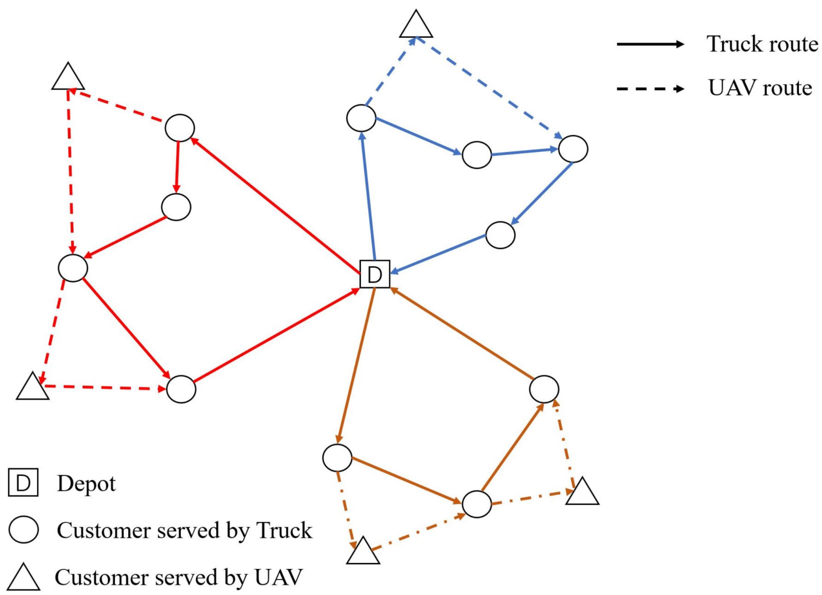

FIGURE 1 A simple representation of the a CVRPDSD solution

operation. A "*truck route failure*" happens when the total actual demand on a route exceeds the vehicle's capacity and a "*drone route failure*" happens when the total actual demand of a drone node exceeds the drone's capacity. A strategy is required for updating the routes in case of such events. The actual action resulting from this strategy is called a *recourse* action. When truck route failure happens, the truck needs to return to the depot to replenish its capacity. In other literature this policy is also called a "*restocking policy*". When drone route failure happens the failed drone node needs to be served by the truck instead. In this research, the truck and the drone will adjust their route based on the given recourse strategy and the detail of this strategy will be introduced later in this section. Some of the additional assumptions of CVRPDSD are:

1. The demands of customers are identical and independent random variables with known discrete or continuous probability distributions. The probability that an individual's demand exceed the vehicle's capacity is zero.
2. The demands of customer are only revealed when the truck or drone arrives.
3. Both the truck and the UAV are capacitated and the UAV has a flight range limit.

The solution of CVRPDSD consists of a set of coordinated routes $s = r_1, r_2, ..., r_m$ where m is the number of available vehicles. For each vehicles i's route $r_i = [ri^T, r_i^D]$, r_i^T specifies the truck i's route while r_i^D specifies all the drone sorties in the truck i's route, where a sortie includes a launch node, a drone node and a retrieve node. For example, assume $r^T = [0, 2, 3, 8, 1, 4, 5, 0]$ and $r^D = [[2, 6, 3], [1, 7, 5]]$. r^T indicates that the truck starts the trip at the depot, visit customer node 2,3,8,1,4,5 consecutively and returns to the depot. r^D indicates there are two UAV sorties in the solution. For the first sortie, the drone is launched at node 2. Then it serves customer node 6 before it is retrieved at node 3. The customer nodes that are planned to be served by the truck and drone are called *truck nodes* and *drone nodes*, respectively. An example of a solution and coordinated routes is shown in Figure 1.

Given a coordinated route r, we can derive several different node sets:

1. launch node set L_r: it includes all the customer nodes that serves as a launch node in one of the drone sorties
2. drone node set D_r: it includes all the customer nodes that are served by the drone in one of the drone sorties
3. retrieve node set R_r: it includes all the customer nodes that serves as a retrieve node in one of the drone sorties
4. intermediate node set I_r: it includes all the customer nodes that served by the truck when the drone is not on the truck.
5. normal truck node set T_r: It includes all the customer nodes that are served by the truck and not in the set $L_r \cup R_r$

For the case where $r^T = [0,2,3,8,1,4,5,0]$ and $r^D = [[2,6,3],[1,7,5]]$, the launch node set $L_r = [2,1]$, drone node set $D_r = [6,7]$, retrieve node set $R_r = [3,5]$, intermediate node set $I_r = [4]$ and normal truck node set $T_r = [8]$

Recourse strategy

In CVRPDSD, both truck route failure and drone route failure might happen. Thus, in this study we adopt a modified version of the classical recourse strategy. The general principles of the modified recourse strategy are:

1. No re-optimization after the failure and the vehicle resumes its deliveries as planned.
2. The truck cannot returns to the depot without the drone.
3. When the UAV is retrieved to the truck and find out that the remaining capacity of the truck is less than the volume of the item that is being carried on the UAV, the truck needs to returns to the depot before serving other customers.
4. At most one return trip can be made on any route to the depot. This means that the solution must be such that the probability of the total demand less than two times the truck capacity is close to 1. In this paper, a route is said to be "feasible" if the probability of the total demand greater than two times the truck capacity less than a pre-defined value α. For the tests conducted in this paper, α is set to be 0.01.
5. No partial pick-up service is allowed.
6. Multiple partial recourse trips could be made and failed drone nodes have a higher priority than unserved truck nodes. A partial recourse is defined when a drone node cannot be served because its demand exceed the drone's capacity and this node will be served first by the truck after the drone is retrieved before truck goes on to its origin planned route.

Based on these assumptions, different recourse actions are deployed when route failure happens at customer nodes of different types.

1. If a drone node fails, it would be served by the truck after the drone is retrieved prior to serving other customer nodes along truck's planned route.
2. If a launch node fails, the truck and the drone will return to the depot and go back to this launch node to launch the UAV and continue on its planned route.
3. If a retrieve node fails with corresponding drone node served, the truck and the drone will return to the depot and travel to this retrieve node or go to next planned node. If a retrieve node fails with corresponding drone node NOT served, the truck and the drone will return to the depot, serve the drone node and go to next planned node.
4. If a intermediate node fails, the truck will retrieve the drone at original planned retrieve

node, returns to the depot, returns to the intermediate node and proceeds with planned route. If the drone node is unserved, the truck will serve it after it serves the retrieve node.

5. If a normal truck node fails, the classical recourse strategy is deployed.
6. When a node which is both a launch node and a retrieve node fails, it should be treated as a retrieve node.

Based on the given policy, in order to illustrate the potential truck route due to the demand uncertainty, a modified truck route r' is presented by inserting all the drone nodes into the original truck's route r^T after the corresponding retrieve node. We do so to account for the possibility that if the drone node cannot be served by the drone it would then be served by the truck after the retrieve node in r_T. For the given example when $r = [r^T, r^D]$ where $r^T = [0,2,3,8,1,4,5,0]$ and $r^D = [[2,6,3],[1,7,5]]$, the corresponding $r' = [0,2,3,6,8,1,4,5,7,0]$. The modified truck route r' is mainly used for calculating the cumulative demand up to a certain node.

Computation of the total expected cost

Consider the kth planned route r_k as a sequence of vertices: $r_k = (v_0, v_1^k, v_2^k, ..., v_{n_k}^k, v_0)$. Given a original planned solution $s = (r_1, ..., r_m)$, the objective function $F(s)$ is the sum of two terms: the deterministic cost and the expected recourse cost during the operation of all the planned routes in s. We denote these two terms as $\phi(s)$ and $E[\psi(s)]$, respectively. As a result,

$$F(s) = \phi(s) + E[\psi(s)] = \sum_{i=0}^{m} (\phi(r_i) + E[\psi(r_i)]) \tag{1}$$

In (1), the computation of the deterministic route cost is straightforward. The calculation of the expected recourse cost of each individual route will be introduced later in this section.

Computation of the probability of failure

Let ξ_i^k denote the demand of ith truck node v_i^k on route $r^{k'}$ (Note here $r^{k'}$ is the modified representation of an individual route) and let X_i^k denotes the cumulative demand up to and including ith node v_i^k. The random variable that represents the vertex demand could be either discrete or continuous. Denote $P_D(t) = P\{X = t\}$ as the probability function of a generic discrete random variable D. Denote $f_D(t)$ as the probability density function of a generic continuous random variable D. In this study, the value of $P_D(t)$ and $f_D(t)$ are known for each vertex in the problem.

Proposition 1. *((15)) The value of $P_{X_i^k}(t)$ and $f_{X_i^k}(t)$ satisfy the following recursion:*

$$P_{X_i^k}(t) = \sum_{l=0}^{t} P_{X_{i-1}^k}(t-l) P_{\xi_i^k}(l) \tag{2}$$

with the boundary condition $P_{X_0^k}(t) = P_{\xi_0^k}(t)$,

$$f_{X_i^k}(t) = \int_{l=0}^{t} f_{X_{i-1}^k}(t-l) f_{\xi_i^k}(l) dl \tag{3}$$

with the boundary condition $f_{X_0^k}(t) = f_{\xi_0^k}(t)$,

Proof. Obviously at the boundary $X_0^k = \xi_0^k$ holds. As X_i^k is the cumulative demand at vertex v_i^k, $X_i^k = X_{i-1}^k + \xi_i^k$. As X_{i-1}^k and ξ_i^k are independent and nonnegative, the probability of X_i^k is the sum of the probability of all the discrete cases or the integral of all the continuous case. □

Computation of the expected cost of recourse

In this subsection, given a modified route r', following additional notations are used.

Q : Capacity of the truck

D : Capacity of the drone

$d(i, j)$: Distance between node i and node j

$\pi(i, j)$: Total travelled distance in truck's a priori route from node i to node j

$next(i)$: Node that is next to node i in r'

$pre(i)$: Node that is previous to node i in r'

$dr(i)$: The corresponding drone node if i serves as a launch node, intermediate node or retrieve node

Now we are ready to calculate the expected recourse cost of each vehicle's route by analyzing customer node of different types. Note that in this process we use the modified route representation r' instead of r^T. Some of the notations are shown at the top of this section. Besides, denote $P\{X_i < Q\}$ as the probability of that the cumulative demand in r' including all the previous nodes up to customer i does not exceed Q. If there exists a customer node i such that $i \in R_r$ or $r \in I_r$ (i is a retrieve node or intermediate node), let $dr(i)$ denotes its corresponding drone node. Note that $dr(i)$ is the next node in r' if i is a retrieve node. To better explain the calculation process of different cases, we use the above-mentioned route as an example. For now, assume that we have a vehicle's route $r = [[0,1,2,3,4,5,6,0],[[0,7,2],[4,8,6]]]$, which indicates that customer 1-6 is served by the truck while the UAV serves customer 7 and 8. The modified route representation is $r' = [0,1,2,7,3,4,5,6,8,0]$.

Let us start with several simple cases:

Case 1: The customer node i is planned to be served by the truck and i is a launch node. As in our example r, $i = 0$ or 4. In this case, recourse action only happens when $X_{pre(i)} \leq Q, X_i \geq Q$ and the truck would need to go back to the depot first and return to node i so that it can launch the UAV. The recourse cost is simply $C_1 = 2d(0,i)$. As a result, the expected recourse cost at node i is

$$E[C(i)] = P\{X_{pre(i)} \leq Q, X_i \geq Q\}C_1 \tag{4}$$

Case 2: The customer node i is planned to be served by the truck and i is normal truck node. As in our example r, $i = 3$. In this case, recourse action only happens when $X_{pre(i)} < Q, X_i \geq Q$. If $X_i = Q$, the truck will need to return to the depot after serving customer i and go to node $next(i)$ after replenishment. The recourse cost is $C_2 = d(i,0) + d(0,next(i)) - d(i,next(i))$. If $X_i > Q$, the truck will need to return to the depot and come back to serve customer i. The recourse cost is $C_3 = 2d(0,i)$. So the expected recourse cost at node i is

$$E[C(i)] = P\{X_{pre(i)} \leq Q\}(P\{X_i = Q\}C_2 + P\{X_i > Q\}C_3) \tag{5}$$

Case 3: The customer node i is planned to be served by the truck and i is an intermediate node of an UAV sortie. As in our example r, $i = 5$. Let $dr(i)$ denotes the corresponding drone node of the sortie and $ret(i)$ denotes the corresponding retrieve node. In this case, recourse action happens when $X_{pre(i)} < Q, X_i \geq Q$. However, different from the first two cases, the truck cannot returns to the depot directly because it cannot returns to the depot without the drone. So, it needs to retrieve the drone at the retrieve node $ret(i)$ first, returns to the depot and come back to serve i or go to $next(i)$. Similarly, if $X_{pre(i)} < Q, X_i = Q$, the recourse cost is $C_4 = \pi(i,ret(i)) + d(ret(i),0) + d(0,next(i)) - d(i,next(i)$. If $X_{pre(i)} < Q, X_i > Q$, the recourse

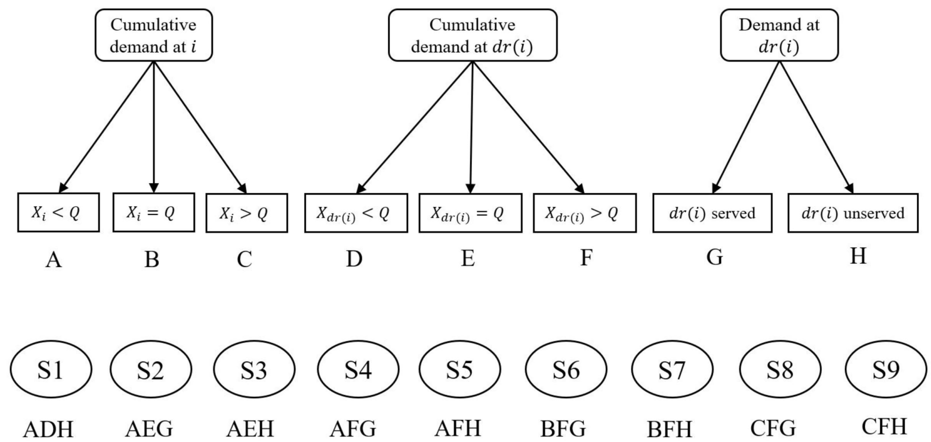

FIGURE 2 Illustration of different scenarios in case 4

cost is $C_5 = \pi(i, ret(i)) + d(ret(i), 0) + d(0, i)$. So the expected recourse cost at node i is

$$E[C(i)] = P\{X_{pre(i)} < Q\}(P\{X_i = Q\}C_4 + P\{X_i > Q\}C_5) \tag{6}$$

Now, for the remaining of this subsection, we will consider a more complicate case when the recourse action decision is made at a retrieve node. The underlying reason for this complication is because we receive more demand information at a retrieve node than nodes of other types. Remember that in the truck-UAV delivery method the truck and UAV operates independently and we only know the exact demand of a customer after it has been visited. So when we retrieve the UAV at a customer node, not only the demand information at the retrieve node is revealed, the demand of the UAV node is also revealed. As a result, different scenarios arises with different demand values and remaining truck capacity. For the rest of this section, all these scenarios will be considered to calculate the expected recourse cost of a retrieve node.

Case 4: The customer node i is planned to be served by the truck and i is a retrieve node of an UAV sortie, e.g. node 6 in the previous example. Remember that in the special cases where a node i is both a launch node and a retrieve node, it should be regarded as a retrieve node when calculating the recourse cost. The recourse action happens at node i only when $X_{pre(i)} < Q$. In this case, different recourse actions are taken under different situations. For the normal case where $i \neq 0$, different scenarios arises, as shown in Figure 2:

- **Scenario 1**: $X_i < Q, X_{dr(i)} < Q$ and $dr(i)$ is NOT served successfully. The probability of this scenario happens is $P_{s1} = P\{X_i < Q, X_{dr(i)} < Q, \xi_{dr(i)} > D\}$. In this scenario, the truck just needs to serve node $dr(i)$ and going to $next(dr(i))$. The recourse cost is $C_{s3} = d(i, dr(i)) + d(dr(i), next(dr(i))) - d(i, next(dr(i)))$.

- **Scenario 2**: $X_i < Q, X_{dr(i)} = Q$ and $dr(i)$ is served successfully. The probability of this scenario happens is $P_{s2} = P\{X_i < Q, X_{dr(i)} = Q, \xi_{dr(i)} \leq D\}$. In this scenario, the truck needs to return to depot and goes to $next(dr(i))$. The recourse cost is $C_{s5} = d(i, 0) + d(0, next(dr(i))) - d(i, next(dr(i)))$.

- **Scenario 3**: $X_i < Q, X_{dr(i)} = Q$ and $dr(i)$ is NOT served successfully. The probability of this scenario happens is $P_{s3} = P\{X_i < Q, X_{dr(i)} = Q, \xi_{dr(i)} > D\}$. In this scenario, the truck needs to serve node $dr(i)$, return to depot and go to $next(dr(i))$. The recourse cost is $C_{s4} = d(i,dr(i)) + d(dr(i),0) + d(0,next(dr(i))) - d(i,next(dr(i))$.

- **Scenario 4**: $X_i < Q, X_{dr(i)} > Q$ and $dr(i)$ is served successfully. The probability of this scenario happens is $P_{s4} = P\{X_i < Q, X_{dr(i)} > Q, \xi_{dr(i)} \leq D\}$. In this scenario, the truck needs to return to depot and goes to $next(dr(i))$. The recourse cost is $C_{s7} = d(i,0) + d(0,next(dr(i))) - d(i,next(dr(i))$.

- **Scenario 5**: $X_i < Q, X_{dr(i)} > Q$ and $dr(i)$ is NOT served successfully. The probability of this scenario happens is $P_{s5} = P\{X_i < Q, X_{dr(i)} > Q, \xi_{dr(i)} > D\}$. In this scenario, the truck needs to return to depot, serves $dr(i)$ and goes to $next(dr(i))$. The recourse cost is $C_{s6} = d(i,0) + d(0,dr(i)) + d(dr(i),next(dr(i))) - d(i,next(dr(i))$.

- **Scenario 6**: $X_i = Q$ and $dr(i)$ is served successfully. The probability of this scenario happens is $P_{s6} = P\{X_i = Q, \xi_{dr(i)} \leq D\}$. In this scenario, the truck needs to return to the depot first before going back to node $next(dr(i))$ in r' and the recourse cost is $C_{s1} = d(i,0) + d(0,next(dr(i))) - d(i,next(dr(i))$.

- **Scenario 7**: $X_i = Q$ and $dr(i)$ is NOT served successfully. The probability of this scenario happens is $P_{s7} = P\{X_i = Q, \xi_{dr(i)} > D\}$. In this scenario, the truck needs to return to the depot first before serving node $dr(i)$ and going to $next(dr(i))$. The recourse cost is $C_{s2} = d(i,0) + d(0,dr(i)) + d(dr(i),next(dr(i))) - d(i,next(dr(i))$.

- **Scenario 8**: $X_i > Q$ and $dr(i)$ is served successfully. The probability of this scenario happens is $P_{s8} = P\{X_i > Q, \xi_{dr(i)} \leq D\}$. In this scenario, the truck needs to return to depot and go to $next(dr(i))$. The recourse cost is $C_{s9} = d(i,0) + d(0,next(dr(i))) - d(i,next(dr(i))$.

- **Scenario 9**: $X_i > Q$ and $dr(i)$ is NOT served successfully. The probability of this scenario happens is $P_{s9} = P\{X_i > Q, \xi_{dr(i)} > D\}$. In this scenario, the truck needs to return to depot, goes back to i, serves $dr(i)$ and goes to $next(dr(i))$. The recourse cost is $C_{s8} = 2d(i,0) + d(i,dr(i)) + d(dr(i),next(dr(i))) - d(i,next(dr(i))$.

So the expected recourse cost at node i is

$$E[C(i)] = P\{X_{pre(i)} < Q\} \sum_{j=0}^{9} P_{sj} C_{sj} \tag{7}$$

where P_{sj}, C_{sj} are shown in each scenario j.

For the special case where the retrieve node i is the depot, recourse only happens when $dr(i)$ is NOT served successfully and the truck needs to serve $dr(i)$ and returns to depot. The recourse cost is $C_6 = 2d(0,dr(i))$. So the expected recourse cost is

$$E[C(i)] = P\{X_i > Q\} C_6 \tag{8}$$

To summarize this section, the expected cost of recourse of a route is simply the sum of the expected recourse cost of all the customers it aims to serve. As a result, we get the following proposition:

Proposition 2. *The expected recourse cost of a cooperated route r is*

$$E[C(r)] = \sum_{i=0}^{n_r} E[C(i)] \tag{9}$$

With Proposition 2, the total route cost of a solution can be calculated using equation (9).

SOLUTION METHOD

In this section, an adaptive large neighborhood search framework is proposed to solve CVRPDSD.

General structure

The concept of large neighborhood search is brought up by (*28*). The general framework of the algorithm used in this research is similar to that proposed in (*22*) and (*23*). Typically in LNS, there are sets of destroy and repair methods available. "Destroy" methods eliminate a certain number of nodes from a solution, while "repair" methods generate a new feasible solution given the partial solution and eliminated nodes from previous step.

There are two main approaches for destroy and repair methods. The first is to simply choose them randomly, while the second one chooses them adaptively. In ALNS, the selection of method is based on the roulette wheel selection rule.

Let A be the set contains of all available destroy methods, and B the set of repair methods. At the start of iteration j, if the performance score of method $i \in A \cup B$ is denoted as w_{ij}, then the probability of method i been chosen is

$$p_{ij} = \frac{w_{ij}}{\sum_i w_{ij}} \tag{10}$$

After each iteration, w_{ij} is updated based on performance in the iteration as follows:

$$w_{i,j+1} = \rho w_{ij} + \frac{\lambda}{\tau}(1-\rho) \tag{11}$$

In the formula, ρ is the score shrinking rate and λ/τ is the score for the method in iteration j, where λ is an indication of the fitness of the newly found solution and τ is an indication of the computation time of the current iteration. Intuitively, the value of λ/τ is higher if we find a better solution with lower cost, or we entered a search space that is not visited before, in a short amount of time. In this study, a random number between 1 and the maximum destroy number m_j is generated as the target number of nodes that would be removed.

To control the evolution of the solution with regard to its sensitivity to the variations of the search space, we control the acceptance probability of a new solution with a global time-varying parameter T, called the temperature. This concept is frequently used in simulated annealing (*13*). In this study, if a better solution is found in a iteration, the new solution is always accepted. If the new solution x^t has a higher cost, the probability of it being accepted is

$$p(x^t) = e^{\frac{f(x)-f(x^t)}{T}} \tag{12}$$

where $f(x)$ denotes the objective value of solution x.

As the solution space is explored, the value of T gradually decreases from the initial value to zero. Given a pre-specified time limit for search phase, the value of T is updated as

$$T = T_0\left(1 - \frac{t}{t_m}\right) \tag{13}$$

where t is the elapsed search time and t_m is the pre-set time limit. The search terminates when the value of T is sufficiently small.

The pseudo-code for the ALNS is given in Algorithm 1.

Algorithm 1 ALNS

Input: Instance and preset parameters
Output: CVRPDSD solution

$s \leftarrow InitialSolution()$
$s^* \leftarrow s$
$noImpro \leftarrow 0$
while $t \leq t_m$ **do**
    Choose a destroy method d and a repair method r
    $s \leftarrow r(d(s^*))$
    $T = T_0(1 - t/t_m)$
    if $random(0,1) < exp(\frac{f(x)-f(x^t)}{T})$ **then**
        $s \leftarrow s^t$
    end if
    if $f(s) < f(s^*)$ **then**
        $s^* \leftarrow s$
        $noImpro \leftarrow 0$
    else
        $noImpro \leftarrow noImpro + 1$
        if $noImpro > noImproMax$ **then**
            $s \leftarrow s^*$
            $noImpro \leftarrow 0$
        end if
    end if
    Update performance score
end while
return s^*

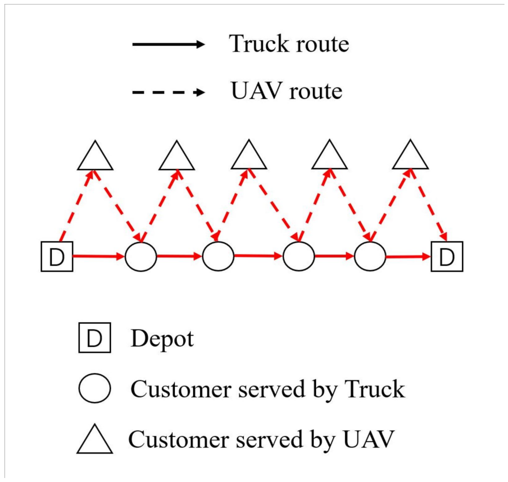

FIGURE 3 Illustration of case when all UAV sorties has no intermediate nodes

Initial solution

In this research, the initial solution is generated using the nearest neighbor method from classical VRP.

Removal methods

In each iteration, the destroy operator removes customer nodes from the current solution and stores them in a set P. In this study, a random generated integer β that lies between 1 and $\lfloor (n-1-m)/2 \rfloor$ is taken as the target number of customer nodes to be removed from the current coordinated route. This process is repeated for each route. After the destroy process, a partial solution s^p and node set P are obtained.

In our study, three different destroy methods are available for selection at each iteration:

Random removal: This method selects customer nodes at random for removal, from the coordinated routes of the current solution.

Nearby removal: In this method, a random customer node is chosen and removed from the current routes. The method then recursively removes the customer node closest to the last customer node removed, until a target number of customers have been selected.

Route removal: This method aims to remove a whole coordinated route from the current solution.

Repair Methods

In each iteration of ALNS, with given partial solution s^p and picked node set P, the repair method rebuilds the partial solution and returns a new feasible solution s. The three repair methods are:

Greedy repair: This is the "cheapest" method of all three repair methods. The detailed process is shown in Algorithm 2.

Nearby repair: This method is more costly than the greedy one. It examines all combinations of launch node and retrieve node in that feasible segment and picks the best one. The detailed process is shown in Algorithm 3.

CP repair: Constraint programming (CP) repair is the most costly methods. In contrast with the first two methods, which maintained existing sorties in the partial route while searching for new ones, CP repair method begins by eliminating all current sorties. Ad-

Algorithm 2 Greedy repair

Input: partial solution s^p and picked node set P

Output: an improved feasible solution of s^p

```
for node i in P do
    Find the insert location with least extra cost among all the insertion locations among all the coordinated routes
    If after insertion the resulting coordinated route is feasible, adopt the move. Else, check next insert location.
end for
for truck route r^T in s^p do
    for node i in r^T do
        if i could be served by UAV with sortie s then
            Add s to the current route
            Delete i from the turck's route
        end if
    end for
end for
return s^p
```

Algorithm 3 Nearby repair

Input: partial solution s^p and picked node set P

Output: an improved feasible solution of s^p

```
for node i in P do
    Insert node into s^p before a customer node that is closest to the chosen node
end for
for truck route r^T in s^p do
    for feasible segment f in r^T do
        for node i in f do
            Examine every feasible sortie that serves i by UAV
            Select sortie s with the greatest saving or with lowest traverse node
            If the resulting route is feasible, add sortie s into s^p
            Break f into two remaining smaller segment and add them into segment set
        end for
    end for
end for
return s^p
```

ditional nodes are chosen randomly from the truck's route, and a constraint programming solver is used to find an optimal way to serve them. This sub-problem is formulated as a scheduling problem in CP and solved by a commercial solver CPLEX. The detailed process of this method is shown in Algorithm 4.

Algorithm 4 CP repair

Input: partial solution s^p and picked node set P
Output: an improved feasible solution of s^p

```
for node i in P do
    Insert node into s^p before a customer node that is closest to the chosen node
end for
for coordinated route r_i in s^p do
    p' ← null set
    Delete all UAV sorties in r_i^D, add UAV nodes into p'
    Randomly select truck nodes in r_i^T and add them into p'
    r_i^{T'} ← remaining truck route
    Solve the CP model or MIP model with r_i^{T'} and p'
end for
return s^p
```

NUMERICAL ANALYSIS

In this section, all the tests are run on a desktop machine with Intel Processor i7-9700K at 3.60GHz with 32 GB RAM.

Experiments on VRPLIB instances

Publicly available VRPLIB instances are chosen to test the proposed ALNS. In this section, all the tested instances are originally created in (*1*) as instances of classical CVRP. Thus, in the case of CVRPDSD, additional parameters and necessary modifications are needed to fit our purpose, which are described and summarized below:

- Rather than being deterministic, customers' demand was made Poisson, with the original deterministic value as mean.
- The truck's capacity is unchanged.
- The UAV's capacity is chosen as the average of the maximum and median value of all the customers' demand, to create interesting (and more challenging) instances.
- The UAV's travel time between two nodes is half of that of the truck, recognizing that its flight is unaffected by congestion or traffic control.
- The UAV's flight range is 50% more than the greatest distance between any two nodes. This means that there are some node pairs that cannot be served by a UAV.

The other parameters are unchanged from the classical VRP.

Experimental Setting

The algorithm parameters are set as follows: initial temperature $T = 1000$, non-improvement parameter *noImprovMax* $= 100$, score shrinking rate $\rho = 0.9$. The initial performance score w_{i0} for each destroy and repair method is set to 100. The time limit of the algorithm is set to be 200 seconds for small cases, and 400 or 600 seconds for larger instances.

TABLE 1 Results of modified CVRPDSD instances

Instance	$\lvert C\rvert$	$\lvert V\rvert$	z_T^*	z_f^*	z_r^*	$Cost_T$	$Cost_f$	$Cost_r$	$\lvert V'\rvert$	$Time$
A-n32-k5	32.0	5.0	918.9	787.8	131.1	802.4	610.9	191.5	3.0	400.0
A-n33-k5	33.0	5.0	760.0	662.7	97.2	669.7	523.8	145.9	4.0	400.0
A-n33-k6	33.0	6.0	843.6	742.8	100.8	507.1	471.1	36.0	4.0	400.0
A-n34-k5	34.0	5.0	905.1	781.3	123.8	748.3	569.3	179.0	3.6	400.0
A-n36-k5	36.0	5.0	947.4	802.1	145.3	911.9	574.5	337.4	3.0	400.0
B-n31-k5	31.0	5.0	785.5	676.7	108.8	629.5	490.6	138.9	3.0	400.0
B-n34-k5	34.0	5.0	963.1	791.2	171.9	612.3	575.3	36.9	3.4	400.0
B-n35-k5	35.0	5.0	1223.4	956.3	267.1	1036.7	677.2	359.5	3.0	400.0
B-n38-k6	38.0	6.0	965.7	809.4	156.3	795.2	591.8	203.4	4.0	400.0
P-n16-k8	16.0	8.0	515.6	451.9	63.7	258.9	227.8	31.0	5.0	200.0
P-n19-k2	19.0	2.0	231.0	212.0	18.4	177.0	163.8	13.2	2.0	200.0
P-n20-k2	20.0	2.0	243.0	216.0	27.0	200.2	178.6	21.6	2.0	200.0
P-n21-k2	21.0	2.0	224.8	212.0	12.0	231.1	198.7	32.4	2.0	200.0
P-n22-k2	22.0	2.0	243.4	216.0	25.5	214.7	195.0	19.6	2.0	200.0
P-n22-k8	22.0	8.0	644.4	603.0	43.0	313.2	313.0	0.2	4.6	200.0
P-n23-k8	23.0	8.0	690.0	529.0	158.8	347.5	306.8	40.7	6.0	200.0
P-n40-k5	40.0	5.0	482.3	461.7	20.6	464.2	371.3	92.8	3.8	400.0
P-n45-k5	45.0	5.0	575.6	512.8	62.8	557.9	461.9	95.9	4.0	400.0
P-n50-k7	50.0	7.0	623.6	559.9	63.7	556.3	404.8	151.5	4.8	600.0
P-n50-k8	50.0	8.0	748.5	634.8	113.6	583.9	416.6	167.3	5.4	600.0
Average	**31.70**	**5.20**	**676.75**	**580.97**	**95.57**	**530.90**	**416.14**	**114.74**	**3.63**	**350.0**

Computational Results

The computational results are shown in Table 1. In the table, $|C|$ and $|V|$ represent the number of nodes and vehicles in the instance, respectively. z_T^*, z_f^* and z_r^* are the route total cost, route fixed cost and route expected recourse cost of the classical VRP's optimal solution, respectively. So z^* equals the minimal route cost of CVRP if the demands are deterministic. $Cost$, $Cost_f$, $Cost_r$ represent the route total cost, route planned cost and route expected recourse cost, respectively found by the proposed algorithm. $|V'|$ represents the number of vehicles in the final LNS solution. $Time$ is the pre-set computational time. For each instance the reported results are the average value of at least 5 independent runs of the algorithm.

As can be seen from Table 1, on average, the CVRPDSD solution has a lower total route cost (553.67) than CVRP optimal solution (676.75). In some instances (such as P-n16-k8, P-n22-k8 and p-n23-k8) where the number of available vehicles is high, the CVRPDSD solution has much lower total route cost than the CVRP optimal solution. There are several reasons for this result. First, in these cases the truck's capacity is usually relatively low and the truck needs to return to depot frequently. However, the hard capacity constraint in classical VRP no longer applies in CVRPDSD. A route that used to be infeasible in CVRP is now considered to be feasible as long as it does not violate the "one return trip assumption." As a result, the number of vehicles used in the final solution is less than CVRP, especially on the above-mentioned cases. Second, the addition of UAVs greatly improves delivery efficiency, as they have a faster travel speed than the truck. This is reflected by the CVRPDSD solution fixed route cost (423.42) being lower than that of the CVRP optimal solution (580.97).

Sensitivity analysis on UAV capacity

In this section, the same experiment setting is adopted as in previous section, but allowing the UAV capacity to vary around the default value (the average of the maximum and median of expected demand of all the customers). In a low capacity setting, this value is set to be the median value of all customers' expected demand while in high capacity setting this value is the maximum value of all customers' demand. The results of modified VRPLIB instances are shown in Table 2.

TABLE 2 Route costs comparison under different drone capacity settings

	Low capacity			Default setting			High capacity		
Instance	$Cost_T$	$Cost_f$	$Cost_r$	$Cost_T$	$Cost_f$	$Cost_r$	$Cost_T$	$Cost_f$	$Cost_r$
A-n32-k5	948.8	657.7	291.1	802.4	610.9	191.5	749.3	612.1	137.2
A-n33-k5	760.4	542.9	217.5	669.7	523.8	145.9	584.5	456.7	127.8
A-n33-k6	812.0	556.5	255.5	507.1	471.1	36.0	502.6	460.7	41.9
A-n34-k5	880.9	627.4	253.5	748.3	569.3	179.0	700.3	523.6	176.7
A-n36-k5	1039.8	730.0	309.8	911.9	574.5	337.4	791.6	552.1	239.5
B-n31-k5	739.4	504.2	235.3	629.5	490.6	138.9	568.3	470.2	98.1
B-n34-k5	806.4	584.5	221.9	612.3	575.3	36.9	595.6	538.8	56.8
B-n35-k5	1127.8	691.8	435.9	1036.7	677.2	359.5	834.3	665.6	168.7
B-n38-k6	937.8	659.9	277.9	795.2	591.8	203.4	655.1	556.6	98.5
P-n16-k8	339.4	265.5	73.9	258.9	227.8	31.0	226.7	220.3	6.4
P-n19-k2	237.4	209.5	27.9	177.0	163.8	13.2	166.6	160.3	6.3
P-n20-k2	251.6	227.0	24.6	200.2	178.6	21.6	176.5	170.6	5.9
P-n21-k2	252.0	214.6	37.4	231.1	198.7	32.4	197.4	188.4	9.0
P-n22-k2	266.5	227.7	38.8	214.7	195.0	19.6	193.0	183.2	9.8
P-n22-k8	434.4	334.9	99.5	313.2	313.0	0.2	305.9	304.9	1.0
P-n23-k8	485.5	343.7	141.7	347.5	306.8	40.7	310.4	293.2	17.2
P-n40-k5	619.5	448.7	170.9	464.2	371.3	92.8	439.7	347.0	92.7
P-n45-k5	749.3	515.0	234.3	557.9	461.9	95.9	459.8	384.5	75.3
P-n50-k7	713.6	522.0	191.6	556.3	404.8	151.5	515.1	373.3	141.8
P-n50-k8	789.0	567.3	221.7	583.9	416.6	167.3	510.1	419.4	90.7
Average	**659.58**	**471.54**	**188.04**	**530.90**	**416.14**	**114.74**	**474.14**	**394.07**	**80.06**

As can be seen in Table 2, the experiment setting with the lowest UAV capacity has the highest overall route cost, which decreases gradually as the UAV capacity increases. Especially in instances P-n16-k8, P-n22-k8, P-n23-k8 and P-n50-k8, where the number of available vehicles is 8, the total overall route cost $Cost_T$ with high UAV capacity is only about 62% of that with low UAV capacity. This results indicate that the UAV capacity has a significant effect on the total overall route cost in CVRPDSD, as low UAV capacity leads to high probability of drone route failure and high recourse cost. As illustrated in Table 2, the recourse cost in capacity setting 2 is only half of that in capacity setting 1.

CONCLUSION

In this paper, we investigate the capacitated vehicle routing problem with drones and stochastic demand. CVRPDSD is a pick-up only problem where it aims to find an *a priori* route minimizing the sum of fixed route cost and expected recourse cost. We propose a new recourse strategy which is modified from the classical VRP recourse strategy. A key difference between the new strategy and the classical one is that the truck cannot returns to the depot without the drones. Additionally, the truck can only returns to the depot at most one time during its route but it can diverts to the customers which are not served by the UAV for multiple times. A closed-form mathematical formulation is proposed to calculate the total overall route cost. A large neighborhood search method which integrates constraint programming modelling is presented to solve CVRPDSD. Numerical analysis are conducted on public available VRPLIB datasets, which shows that the addition of UAV can greatly reduce the total overall cost. Additional sensitivity analyses further indicate that UAV capacity has a significant effect on the total cost.

The future research direction concerning CVRPDSD includes variant which enables multiple return trips or that enables the truck to carry multiple drones.

ACKNOWLEDGEMENTS

This work is supported by the Center for Advanced Multimodal Mobility Solutions and Education (CAMMSE) Tier 1 University Transportation Center, and the National Science Foundation (CMMI-1562291, CMMI-1636154, CMMI-1826320).

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: study conception and design: all authors; data collection: Zhu; analysis and interpretation of results: Zhu; draft manuscript preparation: all authors. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

Philippe Augerat. *Approche polyèdrale du problème de tournées de véhicules*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 1995.

Dimitris J Bertsimas. A vehicle routing problem with stochastic demand. *Operations Research*, 40(3):574–585, 1992.

Paul Bouman, Niels Agatz, and Marie Schmidt. Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, 72(4):528–542, 2018.

ccinsight.org. Covid-19 commerce insight, 2020. URL `https://ccinsight.org/`.

Yong Sik Chang and Hyun Jung Lee. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. *Expert Systems with Applications*, 104:307–317, 2018.

Geoff Clarke and John W Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.

Júlia Cária de Freitas and Puca Huachi Vaz Penna. A randomized variable neighborhood descent heuristic to solve the flying sidekick traveling salesman problem. *Electronic Notes in Discrete Mathematics*, 66:95–102, 2018.

Júlia Cária de Freitas and Puca Huachi Vaz Penna. A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research*, 27 (1):267–290, 2020.

Thai Dinh, Ricardo Fukasawa, and James Luedtke. Exact algorithms for the chance-constrained vehicle routing problem. *Mathematical Programming*, 172(1-2):105–138, 2018.

droneii. The drone delivery report 2019-2024, 2020. URL `https://www.droneii.com/project/the-drone-delivery-report-2019`.

Moshe Dror, Gilbert Laporte, and Francois V Louveaux. Vehicle routing with stochastic demands and restricted failures. *Zeitschrift für Operations Research*, 37(3):273–283, 1993.

Curt Hjorring and John Holt. New optimality cuts for a single-vehicle stochastic routing problem. *Annals of Operations Research*, 86:569–584, 1999.

Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

Gilbert Laporte and François V Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

Gilbert Laporte, Roberto Musmanno, and Francesca Vocaturo. An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands. *Transportation Science*, 44(1):125–135, 2010.

Hongtao Lei, Gilbert Laporte, and Bo Guo. The capacitated vehicle routing problem with stochastic demands and time windows. *Computers & Operations Research*, 38(12):1775–1783, 2011.

Chase C. Murray and Amanda G. Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 2015. ISSN 0968090X. doi: 10.1016/j.trc.2015.03.005.

Mahdi Noorizadegan and Bo Chen. Vehicle routing with probabilistic capacity constraints. *European Journal of Operational Research*, 270(2):544–555, 2018.

Stefan Poikonen, Bruce Golden, and Edward A Wasil. A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS Journal on Computing*, 31(2):335–346, 2019.

QuantumMetric. Covid-19 online sales impact data, 2020. URL `https://www.quantummetric.com/covid-19-online-sales-impact/`.

Walter Rei, Michel Gendreau, and Patrick Soriano. A hybrid monte carlo local branching algorithm for the single vehicle routing problem with stochastic demands. *Transportation Science*, 44(1): 136–146, 2010.

Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472, 2006.

David Sacramento, David Pisinger, and Stefan Ropke. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, 102:289–315, 2019.

Majid Salavati-Khoshghalb, Michel Gendreau, Ola Jabali, and Walter Rei. An exact algorithm to solve the vehicle routing problem with stochastic demands under an optimal restocking policy. *European Journal of Operational Research*, 273(1):175–189, 2019a.

Majid Salavati-Khoshghalb, Michel Gendreau, Ola Jabali, and Walter Rei. A rule-based recourse for the vehicle routing problem with stochastic demands. *Transportation Science*, 53(5):1334–1353, 2019b.

Daniel Schermer, Mahdi Moeini, and Oliver Wendt. A hybrid vns/tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers & Operations Research*, 109:134–158, 2019a.

Daniel Schermer, Mahdi Moeini, and Oliver Wendt. A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C: Emerging Technologies*, 106:166–204, 2019b.

Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. pages 417–431, 1998.

William R Stewart Jr and Bruce L Golden. Stochastic vehicle routing: A comprehensive approach. *European Journal of Operational Research*, 14(4):371–385, 1983.

Kaarthik Sundar and Sivakumar Rathinam. Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. *IEEE Transactions on Automation Science and Engineering*, 11(1):287–294, 2013.

Frank A Tillman. The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, 3(3):192–204, 1969.

Zheng Wang and Jiuh-Biing Sheu. Vehicle routing problem with drones. *Transportation research part B: methodological*, 122:350–364, 2019.

James R Yee and Bruce L Golden. A note on determining operating strategies for probabilistic vehicle routing. *Naval Research Logistics Quarterly*, 27(1):159–163, 1980.

Emine Es Yurek and H Cenk Ozmutlu. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, 91:249–262, 2018.